

# **Open Sales and Distribution Model OSDM**

*This IRS is to be published only in English*

## **International Railway Solution to be classified in volumes of UIC 9**

- Information, Technology, Miscellaneous

### **Application:**

With effect from 1st April 2020

All members of the International Union of Railways

### **Record of updates:**

**2020 March** Version 1.0 Fares only

**2020 December** Version 1.0 OSDM

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Printed by the International Union of Railways (UIC)

ISBN 978-2-xxxxxx



## **The International Railway Solution**

The International Railway Solutions (IRS) are structured in a General Part and in some eventual Application Parts.

The General Part is valid worldwide, while the Application Parts are valid for a specific railway application, based on a geographical or on a service implementation.

The eventual Application Parts may thus be added according to the current needs of the Railway Community.

Structure of the International Railway Solution:

**IRS 90918-10: Open Sales and Distribution Model OSDM**

**General Part**

# Content

General Part .....	3
List of figures: .....	10
<b>Foreword.....</b>	<b>12</b>
Current Situation.....	12
Offline Distribution.....	12
Online Distribution .....	12
Objectives .....	12
Objectives for Offline Distribution .....	12
Objectives for Online Distribution.....	12
Summary .....	13
Normative References.....	14
UIC Leaflets.....	14
International Rail Standards.....	14
International Standards.....	14
Terms and Definitions .....	16
<b>Acronyms .....</b>	<b>18</b>
Requirements .....	20
Common Functional Requirements.....	20
Requirements on product range .....	20
Requirements on price .....	20
Requirements on personal data .....	20
Functional Requirements Distribution .....	20
Requirements on passenger.....	20
Requirements on location .....	20
Requirements on trip .....	21
Requirements on offers.....	21
Requirements on round trips .....	22
Requirements on pre-booking .....	22
Requirements on booking .....	22
Requirements on products.....	23
Requirements on fulfillment .....	23
Requirements on refund .....	23
Requirements on partial refund or exchange .....	23
Functional Requirements Allocation .....	24

Requirements on regional validity .....	24
Requirements on allowed service .....	26
Requirements on validity for usage.....	29
Requirements on validity for passengers / transportables .....	30
Requirements on validity for reductions.....	31
Requirements on prices.....	31
Requirements on the basic fare structure .....	32
Requirements on the after sales conditions .....	32
Requirements on conditions on fulfilment .....	33
Requirements on dynamic fares and train linked tickets.....	33
Requirements on combining fares .....	34
Requirements on reservation.....	38
Architectural Requirements .....	38
Requirements on aligned processes end to end .....	38
Requirements on aligned services.....	38
Requirements on messages.....	38
Requirements on extendibility .....	38
Requirements on security .....	38
Legal Requirements.....	38
Regulations .....	38
Requirements not in Scope .....	40
Actors / Business Capabilities.....	41
Actors .....	42
Common Business Capabilities.....	44
Powerful fare combination.....	44
Simple distribution .....	44
Business Capabilities for Distribution.....	44
Lookup location .....	44
Search trips.....	44
Find offers.....	44
Pre-book offers.....	44
Book pre-booked booking .....	44
Fulfill booking .....	44
Get booking .....	45
Refund booking .....	45

Exchange booking.....	45
Graphical seat reservation .....	45
Edit passenger information .....	45
Retrieve product information .....	45
Retrieve stored personal data.....	45
<b>Business Capabilities for Fare Allocation.....</b>	<b>45</b>
Combine fares .....	45
Service resource location / locate dynamic fares .....	45
Provide bulk fare data .....	46
Provide dynamic fare.....	46
Book offer.....	46
Fulfillment.....	46
Reservation.....	46
Get booking .....	46
Refund booking .....	46
Exchange booking.....	47
Accounting.....	47
Graphical seat reservation .....	48
Edit passenger information .....	49
Retrieve stored personal data.....	49
<b>Common Data Structures in Offline and Online Mode .....</b>	<b>49</b>
General.....	49
Versioning.....	49
Indication of personal data .....	49
Indication of required data.....	49
Detailed data structures.....	50
AfterSalesRules.....	50
Calendar .....	53
CarrierConstraint.....	55
ConnectionPoint.....	56
Fare.....	60
FareCombinationConstraint .....	62
FareResourceLocation .....	66
FareReferenceStationSet.....	79
FulfilmentConstraint.....	80

Line .....	81
PassengerConstraint.....	82
PersonalDataConstraint .....	84
See code list: LanguageCode .....	84
OverruleCode .....	<b>Fehler! Textmarke nicht definiert.</b>
Price.....	87
ReductionCard.....	89
ReductionConstraint.....	91
RegionalConstraint .....	92
ReservationParameter.....	98
StationDetail.....	101
StationNames .....	102
Text.....	103
SalesAvailability .....	104
ServiceClass .....	106
ServiceConstraint .....	107
ServiceLevel.....	108
TravelValidityConstraint .....	109
ZoneDefinition.....	112
Offline Sale and Distribution .....	114
General Scenarios.....	114
Fare exchange.....	114
Versioning of data deliveries .....	114
Automated bulk data exchange .....	114
Asynchronous fare data delivery.....	115
Online Sale and Distribution.....	118
Roles: Fare Provider, Allocator & Distributor.....	118
General scenarios.....	118
Fare Provider Role .....	118
Allocator Role .....	121
Distributor Role .....	123
Data Structures - Online Models .....	126
Trips and Locations Model .....	127
Offers Model.....	134
Booking Model .....	146

State Models.....	148
Service Specification.....	150
Service Overview .....	150
Process Flow .....	151
Trips and Locations Processes .....	152
Offers Processes .....	155
Round trip handling.....	156
Booking Processes .....	163
Confirmation and Fulfilment Processes .....	168
After-Sales Processes .....	171
Example End-to-end Interaction .....	173
Service Level Requirements – Allocator.....	174
Service Level Requirements – Carrier/Fare-Provider .....	174
Code lists .....	175
AccommodationType .....	175
AccommodationSubType .....	175
BarcodeType.....	176
CardType.....	176
ControlDataExchangeType .....	176
Currency .....	176
FareType .....	177
ControlSecurityType .....	177
Gender.....	177
Graphics Icons .....	178
Interface Type.....	186
LanguageCode .....	187
OverruleCode .....	187
Personal data items.....	187
Personal data transfer types .....	187
Personal data change reasons.....	187
Preferences of places .....	188
Preference groups.....	189
Properties of places.....	189
Reduction “cards”.....	191
ServiceLevelType .....	192

Sales Channel Type .....	193
Service Brand Codes .....	193
Service Class / Classic Class .....	193
Station codes .....	194
SupportedOnlineServices .....	194
TaxScope .....	194
TimeReference .....	195
TimeUnit .....	195
TransactionType .....	195
TransportMode .....	195
TransferTypes .....	196
PassengerTypes .....	196
Compliance .....	197
Compliance with the Offline Part .....	197
Compliance with the Online Part .....	198
Compliance as an Allocator .....	198
Compliance as a Fare Provider .....	198

## List of figures:

Figure 1 Actor model .....	41
Figure 2 legacy accounting id in the fare element .....	48
Figure 3 after sales rules data structure.....	52
Figure 4 Calendar data structure.....	54
Figure 5 carrier constraint data structure (offline) .....	55
Figure 6 connection points - simple case 1 .....	57
Figure 7 connection points - complex case 2 .....	58
Figure 8 ConnectionPoint data structure offline.....	58
Figure 9 Fare element data structure (offline).....	61
Figure 10 fare combination constraint data structure .....	62
Figure 11 fare combination constraint data structure .....	63
Figure 12 FareResourceLocation data structure .....	66
Figure 13 FareResourceLocation data structure - carrier link.....	67
Figure 14 FareResourceLocation data structure - train link.....	67
Figure 15 FareResourceLocation data structure - station link .....	68
Figure 16 FareResourceLocation data structure - online resource.....	69
Figure 17 FareReferenceStationSet Definition data structure .....	79
Figure 18 Fulfilmentconstraint data structure .....	80
Figure 19 Line data structure .....	82
Figure 20 PassengerConstraint data structure offline .....	83
Figure 21 Required Personal data structure .....	85
Figure 22 allowed changes on personal data.....	85
Figure 23 cross border conditions for personal data .....	86
Figure 24 Price data structure with a list of currency prices.....	87
Figure 25 Price data structure of one currency.....	87
Figure 26 VAT details within a price .....	88
Figure 27 reduction cards offline .....	90
Figure 28 ReductionCardConstraint data structure offline.....	91
Figure 29 ReductionCardReference data structure.....	92
Figure 30 Connection points and timetable routes .....	93
Figure 31 Regional validity constraint data structure .....	94
Figure 32 regional validity data structure .....	95
Figure 33 route description (ViaStation) data structure .....	96
Figure 34 RegionalValidity data structure - copy of 90918-4.....	97
Figure 35 ReservationParameter data structure.....	99
Figure 36 Reservation Parameter - support for 90918-1 reservation interface .....	100
Figure 37 ReservationParameter data structure - reservation options .....	100
Figure 38 structure Station detail data .....	101
Figure 39 structure Station detail data (offline only).....	102
Figure 40 Text data structure .....	104
Figure 41 SalesAvailability data structure .....	105
Figure 42 service constraints.....	107
Figure 43 Service level description (offline) .....	109
Figure 44 travel validity data structure .....	110
Figure 45 travel validity data structure – validity range .....	111

Figure 46 travel validity data structure – excluded time range .....	111
Figure 47 travel validity data structure- return constraint .....	112
Figure 48 Zone definition structure.....	113
Figure 49 Data structure for bulk data .....	115
Figure 50 fare structiure part 1 .....	116
Figure 51 fare structure part 2 .....	117
Figure 52 Trip data structure.....	130
Figure 53 Segment data structure.....	131
Figure 54 trip data structure - vehicle.....	132
Figure 55 Trip description - stops.....	133
Figure 56 Offer data structure.....	136
Figure 57 CommonOfferPartAttributes data structure.....	139
Figure 58 Admission data structure .....	140
Figure 59 FareOnline data structure .....	144

# Foreword

## Current Situation

### Offline Distribution

The first main goal of this specification is to address the difficulties in the fare data exchange and sales with the current data exchange formats specified in UIC leaflet 108.1 and 2. Some of the difficulties are:

- Missing access to yield managed fares
- Current data exchange is partially non-structured (word, excel, text, ...)
- Fare data are not in line with timetable data (different station codes)
- Missing solution in case of two carriers on the same line
- Data exchange once a year only
- Mapping of fares to passengers is difficult

### Online Distribution

The second main goal of this specification to address the lack of a unified interface for distribution of admissions (i.e., tickets), reservations, integrated reservation and ancillaries. Currently within the rail sector there exists a variety of very complex and incompatible interfaces to distribute rail services within Europe and beyond. Thus, international distributions demands major investments and produces high operating costs.

Furthermore, for a customer the booking of a rail ticket is unnecessarily complicated. This fact weakens rail as a convenient and ecological means of transportation.

## Objectives

The main objectives guiding this specification were:

### Objectives for Offline Distribution

- Create a new tariff model to enable the NRT-carriers/operators to offer customer-friendly and competitive prices for international travel, preferably based on timetables.
- Enable the allocating carrier/operator to offer through-tickets based on different conditions-ranges (e.g. fullflex, semiflex, non-flex, others).
- The portfolio to be offered to the customer shall be set at the decision of the issuing undertaking.
- Focus on easy possibly online-solutions to be implemented within the next two years taking into account upcoming requirements, i.e. new RICS-codes etc.
- The reservation should be fully integrated in the new technology

### Objectives for Online Distribution

- To provide a convenient way for a customer to book an international train service, including refund and exchange processes.
- To define unified process steps for offering, booking, fulfillment and after sale.
- To define unified messages that support the distributor, the allocator as well as the fare provider role.

- To provide a specification that can be supported by existing or upcoming systems without major investments to secure existing investments
- To reduce unnecessary message conversions between callers as they provide no business value

## Summary

The specification covers two aspects:

- Data exchange and sales services for rail products either to provide fare details to combine fares into offers and to provide entire offers for tickets as well. It defines the data structures to define the fares in detail and the combination rules for fares.

The specification covers static fares that can be exchanged as bulk data as well as dynamic fares and offers that need to be requested and booked online. Reservation of places is included to have a harmonized solution for the complete sales service.

A migration is supported by additional data items to cover conversion into the existing data formats 108.1 and to support existing reservation service IRS 90918-1 and accounting data formats IRS 30301.

- A set of services and unified messages to distribute rail content involving all parties in the distribution process.

We started with the customer experience and worked backwards to define the sales and distributions processes supported by OSDM. This resulted in a booking process modelled by the following steps:

- a. Searching for trips
- b. Getting offers
- c. Booking an offer
- d. Confirmation of the booking
- e. Fulfilment of the booking

Analogously, the after-sale process is modelled in the following steps:

- a. Getting a refund/exchange offers
- b. Booking a refund/exchange offer
- c. Fulfilment of the booking

By involving the experts of the parties (distributors, railways and legal experts), we are confident that the OSDM online standard is powerful enough to support the distribution of existing or upcoming commercial products and can be implemented and supported at reasonable cost.

## **Normative References**

### **UIC Leaflets**

**International Union of Railway (UIC)**

*UIC Leaflet 920-1: Standard numerical coding for railway undertakings, infrastructure managers and other companies involved in rail-transport chains*

*UIC Leaflet 920-2: Standard numerical coding of locations*

*UIC Leaflet 920-14: Standard numerical country coding for use in railway traffic*

### **International Rail Standards**

**International Union of Railway (UIC)**

*IRS 30301: Accountancy regulations for international "Passenger" traffic*

*IRS 90918-0: Electronic seat/berth reservation and electronic production of travel documents*

*IRS 90918-1: Electronic reservation of seats/berths and electronic production of travel documents - Exchange of messages*

*IRS 90918-4: e-Ticket Exchange for Control*

*IRS 90918-9: Digital Security Elements for Rail Passenger Ticketing*

### **CIT Manual for International Rail Tickets (MIRT)**

**CIT Guidelines on Protection of Privacy and Processing of Personal Data used in International Passenger Traffic by Rail (GDP CIT)**

### **International Union of Railway (UIC)/PSS**

*Implementation guide reservation systems*

### **International Standards**

#### **International Organization for Standardization (ISO)**

*ISO 3166:2006: Codes for the representation of names of countries and their subdivisions, 2006*

*ISO 4217:2001: Codes for the representation of currencies and funds, 08-2001*

#### **European Union Agency for Railways (ERA)**

*ERA TAP TSI Technical Document B.6: Electronic seat/berth reservation and electronic production of transport documents - Transport documents (RCT2 Standard)*

*ERA TAP TSI Technical Document B.7: International Rail Ticket for Home Printing*

*ERA TAP TSI Technical Document B.8: Standard numerical coding for railway undertakings, infrastructure managers and other companies involved in rail-transport chains*

*ERA TAP TSI Technical Document B.9: Standard numerical coding of locations*

**European Union (EU)**

*Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons regarding the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*

**International Civil Aviation Organization (ICAO)**

*Doc 9303 Machine Readable Travel Documents - Part 3: Specifications Common to all*

*MRTDs, 7th edition, 2015*

## Terms and Definitions

Term	Definition
<b>Tariff</b>	a schedule of <a href="#">prices</a> for the sale or rental of a product or service.  (Wikipedia)  In UIC context the term “tariff” is used for fare structure.
<b>Fare structure</b>	A <b>fare</b> is the <a href="#">fee</a> paid by a <a href="#">passenger</a> for use of a <a href="#">public transport</a> system: <a href="#">rail</a> , <a href="#">bus</a> , <a href="#">taxi</a> , etc. In the case of <a href="#">air</a> transport, the term <i>airfare</i> is often used.  Fare structure is the system set up to determine how much is to be paid by various passengers using a <a href="#">transit vehicle</a> at any given time.  (Wikipedia)
<b>e-Ticket</b>	The electronic representation of the travel contract on a data base. The home print ticket (A4RT or FST) where the contract of travel is represented in a printed or displayed bar code is not an e-ticket according to this definition. Synonym: dematerialized ticket
<b>Integrated Reservation Ticket (IRT)</b>	Ticket for a specific train on a travel day usually including the seats. All tickets for a train are managed in one central system of the <b>allocator</b> . The ticket is valid on that train on a certain day only.
<b>Non-integrated Reservation Ticket (NRT)</b>	A ticket not including an integrated reservation. Multiple <b>allocators</b> can create tickets for the same route independently. The <b>allocator</b> of the ticket is usually the same company that issues the ticket. The ticket might be applicable to a route with many trains or a zone or a list of trains or combinations of these. The validity might be more than one day. Some conditions allow a partial refund on unused parts of the ticket route. Refund can be done via the ticket vendor. These conditions depend on the fare providers and the allocator (i.e. providing the option of reducing the number of passengers or to interrupt the journey). NRTs not linked to a train might be reused in case the use is not tracked.
<b>Ticket</b>	Medium to carry the travel contract or a reference to the travel contract. The ticket might provide proof of the travel contract via its security features. The tickets are sold by <b>ticket vendors</b> <sup>1</sup> . The assembling of the ticket is done by the <b>allocator(s)</b> <sup>1</sup> . The allocator holds the master ticket data/contract of the sold ticket. A ticket can include multiple <b>fare providers</b> in the travel contract. The control of one ticket is done by one or many <b>Ticket Controlling Organizations (TCO)</b> <sup>1</sup> . Other means of ticket checking (e.g. gates) will also be named TCOs.
<b>Conditions</b>	Conditions that limit the use of a travel contract or the changes of a travel contract.
<b>Fare</b>	Proposal to purchase a transport service for specific passenger subject to specific conditions for a specific price. Handling fees by the allocator or ticket vendor are not included.

<sup>1</sup> See chapter Actors

<b>Sales fees</b>	Sales fees are fees added by the allocator or a ticket vendor to the price provided by the fare provider. Sales fees are not part of the fare provider offers defined in the scope of this document.
<b>Station fee</b>	A station fee is a fee for the use of a station by a traveller. It might be included in the fare provider offer.
<b>Station</b>	A station is a location where trains or busses regularly stop to load or unload travellers
<b>Individual ticketing</b>	A separate ticket is created per traveller.
<b>Individual contracts</b>	A separate ticket is created per traveller and these tickets can be treated as individual contracts of carriage. After sales transactions can be applied independently per traveller and ticket.
<b>Fulfilment</b>	A fulfilment is a document (either for paper printing or electronically) provided to the passenger to <ul style="list-style-type: none"> <li>• prove his travel right</li> <li>• facilitate access to trains and stations (e.g. via gates)</li> <li>• provide further information on the travel</li> <li>• provide access to services either directly or via exchange (voucher)</li> </ul>
<b>Admission</b>	The right to travel on a train, aka. as ticket.
<b>Offer Part</b>	An abstraction of things that can be offered. Can be of type Admission, Reservation or Ancillary.
<b>Ancillary</b>	An extra service like meal or WIFI

# Acronyms

Acronym	Description
<b>EWT</b>	East West tariff Tariff used for Non-Integrated-Reservation-Tickets. The fare data model follows UIC IRS 10108.1.
<b>FCB</b>	Flexible Content Bar Code Barcode specification that contains ticket data for control as structured data and is therefore machine interpretable. The Specification provides a data model of a ticket for control. FCB covers various ticket type (IRT, NRT, RPT,...). (UIC 90918-9)
<b>JWT</b>	JSON Web Token Specification to transport authentication information used by the OAUTH2 authorization protocol. <a href="#">JSON Web Token - RFC 7519</a>
<b>NRT</b>	Non-Integrated Reservation Tariff Tariff used for Non-Integrated-Reservation-Tickets. The fare data model follows UIC IRS 10108.1.
<b>NRT</b>	Non-Integrated Reservation Ticket Ticket not including an integrated reservation.
<b>REST</b>	Representational state transfer (REST) REST is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations. Other kinds of Web services, such as SOAP Web services, expose their own arbitrary sets of operations.
<b>SiP</b>	Security in Paper A ticket (representing the contract of carriage) is unique and printed on secured paper to avoid modification or creation by other than a railway company. The security is provided by the quality of the security elements included in the paper and the quality of the stock control process that controls the access to blank secure paper. (UIC IRS 90918-0)
<b>SiD</b>	Security in Data A ticket contains security elements. These are created/calculated based on the content of the ticket, resulting in a non-compliant security element in case of falsification or modification. SiD tickets are usually easy to regenerate or to copy. As SiD does not provide copy protection additional measures must be taken to avoid double use. The tickets are personalized, and the validity of the ticket is limited. (UIC IRS 90918-0)
<b>SIS</b>	Security in System The contract is on a server. Every operation (creation, check, modification, ...) on the ticket is conducted on the record(s) on the server or a synchronized replica. The access to the contract requires an authentication of the traveller. The ticket control id (key to the ticket) or personal data of the traveller (name, date of birth,) are used to retrieve the ticket. (UIC IRS 90918-0)

<b>SIV</b>	Security by Visual elements. The ticket is controlled by a visual element printed with the ticket data on blank paper or displayed on a device. To use an image as a security feature the costs to create the complex image must be higher than the price of the ticket. (UIC IRS 90918-0)
<b>TCO</b>	Ticket Controlling Organization
<b>TLT</b>	Train linked ticket Ticket not including a reservation but restricted to a train run (or multiple train runs along the route)
<b>TLB</b>	Ticket Layout Barcode Barcode specification describing the “printed” layout of a ticket. It is not machine interpretable and does not provide ticket data, only a ticket display.
<b>UML</b>	Unified Modelling Language A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems. OMG Unifier Modelling Language Superstructure v 2.1.2
<b>UUID</b>	Universally Unique Identifier Standard to create a unique id. The specification is published as ISO/IEC 9834-8:2005.

# Requirements

## Common Functional Requirements

*To do all*

### Requirements on product range

It must be possible to distribute and sell all existing products on a fare as well as on an offer basis. Existing products include admissions (a.k.a Tickets), reservations and ancillaries.

### Requirements on price

A price has a currency, an amount and a scale.

A price has a set of value added taxes. A tax is valid for a country and has amount.

### Requirements on personal data

The needed personal data must be indicated. Only personal data needed are allowed to be transferred between the involved parties.

## Functional Requirements Distribution

*To do ASC/Distributors*

### Requirements on passenger

A passenger is the person travelling on a vehicle. A passenger must not necessarily be the person who has bought the booking, i.e. the customer.

A passenger has a gender (male, female or X) and date of birth. A passenger can own reductions, most often in the form of cards.

A reduction has a type, a name and an issuer.

Passenger information has to be collected as sparsely and only if needed for a dedicated process step. Passenger details providing personal information only information required in the offer. It is not allowed to send personal information not required in the offer reply. It is not allowed to send these personal data already in the offer request.

Additional data of a passenger are title, first name, last name, phone numbers, emails or other social account.

For certain context it might be necessary to have information about the passenger's passport. This information can be added to a passenger if needed (e.g. Eurostar trains).

A passenger can have a set of reduction cards.

A passenger can further transport dogs, bicycle, car, motorcycle or trailers if this is supported by the transport vehicle.

### Requirements on location

A location uniquely identifies a place in space. A location can be of type station, point-of-interest, address or geo-coordinate.

For railway stations the UIC code station code must be supported.

To support other means of transportation the types can potentially extended.

### Requirements on trip

A trip must contain the following information.

- origin: a location where the vehicle departs
- destination: a location where the vehicle arrives
- duration: the duration of the trip
- segments: a list of segments

A segment represents a subsection of a trip that is realized with the same transport. In railways it is typically one train (between the moment passenger steps on-board until stepping out of that train) but could be using a different means of transportation. A segment has an origin, a destination and duration.

A vehicle is defined by a number or line and is defined by a service brand.

A transfer is a special kind of segment, define how long the transfer takes.

### Requirements on offers

An offer bundles admissions, reservation and ancillaries into an overall offer presented to an allocator or a distributor.

An offer has an overall comfort class and an overall flexibility.

An offer has a minimal price. The minimal price is the price that does not include optional reservations or ancillaries.

An offer is valid for a given time.

An offer can be pre-booked.

An offer should span the whole trip and include all needed services.

### Requirements on admission

An admission provides the right to travel on a vehicle.

An admission has a price. In general, the price is calculated per passenger, however for DB pricing also collective pricing must be supported.

An admission offer is valid for a given time. An admission offer shows which reductions has been applied.

An admission is linked to one or more passengers.

An admission is in state **CONFIRMED**, **FULFILLED**, **USED**, **REFUNDED**.

In general, there's a one-to-one relationship between offer and product. Only for some combinations of TGV and TER in France an offer must support referencing two products.

An admission might be linked mandatorily or optional to one or more reservations

### ***Requirements on reservation***

A reservation provides the right to sit or lay on dedicated place in a vehicle

A reservation has a price. A reservation can be **OPTIONAL**, **MANDATORY**, **INCLUDED** to an admission.

A reservation offer is valid for a given time. An admission offer shows which reductions has been applied.

A reservation is linked to one or more passengers.

A reservation is in state **CONFIRMED**, **FULFILLED**, **USED**, **REFUNDED**.

A reservation has a 1-to-1 relationship to a product.

An integrated reservation shall be modelled as an admission with an included reservation.

### ***Requirements on ancillary***

An ancillary is a service that can be offered to a customer. Example an ancillary service a Wifi access or on-board meal.

An ancillary has a price.

An ancillary offer is valid for a given time.

An ancillary is linked to one or more passengers.

An ancillary is in state **CONFIRMED**, **FULFILLED**, **USED**, **REFUNDED**.

An ancillary has a 1-to-1 relationship to a product.

### ***Requirements on fees***

Sales fees of an allocator need to be possible for reservations, admissions or auxiliaries or collectively for a set of reservations.

Requirements on fees will be treated in more details within the next version.

### ***Requirements on round trips***

Round trip offers should be possible taking into account both trips when making the offer.

Requirements on round trip will be treated in more detail in the next version.

### ***Requirements on pre-booking***

A pre-booking is an offer in the state **BOOKED**.

After a given time a pre-booking becomes invalid. At this point the allocator is required to free all resources allocated by other allocators or fare providers.

The pre-booking process must be supported by all parties.

### ***Requirements on booking***

A booking consists of a selected offer and optionally reservations or optional ancillaries.

It must be possible to search for bookings:

- Passenger first name, last name or passenger date
- Booking reference
- Fulfillment reference
- Travel date or end
- Origin or destination

## Requirements on products

A product must contain the following information:

- Id: an id uniquely identifying the product, e.g. Sparschiene
- description: A textual description of the product
- conditions: A structured description of the sales or after-sales conditions which can be machine interpreted.
- refundable: Indicates whether a product is refundable, refundable with conditions or not refundable
- exchangeable: Indicates whether a product is refundable, refundable with conditions or not exchangeable
- serviceClass: The service class describing the level of comfort.

Other attributes may define the supported fulfillment media types of the product as well as text defining service or carrier constraint.

## Requirements on fulfillment

A fulfillment must be in a well-defined state (`CONFIRMED`, `FULFILLED`, `CHECKEDIN`, `REFUNDED`) and have a unique control number. The fulfillment must reference the offer parts covered by the fulfillment.

A fulfillment must reference fulfillment documents (a.k.a tickets). Fulfillment documents in form of a UIC PDF ticket must be supported by all parties. Other media types such as `RCT2`, `RCCST`, `UIC_PDF`, `PDF_A4`, `PKPASS`, `TICKETLESS`, `ALLOCATOR_APP`, `SOCIAL_MEDIA_ACCOUNTS` can be supported. Especially the support ticketless is encouraged to be supported by all parties.

**In allocator mode only:** A fulfillment may reference fulfillment items such as visual security elements, additional bar codes or control key,

## Requirements on refund

For a given a booking a refund can be requested.

A refund can have a fee.

Cancellation (a.k.a. as revoke) is a special kind of refund where no fees apply and the complete amount is returned.

Cancellation must be supported by all parties. Total refund must be supported by all parties

## Requirements on partial refund or exchange

Partial refund is regarded as special form of exchange

Partial refund as well as exchange may be supported by all parties.

## Functional Requirements Allocation

The requirements covered by this specification are listed here with references to the implementation. Changes in the requirements during the lifecycle of this specification might lead to changes in the corresponding implementations.

### Requirements on regional validity

Users of the data:

- The allocator to link a journey from the timetable to the valid offers based on the fare data (automated)
- The passenger in a readable form to know which transport connections he can use (manual)
- A controller to validate the ticket on a train or station or station (manual and/or automated (e.g. gates))

Non-functional:

- The fares depend on the timetable. The station data are not part of the fare structure.  
Stations will always be referenced by the station code used in the timetable. Station codes used are those in the timetable according to EU TAP-TSI B.1.

Functional concepts:

#### Station

A station which could be used in timetable data to embark and/or disembark passengers.

#### Fare reference station set (virtual pricing point)

A fare reference station is a list of stations where the fare is valid with a common name.

#### Implementation: → FareReferenceStationSet

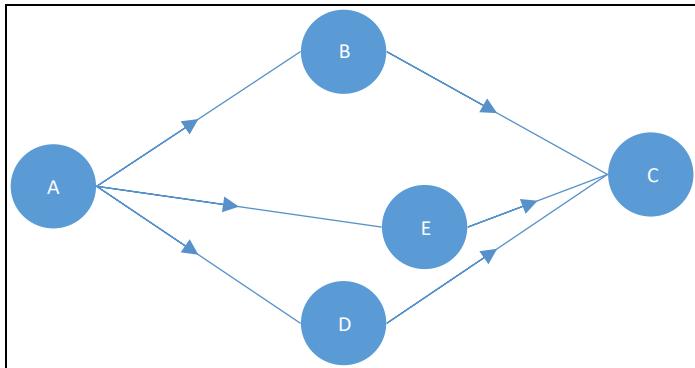
#### Route

A route is defined as an ordered list of stations or “fare reference stations” along a possible travel route. In the human readable form, the stations are separated by “\*”.

A route can split into optional routes indicated in the human readable route by “/”.

The end of a route of one carrier when combined to another route of another carrier is indicated with an additional “(FR)” in the human readable form if it is not at a “real” station.

Are routes used as line routes or as bubble routes?



A\*B/D\*C as line routes: A-E-C is not allowed

A\*B/D\*C as bubble route: A-E-C is allowed

Decision: only the line routes will be supported, "bubbles" must be defined as areas

More than two alternative routes must be possible in the route description.

#### Areas

Areas for the regional validity are needed. The areas defined in IRS 90918-4 (control) and IRS 90918-9 (bard codes) will be available for the fares as well:

- Zones
- Countries
- Geographical polygons

Route for dynamic fares:

- Train bound ticket only
  - No route
  - List of train number, travel date and time, service brand (optional), from/to station
- Route based only
  - Route same as for non-yielded NRT
  - Optionally depending on service brand(s)
- Train bound (long distance) + route (regional)
  - Yield management can be on both parts combined, so the complete part must be requested online
- Train bound (carrier 1) + Train bound (carrier 2)
  - Train bound offer until the border point on both sides
  - Train bound can be combined into one
- Train bound (carrier 1) + route (carrier 2)
  - Train bound offer until the border point

- Route description from the border point onwards

#### Connection Point:

Regions (routes) of different carriers can be connected at defined connection points. The old concept of a central predefined list of border points (as part of TAP-TSI) is replaced by the concept of "connection points" which can be defined by each carrier independently using the station codes.

Connection points will include a border point code to support existing implementations where the border point code is compared with the timetable data. As in principle every station can become a connection point (e.g. all stops from Aachen to Brussels are connection points from DB to SNCB) implementations based on border point codes cannot cover all connections.

As on both sides of a connection multiple small stations could be connected and not all of them might be in the timetable of a train the connections point should allow to connect sets of stations.

#### **Implementation: → ConnectionPoint**

#### **Requirements on allowed service**

##### Allowed services

Open tickets - not linked to a train - might be valid for some carriers or services on the route only.

As the offers should be created based on the timetable the allowed services in an offer should be defined based on the service brand code in the timetable data.

Carriers and service brands can be included or excluded.

#### **Implementation: → CarrierConstraint, ServiceConstraint**

##### Class of Service

List of classes allowed in the ticket. Railways use different notions and names on service classes on their trains. A common type is needed to combine different levels of service from different railways.

There needs to be a marketing name for the class.

Service class is optional for transportables

- Bicycle
- Dog (might depend on the class of the ticket)
- Luggage (might depend on the class of the ticket)
- Oversize Luggage (might depend on the class of the ticket)

#### **Implementation: → ServiceConstraint, ServiceClass**

IRT fares don't use classes but service levels (defined in IRS 90918-1) to cover the more detailed products available via reservation.

In case NRT and Reservation needs to be combined, rules are needed which service levels of the reservation are allowed in combination with a fare.

#### **Implementation: → SalesAvailability**

**Sales** availability defines the constraints on the time when a sale of a fare can start or end. The sales availability is used in the offline data exchange only. A constraint is provided as a list of salesRestrictions that have to be applied.

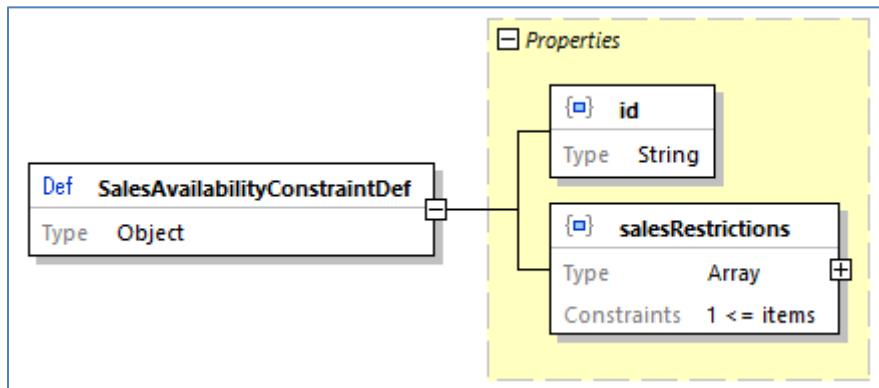


Figure 42: Sales availability structure

Sales restrictions can define a start and end of the sale relative to the date of sale or the date of travel.

A reference to a calendar can be provided to indicate all sales dates.

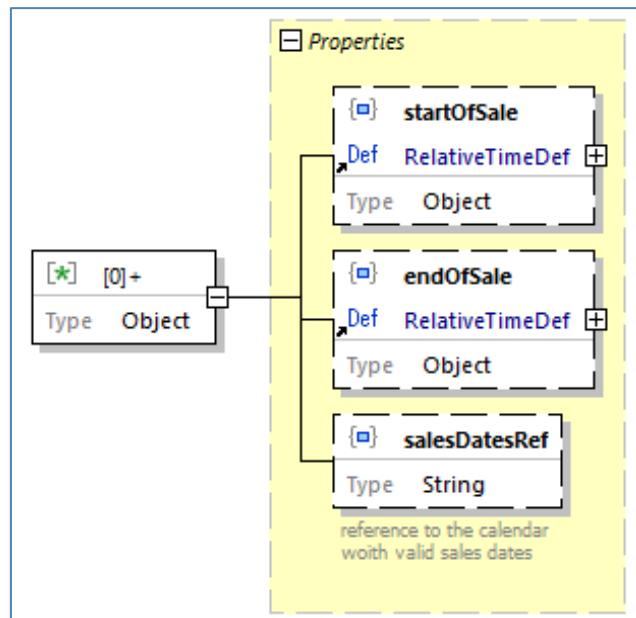


Figure 43 SalesAvailability data structure - calendar

**Data constraints**

startOfSale, endOfSale	$\text{startOfSale} < \text{endOfSale}$
------------------------	---

## ServiceClassRequirements on availability for purchase

An offer is available a specific time range before the start of travel at the first departure station in the time zone of the departure station.

An offer might become unavailable a specific time range before the start of travel at the first departure station in the time zone of the departure station.

An offer might be available from a specific time onwards or in a time range or time ranges (either in UTC or alternatively in the time zone of the ticket vendor).

Example: Offer A is available from 3 months before departure until 2 days before departure and can be purchased in June and July on Thursdays only.

### Real examples

- Available for purchase 180 to 3 days before departure day
- Available for purchase 01 JUN – 30 JUN for travels 01 JUL – 31 AUG for 30 consecutive days of validity
  - purchase 01 JUN – 30 JUN
  - travels in 01 JUL – 31 AUG → validity for usage
  - 30 days of validity → validity for usage
- Available for purchase 180 to 0 days before departure day, valid for 2 consecutive days

The following rules can be defined (and combined):

- Sales start hours or days prior to the departure in the time zone of the departure station
- Sales ends minutes, hours or days prior to the departure in the time zone of the departure station
- Sales start hours or days prior to the start of validity in the time zone of the departure station
- Sales ends minutes, hours or days prior to the start of validity in the time zone of the departure station
- Sales ends minutes, hours after the start of validity in the time zone of the departure station
  - A specific range of days in UTC
- A specific range of days in the time zone of the sales location

## Implementation: → SalesAvailability,

### Requirements on validity for usage

The validity of usage defines the time when the passenger is allowed to use a fare. To define this time there is a need to:

- Simple duration (number of days starting from the first day of validity 00:00 in the time zone of the departure station until the number of days and hours later at a specified time in the time zone of the arrival station:

Example:

Start of Validity: 1.1.2020 00:00 CET  
Validity data: 4 days 5 hours  
End of Validity: 5.1.2020 05:00 GMT

Printed text on the ticket: 1.1.2020 – 4.1.2020

- Duration as number of days and hours + number of days of the journey according to the timetable
- Exclusions (e.g. not valid during peak hours 8:00 – 10:00)
  - Peak hours in case the journey starts in the peak hours (e.g. at NS)
  - Peak hours in general
- Restrictions to specific days
  - Mondays
  - Tuesdays
  - ...
  - Sundays
  - Specific dates or date ranges
- Restriction on return tickets
  - return ticket of the same carrier must be sold
  - the number of nights in between the inbound and outbound part of a return ticket
  - a specific weekday in between the inbound and outbound part of a return ticket is not allowed

*Decision: no return tickets on one “paper” but return fares should be possible*

- Validity for passes
  - Indication that the ticket is a pass
  - Start and end of validity in UTC
  - Number of allowed trips or days
- Examples:
  - Valid Monday – Friday when work day from 09:00 until 03:00 the following day
    - → Mo – Fr without holidays
  - Valid Saturday – Sunday and public holidays from 00:00 until 03:00 the following day

**Implementation:** → [TravelValidityConstraint](#)

### **Requirements on validity for passengers / transportables**

Transportables can be different types of passengers, animals or other items carried by a passenger.

- A passenger might have an upper and / or lower age limit.
- A passenger might have an additional age limit for travelling alone.
- There might be a limit on the number of accompanying passengers of one type a passenger of another type can accompany. (e.g. not more than 8 children with one adult)
- A passenger might have an additional age limit for being entitled for reservation.

- A number of passengers might be entitled to carry a number of passengers of another type for free (1 Adult + 1 accompanying person for PRM).

#### **Implementation:**

Code	Description
WALK	A walk
OTHER	Other types of transfer (e.g. taxi, local city transport not included in the offer,...)

PassengerTypes

A fare might be available with a specific number of passengers only (group fares):

- Minimum number of passengers (adults + seniors + children + youths + Accompanying Person for PRM)
- Minimum number of adults + seniors + (children + youths) / 2

#### **Implementation: → PassengerConstraint**

#### **Requirements on validity for reductions**

Reductions are price reduction due to a reduction “card” an existing ticket or a pass which the passenger already holds. It might be that the physical card does not correspond to a specific reduction but provides the option to carry different reductions.

Different prices due to the age of the passenger are separate fares, not reductions to a fare.

- A card might be valid only for combined tickets only (special NS card)
- Multiple cards might apply to the same route segment, but only one of them would be applied.
- A reduction might grant a 100% price reduction → In this case an NRT is created up to the final station the customer goes with the price to the border of the area. The ticket indicates that the ticket has a reduction of 100% within the area and an indication that it is valid only together with the card. Pricing data are needed for the free travel area to get the route description.

#### **Implementation: → ReductionCard**

#### **Requirements on prices**

##### Price:

Prices might be needed in more than one currency.

- Currency (local currency might be required additionally due to local legislation for two carriers in one country)
- Amount

##### Value Added Tax:

VAT details must be given to the customer to enable a business customer to claim a refund. The VAT details include:

- Country
- VAT-Company-Id
- Percentage
- Amount

The VAT given is the VAT the carrier pays for this fare to the countries where he is providing his service. The VAT might depend additionally on whether the fare is issued as national ticket or integrated in an international ticket. Also, the VAT might depend on whether the fare is used for short distance or integrated in a long-distance ticket

Note: There are national rules on where and when to display the VAT on a ticket or receipt when a ticket is sold in that country. These are not considered here.

#### Prices:

- Fixed prices attached to a route (and fare) including VAT details (country, percentage, amount, VAT id)
- Prices depending on an intermediate distance ("fare kilometer")
- Price depending of other prices.

Decision: *The price will be delivered also in case of reductions or kilometers. No calculation is needed at the receiver side of the data.*

#### [Implementation: → Price](#)

#### **Requirements on the basic fare structure**

The basic fare element links the constraints and the price.

A name of the fare needs to be provided.

#### [Implementation: → Fare](#)

#### **Requirements on the after sales conditions**

After sales conditions define fees to be taken in case of an after sales transaction on behalf of a customer. The after sales transactions considered are:

- Cancellation (Refund)
- Exchange with a new fare of the same carrier
- Exchange with a new fare of another carrier
- Exchange for the same travel day
- Upgrade

After sales transactions due to service violations of the carrier are governed by PRR rules and are not considered here.

Some railways make refunds using other "means of payment" like bonus points, vouchers. These are not considered here and thus will not apply to the fares defined here.

Some railways apply different refund rules depending on the type of payment. These restrictions will not be considered here. It is assumed that the refund will be processed by the allocator who

manages the combined fare. He needs to consider payment restrictions in order to avoid fraud (e.g. no cash refund on electronically payed tickets, no refund unless ticket control data have been received, ...).

The refund fee can be claimed by the carrier.

#### **Implementation: → AfterSalesRules**

#### **Requirements on conditions on fulfilment**

The fulfilment defines the required types of creating a ticket for the passenger and therefore especially the required types of security to be applied.

The fulfilment might be restricted depending on:

- Allowed types of fulfilment
- Accepted / required bar codes
- Required control data exchange
- Individual ticketing

#### **Implementation: → FulfilmentConstraint**

The Required personal data might depend on the fulfilment:

- Required personal data to be provided from the allocator to the carrier depending on type of fulfilment
- Depending on border crossing and train types (Belgium border crossing of high-speed trains requires personal data)
- Data might be required for ticket holder(s) only or for all passengers

#### **Implementation: → PersonalDataConstraint**

#### **Requirements on dynamic fares and train linked tickets**

##### ***Indication of dynamic fares available online***

The allocator needs to find where he can request offers online.

- Solution 1: add the carrier(s) providing offers in their systems to the trains in the timetable
- Solution 2: publish station or ODs (optionally also by country) and/or train types (service brands) and/or the carrier(s) mentioned in the timetable where fares can be requested
- Solution 3: publish for which carriers and service brands (and optionally trains) offers can be requested

Decision: The solution should be independent from the timetable.

#### **Implementation: → FareResourceLocation**

##### ***Indication of train links on the ticket***

Ticket might be linked to the use of specific trains even in case there is no reservation. There are different options on how to indicate this restriction:

- DB solution: The train information replaces the corresponding route part

- ÖBB solution: The route description is identical to the ticket without train link and the trains are added in the condition description

*Decision: in case of a train bound ticket the route of the train should replace the route description for the part of the train bound*

Train link should include:

- Train number
- Service Brand Abbreviation (RJ, ICE,) (can be retrieved from timetable data)
- Date and departure time
- Departure Station (short name)
- Arrival Station (short name)

### ***Request for online fares***

- The complete connection must be sent
  - To check whether it is international
  - To check that it is not inside some regional tariff area
  - To calculate the correct VAT
  - To check for supplements applicable only at the start or end of the journey
- The part where the offer should be built must be provided
  - the station/connection point from and to where the offer is needed

**Implementation:** Fares

### **Requirements on combining fares**

Multiple models are defined for combining fares. The carrier defines in the fare data which model(s) the allocator can apply.

Combining the fares tries to achieve:

- Apply the conditions set by the carrier for the service he provides to secure the business model and financial interests of the carrier
- Create a simple combined fare for the customer

It is not possible to achieve both target at the same time. The different models of combining fares implement different priorities given to these targets.

In general, the basic parameters defining the price must be listed separately on the combined offer:

- route description / train link
- class of service
- passenger types

The combined price is always the sum of the prices of the parts. The allocator might add a handling fee.

### ***SEPARATE\_CONTRACTS model***

This is the model for not combining the fares in one ticket and not allowing the integration in one contract. The rules applied for this ticket are exactly the rules defined by the carrier in the fare data.

The allocator must ensure that it is clear for the customer that no common contract was established.

#### Implementation Aspect

Relevant attributes:

- FareCombinationConstraintDef.combinationModels.model == SEPARATE\_CONTRACT
- FareCombinationConstraintDef.combinationModels.allowedCommonContracts

#### ***Business Rule***

Let CC\_A be the set of allowedCommonContracts for Fare A and let CC\_B be the set of allowedCommonContracts for Fare B.

If the intersection of two sets CC\_A and CC\_B is empty, then separate contracts must be issued. Otherwise a combined contract can be issued.

#### ***SEPARATE\_TICKETS model***

This is the model for not combining the fares in one ticket, so the rules applied for this ticket are exactly the rules defined by the carrier in the fare data. The allocator can form a common contract for the whole journey.

Optionally this might be restricted by a list of carriers where this combination is allowed.

#### Implementation Aspect

Relevant attributes:

- FareCombinationConstraintDef.combinationModels.model == SEPARATE\_TICKET

#### ***CLUSTERING model***

The CLUSTERING model tries to simplify conditions and fares for the customer but sacrifices a part of the control of the carrier on its fares.

Similar types of fares are defined to belong to the same “cluster”. The after sales conditions for a cluster are defined by the allocator. However, the after sales conditions must basic rules on after sales for that cluster.

The clusters correspond to the flexibility a passenger receives to change the booked train. This corresponds directly to the after sales conditions. Hereby the fees to be paid for such an exchange are essential for the definition of clusters and not the complexity of the process to change. Thus, a train bound ticket and an open ticket belong to the same cluster in case the fees to change to different trains / times are comparable.

The after sales fees can be demanded by the carrier.

The other conditions might either be listed per carrier or combined by rules.

The customer buying products from one allocator has a simple unique view on after sales conditions. Optionally this might be restricted by a list of carriers and/or allocators where this combination is allowed.

The validity for usage is combined to be:

*The minimal validity of all included fares but at least the time needed for the combined journey according to a timetable information.*

The combined fare is available for sale only if all parts are available for sale.

The following clusters are defined (with the order from high to low flexibility): BUSINESS → FULL-FLEX → SEMI-FLEX → NON-FLEX → PROMO

Any of the clusters can contain train linked or non train-linked offers.

Offers of a less restrictive cluster can be included in a more restrictive cluster using the more restrictive rules for the combined offer, e.g., BUSINESS + FULL-FLEX → FULL-FLEX

#### BUSINESS

- Refundable after the departure or last day of validity
- Exchangeable after the departure or last day of validity

#### FULL-FLEX

- Refundable before the departure or last day of validity
- Exchangeable before the departure or last day of validity

#### SEMI-FLEX

- Refundable with fee depending on conditions of the allocator
- Exchangeable with fee depending on conditions of the allocator
- Minimum validity applies

#### NON-FLEX

- Non refundable
- Non exchangeable
- Minimum validity applies

#### PROMO

- Non refundable
- Non exchangeable
- Minimum validity applies
- Restricted combination with another cluster offers

#### Implementation Aspect

Relevant attributes:

- FareCombinationConstraintDef.combinationModels.model == CLUSTERING
- FareCombinationConstraintDef.combinationModels.combinableCarriers
- FareCombinationConstraintDef.combinationModels.referencedCluster
- FareCombinationConstraintDef.combinationModels.allowedAllocators

### **Business Rule**

Let A, B be fares.

If A.referenceCluster is element of B.allowedClusters AND if the fare provider of fare B is in A.combinableCarriers AND if the fare provider of fare A is in B.combinableCarriers **then** the fare A and B are combinable according to the CLUSTERING MODEL.

### **COMBINATION model**

The COMBINING model tries to be close to the fare conditions defined by the carrier but sacrifices the simplicity of the fare towards the customer.

The after sales conditions of the different fares will be combined into one condition to best reflect the conditions of all included carriers.

The after sales conditions will thus depend on the combinations of carriers.

Optionally this might be restricted by a list of carriers where this combination is allowed.

The combination model tries to apply all rules of the involved carriers but sacrifices simplicity of rules.

The validity is combined to be:

*The minimal validity of all included fares but at least the time needed for the combined journey according to a timetable information.*

The combined fare is available for sale only if all parts are available for sale.

The after sales fees are combined accordingly:

*At any time, the fees defined by the carriers are applied on the price part of these carriers only. The result is a list of times with increasing fees.*

Example

- Fare 1: 10%    20 days before departure, price: 100€
- Fare 2: 90%    2 days before departure, price: 200 €
- Result:        10€ fee            20 days before departure  
                    10€ + 180€ = 190€ fee    2 days before departure

### Implementation Aspect

Relevant attributes:

- FareCombinationConstraintDef.combinationModels.model == COMBINING
- FareCombinationConstraintDef.combinationModels.combinableCarriers
- AfterSalesCondition.afterSalesRules.fee
- AfterSalesCondition.afterSalesRules.applicationTime

### **Implementation: FareCombinationConstraint**

## **Requirements on reservation**

It should be possible to book reservations within the same technology.

The existing reservation services in IRS 90918-1 should also be supported.

## **Architectural Requirements**

### **Requirements on aligned processes end to end**

The processes must be aligned over all actors to reduce overall complexity and thus costs.

### **Requirements on aligned services**

The services must be aligned such that there is a close mapping to the processes supported by the services.

The services must be aligned such that the call chain between the services does not involve unnecessary mappings between different actors

### **Requirements on messages**

The messages of the online services must contain no unnecessary attributes or data structures.

Unnecessary attributes are attributes that are not needed for the online processes

### **Requirements on extendibility**

The specification must be extendible in various dimensions:

- Support of new products on the fare as well as on the offer level
- Support of new processes, e.g. product-based distribution
- Support of new modes of transportations, e.g. scooters or rail

### **Requirements on security**

The specification must include the protocols to ensure secure authentication and data transfer.

## **Legal Requirements**

The following legal regulations provide requirements that affect the solution:

### **Regulations**

**Rail PRR:** Regulation (EC) 1371/2007 on Rail Passengers' Rights and Obligations

**GDPR:** Regulation (EU) 2016/679 on data protection

- The traveller must be informed on the use of his data and on passing his data to the carrier and TCO
- The traveller must be informed which data are stored including data passed to the carrier and TCO
- The traveller has the right to ask to delete the data in case the data are not required to fulfil the contract of carriage
- The traveller has the right to ask for data correction in case the data are wrong
- Legal basis for processing of personal data with a view of black listing

Although the exchange of blacklists is not in the scope of the specification the data exchanged can be used by the allocator for a local blacklist. He has therefore to obey the regulations when using the data.

Two processing actions (automated profiling) are concerned:

- (i) Collection and analysis of personal data on regular basis for trigger points: consent of passengers or legitimate interests of the rail carrier is needed
  - (ii) Storage of information in the blacklists: legitimate interest of the rail carrier is needed
- General black list for use by multiple companies is allowed
    - (i) No access to the full list is provided
    - (ii) No automatic checking in all cases
  - Pre-cautions to be pursued by the railway undertaking
    - (i) Ensure right of access and objection
    - (ii) Information preceding such processing and notification of inclusion into the blacklist
    - (iii) Safeguards to prevent confusion
    - (iv) Additional organizational and technical safeguards for processing

#### **Art. 101§1 TFEU (competition law)**

All agreements between undertakings, decisions by associations of undertakings and concerted practices which are restrictive of competition are prohibited and void

Sensitive activities are:

- Information sharing
- Joint purchasing/selling
- **Technical standards**
- Standard terms and conditions

The following guidelines apply:

- Technical specifications for data formats should be ok
- Technical specifications for data exchange scenarios should be ok, but excessive error handling scenarios should be optional and agreed bilaterally as unnecessarily high requirements would be a restriction for small companies
- Service Level Requirements should be minimal requirements as unnecessarily high requirements would be a restriction for small companies, higher service levels must be agreed bilaterally
- Information exchange is allowed between the carriers within one contract of carriage as they are all involved in the contract. This does not apply in case of separate contracts.

## **Requirements not in Scope**

- Payment procedures including payment procedures via private currencies alike bonus points  
Information whether such payments are allowed can be included in the fare data, but the required service to handle such payments are not specified here.
- Validation of customer cards
- Combination with non-rail related fares, e.g., flight
- Combination with fares build on pay-per-use basis. e.g., scooters

## Actors / Business Capabilities

Actors are defined according to the UML specification. An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject.

Actors may represent roles played by human users, external hardware, or other subjects.

**Note that an actor does not necessarily represent a specific physical entity but merely a facet (i.e., “role”) of some entity** that is relevant to the specification of its associated use cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.

The following diagram shows the actors and principal use cases involved in rail distribution and control. The principal use case relevant for this specification is marked in yellow.

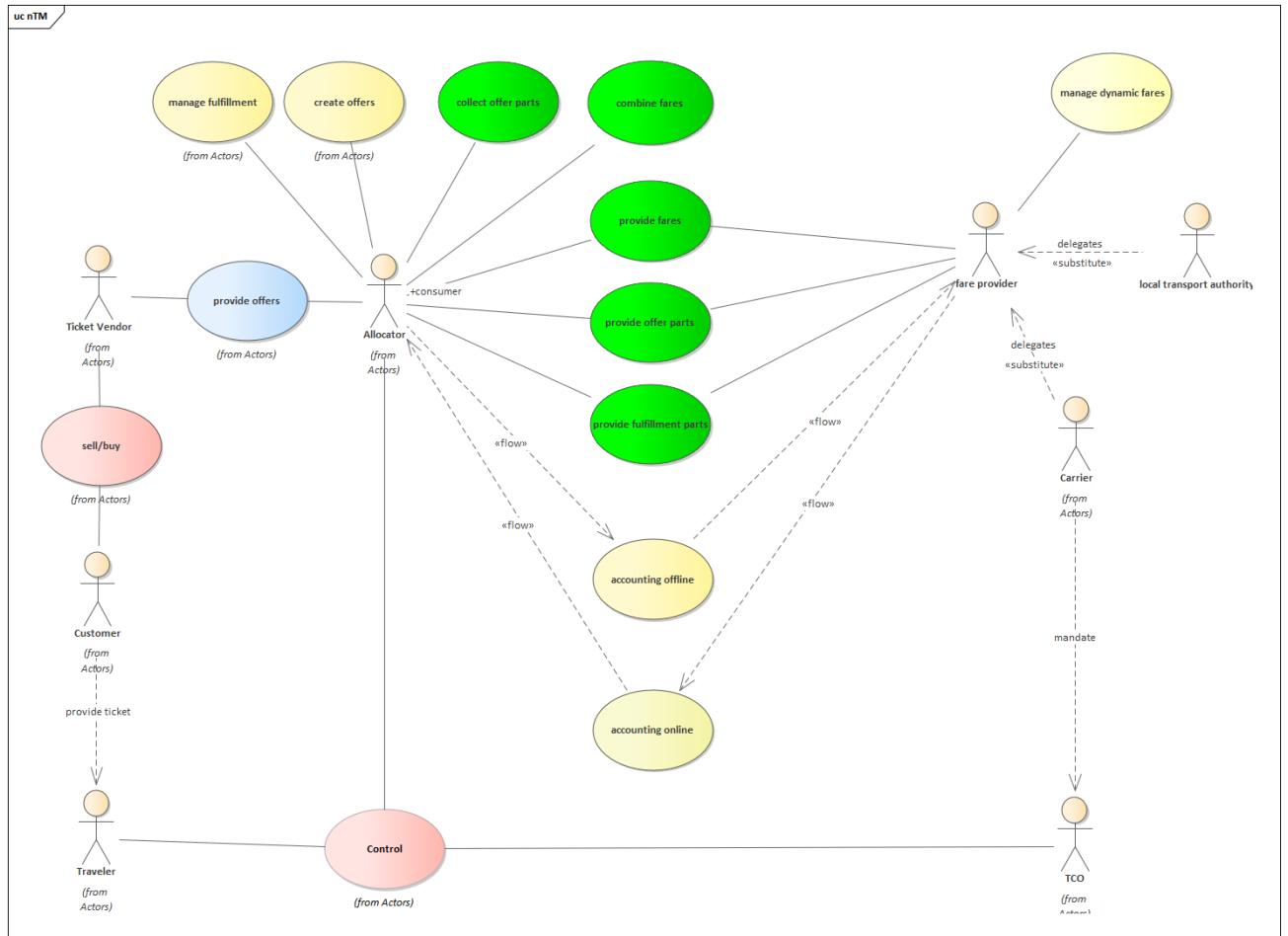


Figure 1 Actor model

The fare data exchanged might already be the result of an allocation process between different local carriers and local transport administrations.

## Actors

Actor	Description	Motivation / Distinction to other roles
<b>Allocator</b>	<p>The <b>allocator</b> manages the lifecycle of a product sold (the travel contract). He therefore needs to establish information exchange with the <b>ticket vendor</b>, <b>carriers</b> and <b>TCOs</b> involved.</p> <p>The <b>allocator</b> makes products available to the <b>ticket vendor</b>.</p> <p>The allocator could provide direct services to the <b>passenger</b> to modify the ticket status (e.g. activate / check in on a ticket).</p> <p>The allocator combines fares defined by the carriers according to their rules.</p> <p>The allocator creates the ticket fulfilment data (e.g. pdf, pkpass, ...).</p> <p><b>Note:</b> To avoid confusion due to usage differences (see the CIT term bank as well as the European TAP-TSI regulation), the terms "Issuer" and "Attributor" have been avoided in this IRS.</p>	The allocator is introduced to separate the role of just selling tickets along a route (Ticket Vendor) from the role of creating the ticket content and providing it to vendors for sale.
<b>Carrier</b>	<p>The <b>carrier</b> is the owner of the fare- He provides the transport of the <b>traveller himself or via a substitute carrier</b>.</p> <p>The travel contract provided to the customer establishes a contract between the traveller and each carrier participating in the product.</p> <p>Carriers include Railway undertaking, Bus companies, Maritime companies.</p>	
<b>Customer</b>	<p>Purchaser of a travel contract for one or more traveller.</p> <p><b>Note:</b> The <b>customer</b> is entitled to receive refund payments.</p>	The <b>customer</b> buys the ticket which represents the travel contract between one or more <b>travellers</b> and one or more <b>carriers</b> .
<b>Distributor</b>	The <b>ticket vendor</b> is the company selling the ticket provided and managed by the <b>allocator</b> to the customer.	Alias: Ticket Vendor
<b>Fare Provider</b>	The fare provider manages fares on behalf of a carrier or a local transport authority.	
<b>Local Transport Authority</b>	<p>The local transport authority organizes the local traffic within an area a behalf of the government or is itself a governmental organisation.</p> <p>It defines a fare structure for the local transport which all carriers included must apply.</p>	
<b>Passenger</b>	Person who travels using a travel contract.	The passenger and the customer can be to distinct person, e.g., if a mom buys a ticket for her daughter.
<b>Ticket Controller</b>	The Other means of ticket checking (e.g. gates) will also be named TCOs.	Alias: Train Agent

<b>Ticket Vendor</b>	The <b>ticket vendor</b> is the company selling the ticket provided and managed by the <b>allocator</b> to the customer. <b>The ticket vendor does not combine fare into one ticket.</b>	Alias: <b>Distributor</b>
----------------------	---	---------------------------

## Common Business Capabilities

### Powerful fare combination

It must be possible to combine fare according to existing fare combinations (e.g. NRT-style PRIFIS) as well as new fare combination models.

### Simple distribution

It must be easily possible to distribute existing and new products. Easily possible means two things: Firstly, for a customer it must be easily possible to find and book and – if needed – refund a booking. Secondly, for the rail sector as a whole the complexity of distribution must be reduced to save costs both for development as well as distribution.

## Business Capabilities for Distribution

### Lookup location

In order to uniquely identify a place of origin and destination a service to look up the unique code is needed. For railway stations this code is the UIC station code.

### Search trips

A service to lookup possible trips from origin to destination is needed, especially as the most attractive offers are bound to trip.

### Find offers

For a given trip possible offers spanning the complete trip need to be calculated to the customer. An offer has an overall flexibility, an overall comfort class and a minimal price. An offer consists of admissions, reservations or ancillaries. Reservations or ancillaries can be included, optional or mandatory.

The overall offer should be “homogenous”, i.e. consisting of offers of the same service class. For the Italian market, non-homogenous offers need to be supported.

**Scope:** Only trip-based offer search is defined in this version of the specification.

**Scope:** The offer must span the complete trip in this version of the specification.

**Scope:** Support for round-trips will be add in the next version of the specification.

### Pre-book offers

If a customer puts an offer into a basket on a distributor channel, it must be possible to retain this offer for a given time using a prebook service. In our design this service creates a booking in the created in the status “prebooked”. If the prebooked booking is not booked after a given time limit it well be freed which also includes freeing all eventual reservations on inventories.

### Book pre-booked booking

After the booking has been paid by the customer, he or she owns the booking and the booking is changed to “booked” by a booking service.

### Fulfill booking

After the booking process the customer needs a set of documents to travel and to prove to a ticket control organization that he or she is eligible to travel. Therefore, a service to fulfill a booking in given form, e.g. a ticket is needed. Internally, the state of the booking is changed to “fulfilled”.

**Scope:** Only support for UIC PDF is mandatory in this version of the specification.

### **Get booking**

To get the booking of a customer a service is needed. Specially care needs to be taken into account that privacy regulations are respected.

### **Refund booking**

If a customer wants to refund a booking a service to refund a booking is needed. The service calculates a refund offer including fees and amount returned which is offered to the customer. If he or she accepts the refund offer the refund offer can be booked. Special refund reasons need to exist, which affect fees and amount returned. Especially, if an agent or a machine makes a mistake a refund reason is needed to refund a booking with no penalties.

By design, the refund process is modelled similarly to the offer/booking process.

**Scope:** Only support for total refund is mandatory in this version of the specification.

### **Exchange booking**

If a customer wants to exchange a booking a service to exchange is needed. Conceptually it takes the existing booking and a new trip and calculates an exchange-offer. This exchange-offer can be booked and fulfilled similarly to refund-offer.

**Scope:** This capability is optional to support in this version of the specification.

### **Graphical seat reservation**

In order to display the layout of a train to a customer a service to access coach layout data and availability pf places is needed.

**Scope:** This capability is optional to support in this version of the specification.

### **Edit passenger information**

To add or in special cases edit passenger information a service is provided. This service is explicitly designed to be fully complaint to GDPR regulation.

### **Retrieve product information**

A service to access the attributes of a product such as detailed sales and after-sales is needed.

### **Retrieve stored personal data**

A customer can request information on the stored personal data. This includes also information on personal data passed on to allocators. The booking data can be used to show the stored personal data.

## **Business Capabilities for Fare Allocation**

### **Combine fares**

The allocator combines fares from different carriers into one offer. The rules on how to combine fares are part of the fare data.

### **Service resource location / locate dynamic fares**

Dynamic fares must be requested online. The allocator needs to find the online resource where to request the offer and book. The fare data provide information on how to find the online service.

## Provide bulk fare data

The carrier provides bulk data on his static fares and additional data for locating online services to the allocators.

## Provide dynamic fare

The carrier provides an online service to retrieve dynamic fares.

## Book offer

The carrier provides online services to book fares and cancel or exchange fares. These can be either as defined in the specification herein or via the interface defined in IRS 90918-1.

## Fulfillment

All necessary information for an allocator to build a valid a ticket including necessary attributes and control elements must be included by the fare provider.

## Reservation

Reservation has been included in the online services and the inventory resolution data for fare or reservation are included in the bulk data (→ FareResourceLocation).

Option/Step 1: using 90918-1 messages for reservation

1. offer (90918-10 REST service) → parameters for 90918-1 soap services are delivered
2. reservation as-if (90918-1 soap service) / graphical place display (90918-1 soap service)
3. reservation (90918-1 soap service) / specific place reservation (90918-1 soap service)
4. prebooking NRT (90918-10 REST service)
5. confirm booking NRT (90918-10 REST service)

Option/Step 2: using REST services 90918-10 for all services

1. offer (90918-10 REST service)
2. checkPreferences (90918-10 REST service) / graphical place display (90918-10 REST service)
3. prebooking NRT / reservation (90918-10 REST service)
4. confirmbooking reservation / NRT (90918-10 REST service)

## Get booking

To get the booking of a customer a service is needed. Specially care needs to be taken into account that privacy regulations are respected.

## Refund booking

If a customer wants to refund a booking a service to refund a booking is needed. The service calculates a refund offer including fees and amount returned which is offered to the customer. If he or she accepts the refund offer the refund offer can be booked. Special refund reasons need to exist, which affect fees and amount returned. Especially, if an agent or a machine makes a mistake a refund reason is needed to refund a booking with no penalties.

By design, the refund process is modelled similarly to the offer/booking process.

**Scope:** Only support for total refund is mandatory in this version of the specification.

## Exchange booking

If a customer wants to exchange a booking a service to exchange is needed. Conceptually it takes the existing booking and a new trip and calculates an exchange-offer. This exchange-offer can be booked and fulfilled similarly to refund-offer.

**Scope:** This capability is optional to support in this version of the specification.

## Accounting

The specification of the accounting data is not part of this document, however some on the fare content defined in this specification must be included in the accounting data.

The accounting data of a booking should include:

- The identification of the entire ticket sold (unique id within the context of the allocator for at least 2 years)
- The booking id provided by the carriers in case of online fares (unique id within the context of the carrier for at least 2 years)
- The identification of each fare included in the ticket (unique id e.g. UUID)
- The identification of individual tickets of the allocator (unique id within the context of the allocator for at least 2 years)
- The identification of individual tickets of the carriers (unique id within the context of the carrier for at least 2 years)
- The price for each fare and carrier included in the ticket
- The VAT does not need to be included in the accounting data (to be verified in RCF-1)

When using the existing 301 data file structure the ids cannot be included. Until the accounting data structures have not been extended the following intermediate solution is included:

For NRT fares distributed in the bulk data exchange:

A legacy accounting identifier is included in the fare element:

- seriesId: the last five digits of the index of a regionalValidity within the list of regional validities
- addId: the remaining digits of the index of a regionalValidity within the list of regional validities (max. 2 digits)
- tariffId: the index of the fare Element in a list of all fare elements referencing the same regionalValidity

Thereby it is possible to identify the fare element uniquely in the context of a fare data delivery.

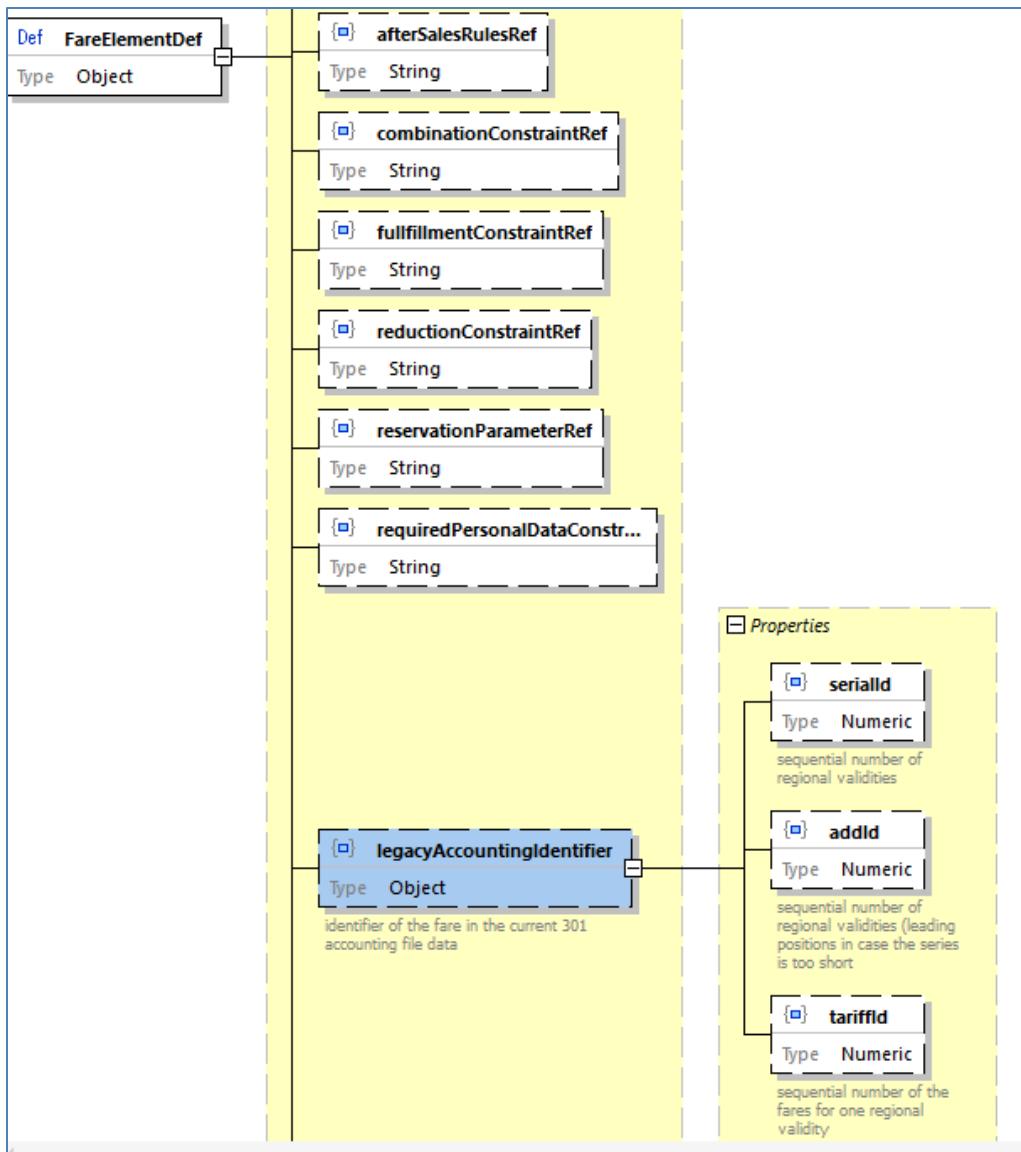


Figure 2 legacy accounting id in the fare element

Accounting data flow:

- In case of NRT fares used from a bulk data exchange:  
The allocator is responsible for the accounting. The data structure for NRT is used.
- In case of fares (IRT or NRT) used with an online booking service:  
The carrier is responsible for the accounting. The data structure for IRT is used.
- In case of fares (IRT or NRT) used with an online booking service but with carrier fees defined by the allocator:

The carrier is responsible for the accounting. The data structure for IRT is used. The allocator will inform the carrier on the applied fees in the cancellation confirmation.

### Graphical seat reservation

In order to display the layout of a train to a customer a service to access coach layout data and availability pf places is needed.

**Scope:** This capability is optional to support in this version of the specification.

## Edit passenger information

To add or in special cases edit passenger information a service is provided. This service is explicitly designed to be fully complaint to GDPR regulation.

## Retrieve stored personal data

A customer can request information on the stored personal data. This includes also information on personal data passed on to fare providers / carriers. The booking data can be used to show the stored personal data.

# Common Data Structures in Offline and Online Mode

The following chapters contain the detailed description of data structures used to describe fares.

The data structure definitions are used in the bulk data exchange and the online services. The requirements listed in chapter “Requirements” reference the data structures that implement the requirement.

## General

The following general data types shall be used:

- DateTime Formats: Date time values must be encoded according to RFC 3339, section 5.6.
- Station Codes: Station codes must be taken from the MERITS code list.
- Station Names: Station names should not include “/”, “\*”. These characters are used to define routes and alternative routes in route descriptions.

## Versioning

The specification (open api specification and schema files for offline data) are published as mayor versions in case they are not interoperable. Minor versions will include interoperable changes on the data structure which also includes providing additional data elements that are optional.

Implementers must be able to ignore additional elements.

Minor minor versions will include additional documentation only.

## Indication of personal data

Within the online part the required personal data are indicated . The general grammar to indicate required data is used.

## Indication of required data

Required data are indicated in a structured way using the following language:

Data elements are indicated by their path to the resource separated by dots:

- passenger.gender
- passenger.email
- passenger.phoneNumber
- ...

The required data elements can be combined using the logical operators:

- AND
- OR

Brackets "(" and ")" can be used in the standard way as for logical expressions

Example: passenger.gender AND (passenger.email OR passenger.phoneNumber)

Keywords can be used for specific topic:

- TICKETHOLDER

Keyword	Meaning	example
TICKETHOLDER	The data element is required for the ticket holder only	passenger.email AND TICKETHOLDER  The passenger e-mail is required for the ticket holder only

## Detailed data structures

The data structures to be used are defined in the schema and open api specification files. This section serves as additional documentation only.

### AfterSalesRules

After sales conditions define fees to be taken in case of an after sales transaction on behalf of a customer. The after sales transactions considered are:

- Cancellation (= Refund)
- Exchange with a new fare of the same carrier
- Exchange with a new fare of another carrier
- Upgrade

See code list: TransactionType

The after sales rules might include rules for a delayed payment to avoid fraud. This might depend in the type of fulfilment. (e.g. no cash refund on electronically payed tickets, no refund unless ticket control data have been received, ...).

The refund fee can be claimed by the carrier.

The after sales rules bundle a set of after sales conditions under an id that can be referenced by a fare.

An after sales condition applies for a set of after sales transactions and specified:

- the fee to be applied
- the time when the fee needs to be applied
- whether the fee needs to be given to the carrier or can be kept by the allocator

The data include the amount to be refunded. The amount is given to avoid any calculations with complex rules (percentage + minimum / maximum value) at the allocator side.:

- The value and currency to be applied
- A percentage for customer information. Due to rounding errors a calculated percentage could result in strange numbers (e.g. 9.99% instead of 10%)
- The unit on which the value is calculated (travellers or bookings)

The time when the fee needs to be applied is defined by:

- The time unit (hours, minutes, ...)
- The time difference value
- The time reference (before departure...)

See code lists:

## TimeReference , TimeUnit

An after sales fee is applied from a time before departure, after sale,..)

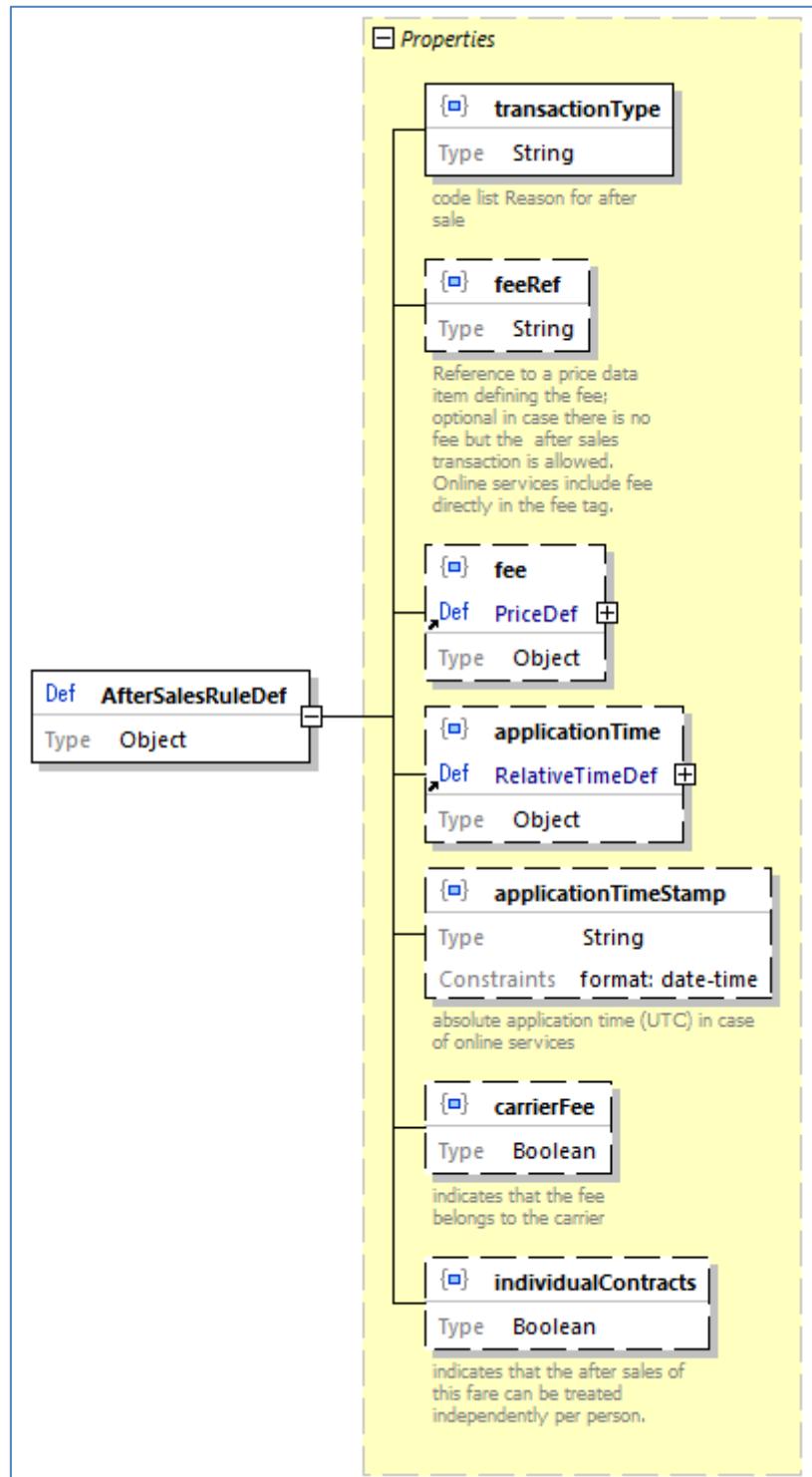


Figure 3 after sales rule data structure

In case multiple rules apply to the same after sales transaction the rule with the closest time in the future must be applied.

Data constraints	
fee/feeRef	In online services a fee is included directly, in bulk data exchange a fee must be included in the list of prices and referenced by an id.  The fee provided must include the currency EUR if not agreed bilaterally otherwise.
applicationTime/applicationTimeStamp	An application time stamp can be used in online services only.  If an application time stamp is provided the allocation Time as relative time must not be included.

## Calendar

A calendar is referenced by a unique id which can be referenced from other data structures linked to the fare.

A Calendar defines a list of days between two dates. If the dates are not provided in UTC the offset to UTC must be provided additionally.

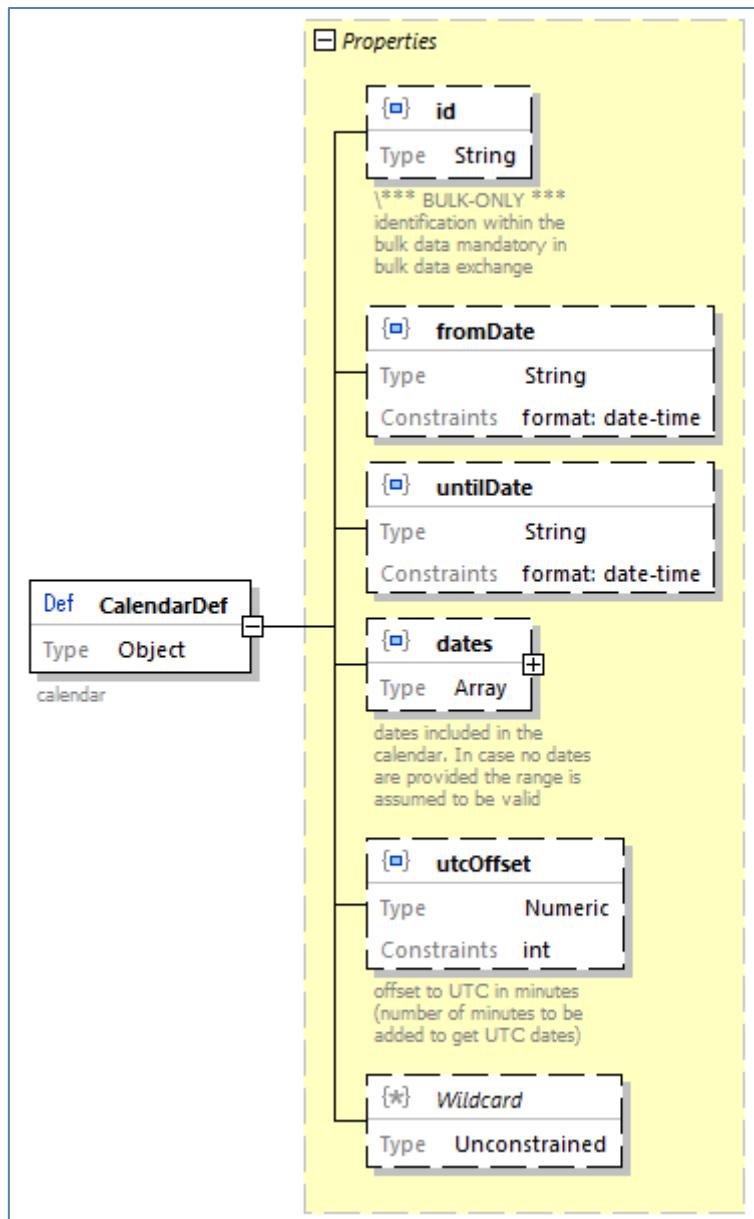


Figure 4 Calendar data structure

Data constraints	
fromDate/untilDate	fromDate and untilDate must be provided.  fromDate <= untilDate
dates	fromDate <= date <= untilDate

## CarrierConstraint

The carrier constraint can be referenced by a fare via the id.

Carrier constraint limits an open fare - not linked to a train - to some carriers. The carriers can be specified either as exclusion list or alternatively as inclusion list.

Carriers are specified by their Company code (RICS code).

The included / excluded carriers are also part of the FCB barcode (IRS 90918-4) content and the ticket control data (IRS 90918-9).

The offline data structure includes an additional id to reference the constraint within a fare data delivery.

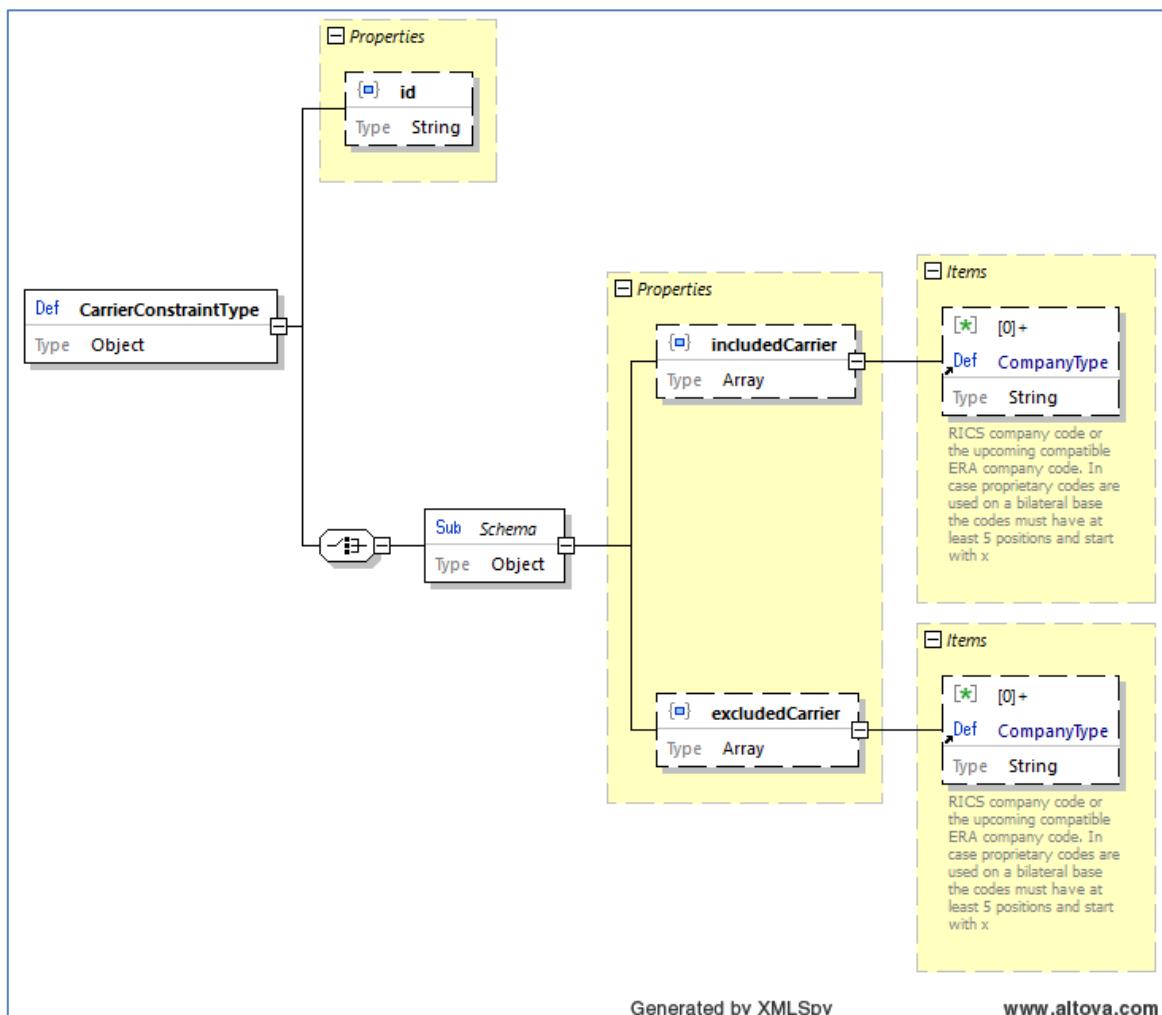


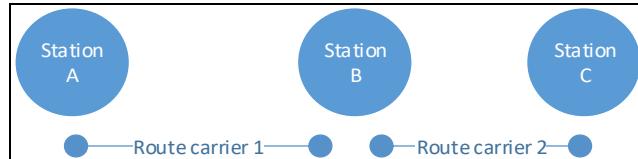
Figure 5 carrier constraint data structure (offline)

Data constraints	
includedCarriers/excludedCarriers	Either a list of included or a list of excluded carriers must be provided. It is not allowed to provide both lists.

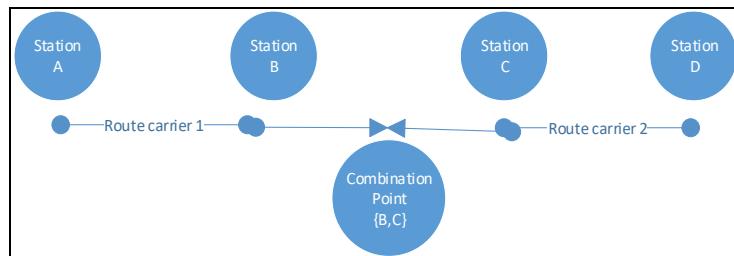
## ConnectionPoint

A connection point defines a point where two regional validities of different carriers can be connected. A connection point is implemented as the list of stations which hit connects.

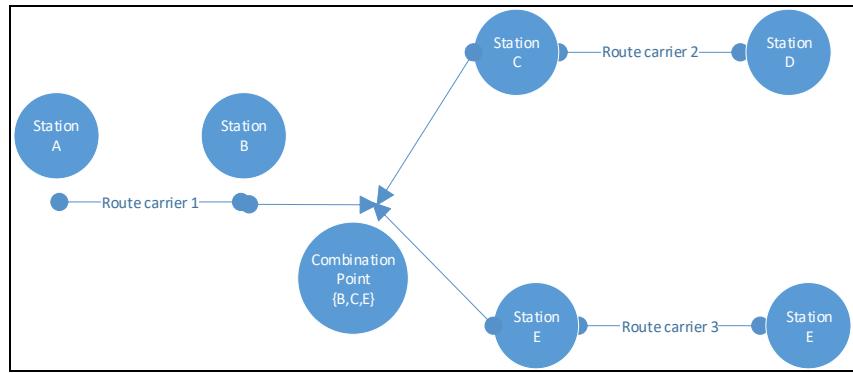
In case a route ends at a real station the connection point includes the real station.



In case the combination is not at a real station an indication is needed to define the allowed combinations. This could be done by listing the next stations of other carriers which would allow a combination. Combinations would be allowed if the combination points of two routes share two common stations.



This would also work with multiple stations.



Connection points will include a border point code to support existing implementations where the border point code is compared with the timetable data. As in principle every station can become a connection point (e.g. all stops from Aachen to Brussels are connection points from DB to SNCB) implementations based on border point codes cannot cover all connections.

As on both sides of a connection multiple small stations could be connected and not all of them might be in the timetable of a train the connection point should allow to connect sets of stations.

1. Two fares can be connected in case their connection points share a common station in the provided station sets if only one set is provided by a connection point.
2. Two fares can be connected in case their connection points share a common station in two if the provided station sets of each connection point.

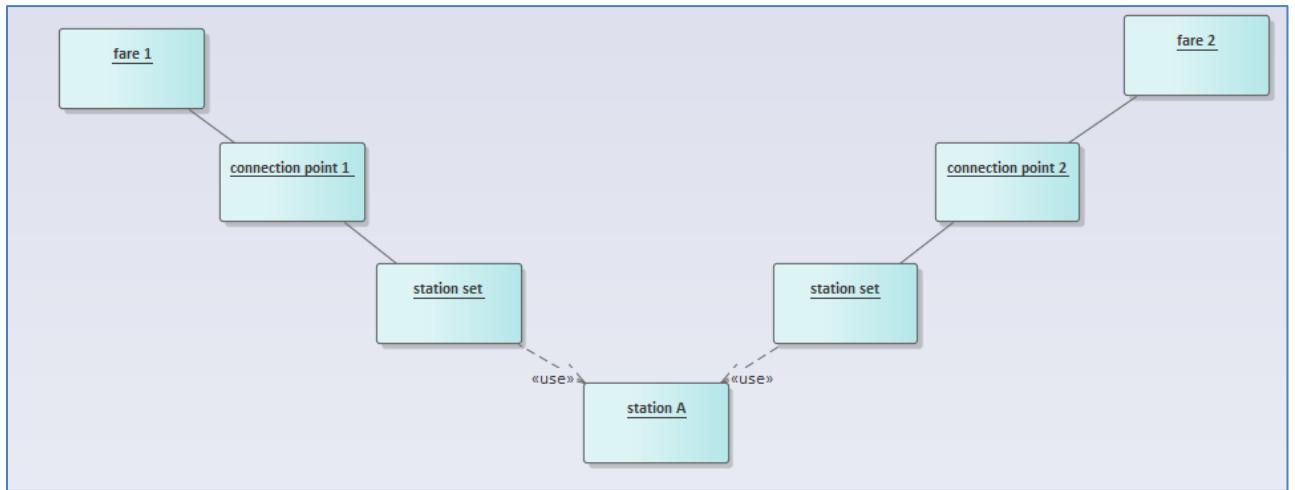
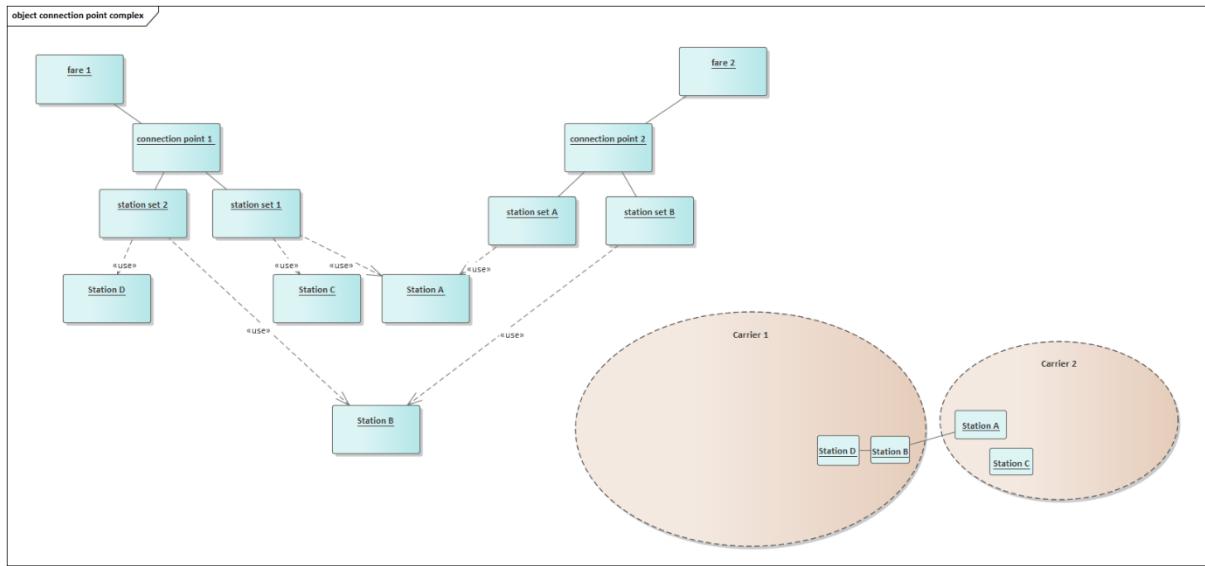
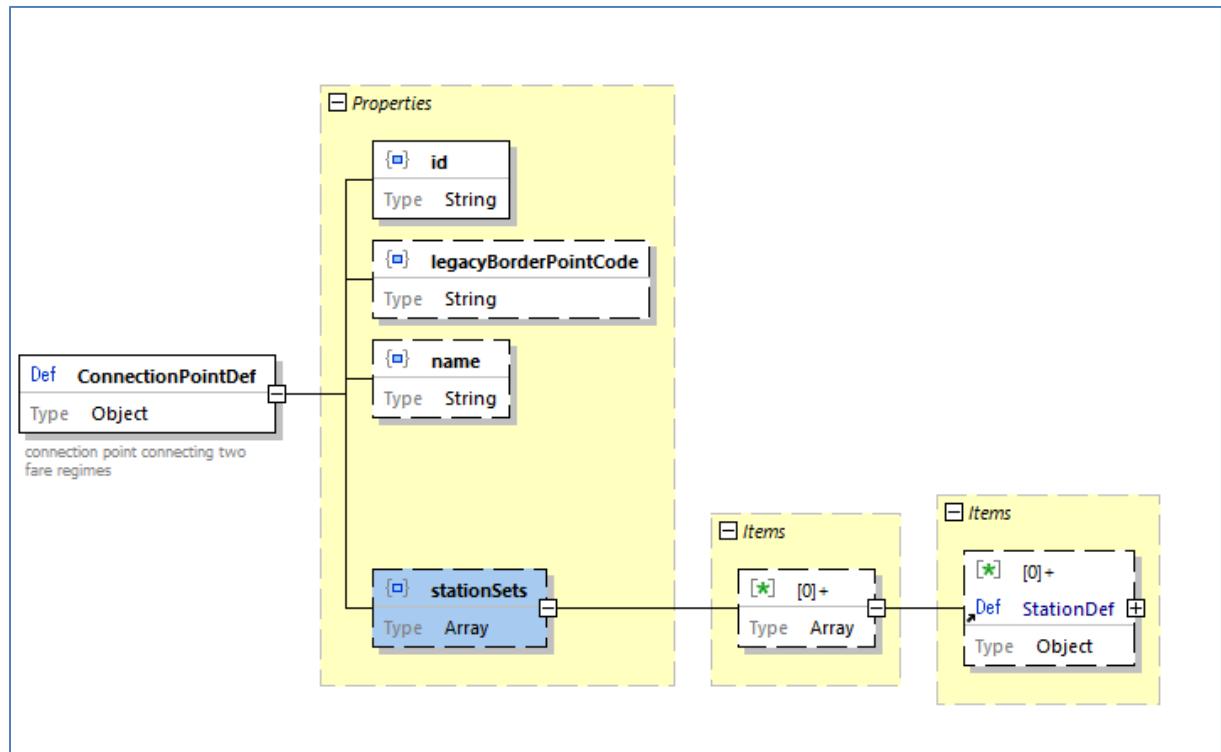


Figure 6 connection points - simple case 1



*Figure 7 connection points - complex case 2*

The online data structure does not include the id and the legacy code.



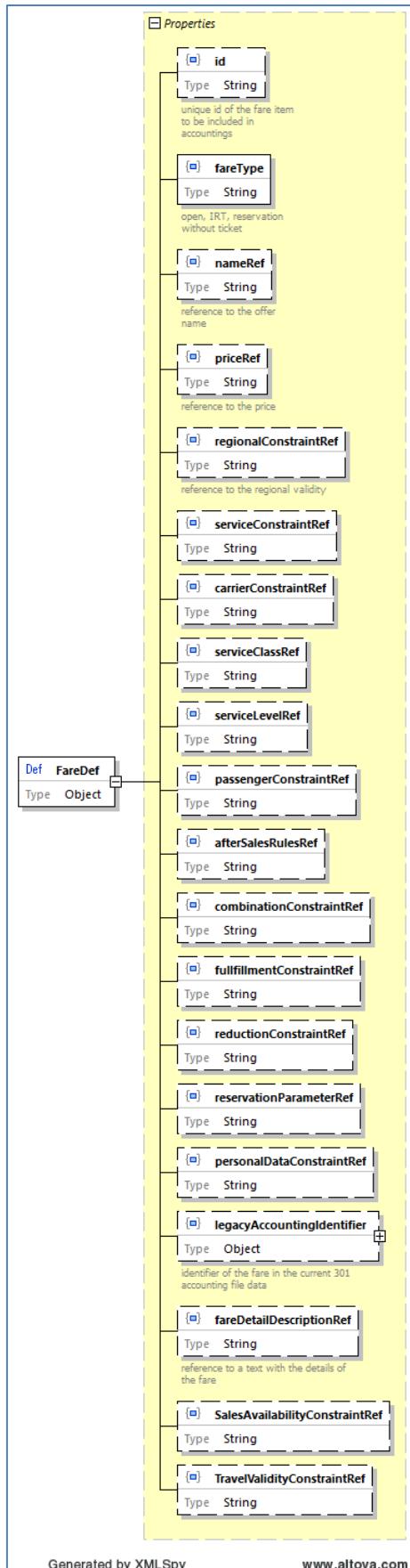
*Figure 8 ConnectionPoint data structure offline*

<b>Data constraints</b>	
stationSets	<p>At least one set with one station must be provided in case the fare border is a real station.</p> <p>Two station sets must be provided in case the fare border is inbetween two real stations.</p>
legacyBorderPointCode	The legacy border point code must be provided for the time being. New implementations should not use the border point code.

## Fare

An elementary fare to create an offer linking all constraints to one price.

Data elements	
fareType	NRT, IRT, Anxilliaries , Reservations
name	Name of the fare
fareDetailDescription	Additional explanation on the fare (e.g. on included fees like Diabolo or Venice fee)
price	Price with currency EUR must be provided if not otherwise agreed bilaterally.
regionalConstraint	Definition of the regional validity of the fare and the geographical combination rules (connection points)
serviceConstraint	Restrictions of the service allowed to be used
carrierConstraint	Restriction on the carriers that can be used with the fare.
serviceClass	Class the passenger can use
serviceLevel	Mode detailed category of places the passenger can use.
passengerConstraint	Rules and restrictions on the passenger types allowed to use the fare and rules on combining passengers.
afterSalesRules	After sales rules for the fare. In case the allocator is responsible for the aftersales rules this is almost empty.
combinationConstraint	Rules on the model of combination of this fare with fares of other carriers.
fulfilmentConstraint	Restrictions and requirements on the fulfilment and security to be applied by the allocator.
reductionConstraint	Rules on reduction cards necessary to apply the fare.
reservationParameter	Information on parameters for reservation via the 90918-1 interface and reservation options.
regulatoryConditions	Legal regimes to be applied to the fare (e.g. COTIV, SMPS regulations)
personalDataConstraint	Rules on the personal data to be provided in a booking
legacyAccountingIdentifier	Data to be included in the current IRS 30301 accounting data format.
salesAvailabilityConstraint	Rules on the allowed sales dates for the fare.
travelValidityConstraint	Rules on the validity for travel of this fare.
legacyConversion	Defines whether this fare is allowed to be converted to the old 108.1 data structure and used according to the old rules. <ul style="list-style-type: none"> <li>• YES</li> <li>• NO</li> <li>• ONLY – this fare is provided for conversion only</li> </ul>



Generated by XMLSpy

[www.altova.com](http://www.altova.com)

Figure 9 Fare element data structure (offline)

Data constraints	
price	A price must be provided for all offline fares including those where the price is zero.
legacyAccountingIdentifier	In case 30301 in the current version is used to accounting these data must be provided for offline fares
serviceClass	Must be provided for offline fares
combinationConstraint	Must be provided for offline fares
travelValidityConstraint	Must be provided for offline fares
SalesAvailabilityConstraint	Must be provided for offline fares

## FareCombinationConstraint

The fare combination constraint defines the rules of combining fares from different carriers. It provides a list of combination models the allocator can choose of.

Content	Description
model	Code of the combination model applied
combinableCarriers	List of carriers that can be combined with this fare. <b>If empty, there is no restriction in combining different carriers.</b> Carriers are listed by their RICS company codes.
onlyWhenCombined	Indicates that this fare can be used only if it is combined with another fare of another carrier.
referenceCluster	Cluster within the clustering model to which this fare belongs
allowedClusters	List of clusters with which this fare can be combined
allowedAllocators	List of allocators which can combine this fare... <b>If empty, there is no restriction in combining different carriers.</b> Carriers are listed by their RICS company codes.  Allowed allocators is not present in the online data.
allowedCommonContracts	List of Carriers with which the allocator can for a common contract. <b>If empty, there is no restriction in indicating common contracts to the passenger except for the SEPARATE_CONTRACT model.</b> Carriers are listed by their RICS company codes.

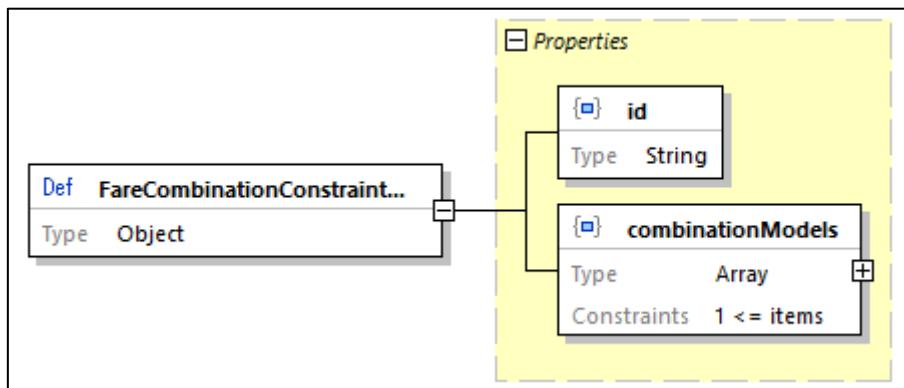


Figure 10 fare combination constraint data structure

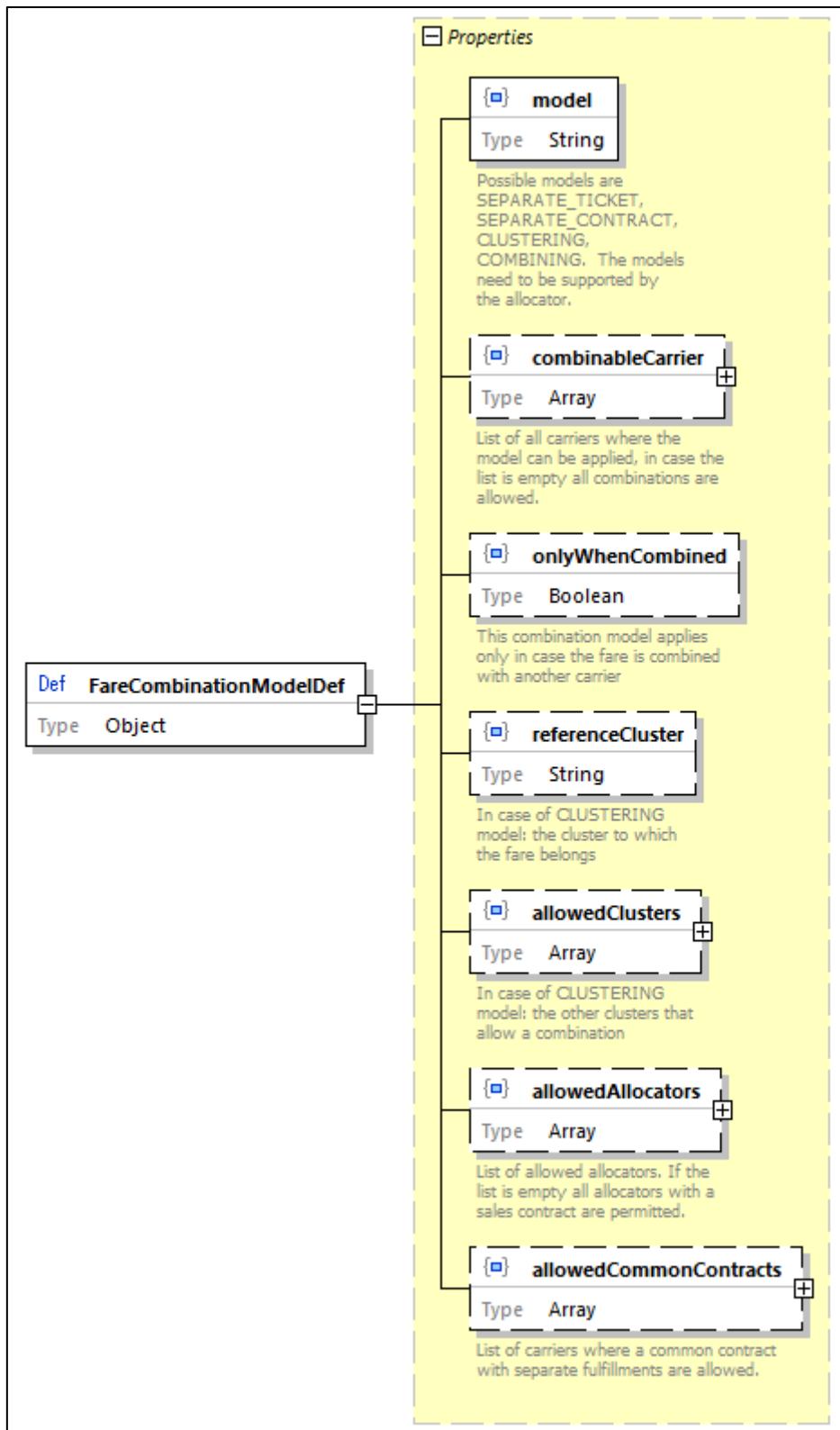


Figure 11 fare combination constraint data structure

Combination model code	Description																
SEPARATE_CONTRACT	This is the model for not combining the fares in one ticket and not allowing the integration in one contract. The rules applied for this ticket are exactly the rules defined by the carrier in the fare data. The allocator must ensure that it is clear for the customer that no common contract was established.																
SEPARATE_TICKET	This is the model for not combining the fares in one ticket, so the rules applied for this ticket are exactly the rules defined by the carrier in the fare data. The allocator can form a common contract for the whole journey.																
CLUSTER	<p>The CLUSTERING model tries to simplify conditions and fares for the customer but sacrifices a part of the control of the carrier on his fares. Similar types of fares are defined to belong to the same “cluster”. The after sales conditions for a cluster are defined by the allocator.</p> <p>However, the after sales conditions must basic rules on after sales for that cluster.</p> <p>The clusters correspond to the flexibility a passenger receives to change the booked train. This corresponds directly to the after sales conditions. Hereby the fees to be paid for such an exchange are essential for the definition of clusters and not the complexity of the process to change. Thus, a train bound ticket and an open ticket belong to the same cluster in case the fees to change to different trains / times are comparable.</p> <p>The after sales fees can be demanded by the carrier.</p> <p>The other conditions might either be listed per carrier or combined by rules.</p> <p>The customer buying products from one allocator has a simple unique view on after salles conditions.</p> <p>The basic parameters defining the price must be obeyed individually within separately on the combined fare/offer:</p> <ul style="list-style-type: none"> <li>○ route description / train link</li> <li>○ class of service</li> <li>○ passenger types</li> </ul>																
COMBINE	<p>The COMBINING model tries to be close to the fare conditions defined by the carrier but sacrifices the simplicity of the fare towards the customer.</p> <p>The after sales conditions of the different fares will be combined into one condition to reflect the conditions of all included carriers.</p> <p>The after sales conditions will thus depend on the combinations of carriers.</p> <p>At any time, the after sales fees defined by the carriers are applied on the price part of these carriers only. The result is a list of times with increasing fees.</p> <p>E.g.:</p> <table> <tbody> <tr> <td>Carrier 1:</td> <td>10%</td> <td>20 days before departure</td> <td>price: 100€</td> </tr> <tr> <td>Carrier 2:</td> <td>90%</td> <td>2 days before departure</td> <td>price: 200 €</td> </tr> <tr> <td>Result:</td> <td>10€ fee</td> <td>20 days before departure</td> <td></td> </tr> <tr> <td></td> <td>190€ fee</td> <td>2 days before departure</td> <td></td> </tr> </tbody> </table>	Carrier 1:	10%	20 days before departure	price: 100€	Carrier 2:	90%	2 days before departure	price: 200 €	Result:	10€ fee	20 days before departure			190€ fee	2 days before departure	
Carrier 1:	10%	20 days before departure	price: 100€														
Carrier 2:	90%	2 days before departure	price: 200 €														
Result:	10€ fee	20 days before departure															
	190€ fee	2 days before departure															

--	--

Additional clustering model data:

Fare clusters reflect the flexibility a fare provides to the customer. Flexibility is defined by the after sales conditions that apply when a passenger wants to change his ticket.

Fare cluster code	description
BUSINESS	Refundable after the departure or last day of validity Exchangeable after the departure or last day of validity
FULL_FLEX	Refundable before the departure or last day of validity Exchangeable before the departure or last day of validity
SEMI_FLEX	Refundable with fee depending on conditions of the allocator Exchangeable with fee depending on conditions of the allocator Minimum validity applies
NON_FLEX	Non refundable Non exchangeable Minimum validity applies
PROMO	Used on a bilateral basis only.  Non refundable Non exchangeable Minimum validity applies

Combinations of fares of different clusters is allowed with the fare clusters listed in allowedClusters. However not all combinations would be provided to the customer. A fare will be combined with a fare of the same cluster and in case this is not available with one of the higher clusters.

E.g.:

Carrier 1:	BUSINESS	→ CombinableClusters: BUSINESS,FULL_FLEX,SEMI_FLEX, NON_FLEX
	SEMI_FLEX	→ CombinableClusters: SEMI_FLEX, NON_FLEX
Carrier 2:	BUSINESS	→ CombinableClusters: BUSINESS,FULL_FLEX,SEMI_FLEX, NON_FLEX
	FULL_FLEX	→ CombinableClusters: FULL_FLEX,SEMI_FLEX, NON_FLEX

Possible combined offers are:

BUSINESS	(Carrier 1 BUSINESS + Carrier 2 BUSINESS)
FULL_FLEX	(Carrier 1 BUSINESS + Carrier 2 FULL_FLEX)
SEMI_FLEX	(Carrier 1 SEMI_FLEX + Carrier 2 FULL_FLEX)

A NON\_FLEX would be formally allowed, but with the same price as the SEMI\_FLEX so it should not be shown to the customer:

NON_FLEX	(Carrier 1 SEMI_FLEX + Carrier 2 FULL_FLEX)
----------	---

Other combinations would also be formally allowed by the data but suppressed as they would only offer a higher price. These should be suppressed by the allocator. E.g.:

FULL\_FLEX (Carrier 1 BUSINESS + Carrier 2 BUSINESS)

Data constraints	
combinationModel	At least one model must be provided

## FareResourceLocation

Fare resource location provides data on where to find online services for fares.

The fare location provides three options:

- Link a resource to a carrier – the carrier must be known from the timetable
- Link a resource to the train – the data must be updated in case of new trains
- Link a resource to stations:
  - The link can be made for stations and for connection points
  - The link is valid if start and end station (or connection points) provide the link

The online link provides information on:

- The type of resource either for a whole train or an area. In case of a train the request must be for the train route between stations (e.g. IRT), whereas for areas there might be multiple splits in-between a train run (e.g. NRT).

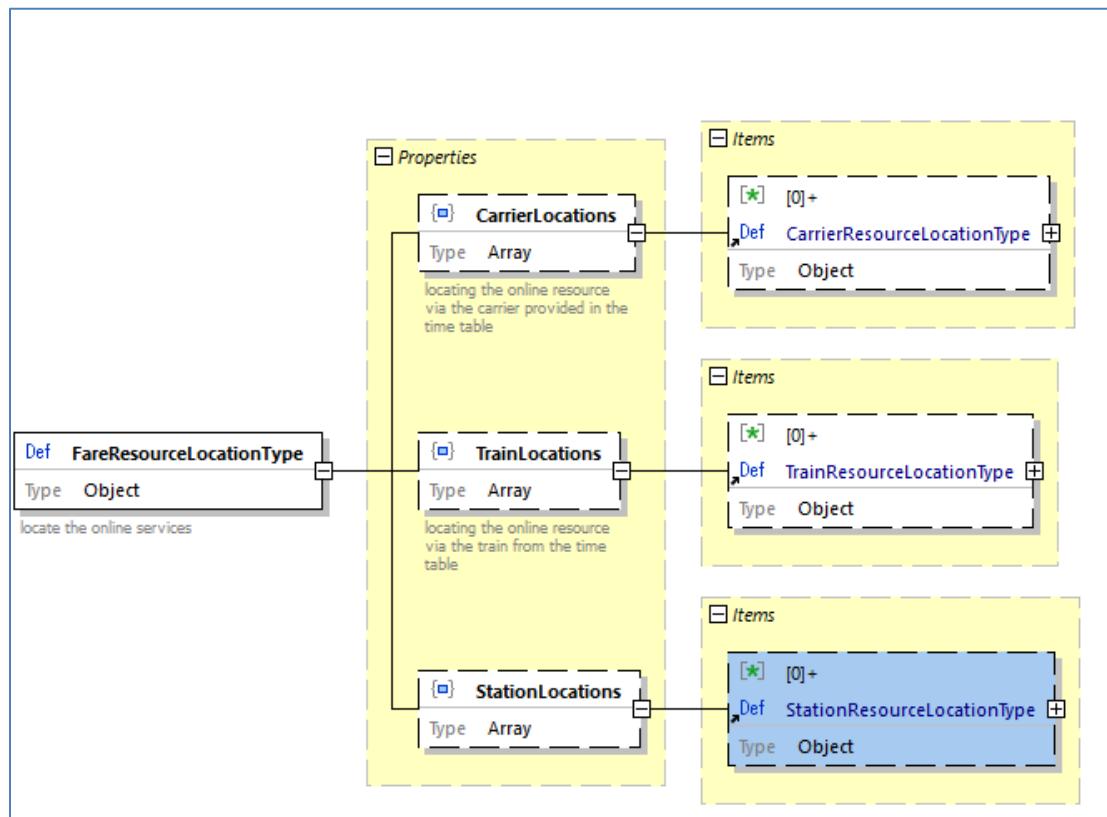


Figure 12 FareResourceLocation data structure

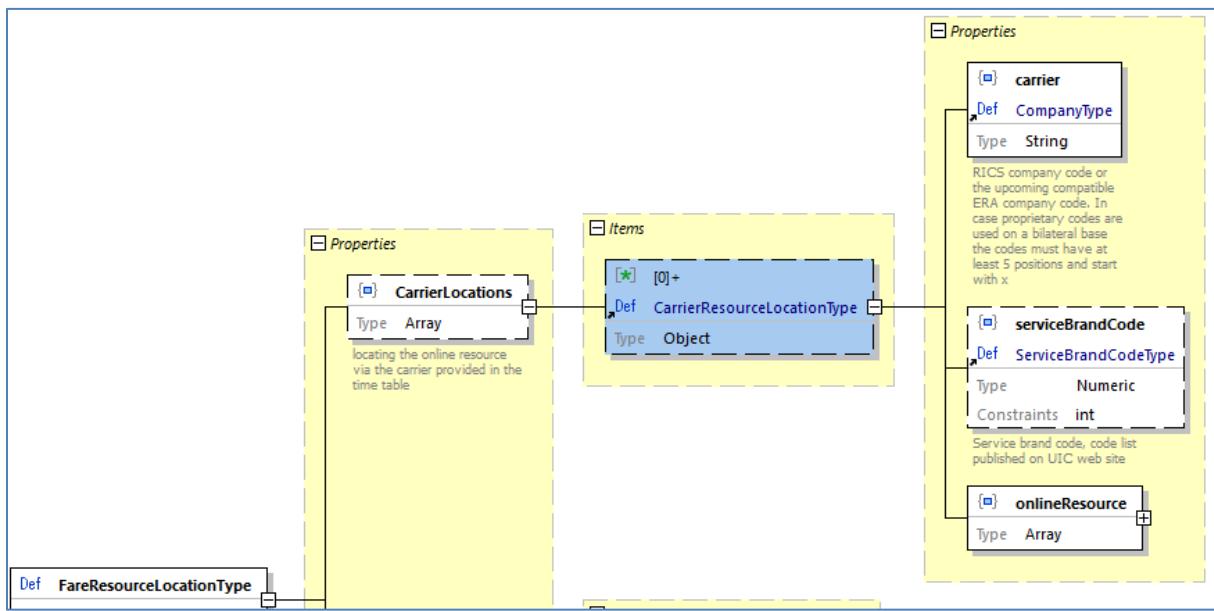


Figure 13 FareResourceLocation data structure - carrier link

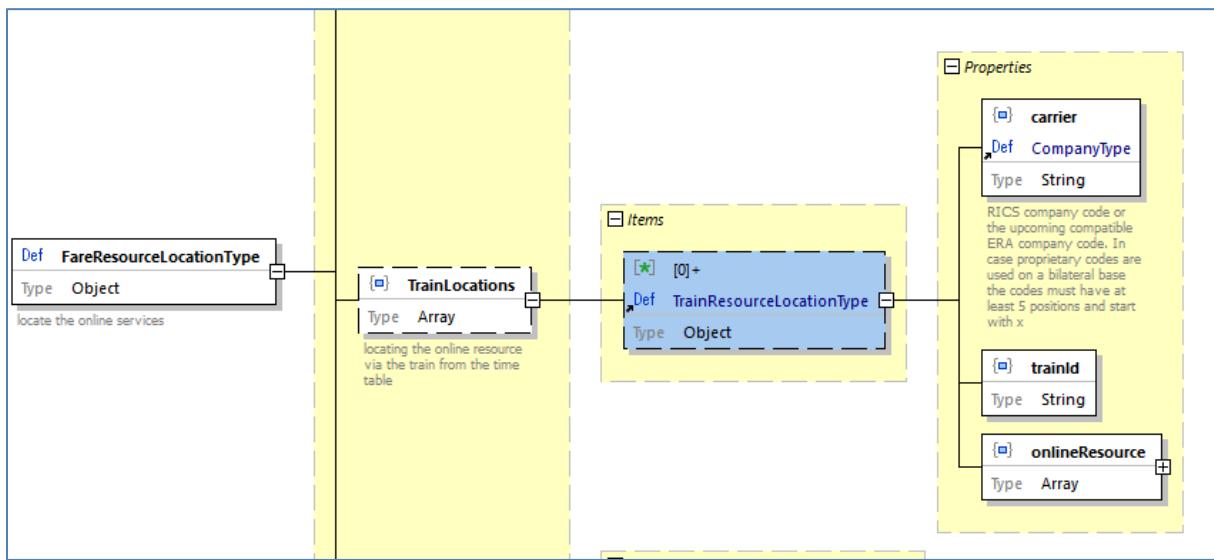


Figure 14 FareResourceLocation data structure - train link

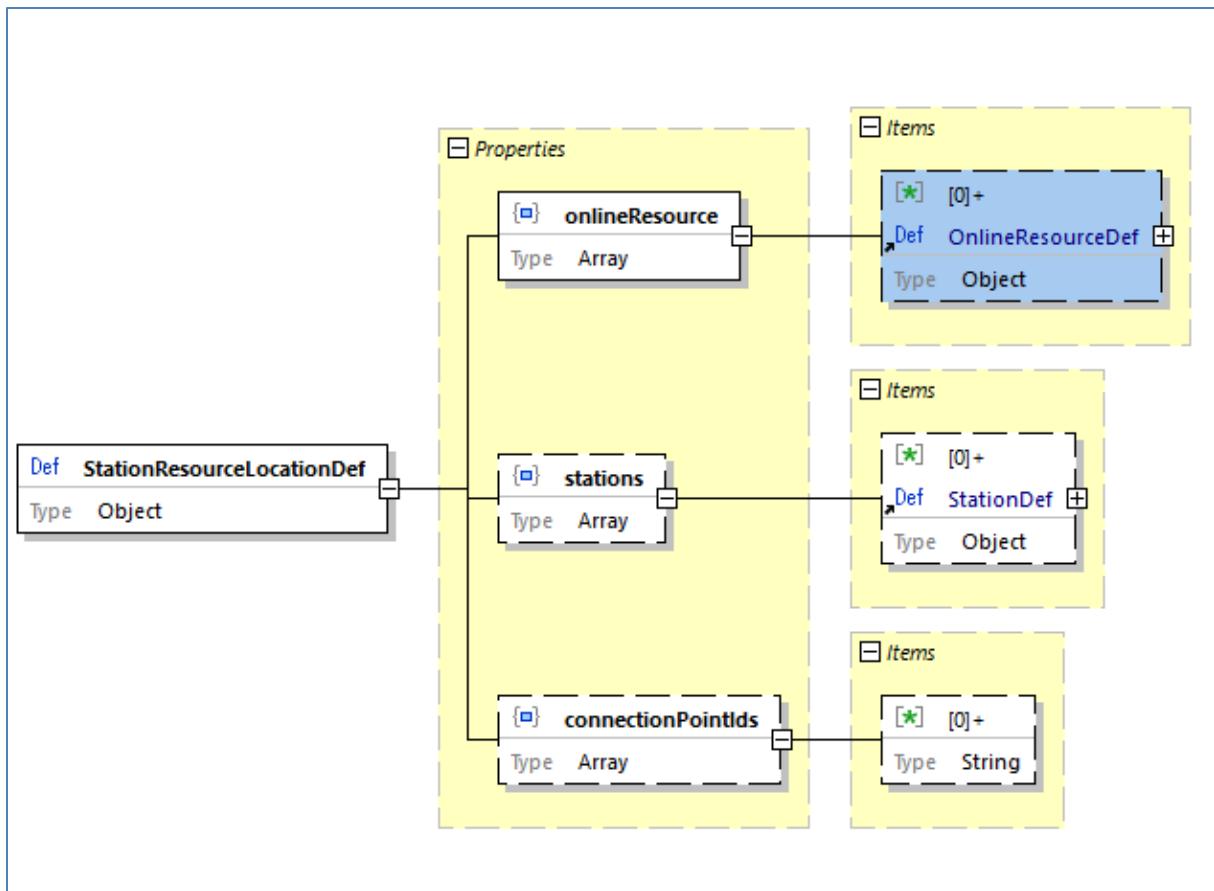


Figure 15 FareResourceLocation data structure - station link

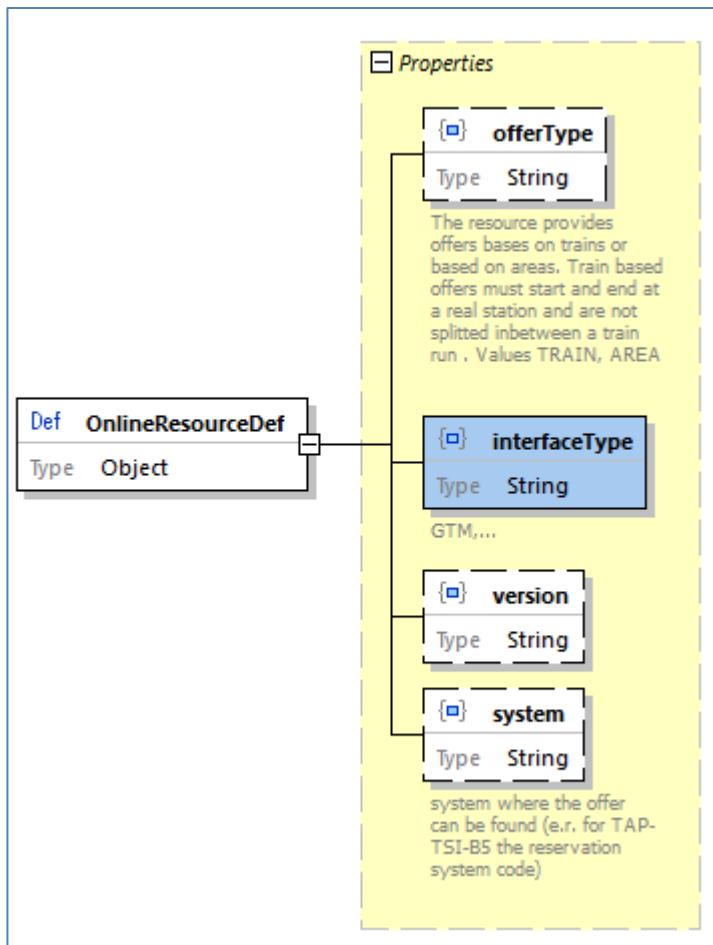


Figure 16 FareResourceLocation data structure - online resource

Code lists:

## Graphics Icons

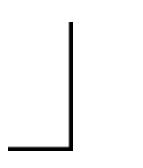
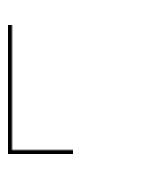
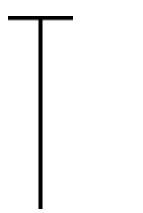
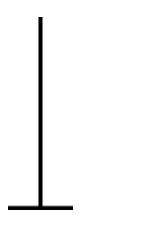
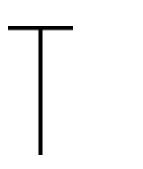
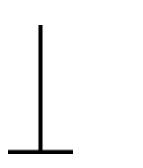
Graphic icons are used to display a coach including its facilities based on the coach layout and availability of places. The graphical items include frames and icons to display seats etc. Graphical items must be provided by the sales application of the issuer application to ensure a unique look and feel of the application.

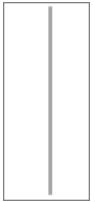
The coach layout provides only the position of graphic items (co-ordinates) not the graphical presentation at the sales application (pictures).

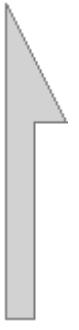
A large table spans two places, whereas a small table spans only one place. A small wall spans two places and a large wall spans 3 places. A very small wall spans one place only.

Icon	description	<u>icon</u>	Orientation uses	Mounting used
	seat	1	to right	-
	seat	1	to left	-
	seat	1	up	-
	seat	1	bottom	-
	berth	2	to left	-
	berth	2	to right	-
	couchette	3	to right	-
	couchette	3	to left	-
	small table	20	-	top
	small table	20	-	bottom

	big table	21	-	top
	big table	21	-	bottom
	wall with angle to left  height covers three places	24	-	bottom
	wall with angle to right  height covers three places	25	-	bottom
	wall with angle to left  height covers three places	24	-	top
	wall with angle to right  height covers three places	25	-	top
	small wall with angle to left  height covers two places	27	-	bottom
	small wall with angle to right  height covers two places	28	-	bottom

	small wall with angle to left  height covers two places	27	-	top
	small wall with angle to right  height covers two places	28	-	top
	wall between two compartments  height covers three places	23	-	bottom
	wall between two compartments  height covers three places	23	-	top
	small wall between two compartments  height covers two places	26	-	bottom
	small wall between two compartments  height covers two places	26	-	top
	small wall between two compartments  height covers two places	29	-	bottom

	small wall between two compartments  height covers two places	29	-	top
	very small wall between two compartments  height covers one places	32	-	bottom
	Very small wall between two compartments  height covers one places	32	-	top
	end-to-end wall	30	-	top-to-bottom
	Mobile wall between sleeping compartments  height covers three places	31	-	top-to-bottom / bottom/top
	arrow indicating upper level	51	-	top-to-bottom

	arrow indicating lower level	52	-	top-to-bottom
	arrow indicating upper level	53	-	top-to-bottom
	arrow indicating lower level	54	-	top-to-bottom
	1st class area	101	-	-
	2nd class area	100	-	-
	mobile phone area	103	-	-
	mobile phone forbidden area	104	-	-
	silence area	116	-	-
	bar area	102	-	-

	dining area	107	-	-
	bicycle area	108	-	-
	luggage area	109	-	-
	luggage locker	110	-	-
	conference compartment	111	-	-
	wardrobe	113	-	-
	dust bin	114	-	-
	children play area	117	-	-
	Space for prams	112	-	-
	family area	106	-	-
	PRM area	105	-	-
	Priority Seats	120	-	-
	Non-smoker area	118	-	-
	WC	115	-	-
	power socket	119	-	-
	WiFi area	130	-	-

	PRM toilet	131	-	-
	Air condition	132	-	-
	USB ports for charging	133	-	-
	Reclining seat	134	-	-
	Window	135	-	-
	stairs upwards	136	-	-
	stairs downwards	137	-	-
	Working compartment	144	-	-
	Cabin for allergic passengers	145	-	-
	Meeting lounge for 7 persons (Pendolino)	150	-	-
	Meeting lounge for 12 persons	151	-	-
	Space for skis	154	-	-
	Conductor working compartment	155	-	-
	Mobile phone booth	156	-	-
	T-loop (audio induction loop)	158	-	-

	Pets compartment	160	-	-
	Sleeping cabin with toilet and shower	161	-	-
	interconnectible sleeping compartments	162	-	-
	Shower	163	-	-
	Baby-care table	168	-	-
	push-button operated wheelchair ramp	170	-	-
	Sink / washbasin	171	-	-
	Ladies compartment	172	-	-

#### icons concerning the entire coach

	1st class coach	121	-	-
	2nd class coach	122	-	-
	mobile phone coach	127	-	-
	mobile phone forbidden in the entire coach	128	-	-
	1silence coach	126	-	-

	coffee bar	123	-	-
	dining car	124	-	-
	bicycle coach	125	-	-
	mixed group/individual coach	129	-	-
	WiFi area	139	-	-
	Air condition	140	-	-
	USB ports for charging	141	-	-
	Business class	142	-	-
	Panorama coach	143	-	-
	Luggage van	173	-	-
	Sleeping car	174	-	-
	Car-carrier	175	-	-

Interface Type: Interface Type

Data constraints	
System	For reservation interface 90810-1 the reservation system code is used.

## FareReferenceStationSet

The fare reference station set defines a set of stations where the fare is valid for all included stations. This set can be used in the regionalValidity description.

The corresponding bar code ab ticket control data will only contain the code of the station set, but the allocator needs the complete list of station to link the fare to the train routes.

A name can be provided.

The station set is referenced by the company code of the fare provider and a code unique within the fare provider.

A legacyCode can be provided to include the current code in the 108.1 data.

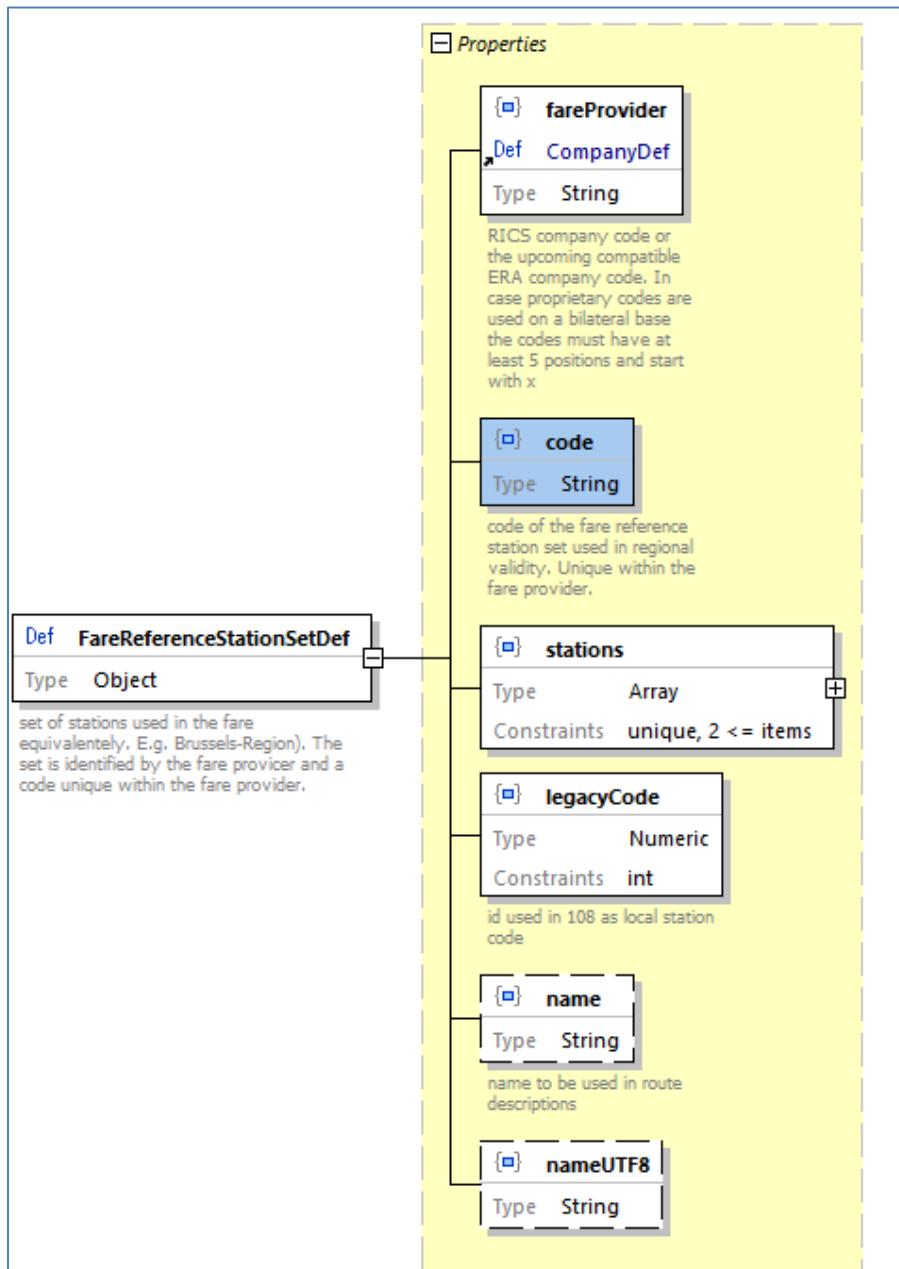


Figure 17 FareReferenceStationSet Definition data structure

Data constraints	
legacyCode	A legacyCode must be provided for the time being. New implementations should not rely on that code.
Name	The name should not include "/.*".

## FulfilmentConstraint

The fulfilment constraint limits the applicable types of fulfilment and defined whether control data need to be transferred via a standard interface (IRS 90918-4).

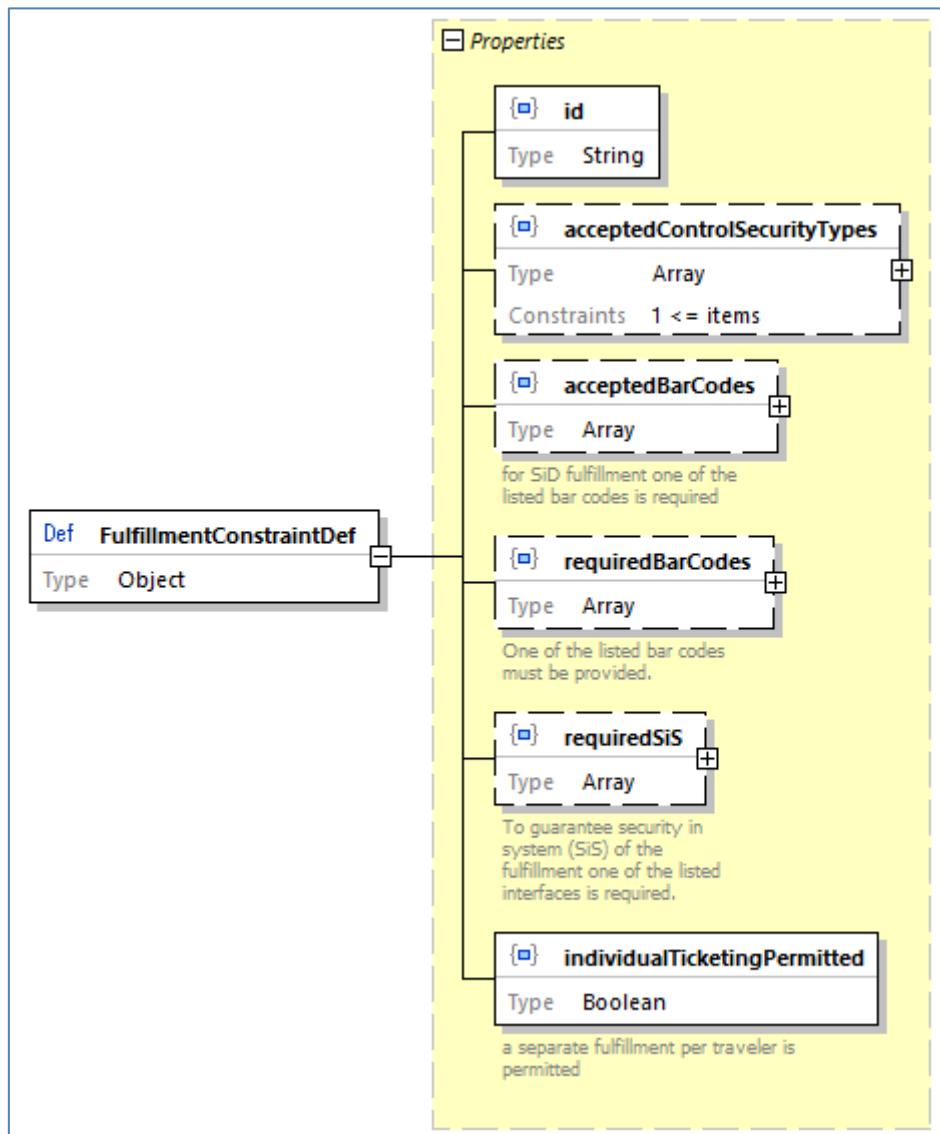


Figure 18 Fulfilment constraint data structure

Code lists:

**for required SiS:**      **CardType**

The following code list defines the card types for cards used

Predefined Card-Ids	Description
LOYALTY_CARD	Loyalty card
REDUCTION_CARD	Card providing reduction
PASS	Pass for travelling

ControlDataExchangeType

**for barcodes:**      **BarcodeType**

**for fulfilment:**      **ControlSecurityType**

Data constraints	
acceptedFulfilmentType	At least one accepted fulfilment type must be provided

## **Line**

Line defines the regional validity on a specific line. It might have additional restrictions to enter or leave at specific stations or to be used within an area or city only.

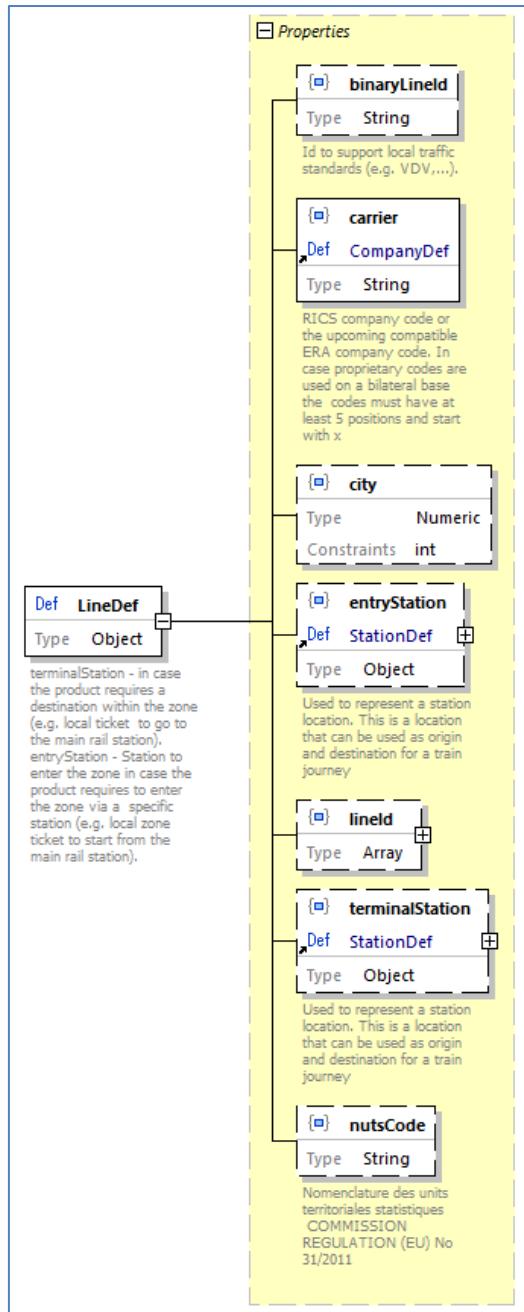


Figure 19 Line data structure

## PassengerConstraint

Passenger constraint defines restrictions of a fare concerning passengers. In online services the structure is reduced to constraints that need to be passed on for control to barcodes and control registries.

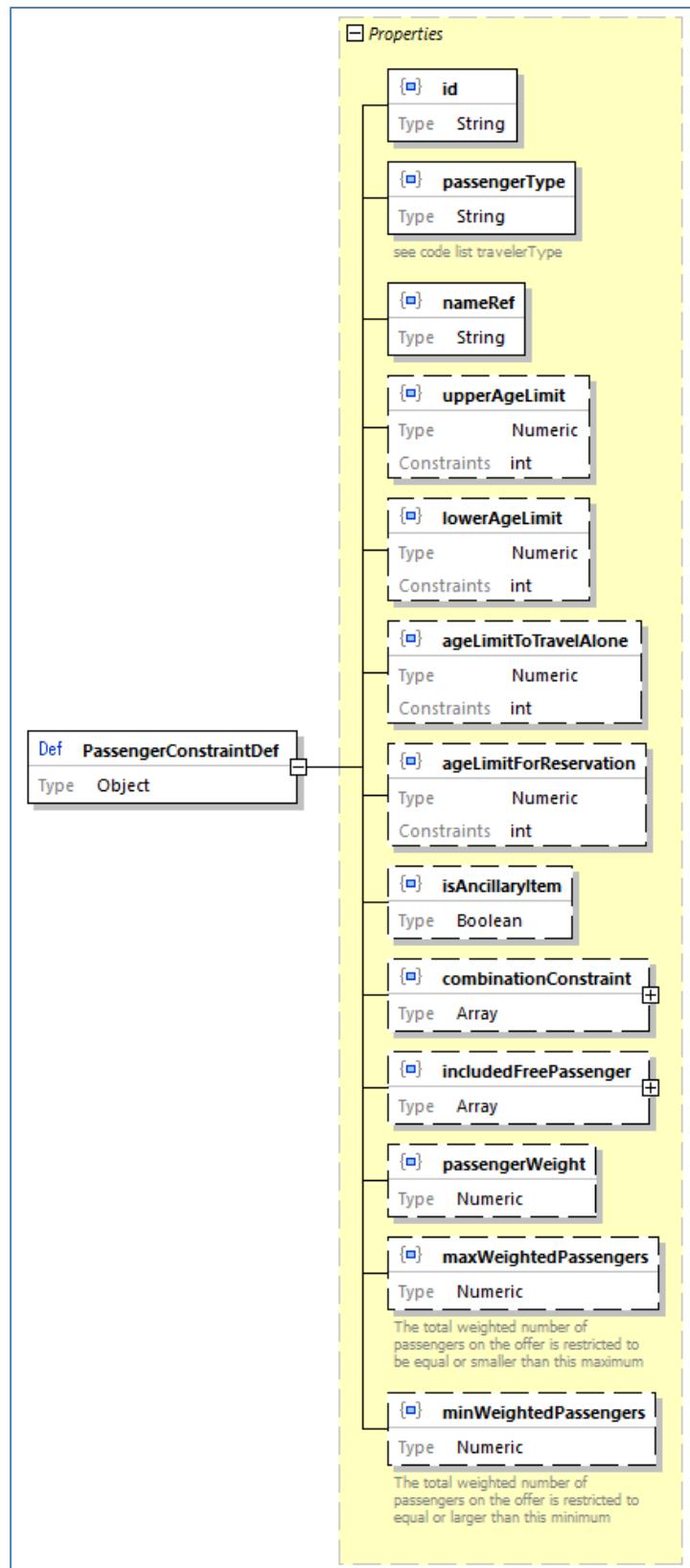


Figure 20 PassengerConstraint data structure offline

Data constraints	
upperAgeLimit, lowerAgeLimit	upperAgeLimit >= lowerAgeLimit

## PersonalDataConstraint

Specification of personal data to be delivered to the carrier. Personal data might be included in:

- Booking service (OSDM and/or IRS 90918-1)
- Control data (bar code and/or control data delivery IRS 90918-4)

The requirement for personal data might depend on the type of fulfilment or on specific border crossings.

acceptedReason	Accepted reason to change personal data after booking confirmation  See code list: Personal data change reasons								
transfer	The way the personal data are transferred.  See code list: Personal data transfer types								
ticketHolderOnly	Personal data are required for the ticket holder only								
dataItem	Code of the data item required  <b>See code list: LanguageCode</b> ISO-639-1 codes  <b>OverruleCode</b> Overrule code to request a refund without fees. <table border="1" data-bbox="393 1455 1378 1590"> <thead> <tr> <th>Code</th><th>Description</th></tr> </thead> <tbody> <tr> <td>STRIKE</td><td>Refund due to strike</td></tr> <tr> <td>SALES_STAFF_ERROR</td><td>Refund due to an error made by the sales staff</td></tr> <tr> <td>PAYMENT_FAILURE</td><td>Refund as the payment failed</td></tr> </tbody> </table> Personal data items	Code	Description	STRIKE	Refund due to strike	SALES_STAFF_ERROR	Refund due to an error made by the sales staff	PAYMENT_FAILURE	Refund as the payment failed
Code	Description								
STRIKE	Refund due to strike								
SALES_STAFF_ERROR	Refund due to an error made by the sales staff								
PAYMENT_FAILURE	Refund as the payment failed								

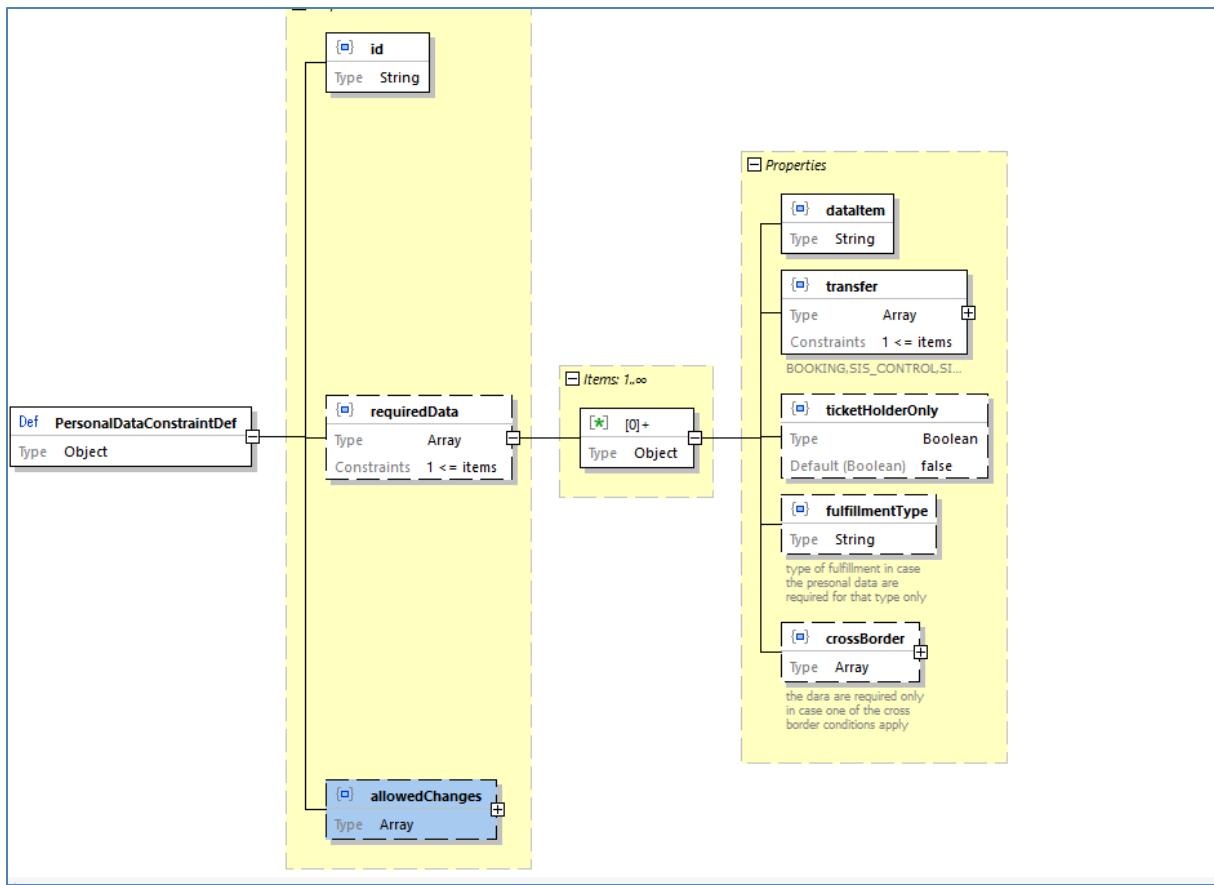


Figure 21 Required Personal data structure

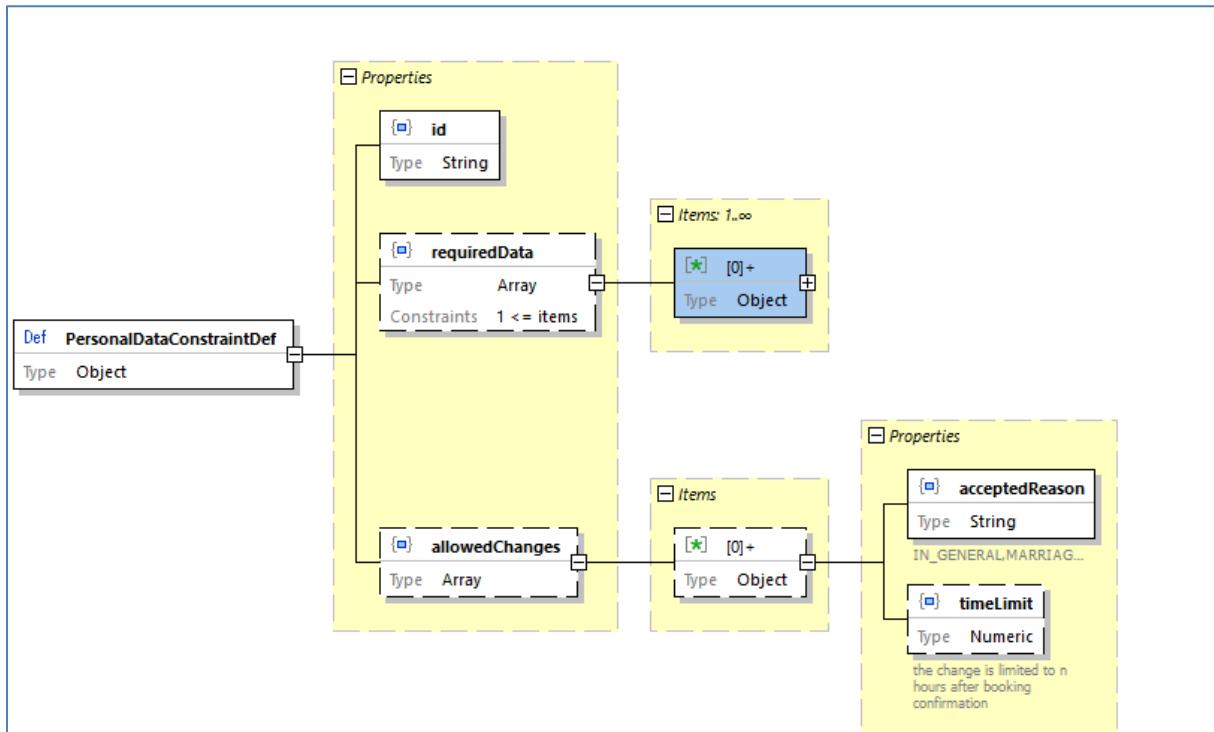


Figure 22 allowed changes on personal data

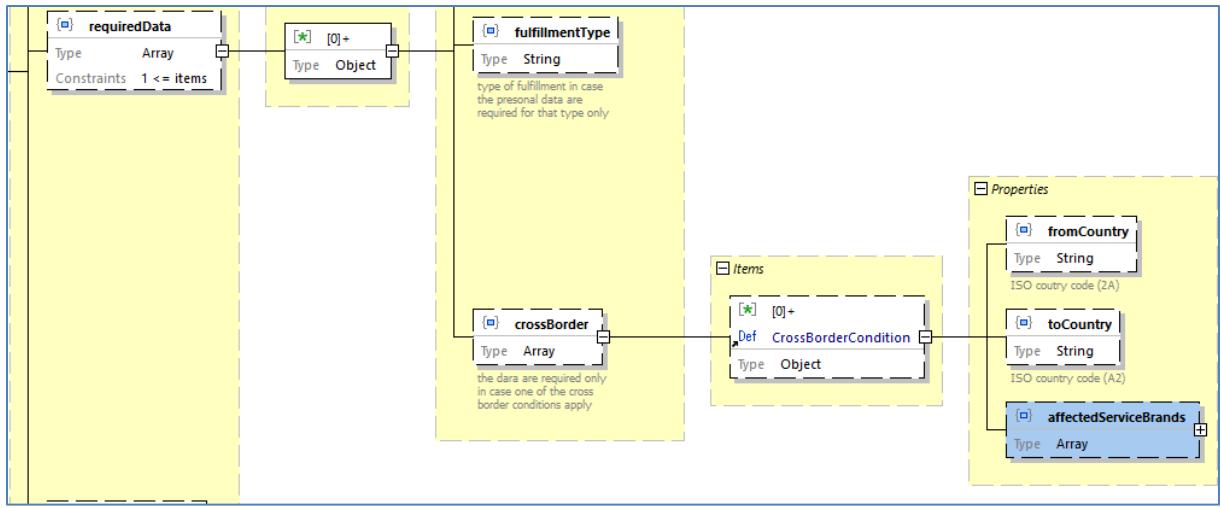


Figure 23 cross border conditions for personal data

## Price

The price data structure provides the price or a fee including the VAT details optionally in different currencies.

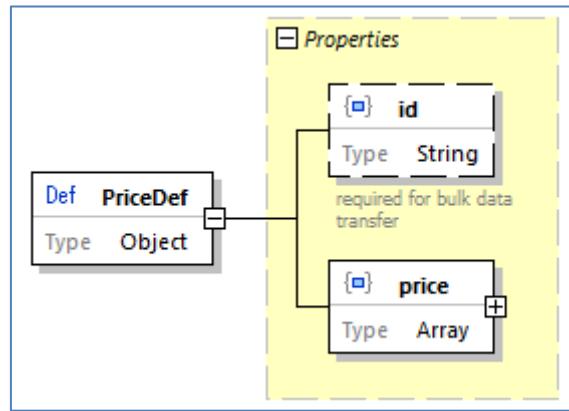


Figure 24 Price data structure with a list of currency prices

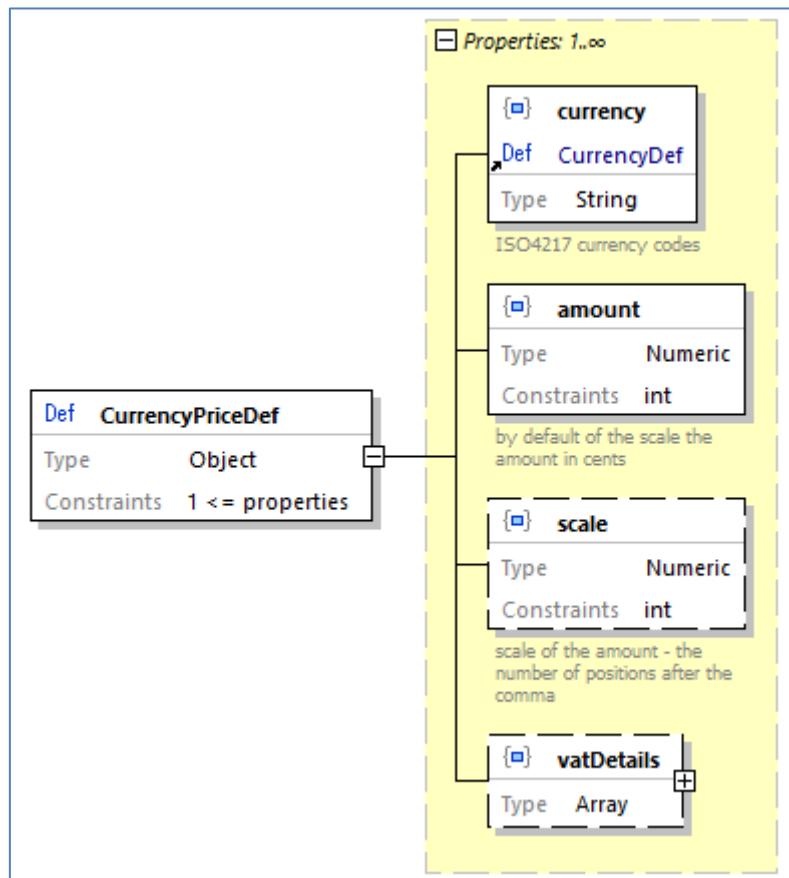
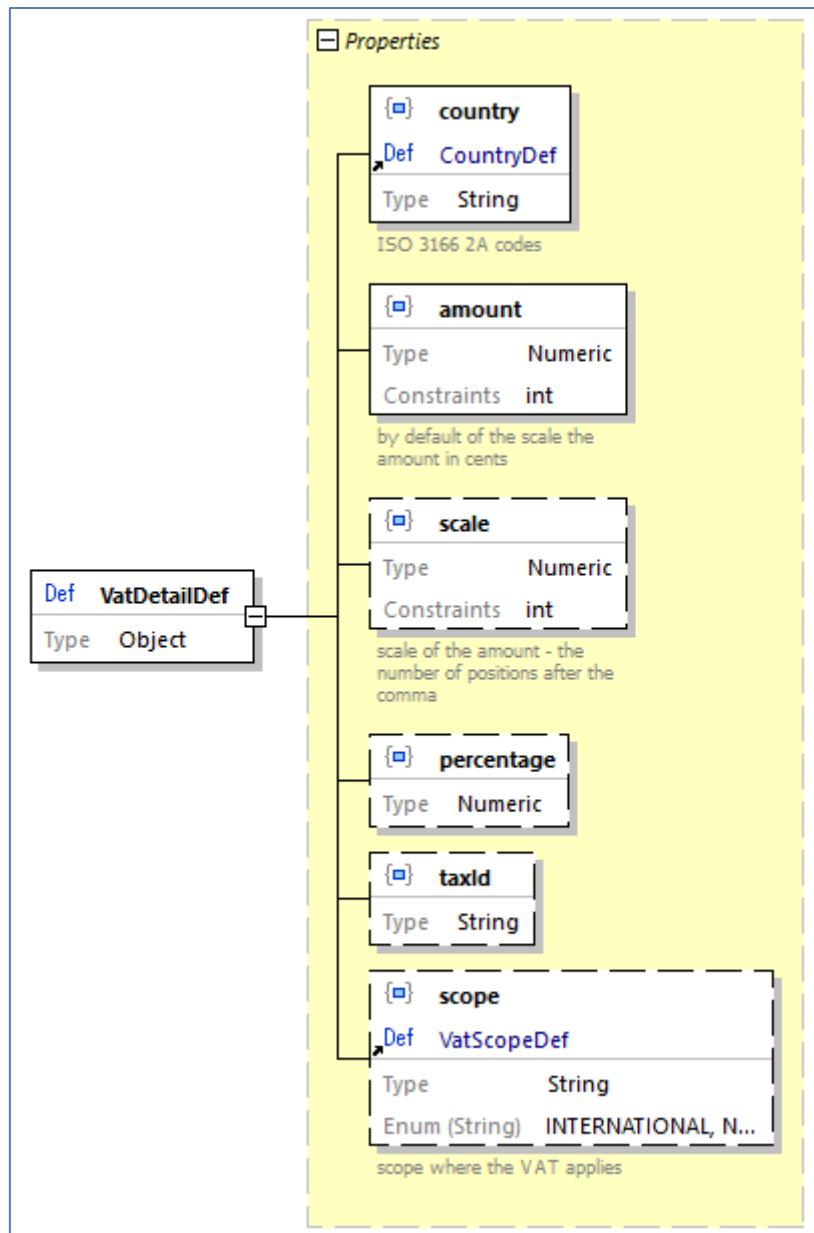


Figure 25 Price data structure of one currency



*Figure 26 VAT details within a price*

## Scope: see code list TaxScope

Data constraints	
amount	Amount >= sum of VAT-amounts

## ReductionCard

The reduction cards of a carrier are listed in the bulk data.

List of cards of the carrier:

Id	Unique id of the card. The id must start with the RICS code of the carrier
name	Name and short name of the card. The name should be used for the card selection by the customer, the short name should be used for barcodes. Usually the card name is not translated, but the card name might be provided in different languages by carriers in multilingual countries.
serviceClass	Service class indicated for the class
issuer	Issuer of the card. Usually the carrier providing the fare data.
type	Type of the cards to separate between loyalty cards, cards that are tickets (passes), and reduction cards.  LOYALTY_CARD,REDUCTION_CARD,PASS
cardIdRequired	Indicates that the card id must be provided in the prebooking request to validate the card. This card cannot be used without the online services for booking

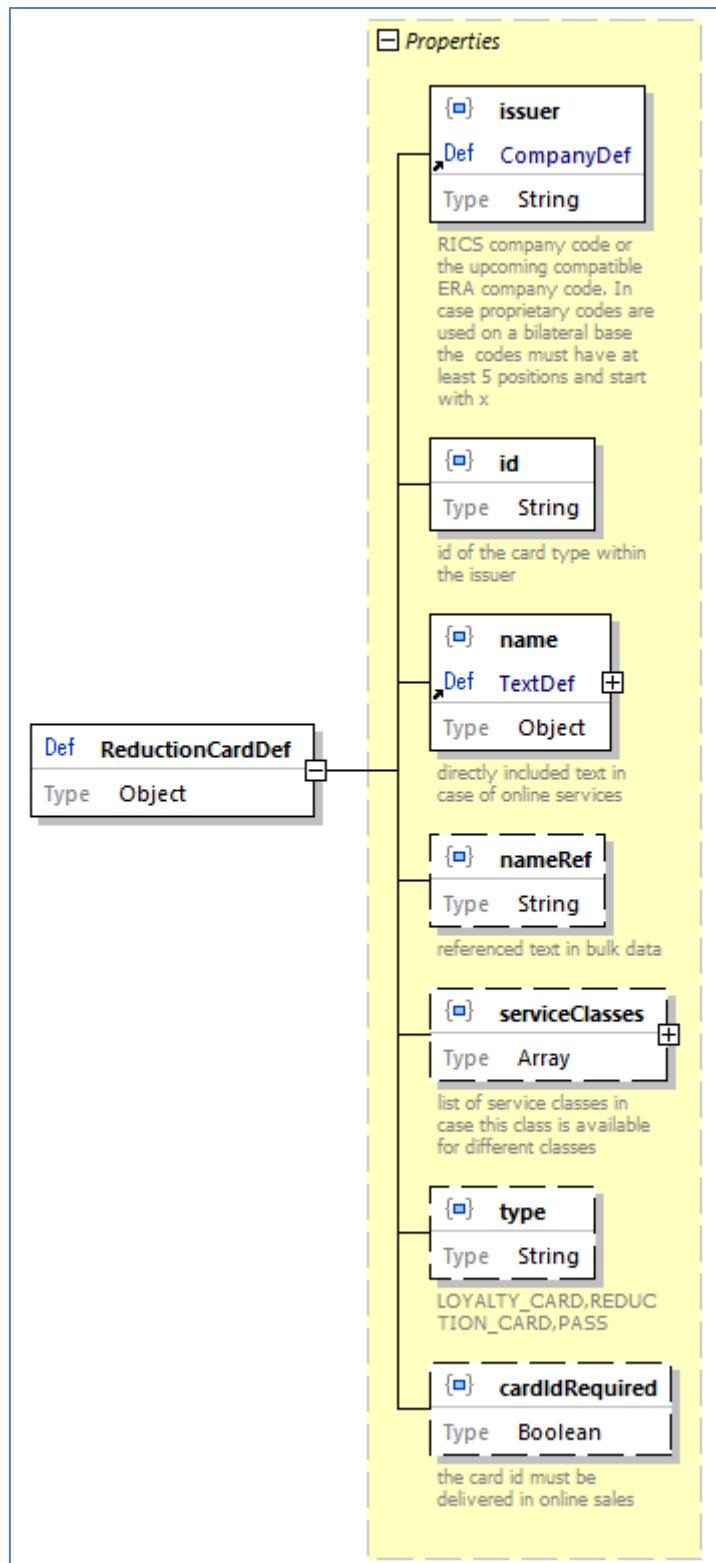


Figure 27 reduction cards offline

## ReductionConstraint

A fare associated with this constraint requires one of the listed cards to be presented by the passenger on the trip. Card Ids can be taken from the listed cards provided within the fare data delivery or from the common code list in **Reduction “cards”**.

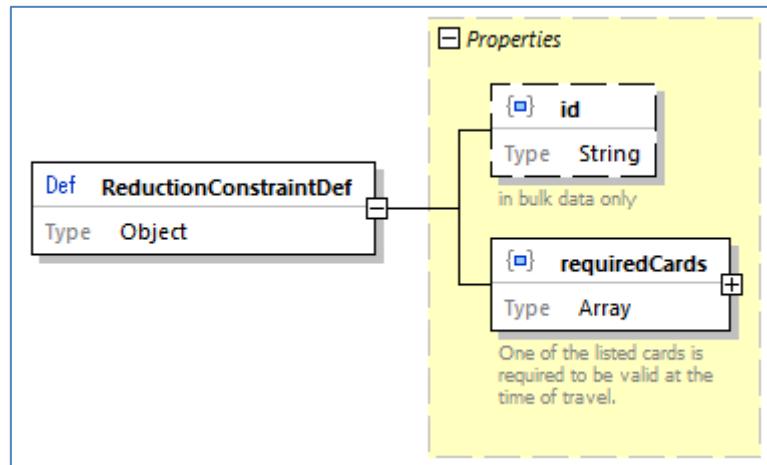


Figure 28 ReductionCardConstraint data structure offline

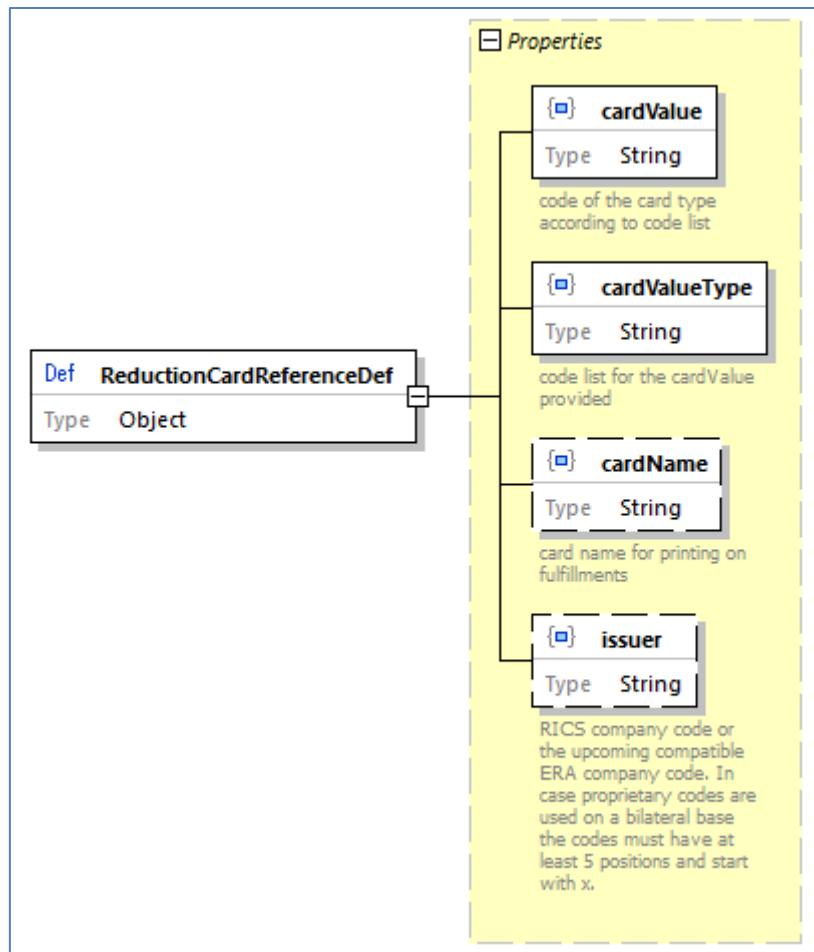


Figure 29 ReductionCardReference data structure

## RegionalConstraint

Definition of a regional validity of a fare. The regional validity constraint is defined by an entry connection point and an exit connection point to combine this regional validity with other regional validities of other carriers and the specification of the regional validity that is sued and described in IRS 90918-4 for ticket control. The entry or exit connection point might be missing in case the fare cannot be combined or can be combined on one side only.

Content	description
entryConnectionPoint	Defines the connection point for connecting this fare at the start of regional validity (see <code>ConnectionPoint</code> )
exitConnectionPoint	Defines the connection point for connecting this fare at the start of regional validity (see <code>ConnectionPoint</code> )
regionalValidity	Definition of the regional validity as defined in IRS 90918-4. I provide data structures for zones, Lines, train links, geographical polygons and routes.

The connection points are included for combining regions. When combining two regional validities from two carriers the connection points will disappear in the combined data structure for bar codes and ticket control and from the textual description for the passenger.

E.g.:

Carrier 1: RegionalConstraint {Exit (A,B), RegionalValidity X – Y/Z- A}

Carrier 2: RegionalConstraint {Entry (A,B), RegionalValidity B – C/D – E}

→ X\*Y/Z\*A\*B\*C/D\*E

The allocator might need to remove doubled stations in routes in case the connection point is a real station used in both regional validity descriptions in case it is displayed as one combined text:

Carrier 1: RegionalConstraint {Exit (A), RegionalValidity X – Y/Z- A}

Carrier 2: RegionalConstraint {Entry (A), RegionalValidity A – C/D – E}

→ X\*Y/Z\*A\*C/D\*E → X\*Y/Z\*A\*C/D\*E

Connecting Regional Validity to trips:

The regional constraint is connected to the timetable via the regional validity, the connection points are used to combine regional constraints.

To support legacy implementations the connection points can provide a border point code linked with the timetable.

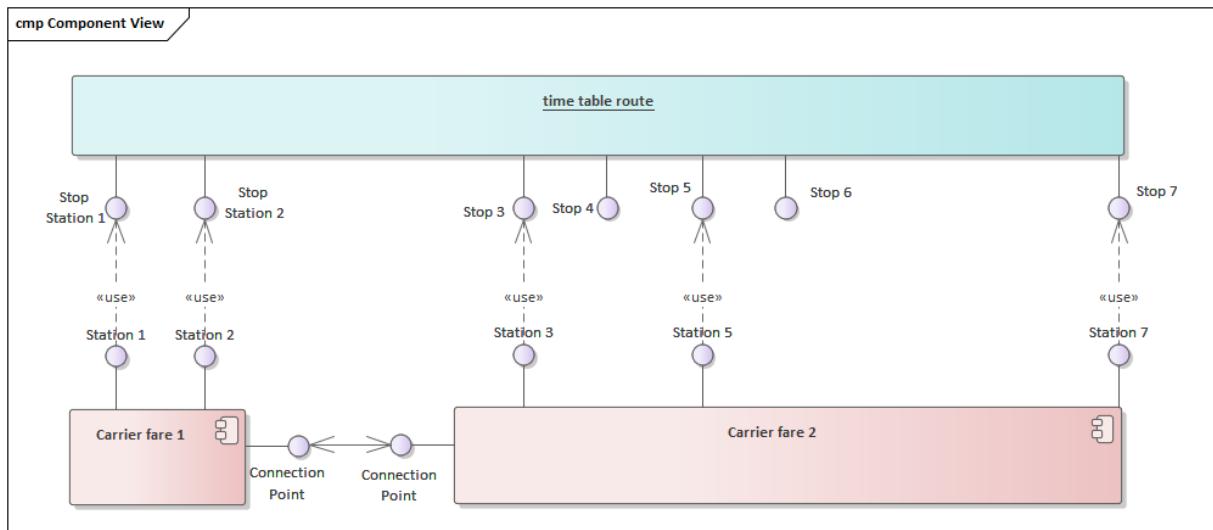


Figure 30 Connection points and timetable routes

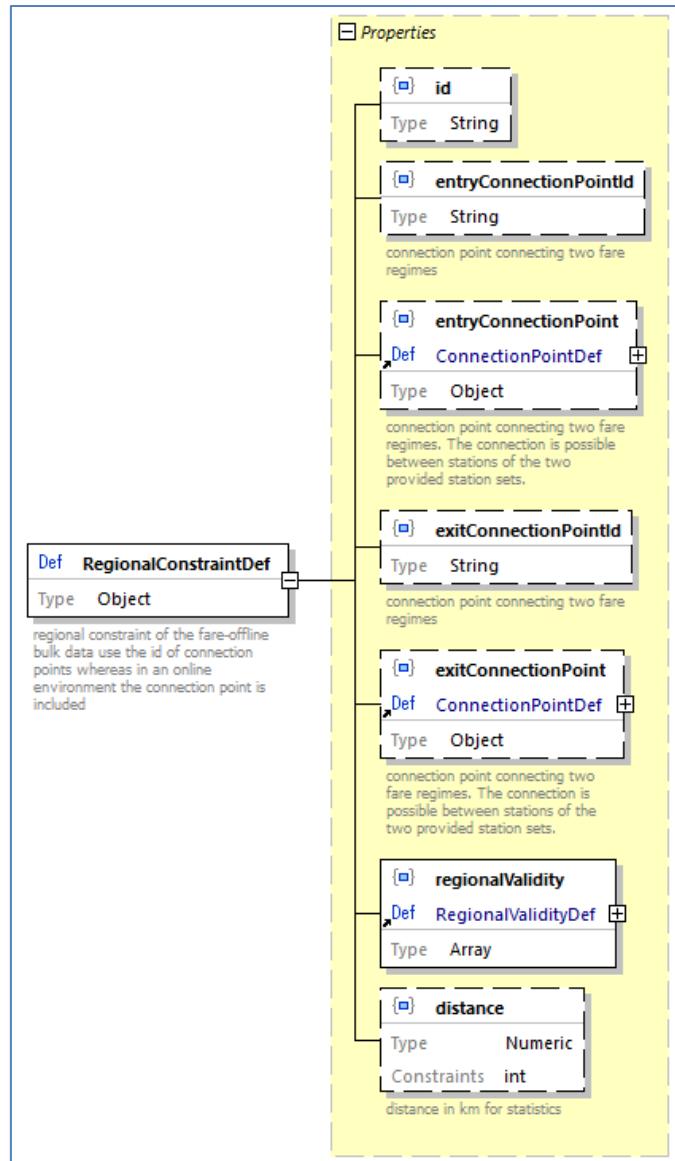


Figure 31 Regional validity constraint data structure

The online data structure will not use the id and will directly include the entry and exit connection point, whereas the offline structure will include the id of the connection point pointing to a connection point within the same data delivery..

The regional validity contains also content that is applicable to synchronous data transfer only (e.g. train links for train bound offers).

The data structure “RegionalValidity” is defined in UIC IRS 90918-4 and included by reference only. It provides a sequential list of region definitions that can be defined as zones, lines, train links (online version only) geographical areas (polygons) and route descriptions (via-stations). The route description is extended to include fare reference station sets within the route.

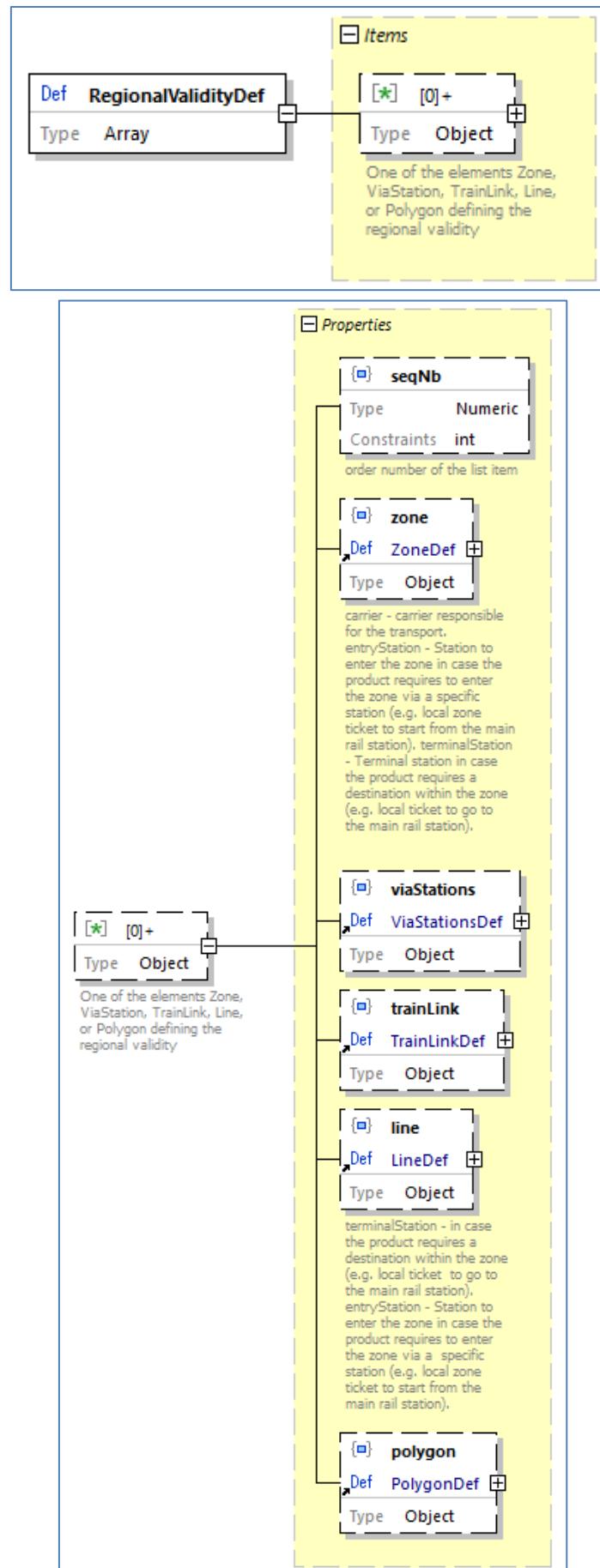


Figure 32 regional validity data structure

Extended route data structure including fare reference station sets.

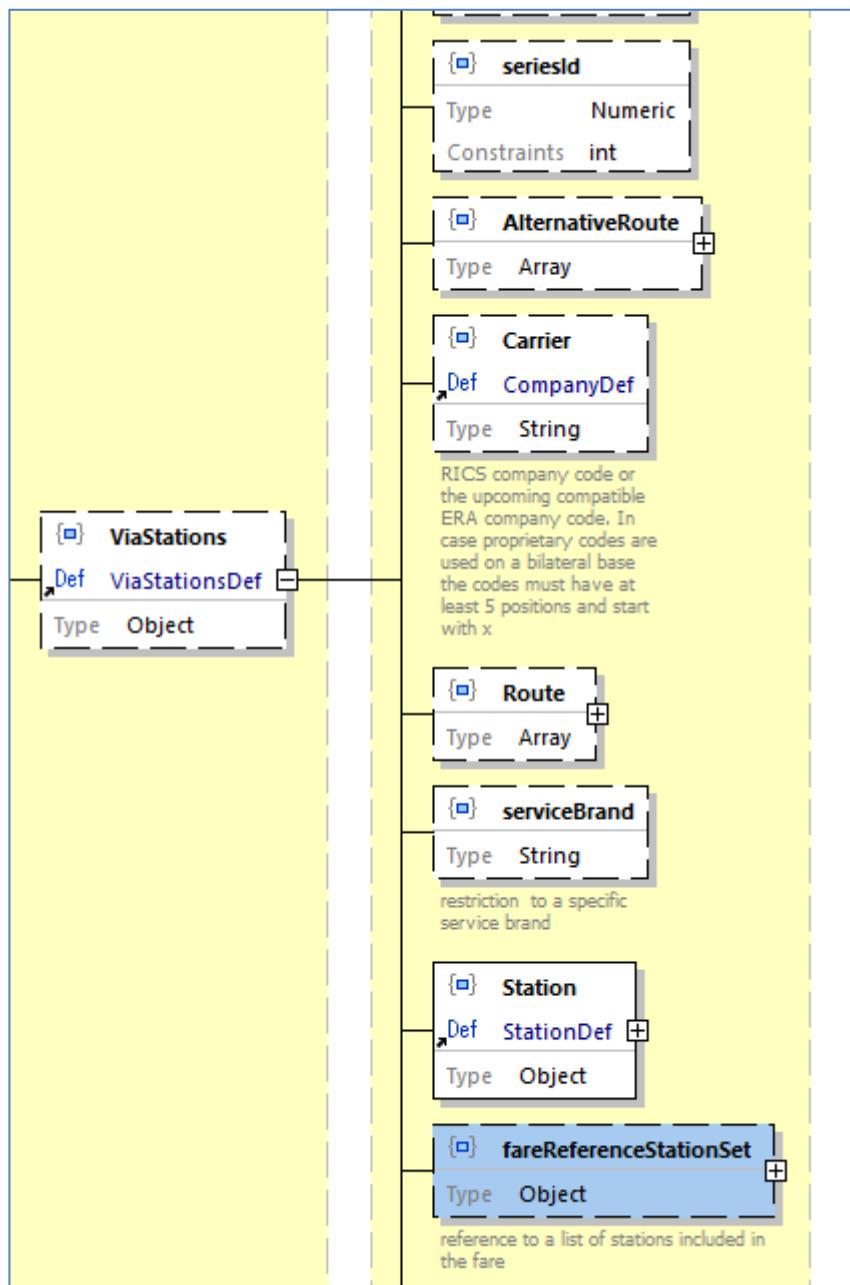


Figure 33 route description (ViaStation) data structure

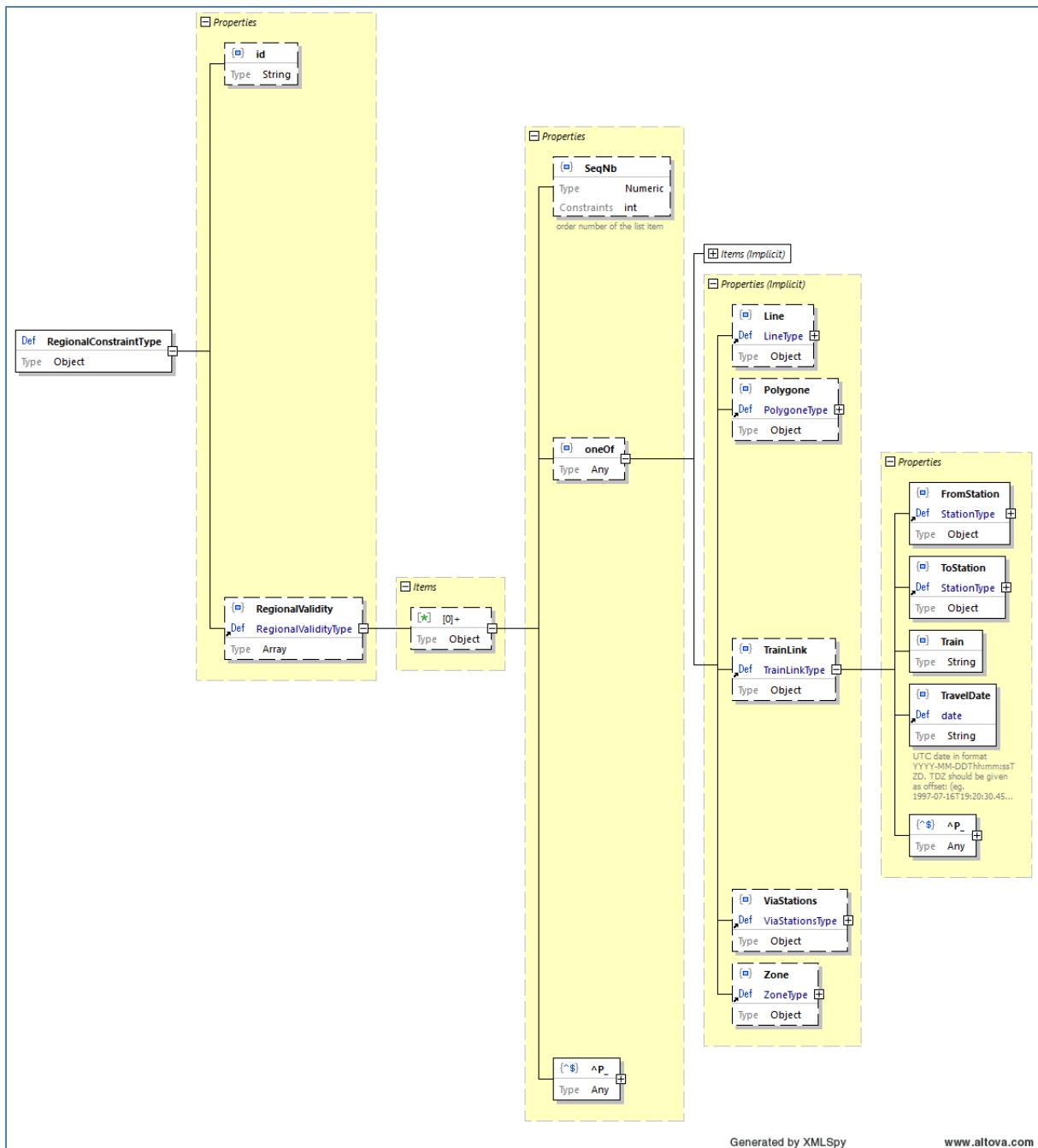


Figure 34 `RegionalValidity` data structure - copy of 90918-4

Generated by XMLSpy

[www.altova.com](http://www.altova.com)

## ReservationParameter

ReservationParameter provide data on how to combine reservations with NRT fares, how to book reservations via the 90918-1 interface and which options a passenger has for reservation.

reservationRequired	A reservation must be made accompanying an NRT ticket.
reservationParameters981-1	Parameters to request the correct reservation using the interface according to IRS 90918-1
reservationOptions	<p>Reservation options available that would not change the offer (same price and conditions) (e.g. Aisle or Window).</p> <p>The information is static and does not mean that such an option is still available.</p> <p>The preferences are grouped in case a selection is required (Aisle or Window).</p>

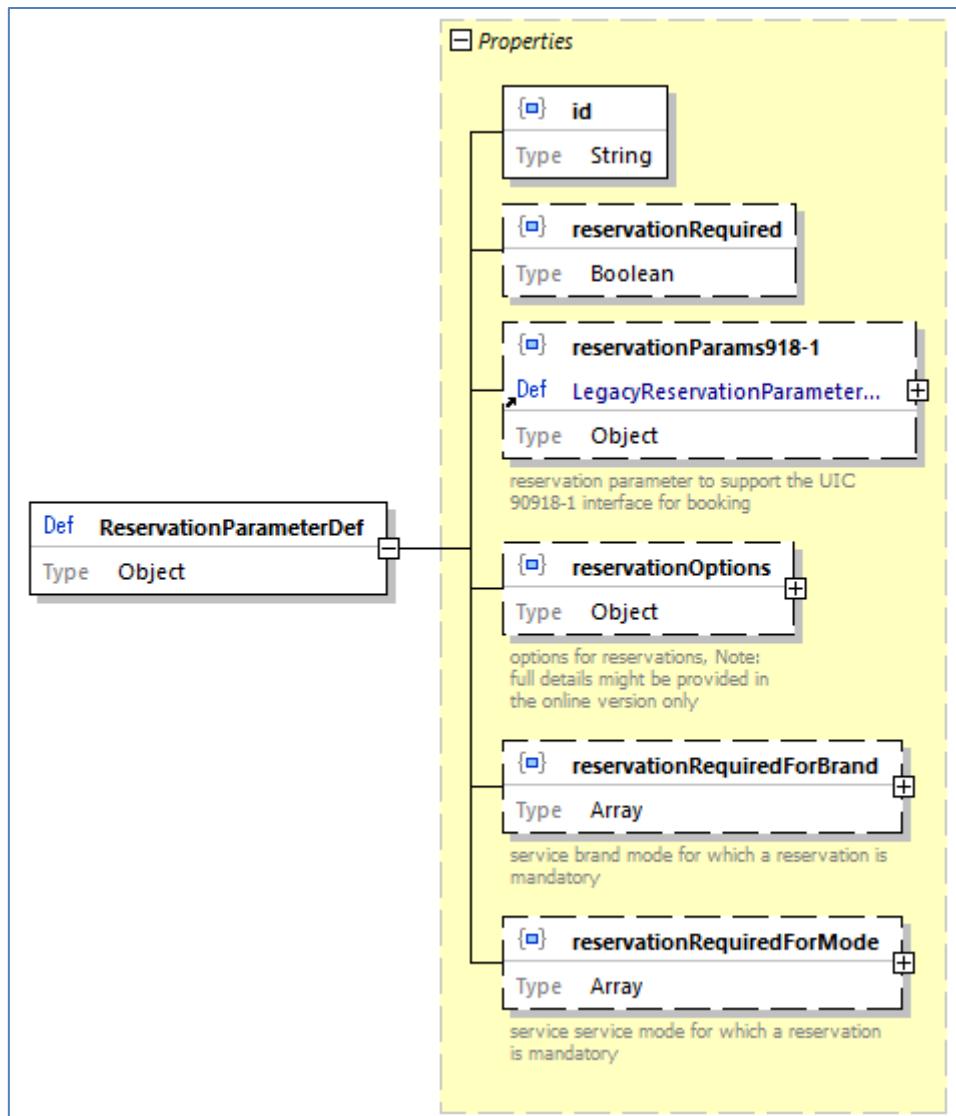


Figure 35 ReservationParameter data structure

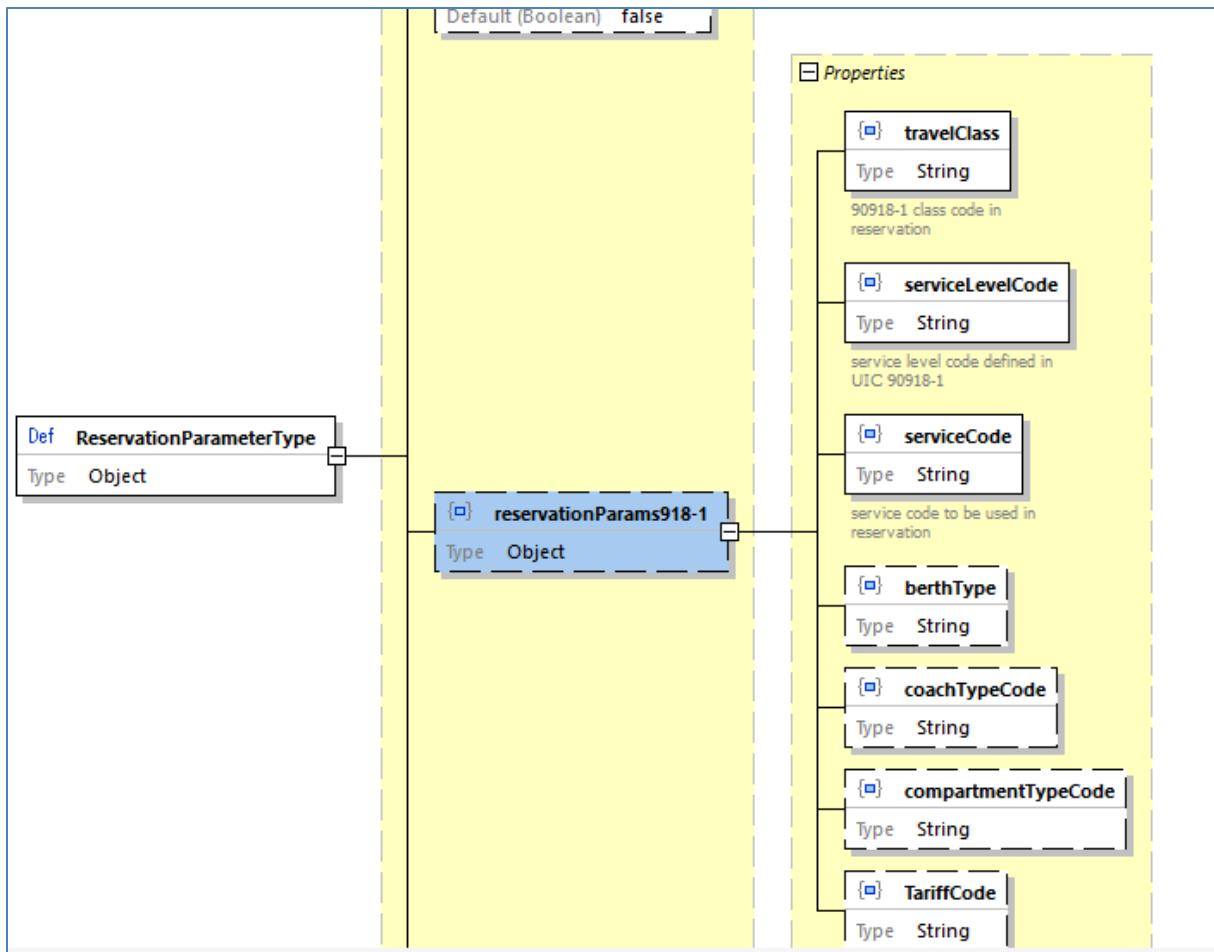


Figure 36 Reservation Parameter - support for 90918-1 reservation interface

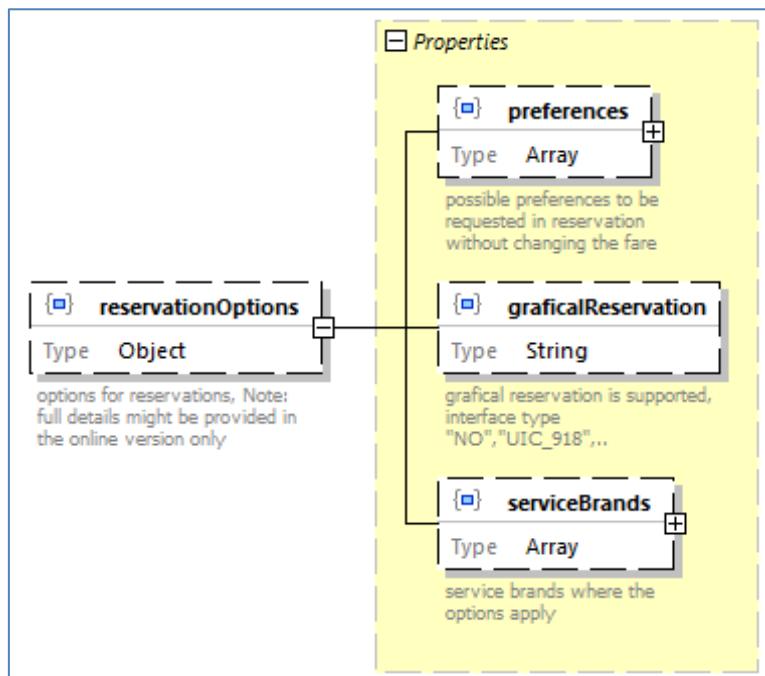


Figure 37 ReservationParameter data structure - reservation options

- Code list Preference Groups: see Preference groups
- Code list Preferences: see Preferences of places

## StationDetail

Details on stations including codes and names. Codes must include the MERITS code in case it is defined for a station.

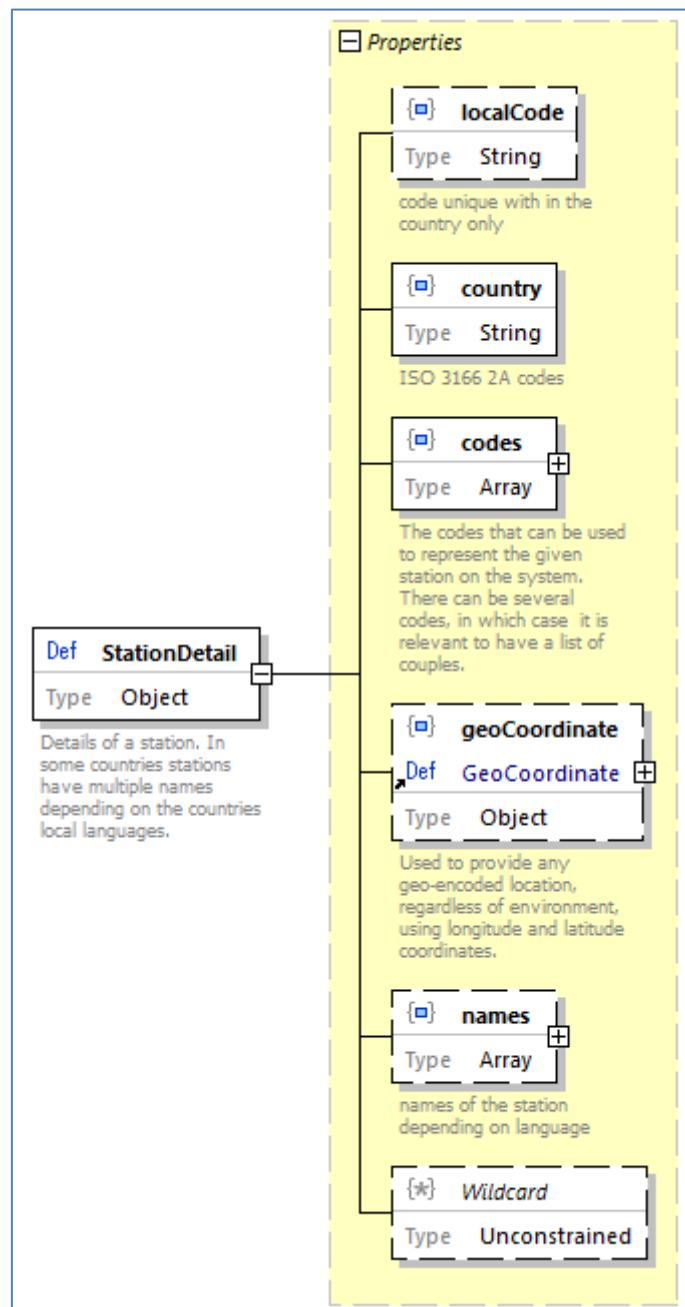


Figure 38 structure Station detail data

## StationNames

Station names provides multilanguage names in short and long form as currently no other data source can provide these names. Short names are used within the route descriptions whereas the long for is used for entry and exit stations.

A legacy border point code can be provided during the migration to the OSDM data model.

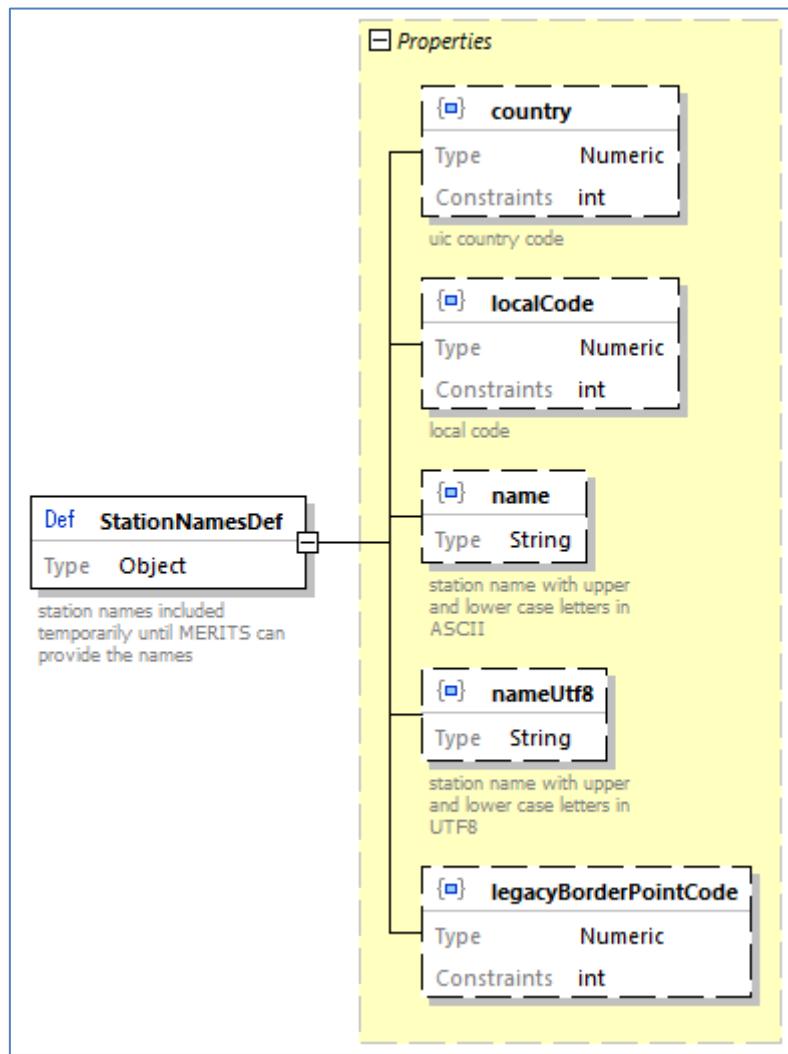


Figure 39 structure Station detail data (offline only)

## Text

Used for all textual descriptions where translations might be needed.

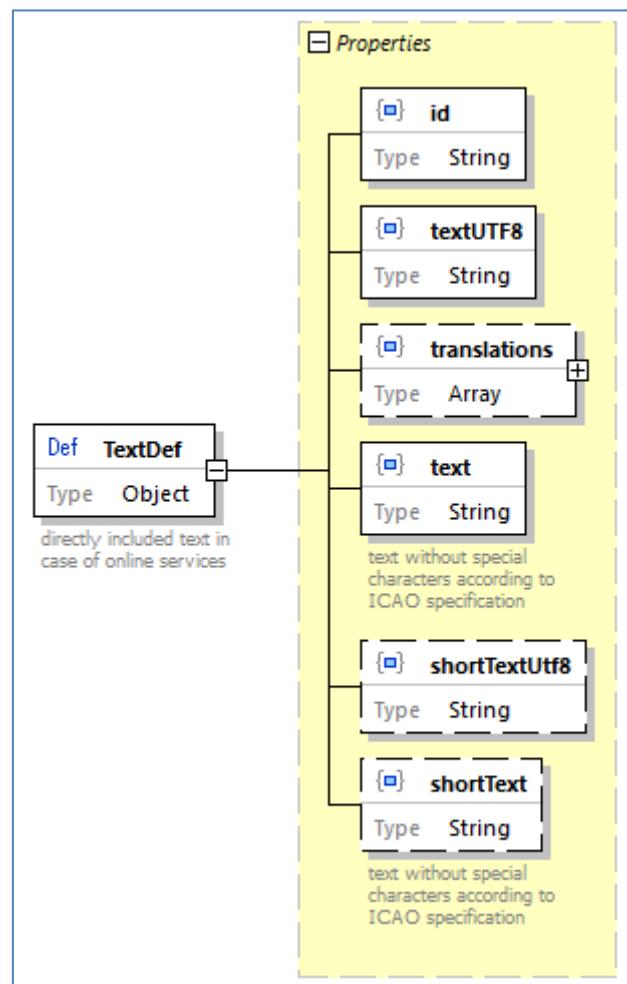


Figure 40: text data structure

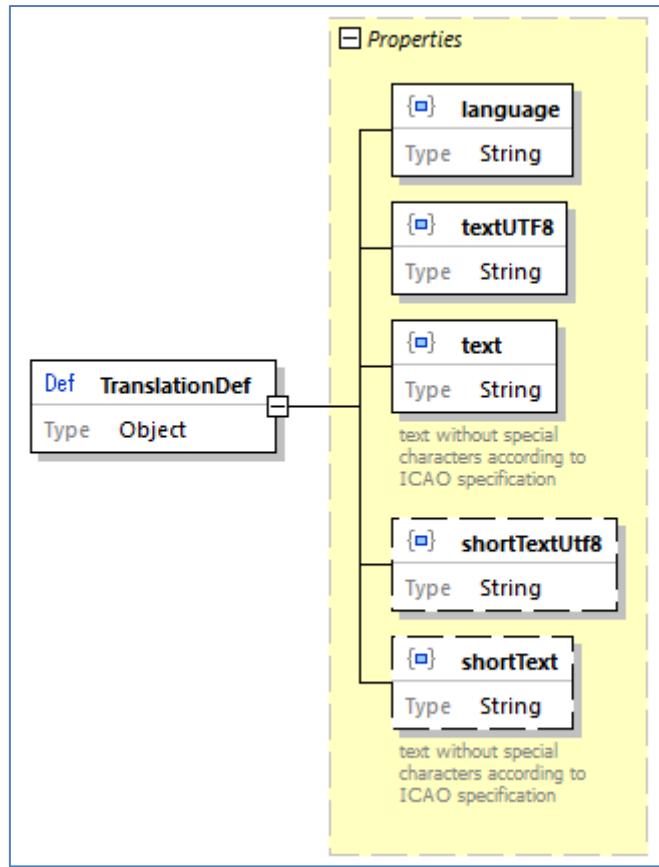


Figure 41 Translation data structure

## SalesAvailability

Sales availability defines the constraints on the time when a sale of a fare can start or end. The sales availability is used in the offline data exchange only. A constraint is provided as a list of salesRestrictions that have to be applied.

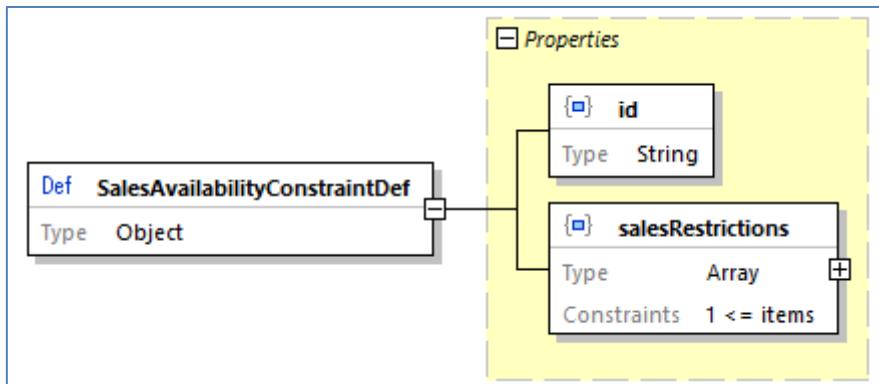


Figure 42: Sales availability structure

Sales restrictions can define a start and end of the sale relative to the date of sale or the date of travel.

A reference to a calendar can be provided to indicate all sales dates.

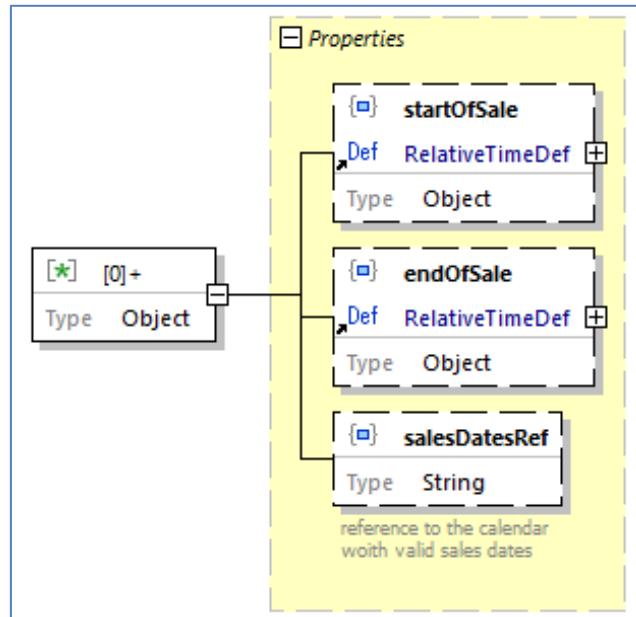


Figure 43 SalesAvailability data structure - calendar

Data constraints	
startOfSale, endOfSale	$\text{startOfSale} < \text{endOfSale}$

## ServiceClass

ServiceClass provides textual descriptions for the predefined service classes.

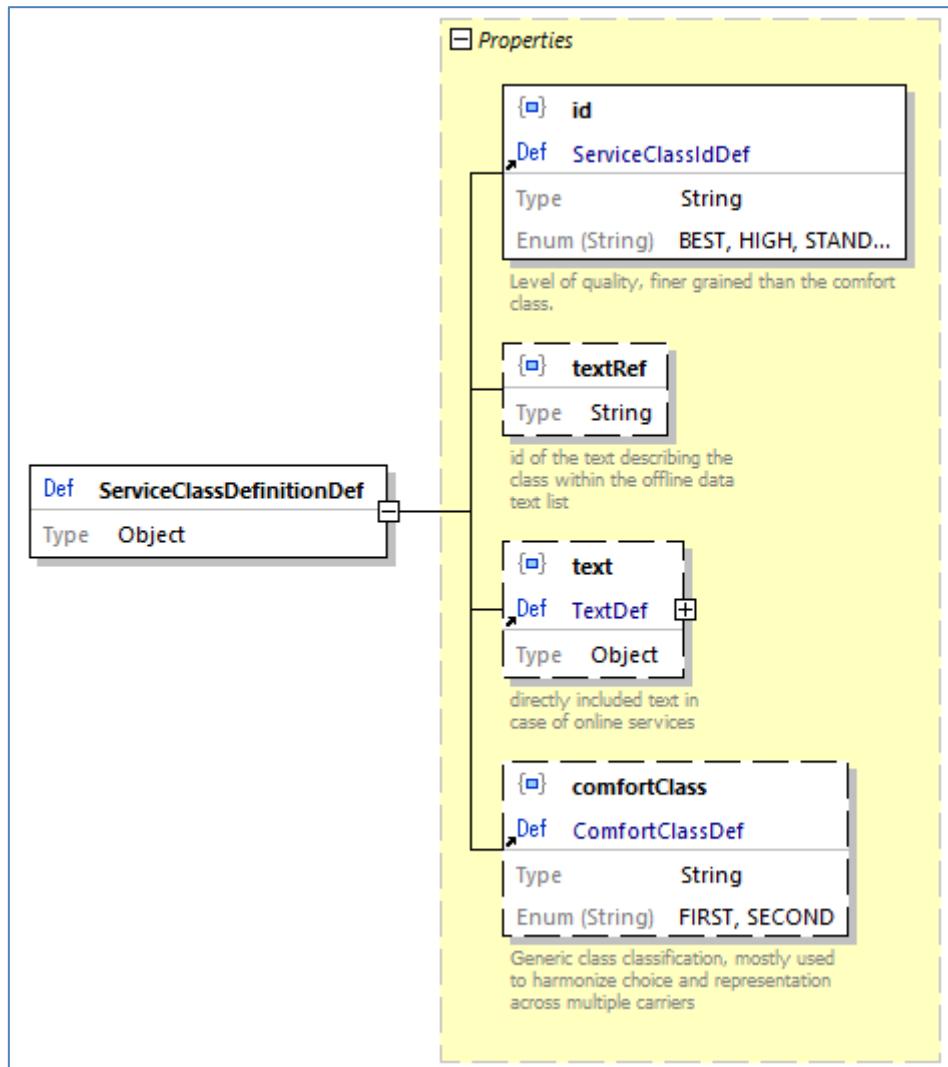


Figure 44: Service Class

## ServiceConstraint

The service constraint limits a fare to specific service brands (train types). The constraint can either be defined as a list of service brands included or as a list of service brands excluded for the fare.

The online data structure will not provide the id.

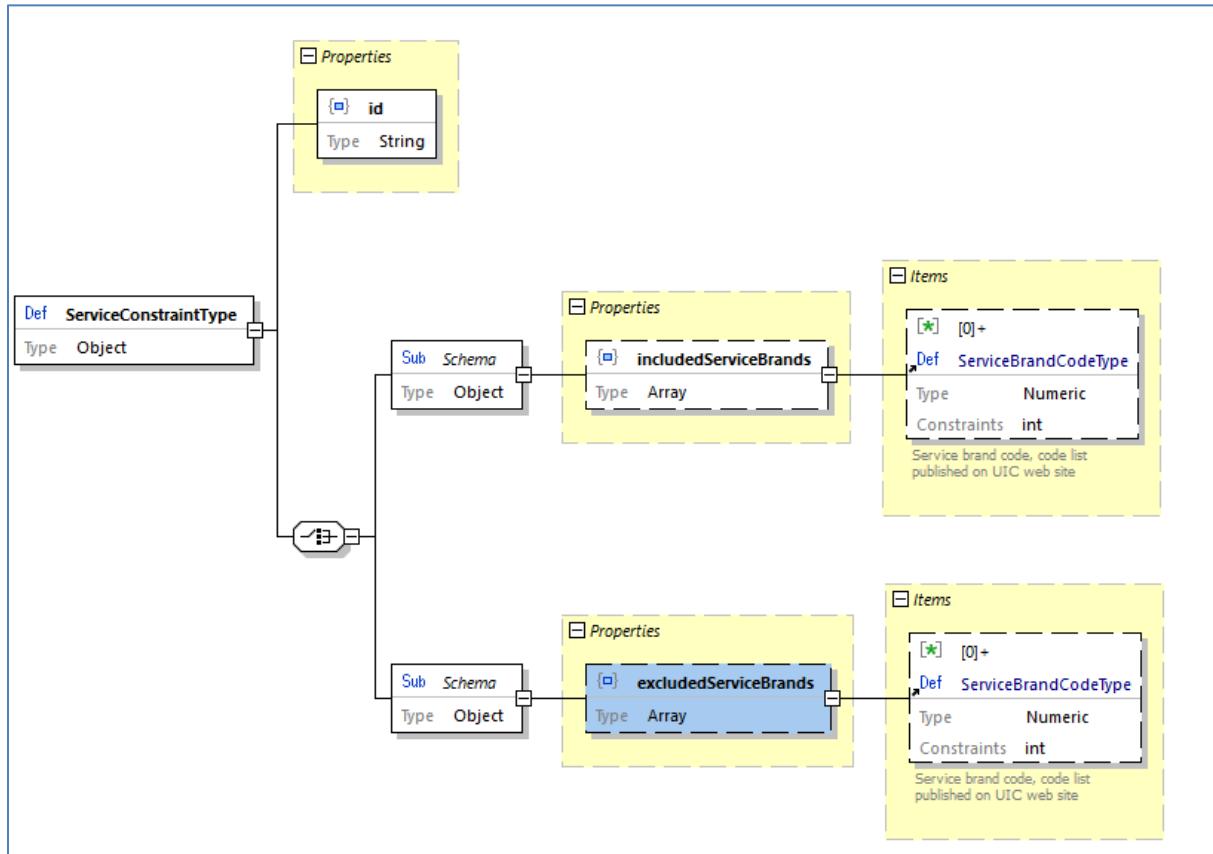


Figure 45 service constraints

Data constraints	
includedServiceBrands, excludedServiceBrands	Only one of the lists can be used. Using both lists is forbidden.

## **ServiceLevel**

The service level data provide additional informations (e.g. text) in the offline data exchange in case the reservation API IRS90918-1 is used.

Description of a service level. The service level defines a specific product on a train which can have a price (e.g. Double places with shower, ...). It is more specific than just the classic travel class.

The available service levels are defined in IRS 90918-1 element 308 (Service level code). The data indicate the service class that needs to be booked in case the reservation is not an IRT and parameters needed for reservation via the 90918-1 interface.

Some service levels might require a mandatory reservation.

Additional to a service level there might be reservation options that do not affect the price. There are listed in reservation options. (e.g. Upper or lower berth in the service level for double Sleeper compartment).

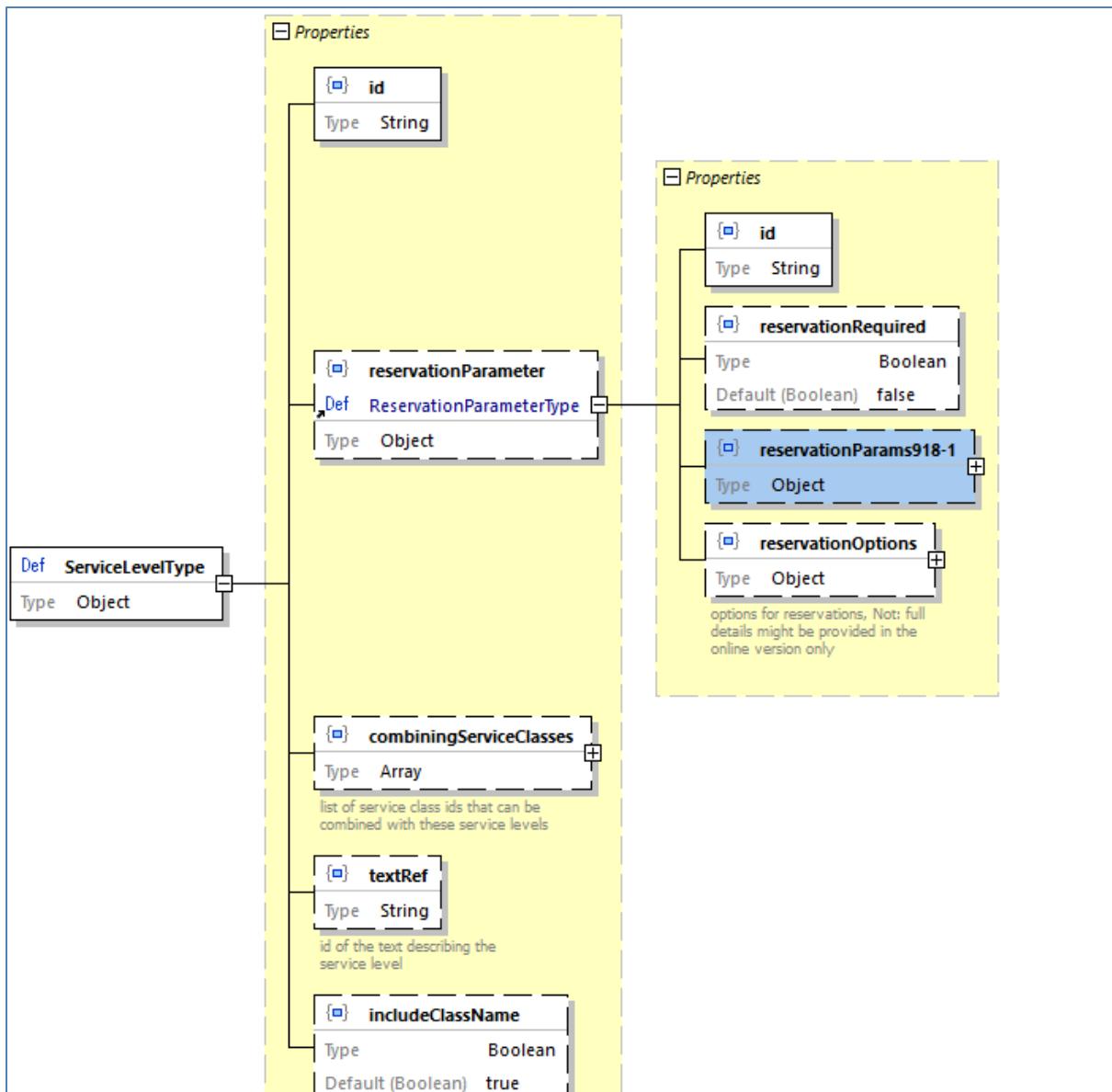


Figure 46 Service level description (offline)

## TravelValidityConstraint

The travel validity constraint defines at which times the passenger is permitted to travel.

Data	
<code>travelValidity</code>	Calendar defining the days on which travelling is allowed
<code>validityRange</code>	Duration of validity for a travel contract including extensions for some hours after midnight.
<code>ExcludedTimeRange</code>	Time range(s) excluded from travelling
<code>numberOfTravelDays</code>	Number of allowed travel days
<code>returnConstraint</code>	Restrictions on the time between the trip and the return trip.

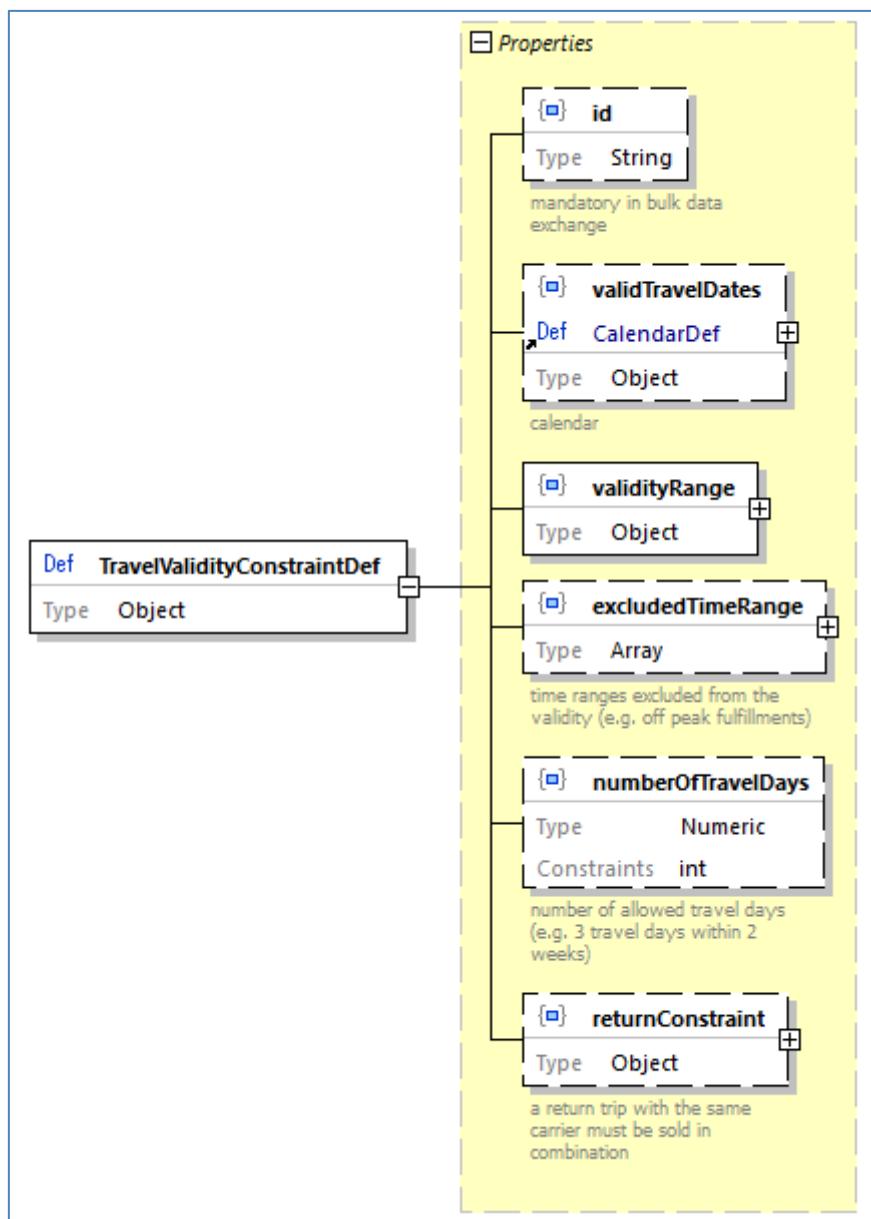


Figure 47 travel validity data structure

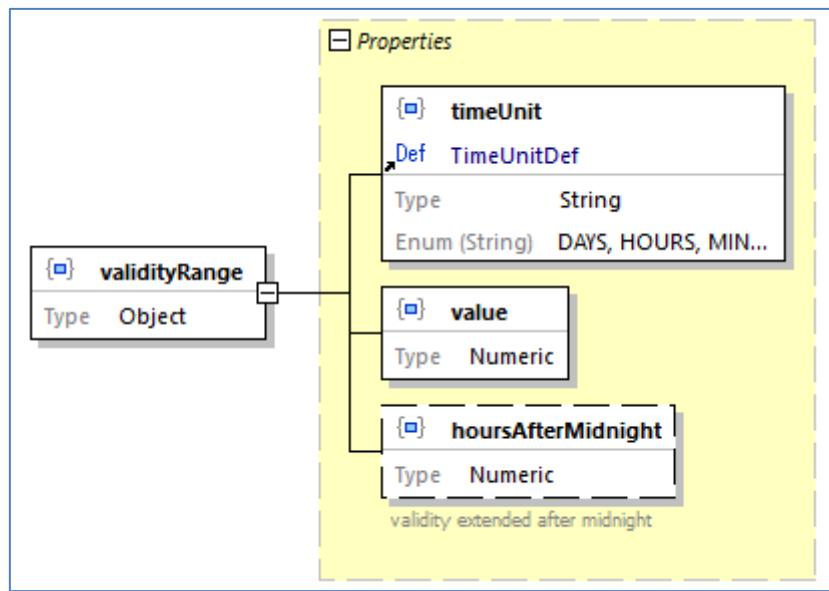


Figure 48 travel validity data structure – validity range

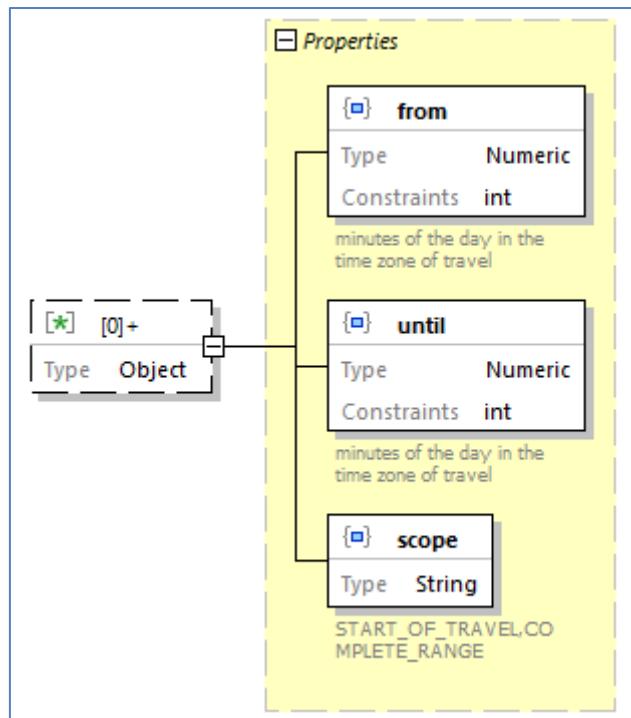


Figure 49 travel validity data structure – excluded time range

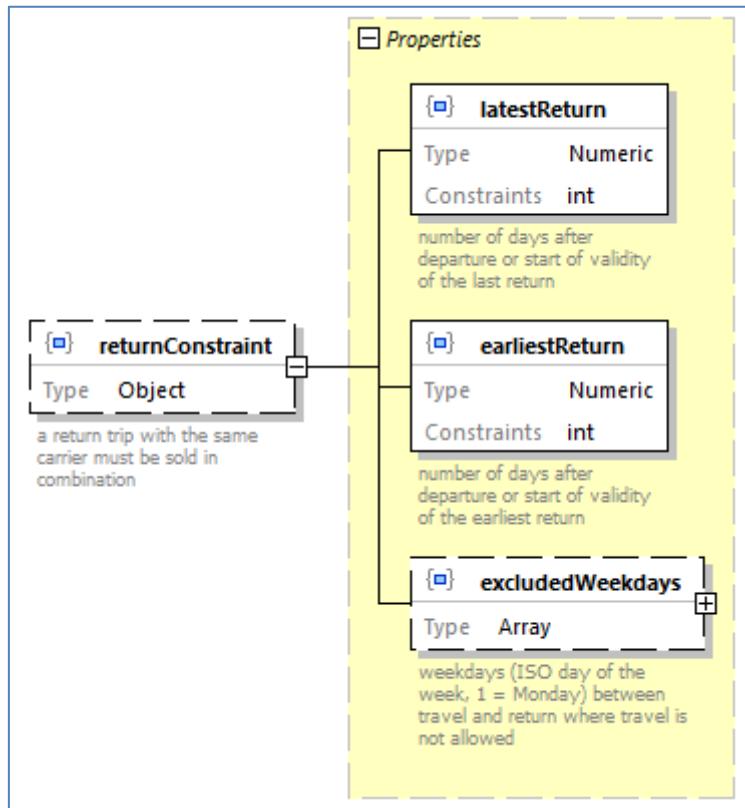


Figure 50 travel validity data structure- return constraint

Data constraints	
ExcludedTimeRange	from time < to time
numberOfTravelDays	A duration must be provided
returnConstraint	earliestReturn < latestReturn

## ZoneDefinition

Definition of zones used in regional validity.

The area of a zone can be defined by either a list of stations, geographical polygon or a list of NUTS codes. Multiple definitions are allowed in case they define the same area.

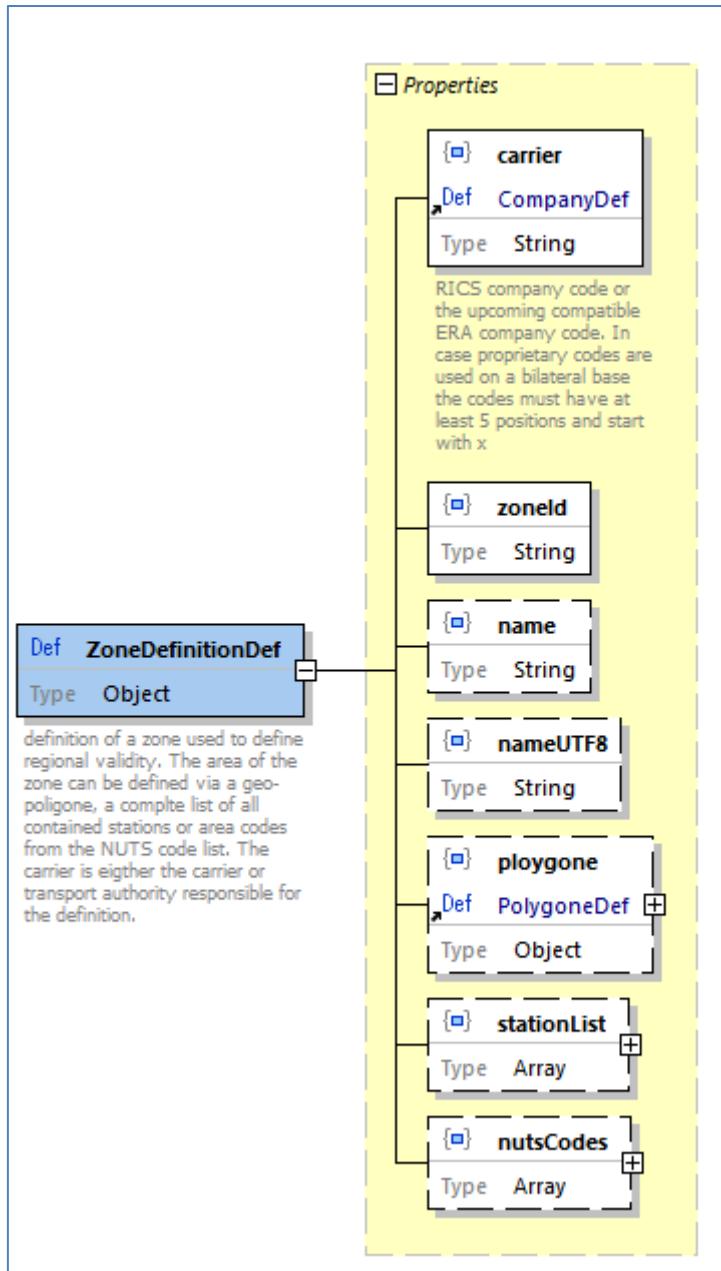


Figure 51 Zone definition structure

# Offline Sale and Distribution

## General Scenarios

Within the offline sales model the participating companies agreed to allow sales based on the provided fare data. The receiving company is responsible to apply the rules defined within the fare data. In case the implementation does not cover some features it is not allowed to sell fares that use these features.

### Fare exchange

Fares can be exchanged by bilateral file exchange, via a queue provided according to this specification or via some common exchange platform like the upcoming OSDM data exchange platform in case the company is a member of the platform.

Exchanged fare data deliveries can be defined to be implemented mandatory or to be optional thus allowing to continue the sales with the previous version. In case a mandatory version replaces a previous version it also replaces all previous optional version with-in the chain.

A data delivery might specify a minimal version number of the schema that needs to be supported to use the data.

## Versioning of data deliveries

The data delivery will contain the version number and the version number which is required to process the data. Also, a change in a minor version might restrict the usage of older version in case a carrier used a new optional feature which is mandatory to his fares.

### Automated bulk data exchange

Automated asynchronous bulk data transfer is an option implemented by queues. The queues must implement the AMQP 1.0 specification (<https://www.amqp.org/about/what> ).

On bilateral agreement other queue technologies might be used between two systems.

Queue authentication and encryption must use TLS version 1.2.

AMQP Header Parameter:

Parameter	Usage
message-id	Technical id of the data transfer, not the data delivery id in the data.
user-id	
to	
subject	„fare-data-delivery_<version>

reply-to	N/A
correlation-id	N/A
content-type	application/json
absolute-expiry-time	1 year ahead
creation-time	Time stamp when the data are put to the queue
group-id	
group-sequence	
reply-to-group-id	

## Asynchronous fare data delivery

The fare structure delivery is the bulk data object collecting the fare data (“FareStructure”) of a delivery and the delivery meta data (“Delivery”).

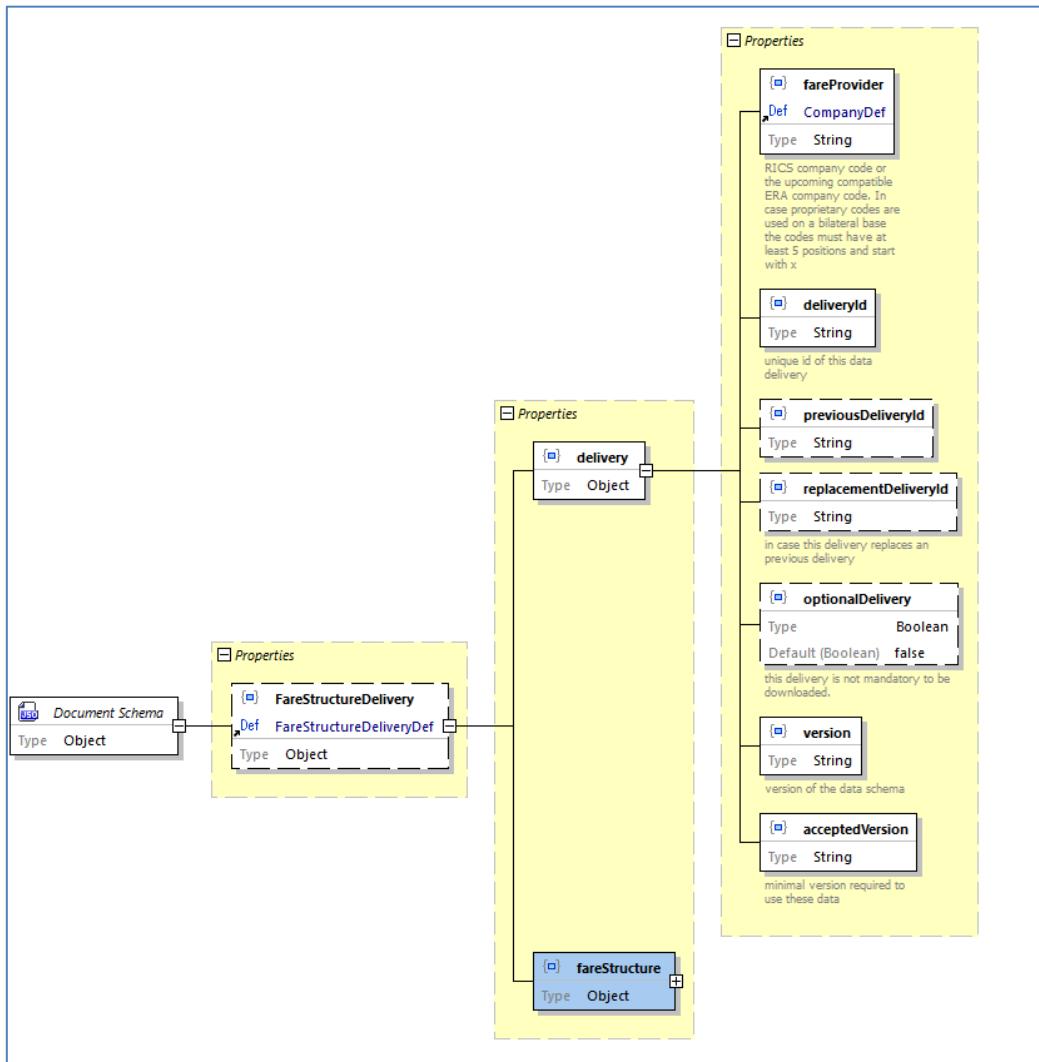


Figure 52 Data structure for bulk data

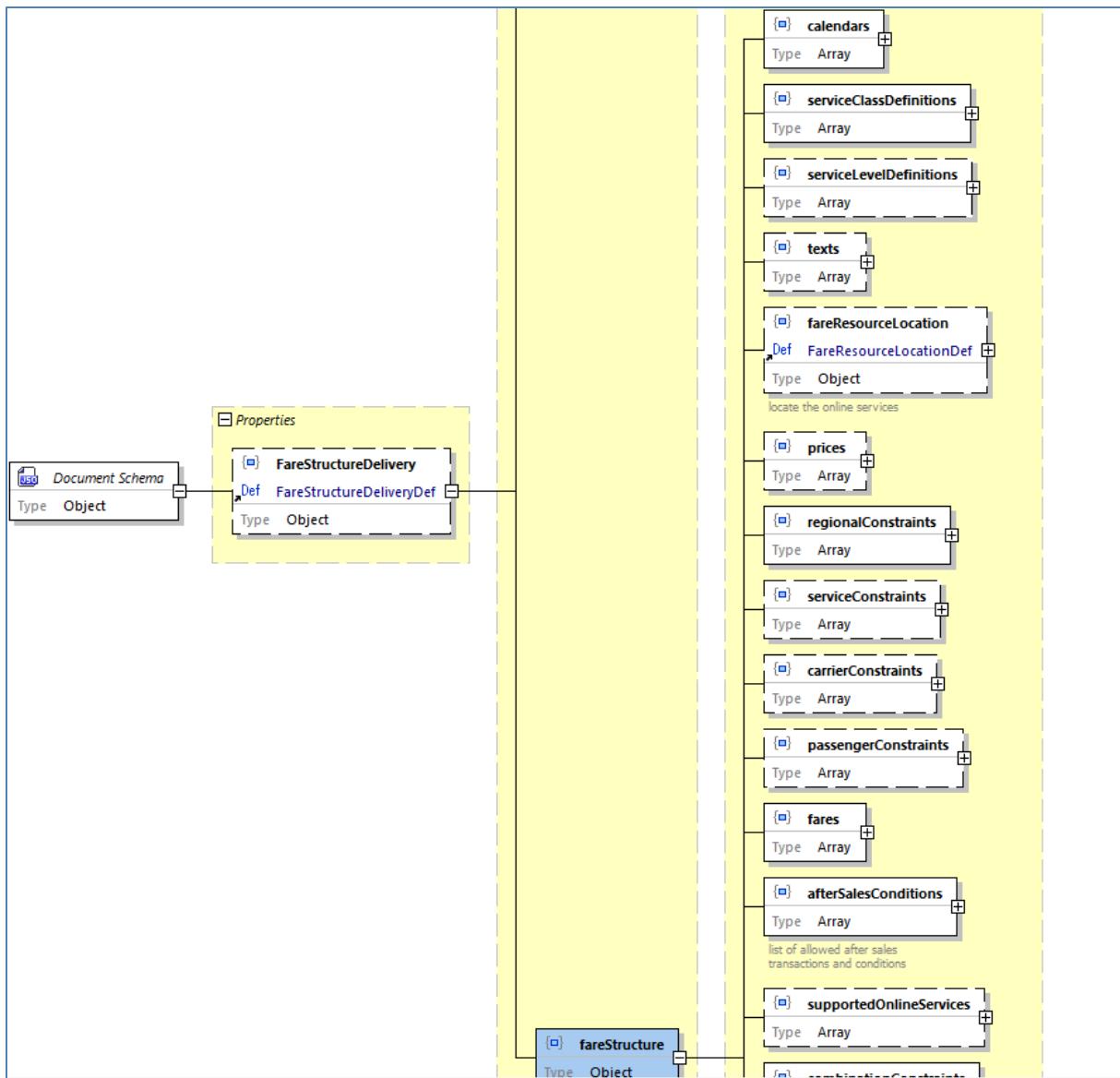


Figure 53 fare structure part 1

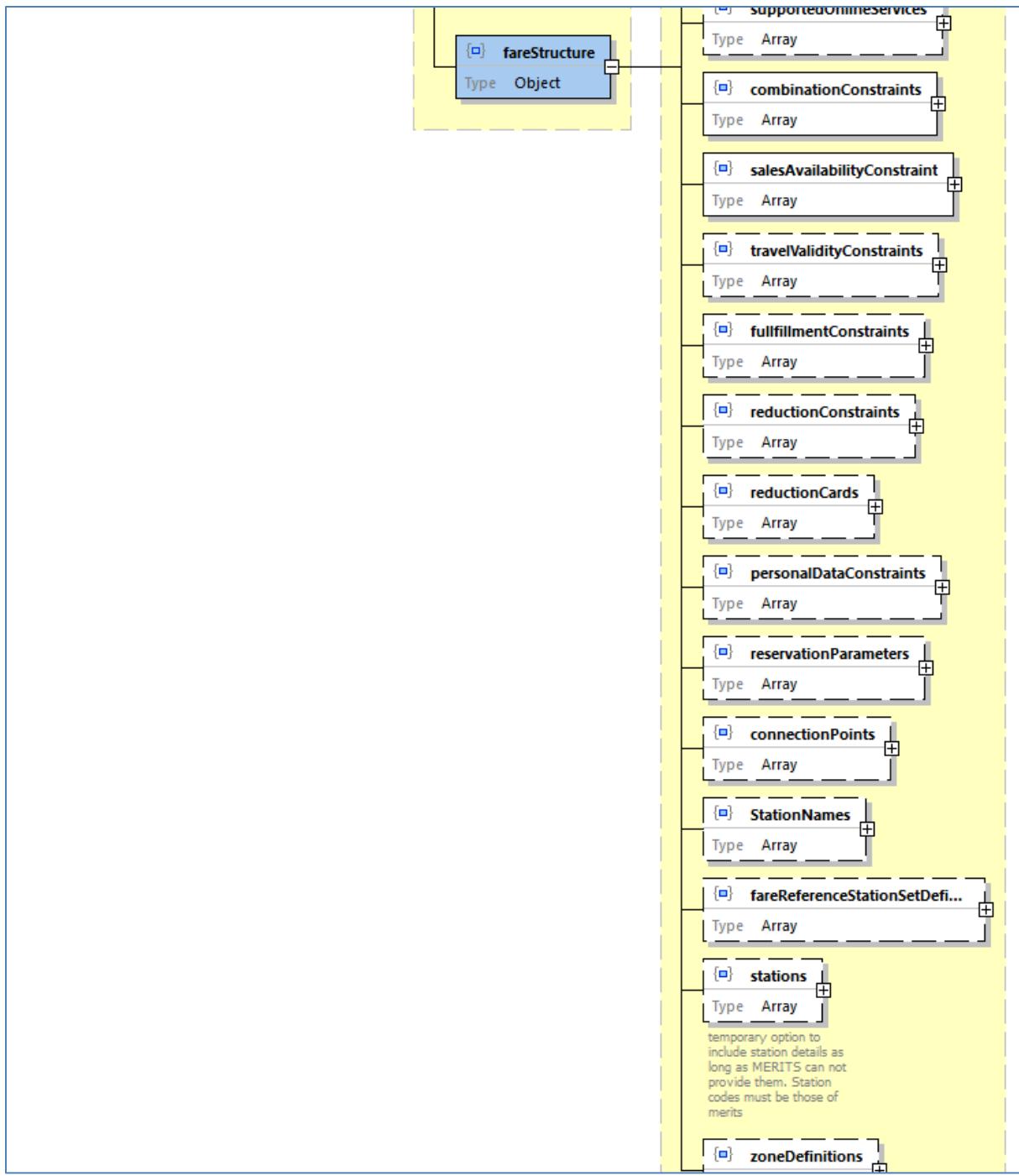


Figure 54 fare structure part 2

# Online Sale and Distribution

## Roles: Fare Provider, Allocator & Distributor

A participant can play the following roles:

- **Fare Provider** – defining the fare and combination rules and providing them to allocators offline or online.
- **Allocator** – combining fares, defining after sales rules within the frames set by the fare provider, providing combined offers and managing the booking transaction, managing the ticket security (barcode, control processes), managing compensations processes, managing the stock control process.
- **Distributor** – selling tickets from one or more allocators to the customer. Selecting the allocators and joining multiple independent bookings.

A RU or system provider can support more than one role.

## General scenarios

### Fare Provider Role

The services defined to exchange fares online are designed to support general sales and after sales scenarios.

Here the general flow is listed, and the supporting services of the fare exchange are indicated as well as other services of UIC specifications.

#### *Distribution Scenario – Sales*

##### Timetable search

Timetable data are not in the scope of this specification. It is assumed, that the allocator has retrieved a valid travel solution from a timetable search engine before requesting fares.

##### Offer

Requesting fares of a carrier or train for a specified set of passengers. Splitting of the entire travel solution into appropriate parts is the task of the allocator. Data supporting the selection of the carriers/fare providers to be requested are part of this specification and can be transferred beforehand (see **FareResourceLocation**).

The carrier should deliver all applicable fares for the requested offer and the rules for combining them.

The prices provided within the offer might be fixed for a specific time range depending on the fare, but this is not mandatory. Some carriers don't guarantee prices at all.

In case of offers with reservation the offer reply will contain the availability of the different fares and services, but no availability on other place features (e.g. window, aisle, ...) if they don't affect the price.

In case a trip description needs to be changed as the carrier/ fare provider has more accurate data on the trip the reply contains a new trip and an indication of change.

## Preliminary Booking

The preliminary booking allocates the fare for a period defined by the carrier. Technically it is implemented as a post /booking using the offerId.

For an intermediate time the booking and fulfillment of reservations might be implemented using IRS 90918-1 and not via 90918-10. In this case only the offer service is used.

In case of reservations the currently defined flow in IRS 90918-1 is different, as the preliminary booking must be implemented by the allocator receiving the reservation by cancelling the allocated fare and places in due time and with a specific reason for cancellation.

The current 90918-1 service require a detailed specification of fares and place features. The fare data of the offer defined in this specification will provide these data to allow an integration with the existing services of reservation systems.

## Confirm Booking

The booking is confirmed by the allocator to the carrier. This is the trigger for the accounting processes between the allocator and the carrier.

## Fulfillment

For combined tickets the allocator creates the ticket and is responsible to for the fulfilment. The fare provider might provide additional security features to be included within the fulfillment of the allocator.

The allocator is responsible to provide the ticket data for the control by the carriers. This is not in the scope of this specification but defined in IRS 90918-4.

## Place Selection

Place selection is implemented as a patch to the selected offer.

Place selection does not allocate places.

For an intermediate time the booking and fulfillment of reservations might be implemented using IRS 90918-1 and not via 90918-10. In this case only the offer service is used.

The current 90918-1 service require a detailed specification of fares and place features. The fare data of the offer defined in this specification will provide these data to allow an integration with the existing services of reservation systems. (see Reservation Parameter).

## Payment

Payment is not in the scope of this specification. The carrier/fare provider is not involved in this step.

## *After sales scenario - cancellation*

### Refund Offer

The fare description obtained with the original offer includes all information on cancellation fees in case the fees are not calculated by the allocator.

A refund offer from the carrier is therefore not required for the process.

A refund offer request can indicate a special reason for cancellation without fees due to an error of the allocator or unavailability of the service. The reason might not be accepted by the carrier.

The refund offer might indicate that the payment of the refund needs to be delayed checking whether the ticket has been used.

#### Confirm Refund Offer

A confirmed booking is cancelled.

#### *After sales scenario - exchange*

A cancellation offer is requested using special exchange reasons.

An offer for the new booking is requested with a reference to the old booking(s).

These covers:

- Exchange
- Upgrade
- Increase of passengers
- Decrease of passengers

#### Return Payment

No data exchange is foreseen in this step between allocator and carrier.

In case of a delayed payment of refunds the allocator needs to validate the ticket control data (IRS 90918-4) before the payment to the customer.

#### *Change of personal data*

##### Change personal data

The change of personal data can be provided as an optional feature.

Note: according to GDPR it must be possible to correct errors in personal data.

#### Reticket

The reticketing is not in the scope of this specification.

The allocator is responsible to provide the ticket data for the control by the carriers. This is not in the scope of this specification but defined in IRS 90918-4.

#### *Information on personal data*

In order to fulfill GDPR requirements information on stored personal data of the customer or passenger must be provided. As personal data are transferred from the allocator to the carrier the customer can ask at the issuer or allocator for information on the still stored personal data. The allocator then needs a possibility to request this information from the carrier.

To retrieve the stored personal data the booking has to be requested.

## Allocator Role

### *Distribution Scenario – Sales*

#### Timetable search

The allocator makes a time table search himself or retrieves the trip from a distributor. The allocator has to specify additionally for which part of the trip he requests the fares.

#### Offer

Requesting fares of a carrier or train for a specified set of passengers. Splitting of the entire travel solution into appropriate parts is the task of the allocator. Data supporting the selection of the carriers/fare providers to be requested are part of this specification and can be transferred beforehand (see **FareResourceLocation**).

The prices provided within the offer might be fixed for a specific time range depending on the fare, but this is not mandatory. Some carriers don't guarantee prices at all.

In case of offers with reservation the offer reply will contain the availability of the different fares and services, but no availability on other place features (e.g. window, aisle, ...) if they don't affect the price.

In case the fare provider delivers a changed trip the allocator has to take this into account. The reply contains a new trip and an indication of change.

#### Preliminary Booking

The preliminary booking allocates the fare for a period defined by the carrier. Technically it is implemented as a post /booking using the offerId. The allocator is responsible to manage the transactions including multiple carriers/fare providers.

For an intermediate time the booking and fulfillment of reservations might be implemented using IRS 90918-1 and not via 90918-10. In this case only the offer service is used.

In case of reservations the currently defined flow in IRS 90918-1 is different, as the preliminary booking must be implemented by the allocator receiving the reservation by cancelling the allocated fare and places in due time and with a specific reason for cancellation.

The current 90918-1 service require a detailed specification of fares and place features. The fare data of the offer defined in this specification will provide these data to allow an integration with the existing services of reservation systems.

#### Confirm Booking

The booking is confirmed by the allocator to the carrier. This is the trigger for the accounting processes between the allocator and the carrier.

In case the confirmation of a booking fails the allocator is responsible to restore a consistent status of the booking either by retries of the failed confirmation or by deleting the confirmed booking.

The allocator is responsible to delete bookings in case of failures and to repeat delete requests until the request is successful of the departure date of the trip has passed. The repletion of delete requests should avoid network bottlenecks (e.g. one retry every hour).

## Fulfillment

For combined tickets the allocator creates the ticket and is responsible for the fulfilment. The fare provider might provide additional security features to be included within the fulfillment of the allocator.

The allocator is responsible to provide the ticket data for the control by the carriers/fare providers. This is not in the scope of this specification but defined in IRS 90918-4.

## Place Selection

Place selection is implemented as a patch to the selected offer.

Place selection does not allocate places.

For an intermediate time the booking and fulfillment of reservations might be implemented using IRS 90918-1 and not via 90918-10. In this case only the offer service is used.

The current 90918-1 service require a detailed specification of fares and place features. The fare data of the offer defined in this specification will provide these data to allow an integration with the existing services of reservation systems. (see Reservation Parameter).

## Payment

Payment is not in the scope of this specification. The carrier/fare provider is not involved in this step.

## *After sales scenario - cancellation*

### Refund Offer

A refund offer from the carrier is required in case of online sales for the process

A refund offer request can indicate a special reason for cancellation without fees due to an error of the allocator or unavailability of the service. The reason might not be accepted by the carrier.

The refund offer might indicate that the payment of the refund needs to be delayed checking whether the ticket has been used.

### Confirm Refund Offer

A confirmed booking is cancelled. In case after sales fees for the carrier are applied that need to be transferred in the booking confirmation.

## Return Payment

In case of a delayed payment of refunds the allocator needs to validate the ticket control data (IRS 90918-4).

## *After sales scenario - exchange*

A exchange offer is requested using special exchange reasons.

An offer for the new booking is requested with a reference to the old booking(s).

These covers:

- Exchange
- Upgrade
- Increase of passengers

- Decrease of passengers

### **Return Payment**

No data exchange is foreseen in this step between allocator and carrier. The allocator has to pass on information on delayed payment to the distributor.

In case of a delayed payment or refunds the allocator needs to validate the ticket control data (IRS 90918-4) before the payment to the customer.

### ***Change of personal data***

#### **Change personal data**

The change of personal data can be provided as an optional feature.

Note: according to GDPR it must be possible to correct errors in personal data.

#### **Reticket**

The reticketing is not in the scope of this specification.

The allocator is responsible to provide the ticket data for the control by the carriers. This is not in the scope of this specification but defined in IRS 90918-4.

### ***Information on personal data***

In order to fulfill GDPR requirements information on stored personal data of the customer or passenger must be provided. As personal data are transferred from the allocator to the carrier the customer can ask at the issuer or allocator for information on the still stored personal data. The allocator then needs a possibility to request this information from the carrier.

To retrieve the stored personal data the booking has to be requested.

### **Distributor Role**

#### ***Distribution Scenario - sales***

##### **Timetable search**

The distributor makes a time table search himself or retrieves the trip from an allocator.

##### **Offer**

Requesting offers of an allocator for a specified set of passengers and optionally for a trip.

The prices provided within the offer might be fixed for a specific time range depending on the offer, but this is not mandatory. Some carriers don't guarantee prices at all.

In case of offers with reservation the offer reply will contain the availability of the different services, but no availability on other place features (e.g. window, aisle, ...) if they don't affect the price.

In case the allocator delivers a changed trip the distributor has to take this into account. The reply contains a new trip and an indication of change.

## Preliminary Booking

The preliminary booking allocates the offer for a period defined by the allocator. Technically it is implemented as a post /booking using the offerId. The allocator is responsible to manage the transactions including multiple carriers/fare providers.

## Confirm Booking

The booking is confirmed by the allocator to the allocator.

In case the confirmation of a booking fails the distributor is responsible to restore a consistent status of the booking either by retries of the failed confirmation or by deleting the confirmed booking.

The distributor is responsible to delete bookings in case of failures and to repeat delete requests until the request is successful of the departure date of the trip has passed. The repletion of delete requests should avoid network bottlenecks (e.g. one retry every hour).

## Fulfillment

The distributor requests fulfillments from the allocator and hands them to the traveler.

## Place Selection

Place selection is implemented as a patch to the selected offer.

Place selection does not allocate places.

## Payment

Payment is not in the scope of this specification. The allocator is not involved in this step.

## *After sales scenario - cancellation*

### Refund Offer

The distributor needs to get a refund offer from the allocator to start the refund process. The refund offer is part of the booking to be refunded.

A refund offer request can indicate a special reason for cancellation without fees due to an error of the sales staff or unavailability of the service. The reason might not be accepted by the allocator.

The refund offer might indicate that the payment of the refund needs to be delayed checking whether the ticket has been used.

### Confirm Refund Offer

A confirmed booking is cancelled.

### Return Payment

The distributor needs to obey the delayed payment instruction provided by the allocator.

## *After sales scenario - exchange*

An exchange offer is requested using special exchange reasons.

An offer for the new booking is requested with a reference to the old booking(s).

These covers:

- Exchange

- Upgrade
- Increase of passengers
- Decrease of passengers

#### Return Payment

The distributor needs to obey the delayed payment instruction provided by the allocator.

#### *Change of personal data*

##### Change personal data

The change of personal data can be provided as an optional feature.

Note: according to GDPR it must be possible to correct errors in personal data.

##### Reticket

The reticketing is not in the scope of this specification.

#### *Information on personal data*

In order to fulfill GDPR requirements information on stored personal data of the customer or passenger must be provided. As personal data are transferred to the allocator the customer can ask at the distributor for information on the still stored personal data. The distributor then needs a possibility to request this information from the carrier.

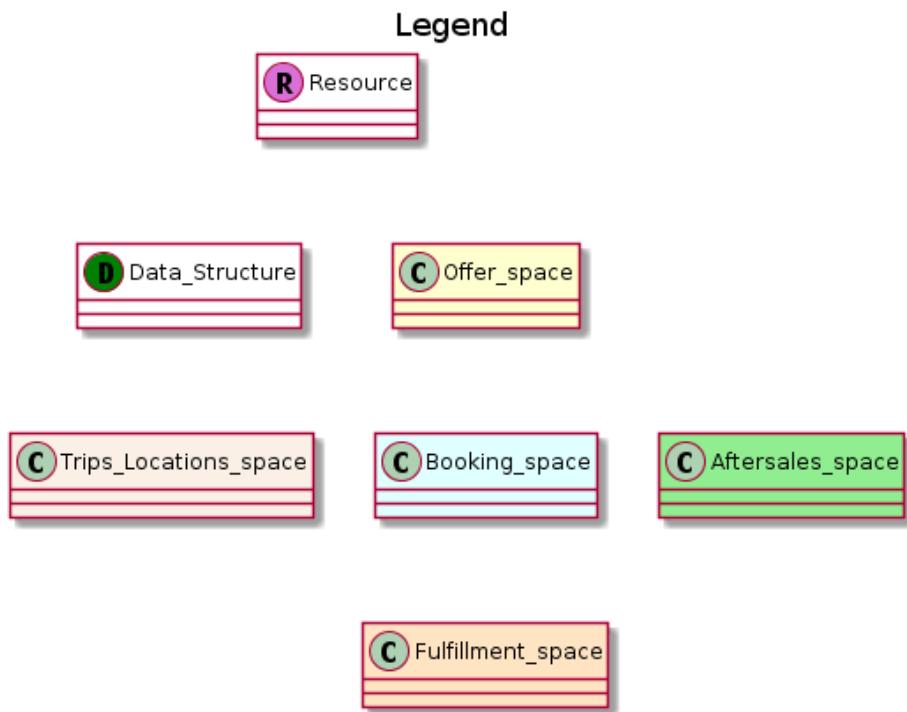
To retrieve the stored personal data the booking has to be requested.

## Data Structures - Online Models

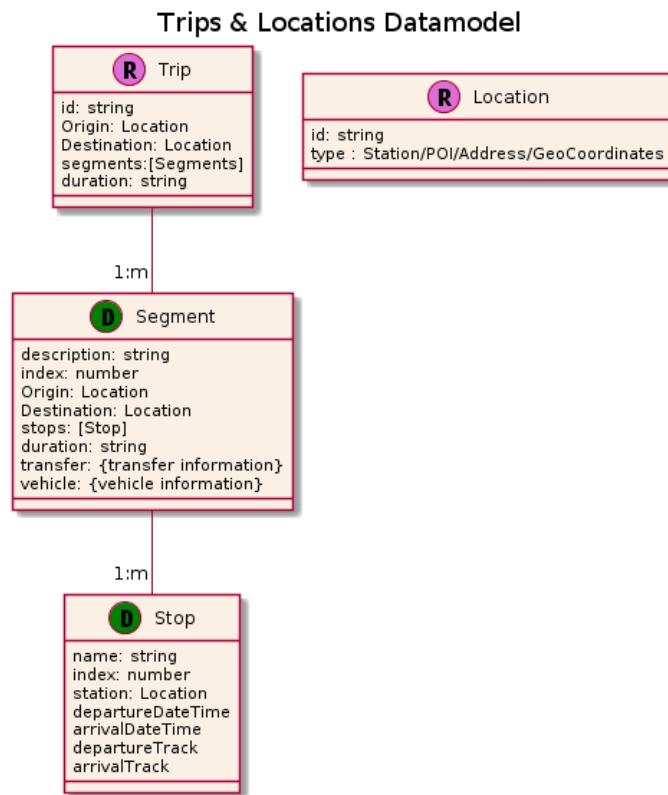
This chapters present the data models underlying the API specifications. It is therefore not a strict representation of the resources themselves (those are self-represented in the OpenAPI specifications.)

As such, some of the details of how the information is structured in the API are not represented or simplified in the data models. The main purpose of this data model is therefore to help a quicker understanding of the API and its underlying concepts.

### Legend:



## Trips and Locations Model

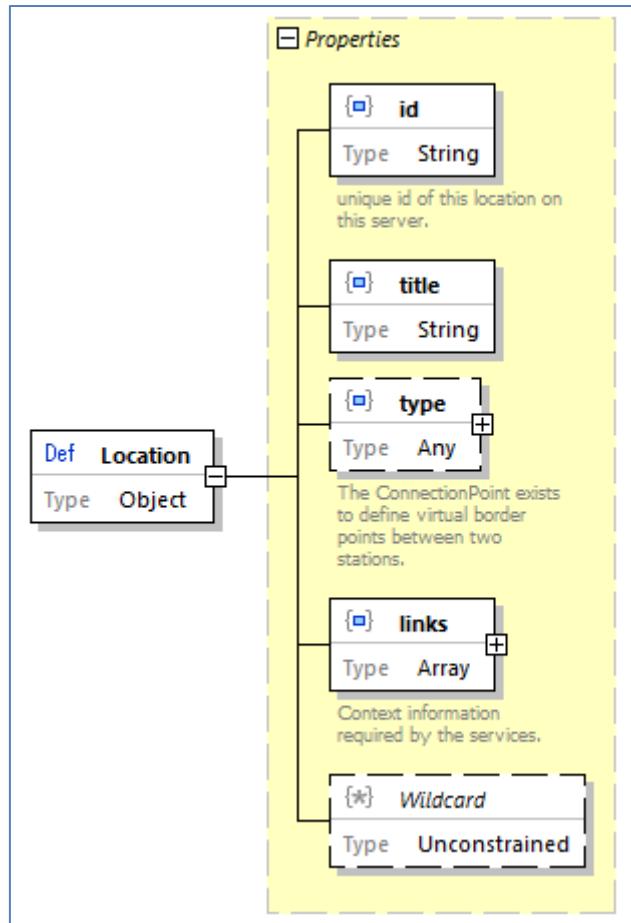


## Locations

Locations are resources representing a specific location in a trip: departure, origin, intermediate stop or other. A location can be of different types:

- **Station:** represent a train station. It is obviously the most relevant type for OSDM. Stations can be indicated as codes from different code sets. As with other code list based representations in the API, using the UIC code set is highly recommended.
- **Address:** any street address can be represented here. Is included to foresee extension towards multi modality and first/last miles solutions
- **POI:** used to represent a specific point of interest
- **GeoCoordinate:** allows providing any location on the globe using its geographical coordinates.
- **ConnectionPoint:** allows to model virtual border points by defining stations within the connection point lies.

Locations are modelled in the API as resources with a long time-to-live, which should allow efficient caching of this data, therefore removing the need of getting full location details in transactional operations.



## *Trips*

Trips represent the concrete realization of a trip going from departure station to destination station. A trip is composed of one or more segments.

Each segment (also sometimes called legs) represent a connection between two locations where the passenger will either step in a transport or step out of a transport (most likely a train). Three types can be distinguished:

- **origin:** departure location of the segment
- **destination:** arrival location of the segment
- **stops:** intermediate locations encountered between origin and destination of the segment

Regardless of whether the products to travel these segments are train-bound, or based on with a validity period of any duration, segments (and by extension trips) are always train-bound and represent the realization of the travel wish using specific trains at a specific moment in time.

Trips can be retrieved with or without details of all intermediate stops on the way between departure and arrival of each segment.

The trip describes the travel solution the customer has chosen from a timetable system. The trip is an ordered list of segments where each segment contains a vehicle for the transport, a list of stops which must include departure and arrival of the vehicle and optionally a transfer to the next vehicle in the list. The transfer will not be part of the offer.

Note: The allocator is responsible to create the trip data and to obey the minimal connection times provided by the carriers and station managers as part of the timetable data. Providing an offer does not imply that the connection time in the transfer has been validated by the carrier. However the carrier is allowed to reject an offer due to a violation of the minimal connection time.

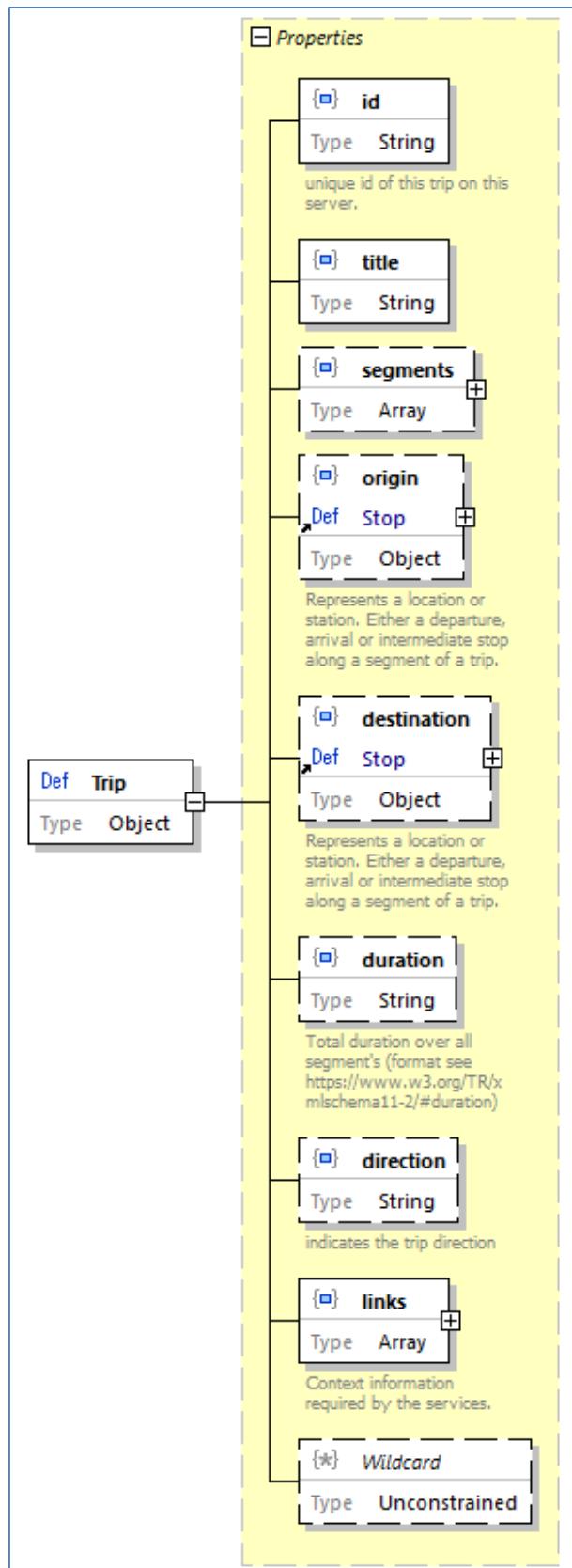


Figure 55 Trip data structure

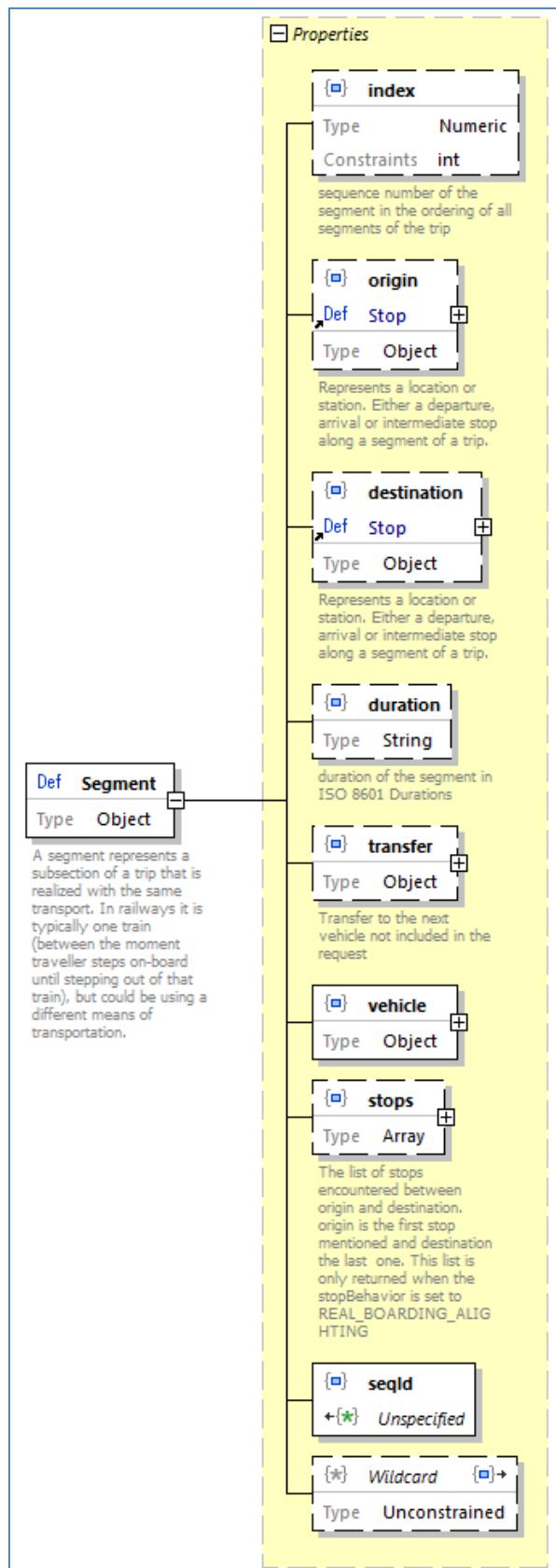


Figure 56 Segment data structure

The vehicle is identified via an array of numbers or a line number (in case of local transport there might not be dedicated number for the vehicle). Multiple numbers can be included in case of coupled trains running together but with different train numbers.

**Note:** The number is the train number not combined with the service brand abbreviation (e.g. TGV 4711 has number 4711).

The service brand code contains valid codes from the service brand code list (UIC/ TAP-TSI) only.

The valid codes of the transport mode must comply with the service brand code list (UIC MERITS/TAP-TSI).

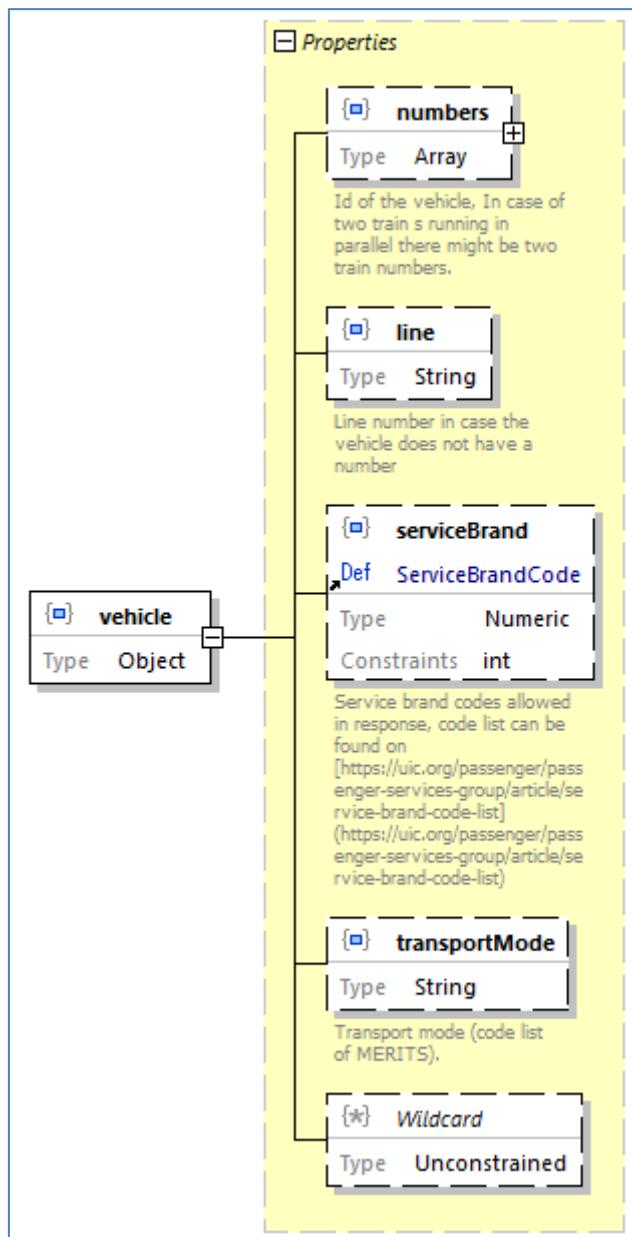


Figure 57 trip data structure - vehicle

The stops include the station and the arrival and departure time. Intermediate stops can be added as well.

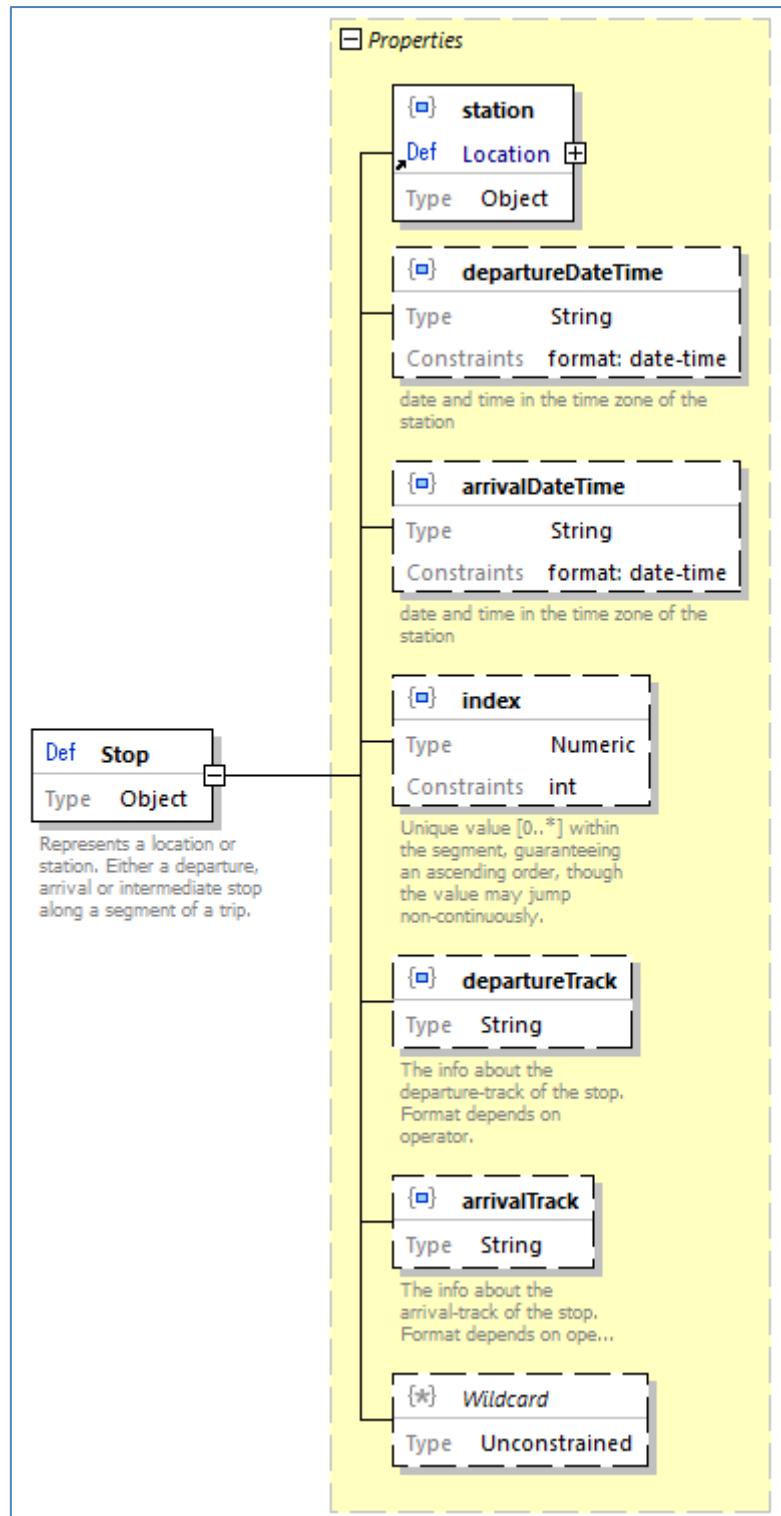
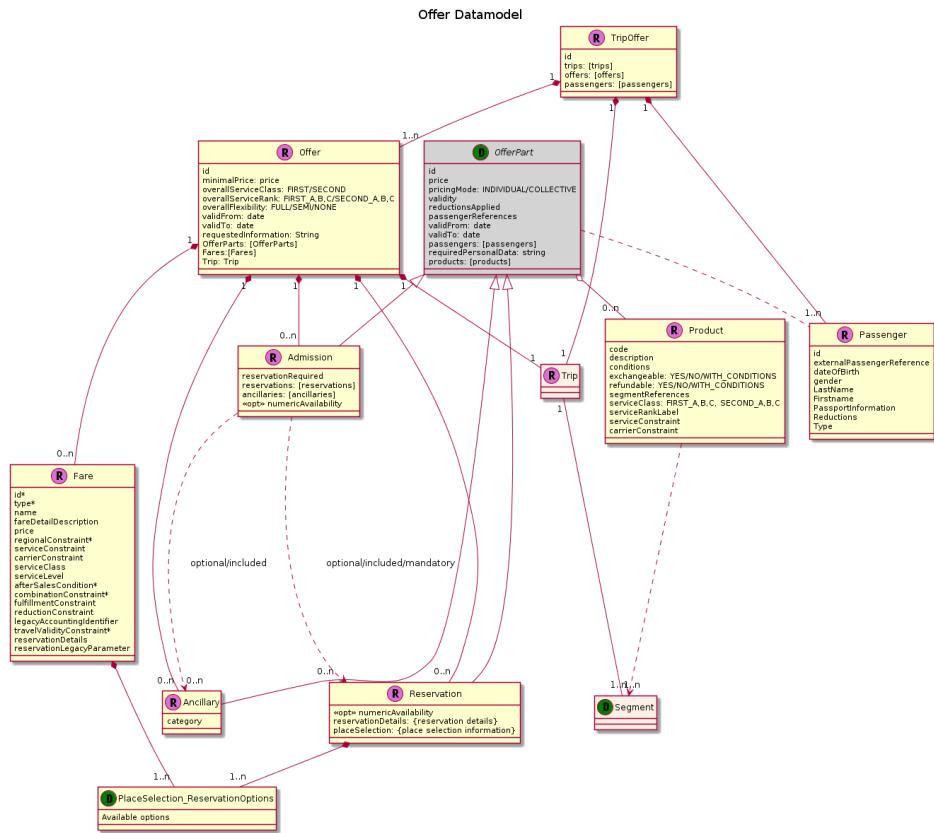


Figure 58 Trip description - stops

## Offers Model



## Offers

In Distributor Mode, offers represent a collection of "offer parts", representing bookable elements that covers exactly one a specific trip (or the requested section of a trip in case of allocator request to a Fare Provider). Note that the offer contains a reference to the trip resource it is built for, although this reference can be redundant when the `tripOffer` the offer is part of is already known to the API consumer.

In Allocator Mode, an offer will not contain offer parts but it will contain a fare element, that the allocator can use to build the final product to be distributed to passengers. There can also be hybrid situations where offers will contain both offer parts and fares if the provider offers product in the two flavors.

In some cases, API consumers will be required to provide some additional information in order to proceed with the provisional booking. In this case, the information needed will be specified in the `requestedInformation` element using a notation akin to regular expressions.

Offers should always contain a minimal price (= the price of the offer without any of the optional offer parts), a global service class, service level and flexibility. Although the calculation rules for these global values are up to the provider, the following rules are recommended:

- Service Class: the lowest class of a significant offer part should be the service class of the offer (1st class + 2nd class = 2nd class)
- Service Level: the highest service level should be the service level of the offer (bed + berth in compartment of 2 + berth in compartment of 4 = berth in compartment of 2)
- Flexibility: the lowest flexibility of a significant offer part should be the service class of the offer (full flex + mid flex = mid flex)

Offers will usually have a validity period, that is the period over which, from the fare and tariff conditions, the offer is likely to be proposed. It is **not** a guarantee that the offer remains available for that period.

Offer resources and all related resources (tripoffers and all subresources) should have a limited lifetime (recommended value 30 minutes) and be discarded when expired or at booking time.

The price of an offer is not guaranteed. The price might change when the offer is confirmed.

The offer lists all fares and offer parts for the requested passengers and links them to the passengers.

Fares are delivered to allocators only, whereas ticket vendors will receive offerParts only. OfferParts are Admissions, Reservations or Anxilliaries.

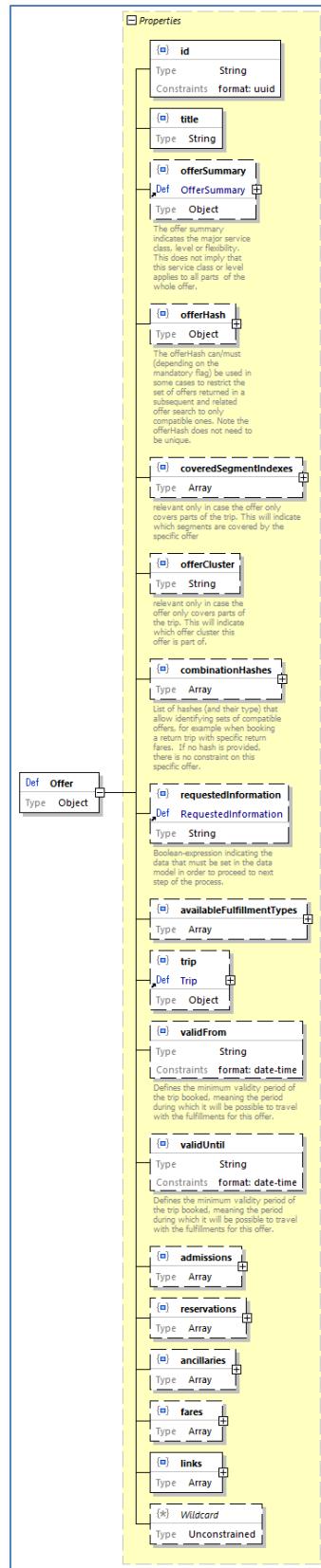


Figure 59 Offer data structure

### TripOffers

TripOffers are the resources grouping all the offers relating to one same trip. Indeed, in most cases the provider will propose several offers of different comfort and flexibility levels.

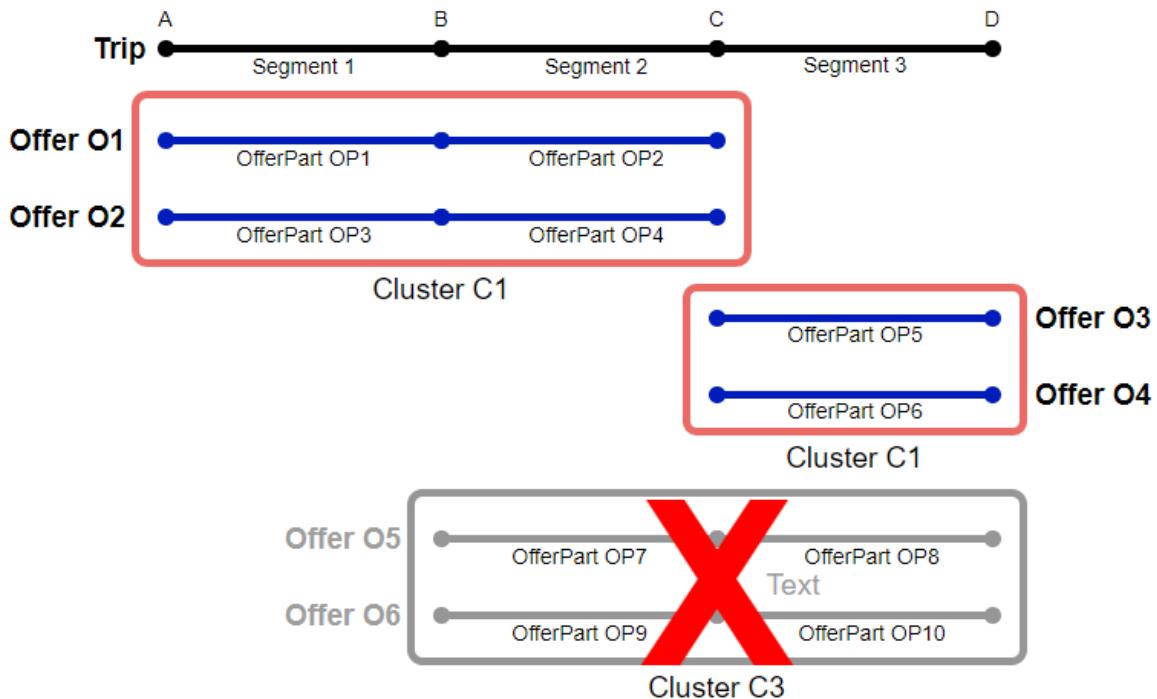
This resources, the trip resource representing the trip the offers are for and the passengers for the trip.

#### *Offers with Partial Coverage*

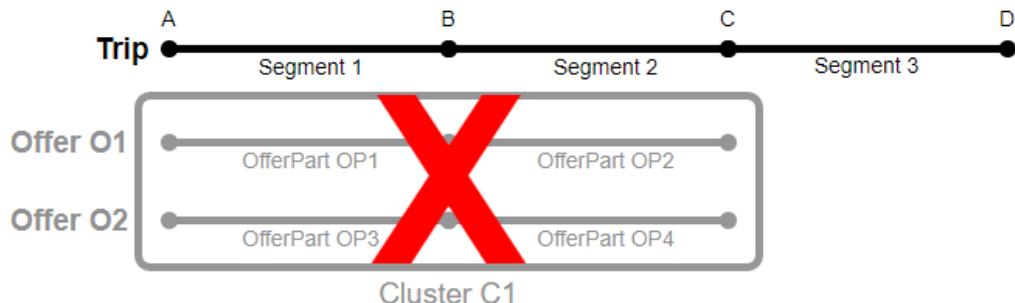
It is possible in OSDM to propose offers covering only a subset of the requested trip under specific conditions:

- the segments covered by a given offer are indicated through the `coveredSegmentIndexes` property
- all offers covering the same set of segments belong to the same `offerCluster`. As such, their `offerCluster` identifier must be filled-in and identical
- a `segment` can only be covered in one `offerCluster` within a `tripOffer` (no overlap)
- each `segment` of the `trip` must be covered by at least one `offer` in each `TripOffer` (no gap)

*Example with no overlap*



*Example with no gap*



## *Offer Parts*

Although offer parts are by themselves not a resource, they deserve a separate section in order to for the reader to clearly understand the data model design. The idea is that offer parts represent an instantiation of a product that can be sold.

Example:

- A carrier proposes an "Early bird Holiday Fare" **product** for all its high-speed trains riding towards the seaside of the country, offered when sales date is at least 15 days prior to travel.
- Whenever an offer request is received and this fare can apply, an **offer part** is created and proposed, specifically to the date, origin and destination of that specific request, and referencing this product. As such it is therefore the offer part that for example will carry the actual price

These offer parts can be of different type, depending on what they represent:

- Admissions
- Reservations
- Integrated reservations
- Ancillaries

However, these different type share a significant amount of characteristics: they all apply to a defined set of passengers, have a price (calculated individually or collectively), and a few additional attributes. They also fill in the same fundamental role in the offer, which is why they are represented here as an abstract parent class.

## **Multiplicity**

Offer parts are not normalized but will reflect the reality of the products generated. This means that one offer part will almost always equal one fulfillment in the resulting booking, should it be completely realized and confirmed.

So, two passengers travelling together happen to get exactly the same product (because their profile is identical in terms of age, reductions etc), will still get 2 individual offer parts (one per passenger) if the product covered has individual pricing and fulfillment, while they would be grouped in one offer part in case of collective pricing and fulfillment. (see examples at the end of the offer section).

The offer part is an immutable part of an offer provided for a distributor. It does not include any detailed information needed to combine fares or to support the creation of bar codes or control data or to create complex textual descriptions from regional validity data.

Where details must be displayed to the customer these are included as textual descriptions.

OfferParts are not included as such within the messages but as admissions, reservations or ancillaries. They share common attributes defined in CommonOfferPartAttributes.

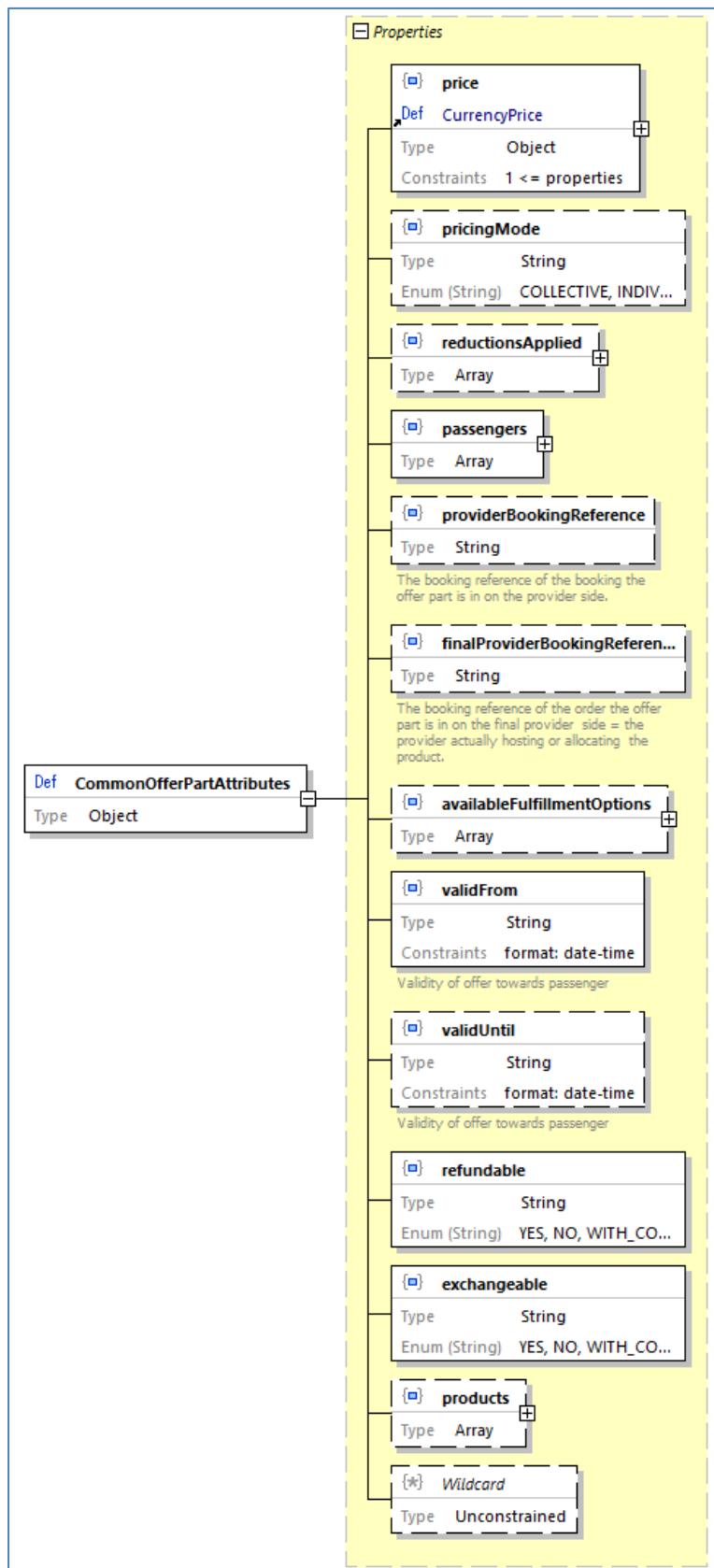


Figure 60 CommonOfferPartAttributes data structure

## *Offer Parts - Admissions*

Admission offer parts represent a travel right, or the entitlement to travel onboard a train between the given origin and destination, following the given route, without a seat reservation. In most cases, these train products will not be train-bound either.

In some transportations, seat reservations or an ancillary products (such as a WIFI connection or a meal onboard) can be associated with the admission for one or more of the segments. A link will in this case point from the admission to the reservations or ancillaries, and the link will be qualified. Ancillaries can be either included or optional, while reservation can also be mandatory to travel. Finally, there can be cases where all reservations associated are optional while it is mandatory to pick at least one (it can be the case for night trains for example). In this case the reservations will all be qualified as optional, but the `reservationRequired` flag of the admission will be set to true.

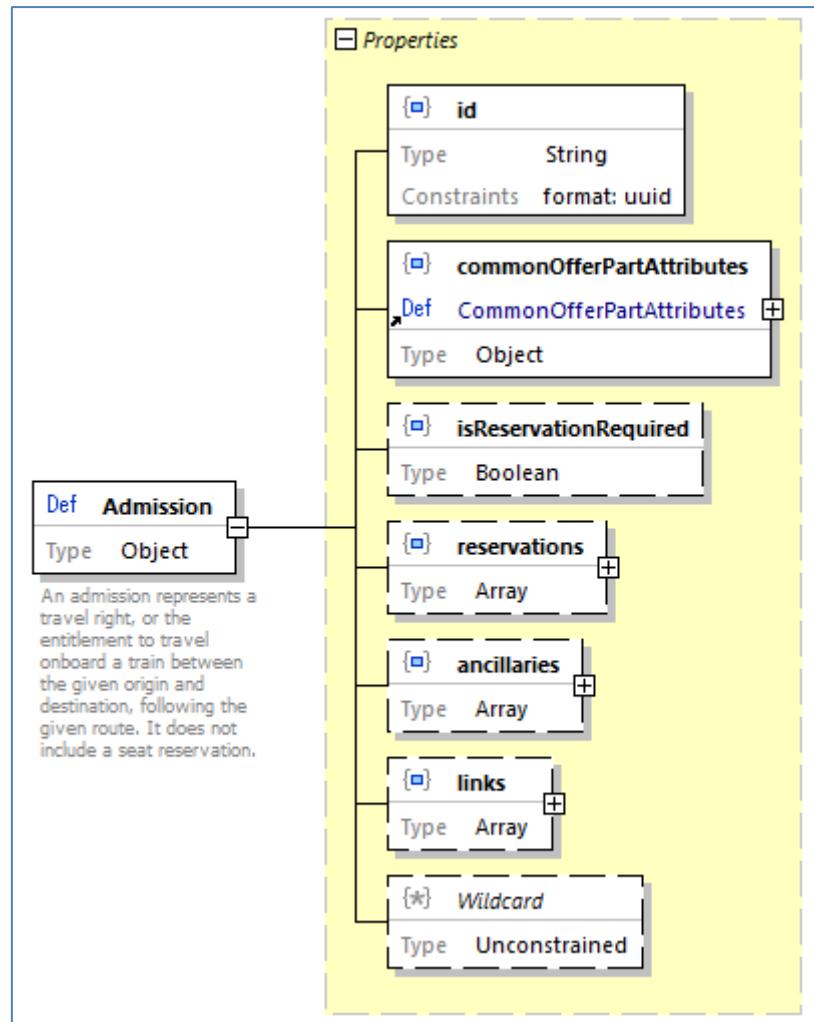


Figure 61 Admission data structure

### *Offer Parts - Reservations*

Reservation offer parts represent a seat (or other accommodation) reservation on the transportation. In contrast with admissions, a reservation is bound to a specific train, although it normally does not include the entitlement to board the train. Passengers will therefore typically need an associated admission offer part or other entitlement (such as a pass) in order to actually travel.

Note booking an offer will not book the reservations in the offer unless they have an "included" relationship with an admission of that offer. In order to add a non-included reservation to a booking, the reservation ids will have to be passed.

Reservations have several additional attributes due to their specificities compared to admission products:

- Reservation Details provide additional information on the service Level and, once the offer will have been booked, the exact reserved places, with their properties and links to the concerned passengers
- Place Selection Details: contains several elements related to the selection of places:
  - ReservationOptions show, at offer retrieval stage which options are available for this reservation.
  - SelectedOptions allows the API consumer to specify which options are desired.
  - SelectablePlaces and SelectedPlaces are only relevant to graphical selection of seats (seatmap), not in scope of this initial version.

### Modelling lump sum reservations

For some trains, especially in Germany and Austria today, a specific form of reservation booking can be found where the price for adding an optional or mandatory (but not free) reservation remains the same regardless of the number of reservations actually booked. In order to represent this type of reservation with the current model, 2 approaches are proposed to implementers:

- Generate two distinct offers: one with all (available) reservations proposed as included, the lump sum for the reservations being integrated in the admission price. In this approach it is assumed that a passenger will always book all available reservation, since the price is the same anyway. This approach also allows to not propose a reservation if there is none available on one of the segments, while still offering the offer for the complete trip with reservations on all segments where it is available.
- Propose all reservations as optional reservations with an identical unit price equals to 0 or to the reservation lump sum, associated with specific information in the product conditions or the offer messages. At booking time, a price update (increase or decrease) is then applied so that the lump sum is counted once and only once, associated with a booking message warning that the price update took place.

### *Offer Parts - Ancillaries*

Ancillaries are used to represent non-transport products associated with the transportation request submitted. It could be onboard services such as a Wi-Fi connection or a meal, or services associated with one of the stops, or origin/destination, like a parking spot or lounge access.

This offer part is significantly simpler than those instantiating transport products, and only has one additional attribute, being the category of the ancillary.

### ***Products***

Products are the products actually offered by the provider, either directly or either as distributor if the provider itself retrieves the products (or constitutive fares) from another provider. Products resources contain all the conditions and attributes of the product, regardless of the actual sale case. Typically, this matches commercial products having the same name and recognizable common sales & aftersales characteristics.

Although no manipulation is performed on products, it is nevertheless proposed as a resource mainly to allow caching of the information. Indeed, since product information usually does not change too frequently (and usually at well defined dates), those resources can then be exposed with a significantly longer time-to-live and save significant bandwidth. It also allows for a "product catalog" functionality to be built by the API consumer, should he or she want to do that.

## Fares

Fares should be seen as the counter part of OfferParts in case of interactions between an allocator and a fare provider (allocator context). The key difference here is that where offer parts are products defined by the provider and fulfilled by it as well, the fares do not constitute a distributable product. It is up to the allocator to build the distributable product (that he could then distribute as an offer part via an API), based on one single fare or by combining fares coming from different providers. In consequence, the fulfillment of the resulting product is the responsibility of the allocator as well.

For distribution systems also able and allowed to act as allocators, encapsulating both fares and offer product in offers allows to have a similar flow of interactions regardless of the type of provider.

An elementary fare used in online services.

The data elements which are referenced in bulk data fare are here included as objects.

An elementary fare to create an offer linking all constraints to one price.

Data elements	
fareType	Admission, IRT, Anxillaries , Reservations
name	Name of the fare
fareDetailDescription	Additional explanation on the fare (e.g. on included fees like Diabolo or Venice fee)
price	Price with currency EUR must be provided if not otherwise agreed bilaterally.
regionalConstraint	Definition of the regional validity of the fare and the geographical combination rules (connection points)
serviceConstraint	Restrictions of the service allowed to be used
carrierConstraint	Restriction on the carriers that can be used with the fare.
serviceClass	Class the passenger can use
accommodationType	Category of places the passenger can use.
accommodationSubType	Detailed category of places the passenger can use.
passengerConstraint	Rules and restrictions on the passenger types allowed to use the fare and rules on combining passengers.
afterSalesRules	After sales rules for the fare. In case the allocator is responsible for the aftersales rules this is almost empty.
combinationConstraint	Rules on the model of combination of this fare with fares of other carriers.
fulfilmentConstraint	Restrictions and requirements on the fulfilment and security to be applied by the allocator.
reductionConstraint	Applied reduction cards only

reservationParameter	Information on parameters for reservation via the 90918-1 interface and reservation options.
personalDataConstraint	Rules on the personal data to be provided in a booking
legacyAccountingIdentifier	Data to be included in the current IRS 30301 accounting data format.
travelValidityConstraint	Rules on the validity for travel of this fare.

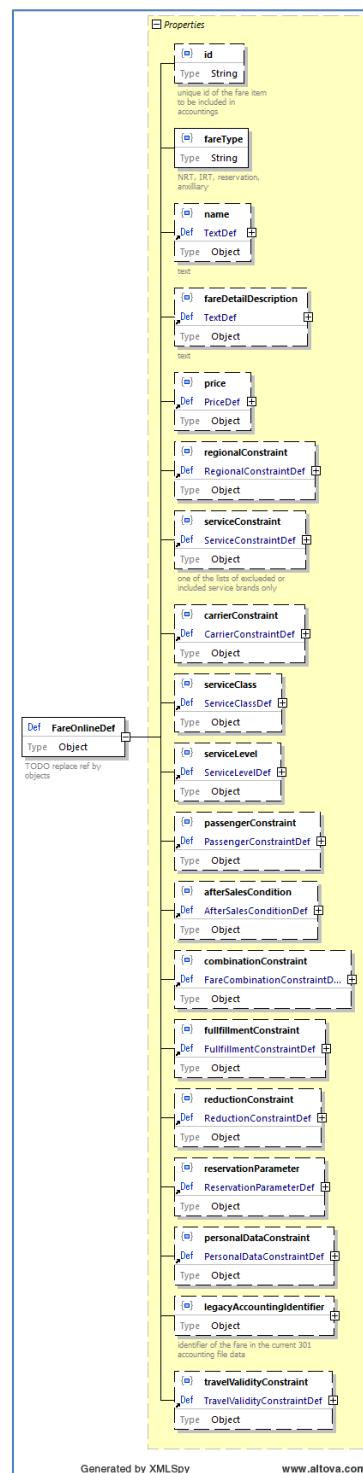


Figure 62 FareOnline data structure

Data constraints	
price	A price must be provided for all offline fares including those where the price is zero.
legacyAccountingIdentifier	In case 30301 in the current version is used for accounting
serviceClass	Must be provided
combinationConstraint	Must be provided
travelValidityConstraint	Must be provided

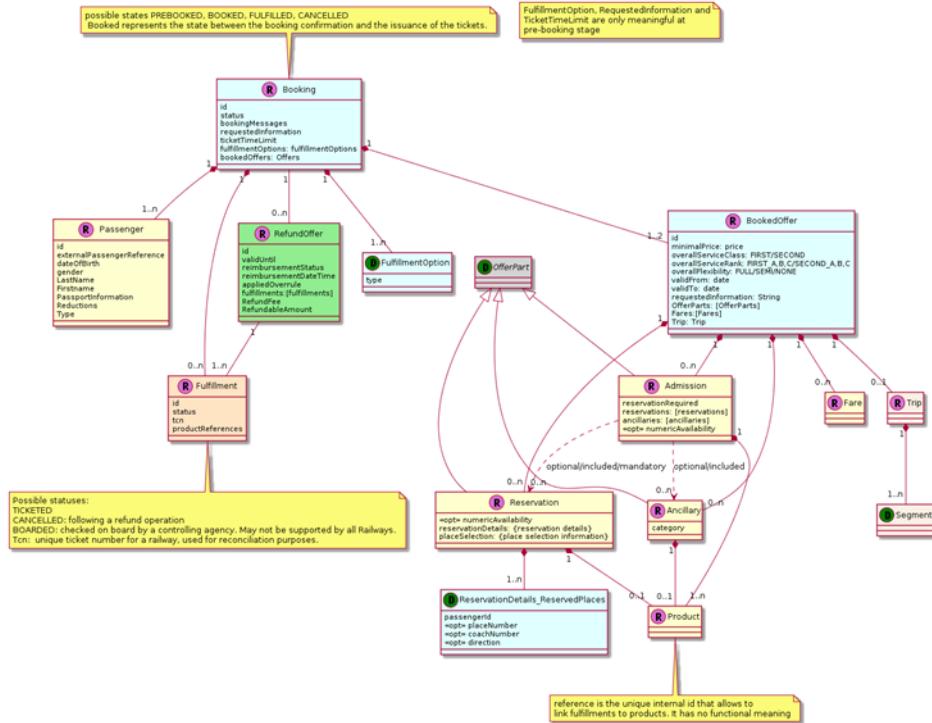
### *Passengers*

As the name suggests, passenger resources represent the passengers for whom the offers are proposed. All offers generated are always proposed for the complete set of passengers (no partial offers covering only a part of the passengers is generated). However, it is possible that because of age, reductions or other, some passengers are permitted to travel some segments without needing a travel right or reservation. It is for example usually the case for infants travelling on their parents lap.

While a lot of attributes can be set for passengers, only a few are required at this stage (and even later). The key elements at offer stage are already specified in the offer request. The link between the possibly anonymous passenger profiles (in most basic form: a unique (in the booking) reference, age and reduction entitlement if any) created in the offer request and the passenger resources in the TripOffers can be made based on the passenger Reference attribute.

The passenger resources created in the context of offers have their lifetime strictly limited to the lifetime of the offer resource they are part of. The resource and all local traces of it should be discarded once the offer has been booked or reached the end of the lifetime, in order to avoid any privacy concern.

## Booking Model



### Booking

The booking represents the local (to the provider) booking for the offers that have been selected. It contains a set of sub resources, most of which were encountered in the offer stage, but also adds a few specific attributes and information, the most important undoubtedly being the booking status (see for the state model below). The booking will indeed evolve over time based on API consumer actions, time elapsed or other business events.

The booking also contains additional attributes that are needed to manage and control the confirmation of the booking when it is in provisional state, such as the ticket time limit or the fulfillment options. The ticket time limit is the time during which the booking is guaranteed to remain available for confirmation for the price and possible reservations assigned at provisional booking time. Basically, it is the time given to the API consumer to perform all updates needed to confirm the booking and trigger that confirmation.

**FulfillmentOptions** allows the API consumer to specify the format desired for the fulfillment.

### *BookedOffers*

BookedOffers are actually the same resources as the offers except that they are now booked. Most of the resource remains unchanged, except for the sections on reservation details (either in reservation or integrated reservation offerparts , or in fares), where but the sections related to the reserved places (in `reservationDetails`) will now be populated with the references to the space allocated by the provider system where the transport product is hosted.

### *Fulfillments*

Fulfillments could once have been called tickets. But the evolutions in the industry have led this to be a limitative naming, as various kinds of ticketless onboard controls are rapidly taking over and become the norm rather than the exception. Since only the distribution part of the process is in scope, the details of how to produce or control fulfillment are not covered. From a distribution standpoint, the only needs are

- The possibility to point at a fulfillment representing an offer part (= the id) for aftersales operations.
- The capability to link this fulfillment to that associated offer part they relate to.
- A business identifier that can be used in associated processes. For railways, that would be the Ticket Control Number, or TCN.
- Links to the documents or other security features that can be used to represent and control fulfillment status. In most case it is a PDF document and/or a barcode.

These are all provided in the fulfillment sub-resource.

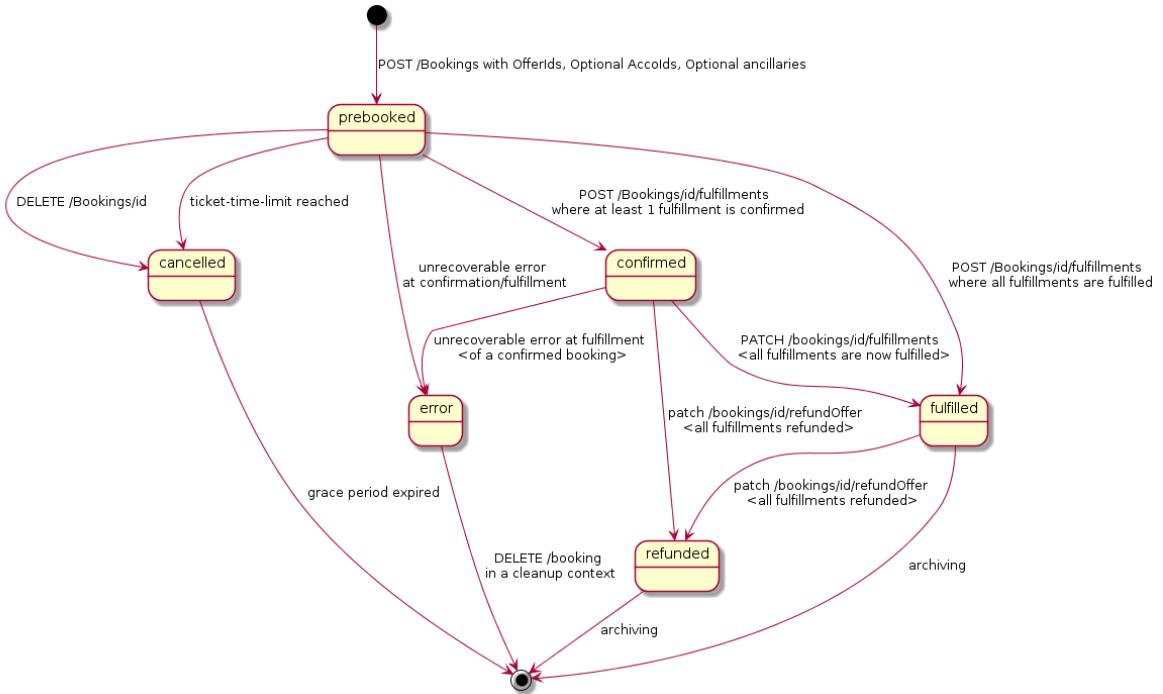
### *Passengers*

The passengers sub-resource in the booking is actually the same as the one in the `tripOffers`, but it is worth mentioning it separately here as

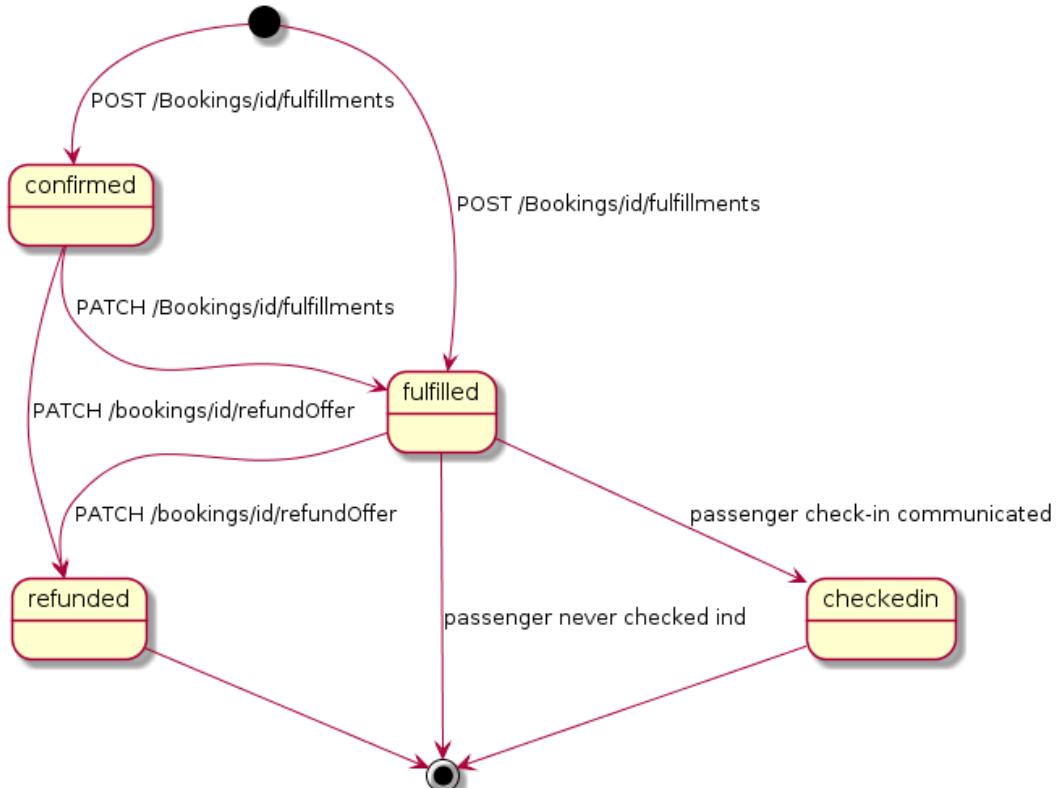
- being a sub-resource, it will have a different path
- as mentioned in the section about offers, the passengers in the `tripOffers` will disappear with the booking or the time-to-live expiry of the offers, and the passengers created in the booking will have a different id. In case the caller wants to keep a passenger identification within the booking he needs to provide his reference within the passenger data.

## State Models

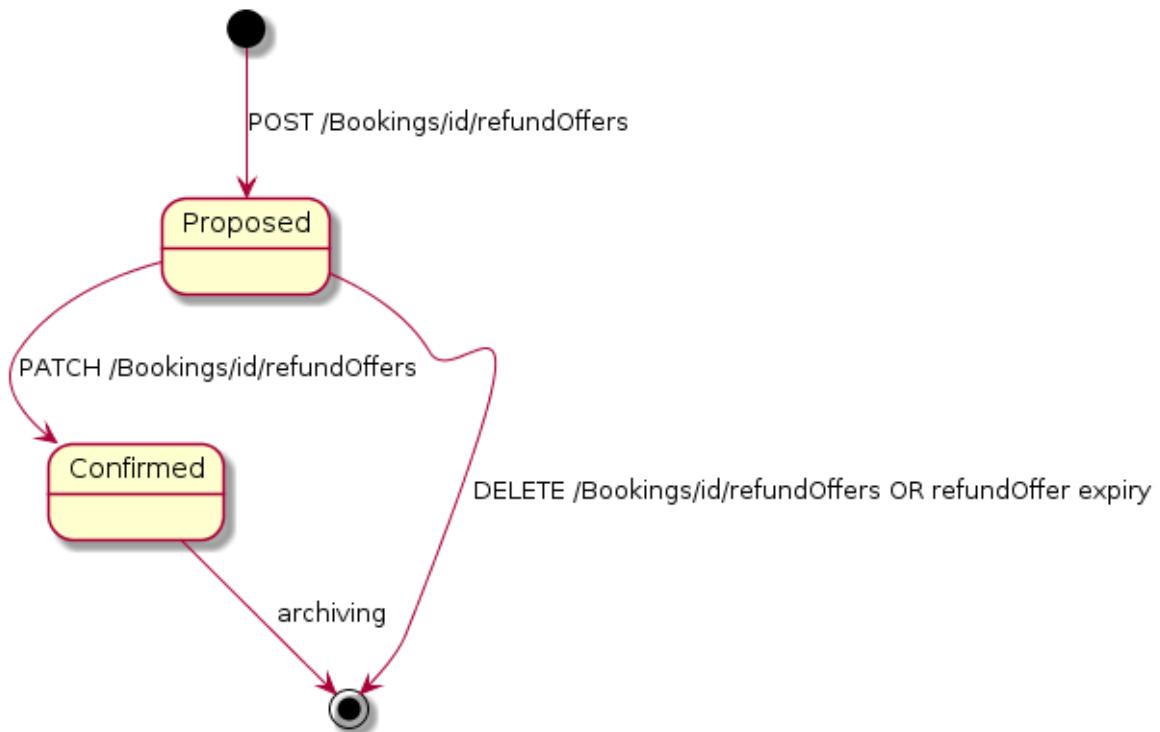
### Booking State Model



### Fulfillment State Model



### *Refund State Model*



## Service Specification

### Service Overview

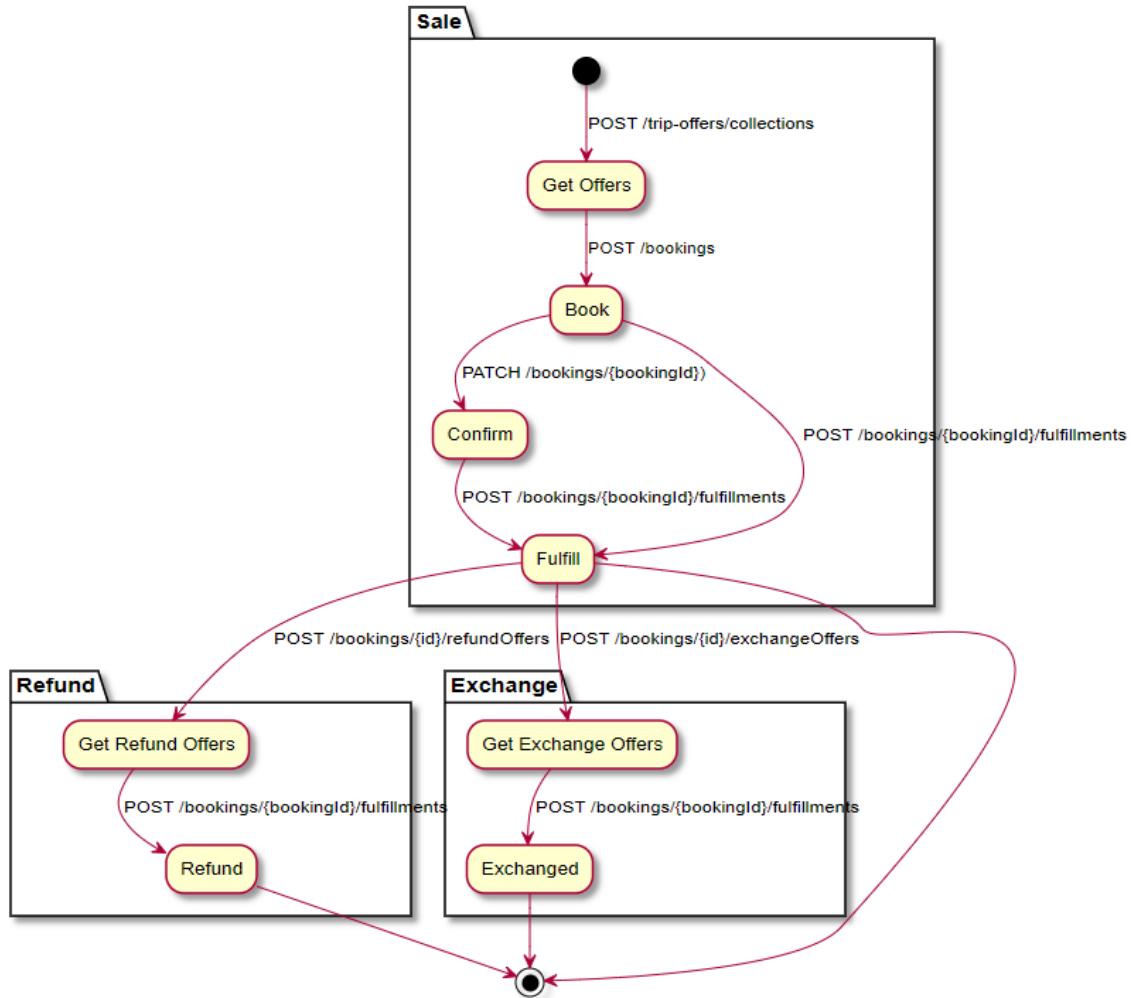
To fulfil the requirements expressed by the business expert and provide the capabilities a set of services has been developed.

*Table 1: OSDM Online API*

Resources	Description
/locations /trips	Resources to search for trip and locations
/trip-offers-collection /trip-offers /offers	Resources to get bookable offers
/offers/{id}/admissions /offers/{id}/reservations /offers/{id}/ancillaries /offers/{id}/fares	Resources to manipulate parts of an offer consisting of, e.g., admissions, reservations or ancillaries; if permitted, also fares are offered.
/bookings	Resources to manipulate bookings
/offers/{id}/passengers /bookings/{id}/passengers	Resources to manipulate a passenger's information at every stage of the flow
/products	Resources to retrieve products information on one or more products
/bookings/{id}/fulfillements /fulfillements	Resources to retrieve fulfilments, e.g. tickets
/bookings/{id}/refundOffers	Resources to get and accept a refund offer
/bookings/{id}/exchangeOffers	Resources to get and accept a exchange offer
/coachLayouts	Resources to get layouts of coaches

In the rest of this chapter a detailed walk through thru the usage of the API is provided.

## Process Flow

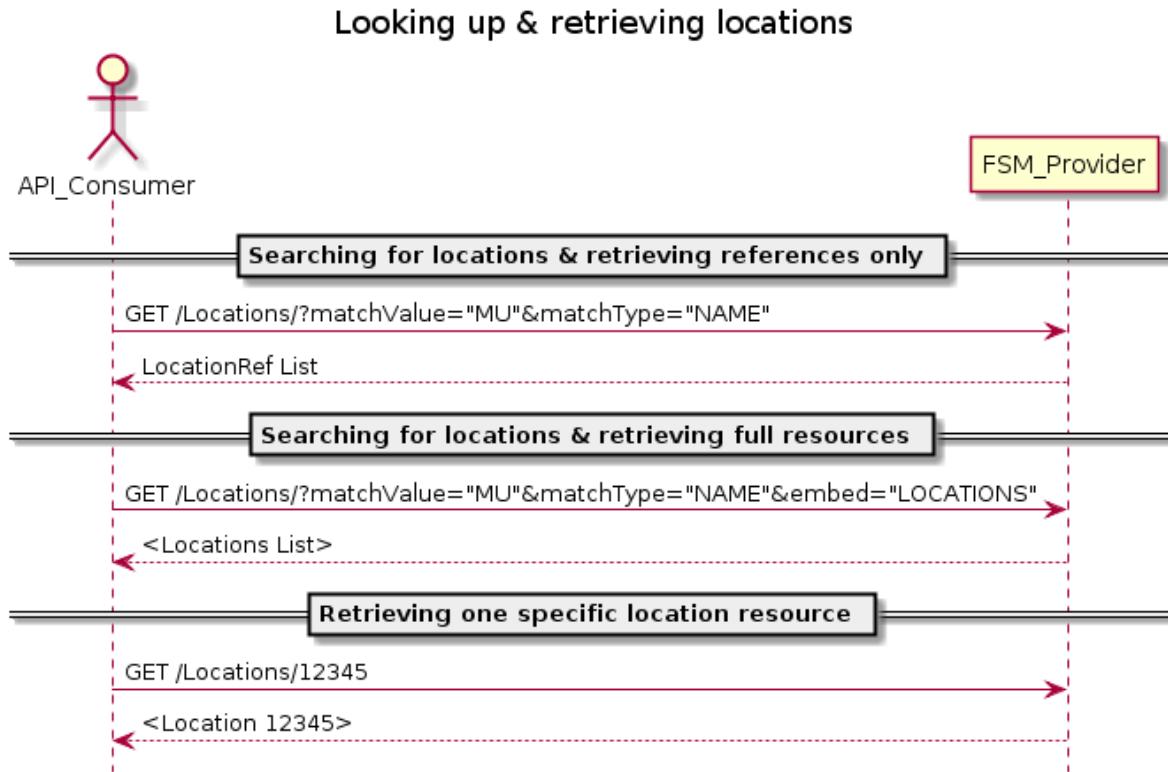


The process flow starts with getting offers which can be chosen by the customer. Once selected they can be prebooked and after the payment process (which is outside of the scope of this document) they can be booked. The fulfillment of the booking can either be on paper or paperless.

If needed bookings can either be refunded or exchanged by providing the customer with a refund or exchange offer which can then be booked by the customer.

## Trips and Locations Processes

### *Looking up locations*



`GET /locations` lookup can be used by an API provider in order to search for locations. Two typical uses cases would be

- getting a set of locations (in full or as reference) from a substring of the name
- getting full details on a location based on one of its codes

Note that the functionality is not intended to trigger a "dump" of the complete locations list or to build a full "browsing" functionality, hence the lack of pagination features here.

Given the high stability of this information, locations are given a long time to live and get responses can be cached for a long period, so these operations should not be too costly in terms of calls or bandwidth.

### Error handling

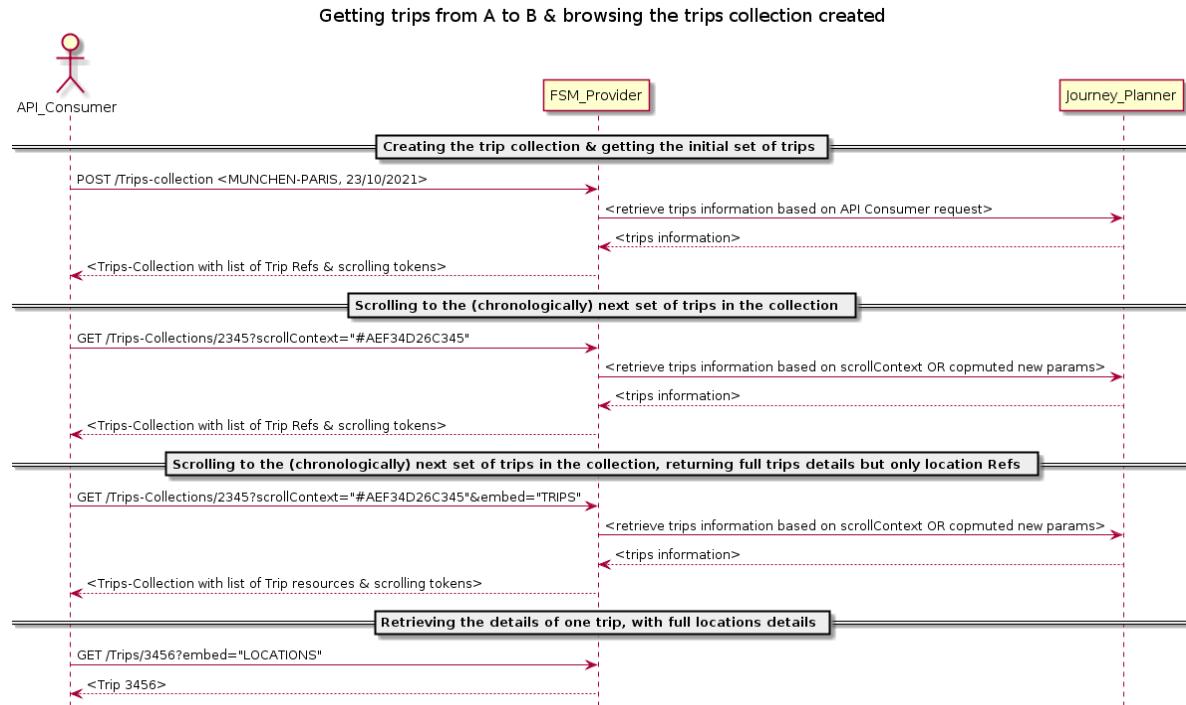
Error handling by the provider remains basic here as a handful of cases have to be handled:

- invalid characters in the search string.
- no result found for the given criteria.
- The search did not return any result
- unknown error on server side

In all cases, the error handling starts and stops with the ODSM provider returning the appropriate JSON Problem element.

In case the error can apply to multiple fields, it is recommended to provide additional details such as the incriminated field in the detail property of the Problem element.

### Getting and Browsing Trips



If the API consumer only needs a schedule, and no bookable offer, it has the possibility to create a trips collection using `POST /trip-collection`. If the query is successful, the initial response to this resource will be a set of trips matching the provided search criteria. Please refer to the YAML specifications for the list of search criteria available.

Depending on their respective journey planner capabilities, it could be that some criteria cannot be supported by one or the other provider. In this case it is up to the implementing party to clearly document those limitations together with the publishing of its endpoints. In all cases, at least origin, destination and `travelDateTime` must be supported.

Based on an initially returned trips collection, it is then possible to retrieve earlier or later trips using `GET` on the trip-collection by specifying the appropriate scrolling-tokens. As with all cases where nested resources can be returned, individually or in list, the embed feature allows specifying whether complete trips should be returned or only a title and a link. A `GET` verb without any scrolling-token will simply return the last set of trips return.

It is important that once a trip has been generated, the trip's time to live has a sufficient duration to allow the possible subsequent uses:

- A trip or a trip id could be used in subsequent offer operations (see further)
- When scrolling back and forth over time, a same trip should maintain the same id, so the API consumer can, if desired, expand the set of trips in its own context and have the guarantee that one same trip (in terms of content) will remain with the same id (in terms of resource id).

## Error handling

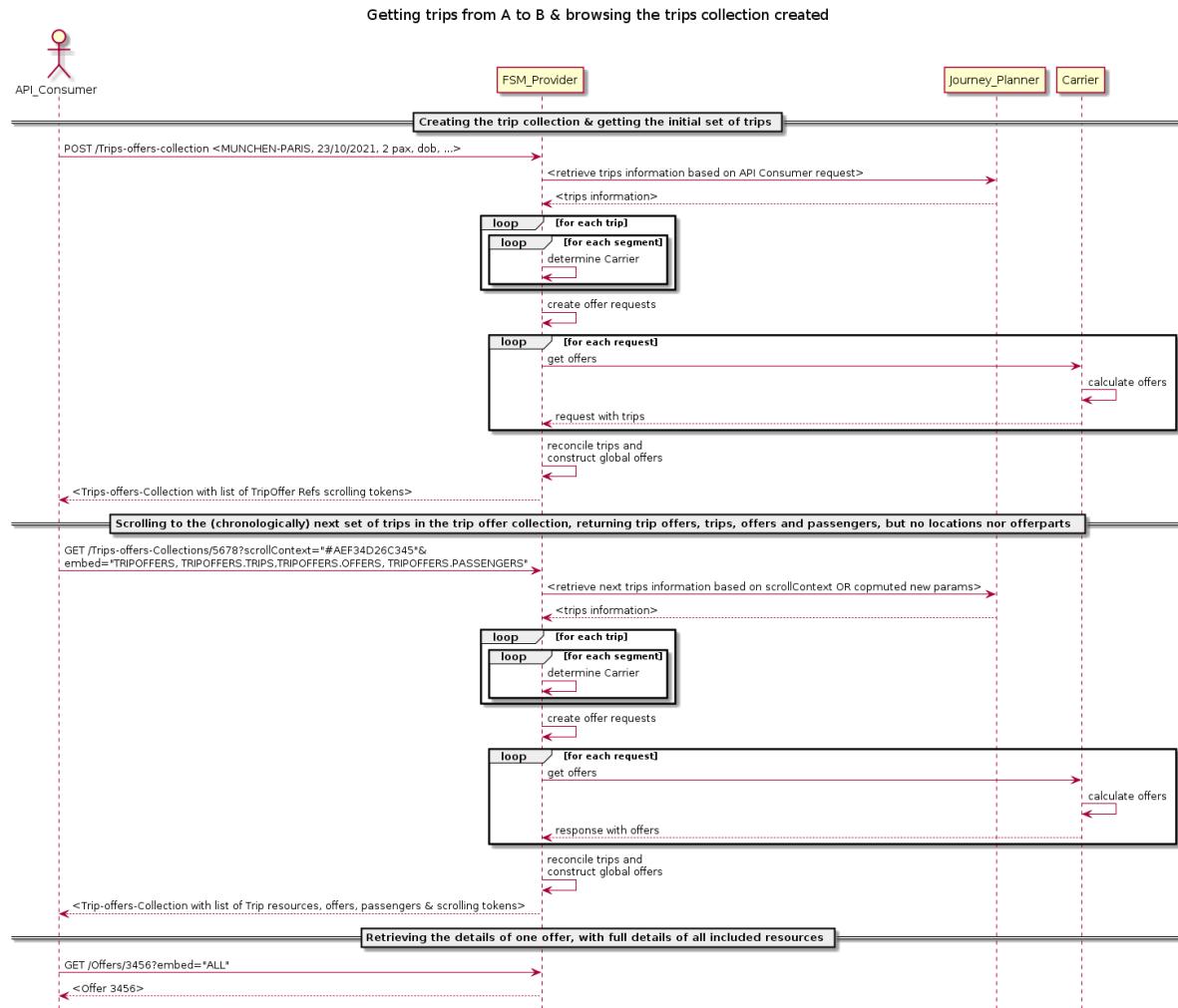
Since requesting trips still does not involve any transactional operation, the error handling is also limited to returning a JSON problem element. The following cases are to be considered:

- A search criteria value contains invalid value or invalid characters
- A search criteria that lies outside accepted boundaries: it could be the date in the past, or too far in the future, or value outside bounds for the max number of changes
- The origin or destination is not known
- The search did not return any result
- Unknown error on server side

In case the error can apply to multiple fields, it is recommended to provide additional details such as the incriminated field in the detail property of the Problem element.

## Offers Processes

### Getting and Browsing Offers



Requesting and browsing offers works a lot like the requesting and browsing trips: the API consumer submits search criteria, and a collection of "trip offers" is returned. This collection can be browsed to earlier and later trips the same way as the trips collections.

The search criteria for offers extend the search criteria available for trips with additional criteria applicable to the fares and products that can be returned such as the fare flexibility, the service class or the currency the offers should be proposed in.

Although the trip-related search criteria are present and will likely be the easiest and most used option, there are two alternative ways to search offer if a specific trip is already known:

1. Provide the id of a trip previously searched on the provider (obviously, that trip resource should still be alive)
2. Provide a complete trip structure. It is only way to go for a request to a fare provider working according to OSDM rules. In this case, the trip provided may be larger than the part for which fares are requested. For this reason, the requested section must then be provided so that the provider knows which portion to work on.

An offer request to a provider can lead to offers with multiple `OfferParts`, potentially coming from different sub-providers (OSDM compliant or not). However, in preparing offers with multiple offer parts for the API consumer, the provider must follow the following rules:

- The POST `/tripoffers-collection` only generates complete offers covering the complete trip (or complete section) requested.
- While the combination logic is left to the implementor, it is recommended to only build and retain offers that are homogeneous (as much as possible) in terms of flexibility and comfort class.
- Each offer request should create a new offer context with a dedicated `TripOffer` collection resource and dedicated sub resources, since it is possible, and may even be required to patch offers with data that is specific to the booking dialog at hand in order to perform the booking.

The resources used at offer steps offer various levels of embedding and multiple granularity for the retrieval of information, so each implementing party can fine-tune the queries in order to get all the information needed for the processing at hand, and only that information.

## Offer Messages

During the offer construction, the provider could encounter events that, while not halting the process or constituting an error, may be relevant for handling of the response by the API consumer. These events can then be passed on using the offer Message element. The following events are identified and relevant to this section

- Overbooking
- Schedule correction applied

## Round trip handling

We define a round trip as a mirrored couple of trips (*A-B B-A*), each made of one or more segments.

The construction of a round trip in OSDM is always a two-step process, where the outward offers are requested separately from the inward offers.

## *Receiving offers with return products and fares*

In order to indicate to the provider that the intention is to build a return trip, the `returnSearchParameters` are used:

When requesting offers for the outward travel, the API consumer has to provide a return date. The response will contain a set of offers. Each of these offers will have a `offerHash`. Usage of it is described further below.

To get offer for the inward travel, the API consumer will have to provide

- The id of the outward `tripCollectionID` (allows knowing the context in which the outward offers are made)
- Depending on the targeted fare provider, the `offerHash` for the selected outward offer, or the set of potential offers (as the `offerHash` does not have to be unique. E.g. all offers for a given date might have the same offerHash if the constraint is only on date) can or must be provided. Whether the `offerHash` is mandatory in the inward offer request is indicated by the “mandatory flag” that is provided in the outward offer response next to each `offerHash`. If the `offerHash` is provided in the inward offer request, the provider should then only return offers that are compatible with the indicated (set of) outward offers.

Note that depending on whether the offerHash is mandatory or not and whether it is unique per outward offer, it may or may not be mandatory to select the outward offer before the inward offer request can be constructed.

### ***Using combinationHashes***

Besides the offerHash discussed above, some offers may have one or more **combinationHashes** as well. As the name suggests, these can be used in order to determine how to combine offers. For round trips constructions, we are specifically interested in combination types of the type “return”.

The idea is actually fairly simple: in case no filtering is applied on the inward offers using the **offerHash** filter mentioned above, the returned inward offers may not all be compatible with all outward offers. Identifying compatible pairs are simply identified by the fact that they have the same (set of) return combinationHash(es). Offers with no return combinationHashes have no constraints.

Hereunder an example illustrating this concept:

#### **Outward Offers**

- Offer1: -
- Offer2: #123
- Offer3: #234, #123
- Offer4: -

#### **Inward Offers**

- Offer5: -
- Offer6: #123
- Offer7: #234
- Offer8: #123, #234

#### **Valid Combinations**

- Offer1 + Offer5 (no constraint on hashes)
- Offer4 + Offer5
- Offer2 + Offer6
- Offer3 + Offer8

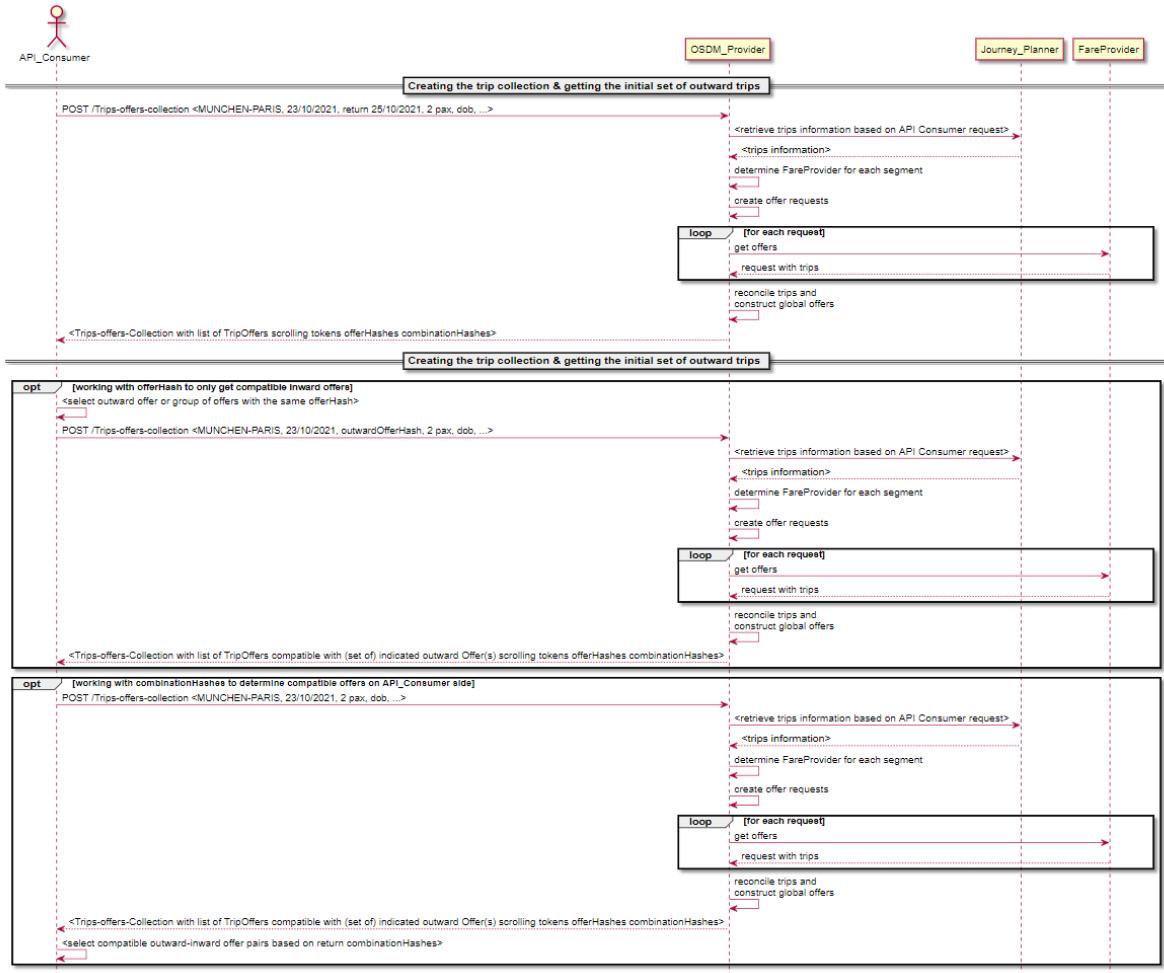
Offer7 cannot be combined with any offer on the outward set.

### ***Products Covering Both Directions***

While in most cases the two trips are materialized with distinct products/fares for the fare provider, there are fare providers still proposing unique products covering the outward as well as the return. In this case, the product element can be flagged as covering the mirrored segment as well. As for the offer construction process, the provider will simulate the two steps approach by using one of the following approaches:

- The same product covering both outward and return is proposed in the offers for the two directions
- For one of the two directions, a dummy product is returned.

Regarding the price, it can either be placed in full on the offers in the two directions (but then the total price will be incorrect when looking at the complete return travel), or split in any way desired between the outward and the return.



## Error Handling

- The referenced trip cannot be found (if e.g. the trip has expired)
- A search criteria value contains invalid value or invalid characters
- A search criteria lies outside accepted boundaries: it could be the date in the past, or too far in the future, or value outside bounds for the max number of changes
- The origin or destination is not known
- The trip search did not return any result
- No offer could be built for any of the discovered trips
- Schedule mismatch between systems

### *A Complex Example Mixing Offers and Fares*

Request from the customer expressed at the front-end

I want to go from Rotterdam to Wien Stephansplatz via Antwerp.

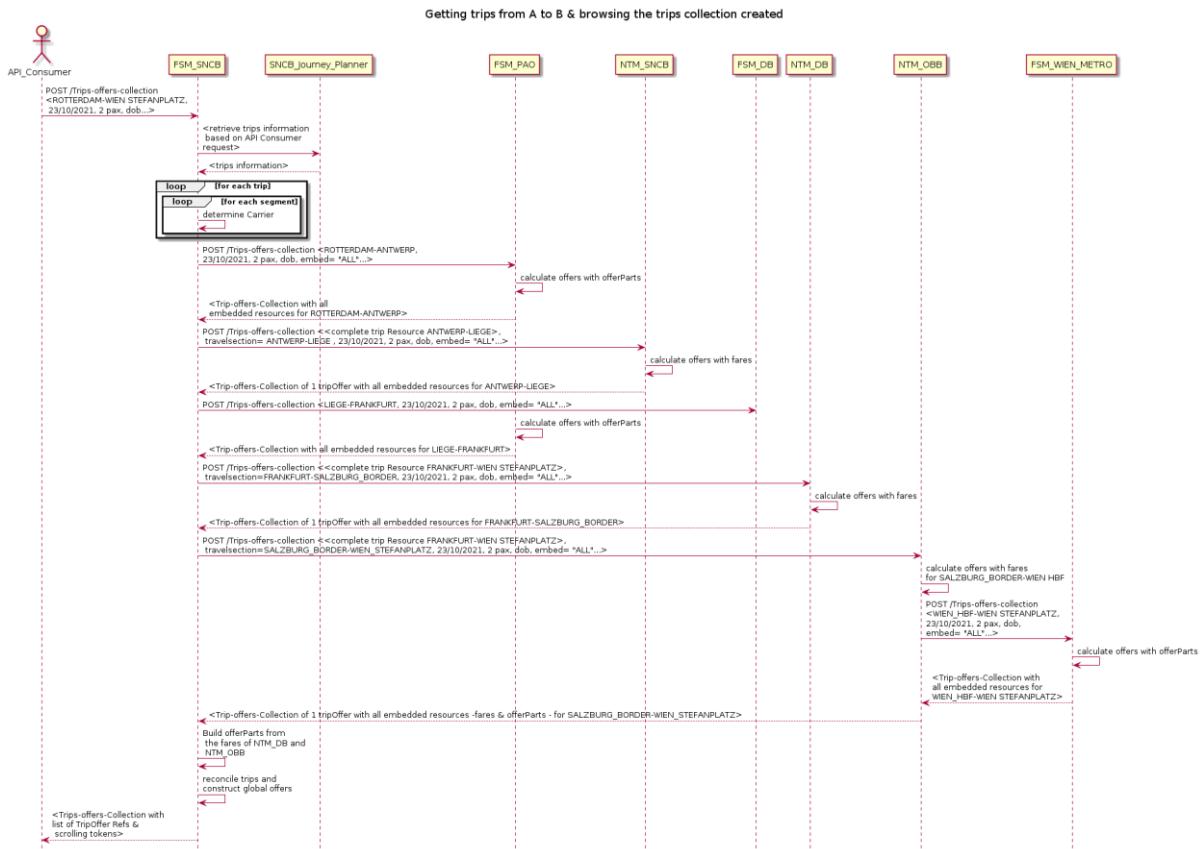
Request submitted to SNCB

Proposed trip by timetable system

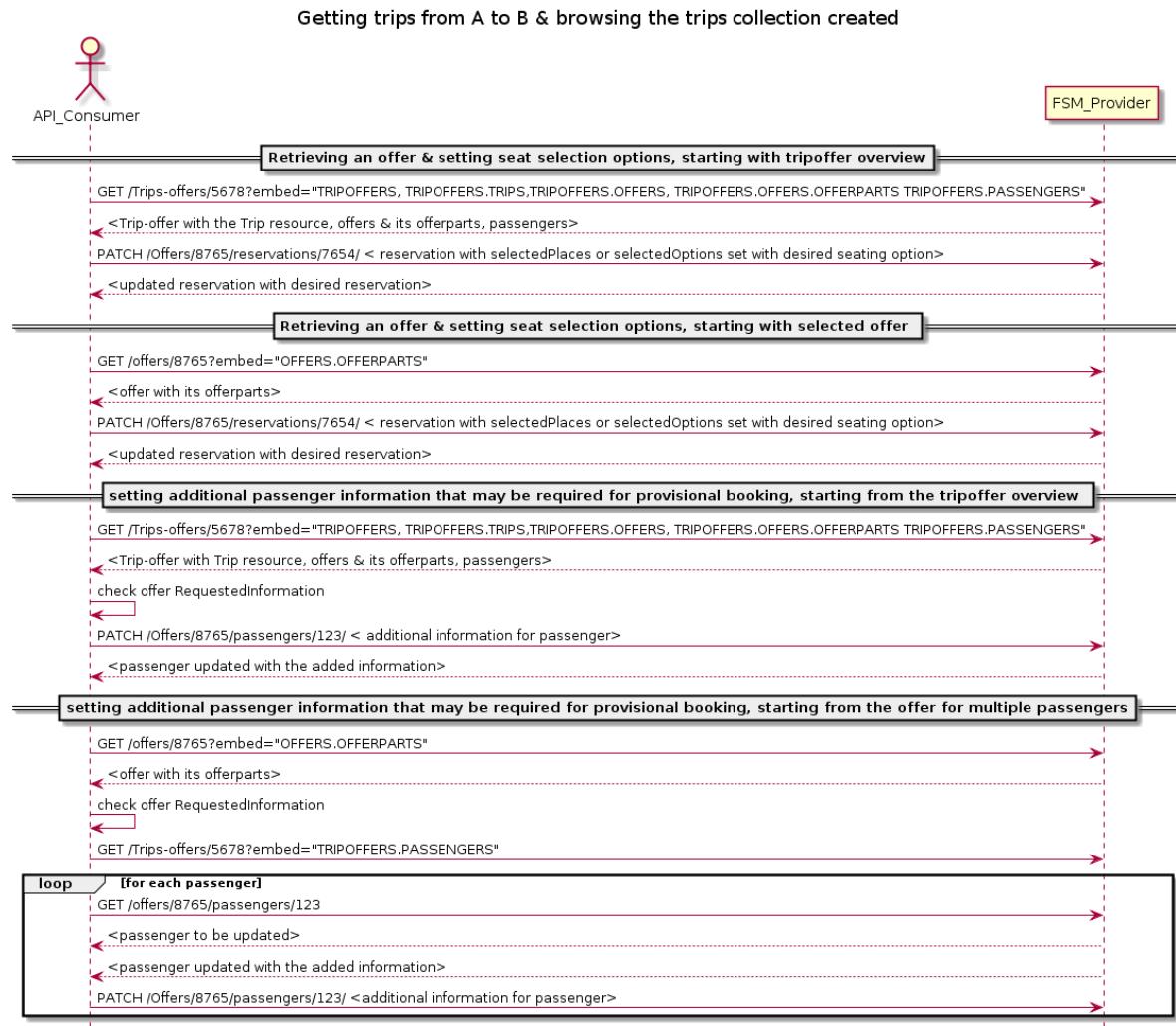
Segment	Train
Rotterdam → Antwerp	Thalys 9324 (integrated reservation)
Antwerp → Liège	IC 2345 + IR 5567
Liège → Frankfurt	ICE 122 (mandatory reservation)
Frankfurt → Wien Hbf	RailJet RJ 23 (optional reservation)
Wien Hbf → Wien Stephansplatz	Metro

Fare Provider Resolution Returns

Segment	Train	Carrier System	Carrier System (Consolidated)
Rotterdam → Antwerp	Thalys 9324 (integrated reservation)	PAO	PAO
Antwerp → Liège	IC 2345 + IR 5567	Fare SNCB	Fare SNCB
Liège → Frankfurt	ICE 122 (mandatory reservation)	NVS-Gus	NVS-Gus
Frankfurt → Wien Hbf	RailJet RJ 23 (optional reservation)	Frankfurt → Salzburg (Border)	Fare DB
		Salzburg (Border) → WienHbf	Fare ÖBB
		Frankfurt → Wien Hbf (reservation)	
Wien Hbf → Wien Stephansplatz	Metro	Fare ÖBB	



## Completing offers for provisional booking



Once an offer has been selected some additional steps can be taken to complete the information:

- Adding passenger information
- Adding accommodation preferences regarding the accommodation (most of the times a seat, but could be sleeping accommodation for night trains), either for the type of accommodation space, or its exact location.

While selecting accommodation preferences will be most of the time optional, some information (usually on passengers) may be mandatory in order to proceed with the booking.

The `RequestedInformation` property will provide the details of what needs to be specified in order to book a given offer. These details are provided in the form of a Boolean expression, referring to the offer model elements using dot notation (with the `TripOffer` as the root). For example, if it is required that name and first name are set to proceed `RequestedInformation` would be:

`"passengers.passengerDetails.firstName AND passengers.passengerDetails.firstName"`

Another example, if on top of first and last names, at least one email or one phone number is needed:

```
"passengers.passengerDetails.firstName AND passengers.passengerDetails.firstName"  
AND (passengers.passengerDetails.eMails[0] OR  
passengers.passengerDetails.phones[0])
```

By parsing this structure, the API consumer is able to identify the elements that need to be filled-in to proceed.

The two types of updates (accommodation preferences and passenger data updates) are applied using a PATCH verb on the related resources. While the resource in its whole is presented, only property explicitly listed as updateable can be provided with a value. Attempting to modify another property must result in an error.

For accommodation preferences, the following properties can be updated:

- Reservation:
  - `reservation.placeSelection.selectedOptions`
  - `reservation.placeSelection.selectedPlaces`
- Fare
  - `fare.placeSelection.selectedOptions`
  - `fare.placeSelection.selectedPlaces`

For passengers, all properties are updateable **except**

- `id`
- `reference`
- `type`

Note however that updating a property that could have influenced the product in the offer (such as date of birth or reduction cards) will not influence the offer content anymore: the booked offer will be the offer initially generated, even though the products booked may be inconsistent with the new values of the passenger properties.

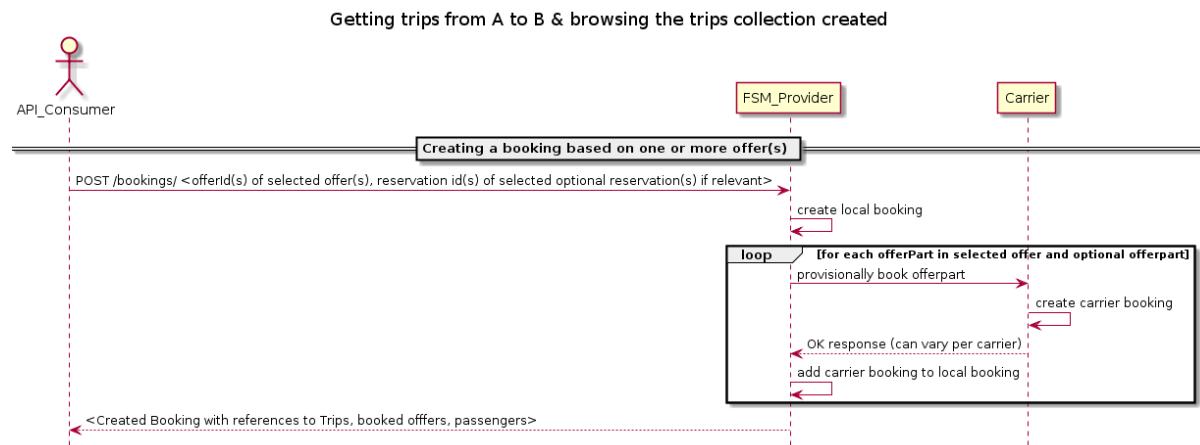
Reminder: the accommodation preferences can be found in the `reservationOptions` elements (`offer.fare|integratedReservation|reservation.placeSelection.reservationOptions`)

#### Error handling

- The requested reservation option is not available on this transport
- An invalid value is provided for a passenger property
- Attempted to modify a read-only property

## Booking Processes

### Creating a booking based on offers



Once the offer selected has been completed and all requested information provided, the consumer can continue to the booking of that offer. Along with the offer, optional or mandatory reservations or ancillaries could be booked as well. Those optional offer parts can be identified easily in the offers as they will always be linked with an admission product (in `admission.reservations` or `admission.ancillaries`). The link contains the `relationType` property, which indicates whether the pointed reservation is *included* (in which case it is not needed to explicitly add it in the booking request), *mandatory* (the reservation must be added in the booking request) or *optional* (the reservation can be added in the booking request). Ancillaries are never mandatory (only included or optional). Adding optional or mandatory elements is simply done by adding the respective offer part in the booking request `POST /bookings`.

It is also possible to book several offers in one operation to the same booking. This is especially relevant to support return trips. If this is the case, a collection of offer ids (and associated reservations and ancillaries) is given instead of just one. However, note that in this case the passengers party for all booked offers needs to be the same. To ensure this, the passenger reference of each member of the passenger party must remain the same from one offer to the other.

If the booking succeeds, a new booking resource is created. In this booking, the booked offers can be found and should look a lot like the offers as they were in the offer responses, with the exception that for (integrated)reservations and fares, the `reservedPlaces` element will now be populated with the places that have actually be assigned to the passengers for this offer part.

The passengers in the booking resources are also the same type of resources as the ones manipulated in offers. However, they have to be different resources with different ids (the passenger's references do remain unchanged).

Initially, a booking will have the status "PREBOOKED" (see also the booking status model).

At the root of the booking structure, two balance elements are provided, in order to clarify the state of the financial exchange between an API consumer or booker and the provider:

1. conditional balance is the balance of the booking that is not confirmed. It is the amount that will be due to the provider if the booking is further confirmed
2. confirmed balance is the balance of the booking that is confirmed. Unless aftersales takes place on one or more fulfilments in the booking, this amount now must be paid to the provider.

At the root of the booking structure the ticket time limit is located. This is the time for which the provider will hold a booking in pre-booked state, waiting for the confirmation while guaranteeing the booking for the given products, spaces at the announced price. Obviously, this value only has a meaning for a booking in prebooked state. A commonly accepted value would be around 30 minutes, which is normally sufficient to allow finalising the booking, while not monopolizing resources too long in case the booking is abandoned without properly cancelling it. However, some systems may decide a longer time. Obviously, the value for the booking ticket-time limit can never exceed the earliest ticket time limit of any of its offer parts.

#### Provisionally Booking a Return Trip

While this may not be true for all providers, most of them require that the outward and the return parts of a return trips are booked together in order to actually book a return-specific product. Therefore, when building a return travel, the API consumer should always specify the outward offer(s) and return offer(s) in the same POST /bookings operation.

#### Provisionally booking a trip with offers clusters

When booking for a trip for which several offer clusters were provided (see offer clusters)), the API consumer must be careful to always select one and only one offer from each offer cluster in the tripOffer. This ensures that even though the selection is done per offer cluster, the complete trip is covered exactly without any gap nor overlap. However, the provider implementers must verify and validate the set of offers selected is valid. If the trip being booked is also a return trip, then the rule applies for each direction.

#### *Handling partial success of pre-booking*

As an OSDM provider, partial pre-booking is not expected. Consequently, all pre-booking operations are either fully successful or not executed at all. However, as an allocator, an provider system may be configured in such ways that it is able to combine offers from different sub-providers (via an OSDM compliant API or not) and propose them in turn as one offer to its API consumers, as one undividable product or as a bundled pack.

Unfortunately, when the booking is attempted, the process with some of the sub-providers may encounter an error leading to the booking failing, while it will have succeeded for other parts of the offer, directed to other sub-providers. The result is a partially pre-booked booking. Since this situation is not compliant with the OSDM specification, this situation needs to be rolled back. This can be done by cancelling the pre-bookings that were successful (on an sub-provider, it would be performed using the DELETE /booking/*id*

verb). An appropriate error message is then returned in the booking response, under the form of a JSON problem element

## Booking Messages

As it was the case with offers, during the booking process, some events may occur that are worth communicating to the API consumer, while they do not really constitute an error nor should interrupt the booking process. These events and situations can be communicated through the booking messages

- Price change: the booking succeeded, but the price of the offer has been modified between the offer generation and its actual booking
- Overbooking

## Error Handling

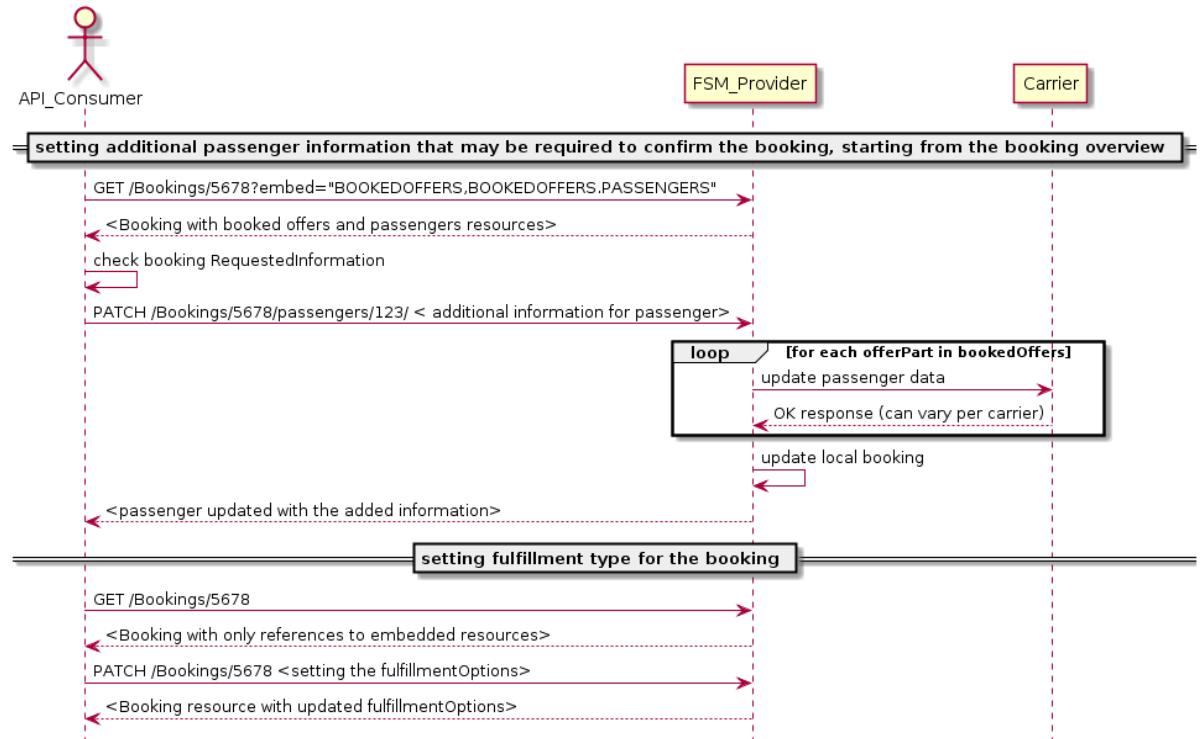
- Referenced Offer or OfferPart not found (if e.g. the offer has expired)
- No rights to access referenced offer
- Incompatible offer part with the offer
- Missing information
- Reservation to sub-system failed for one or more offer parts
- Insufficient availability for one of the requested products
- Requested place not available

## Notes

- Booking an offer will not book the reservations in the offer unless they have an "included" relationship with an admission of that offer. In order to add a non-included reservation to a booking, the reservation ids will have to be passed additionally or it will not be booked.
- It is up to the API implementing party to decide whether booked offers can have the same resource ids as the offers in the shopping stage. However, it is assumed in the specifications that this is not the case, and the API Consumer should not rely on this possibility.
- In case the passengers' details are different in the different offers added together in a booking, the passenger information of the first offer will be copied in the booking, and those of the following offers will be ignored.

## Completing Booking for Confirmation and Fulfilment

Getting trips from A to B & browsing the trips collection created



When the booking has been successfully created, some additional changes may be desired or even required before the booking can be confirmed.

- As with offers, some passenger information may be required. If this is the case, the mechanism used is exactly the same as for offers: the `requestedInformation` property at booking level will indicate which information is needed to confirm using Boolean expressions and dot notation. Updating the values is done via a PATCH on passenger subresources of the booking (as for the offer). Even if all the required data is already present, it could still be relevant to update these values. For example, a dummy date of birth might, due to the selected fulfilment type now be requested to be the exact date and require an update, even though the property is already filled-in.
- It may be needed or desired to change or set fulfilment type & options. It is however recommended to the provider implementors to set a default value for these properties (especially if only one value is possible). Note that the choice of the fulfilment type & options may impact the `requestedInformation`. This property should therefore be re-evaluated whenever the fulfilment type is modified.

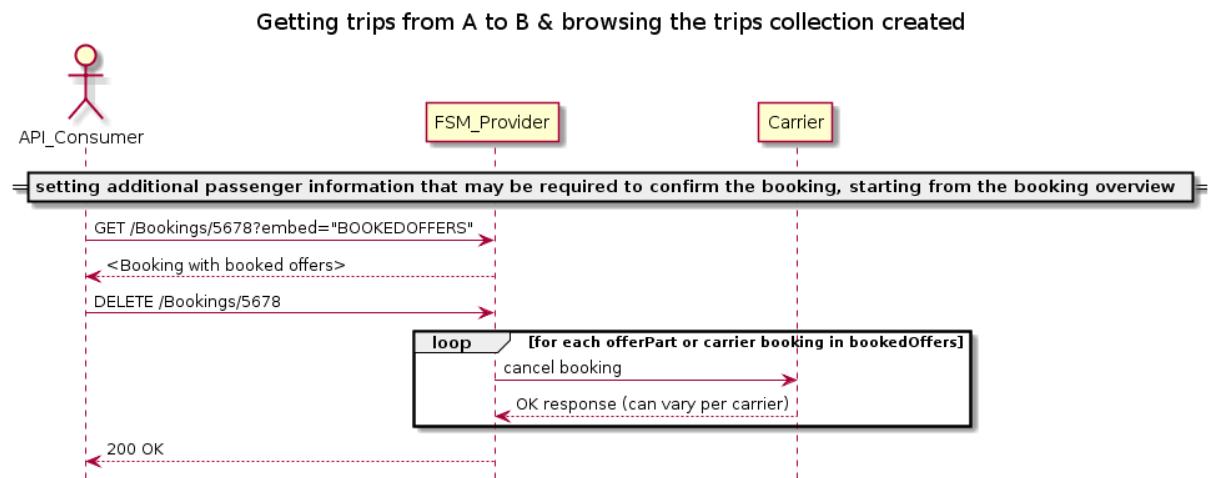
## Notes

- As in the offer, the modifications on the passenger's properties will never impact the products in the offer (thus also not the price), even if this leads to an inconsistency between the offered product and the updated passenger property.

## Error handling

- A invalid value is provided for a passenger property
- Attempted to modify a read-only property
- The booking is confirmed/refunded/cancelled and does not allow modifications

### *CANCEL a Not Confirmed Booking*



In case a prebooked booking is abandoned by its user, and this event is captured, it is recommended for the API consumer to properly cancel the booking on the provider side. In case this is not done the booking will be cancelled when the ticket time limit is reached, but in the meantime all related resources (seats etc) will remain unavailable for other requests. Upon receiving a DELETE verb for a given booking, the provider should obviously do its own cleaning as well, and if needed pass on the cancel to its sub-providers.

In case of a partial success for booking, the DELETE verb can also be used to clean-up the bookings on sub-providers where the pre-booking succeeded and who support the protocol.

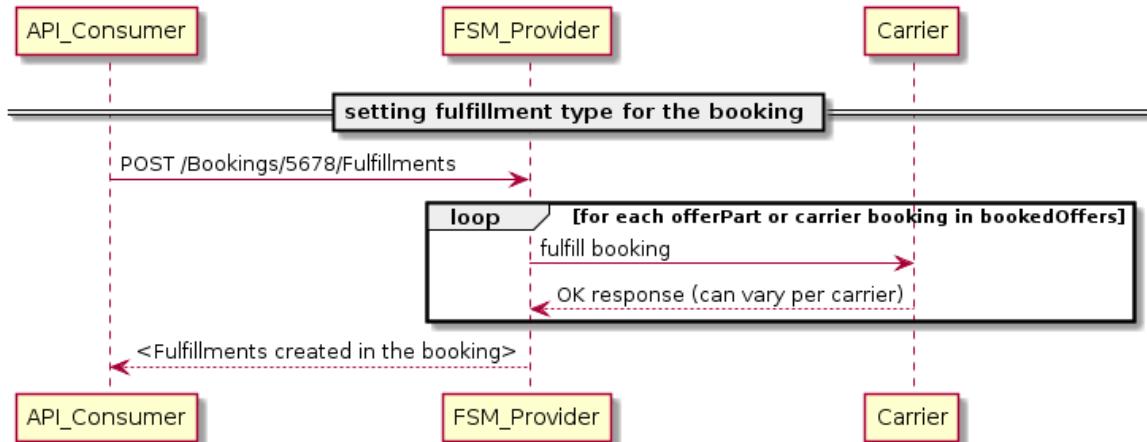
Regardless of whether the cancel occurred through an explicit DELETE or expiry of the ticket-time-limit, the booking state will then change to "CANCELLED" for a short "grace" period, before being completely cleaned-up (offer parts are well cleaned-up immediately). This grace period aims at ensuring that any ongoing operation with the booking is given sufficient time to get explicit information on the cancelled status of the booking. The choice of the duration of that grace period is left to the implementer.

## Error handling

- the booking is already confirmed
- the booking is already cancelled
- unknown error on the server side

## Confirmation and Fulfilment Processes

### Fulfilment



The fulfilment is the final step of the booking. In most cases, the booking will be confirmed and fulfilled in one step from the API consumer standpoint:

- fulfilments elements are created with the appropriate status (see below)
- the provisional balance becomes confirmed
- the status of the booking changes to FULILLED (for most systems) or CONFIRMED (see below)
- if relevant the documents elements in the fulfilment resources are created and linked

However, in case the provider acts as a distributor for products or fares actually hosted in sub-provider systems (OSDM compliant or not), a lot more takes place behind the scene. Indeed, the provider will have to

- confirm or fulfil the bookings towards all the sub providers
- retrieve the fulfilment details to populate its own booking responses (and databases, most likely)
- build the fulfilment elements
- update relevant booking properties as described above.

### Error handling

- Partial success
- Unknown error on provider side
- Missing information in the booking
- No fulfilment type selected
- Booking already confirmed/fulfilled/cancelled

### *The Special Case of Partial Success*

If a booking is composed of multiple offer parts, some of them potentially coming from sub-providers, it could be that at confirmation (or fulfillment) time, the operation only succeeds for some of the bookings. Unfortunately, a clean roll back to the previous state is not possible here for the

succeeded confirmation. The middle **Distributor** (combining offers of its sub-providers on request of its API consumer) has several options to handle the situation:

In all cases, the middle **Distributor** obviously has the option of proactively retrying to confirm on OSDM subproviders where the confirmation failed. But this may keep on failing beyond a reasonable waiting time for the API consumer. A different strategy then needs to be applied:

The first possibility is to completely clean up the booking by:

- cancelling unconfirmed content
- refunding confirmed content (with overrule if needed)
- returning an error message to the API consumer

In this case, the specific error handling remains concealed for the API consumer, who only will be informed of the final result, being the the booking has failed and been completely cancelled.

The second option is to expose the situation to the API Consumer and let it decide of the course to be taken. In this case, the resulting partial booking is returned to the API consumer with an error state

The choice of the strategy to follow here is left to the implementers. However, the implementer who would choose to expose the situation and let the API consumer handle it, also needs to implement the logic described hereunder. This may be slightly more complex than proactively cleaning up the booking in its entirety.

If this strategy is chosen, the partial booking will then be returned with the following specific characteristics:

- the returned booking has an **ERROR** status
- fulfillment is available/fulfilled only for some of the offerparts
- the confirmed balance amount only totals offer parts where the confirmation actually succeeded, while the provisional balance amounts to the total of the offer parts where the error occurred (or where the confirmation was never attempted because the error came too soon)

The following options are then available to the API Consumer:

- Explicitly request a retry on the confirmation, by re-triggering a POST or PATCH / Fulfillment. The **Distributor** will then re-attempt to confirm the not-yet confirmed content in the booking, while leaving the confirmed unchanged.
- Either directly, or after a few attempts on re-confirming, the booking needs to be cleaned-up so it can have a consistent status again (meaning the totality of the content is confirmed). To do so:
  - The API consumer must start by cancelling the non-confirmed content. He can do so by sending a PATCH on the booking where the **cleanupPartialBooking** property set on TRUE. This will result in
    - the cancellation of all non confirmed content,
    - adaptation of the balance values (provisional balance = 0, confirm balance = sum of confirmed products)
    - a reset of the booking status to **FULFILLED** (or **CONFIRMED**, depending on the confirmed content fulfillment status)
  - If deemed relevant, the API consumer can even completely remove the booking by refunding the confirmed part, if needed using an overrule code.

### **Confirm booking without fulfillment**

For some providers or products, the booking confirmation and the fulfillment step are distinct steps, while for others/most, booking confirmation and fulfillment are performed together. For products where this is the case, the fulfillment item generated by the **POST fulfillment** will show several differences from those where the product is confirmed and fulfilled in one step:

- The most obvious difference is the status, that is set to **CONFIRMED** instead of **FULFILLED**
- No document nor fulfillment item will be provided
- The fulfillment may not have a controlNumber.

In terms of process, creating this fulfillment at this stage allows an uniform confirmation process (the totality of the booking is confirmed in one step) for bookings that would mix the two kinds of fulfillment processes. The fulfillments can later get PATCHed in order to trigger the actual fulfillment.

When a confirmation request is received by the **Distributor**, it should first ensure that the operation is indeed supported for all offer parts in the booking (whether the **Distributor** is hosting those or they are coming from sub-providers). Indeed, OSDM (in MVP phase at least) will not support partial confirmation or partial fulfillment.

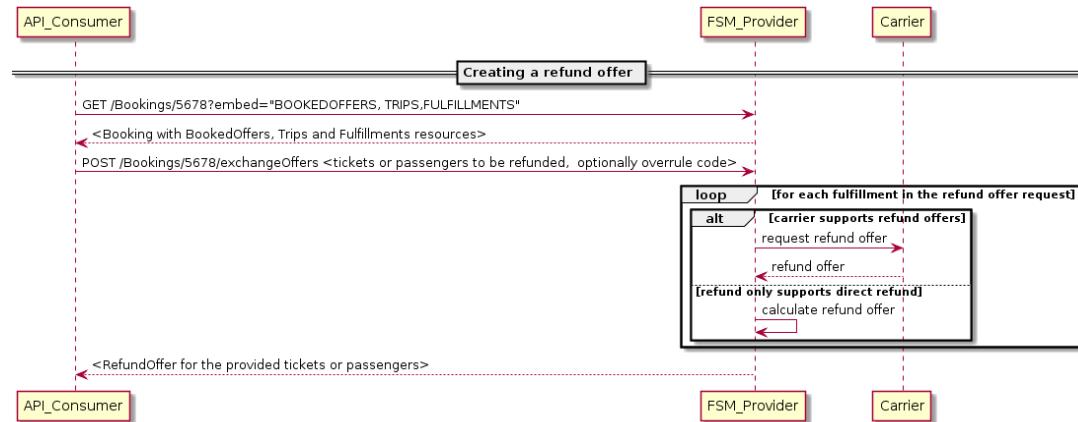
If this check is successful, then the execution of the confirm can start:

- All offer parts will be confirmed (locally or via requests to sub-providers), in parallel or sequentially
- The ticket-time-limit is invalidated (set to 0)
- The state of the booking is set to **CONFIRMED**
- The provisional balance is set to 0
- The confirmed balance is set to the total amount of the booking
- Response is sent to the API consumer

As of that point, cancelling the order becomes impossible (except for cleaning up cases, cf below) and any subsequent change should be handled as an after sales operation. Once the booking is confirmed, it becomes also impossible to modify any element in the booking (such as fulfillment type or passenger information).

## After-Sales Processes

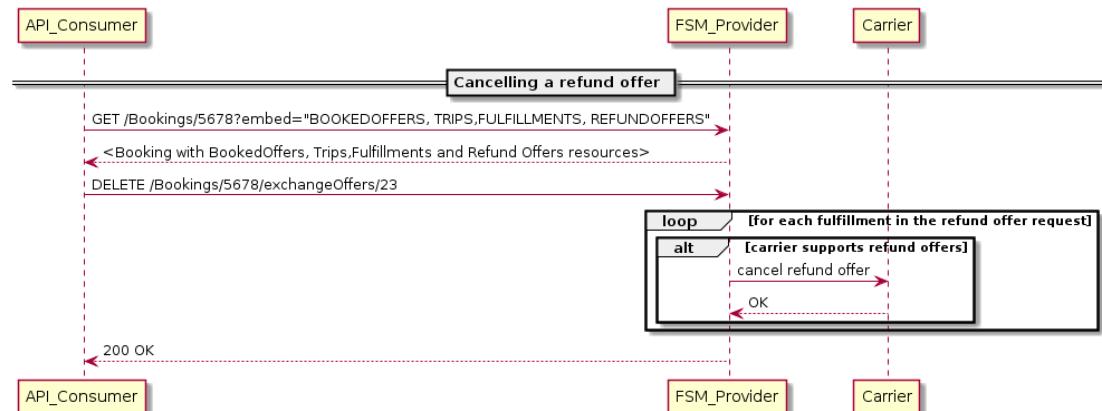
### Request a Refund offer



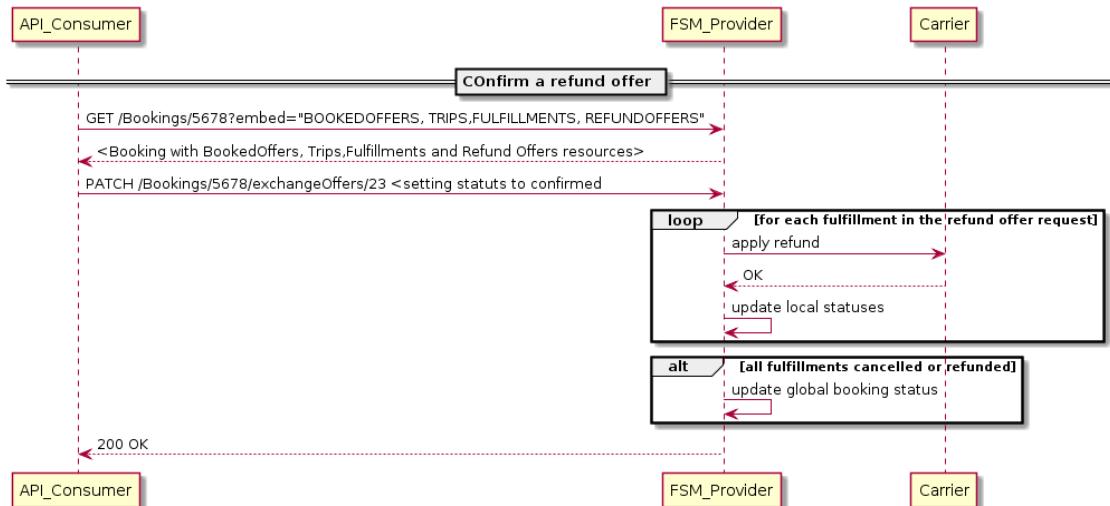
On a confirmed booking, and if it is allowed, after sales operations are also possible via the OSDM API. In OSDM, the refunds are taking place based on fulfillment resources. There is no partial refund of one fulfillment possible. This also means that in case of collective ticketing, all passengers will be refunded in one go.

In order to perform a refund, the API consumer first has to create a refundOffer in the booking where the fulfillments to refund are located with a [POST refundOffer](#). If the set of fulfillments provided is a valid set for refund, the operation creates a refundOffer that contains the information that is relevant to the refund operation at the moment the refund offer was created. This includes information such as the amount that will be refunded, any potential refund fee, etc (see the model for more details).

### Cancel a refund

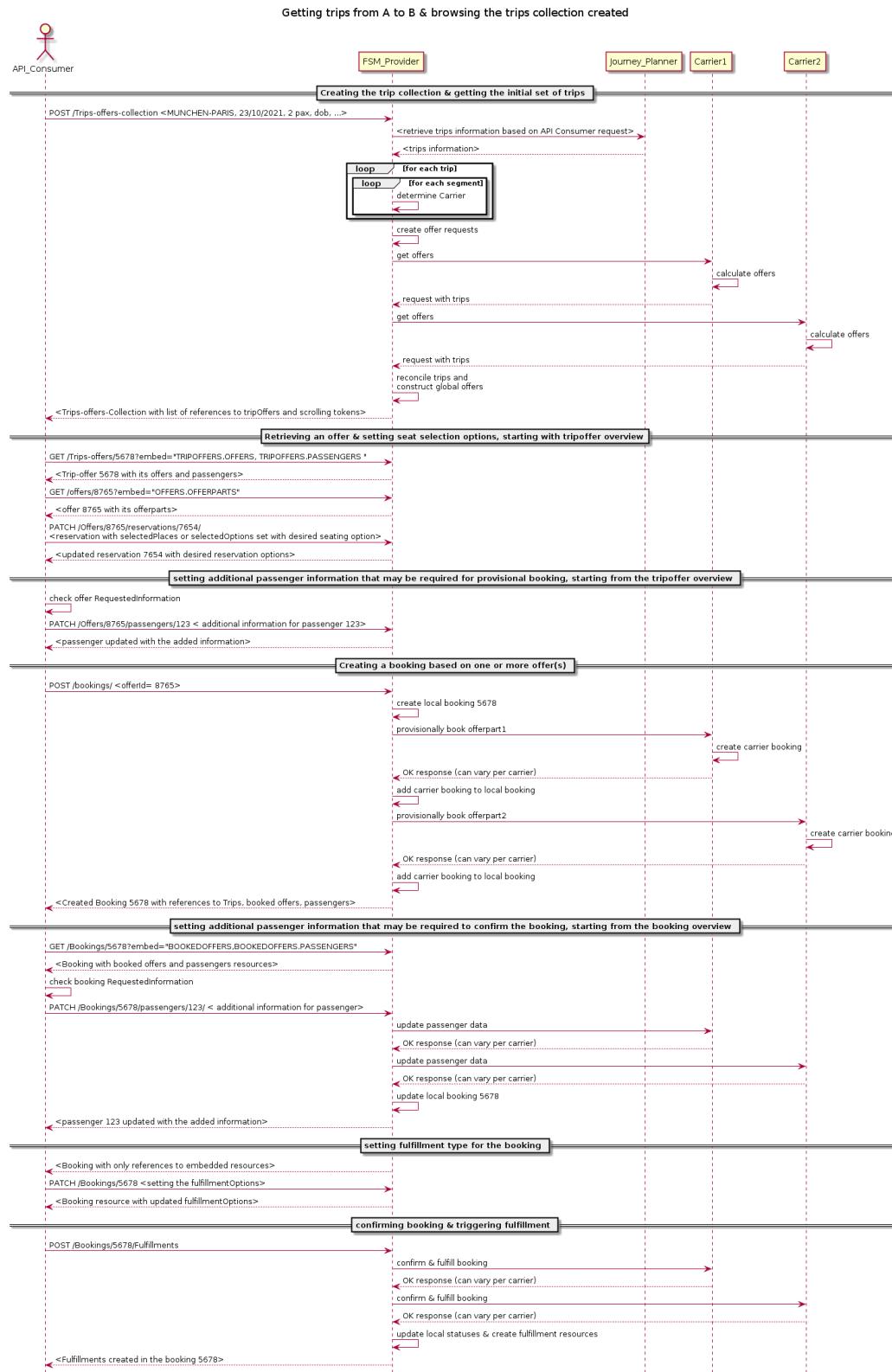


## Confirm a refund



## Example End-to-end Interaction

### Canonical case



## Service Level Requirements – Allocator

Resources	Supported Look-2-Book Rate	Response Times (msec)	Required Error Handling
/trip-offers-collection	1000:1	800	
/offers	100:1	800	
/offers/{id}/fares	5:1	600	
/bookings	1:1	80	Retry of the booking request followed by a delete booking/{bookingId} in case the booking is not needed any more.  The error handling must be repeated for 3 days but no further than the train departure or until an appropriate reply was received indicating that the booking was not made.
/offers/{id}/passengers	0,01:1	400	retry
/bookings/{id}/passengers	0,01:1	400	retry
/offers/{offerId}/fares/{fareId}	5:1	400	
/bookings/{id}/fulfillements	1:1	600	
/fulfillements	1,1:1	400	
/bookings/{id}/refundOffers	0,5:1	800	
/bookings/{id}/exchangeOffers	0,01:1	800	
/coachLayouts	Once per day	10000	
/coachLayouts/{layoutId}	2:1	300	
/coachLayouts/offerId=1234567890	2:1	300	

## Service Level Requirements – Carrier/Fare-Provider

Mandatory service level requirements to be fulfilled by a fare provider / carrier.

The Look-2-Book rate relates to the number of bookings created by the type of offer request.

The expected average response time in milliseconds that a service must provide includes the infrastructure of the provider of the server (firewalls, load balancer, circuit breaker and application) but not the network in-between the sender and receiver.

Deviating service level requirements can be defined bilaterally.

Resources	Supported Look-2-Book Rate	Response Times (msec)	Required Error Handling
/trip-offers-collection	1000:1	400	
/offers	100:1	400	
/offers/{id}/fares	5:1	300	

/bookings	1:1	200	Retry of the booking request followed by a delete booking/{bookingId} in case the booking is not needed any more.  The error handling must be repeated for 3 days but no further than the train departure or until an appropriate reply was received indicating that the booking was not made.
/offers/{id}/passengers	0,01:1	200	retry
/bookings/{id}/passengers	0,01:1	200	retry
/offers/{offerId}/fares/{fareId}	5:1	300	
/bookings/{id}/fulfillements	1:1	200	
/fulfillements	1,1:1	200	
/bookings/{id}/refundOffers	0,5:1	400	
/bookings/{id}/exchangeOffers	0,01:1	400	
/coachLayouts	Once per day	10000	
/coachLayouts/{layoutId}	2:1	200	
/coachLayouts/offerId=1234567890	2:1	200	

## Code lists

### AccommodationType

Code	Description Description
SEAT	Seat (includng reclining seats)
COUCHETTE	Couchette
BERTH	Berth
VEHICLE	Vehicle (Bicycles or car carriage)

### AccommodationSubType

The AccomodationSubType reflects the Service level defined in IRS 90918-1 / TAP-TSI B.5. In case of new service levels it has to be checked whether a new AccommodationSubType is needed.

Code	Description Description
AS	Quiet Compartment (Seat)
AM	Seat in separate Compartment
AH	Seat in historic coach
AV	Seat with front-view
D5	Couchette five-berth
D4	Couchette four-berth
D6	Couchette six-berth
F4	Ladies compartment couchette - 4-couchettes

F6	Ladies compartment couchette - 6-couchettes
D2	Cabin-couchette coach
N	Double
W	Double with shower
T	Double with shower and WC
TL	Double with shower and WC and double bed
PS	Private compartment berth
PD	Private compartment berth deluxe
DP	Private compartment couchettes
BP	private compartment seats
BE	Restaurant places in a dining car
L	Single
S	Single with shower and WC
SL	Single with shower and WC and double bed
E	Sleeperette
M	Special
P	T2
Q	T3
U	T3 with shower and WC
R	T4
AR	Wheel chair seat
DR	Wheelchair in sleeping car
AC	Seat near children's play area
O4	Motorcycle
O5	Motorcycle with sidecar
O3	Vehicle parking place category 1 - 3
O8	Vehicle parking place category 6 - 8

## BarcodeType

Code	Description
FCB	See IRS 90918-9
TLB	See IRS 90918-9
SBB	See IRS 90918-9

## CardType

The following code list defines the card types for cards used

Predefined Card-Ids	Description
LOYALTY_CARD	Loyalty card
REDUCTION_CARD	Card providing reduction
PASS	Pass for travelling

## ControlDataExchangeType

Code	Description
REGISTRY	See IRS 90918-4, data are delivered to the UIC registry
PEER2PEER	See IRS 90918-4, data are exchange peer 2 peer

## Currency

ISO4217 currency codes

## FareType

Code	Description
NRT	NRT
IRT	IRT
RES	RES (with or without supplement)
ANCILLARY	Ancillary

## ControlSecurityType

Code	Description
SIP	Security in Paper -CIT secure background tSee IRS 90918-9
SID	Security in Data (e.g. signed barcode) See IRS 90918-9
SIS	Security in System (Onlice control vs. Tickte registry) See IRS 90918-9/90918-4

## Gender

Gender code	Description
MALE	Male
FEMALE	Female
X	Diverse

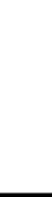
## Graphics Icons

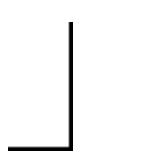
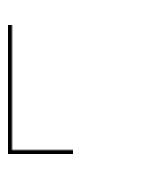
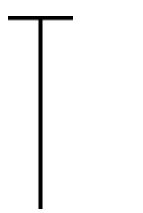
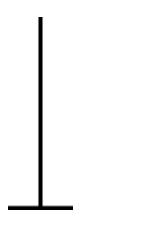
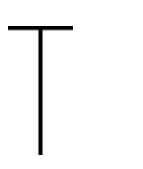
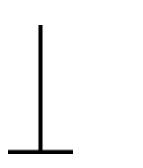
Graphic icons are used to display a coach including its facilities based on the coach layout and availability of places. The graphical items include frames and icons to display seats etc. Graphical items must be provided by the sales application of the issuer application to ensure a unique look and feel of the application.

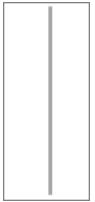
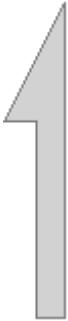
The coach layout provides only the position of graphic items (co-ordinates) not the graphical presentation at the sales application (pictures).

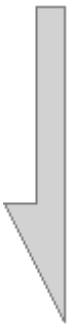
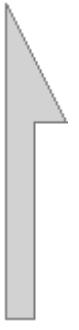
A large table spans two places, whereas a small table spans only one place. A small wall spans two places and a large wall spans 3 places. A very small wall spans one place only.

Icon	description	<u>icon</u>	Orientation uses	Mounting used
	seat	1	to right	-
	seat	1	to left	-
	seat	1	up	-
	seat	1	bottom	-
	berth	2	to left	-
	berth	2	to right	-
	couchette	3	to right	-
	couchette	3	to left	-
	small table	20	-	top
	small table	20	-	bottom

	big table	21	-	top
	big table	21	-	bottom
	wall with angle to left  height covers three places	24	-	bottom
	wall with angle to right  height covers three places	25	-	bottom
	wall with angle to left  height covers three places	24	-	top
	wall with angle to right  height covers three places	25	-	top
	small wall with angle to left  height covers two places	27	-	bottom
	small wall with angle to right  height covers two places	28	-	bottom

	small wall with angle to left  height covers two places	27	-	top
	small wall with angle to right  height covers two places	28	-	top
	wall between two compartments  height covers three places	23	-	bottom
	wall between two compartments  height covers three places	23	-	top
	small wall between two compartments  height covers two places	26	-	bottom
	small wall between two compartments  height covers two places	26	-	top
	small wall between two compartments  height covers two places	29	-	bottom

	small wall between two compartments  height covers two places	29	-	top
	very small wall between two compartments  height covers one places	32	-	bottom
	Very small wall between two compartments  height covers one places	32	-	top
	end-to-end wall	30	-	top-to-bottom
	Mobile wall between sleeping compartments  height covers three places	31	-	top-to-bottom / bottom/top
	arrow indicating upper level	51	-	top-to-bottom

	arrow indicating lower level	52	-	top-to-bottom
	arrow indicating upper level	53	-	top-to-bottom
	arrow indicating lower level	54	-	top-to-bottom
	1st class area	101	-	-
	2nd class area	100	-	-
	mobile phone area	103	-	-
	mobile phone forbidden area	104	-	-
	silence area	116	-	-
	bar area	102	-	-

	dining area	107	-	-
	bicycle area	108	-	-
	luggage area	109	-	-
	luggage locker	110	-	-
	conference compartment	111	-	-
	wardrobe	113	-	-
	dust bin	114	-	-
	children play area	117	-	-
	Space for prams	112	-	-
	family area	106	-	-
	PRM area	105	-	-
	Priority Seats	120	-	-
	Non-smoker area	118	-	-
	WC	115	-	-
	power socket	119	-	-
	WiFi area	130	-	-

	PRM toilet	131	-	-
	Air condition	132	-	-
	USB ports for charging	133	-	-
	Reclining seat	134	-	-
	Window	135	-	-
	stairs upwards	136	-	-
	stairs downwards	137	-	-
	Working compartment	144	-	-
	Cabin for allergic passengers	145	-	-
	Meeting lounge for 7 persons (Pendolino)	150	-	-
	Meeting lounge for 12 persons	151	-	-
	Space for skis	154	-	-
	Conductor working compartment	155	-	-
	Mobile phone booth	156	-	-
	T-loop (audio induction loop)	158	-	-

	Pets compartment	160	-	-
	Sleeping cabin with toilet and shower	161	-	-
	interconnectible sleeping compartments	162	-	-
	Shower	163	-	-
	Baby-care table	168	-	-
	push-button operated wheelchair ramp	170	-	-
	Sink / washbasin	171	-	-
	Ladies compartment	172	-	-

#### icons concerning the entire coach

	1st class coach	121	-	-
	2nd class coach	122	-	-
	mobile phone coach	127	-	-
	mobile phone forbidden in the entire coach	128	-	-
	1silence coach	126	-	-

	coffee bar	123	-	-
	dining car	124	-	-
	bicycle coach	125	-	-
	mixed group/individual coach	129	-	-
	WiFi area	139	-	-
	Air condition	140	-	-
	USB ports for charging	141	-	-
	Business class	142	-	-
	Panorama coach	143	-	-
	Luggage van	173	-	-
	Sleeping car	174	-	-
	Car-carrier	175	-	-

## Interface Type

Code	Description
IRS90918_1_RESERVATION_BINARY	Reservation interface according to IRS90918-1 binary message format
IRS90918_1_RESERVATION_XML	Reservation interface according to IRS90918-1 XML message format
OSDM	OSDM API

## LanguageCode

ISO-639-1 codes

## OverruleCode

Overrule code to request a refund without fees.

Code	Description
STRIKE	Refund due to strike
SALES_STAFF_ERROR	Refund due to an error made by the sales staff
PAYMENT_FAILURE	Refund as the payment failed

## Personal data items

Codes for personal data items that might be transferred to the carrier if required.

Code	Description
DATE_OF_BIRTH	Date of birth
E_MAIL	e-mail
PHONE	Phone number
FULL_NAME	Full name (first and last name)
LAST_NAME	Last name
SOCIAL_MEDIA_ACCOUNT	A social media account
LANGUAGE	Languages of the passenger
TITLE	Title of the passenger
DOC_TYPE	Type of the document if passport or id card are possible
DOC_ID	ID of the document
DOC_NAME	Name of the passenger as written on the document
DOC_CITY_OF_RESIDENCE	City of residence as written in the document
DOC_COUNTRY_OF_BIRTH	
DOC_COUNTRY_OF_ISSUE	
DOC_COUNTRY_OF_RESIDENCE	
DOC_DATE_OF_ISSUE	
DOC_GENDER	Gender as written in the document
GENDER	Gender to be used for special bookings
DOC_NATIONALITY	Nationality as given in the document
DOC_LIMIT_OF_VALIDITY	Limit of validity as written on the document
DOC_CITY_OF_ISSUE	
DOC_CITY_OF_BIRTH	
DOC_TYPE_PASSPORT	Document provided must be a passport
CARD_ISSUER	Card issuer
CARD_NUMBER	Card identifier

## Personal data transfer types

Code	Description
BOOKING	The data will be transferred via the booking services
PRE_BOOKING	The data will be transferred via the pre-booking (offer) services
SIS_CONTROL	The data will be transferred via security in system control data exchange (IRS 90918-4)
SID	The data will be transferred via a barcode

## Personal data change reasons

Code	Description
IN_GENERAL	No specific reason
MARRIAGE	

DOCUMENT_CHANGE	E.g. passport was lost and replaced
AGENT_ERROR	Personal data were entered wrongly by the sales agent

## Preferences of places

Code	Group	Description
AISLE	PLACE_LOCATION	
WINDOW	PLACE_LOCATION	
UPPER_BED	BED_LOCATION	
LOWER_BED	BED_LOCATION	
MIDDLE_BED	BED_LOCATION	
UPPER_COUCHETTE	BED_LOCATION	
MIDDLE_COUCHETTE	BED_LOCATION	
LOWER_COUCHETTE	BED_LOCATION	
UPPER_DECK	LEVEL	
LOWER_DECK	LEVEL	
COMPARTMENT	PLACE_GROUPING	
OPEN_SPACE	PLACE_GROUPING	
TABLE		Places at a table
BICYCLE	VEHICLE	Bicycle
TANDEM	VEHICLE	Tandem bicycle
PRAM	VEHICLE	Space for a pram
AIR-CONDITIONED		
PANORAMA		Panorama coach
MANAGER		Manager compartment / business
VIDEO		Place with video entertainment
CABIN8	PLACE_GROUPING	
DUO_F2F	PLACE_GROUPING	Duo face to face (2 seats facing)
DUO_SBS	PLACE_GROUPING	Duo side by side (2 seats side by side)
CLUB_2	PLACE_GROUPING	Club Duo (2 seats facing in a separate compartment)
CLUB_4	PLACE_GROUPING	Club 4 (4 seats facing)
CARRE	PLACE_GROUPING	Carré (4 seats facing normally 2nd Class)
SALON	PLACE_GROUPING	Salon (6 seats facing in separate compartment)
KIOSQUE	PLACE_GROUPING	Kiosque (special seats in edge area)
SOLO	PLACE_GROUPING	Separate place
SILENCE	USAGE	
OFFICE	USAGE	
ACC_BICYCLE	ACC_VEHICLE	Place with bicycle
ACC_TANDEM	ACC_VEHICLE	Place with tandem bicycle
BISTRO		Places in a coach with self-service bistro
CLASSIC		Classic coach
RESTAURANT	USAGE	Places in the restaurant coach
FRONT_VIEW		Places with view to the front
PRM	PRM	Places for passenger needing assistance / disabled
EASY_ACCESS	PRM	Place with easy access for PRMs
WHEELCHAIR	PRM	Wheelchair place with additional seat
WHEELCHAIR_WS	PRM	Wheelchair place without additional seat

PHONE		Places in an area with mobile phone amplifier
NEAR_ANIMALS	ANIMALS	Places close to place with animals
SILENCE		Places in silence area
NEAR_DINING		Places near the dining car
WIFI		Places with WiFi access point
POWER		Place with power socket
WITH_ANIMALS	ANIMALS	Place with animals (animals allowed)
WITHOUT_ANIMALS	ANIMALS	Place in an area where animals are not allowed
WITH_SMALL_CHILDREN	FAMILY	Places for passengers with small children
ACC_PRAM	FAMILY	Place with space for a pram
FAMILY	FAMILY	Places in family area
NEAR_PLAY_AREA	FAMILY	Places near a child play area
FEMALE	USAGE	Female compartment
DOUBLE_BED		Sleeper with double bed
CONNECTING_DOOR		Compartments with connecting Door (in Sleepers)
CONFERENCE	USAGE	Conference compartment

## Preference groups

Code	description
PLACE_LOCATION	Place location (Aisle, Window)
BERTH_LOCATION	Location of a berth or couchette
LEVEL	Upper or lower deck
PLACE_GROUPING	Compartment / Open Space
VEHICLE	Different types of vehicles
USAGE	Different usage types for different coach areas
ACC_VEHICLE	Different types of accompanying vehicles
FAMILY	Different types of places for families

## Properties of places

Code	Description
AILE	
WINDOW	
UPPER_BED	
LOWER_BED	
MIDDLE_BED	
UPPER_COUCHETTE	
MIDDLE_COUCHETTE	
LOWER_COUCHETTE	
UPPER_DECK	
LOWER_DECK	
COMPARTMENT	

OPEN_SPACE	
TABLE	Places at a table
BICYCLE	Bicycle
TANDEM	Tandem bicycle
PRAM	Space for a pram
AIR_CONDITIONED	
PANORAMA	Panorama coach
MANAGER	Manager compartment / business
VIDEO	Place with video entertainment
CABIN8	
DUO_F2F	Duo face to face (2 seats facing)
DUO_SBS	Duo side by side (2 seats side by side)
CLUB_2	Club Duo (2 seats facing in a separate compartment)
CLUB_4	Club 4 (4 seats facing)
CARRE	Carré (4 seats facing normally 2nd Class)
SALON	Salon (6 seats facing in separate compartment)
KIOSQUE	Kiosque (special seats in edge area)
SOLO	Separate place
SILENCE	
OFFICE	
ACC_BICYCLE	Place with bicycle
ACC_TANDEM	Place with tandem bicycle
BISTRO	Places in a coach with self-service bistro
CLASSIC	Classic coach
RESTAURANT	Places in the restaurant coach
FRONT_VIEW	Places with view to the front
PRM	Places for passenger needing assistance / disabled
EASY_ACCESS	Place with easy access for PRMs
WHEELCHAIR	Wheelchair place with additional seat
WHEELCHAIR_WS	Wheelchair place without additional seat
PHONE	Places in an area with mobile phone amplifier
NEAR_ANIMALS	Places close to place with animals
SILENCE	Places in silence area
NEAR_DINING	Places near the dining car
WIFI	Places with WiFi access point
POWER	Place with power socket
WITH_ANIMALS	Place with animals (animals allowed)
WITHOUT_ANIMALS	Place in an area where animals are not allowed
WITH_SMALL_CHILDREN	Places for passengers with small children
ACC_PRAM	Place with space for a pram
FAMILY	Places in family area
NEAR_PLAY_AREA	Places near a child play area
FEMALE	Female compartment
DOUBLE_BED	Sleeper with double bed
CONNECTING_DOOR	Compartments with connecting Door (in Sleepers)
CONFERENCE	Conference compartment
COMPARTMENT_SHOWER_WC	Compartment with shower and WC
COMPARTMENT_WC	Compartment with shower
SLEEPERETTE	Sleeperette (reclining seat)
BUSINESS	Manager compartment/business
CONFERENCE_ROOM	Conference room

BICYCLE_INCL_SEAT	Bicycle including seat
CLUB	Club (RENFE)
PREFERENTE	Preferente (RENFE)
TOURIST	Tourist (RENFE)
COUCHETTE_4	Four-berth couchette compartment
COUCHETTE_EMPTY	Couchette coach without equipment
WITH_DRINK	with drink
WITH_MEAL	with meal
PREMIUM	PREMIUM
WITH_BREAKFAST	Breakfast included
COUCHETTE_5	Five-bed couchette compartment
DOUBLE_BED	Double bed (Suite DeLuxe)
EXCELLENCE	Excellence Class
RESTAURANT	Restaurant (places in a dining car)
COUCHETTE_2	Two-bed couchette compartment
COUCHETTE_6	Six-bed couchette compartment
BED_SINGLE	Single sleeper compartment
BED_SPECIAL	Special (small) one bed sleeper compartment
BED_DOULBE	Double sleeper compartment
BED_T2	Small double sleeper compartment
BED_T3	Three bed sleeper compartment
BED_T4	Four bed sleeper compartment

## Reduction “cards”

The following code lists defines the commonly used cards which are not provided by a specific carrier:

Predefined Card-Ids	Issuer	Description	type
UIC_EURAIL	Eurail	Eurail Pass	PASS
UIC_INTERRAIL	Eurail	Interrail Pass	PASS
UIC_FIP_LEASURE_RED	FIP	FIP reduction (50%)	REDUCTION_CARD
UIC_FIP_DUTY	FIP	FIP duty	PASS
UIC_FIP_LEASURE_FREE	FIP	FIP free personal use	PASS
UIC_RAILPLUS	*	A pure rail plus card	REDUCTION_CARD
UIC_RIT_1	*	RIT reduction for RIT 1 members	REDUCTION_CARD
UIC_RIT_2	*	RIT reduction for RIT 2 members	REDUCTION_CARD
UIC_RIT_3	*	RIT reduction for RIT 3 members	REDUCTION_CARD
UIC_EURAIL_1	Eurail	Eurail Pass First Class	PASS
UIC_INTERRAIL_1	Eurail	Interrail Pass First Class	PASS
UIC_FIP_LEASURE_RED_1	FIP	FIP reduction (50%) First Class	REDUCTION_CARD
UIC_FIP_DUTY_1	FIP	FIP dut First Class	PASS
UIC_FIP_LEASURE_FREE_1	FIP	FIP free personal use First Class	PASS
UIC_RAILPLUS_1	*	A pure rail plus card First Class	REDUCTION_CARD
UIC_RIT_1_1	*	RIT reduction for RIT 1 First Class members	REDUCTION_CARD
UIC_RIT_2_1	*	RIT reduction for RIT 2 First Class members	REDUCTION_CARD
UIC_RIT_3_1	*	RIT reduction for RIT 3 First Class members	REDUCTION_CARD

<b>UIC_EURAIL_2</b>	<b>Eurail</b>	<b>Eurail Pass Second Class</b>	<b>PASS</b>
<b>UIC_INTERRAIL_2</b>	<b>Eurail</b>	<b>Interrail Pass Second Class</b>	<b>PASS</b>
<b>UIC_FIP_LEASURE_RED_2</b>	<b>FIP</b>	<b>FIP reduction (50%) Second Class</b>	<b>REDUCTION_CARD</b>
<b>UIC_FIP_DUTY_2</b>	<b>FIP</b>	<b>FIP duty Second Class</b>	<b>PASS</b>
<b>UIC_FIP_LEASURE_FREE_2</b>	<b>FIP</b>	<b>FIP free personal use Second Class</b>	<b>PASS</b>
<b>UIC_RAILPLUS_2</b>	*	<b>A pure rail plus card Second Class</b>	<b>REDUCTION_CARD</b>
<b>UIC_RIT_1_2</b>	*	<b>RIT reduction for RIT 1 members Second Class</b>	<b>REDUCTION_CARD</b>
<b>UIC_RIT_2_2</b>	*	<b>RIT reduction for RIT 2 members Second Class</b>	<b>REDUCTION_CARD</b>
<b>UIC_RIT_3_2</b>	*	<b>RIT reduction for RIT 3 members Second Class</b>	<b>REDUCTION_CARD</b>

## ServiceLevelType

The master code list is part of IRS 90918-1

<b>Code</b>	<b>Description Description</b>
<b>A</b>	Seat 1st Class
<b>AH</b>	Classic (places in a historic coach) (1st class)
<b>AM</b>	Business compartment 1st Class
<b>AR</b>	Wheel chair seat 1st Class
<b>AS</b>	Quiet compartment 1st Class
<b>AV</b>	Front view places (1st class)
<b>AW</b>	1st Class with WiFi Access Point
<b>B</b>	Seat 2nd Class
<b>BC</b>	place near children's play area 2nd Class
<b>BD</b>	place with animal 2nd Class
<b>BE</b>	Restaurant (places in a dining car) (2nd class)
<b>BH</b>	Classic (places in a historic coach) (2nd class)
<b>BN</b>	place close to place with animal 2nd Class
<b>BR</b>	Wheelchair seat 2nd Class
<b>BS</b>	Quiet compartment 2nd Class
<b>BW</b>	2nd Class with Wifi Access Point
<b>C</b>	Couchette 1st Class
<b>D</b>	Couchette 2nd Class
<b>DR</b>	Wheelchair in sleeping car
<b>D2</b>	Cabin-couchette coach 2nd Class
<b>D4</b>	Couchette 2nd Class four-berth
<b>D5</b>	Couchette 2nd Class five-berth
<b>D6</b>	Couchette 2nd Class six-berth
<b>E</b>	Sleeperette 1st Class
<b>EX</b>	Excellence Class
<b>F</b>	Sleeperette 2nd Class
<b>G</b>	Premium 1st Class
<b>G2</b>	Premium 2nd Class
<b>H</b>	Leisure 1st Class
<b>I</b>	- blocked
<b>J</b>	- blocked
<b>K</b>	Euraffaires
<b>L</b>	Single
<b>M</b>	Special
<b>N</b>	Double

O	Vehicle parking place
O3	Vehicle parking place category 1 – 3
O4	Motorcycle
O5	Motorcycle with sidecar
O8	Vehicle parking place category 6 – 8
OB	Bicycle
OL	Long trailer category 3 – 5
OS	Short trailer category 1 – 2
OT	Tandem Bicycle
P	T2
Q	T3
R	T4
S	Single with shower & WC
SL	Single with shower & WC & double bed
T	Double with shower & WC
TL	Double with shower & WC & double bed
U	T3 with shower & WC
V	- blocked
W	Double with shower
X	Club
Y	Preferente
Z	Tourist
BP	private compartment seats
F4	Ladies compartment couchette 2nd Class 4-berth
F6	Ladies compartment couchette 2nd Class 6-berth
DP	Private compartment couchettes
PS	Private compartment berth
PD	Private compartment berth deluxe

## Sales Channel Type

Channel type code	description
STATION_OFFICE	
VENDING_MACHINE	
WEB_MOBILE_SALE	
ON_BOARD_SALE	
AGENCY_SALE	

## Service Brand Codes

Codes from the official UIC/TAP-TSI code list must be used.

## Service Class / Classic Class

Code	Description	Classic Class
BEST	1st class A (highest service level)	FIRST
HIGH	1st class B - todays standard 1st class	FIRST
STANDARD	2nd class C	SECOND
BASIC	2nd class D (lowest service level) - todays standard 2nd class	SECOND

Examples:

ÖBB: BEST = Business, HIGH = 1<sup>st</sup> class, STANDARD = 2<sup>nd</sup> class

DB: BEST = 1<sup>st</sup> class, STANDARD = 2<sup>nd</sup> class

RHB: BEST = Excellence Class, HIGH = 1<sup>st</sup>. class, STANDARD = 2<sup>nd</sup> class

## Station codes

The station codes are the station codes defined in the timetable data according to TAP-TSI. A valid source for the code list is MERITS.

## SupportedOnlineServices

Online services supported by the carrier.

Supported online services	Description
OFFER	Offer service (without after sales offers)
BOOKING	Preliminary booking, confirm booking and cancellation
RESERVATION_LEGACY_918_1	Services according to the 90918-1 XML or binary specification
RESERVATION_PREF	Place allocation using preferences
RESERVATION_GRAPH	Place allocation using graphical place display
UPGRADE	Upgrade to an existing booking
INCREASE	Exchange with an increase of the number of passengers
DECREASE	Exchange with an decrease of the number of passengers
EXCHANGE	Exchange with the same number of passengers
FULFILLMENT	Only in case of separate ticket! – providing an entire tickets
FULFILLMENT_ITEMS	Providing security items for a ticket (e.g. additional bar codes)

## TaxScope

Code	description
INTERNATIONAL	VAT applies in international tickets only This needs to be applied in case the fare is integrated into an international ticket
NATIONAL	VAT applies in national tickets only
SHORT_DISTANCE	VAT applies in short distance tickets only
LONG_DISTANCE	VAT applies in long distance tickets only This needs to be applied in case the fare is integrated in a longer journey.

## TimeReference

Code	Description
BEFORE_DEPARTURE	Time value calculated relative to the departure (subtract from departure) The time zone of the departure station applies.
AFTER_DEPARTURE	Time value calculated relative to the departure (add to departure) The time zone of the departure station applies.
AFTER_SALE	Time value calculated relative to the sales time (add to sales time). The time zone of the sale applies.
BEFORE_START_VALIDITY	before the start of the validity. The time zone of the departure station applies.
AFTER_END_VALIDITY	after the start of the validity. The time zone of the departure station applies.

## TimeUnit

Code	Description
DAY	
MINUTE	
HOUR	

## TransactionType

Transaction type used in after sales rules for fares

Code	Description
REFUND	
EXCHANGE	
CARRIER_CHANGE	Exchange with a new fare of another carrier
EXCHANGE	Exchange with a new fare of the same carrier
UPGRADE	

## TransportMode

Corresponding to numerical codes in TAP-TSI / MERITS

Code	Description
HIGH_SPEED_TRAIN	
HISTORIC_TRAIN	
INTERCITY	
REGIONAL	
INTERREGIONAL	
TRAIN	
URBAN	
TRAM	
UNDERGROUND	
NIGHT_TRAIN	
SHARED_TAXI	
MOTOR_RAIL	Car carriage trains
MOUNTAIN_TRAIN	

PLANE	
COACH_GROUP	Group of coaches included in multiple trains (through coaches)
SHIP	
BUS	

## TransferTypes

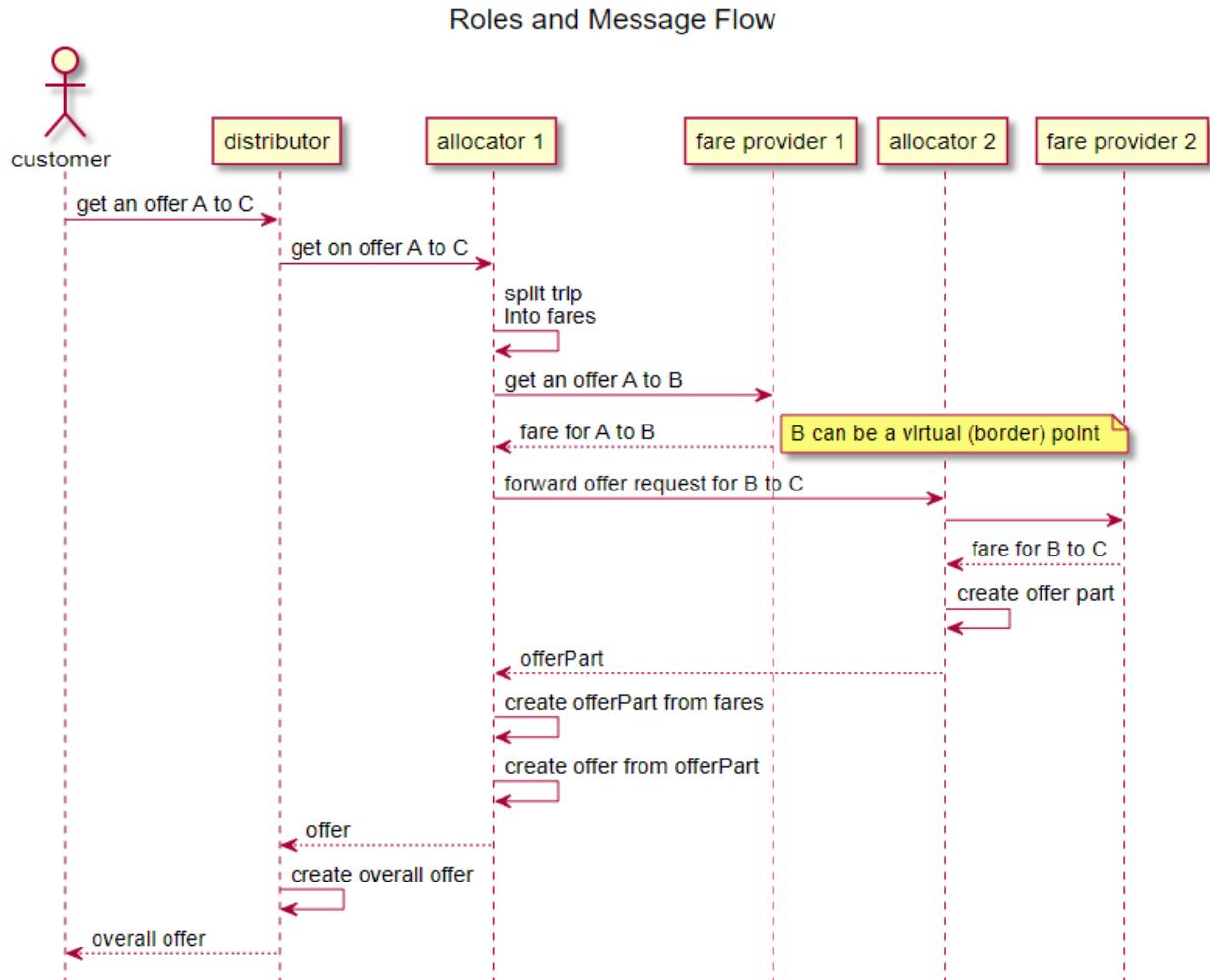
Code	Description
WALK	A walk
OTHER	Other types of transfer (e.g. taxi, local city transport not included in the offer,...)

## PassengerTypes

PassengerType/Anxillary	Description	Anxillary service	bulk	Online request	Online reply
YOUNG_CHILD	Young child - defined by the carrier depending on the age		X		X
CHILD	Child - defined by the carrier depending on the age		X		X
YOUTH	Youth - defined by the carrier depending on the age		X		X
ADULT	Adult - defined by the carrier depending on the age		X		X
SENIOR	Senior - defined by the carrier depending on the age		X		X
FAMILY_CHILD	Child associated with a family travelling together		X	X	X
ACCOMP_PRM	Accompanying Person for PRM		X		X
PRM_CHILD	Handicapped young child accompanied by one person where the usual child according to the age price would be zero and the accompanying person would also be free		X		X
WHEELCHAIR	Passenger with wheelchair			X	
PERSON	Used in requests together with date of birth			X	
PRM	Person with reduced mobility – to be used in case of accompanying person or dog, date of birth must be provided additionally			X	
DOG	A dog	X	X	X	X
PET	A pet			X	X
LUGGAGE	Oversize luggage	X	X	X	X
BICYCLE	A bicycle	X	X	X	X
ACCOMP_DOG	An accompanying dog for a PRM	X	X	X	X
CAR	A car for carriage trains	X	X	X	X
MOTOCYCLE	A motorcycle for carriage trains	X	X	X	X
TRAILER	A trailer for carriage trains	X	X	X	X

## Compliance

To be compliant with the OSDM specification in total a party must be compliant with the offline as well as the online part of the specification. However, a party can decide based on their business need to implement the offline or the online part online based on the role they want to play in the distribution and sales process.



## Compliance with the Offline Part

An implementation of the bulk data exchange specification is compliant with the specification if

- A feature specified in the data structure is implemented  
Or
- A fare providing the feature in its data is excluded from sale

A system receiving data for a fare must be able to understand all features and rules of the fare defined in the data and obey these features and rules or must not sell such a fare.

## Compliance with the Online Part

### Compliance as an Allocator

The following services/features are mandatory/optional to implement:

Resources	Description	Need to support
/locations /trips	Resources to search for trip and locations	Mandatory. Especially for train stations UIC code need to be supported.
/trip-offers-collection /trip-offers /offers	Resources to get bookable offers	Mandatory is to provide at least admission offers.
/offers/{id}/admissions /offers/{id}/reservations /offers/{id}/ancillaries /offers/{id}/fares	Resources to manipulate parts of an offer consisting of, e.g., admissions, reservations or ancillaries;	Mandatory
/offers/{offerId}/fares/{fareId} <b>/offers/{offerId}/reservations/{reservationId}</b>	Resources to manipulate seat assignment	Conditional; mandatory to be supported in case seat assignment is provided.
/bookings	Resources to manipulate bookings	Mandatory
/offers/{id}/passengers /bookings/{id}/passengers	Resources to manipulate a passenger's information at every stage of the flow	Mandatory
/products	Resources to retrieve products information on one or more products	Mandatory
/bookings/{id}/fulfillements /fulfillements	Resources to retrieve fulfilments, e.g. tickets	Mandatory is to support entire tickets.
/bookings/{id}/refundOffers	Resources to get and accept a refund offer	Mandatory is to support full refund.
/bookings/{id}/exchangeOffers	Resources to get and accept an exchange offer	Optional
/coachLayouts	Resources to get layouts of coaches	Optional

All non-functional requirements defined in the services must be fulfilled.

Especially all implementations must support the Tolerant Reader pattern. This integration pattern helps creating robust communication systems. The idea is to be as tolerant as possible when reading data from another service. This way, when the communication schema changes, the readers must not break.

### Compliance as a Fare Provider

The following services/features are mandatory/optional to implement:

Resources	Description	Need to support
/locations /trips	Resources to search for trip and locations	Optional
/trip-offers-collection /trip-offers /offers	Resources to get bookable offers	Mandatory is to provide fare offers on a /trip-offers-collection.

		Optional: /offers for non trip based offers.
/offers/{id}/admissions /offers/{id}/reservations /offers/{id}/ancillaries /offers/{id}/fares	Resources to manipulate parts of an offer consisting of, e.g., admissions, reservations or ancillaries; if permitted, also fares are offered.	Mandatory is to support pre-booking and booking of fares.
/bookings	Resources to manipulate bookings	Mandatory is to support bookings consisting of fares except in the special case of direct sale fare offers
/offers/{id}/passengers /bookings/{id}/passengers	Resources to manipulate a passenger's information at every stage of the flow	Conditional, mandatory to be supported in case personal data are required by the fare provider
/products	Resources to retrieve products information on one or more products	Mandatory
/offers/{offerId}/fares/{fareId} /offers/{offerId}/reservations/{reservationId}	Resources to manipulate seat assignment	Conditional; to be supported in case seat assignment is provided.
/bookings/{id}/fulfillements /fulfillements	Resources to retrieve fulfilments, e.g. tickets	Conditional, mandatory to be supported in case fulfillment items need to be provided
/bookings/{id}/refundOffers	Resources to get and accept a refund offer	Mandatory is to support for full refund, partial refund is optional Not required in the special case of direct sale offers
/bookings/{id}/exchangeOffers	Resources to get and accept an exchange offer	Optional
/coachLayouts	Resources to get layouts of coaches	Optional

All non-functional requirements defined in the services must be fulfilled.

On a technical level the implementation must support the Tolerant Reader pattern. This integration pattern helps creating robust communication systems. The idea is to be as tolerant as possible when reading data from another service. This way, when the communication schema changes, the readers must not break.

An implementation of specification is compliant with the specification if for a given version

- A feature specified in the data structure is implemented
- Or
- A fare providing the feature in its data is excluded from sale

A system receiving data for a fare must be able to understand all features and rules of the fare defined in the data and obey these features and rules or must not sell such a fare.

**Copyright 2020 – International Union of Railways (UIC)**

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.