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
2021 February	Version 1.1 OSDM (Offline & Online)
2021 June	Version 1.2 OSDM (Offline & Online)
2021 November	Version 1.3 OSDM (Offline & Online)
2022 February	Version 1.4 OSDM (Offline & Online)
2022 October	Version 2.0 OSDM (Offline & Online)
2023 February	Version 3.0 OSDM (Offline & Online)
2023 October	Version 3.1 OSDM (Offline & Online)
2024 February	Version 3.2 OSDM (Offline & Online)

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

Inhalt

Foreword

Current Situation

Offline Distribution

The first main goal of this specification is to addresses 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 possible only once a year
- 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 distributor to offer through-tickets based on different conditions-ranges (e.g. fully flexible, semi flexible, 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 are support the distributor, the distributor 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:

1. Searching for trips
2. Getting offers
3. Booking an offer
4. Confirmation of the booking
5. Fulfillment of the booking

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

1. Getting a refund/exchange offers
2. Booking a refund/exchange offer
3. Fulfillment 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.

What’s New in OSDM Version 3.2

The following features have been added with version 3.2 of OSDM.

- Partial refund has been simplified (All)
- Split of a booking improved (Amadeus)
- Second last name for Spain (Renfe)
- Reservation on leg parts (if no seat is available for the entire leg) (Sweden)
- Graphical reservation offer selection for busses in Sweden (Sweden)
- Special accounting reference (DB/Eurostar/SNCF)

Other Work

Other work addressed includes:

- Improving the API’s description
- Fixing minor inconsistencies
- Incorporating feedback from the first implementors

Breaking Changes

We adhere to Semantic Versioning, thus this version is *backwards compatible with version 3.0.*

A detailed list of non-breaking changes are available in the changelog.

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
Admission	The right to travel on a train, aka. as ticket.
Ancillary	An extra service like meal or WIFI.
Complaint	Complaint of a customer due to a service not provided as contracted. The handling of complaints in case of delays is subject to EU PRR and COTIV defining minimal refund amounts and time-line for decisions.
Conditions	Conditions that limit the use of a travel contract or the changes of a travel contract.
e-Ticket	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.
Fare	Proposal to purchase a transport service for specific passenger subject to specific conditions for a specific price. Handling fees by the distributor or retailer are not included.
Fare structure	A fare is the fee paid by a passenger for use of a public transport system: rail, bus, taxi, etc. In the case of air transport, the term airfare is often used. Fare structure is the system set up to determine how much is to be paid by various passengers using a transit vehicle at any given time.
Fulfillment	A fulfillment is a document (either for paper printing or electronically) provided to the passenger to prove his travel right,facilitate access to trains and stations (e.g. via gates), provide further information on the travel and provide access to services either directly or via exchange (voucher).
Individual ticketing	A separate ticket is created per passenger.
Individual contracts	A separate ticket is created per passenger and these tickets can be treated as individual contracts of carriage. After sales transactions can be applied independently per passenger and

Term	Definition
	ticket.
Integrated Reservation Ticket (IRT)	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 distributor. The ticket is valid on that train on a certain day only.
Non-integrated Reservation Ticket (NRT)	A ticket not including an integrated reservation. Multiple distributors can create tickets for the same route independently. The distributor 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 retailer. These conditions depend on the fare providers and the distributors (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.
Offer Part	An abstraction of things that can be offered. Can be of type Admission, Reservation or Ancillary.
Reimbursement Claim	Special customer request for refund after the ticket validity has started where the customer needs to provide proof for not using a ticket.
Tariff	A schedule of prices and conditions for the sale or rental of a product or service. In UIC context the term “tariff” is used for fare structure.
Ticket	Medium of the passenger to prove a travel contract. Contains a description or a reference to the travel contract. The ticket might provide direct proof of the travel contract via its security features or via a link to the contract. The tickets are sold by retailers. The assembling of the ticket is done by the distributors(s). The distributor holds the master ticket data/contract of the sold ticket. A ticket can include multiple fare providers in the travel contract. The control of one ticket is done by one or many Ticket Controlling Organizations (TCO) 1. Other means of ticket checking (e.g. gates) will also be named TCOs.
Sales fee	Sales fees are fees added by the distributor or retailer 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.
Station fee	A station fee is a fee for the use of a station by a passenger. It might be included in the fare provider offer.
Station	A station is a is a location where trains or busses regularly stop to load or unload passengers.

Acronyms

Acronyms	Acronym Description
ERA	**European Union Agency for Railways** (https://www.era.europa.eu/)
EWT	**East West tariff**: Tariff used for Non-Integrated-Reservation-Tickets. The fare data model follows UIC IRS 10108.1.
FCB	**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).
IRT	**Integrated Reservation Tariff**: Tariff used for Integrated-Reservation-Tickets. The fare data model replaced UIC IRS 10108.2.
IRT	**Integrated Reservation Ticket**: Ticket including mandatory reservation.
JWT	**JSON Web Token**: Specification to transport authentication information used by the OAUTH2 authorization protocol. JSON Web Token - RFC 7519
NRT	**Non-Integrated Reservation Tariff**: Tariff used for Non-Integrated-Reservation-Tickets. The fare data model replaced UIC IRS 10108.1.
NRT	**Non-Integrated Reservation Ticket**: Ticket not including an integrated reservation.
REST	**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.
RPT	**Rail Pass Ticket** Ticket valid for multiple trips in a region
SiP	**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)
SiD	**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).
SiS	**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 passenger. The ticket control id (key to the ticket) or

Acronyms	Acronym Description
	personal data of the passenger (name, date of birth,) are used to retrieve the ticket. (UIC IRS 90918-0).
SiV	**Security by Visual** elements: The ticket is controlled by a visual element printed with the ticket data on blank paper of 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).
TAP TSI	European regulation on Interoperability: **Telematics Applications for Passenger Service Technical Specification for Interoperablility**
TCO	**Ticket Controlling Organization**
TLT	**Train Linked Ticket**: Ticket not including a reservation but restricted to a train run (or multiple train runs along the route).
TLB	**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.
UML	**Unified Modelling Language**: A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.
UUID	**Universally Unique Identifier**: Standard to create a unique id. The specification is published as ISO/IEC 9834-8:2005.

Requirements

Common Functional Requirements

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. Per default the scale is set to two.

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 for the given business process can be transferred between the parties involved.

Functional Requirements of the Retailer

Requirements on Passenger

A passenger is the person travailing on a vehicle. A passenger may not necessarily be the person who has performed 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 must be collected sparsely and only if needed for a dedicated process step. Passenger details providing personal information shall only be used to meet the requirements of 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 personal 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, bicycles, cars, motorcycles 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 station codes most be supported.

To support other means of transportation the types can potentially be 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
- `tripLegs`: a list of tripLegs

A trip is composed of one or more `tripLegs` and can be of one the following type:

- **TimedLeg**: A type of leg with a timetable schedule such a provided by public transport
- **TransferLeg**: A type of leg that links two legs such as walking from one stop to another

- **ContinuosLeg**: A type of leg that is not bound to a timetable. This leg is mainly aimed at new modes such as scooter, taxis,..

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

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

A transfer is a special kind of tripLeg, defining how long the transfer takes.

Requirements on Offers

An overall offer presented to a retailer or a distributer bundles offers that contain admissions, reservation and ancillaries.

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 at least one tripLeg of the 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.

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. Additionally, an included reservation does not reference a product.

An admission may be linked mandatorily or optionally 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 have been applied.

A reservation is linked to one or more passengers.

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

A reservation has a one-to-one 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. Examples for ancillary services: Wifi access or on-board meal.

An ancillary has a price.

An ancillary can be **OPTIONAL, MANDATORY,INCLUDED** to an admission or reservation.

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 one-to-one relationship to a product.

Requirements on Fees

Fees of a distributor or a carrier can be required upon the sale of reservations, admissions or ancillaries or collectively for a set of reservations, or for a booking.

A fee has a value.

A fee applies to one or more offer parts, and to one or more passengers.

A fee is automatically added to a booking if the relevant offer parts are included, e.g. a reservation fee is automatically added when any reservation has been selected.

Whether a fee is refundable is defined by the tariff.

The state of a fee depends on the state of the associated product.

Requirements on Prolonging an Offer

Optional requirement

Especially for travel agency a common use case is to prolong the validity of an offer, so that the customer has more time to decide. This process can also be understood as putting an offer on hold.

If a supported by the underlying system it must be possible to request an on hold offer to put an offer on hold for given time period.

If accepted a deposit will be charged from the requestor. Additionally, a fee can be requested once the prolonged offer is actually booked.

Requirements on Offer Combination

Offers from different providers might have a restriction to be sold in combination only.

The combination might not be be required by all offers. E.g. an offer from provider A might only be sold if offer B of another provider is sold but offer B can be sold stand alone.

The combination logic needs to be fast (<20ms).

Requirements on Round Trips

Round trip offers should be possible considering both trips when making the offer.

Support for round trips consisting of one or two products need to be supported.

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

To support stateless booking a explicit pre-booking step is *not* supported by design.

A booking has a purchaser which has booked the booking. The booking can but most not be a passenger.

Booking must be supported by all parties.

Requirements on Products

A product must contain the following information:

- An id uniquely identifying the product, e.g. "Sparschiene".
- A textual description of the product.
- A structured description of the sales or after-sales conditions which are machine interpretable as well as a human readable.
- A indication whether a product is *refundable*, *refundable with conditions* or *not refundable*.
- A indication whether a product is *exchangeable*, *exchangeable with conditions* or *not exchangeable*.
- The service class describing the level of comfort.
- Define the supported fulfillment media types.

A product does not have a price, as the price is bound to an offer as an instantiation of a product.

Requirements on Fulfillment

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

A fulfillment must reference fulfillment documents (aka. tickets). Fulfillment documents in form of a UIC PDF ticket most be supported by all parties. A fulfillment is provided for a specific FulfillmentType (e.g. `CIT_PAPER`) and FulfillmentMedia specifying the format (e.g. `RCT2`).

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

Requirements on Documents

It must be possible to add and delete a document to a booking. Such a document can be a booking receipt, a CO2 report, a explanatory documentation to a complaint or any other general document.

A document can reference a booking or a passenger in a booking.

Requirements on Refund

For a given a booking a refund can be requested.

A refund can have a fee.

Cancellation (a.k.a. revoke) is a special kind of refund where no fees apply, and the complete amount is returned triggered by an overrule code.

Cancellation must be supported by all parties.

Total refund must be supported by all parties.

Requirements on Partial Refund

Partial refund allows to remove passengers and booking parts (only if supported by the underlying tarif) from a booking.

A partial refund can have a fee.

Partial refund may be supported by all parties.

Requirements on Exchange

Exchange allows to change trip and passengers.

An exchange can have a fee.

Exchange may be supported by all parties.

Requirements on Seat Change

Optional requirement

If supported by the underlying system, it must be possible to change of seat after the booking is confirmed and fulfilled. This change can either be free or for a fee when requested by the carrier.

The following scenarii exist:

1. Specific seat and coach requested: optionally, a seat map can be requested, so that the user knows which seat is available. The user can then select a specific coach and seat number.
2. Near to a given seat: The user provides in the request a seat number he wishes to be seated next to.
3. Seat preferences: The user provides seat arrangement such as window, aisle etc…

Some providers give the possibility to up-sell to a better seat after booking. This is an adjacent but not equivalent case, as the change of seat should not affect allocation or update the inventory.

Requirement to Release Allocated Resources

Optional requirement

For some systems (e.g in the French or Swedish market) it must be possible to allocated resources such as places, meal or others.

Requirement to Cancel a Fulfillment

Optional requirement

For some systems (e.g. in the French) it must be possible to cancel a fulfillment (*Void PNR*).

Requirements on Complaints

It must be possible to manage a complaint according to (EU) **2021/782**.

It must be possible that the claim is managed by a separate person on behalf of the passenger.

It must be possible to retailers or distributors to submit a claim for a booking or parts of it (e.g. in case of a return trip) where the service has not been provided as promised. This includes:

- Delays of trains
  - In case the fare was not restricted to specific trains this also includes trains which were not mentioned in the trip of the booking
  - A description of the cause is needed delayed trains incl. the delay, missed connections, cancelled trains, …

    - Abandoned travel (e.g. in case of a return, subsequent trains, …)
- Service derogation (e.g. first class not available, …)

It must be possible to provide documentation to support the claim (e.g. scanned manual remark of train staff) (Alternatively require exchange via **UIC 90918-4**).

It must be possible to demand that the claimed amount is issued as a voucher or transferred to a bank account.

It must be possible to lookup the status of the claim made.

In case a claim is accepted or rejected the actor that made the request must be informed. The accepted or rejected claim must provide an explanation for the decision especially if the amount covers the ticket price only partially.

It must be possible to inform the distributor / fare provider that the payment was made and the claim is settled.

It must be possible for a distributor to inform the fare provider that the claim was accepted because the fare provider missed a legal timeline to handling the claim.

Requirements on Reimbursements

Customers who have bought a ticket which allows reimbursement and which have not traveled or traveled partially only can claim to be reimbursed. The customer must prove that he has not or only partially used the ticket.

A partial use might be a only a part of the trip was traveled or only some of the travelers were traveling or a combination of both.

The non-use of a ticket may be proven by documents that were provided to the customer/traveler on a train or at a station. This proof can also be provided in electronic form by a carrier.

The customer must be able to make the claim via a retailer to the disributor who needs to forward the request to the involved carriers.

Functional Requirements of a Distributor

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 distributor 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.

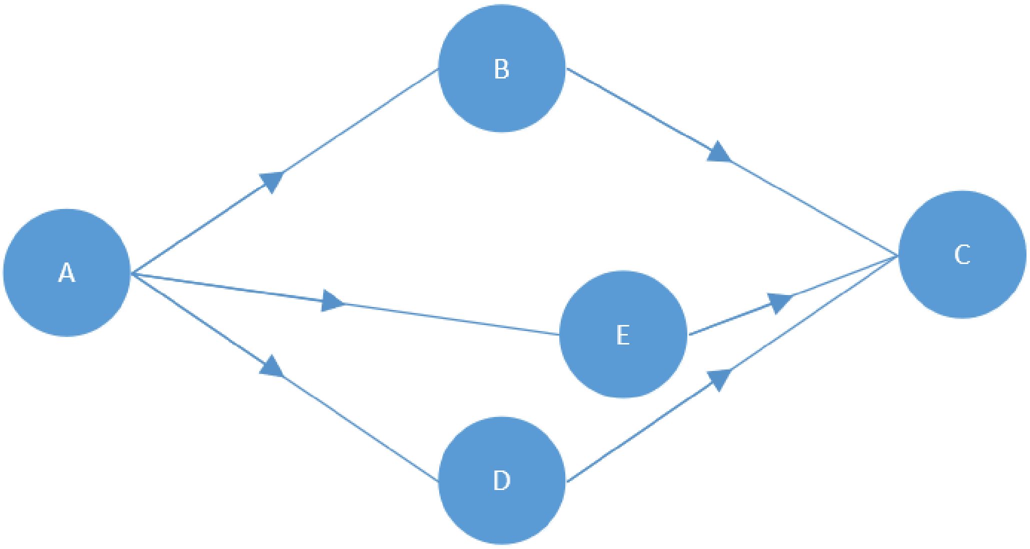
Route

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 might be indicated with an additional text (e.g. FR or GR) in the human readable form if it is not at a "real" station.

Are routes used as line routes or as bubble routes?



Line- vs. Bubble Route interpretation

`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 (bar codes) will be available for the fares as well:

- Zones
- Countries
- Geographical polygons

Route for dynamic fares:

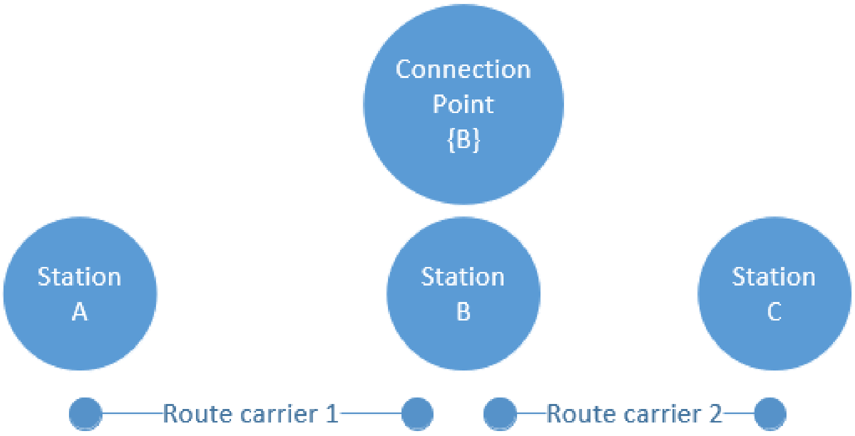
- Train bound ticket only
  - No route
  - List of train number, travel date and time, service brand (optional), from/to station
- Route based only
  - Route same as for non-yielded NRT
  - Optionally depending on service brand(s)
- Train bound (long distance) + route (regional)
  - Yield management 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/connection 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/connection point
  - Route description from the border point/connection point onwards

Connection Point

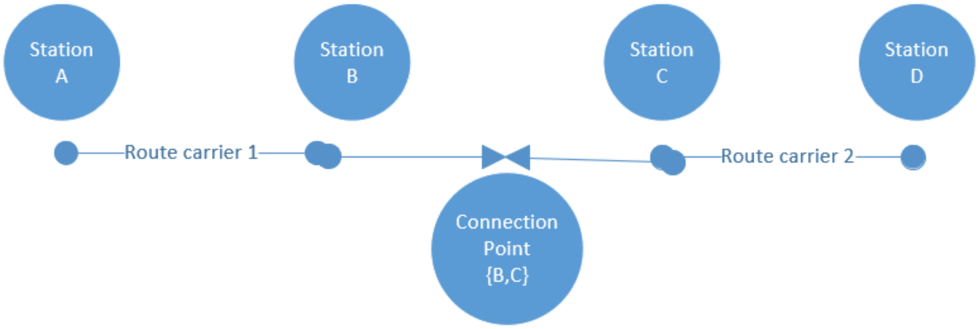
Regions (routes) of different carriers can be connected at defined connection points. The old concept of a central predefined list of points (as part of TAP-TSI) is replaced by the concept of "connection points" which can be defined by each carrier independently using the station codes.

ConnectionPoints provide information on the points where two fares of different provides can be glued together.

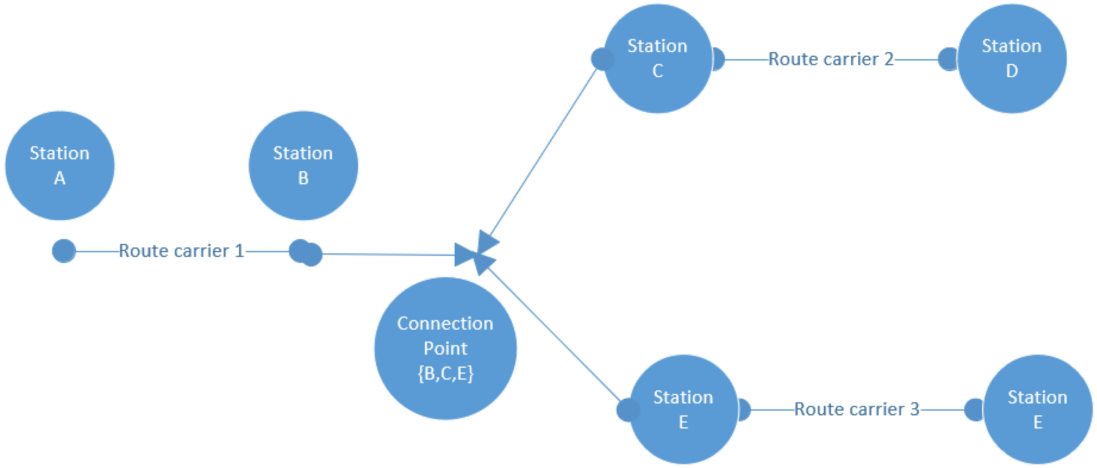
As in principle every station can become a connection point (e.g. all stops from Aachen to Brussels are connection points from DB to SNCB as the train is a DB train also inside of Belgium).



Connection point



Connection point



Connection point

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.

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.

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.

A marketing name is required 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)

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.

Requirements 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).

SalesAvailability 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.

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.

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 two 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

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
  - Start and end date of validity in local dates (e.g. for passes that cover more than one time zone)
  - Number of allowed trips or days
  - The validity might depend on the time of Boarding or De-Boarding of a train. E.g. a pass might be valid on the train if the the passenger boards the train during the validity of the pass.
- Examples:
  - Valid Monday – Friday if work day from 09:00 until 03:00 the following day
  - Valid Saturday – Sunday and public holidays from 00:00 until 03:00 the following day

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 traveling 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).

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

The passenger weight of each passenger type needs to be considered.

Requirements on validity for reductions

Reductions are price reductions 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 tripLeg, 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.

Requirements on prices

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, international 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.

Possible Price formats are:

- 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.

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.

Requirements on the after sales conditions

After sales conditions define fees to be taken in case of an aftersales 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 distributor 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.

Requirements on conditions on fulfillment

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

The fulfillment might be restricted depending on:

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

The Required personal data might depend on the fulfillment:

- Required personal data to be provided from the distributor to the carrier depending on type of fulfillment

- Depending on border crossing and train types (Belgium border crossing of high-speed trains requires personal data)

- Data might be required for ticket holders only or for all passengers

Fares should provide the involved TCOs that need to have access to the ticket data for control of fulfillments.

Requirements on dynamic fares and train linked tickets

Indication of dynamic fares available online

The distributor 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.

Indication of train links on the ticket

Tickets may be linked to the use of specific trains even in the case there is no seat 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:

- Date and departure time
- Service Brand Abbreviation (e.g. RJ, ICE) (can be retrieved from timetable data)
- Train number
- 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

Requirements on combining fares

Multiple models are defined for combining fares. The carrier defines in the fare data which model(s) the distributor 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 distributor 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 distributor 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 most be issued. Otherwise a combined contract can be issued.

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 distributor. However, the after sales conditions must respect 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 distributor has a simple unique view on after sales conditions. Optionally this might be restricted by a list of carriers and/or distributors 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 leads to 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 distributor

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

NON-FLEX:

- Non refundable
- Non exchangeable
- Minimum validity applies

PROMO:

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

Implementation Aspect

Relevant attributes:

```
-FareCombinationConstraintDef.combinationModels.model ==
  CLUSTERING -
    FareCombinationConstraintDef.combinationModels.combinableCarriers -
    FareCombinationConstraintDef.combinationModels.referencedCluster -
    FareCombinationConstraintDef.combinationModels.allowedDistributors;
```

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;
```

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.

Requirements on Trip Interruptions

Restrictions oon allowed trip interruptions must be indicated in a fare.

Restrictions on interruptions can specify the maximal number of interruptions and the maximal duration of interruptions.

The fare must provide information whether the interruption has to be confirmed by staff of by deactivation of the electronic ticket.

Requirements on Multi-Journeys Tickets

The restrictions on the use of a multi-journey ticket for individual trips must be described in the fare.

Restrictions can specify the number of das, trips or time units to be used.

The process to use an individual trip with a multi journey ticket must be indicated. These include separate fulfillments per individual trips or activation for a trip.

Requirements on Fare Exchange

In principle prices for customers are created by addition of prices per fare provider. Nevertheless, bi-multilateral agreements may include other regulations.

Conversion from Euro into a national currency (if necessary, vice versa) is subject to national distribution systems of the carrier/distributor concerned.

Architectural Requirements

Requirements on aligned processes end to end

The processes must be are 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 transportation, e.g. scooters or rail

Requirements on security

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

Legal Requirements

The flowing legal regulations provide requirements that affect the solution:

Rail PRR Regulation (EC) 1371/2007 on Rail Passengers’ Rights and Obligations

This regulation must be fulfilled.

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

- The passenger must be informed on the use of his data and on passing his data to the carrier and TCO
- The passenger must be informed which data are stored including data passed to the carrier and TCO
- The passenger has the right to ask to delete the data in case the data are not required to fulfil the contract of carriage
- The passenger 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 distributor for a local blacklist. He has therefore to obey the regulations when using the data.

  Two processing actions (automated profiling) are concerned:

  1. Collection and analysis of personal data on regular basis for trigger points: consent of passengers or legitimate interests of the rail carrier is needed
  2. Storage of information in the blacklists: legitimate interest of the rail carrier is needed
- General black list for use by multiple companies is allowed
  1. No access to the full list is provided
  2. No automatic checking in all cases
- Pre-cautions to be pursued by the railway undertaking
  1. Ensure right of access and objection
  2. Information preceding such processing and notification of inclusion into the blacklist
  3. Safeguards to prevent confusion
  4. 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

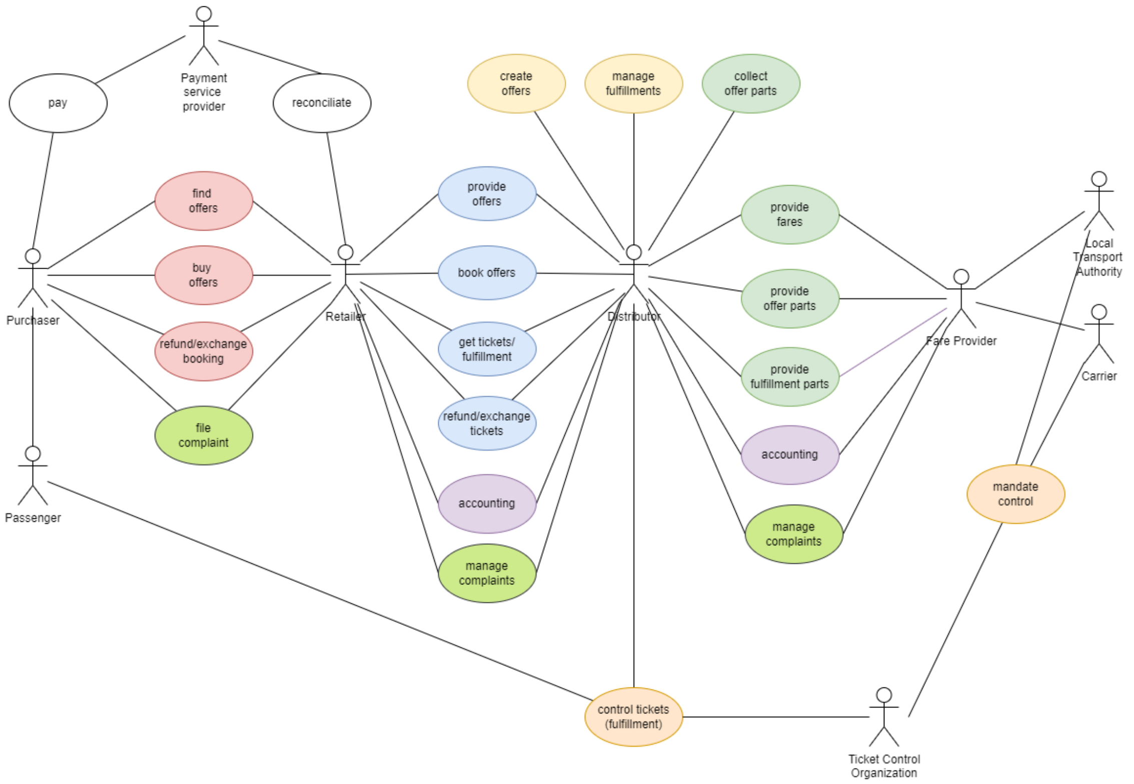
Actor Model and Business Use Cases

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 entity but merely a facet (e.g., "role") of some entity** that is relevant to the specification of its associated use cases. Thus, a single 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.



Actor Model

Actors in OSDM

Actor	Description	Motivation / Distinction to other roles
Distributor	The **distributor** manages the lifecycle of a product sold (the travel contract). He therefore needs to	

Actor	**Description**	Motivation / Distinction to other roles
	establish information exchange with the **retailer**, **carriers** and **TCOs** involved. The **distributor** makes products available to the **retailer**. The distributor could provide direct services to the **passenger** to modify the ticket status (e.g. activate / check in on a ticket). The distributor combines fares defined by the carriers according to their rules.The distributor creates the ticket fulfillment data (e.g. pdf, pkpass, …). The distributor is introduced to separate the role of just selling tickets along a route (retailer) from the role of creating the ticket content and providing it to retailers for sale.	
Carrier	The **carrier** provides the transport service to the **traveller himself or via a substitute carrier**. The travel contract provided to the customer	

Actor	**Description**	Motivation / Distinction to other roles
	establishes a contract between the traveller and each carrier participating in the service. Carriers include Railway undertaking, Bus companies, Maritime companies. Unless some intermeta fare provider organizes the fare the carrier is the owner of the fare.	
Customer	The **customer** purchases a travel contract for one or more traveller. *Note*: **The customer** is entitled to receive refund payments.	
Retailer	The **retailer** is the company selling the ticket provided and managed by the distributor to the customer.	
Fare Provider	The **fare provider** manages fares on behalf of a carrier or a local transport authority.	
Local Transport Authority	The **local transport authority** organizes the local traffic within an area a behalf of the government or is itself a governmental organization. It	

Actor	**Description**	Motivation / Distinction to other roles
	defines a fare structure for the local transport which all carriers included must apply.	
Passenger	The **passenger** is the 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.
Ticket Controller	Person (train staff) or machine (e.g. gates) responsible for controling the ticket. The ticket controler is always part of a **ticket controlling organization** (**TCO**) acting on behalf ot the carrier.	

Common Business Capabilities

Powerful Fare Combination

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

Simple Sales

It must be easily possible to sell 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 Retailers

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 if possible. For the Italian market, non-homogenous offers need to be supported. Some trains of the trip might not support all service classes.

Searching for non-trip based offers is supported by the same service. Search criteria can be tags, regions, geo-coordiantes.

Offer combination

Offers can provide tags to indicate that some products from a provider can be sold only when in conjunction with product(s) from another provider using the same tag. At least one, not all, combinationTags must be in common to allow combination. No combinationTags indicate that there are no combination constraints on the product. The comination might be mandatory or optional.

The combination tags must be unique across all partners involved on a given trip.

If offers have no combination restrictions they can be combined freely. However if business rule require, it must be to express combination constraints to secure the tariff validity.

Pre-book Offers

If a customer puts an offer into a basket on a retail channel, it most 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 "pre-booked". If the pre-booked 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".

Multiple formats and media are supported including pdf, pkpass. Parts to be included in a fulfillment (visual security elements, separate bar code) can be used as well in case of fares.

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.

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.

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 optional. Product information is part of the offer or booking and is included there by default.

Retrieve Stored Personal Data

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

There is no specific service to retrieve the stored personal data. The booking can be retrieved to get the passenger references and then the passengers can be retireved. This will provide all stored personal data.

Manage complaints

Complaints can be created on behalf of the passenger by Retailers. The distributor can request additional documents to proof the complaint. The passenger can add documents to prove the claim and change his data (e.g. bank account). The Retailer is informed on changes on the claim.

Provide Masterdata

Reduction Card Definitions

The Definitions of reduction cards are provided as Master Data. The reduction card definitions can be used by the Retailers for selection lists in the sales process or in customer accounts.

Business Capabilities for Fare Allocation

Combine Fares

The distributor 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 distributor 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 distributors.

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.

Fulfillment

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

Reservation

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

Option/Step 1: Using old messages for reservation

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

Option/Step 2: Using REST services for all services

1. offer (REST service)
2. checkPreferences (REST service) / graphical place display (REST service)
3. pre-booking NRT / reservation (REST service)
4. confirm booking of reservation / NRT (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.

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.

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 distributor 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 distributor (unique id within the context of the distributor 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

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.

Accounting data flow:

- In case of NRT fares used from a bulk data exchange:

The distributor 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 distributor:

The carrier is responsible for the accounting. The data structure for IRT is used. The distributor 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 of places is needed.

Passenger information

Passenger names are supported with an element for fits names and an element for last names. As an additional option it is possibe to provide a first and a last familiy name additionally. This is used to cover lewgal requirements in some spanish speaking countries where both family names might be required separately.

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

There is no specific service to retrieve stored personal data. The booking can be retrieved to get the passenger references and then the passengers can be retireved. This will provide all stored personal data.

Manage complaints

Complaints can be created on behalf of the passenger by the reteiler. The distributor and the carrier can request additional documents to proof the complaint. The passenger can add documents to prove the claim and change his data (e.g. bank account). The distributor can decide on a claim himself in case the carrier has not kept the legal time line.

Manage reimbourcements

Reimbourcements can be requested by the retailer to the distributor in case the fare allows the reimbourcement of unused tickets. The distributor can forward the request to fare providers.

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 TAP TSI retail station code list (MERITS).
- 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, e.g.: `passenger.gender AND (passenger.email OR passenger.phoneNumber)`

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 fulfillment. (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 distributor
- 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 distributor 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,..)

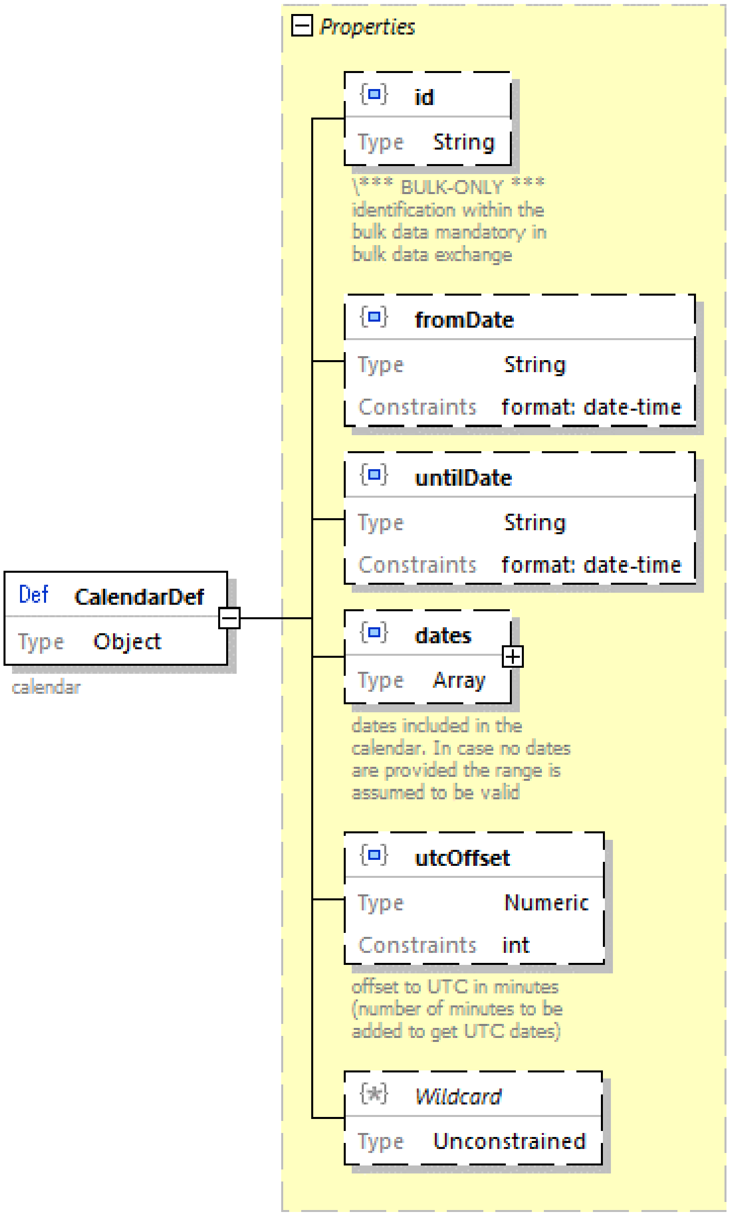
In case multiple rules apply to the same after sales transaction the rule with the closest time in the future must be applied.

Data Constraints on AfterSaleRule

Code	Description
`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 € 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 application 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.



Calendar

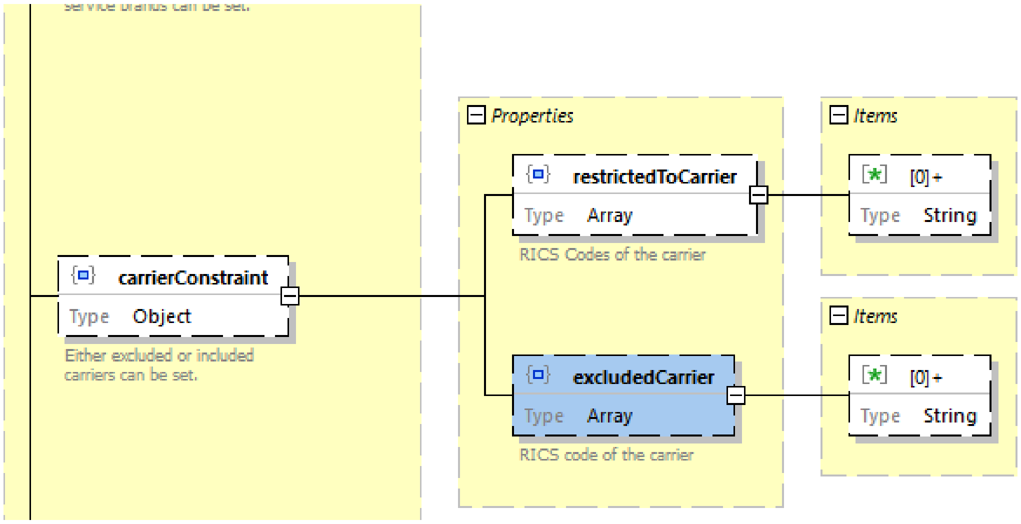
Data Constraints on Calendar

Code	Description
`fromDate`, `untilDate`	fromDate and untilDate must be provided and `fromDate <= untilDate`
`dates`	`fromDate <= date <= untilDate`

CarrierConstraint

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. In case of admissions (train linked or non-train linked) the included carriers specify the responsible carriers to be listed for the customer on an fulfillment.

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



Carrier Constraint Type

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.

Data Constraints on CarrierConstraint

Code	Description
`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.

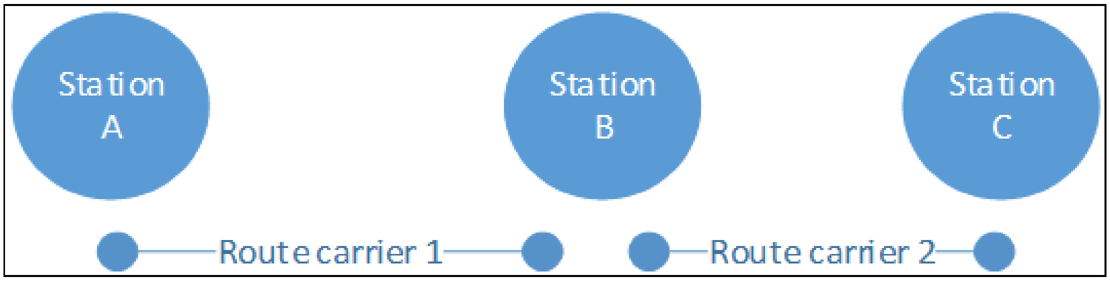
In case the combination is not at a real station an indication is needed to define the allowed combinations. This could be done by listing the next stations of other carriers which would allow a combination. Combinations would be allowed if the combination points of two routes share two common stations.

This would also work with multiple stations.

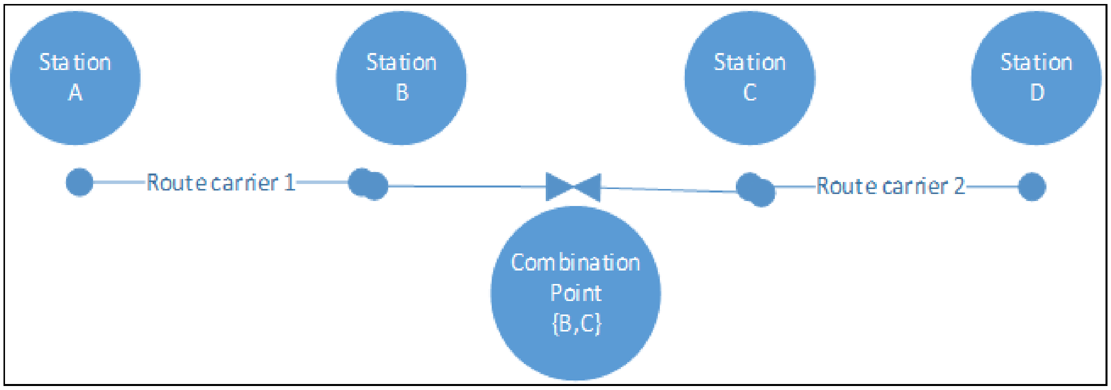
Connection points will include a border point code to support existing implementations where the border point code is compared with the timetable data. As in principle every station can become a connection point (e.g. all stops from Aachen to Brussels are connection points from DB to SNCB) implementations 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 connections 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.

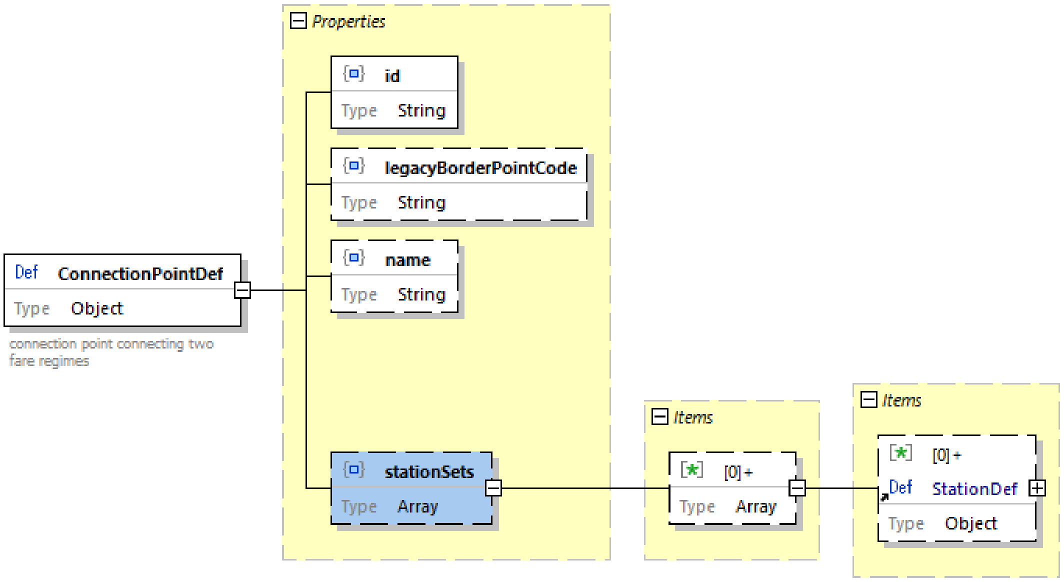


Fare Connection Point - Simple Case



Fare Connection Point - Complex Case

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



Fare Connection Point

Data Constraints on ConnectionPoint

Code	Description
`stationSets`	At least one set with one station must be provided in case the fare border is a real station. Two station sets must be provided in case the fare border is between two real stations.
`legacyBorderPointCode`	The legacy border point code must be provided for the time being. New implementations should not use the border point code.

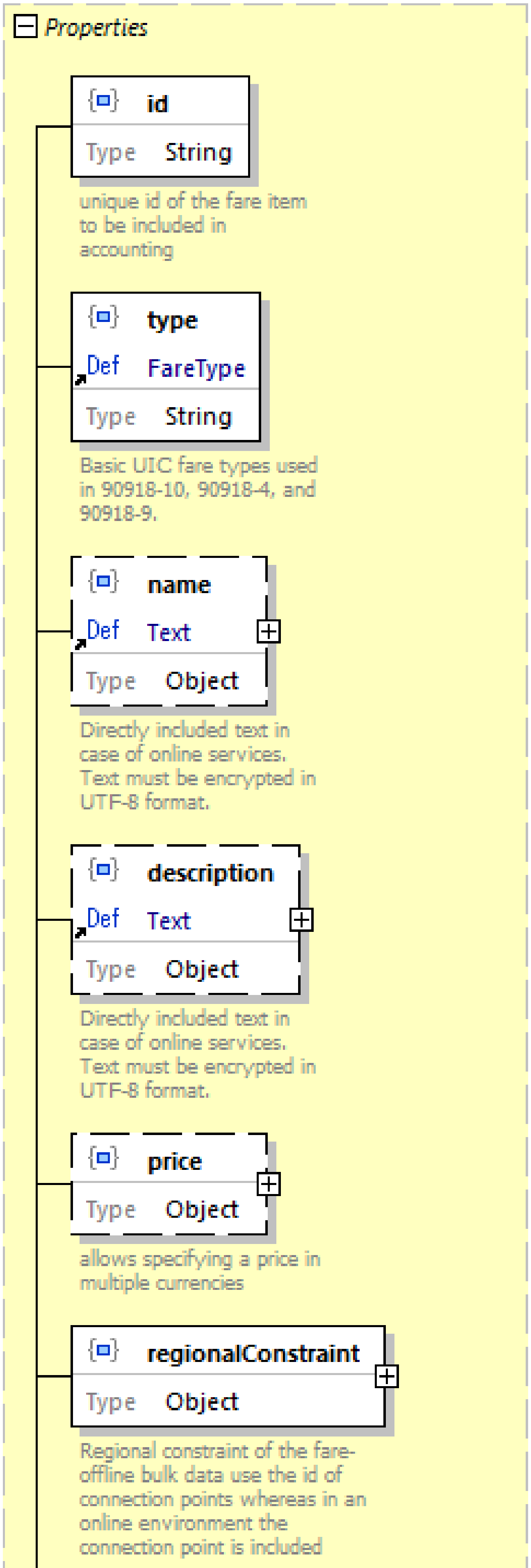
Fare

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

Data elements	Description
`fareType`	**NRT**, **IRT**, **Ancillaries** , **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 € must be provided if not otherwise agreed bilaterally.
`regionalConstraint`	Definition of the regional validity of the fare and the geographical combination rules (connection points).
`serviceC`	Restrictions of the service allowed to be used.

Data elements	Description
`onstraint`	
`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 distributor is responsible for the after sales rules this is almost empty.
`combinationConstraint`	Rules on the model of combination of this fare with fares of other carriers.
`fulfillmentConstraint`	Restrictions and requirements on the fulfillment and security to be applied by the distributor.
`reductionConstraint`	Rules on reduction cards necessary to apply the fare.
`reservationParameter`	Information on parameters for reservation via the *IRS 90918-1* interface and reservation options.
`regulatoryConditions`	Legal regimes to be applied to the fate (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 sates dates for the fare.
`travelValidityConstraint`	Rules on the validity for travel of this fare.
`luggageC`	Rules on the luggage to be taken with this fare.

Data elements	Description
`onstraint`	
`legacyConversion`	Defines whether this fare is allowed to be converted to the old 108.1 data structure and used according to the old rules (`YES`, `NO`, `ONLY` (this fare is provided for conversion only)).



Properties

id
Type String

unique id of the fare item to be included in accounting

type
Def FareType
Type String

Basic UIC fare types used in 90918-10, 90918-4, and 90918-9.

name
Def Text
Type Object

Directly included text in case of online services. Text must be encrypted in UTF-8 format.

description
Def Text
Type Object

Directly included text in case of online services. Text must be encrypted in UTF-8 format.

price
Type Object

allows specifying a price in multiple currencies

regionalConstraint
Type Object

Regional constraint of the fare-offline bulk data use the id of connection points whereas in an online environment the connection point is included

Fare (Online)

Data Constraints on Fare

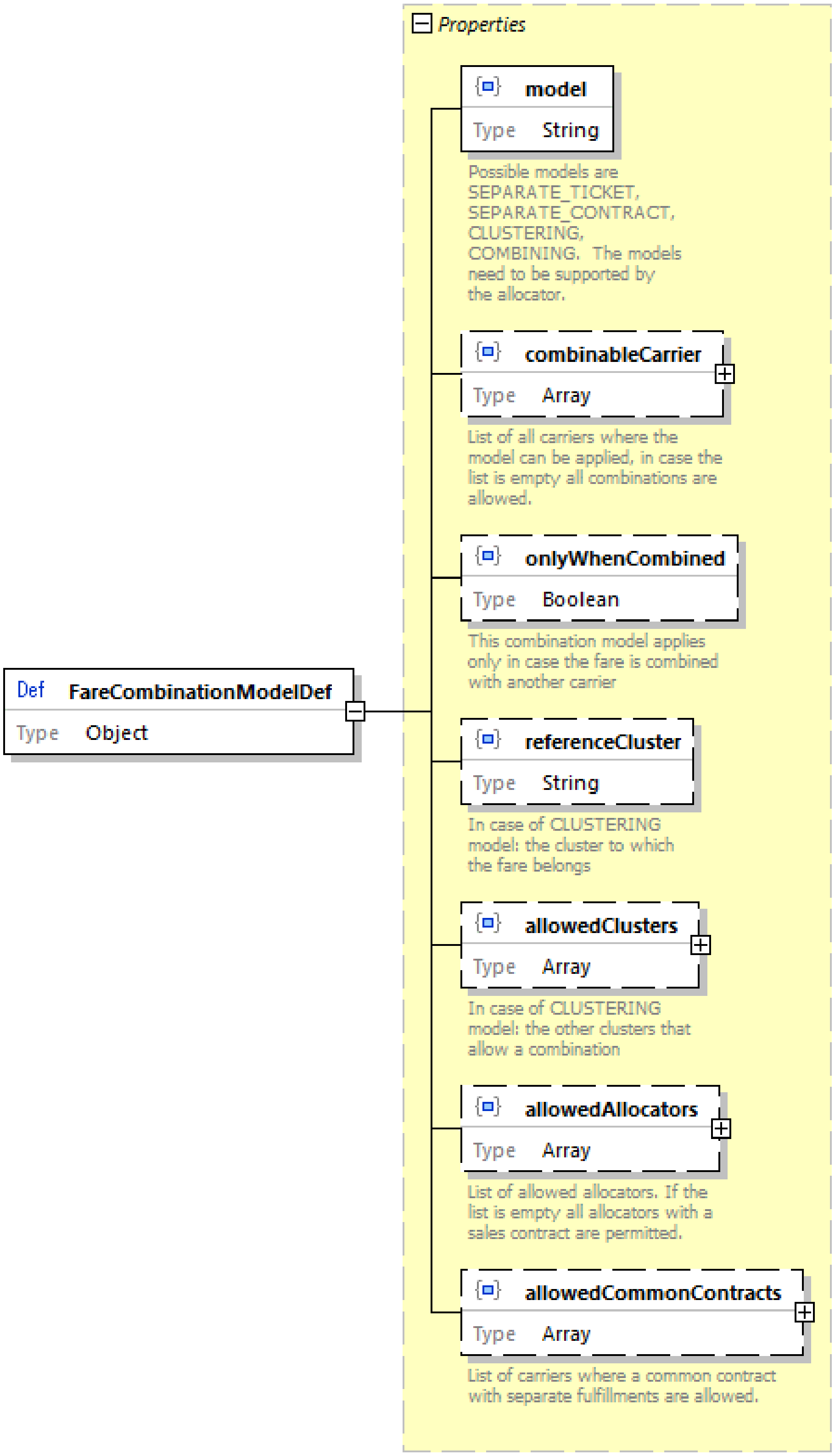
Code	Description
`price`	A price must be provided for all offline fares including those where the price is zero.
`legacyAccountingIdentifier`	In case *IRS 30301* in the current version is used to accounting these data must be provided for offline fares
`serviceClass`	Must be provided for offline fares
`combinationConstraint`	Must be provided for offline fares
`travelValidityConstraint`	Must be provided for offline fares
`salesAvailabilityConstraint`	Must be provided for offline fares

FareCombinationConstraint

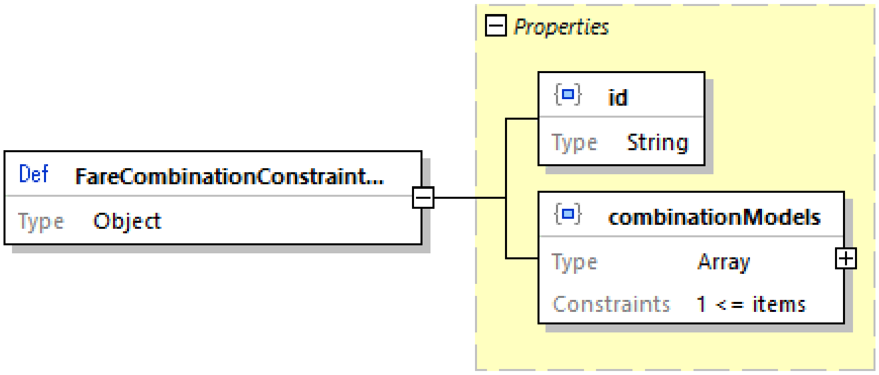
The fare combination constraint defines the rules of combining fares from different carriers. It provides a list of combination models the distributor can choose of.

Content	Description
`model`	Code of the combination model applied
`combinableCarriers`	List of carriers that can be combined with this fare. If empty, there is no restriction in combining different carriers. 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
`allowedDistributors`	List of distributors which can combine this fare. If empty, there is no restriction in combining different carriers. Carriers are listed by their RICS company codes. This is not present in the online data.
`allowedCommonCo`	List of Carriers with which the distributor can combine for a common contract. If empty, there is no restriction in indicating common contracts to the passenger except for the `SEPARATE_CONTRACT` model. Carriers are listed by their RICS company

Content	Description
`ntracts`	codes.


Properties
model
Type String
Possible models are SEPARATE_TICKET, SEPARATE_CONTRACT, CLUSTERING, COMBINING. The models need to be supported by the allocator.
combinableCarrier
Type Array
List of all carriers where the model can be applied, in case the list is empty all combinations are allowed.
onlyWhenCombined
Type Boolean
This combination model applies only in case the fare is combined with another carrier
Def FareCombinationModelDef
Type Object
referenceCluster
Type String
In case of CLUSTERING model: the cluster to which the fare belongs
allowedClusters
Type Array
In case of CLUSTERING model: the other clusters that allow a combination
allowedAllocators
Type Array
List of allowed allocators. If the list is empty all allocators with a sales contract are permitted.
allowedCommonContracts
Type Array
List of carriers where a common contract with separate fulfillments are allowed.

Fare Combination Model



Fare Combination Constraint

Combination Model

SEPARATE_CONTRACT Model

This SEPARATE_CONTRACT model 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 distributor must ensure that it is clear for the customer that no common contract was established.

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 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 distributor. 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 distributor has a simple unique view on after sales conditions.

The basic parameters defining the price must be obeyed individually within separately on the combined fare/offer:

- route description / train link
- class of service

- passenger types

COMBINING 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 reflect the conditions of all included carriers.

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

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.

COMBINING Model Example

- 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/Exchangeable after the departure or last day of validity
FULL_FLEX	Refundable/Exchangeable before the departure or last day of validity
SEMI_FLEX	Refundable/Exchangeable with fee depending on conditions of the distributor. 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 his is not available with one of the higher clusters.

Clustering Model Example

- 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 distributor. E.g.:

- FULL_FLEX (Carrier 1 BUSINESS + Carrier 2 BUSINESS)

Data Constraints on FareCombinationConstraint

Code	Description
combinationModel	At least one model must be provided

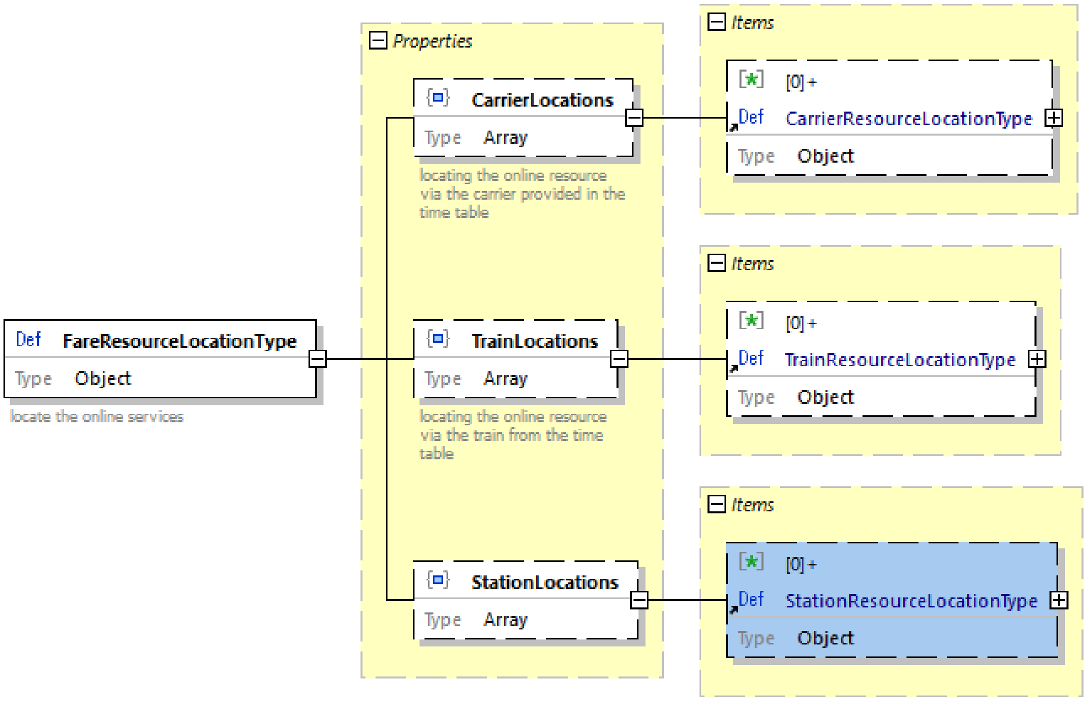
FareResourceLocation

Fare resource location provides data on where to find online services for fares. The fare location provides three options:

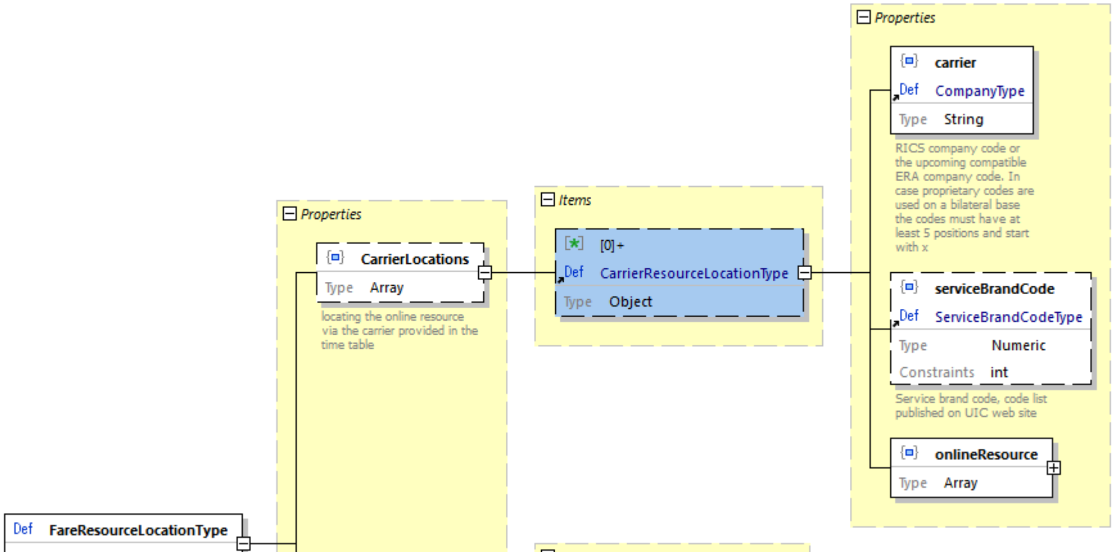
- Link a resource to a carrier – the carrier must be known from the timetable
- Link a resource to the train – the data must be updated in case of new trains
- Link a resource to stations:
  - The link can be made for stations and for connection points
  - The link is valid if start and end station (or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

The online link provides information on:

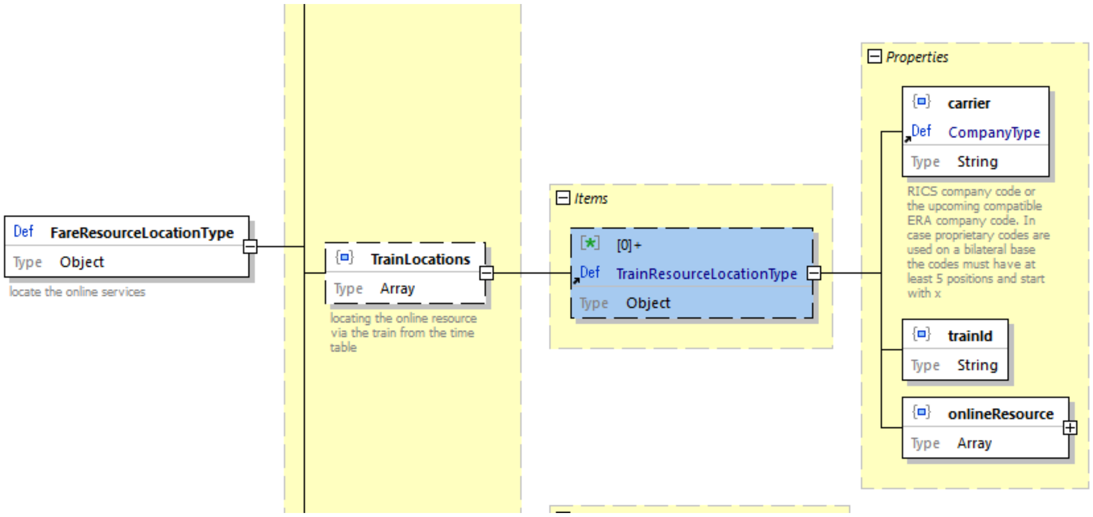
- The type of resource either for a whole train of an area. In case of a train the request must be for the train route between stations (e.g. IRT), whereas for areas there might be multiple splits in-between a train run (e.g. NRT).


Def FareResourceLocationType
Type Object
locate the online services
Properties
CarrierLocations
Type Array
locating the online resource via the carrier provided in the time table
TrainLocations
Type Array
locating the online resource via the train from the time table
StationLocations
Type Array
Items
[0] +
Def CarrierResourceLocationType
Type Object
Items
[0] +
Def TrainResourceLocationType
Type Object
Items
[0] +
Def StationResourceLocationType
Type Object

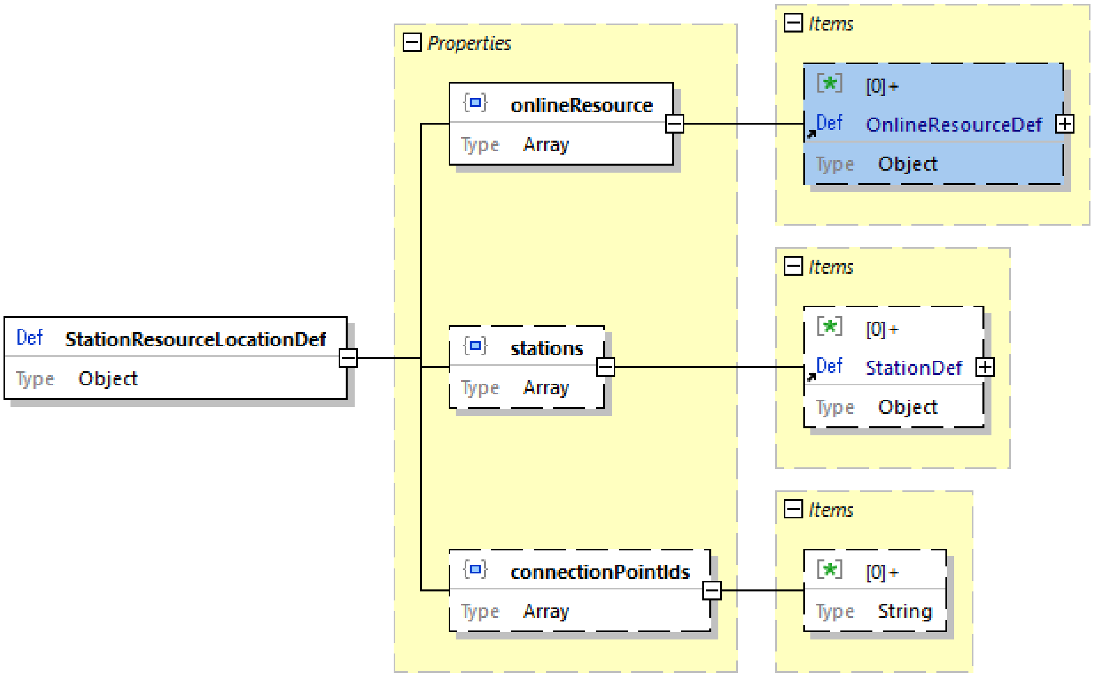
Fare Resource Location Type


Def FareResourceLocationType
Properties
CarrierLocations
Type Array
locating the online resource via the carrier provided in the time table
Items
[0] +
Def CarrierResourceLocationType
Type Object
Properties
carrier
Def CompanyType
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
serviceBrandCode
Def ServiceBrandCodeType
Type Numeric
Constraints int
Service brand code, code list published on UIC web site
onlineResource
Type Array

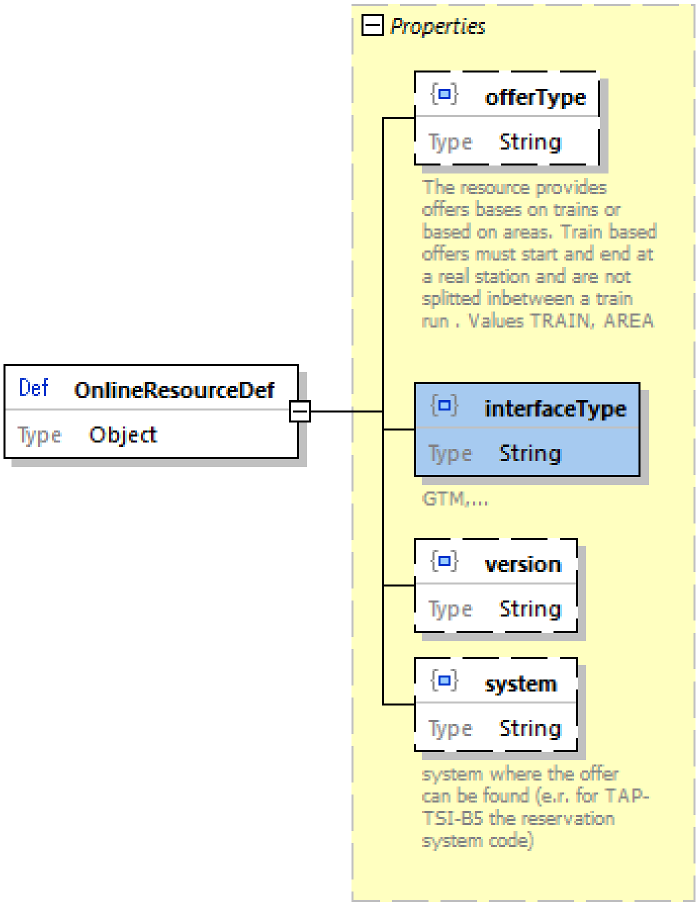
Fare Resource Location Type - Carrier Link


Def FareResourceLocationType
Type Object
locate the online services
TrainLocations
Type Array
locating the online resource via the train from the time table
Items
[0]+
Def TrainResourceLocationType
Type Object
Properties
carrier
Def CompanyType
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
trainId
Type String
onlineResource
Type Array

Fare Resource Location Type - Train Link



Fare Resource Location Type - Station Link


Properties
offerType
Type String
The resource provides offers bases on trains or based on areas. Train based offers must start and end at a real station and are not splitted inbetween a train run . Values TRAIN, AREA
Def OnlineResourceDef
Type Object
interfaceType
Type String
GTM,...
version
Type String
system
Type String
system where the offer can be found (e.r. for TAP-TSI-B5 the reservation system code)

Fare Resource Location Type - Online Resource

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.

Data constraints on FareResourceLocation

Code	Description
`System`	For reservation interface 90810-1 the reservation system code is used.

FareReferenceStationSet

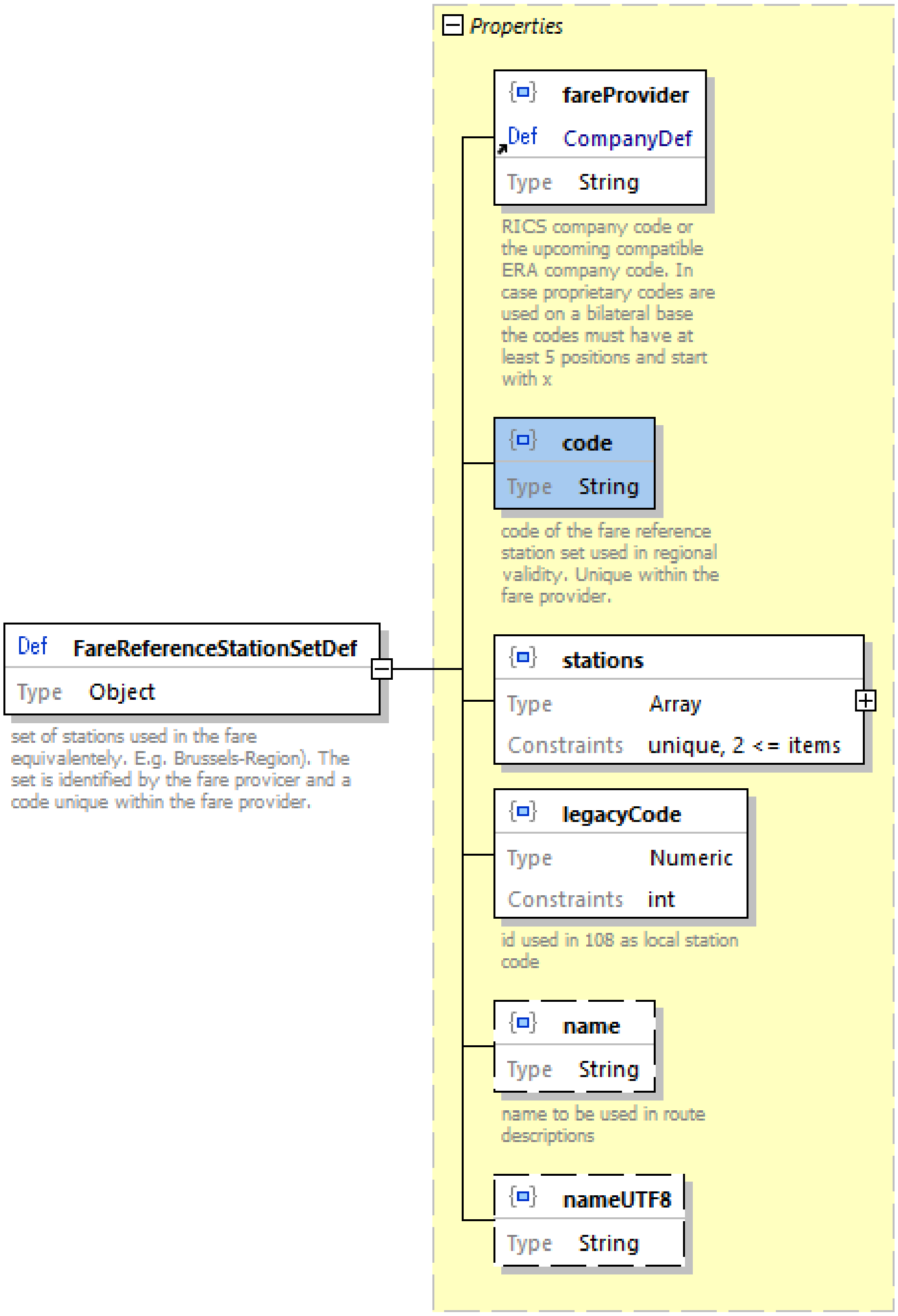
The fare reference station set defines a set of stations where the fare is valid for all included stations. This set can be used in the regionalValidity description.

The corresponding bar code ab ticket control data will only contain the code of the station set, but the distributor 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.


Properties
fareProvider
Def CompanyDef
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
code
Type String
code of the fare reference station set used in regional validity. Unique within the fare provider.
Def FareReferenceStationSetDef
Type Object
set of stations used in the fare equivalentely. E.g. Brussels-Region). The set is identified by the fare provicer and a code unique within the fare provider.
stations
Type Array
Constraints unique, 2 <= items
legacyCode
Type Numeric
Constraints int
id used in 108 as local station code
name
Type String
name to be used in route descriptions
nameUTF8
Type String

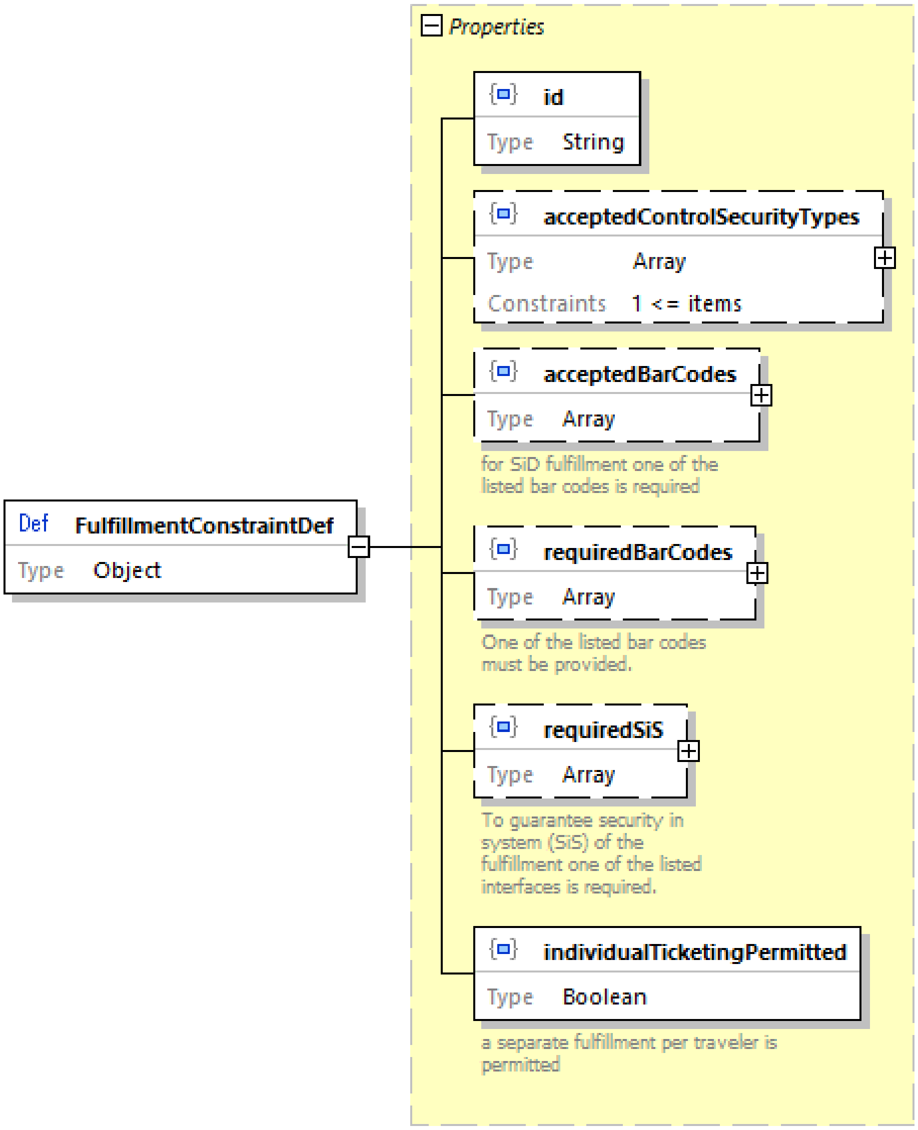
Fare Reference Station Set

Data Constraints on FareReferenceStationSet

Code	Description
`legacy Code`	A legacyCode must be provided for the time being. New implementations should not rely on that code.
`name`	The name should not include "/"."*".

FulfillmentConstraint

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


Properties
Def FulfillmentConstraintDef
Type Object
id
Type String
acceptedControlSecurityTypes
Type Array
Constraints 1 <= items
acceptedBarCodes
Type Array
for SiD fulfillment one of the listed bar codes is required
requiredBarCodes
Type Array
One of the listed bar codes must be provided.
requiredSiS
Type Array
To guarantee security in system (SiS) of the fulfillment one of the listed interfaces is required.
individualTicketingPermitted
Type Boolean
a separate fulfillment per traveler is permitted

Fulfillment Constraint

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`:

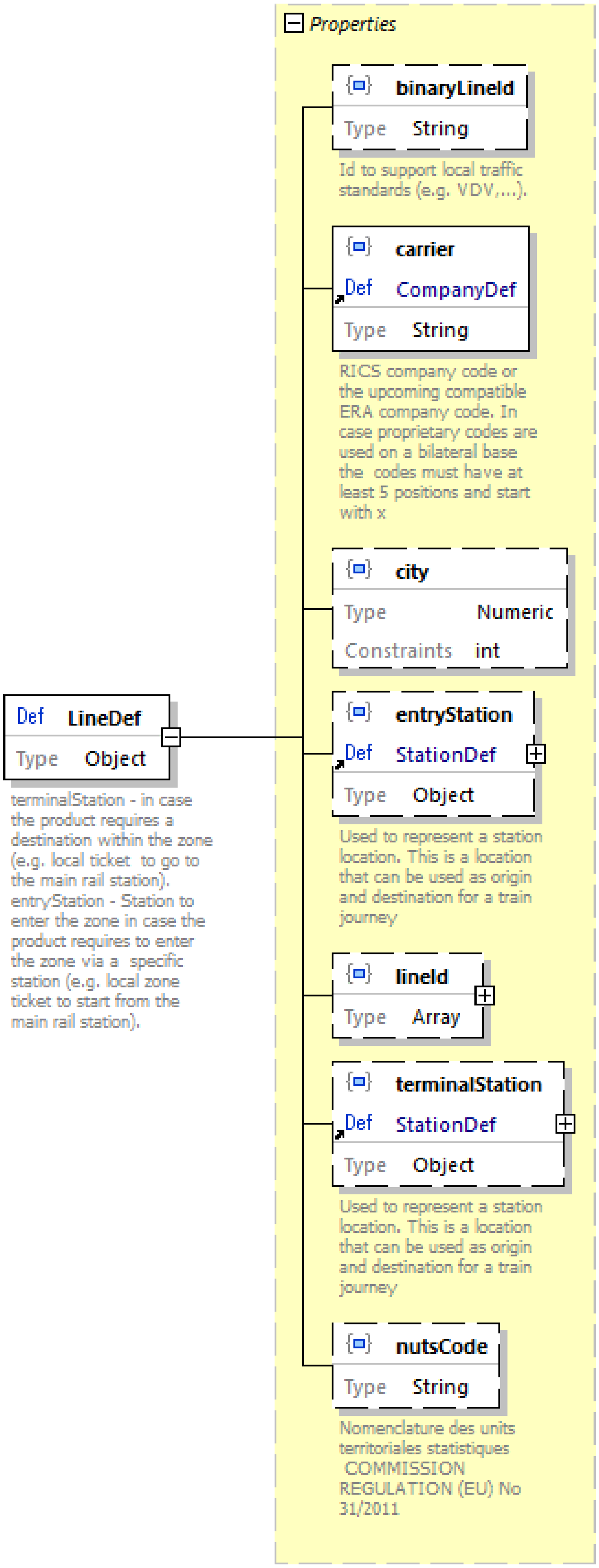
- for bar codes: `BarcodeType`
- for fulfillment: `ControlSecurityType`

Data Constraints on FulfillmentConstraint

Code	Description
`acceptedFulfillmentType`	At least one accepted fulfillment 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.



terminalStation - in case the product requires a destination within the zone (e.g. local ticket to go to the main rail station). entryStation - Station to enter the zone in case the product requires to enter the zone via a specific station (e.g. local zone ticket to start from the main rail station).

binaryLineId: Id to support local traffic standards (e.g. VDV,...).

carrier: RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

entryStation: Used to represent a station location. This is a location that can be used as origin and destination for a train journey

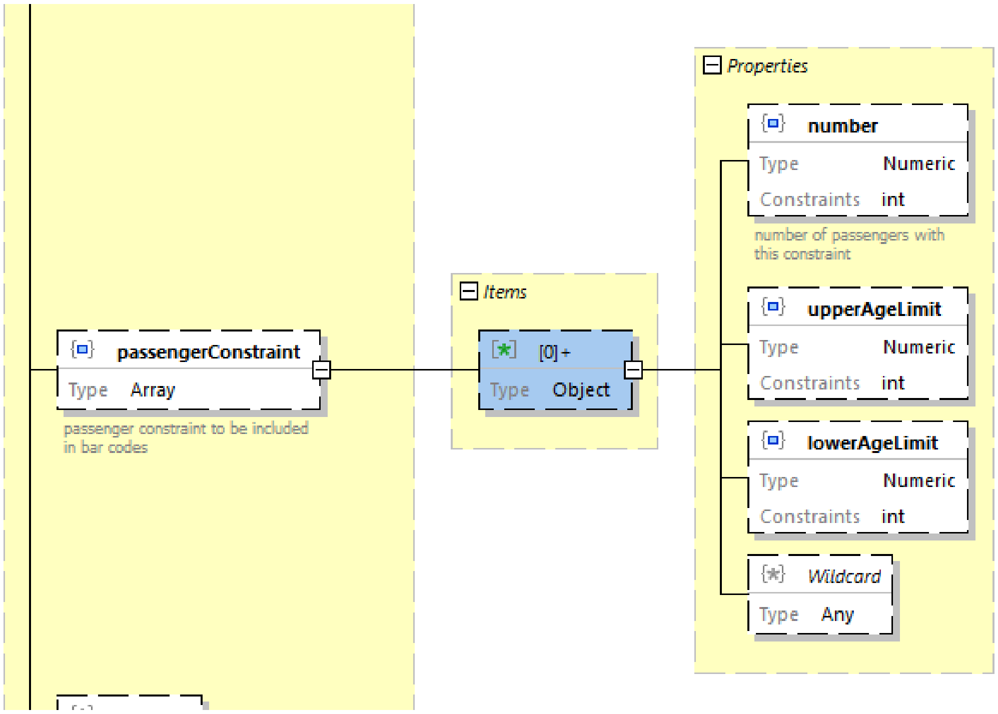
terminalStation: Used to represent a station location. This is a location that can be used as origin and destination for a train journey

nutsCode: Nomenclature des units territoriales statistiques COMMISSION REGULATION (EU) No 31/2011

Line

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 bar codes and control registries.



Passenger Constraint

Data Constraints on PassengerConstraint

Code	Description
`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*)

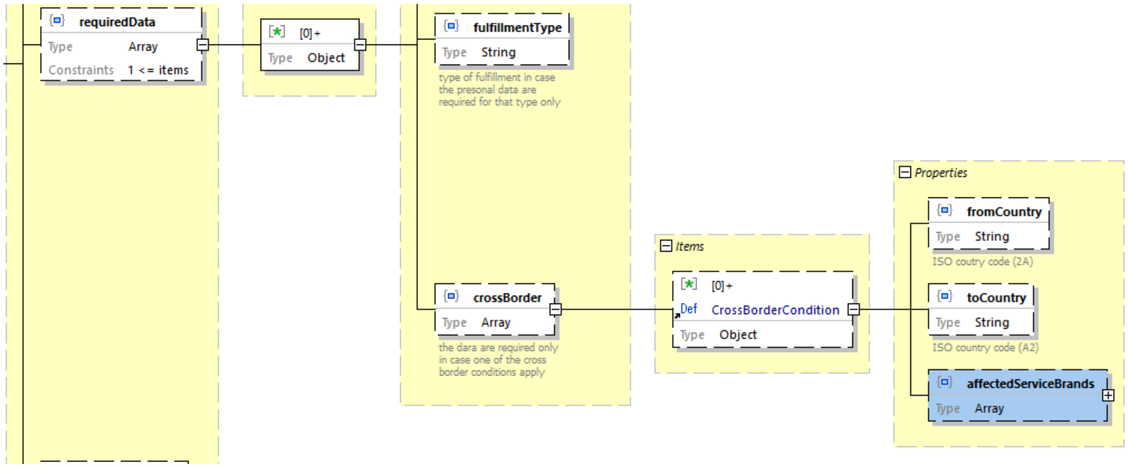
The requirement for personal data might depend on the type of fulfillment or on specific border crossings.

Code	Description
`acceptedReason`	Accepted reason to change personal data after booking confirmation. See code list: Personal data change reasons

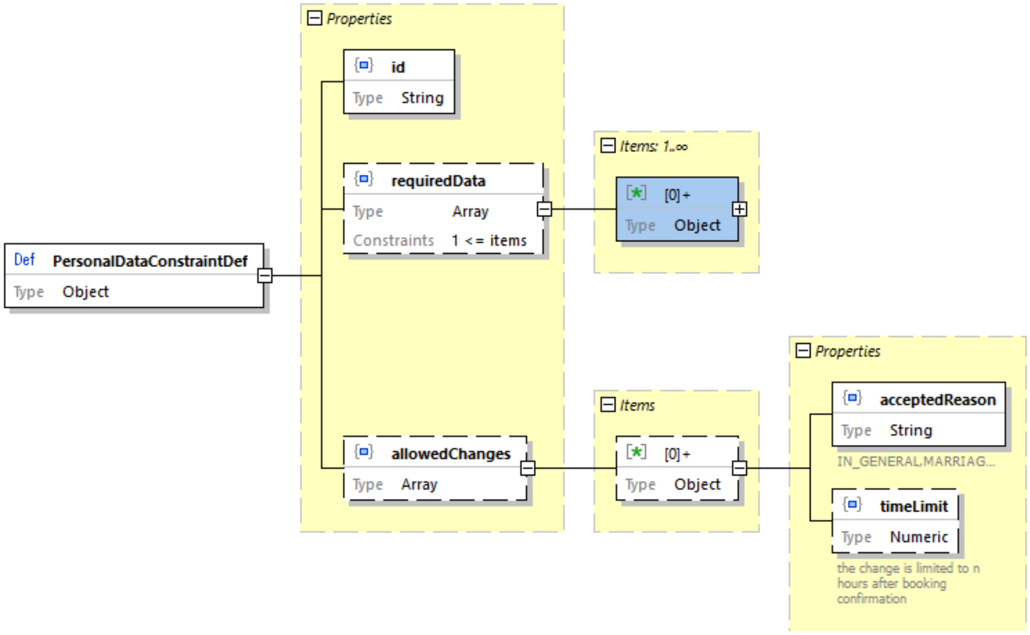
Code	Description
`transfer`	The way the personal data are transferred. See code list: Personal data transfer types
`ticketHolderOnly`	Personal data are required for the ticket holder only
`dataItem`	Code of the data item required. Consists of `languageCode`, `overruleCode` (see below) and personal data items.

Overrule Code

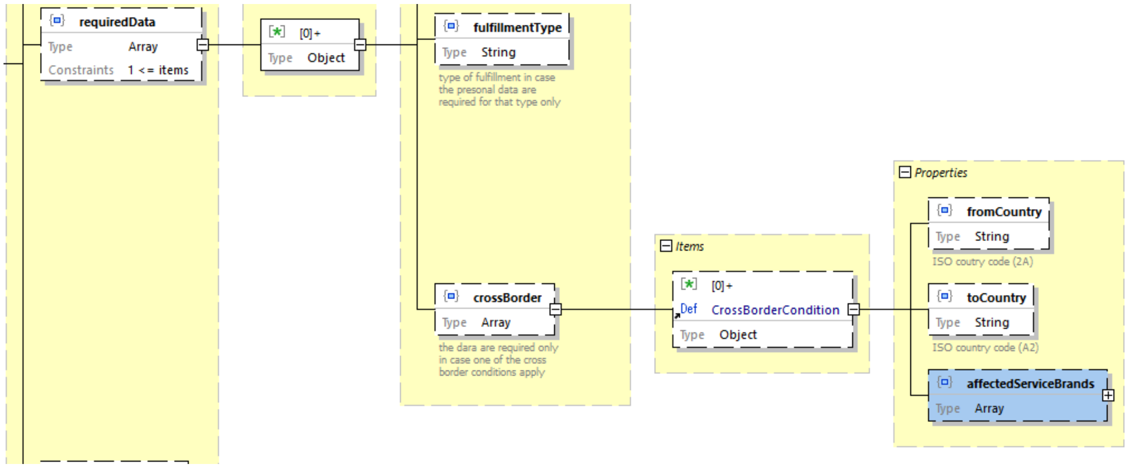
Code	Description
`STRIKE`	Refund due to strike
`SALES_STAFF_ERROR`	Refund due to an error made by the sales staff
`PAYMENT_FAILURE`	Refund as the payment failed



Personal Data



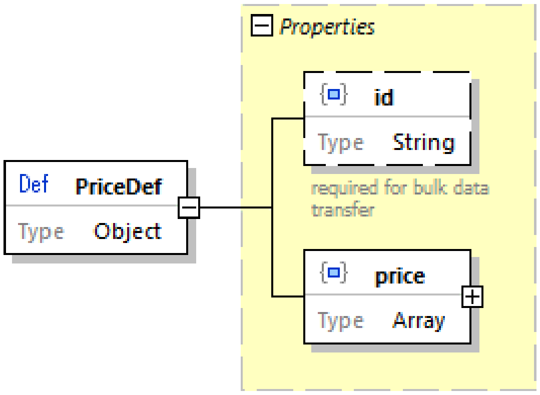
Personal Data - Allowed Changes



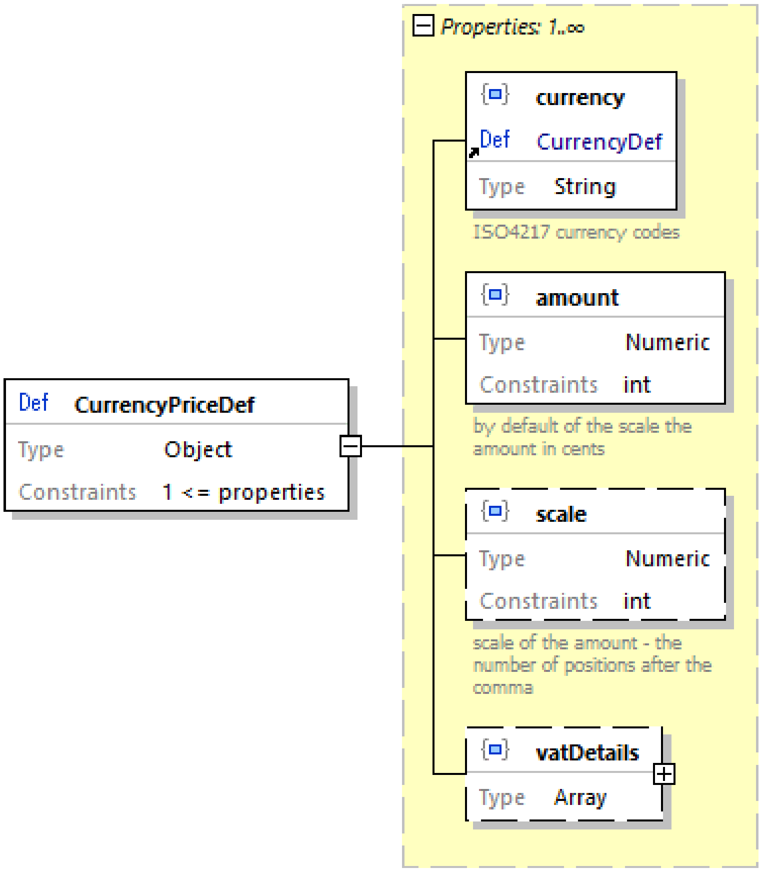
Personal Date - Cross Border Conditions

Price

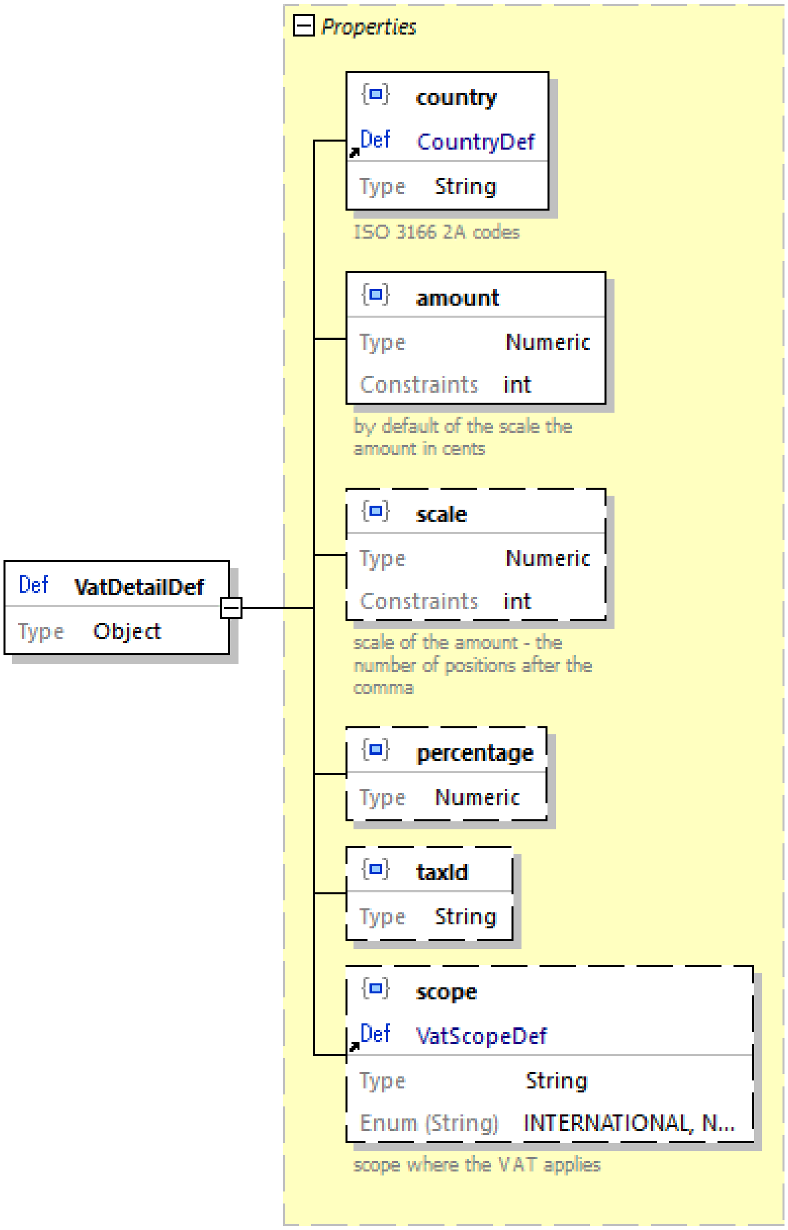
The price data structure provides the price or a fee including the VAT details optionally in different currencies.



Price



Currency Price



VAT Detail

Scope: see code list TaxScope

Data Constraints on Price

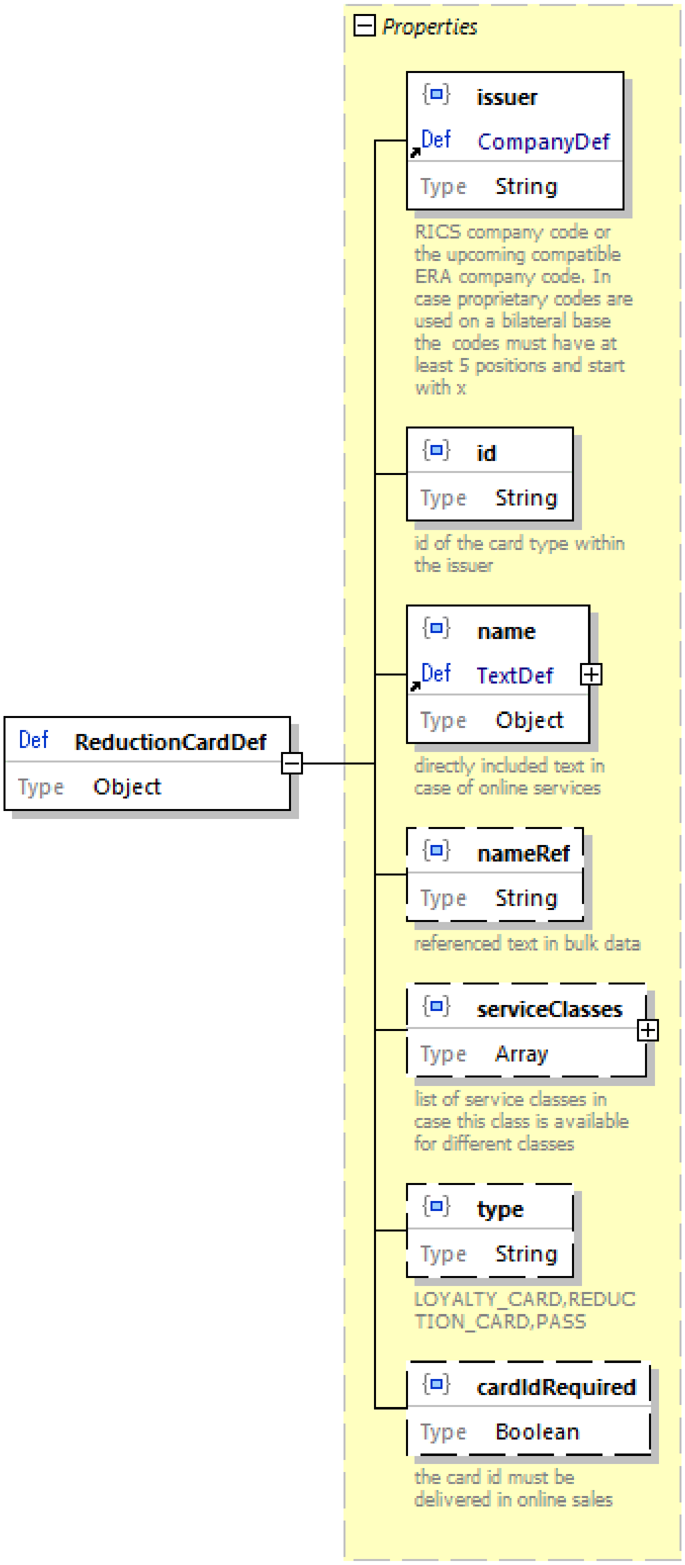
Code	Description
amount	Amount >= sum of VAT-amounts

ReductionCard

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

List of Carrier Cards

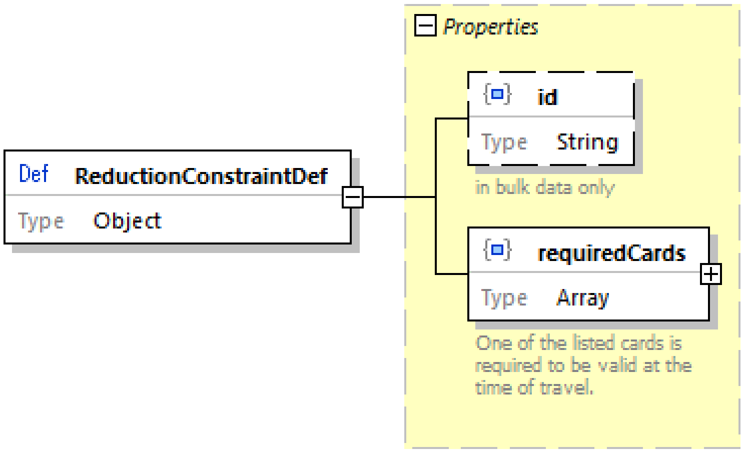
Code	Description
id	Unique id of the card. The id must start with the RICS code of the carrier
name	Name and short name of the card. The name should be used for the card selection by the customer, the short name should be used for bar codes.Usually the card name is not translated, but the card name might 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 pre-booking request to validate the card. This card cannot be used without the online services for booking


Properties
issuer
Def CompanyDef
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
id
Type String
id of the card type within the issuer
name
Def TextDef
Type Object
directly included text in case of online services
nameRef
Type String
referenced text in bulk data
serviceClasses
Type Array
list of service classes in case this class is available for different classes
type
Type String
LOYALTY_CARD,REDUCTION_CARD,PASS
cardIdRequired
Type Boolean
the card id must be delivered in online sales
Def ReductionCardDef
Type Object

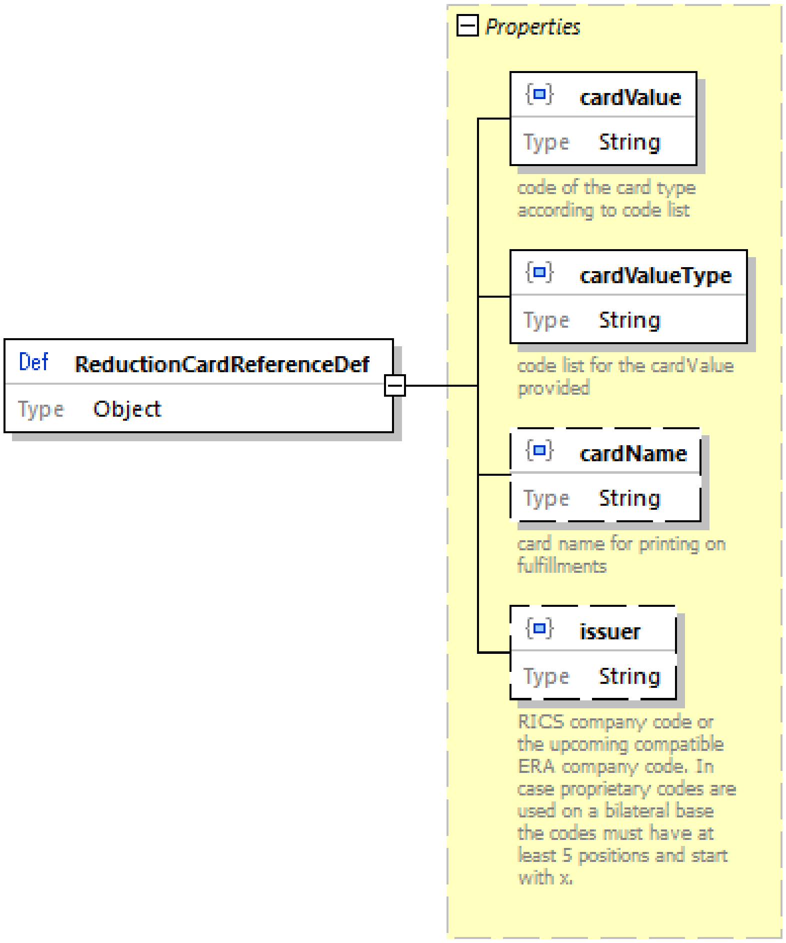
Reduction Card

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".


Properties
Def ReductionConstraintDef
Type Object
id
Type String
in bulk data only
requiredCards
Type Array
One of the listed cards is required to be valid at the time of travel.

Reduction Constraint


Properties
cardValue
Type String
code of the card type according to code list
cardValueType
Type String
code list for the cardValue provided
Def ReductionCardReferenceDef
Type Object
cardName
Type String
card name for printing on fulfillments
issuer
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x.

Reduction Card Reference

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 used 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 ConnectionPoint)
`exitConnectionPoint`	Defines the connection point for connecting this fare at the start of regional validity (see ConnectionPoint)
`regionalValidity`	Definition of the regional validity as defined in *IRS 90918-4*. It 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}
- *Result*: `X*Y/Z*A*B*C/D*E`

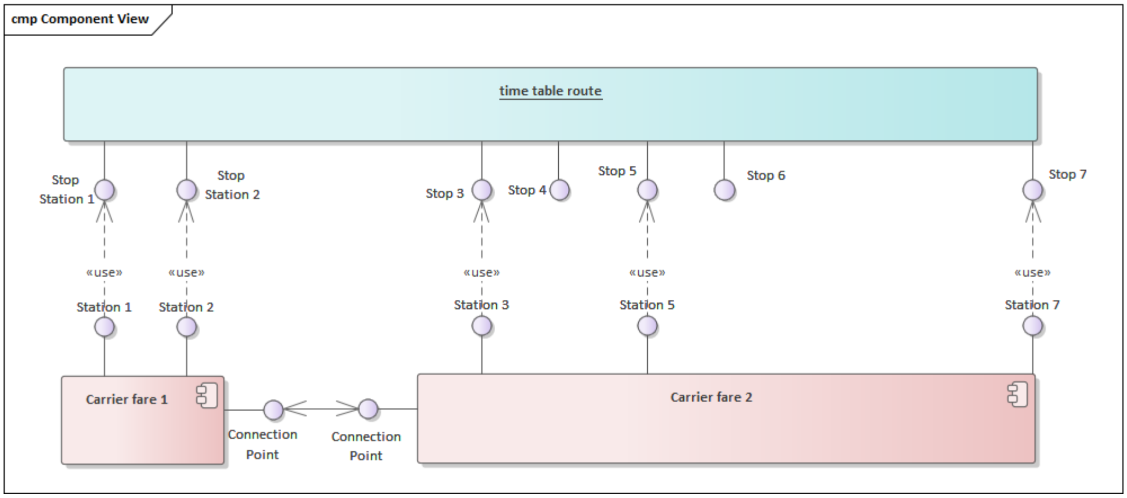
The distributor might need to remove doubled stations in routes in case the connection point is a real station used in both regional validity descriptions in case it is displayed as one combined text:

- Carrier 1: RegionalConstraint {Exit (A), RegionalValidity X – Y/Z- A}
- Carrier 2: RegionalConstraint {Entry (A), RegionalValidity A – C/D – E}
- *Result*: `X*Y/Z*A*A*C/D*E` –> `X*Y/Z*A*C/D*E`

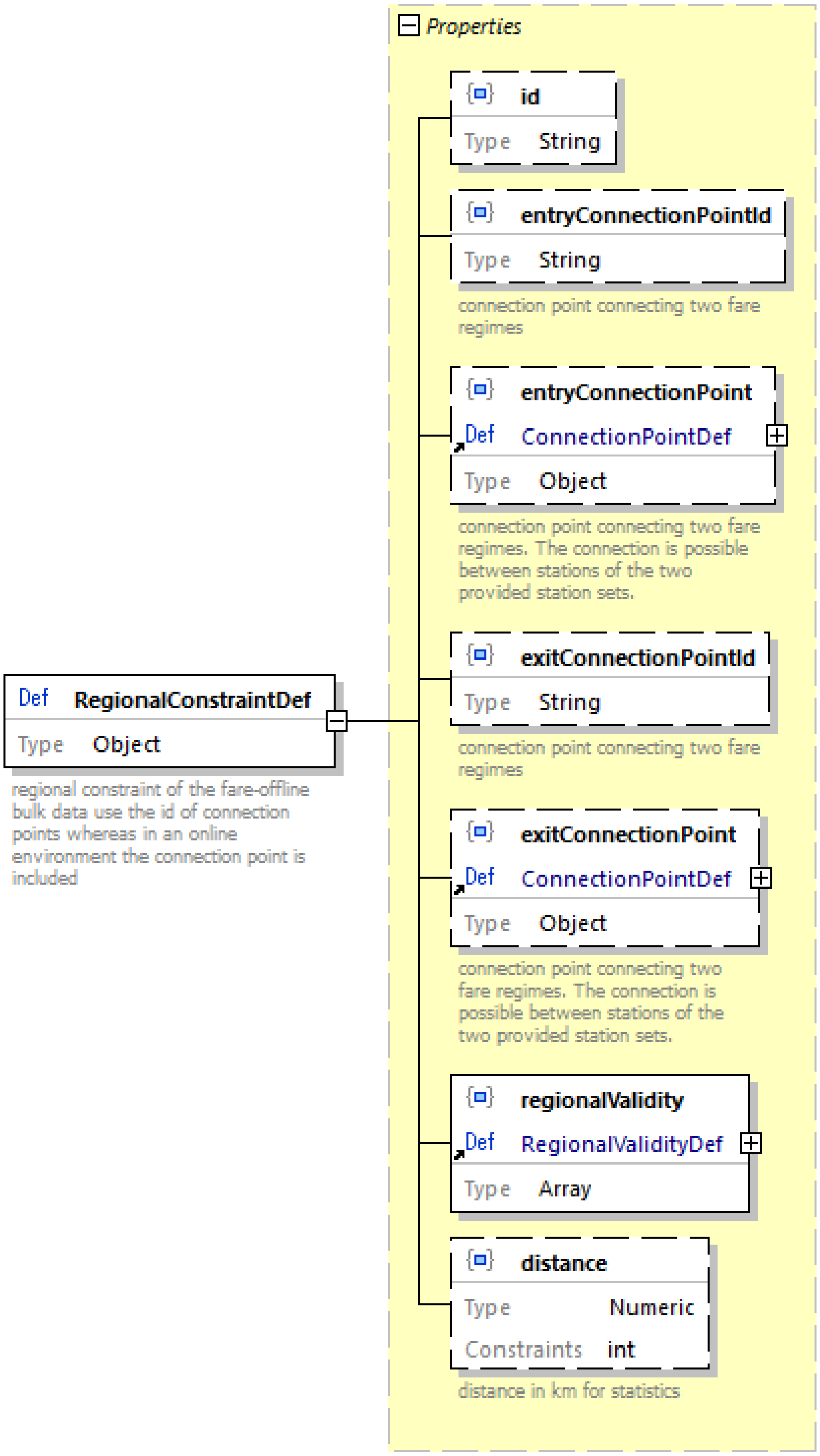
Connecting Regional Validity to Trips

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.



Connection Points and Timetable Routes

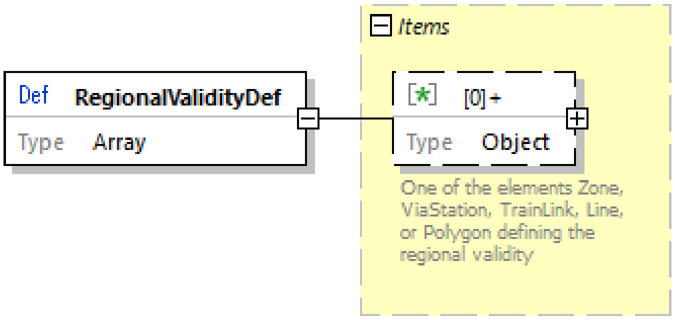

Properties
id
Type String
entryConnectionPointId
Type String
connection point connecting two fare regimes
entryConnectionPoint
Def ConnectionPointDef
Type Object
connection point connecting two fare regimes. The connection is possible between stations of the two provided station sets.
exitConnectionPointId
Type String
connection point connecting two fare regimes
exitConnectionPoint
Def ConnectionPointDef
Type Object
connection point connecting two fare regimes. The connection is possible between stations of the two provided station sets.
regionalValidity
Def RegionalValidityDef
Type Array
distance
Type Numeric
Constraints int
distance in km for statistics
Def RegionalConstraintDef
Type Object
regional constraint of the fare-offline bulk data use the id of connection points whereas in an online environment the connection point is included

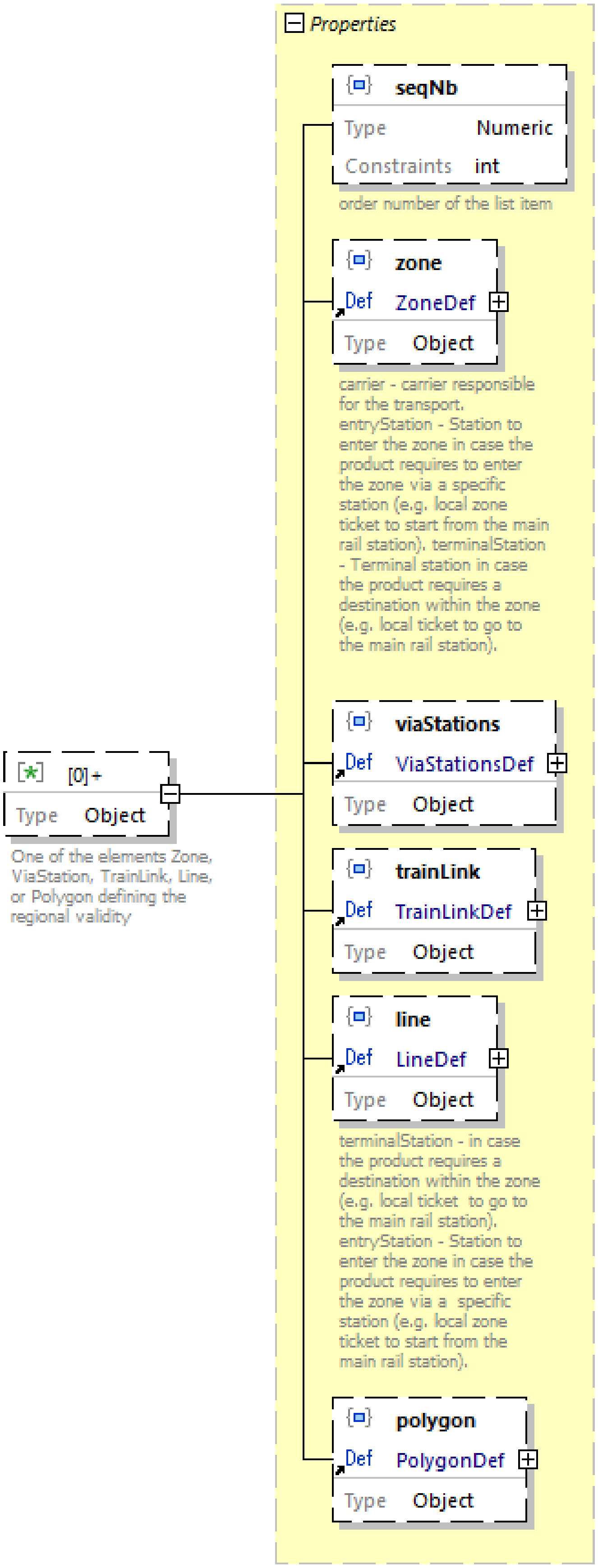
Regional Constraint

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 *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.


Def RegionalValidityDef
Type Array
Items
[*] [0]+
Type Object
One of the elements Zone, ViaStation, TrainLink, Line, or Polygon defining the regional validity



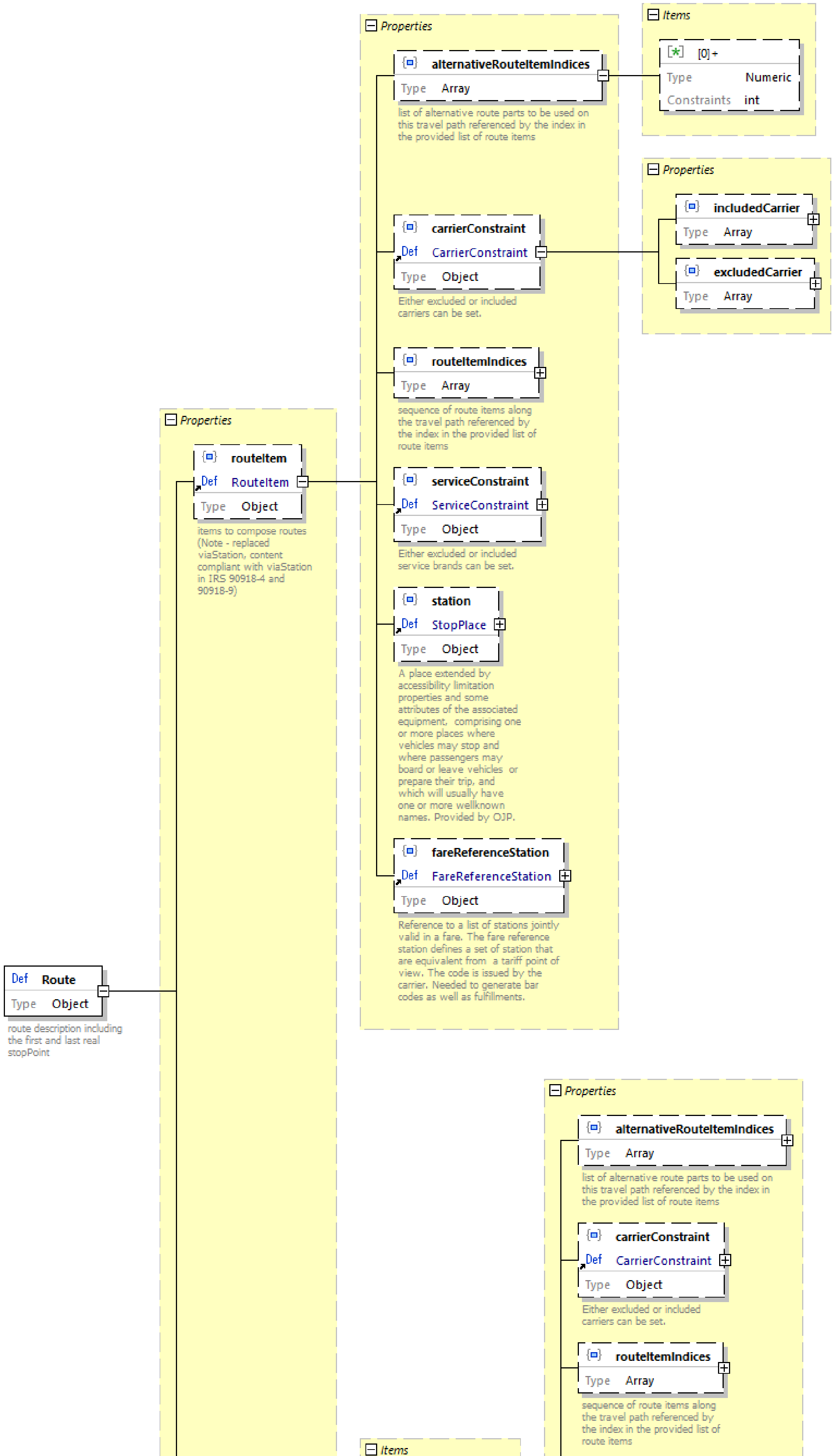
order number of the list item

carrier - carrier responsible for the transport. entryStation - Station to enter the zone in case the product requires to enter the zone via a specific station (e.g. local zone ticket to start from the main rail station). terminalStation - Terminal station in case the product requires a destination within the zone (e.g. local ticket to go to the main rail station).

terminalStation - in case the product requires a destination within the zone (e.g. local ticket to go to the main rail station). entryStation - Station to enter the zone in case the product requires to enter the zone via a specific station (e.g. local zone ticket to start from the main rail station).

One of the elements Zone, ViaStation, TrainLink, Line, or Polygon defining the regional validity

Extended route data structure including fare reference station sets.



route description including the first and last real stopPoint

items to compose routes (Note - replaced viaStation, content compliant with viaStation in IRS 90918-4 and 90918-9)

list of alternative route parts to be used on this travel path referenced by the index in the provided list of route items

Either excluded or included carriers can be set.

sequence of route items along the travel path referenced by the index in the provided list of route items

Either excluded or included service brands can be set.

A place extended by accessibility limitation properties and some attributes of the associated equipment, comprising one or more places where vehicles may stop and where passengers may board or leave vehicles or prepare their trip, and which will usually have one or more wellknown names. Provided by OJP.

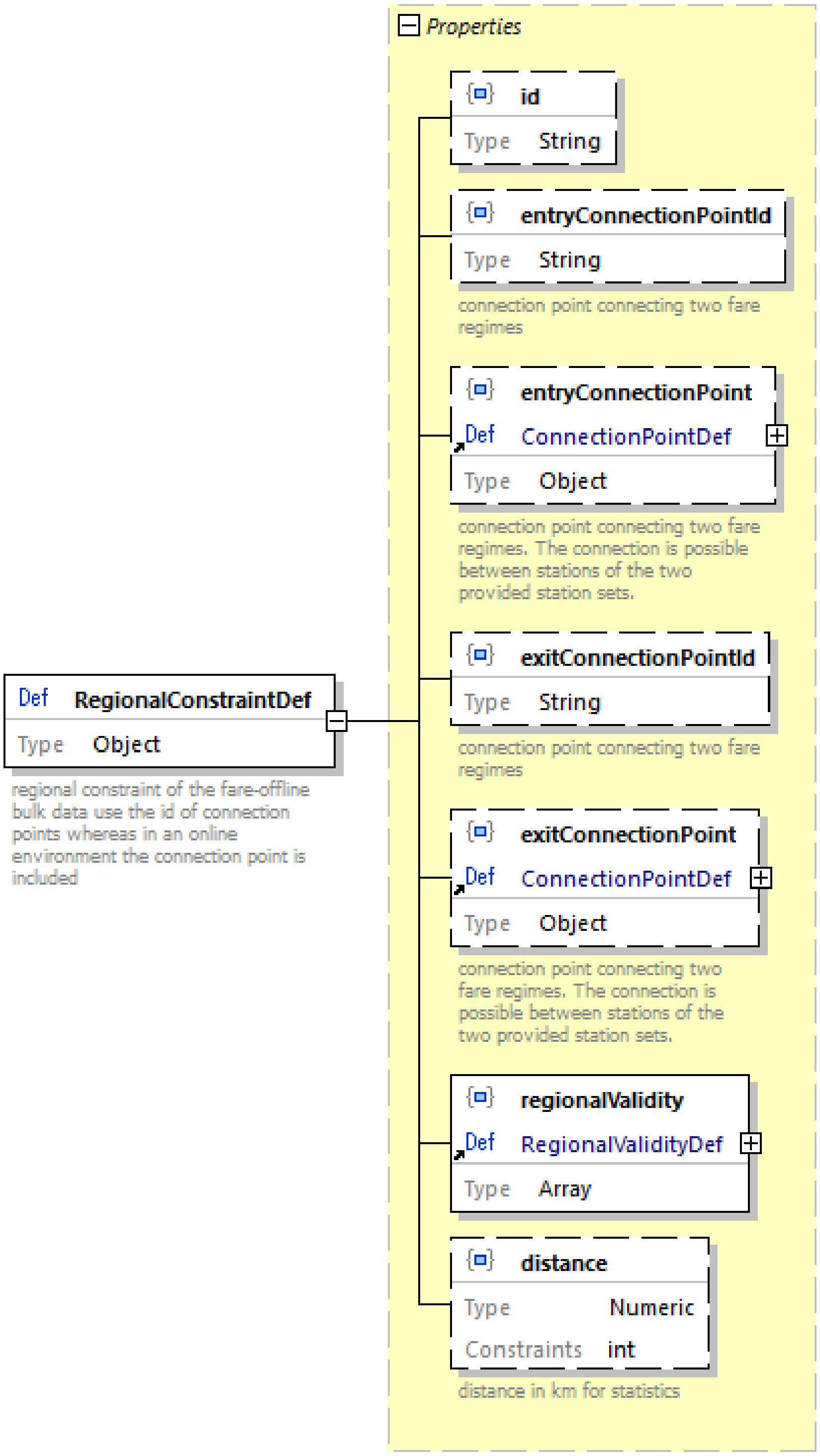
Reference to a list of stations jointly valid in a fare. The fare reference station defines a set of station that are equivalent from a tariff point of view. The code is issued by the carrier. Needed to generate bar codes as well as fulfillments.

list of alternative route parts to be used on this travel path referenced by the index in the provided list of route items

Either excluded or included carriers can be set.

sequence of route items along the travel path referenced by the index in the provided list of route items

Route

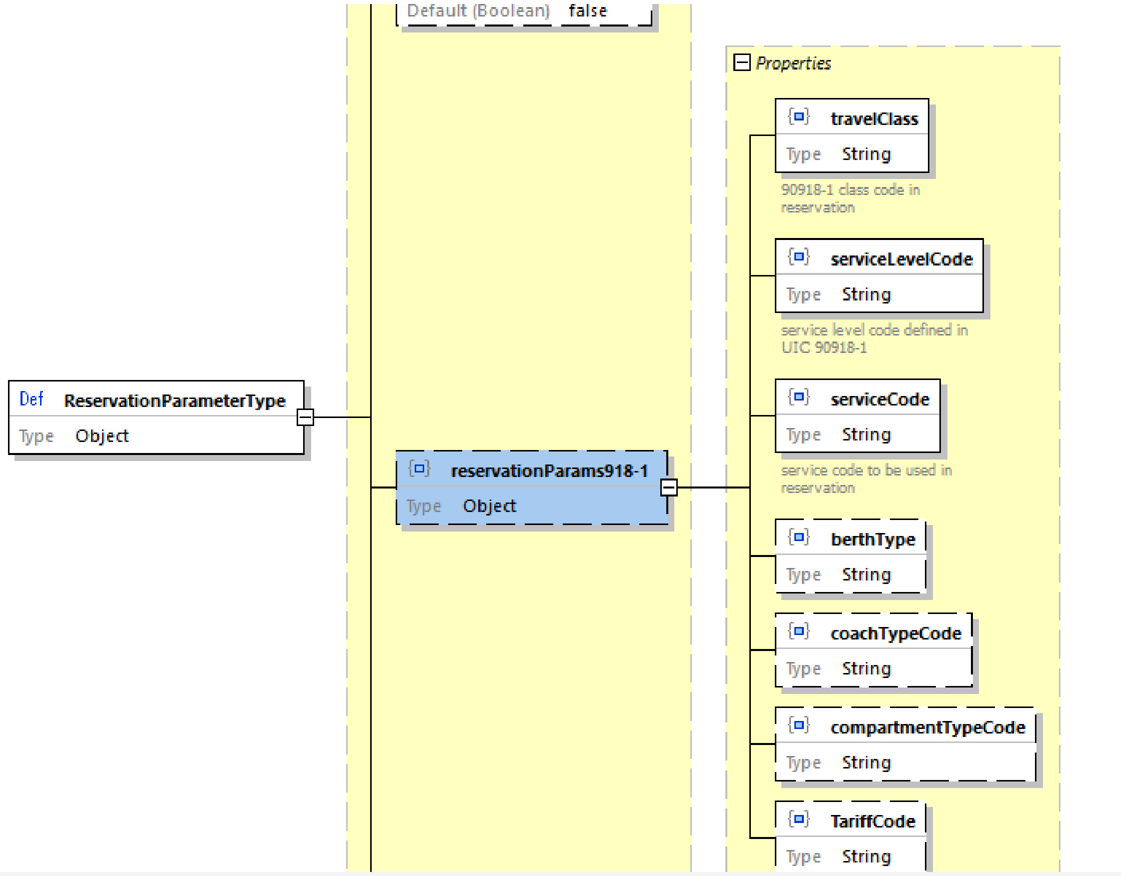

Properties
Def RegionalConstraintDef
Type Object
regional constraint of the fare-offline bulk data use the id of connection points whereas in an online environment the connection point is included
id
Type String
entryConnectionPointId
Type String
connection point connecting two fare regimes
entryConnectionPoint
Def ConnectionPointDef
Type Object
connection point connecting two fare regimes. The connection is possible between stations of the two provided station sets.
exitConnectionPointId
Type String
connection point connecting two fare regimes
exitConnectionPoint
Def ConnectionPointDef
Type Object
connection point connecting two fare regimes. The connection is possible between stations of the two provided station sets.
regionalValidity
Def RegionalValidityDef
Type Array
distance
Type Numeric
Constraints int
distance in km for statistics

Regional Constraint

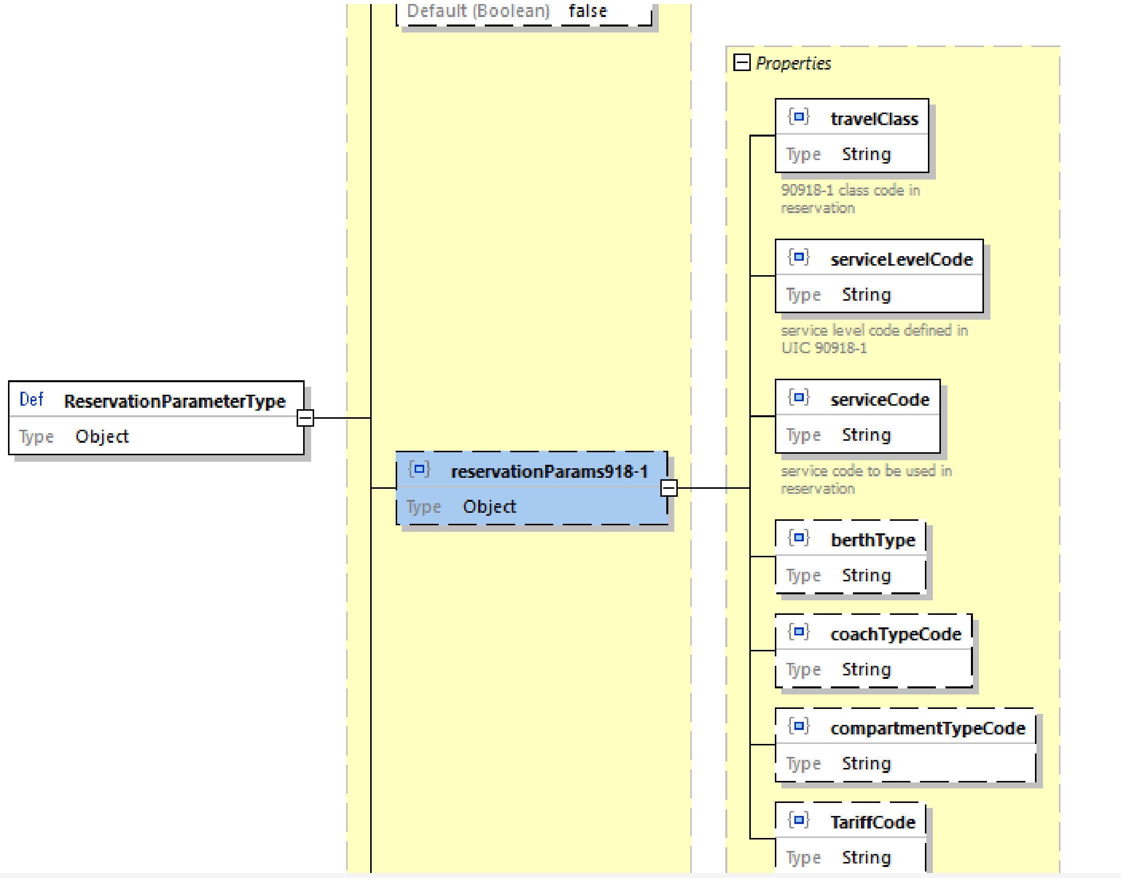
ReservationParameter

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

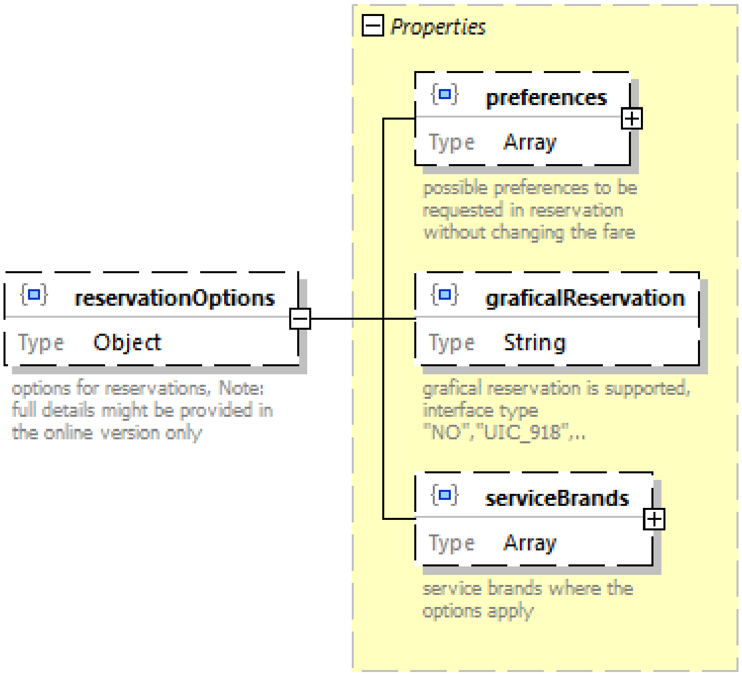
Code	Description
`reservationRequired`	A reservation must be made accompanying an NRT ticket.
`reservationParameters981-1`	Parameters to request the correct reservation using the interface according to *IRS 90918-1*.
`reservationOptions`	Reservation options available that would not change the offer (same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



Reservation Parameter



Reservation Parameter - 90918-1 Parameters


Properties
preferences
Type Array
possible preferences to be requested in reservation without changing the fare
reservationOptions
Type Object
options for reservations, Note: full details might be provided in the online version only
graficalReservation
Type String
grafical reservation is supported, interface type "NO","UIC_918",..
serviceBrands
Type Array
service brands where the options apply

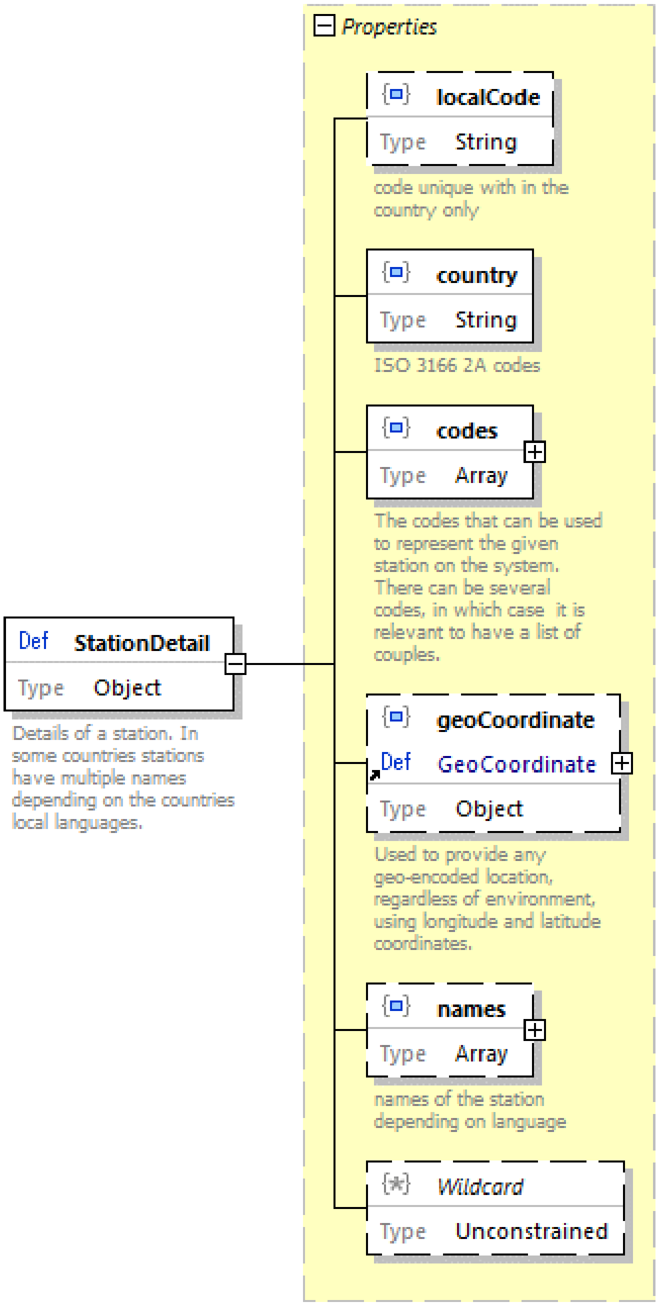
Reservation Parameter - Reservation Options

Code Lists

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

StationDetail

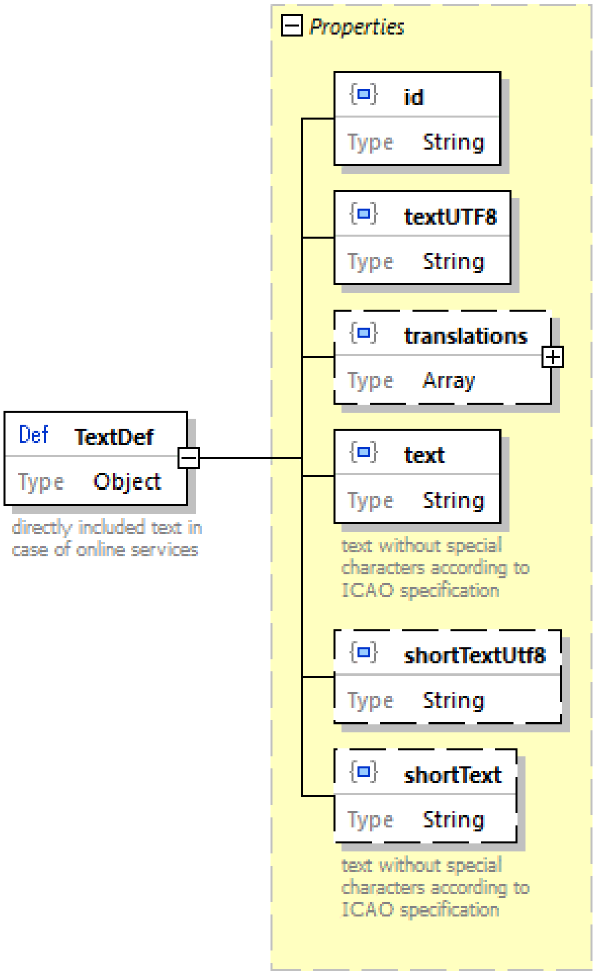
Details on stations including codes and names. Codes must include the MERITS code in case it is defined for a station.


Properties
Def StationDetail
Type Object
Details of a station. In some countries stations have multiple names depending on the countries local languages.
localCode
Type String
code unique with in the country only
country
Type String
ISO 3166 2A codes
codes
Type Array
The codes that can be used to represent the given station on the system. There can be several codes, in which case it is relevant to have a list of couples.
geoCoordinate
Def GeoCoordinate
Type Object
Used to provide any geo-encoded location, regardless of environment, using longitude and latitude coordinates.
names
Type Array
names of the station depending on language
Wildcard
Type Unconstrained

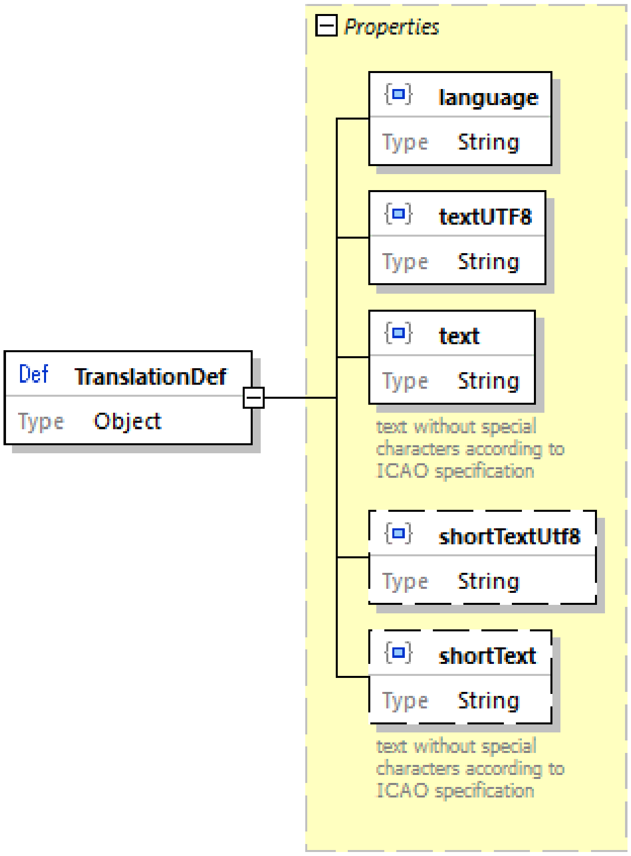
Station Detail

Text

Used for all textual descriptions where translations might be needed.


Properties
Def TextDef
Type Object
directly included text in case of online services
id
Type String
textUTF8
Type String
translations
Type Array
text
Type String
text without special characters according to ICAO specification
shortTextUtf8
Type String
shortText
Type String
text without special characters according to ICAO specification

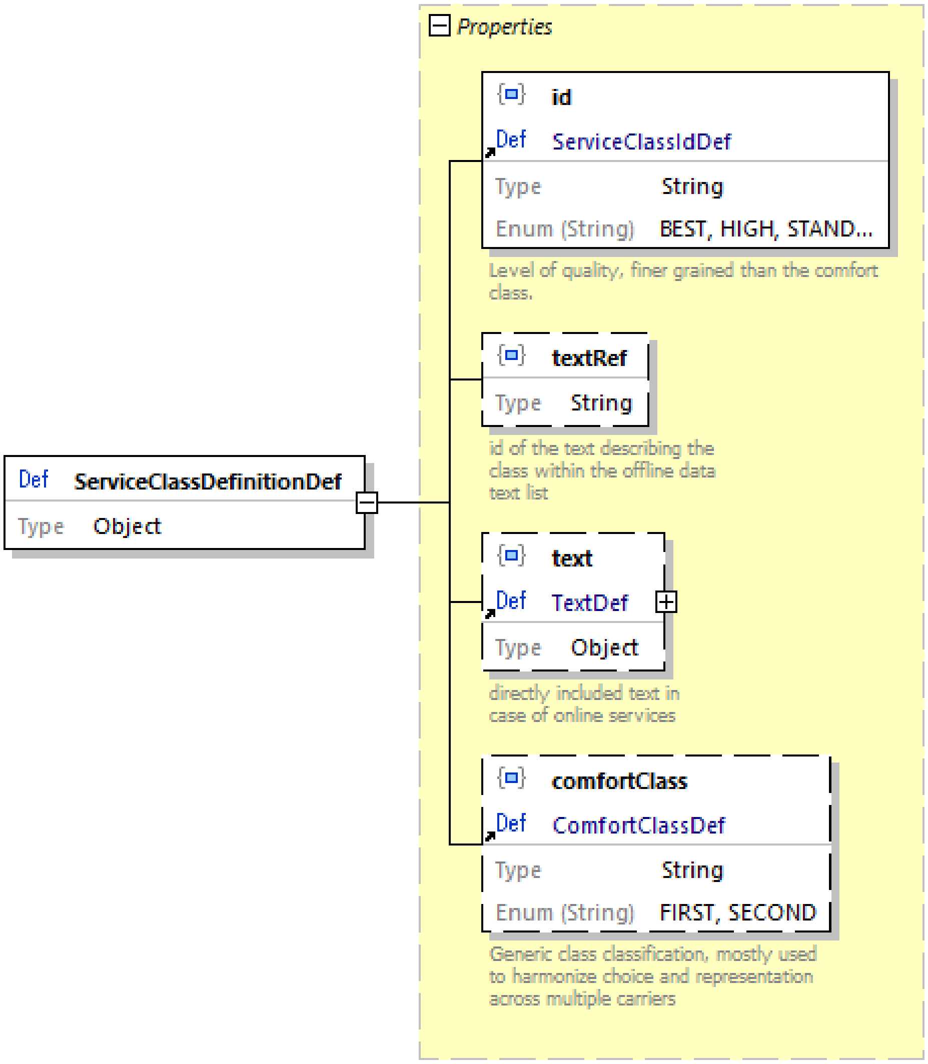
Text



Translation

ServiceClass

Service class provides textual descriptions for the predefined service classes.

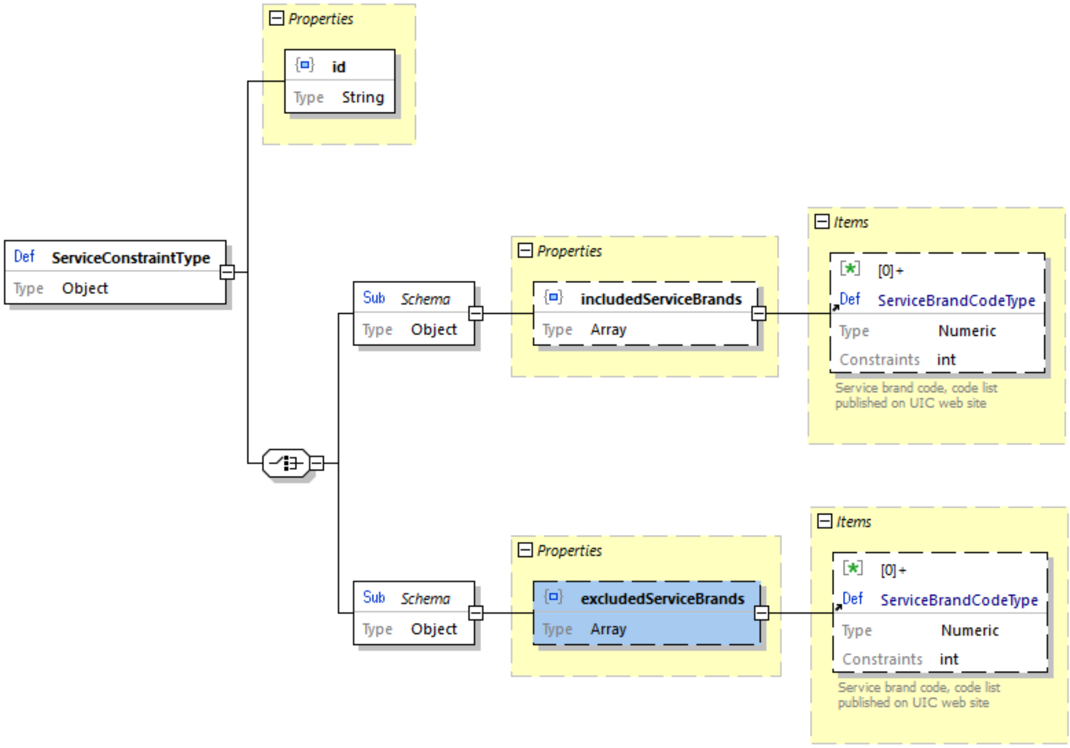

Properties
Def ServiceClassDefinitionDef
Type Object
id
Def ServiceClassIdDef
Type String
Enum (String) BEST, HIGH, STAND...
Level of quality, finer grained than the comfort class.
textRef
Type String
id of the text describing the class within the offline data text list
text
Def TextDef
Type Object
directly included text in case of online services
comfortClass
Def ComfortClassDef
Type String
Enum (String) FIRST, SECOND
Generic class classification, mostly used to harmonize choice and representation across multiple carriers

Service Constraint

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.



Service Constraint

Data Constraints on ServiceConstraint

Code	Description

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

ServiceLevel

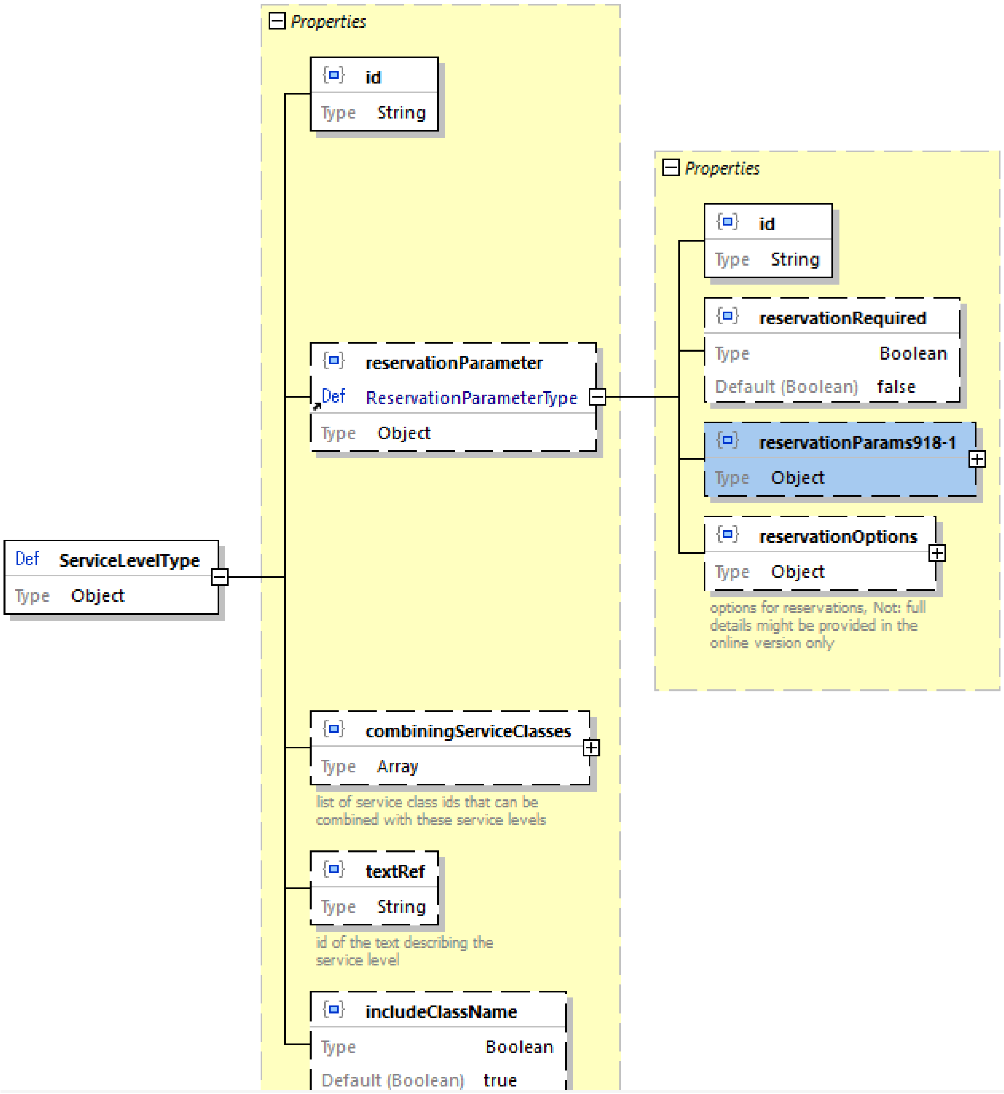
The service level data provide additional information (e.g. text) in the offline data exchange in case the reservation API of *IRS 90918-1* is used.

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 *IRS 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).

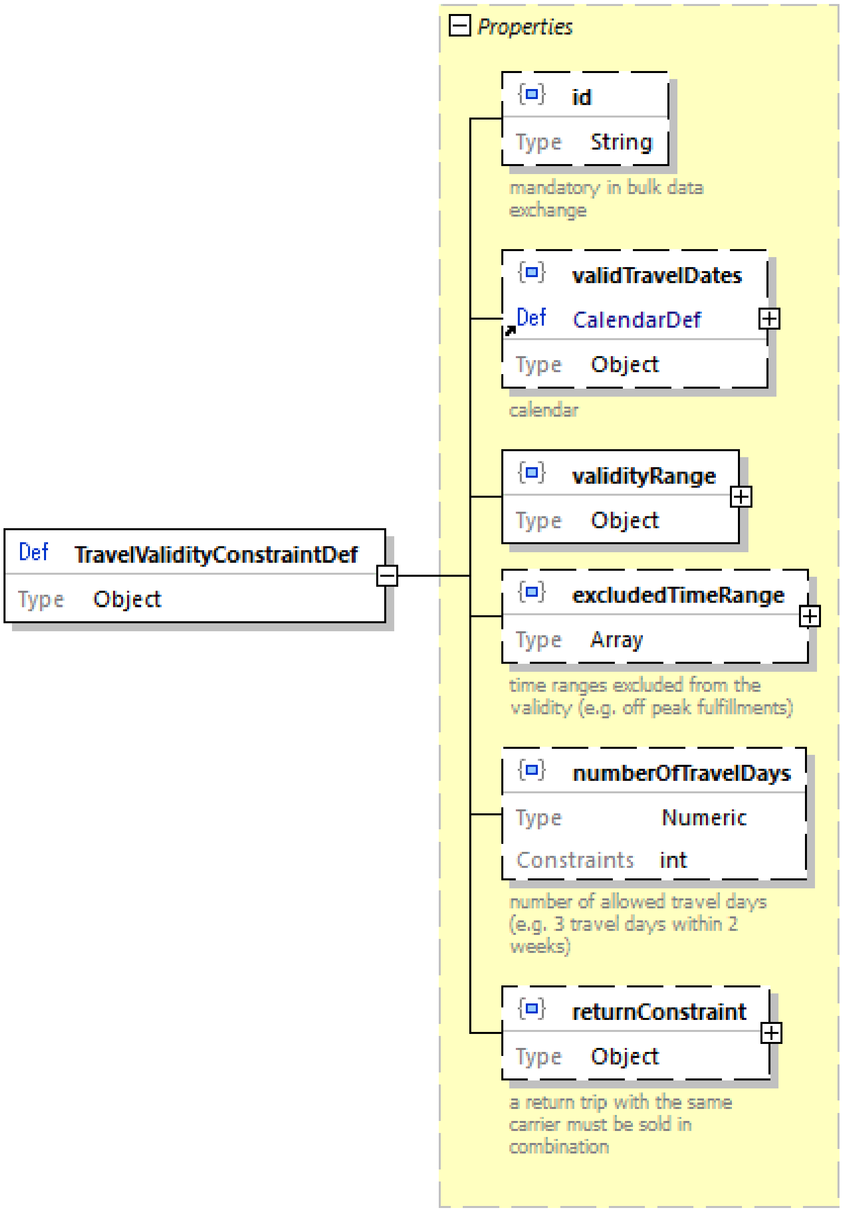


Service Level

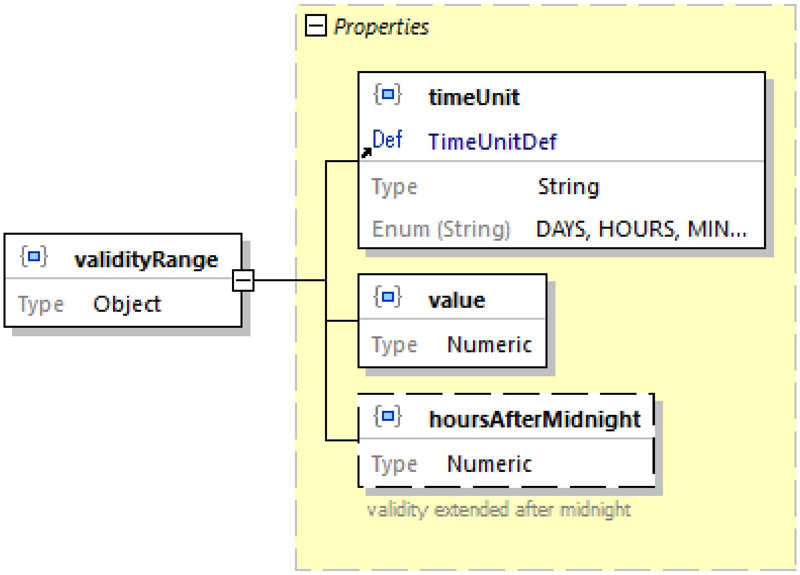
TravelValidityConstraint

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

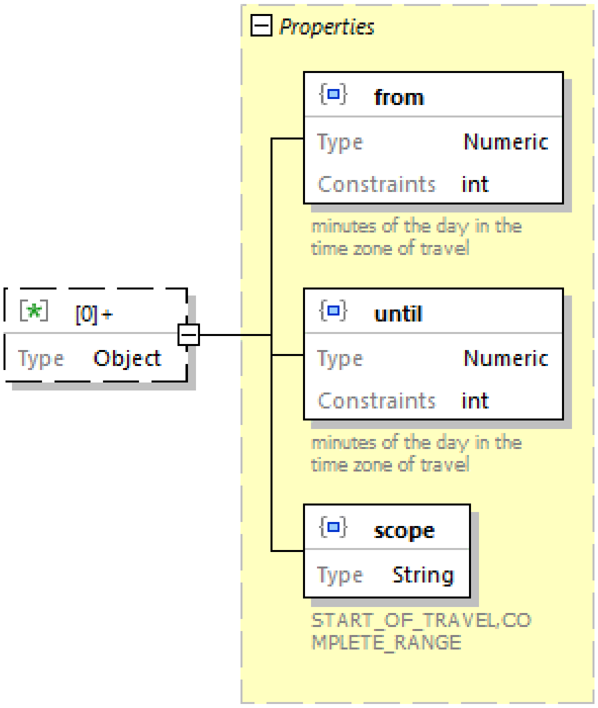
Data Constraints on TravelValidityConstraint


Properties
id
Type String
mandatory in bulk data exchange
validTravelDates
Def CalendarDef
Type Object
calendar
validityRange
Type Object
Def TravelValidityConstraintDef
Type Object
excludedTimeRange
Type Array
time ranges excluded from the validity (e.g. off peak fulfillments)
numberOfTravelDays
Type Numeric
Constraints int
number of allowed travel days (e.g. 3 travel days within 2 weeks)
returnConstraint
Type Object
a return trip with the same carrier must be sold in combination

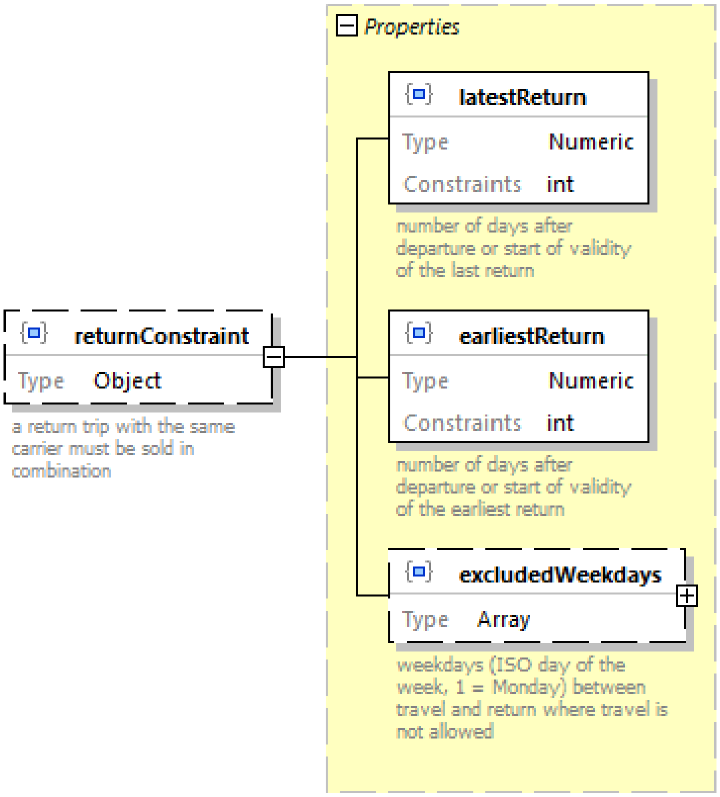
Travel Validity



Travel Validity - validity range


Properties
from
Type Numeric
Constraints int
minutes of the day in the time zone of travel
[0]+
Type Object
until
Type Numeric
Constraints int
minutes of the day in the time zone of travel
scope
Type String
START_OF_TRAVEL,COMPLETE_RANGE

Travel Validity - excluded time range


Properties
latestReturn
Type Numeric
Constraints int
number of days after departure or start of validity of the last return
returnConstraint
Type Object
a return trip with the same carrier must be sold in combination
earliestReturn
Type Numeric
Constraints int
number of days after departure or start of validity of the earliest return
excludedWeekdays
Type Array
weekdays (ISO day of the week, 1 = Monday) between travel and return where travel is not allowed

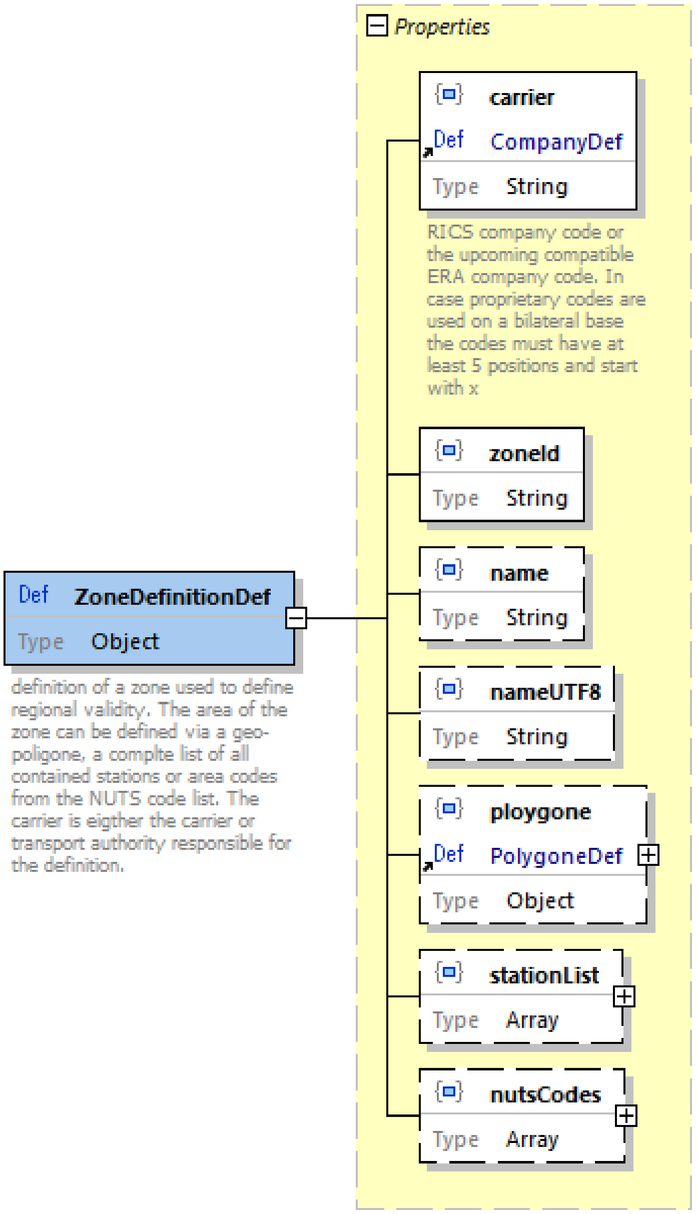
Travel Validity - return constraint

Code	Description
`excludedTimeRange`	`from` time < `until` 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 of a list of NUTS codes. Multiple definitions are allowed in case they define the same area.



RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

definition of a zone used to define regional validity. The area of the zone can be defined via a geo-poligone, a complte list of all contained stations or area codes from the NUTS code list. The carrier is eigther the carrier or transport authority responsible for the definition.

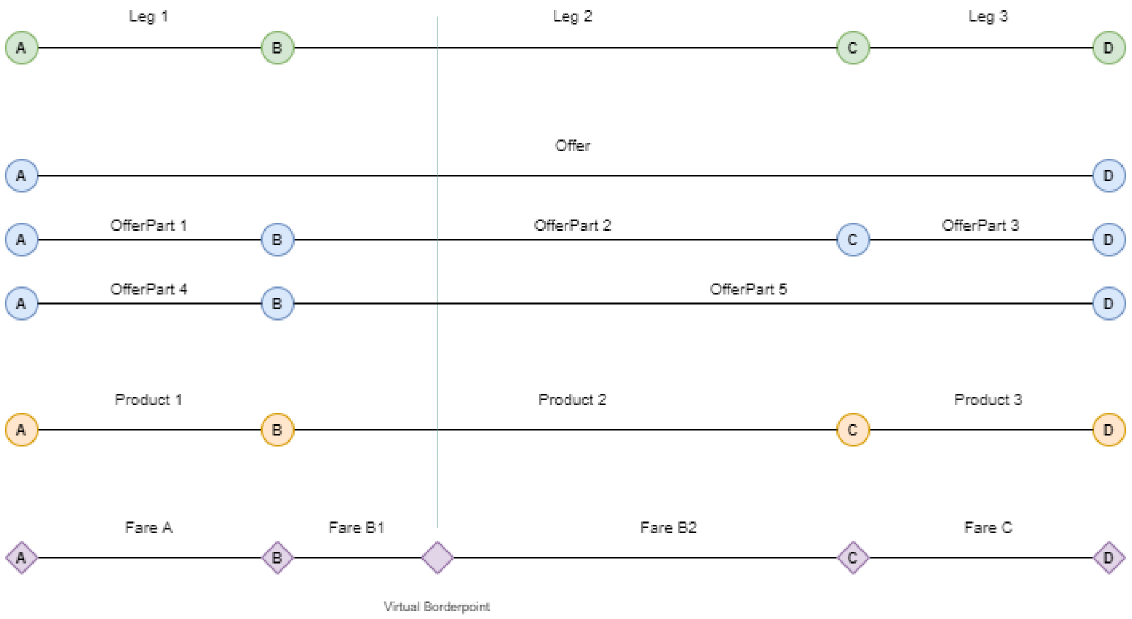
.

Relationships between Offers, OfferParts, Products and Fares

The entities `offer`, `offerpart`, `product` and `fare` represent different concepts.

An `offer` spans the trip consisting of segments. An `offerpart` spans a segment or - in the case of through fares - multiple segments. An `offerpart` references zero or two products. Mostly, the relationship is one-to-one (zero in case of included seat reservation, whereas two products are only referenced in some rare TGV-TER scenarios).

A `fare` is not necessarily bound to segments but reflects the tariff worlds. A `fare` often starts or ends at country borders where no train station exist. This border points are thus called `virtual border points` which can be modelled by `ConnectionPoints`. Fares are combined following a `fare combination model` to an offer.



Relationships between Offers, OfferParts, Products and Fares

Note: This example assumes that the complete trip can be priced.

Ways to Access Fares

There are three ways for a retailer to access fares.

First Scenario: Complete Offline Import of Fares

A retailer imports all fares of a given railway using a batch import from the OSDM-Offline platform.

The fares contain all relevant information to create offers, bookings and fulfillments (aka. tickets). If a booking occurs the RU informs the RU of the fare sold using UIC 301.

Second Scenario: Import of Relevant Fares at Offer Step

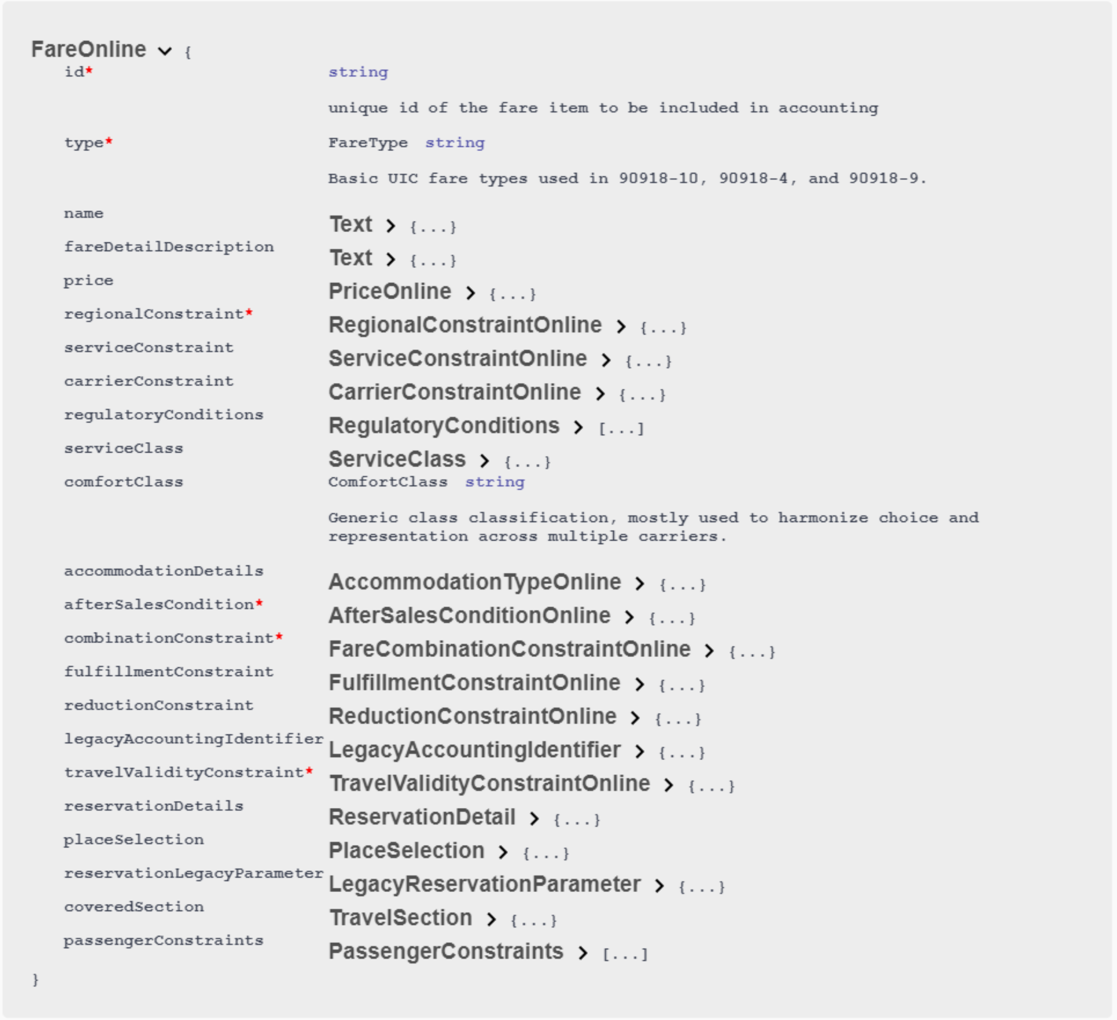
At offer time, a retailer looks up the fares for a given origin/destination in its database. If it doesn’t find it the retailer imports the relevant fares online. If a booking occurs the RU informs the RU of the fare sold using UIC 301.

Third Scenario: No Explicit Import of Fares

For a given origin/destination and date of travel the relevant fares are returned as part of the `Offer`.

If the fare is sold as part of an offer then the fare is booked online at the offering retailer.

Attributes of an Online Fare



```
FareOnline ⌄ {
   id*                          string

                                unique id of the fare item to be included in accounting

   type*                        FareType  string

                                Basic UIC fare types used in 90918-10, 90918-4, and 90918-9.

   name                         Text > {...}
   fareDetailDescription        Text > {...}
   price                        PriceOnline > {...}
   regionalConstraint*          RegionalConstraintOnline > {...}
   serviceConstraint            ServiceConstraintOnline > {...}
   carrierConstraint            CarrierConstraintOnline > {...}
   regulatoryConditions         RegulatoryConditions > [...]
   serviceClass                 ServiceClass > {...}
   comfortClass                 ComfortClass  string

                                Generic class classification, mostly used to harmonize choice and
                                representation across multiple carriers.

   accommodationDetails         AccommodationTypeOnline > {...}
   afterSalesCondition*         AfterSalesConditionOnline > {...}
   combinationConstraint*       FareCombinationConstraintOnline > {...}
   fulfillmentConstraint        FulfillmentConstraintOnline > {...}
   reductionConstraint          ReductionConstraintOnline > {...}
   legacyAccountingIdentifier   LegacyAccountingIdentifier > {...}
   travelValidityConstraint*    TravelValidityConstraintOnline > {...}
   reservationDetails           ReservationDetail > {...}
   placeSelection               PlaceSelection > {...}
   reservationLegacyParameter   LegacyReservationParameter > {...}
   coveredSection               TravelSection > {...}
   passengerConstraints         PassengerConstraints > [...]
}
```

Attributes of An Online Fare

Roles of Fare Attributes in the Booking and After Sales Processes

The fare contains all information so that a retailer can calculate correct prizes, render a valid fulfillment and account correctly.

The following sections describe which fare attributes are used at which step:

Offer Creation Step

Relevant fare attributes for the offer creation step are:

- `price`
- `regionalConstraint`

  Examples: a list of stations, a list of zones or a list of train link,…

  A retailer needs to guarantee that the whole trip is covered by fares.
- `serviceConstraint`

  Examples: `IC`, `TGV`, `BEX`, …

  A retailer can create offers only if the trip/segment is run by the service.
- `carrierConstraint`

  Examples: Thalys, Eurostar, …

  A retailer can create offers only if the trip/segment is run by the carrier.
- `regulatoryConditions`

  Examples: `CIV`, `MD` or `EU-PER`

  The regulatory conditions need to be indicated in the offer to inform the customer.
- `serviceClass`

  Examples: `HIGH`, `BEST`, `STANDARD` or `BASIC`

  A retailer needs to consider `serviceClass` depending on the `FareCombinationModel` applied.

  The service class needs to be indicated in the offer to inform the customer.
- `travelClass`

  Examples: `FIRST` or `SECOND`

  The comfort class needs to be indicated in the offer to inform the customer.
- `accommodationDetails`

  Example: `SEAT`, `COUCHETTE`, `BERTH` or `VEHICLE`

  If the train is a night train or car carriage the accommodation details need to be reflected in the offer.

- afterSalesCondition

  Example: Non-refundable after departure.

  A retailer needs to consider afterSalesCondition depending on the FareCombinationModelapplied.

  The refund/exchange conditions need to be indicated in the offer to inform the customer.

- combinationConstraint

  Examples: SEPARATE_CONTRACTS model, SEPARATE_TICKETS model, CLUSTERING model or COMBINATION model

  A retailer can only combine fares respecting the combination models.

- fulfillmentConstraint

  Examples: SIP, SID or SIS

  A retailer can only create offers which respect the constraints concerning the ticket control. If fulfillmentConstraint are mixed, then all constraints need to be served.

- reductionConstraint

  Example: 1085_GA where 1085 denotes SBB, 1080_Bahncard50, where 1080 denotes DB, …

  A retailer can create offers only if the passenger(s) own(s) the reduction(s).

- travelValidityConstraint

  Example: Valid 24h after departure

  The travel validity constraint needs to be communicated in the offer to inform the customer.

- placeSelection

  Example: Selected places in a graphical seat map, reference place for adjacent reservation or place preferences

  A retailer most respect the selected places. The other types are optional to be respected by the retailer.

- coveredSection

  Example: Start and end location

  A retailer has to create an offer that covers the whole trip from start location to end location.

Booking Step

Relevant fare attributes for the booking process step are:

- `price`
- `reservationDetails`

  Example: Wagon 19, Seat 44

  The reservation details need to be communicated at the pre-booking step to the customer.
- `legacyAccountingIdentifier`

  The legacy accounting identifier information is used to write a correct 301 record.

Fulfillment Step

Relevant fare attributes for the fulfillment step are:

- `price`

  The price needs to be communicated to the passenger(s), e.g. printed on the ticket.

  Additionally, it needs to be encoded in the security element(s).
- `regionalConstraint`

  The regional constraint need to be communicated to the passenger(s), e.g. printed on the ticket.

  Additionally, it needs to be encoded in the security element(s).
- `regulatoryConditions`

  The regulatory conditions need to be communicated to the passengers(s), e.g. printed on the ticket.
- `reservationsDetails`

  The reservation details need to be communicated to the passengers(s), e.g. printed on the ticket.

  Additionally, it needs to be encoded in the security element(s) in case of mandatory reservations.
- `serviceClass`

  The service class need to be communicated to the passengers(s), e.g. printed on the ticket.

  Additionally, it needs to be encoded in the security element(s).

- `travelValidityConstraint`

  The date are needed to create valid barcode and control data.

  Additionally, it needs to be encoded in the security element(s).

- `passengerConstraint`

  Example: Age between 6 and 16 years

  The passenger constraint need to be communicated to the passengers(s), e.g. printed on the ticket.

After Sale

Refund Offer Creation Step

Relevant fare attributes for the creation of a refund offer are:

- `price`
- `afterSaleConditions`

  A retailer can create offers only if the after sale condition support its creation.

Exchange Offer Creation Step

Relevant fare attributes for the creation of a refund offer are the `afterSaleConditions` as well as all the attributes for offer creation.

Technical Principles

Design Guidelines

- **Do not reinvent the wheel** - Use existing concepts whenever possible (e.g. type system of OpenAPI, Problem details,…).
- Strive for a Level 3 of REST maturity.
- Use semantic versioning.

Derived Guidelines

- Whenever a resource returned in a response can contain embedded resources, the request must allow specifying whether and which embedded resources should be returned in full or as references.
- Follow Zalando RESTful API and Event Scheme guidelines
- Use of the JSON Problem element
- Standard Patch operations (not JSON PATCH)
- A resource is either represented in full or as a reference. The reference element has the name of the resource post-fixed with “Ref”. References normally only contains the URL to the referenced resource and a title element allowing to summarize the resource in one short string
- Although examples or recommendations are provided as to which information should best be represented in the title string, each implementor as the freedom to modify it to best suit his needs.
- Enumerations for very stable entities with limited set only, otherwise code lists. Stations codes are code lists.
- Where possible, existing UIC code lists should be favored.
- Creation/ modification calls return the created/modified resource (not just an ok code)

Error Handling

In order to communicate errors to a consumer we support RFC7807.

This RFC defines a “problem detail” as a way to carry machine- readable details of errors in a HTTP response to avoid the need to define new error response formats for HTTP APIs.

A problem details object can have the following members:

- `type`: A URI reference RFC3986 that identifies the problem type. This specification encourages that, when dereferenced, it provide human-readable documentation for the problem type (e.g., using HTML [W3C.REC-html5-20141028]). When this member is not present, its value is assumed to be “about:blank”.
- `title`: A short, human-readable summary of the problem type. It SHOULD NOT change from occurrence to occurrence of the problem, except for purposes of localization (e.g., using proactive content negotiation; see RFC7231, Section 3.4).
- `status`: The HTTP status code (RFC7231, Section 6) generated by the origin server for this occurrence of the problem.

- `detail`: A human-readable explanation specific to this occurrence of the problem.
- `instance`: A URI reference that identifies the specific occurrence of the problem. It may or may not yield further information if dereferenced.

Consumers MUST use the `type` string as the primary identifier for the problem type; the `title` string is advisory and included only for users who are not aware of the semantics of the URI and do not have the ability to discover them (e.g., offline log analysis). Consumers SHOULD NOT automatically dereference the type URI.

Functional Errors and Error Codes

In order that OSDM implementations behave consistently in error situations, a list of error codes and warnings have been defined that must be supported in case of functional errors by all implementations.

Authentication

The following three design principles are binding for each implementor:

1. Don’t reinvent crypto, thus we are using **OAuth2**.

2. The JWTs in use for the authentication should be **short-lived** (think of timeout duration single-digit multiples)

3. The JWTs sent by the consumer, regardless of where they are generated, must be **digitally signed** using a private key for which the provider is able to find the matching public key

These principles can be implemented as follows:

- A **registration service** allows a consumer to register the necessary data (like: organization, technical admin, commercial admin, support line, other non-functional requirements like throttling limits) for approvement by the team providing access. any registered consumer will be approved or rejected if approved a unique, technical “ClientId” results and will be passed to the consumer

- A **login service** allows to request a valid token (for e.g. OAauth2) by the registered ClientId and related secret per configuration the validity duration of the token may be set (for e.g. 10min.), after that the token must be renewed by the consumer optionally the token might be revoked (for e.g. if your devOps realizes the consumer does not behave as expected)

In multi-environments (like DEV, TEST, INT, PROD) consumers might register for each environment separately.

User Lookup

From a devOps perspective it might by hard to control who uses your API (for e.g. `ClientId` and secret might by passed around or hacked).

The following issues should be defined outside of OSDM:

- Additional legal contracts with consumer
- IP white listing
- Mapping of related user properties to the technical `ClientId`

Offline Sales 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 Delivery Schemas

The data delivery will contain the version number of the used json schema 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.

Versioning of Data Delivery Data

The data delivery has a unique id. It can indicate that it replaces a previous delivery by indicating the data delivery id of the delivery to be replaced. Deliveries can be marked as optional. In this case a user of the data delivery might ignore the delivery. Deliveries marked as mandatory have to be used.

Automated Bulk Data Exchange

Automated asynchronous bulk data transfer is an option implemented by queues. The queues must implement the AMQP 1.0 specification.

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

Queue authentication and encryption must use TLS version 1.2.

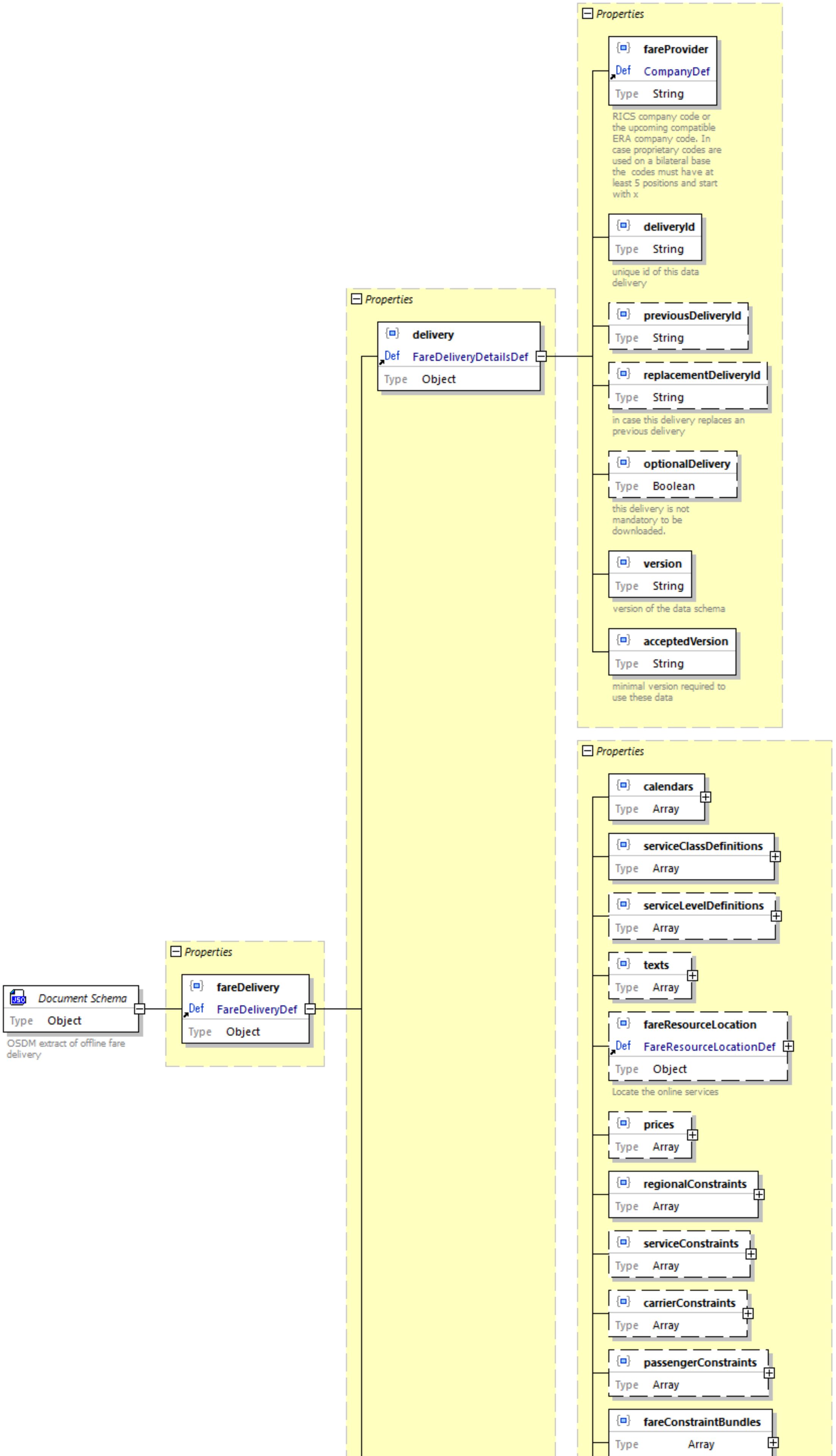
AMPQ Header Parameter

Parameter	Usage
`message-id`	Technical id of the data transfer, not the data delivery id in the data.

Parameter	Usage
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.



OSDM extract of offline fare delivery

RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

unique id of this data delivery

in case this delivery replaces an previous delivery

this delivery is not mandatory to be downloaded.

version of the data schema

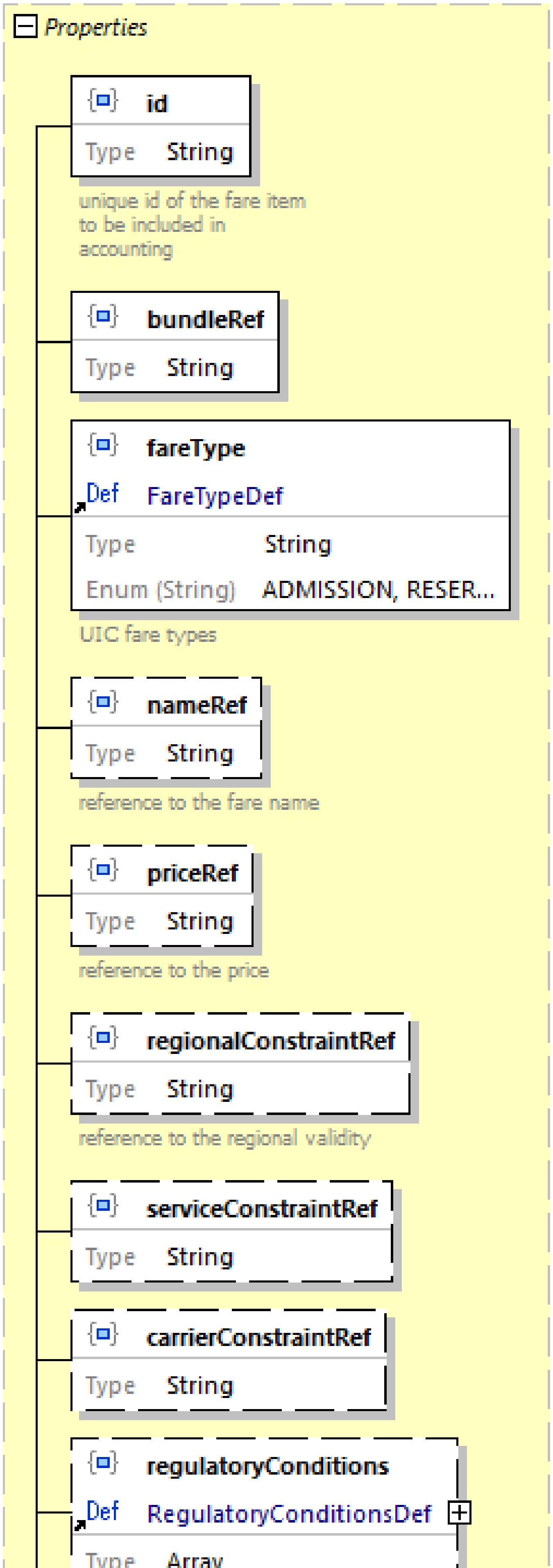
minimal version required to use these data

Locate the online services

Data Structure for Bulk Data

Definition of a single fare

The single fare represents the smallest unit to be integrated in an offer. Within the offline data the fare collects the references to the constraints that need to be applied and the price.



Properties

id
Type String
unique id of the fare item to be included in accounting

bundleRef
Type String

fareType
Def FareTypeDef
Type String
Enum (String) ADMISSION, RESER...
UIC fare types

nameRef
Type String
reference to the fare name

priceRef
Type String
reference to the price

regionalConstraintRef
Type String
reference to the regional validity

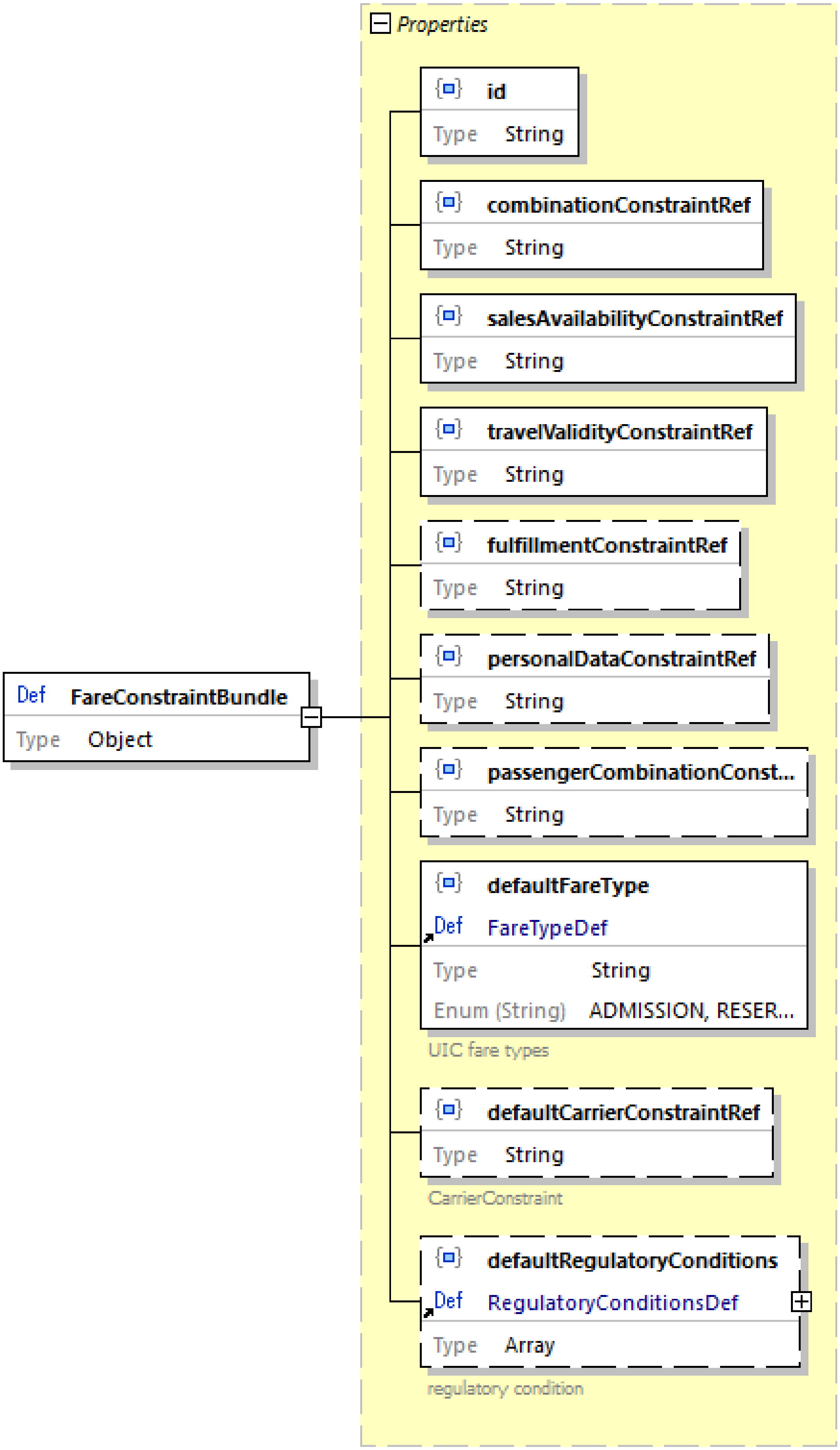
serviceConstraintRef
Type String

carrierConstraintRef
Type String

regulatoryConditions
Def RegulatoryConditionsDef
Type Array

Fare Structure

Some constraints are bundled within the fare constraint bundle to avoid repeating the same data too many times:

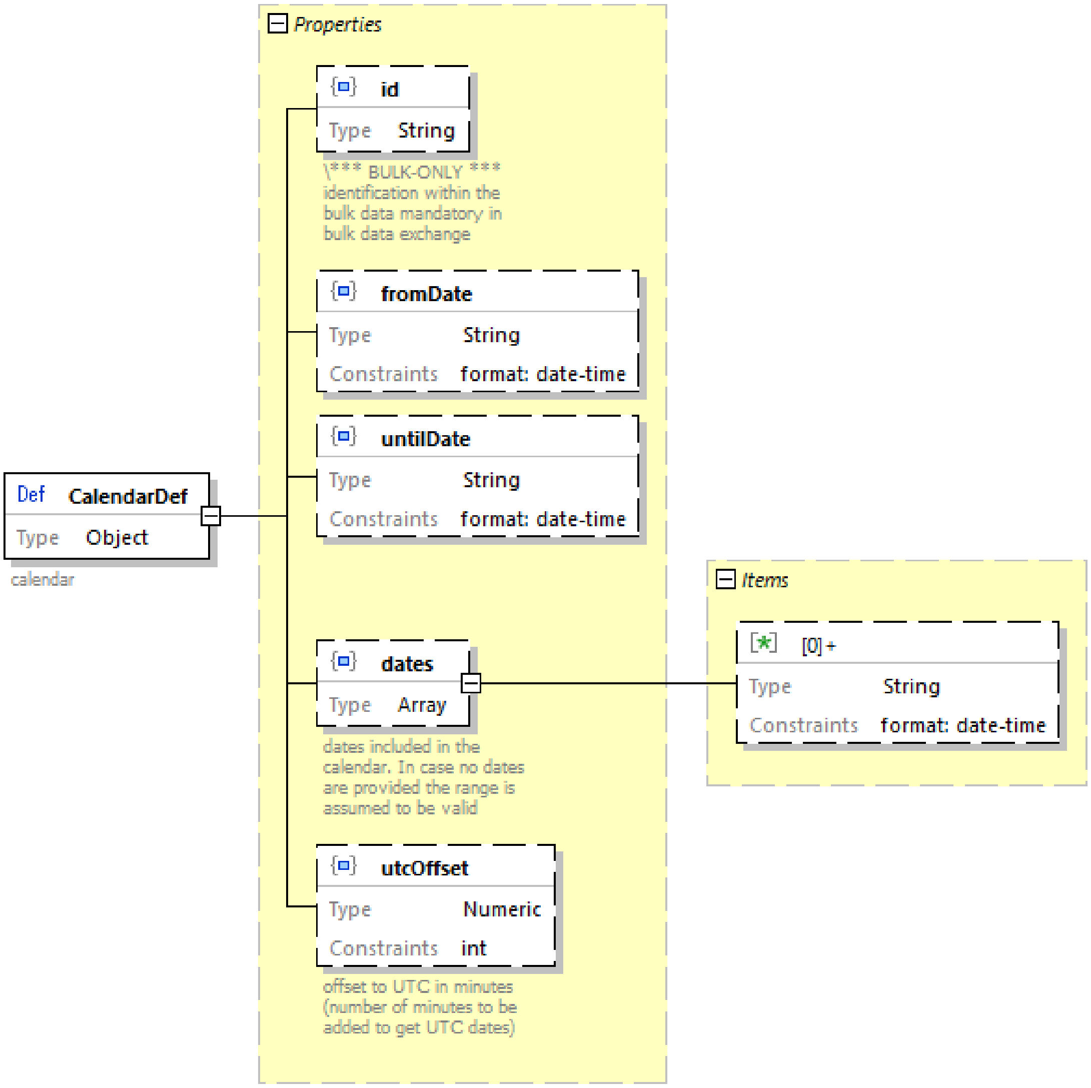

Properties
Def FareConstraintBundle
Type Object
id
Type String
combinationConstraintRef
Type String
salesAvailabilityConstraintRef
Type String
travelValidityConstraintRef
Type String
fulfillmentConstraintRef
Type String
personalDataConstraintRef
Type String
passengerCombinationConst...
Type String
defaultFareType
Def FareTypeDef
Type String
Enum (String) ADMISSION, RESER...
UIC fare types
defaultCarrierConstraintRef
Type String
CarrierConstraint
defaultRegulatoryConditions
Def RegulatoryConditionsDef
Type Array
regulatory condition

Basic definitions included in the data delivery

Basic definitions are provided within each data delivery. The basic definitions are included only once and are references within the data via their id.

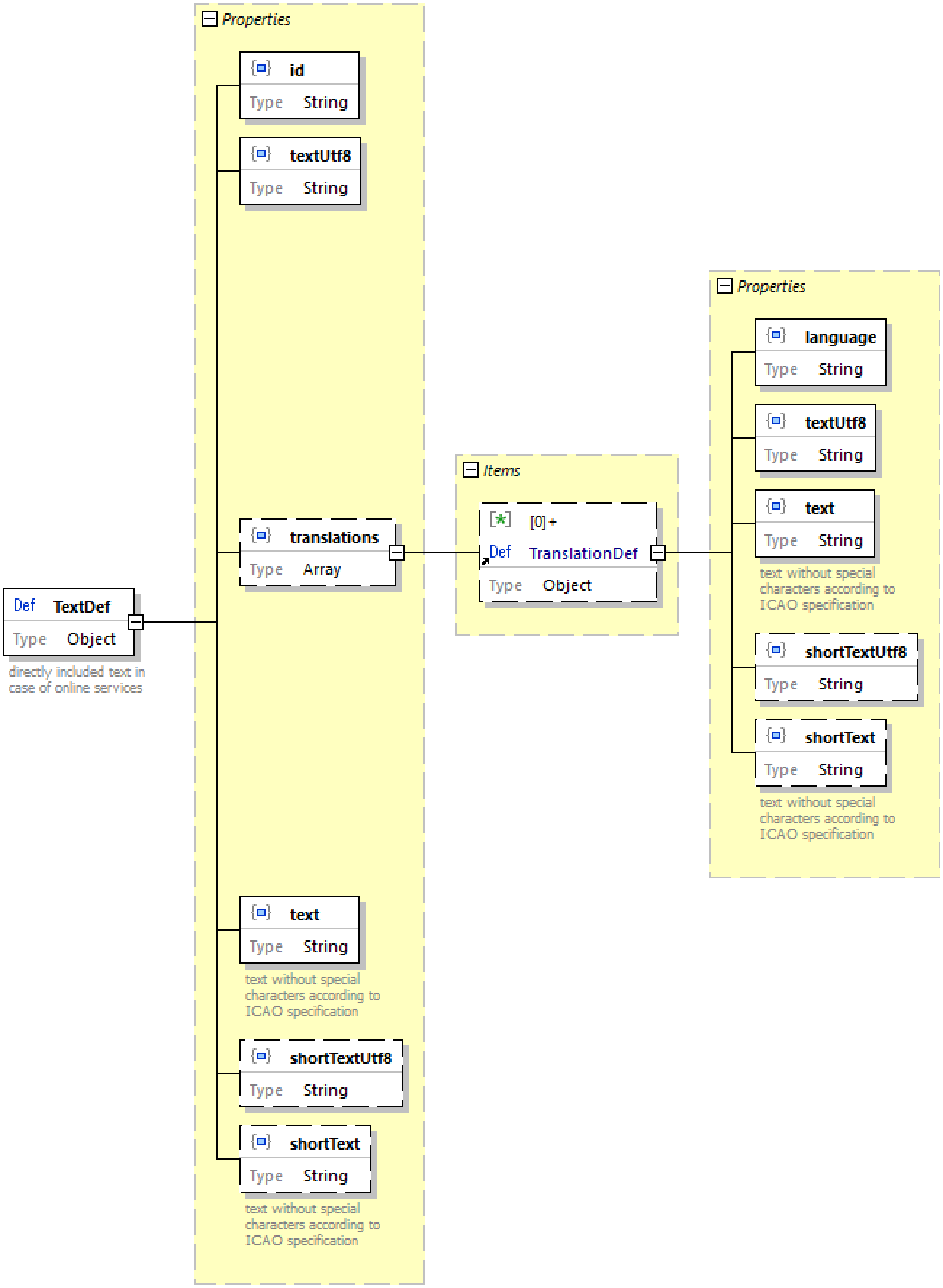
Calendar

A data structure to define a calendar e.g. used in sales availability.



Text

All texts provided wit the data use the text data structure providing short and long texts and translations in different languages. To support legacy implementations and the conversation to the 108.1 specification additional texts without special characters can be defined.


Properties
id
Type String
textUtf8
Type String
translations
Type Array
text
Type String
text without special characters according to ICAO specification
shortTextUtf8
Type String
shortText
Type String
text without special characters according to ICAO specification
Def TextDef
Type Object
directly included text in case of online services
Items
[0] +
Def TranslationDef
Type Object
Properties
language
Type String
textUtf8
Type String
text
Type String
text without special characters according to ICAO specification
shortTextUtf8
Type String
shortText
Type String
text without special characters according to ICAO specification

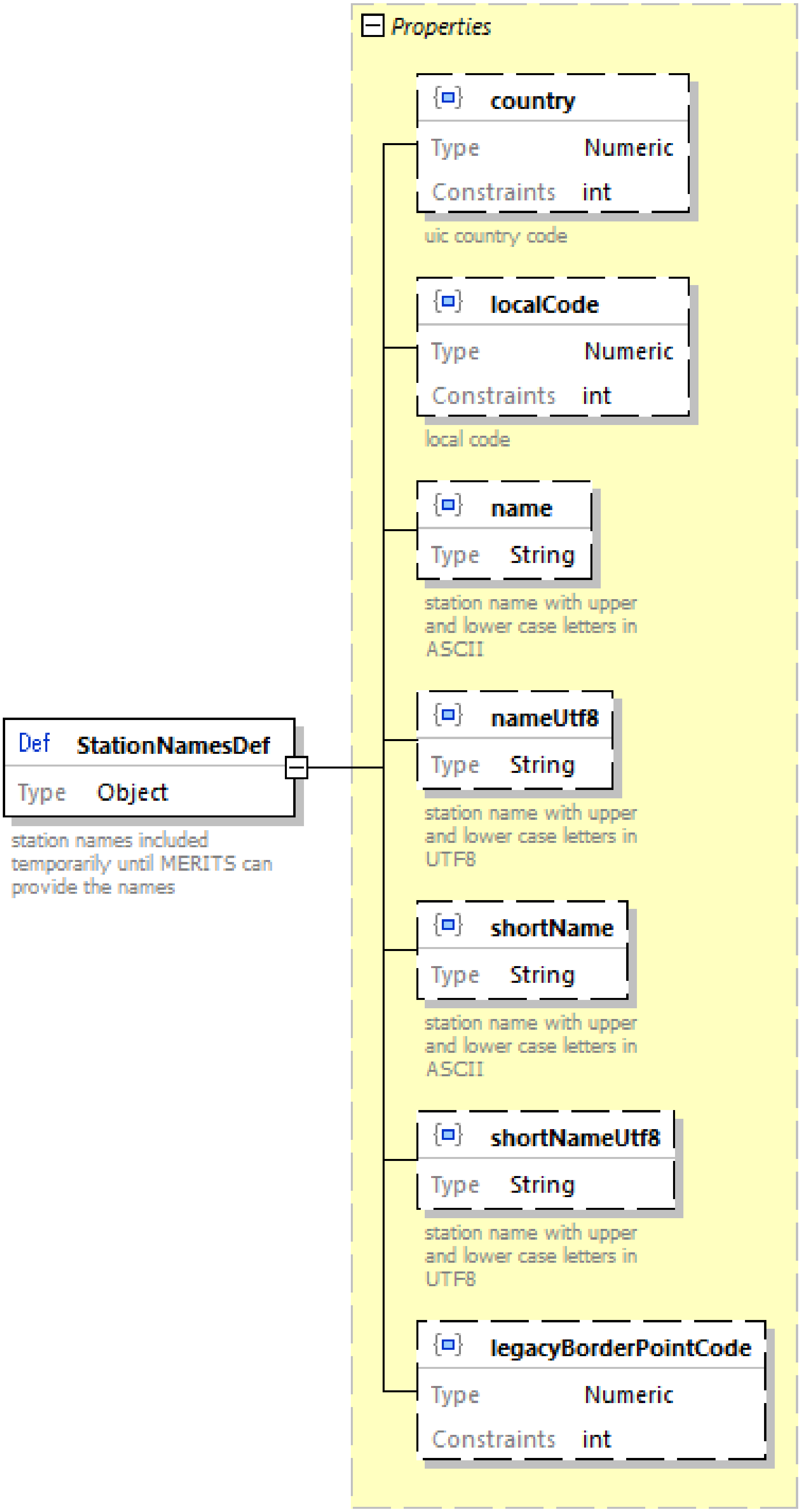
Fare Structure

Station Names

Station names have been included within the data exchange to support names including special characters and names of different length. If in the future the station data exchange of MERITS is capable of providing these names they can be removed here. The station codes used must be codes as defined in MERITS / TAP-TSI.

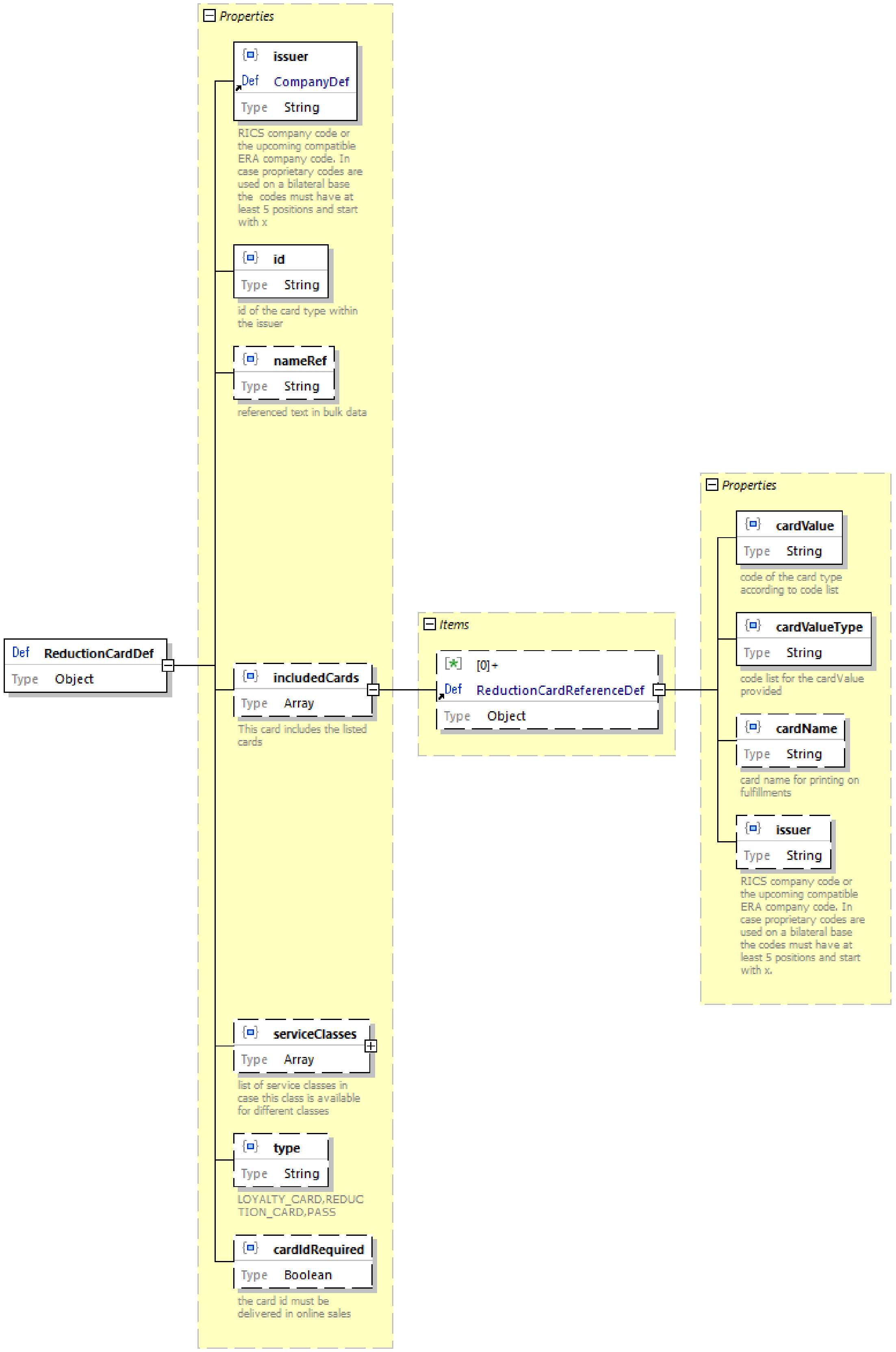
Station names provides multi language names in short and long form as currently no other data source can provide these names. Short names are used within the route descriptions whereas the long for is used for entry and exit stations.

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


Def StationNamesDef
Type Object
station names included temporarily until MERITS can provide the names
Properties
country
Type Numeric
Constraints int
uic country code
localCode
Type Numeric
Constraints int
local code
name
Type String
station name with upper and lower case letters in ASCII
nameUtf8
Type String
station name with upper and lower case letters in UTF8
shortName
Type String
station name with upper and lower case letters in ASCII
shortNameUtf8
Type String
station name with upper and lower case letters in UTF8
legacyBorderPointCode
Type Numeric
Constraints int

Reduction Cards

Tis covers the definition of reduction cards used in the fares. The name and some basic information of the cards can be defined. The reduction itself (percentage) is not included as the fare price already provides the reduced price. Some provider specific cards are accepted by other carriers as a generic card. This can be expressed via the included cards feature. E.g. MyCard could be accepted as RailPlus card by others, so MyCard includes RailPlus.



RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

id of the card type within the issuer

referenced text in bulk data

This card includes the listed cards

code of the card type according to code list

code list for the cardValue provided

card name for printing on fulfillments

RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x.

list of service classes in case this class is available for different classes

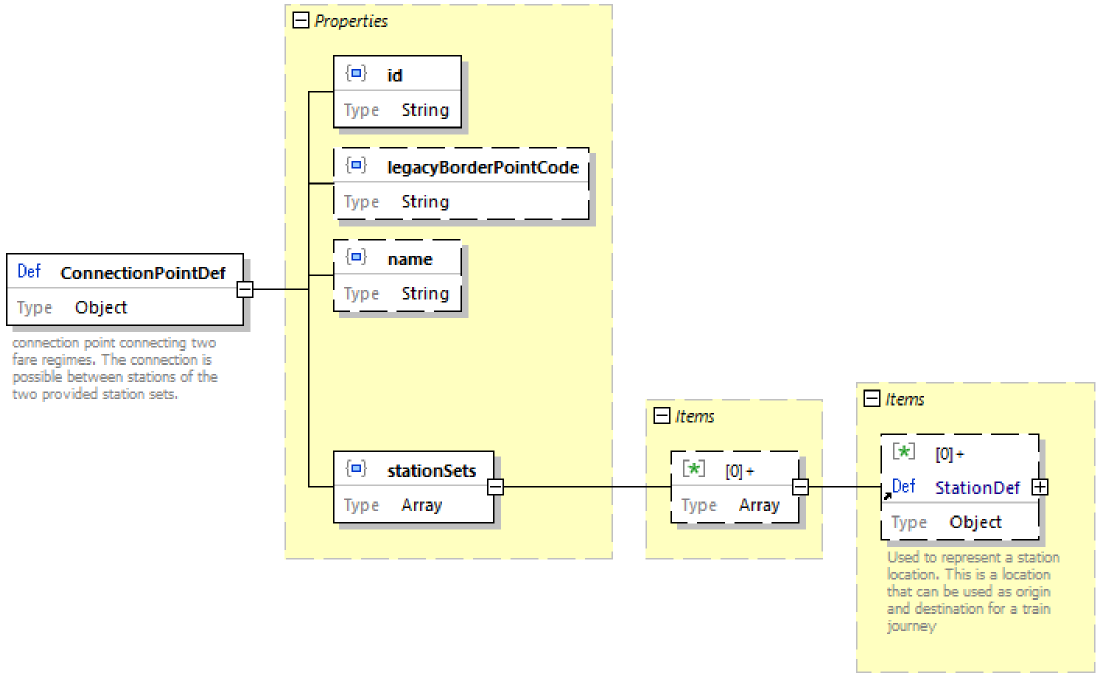
LOYALTY_CARD,REDUCTION_CARD,PASS

the card id must be delivered in online sales

Fare Structure

Connection Points

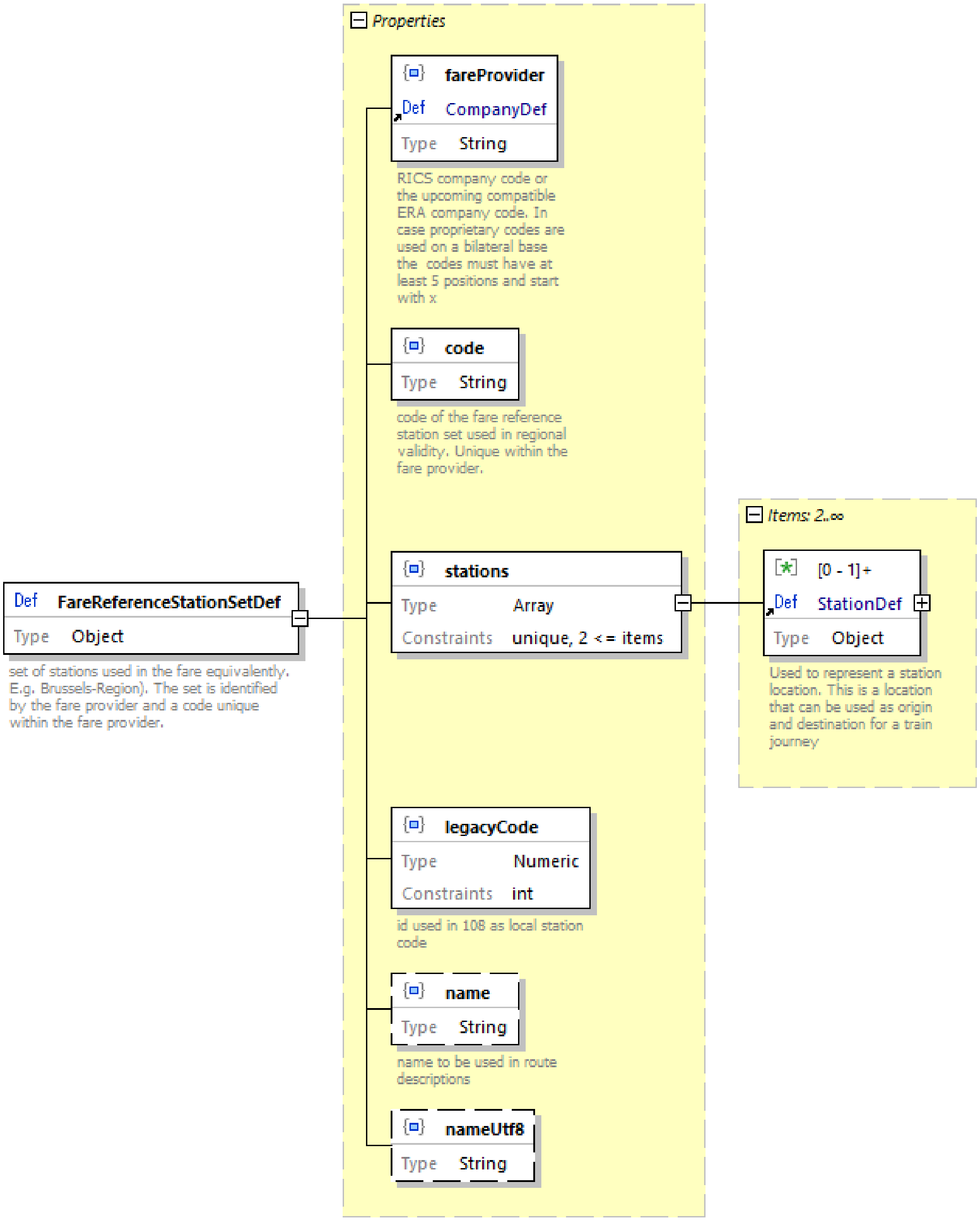
Connection points define the options to connect one fare with another fare a a point. In case the connecting point is a real station the connection point is defined by a set including just that station. In case the fares are connected between two stations the connection point includes two sets each including the station on one side. There light be cases where a connection is possible between more than two stations, in this rare case the set(s) might contain more than one station (e.g. Stations *A* and *B* for carrier 1 are connected to stations *C* and *D* of carrier 2 and allowed route go via *A*-*C* or *B*-*D*).


Properties
id
Type String
legacyBorderPointCode
Type String
name
Type String
Def ConnectionPointDef
Type Object
connection point connecting two fare regimes. The connection is possible between stations of the two provided station sets.
stationSets
Type Array
Items
[0]+
Type Array
Items
[0]+
Def StationDef
Type Object
Used to represent a station location. This is a location that can be used as origin and destination for a train journey

Fare Structure

fare reference station set

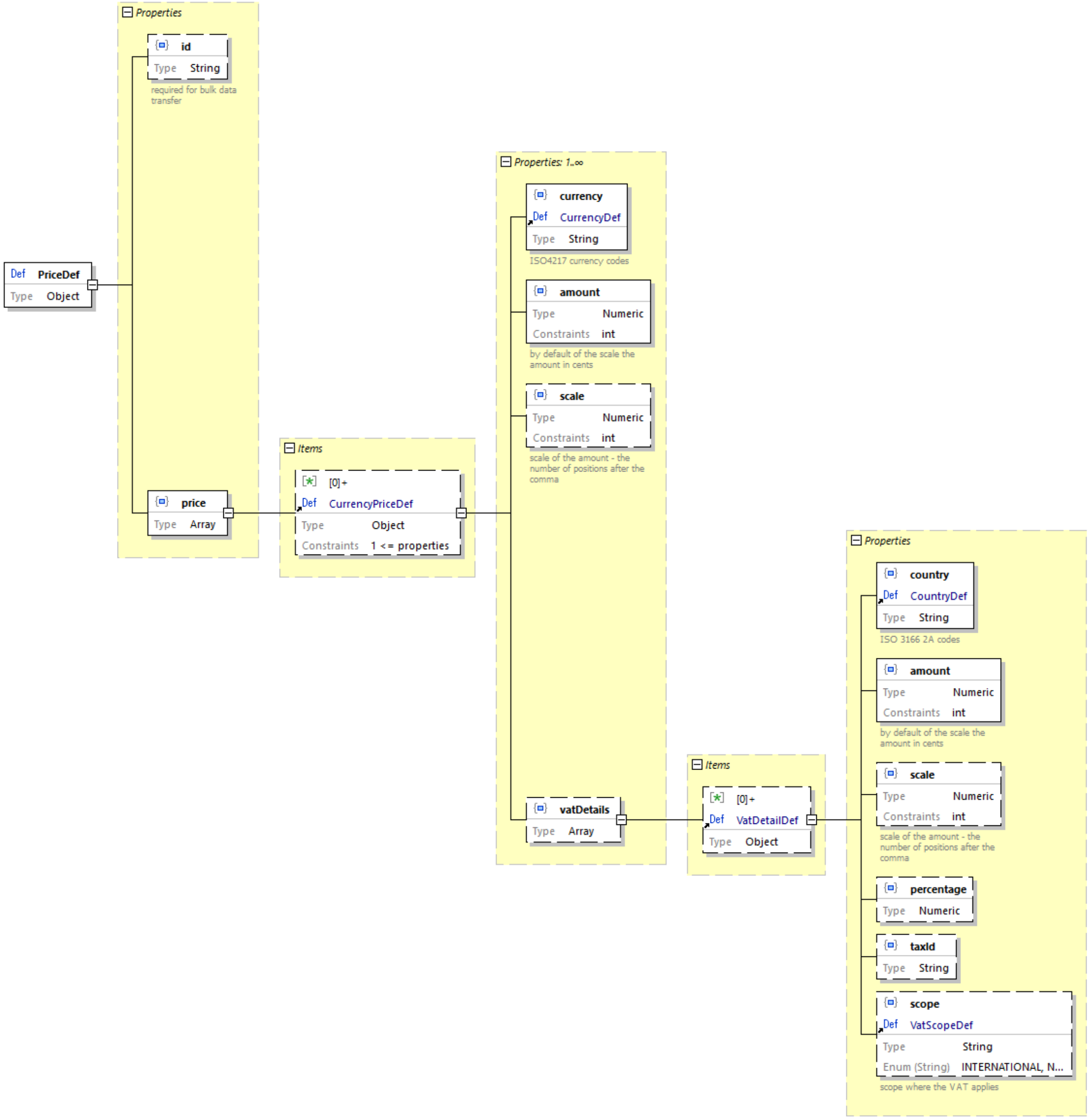
Fare reference station set defines a set of stations that can be used in a route. All station(s) of the set can be used by the passenger.


Properties
fareProvider
Def CompanyDef
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
code
Type String
code of the fare reference station set used in regional validity. Unique within the fare provider.
Def FareReferenceStationSetDef
Type Object
set of stations used in the fare equivalently. E.g. Brussels-Region). The set is identified by the fare provider and a code unique within the fare provider.
stations
Type Array
Constraints unique, 2 <= items
Items: 2..∞
[0 - 1]+
Def StationDef
Type Object
Used to represent a station location. This is a location that can be used as origin and destination for a train journey
legacyCode
Type Numeric
Constraints int
id used in 108 as local station code
name
Type String
name to be used in route descriptions
nameUtf8
Type String

Fare Structure

Price

The price of a fare of the refund fee on an after-sale. VAT details can be provided for the price. The default currency is EUR, but other currencies might be used based on bilateral agreements.

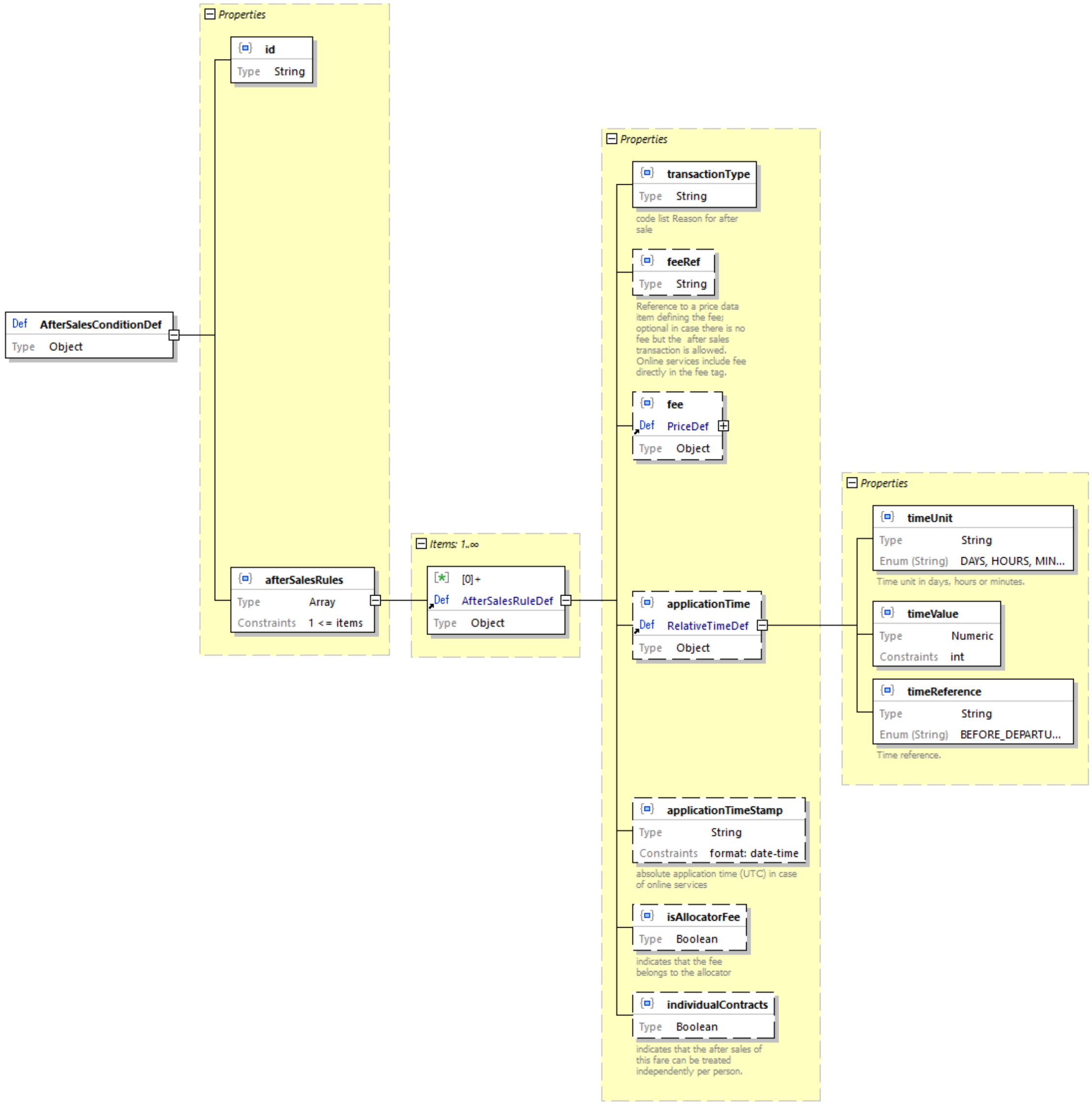


Fare Structure

Definitions of a Different Fare Constraints

Fare After Sales Constraint

Definition of the after sales conditions to be applied. Depending on the fare combination mode the after sales constraint can be omitted in case the distributor is responsible for the after sales fees.


Def AfterSalesConditionDef
Type Object
Properties
id
Type String
afterSalesRules
Type Array
Constraints 1 <= items
Items: 1..∞
[0]+
Def AfterSalesRuleDef
Type Object
Properties
transactionType
Type String
code list Reason for after sale
feeRef
Type String
Reference to a price data item defining the fee; optional in case there is no fee but the after sales transaction is allowed. Online services include fee directly in the fee tag.
fee
Def PriceDef
Type Object
applicationTime
Def RelativeTimeDef
Type Object
Properties
timeUnit
Type String
Enum (String) DAYS, HOURS, MIN...
Time unit in days, hours or minutes.
timeValue
Type Numeric
Constraints int
timeReference
Type String
Enum (String) BEFORE_DEPARTU...
Time reference.
applicationTimeStamp
Type String
Constraints format: date-time
absolute application time (UTC) in case of online services
isAllocatorFee
Type Boolean
indicates that the fee belongs to the allocator
individualContracts
Type Boolean
indicates that the after sales of this fare can be treated independently per person.

Fare Structure

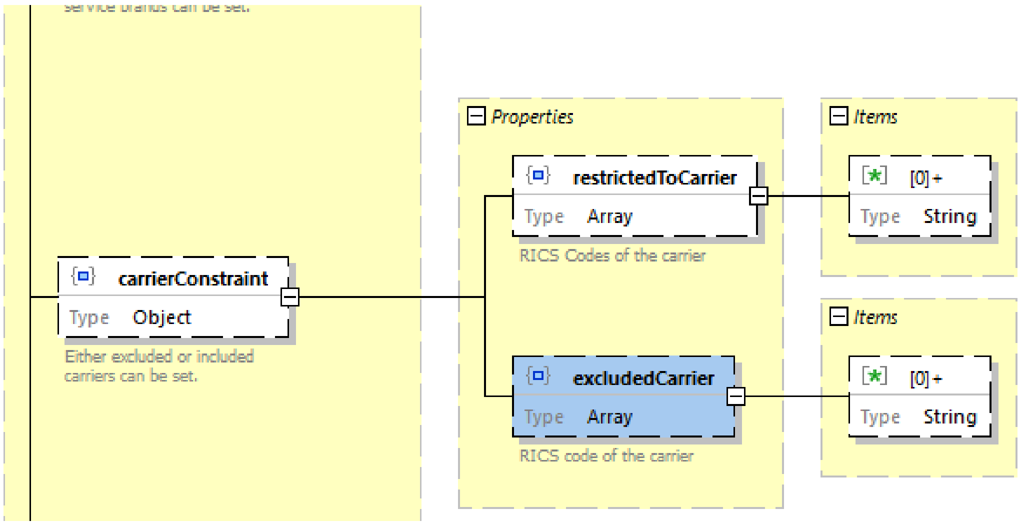
Fare Carrier Constraint

The carrier constraint defines the carriers that can be used. Either a list of the allowed carriers can be provided or a list on excluded carriers. In the case of excluded carriers all carriers not listed can be used.

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

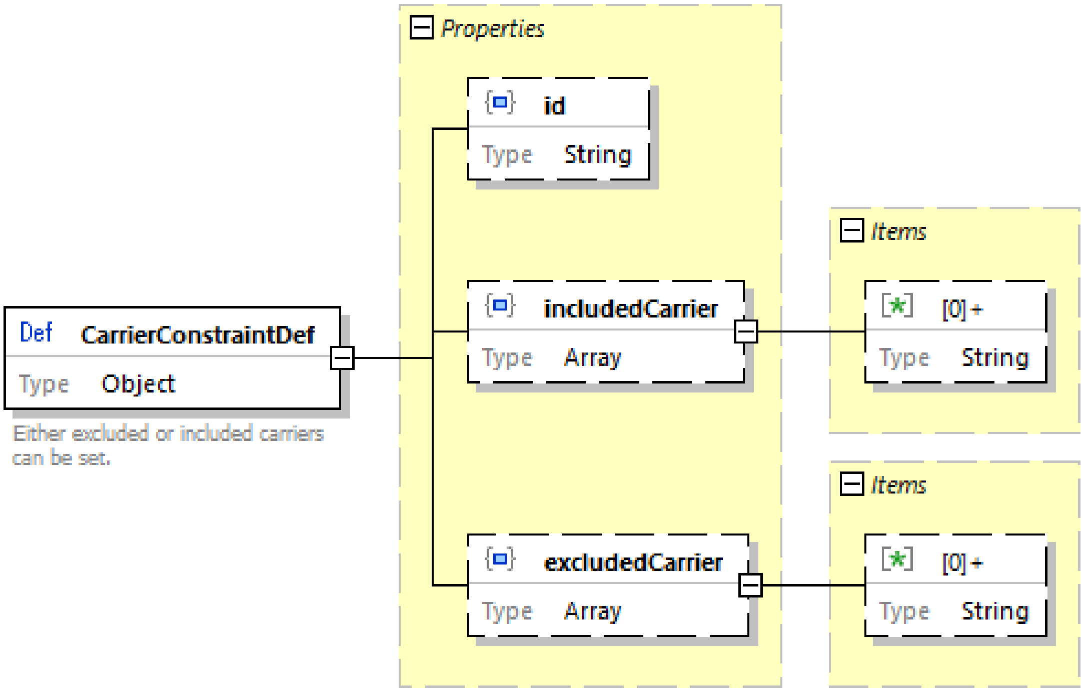
Carrier constraint limits an open fare - not linked to a train - to some carriers. The carriers can be specified either as exclusion list or alternatively as inclusion list.

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



Carrier Constraint Type

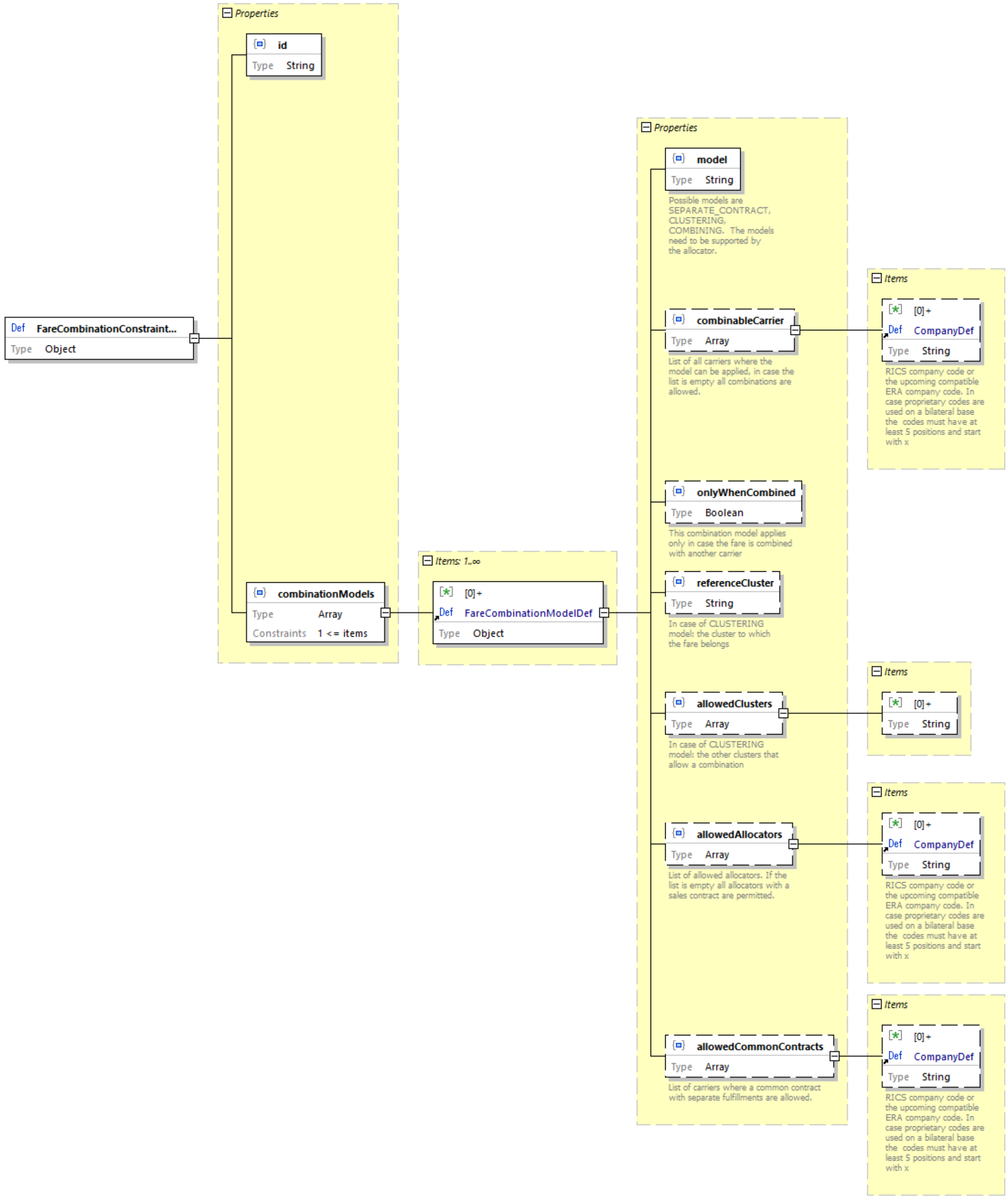
The included / excluded carriers are also part of the FCB barcode (*IRS 90918-4*) content and the ticket control data (*IRS 90918-9*).



Fare Structure

Fare Combination Constraint

Constraints ruling the possible combinations with other fares of other providers.



Possible models are SEPARATE_CONTRACT, CLUSTERING, COMBINING. The models need to be supported by the allocator.

List of all carriers where the model can be applied, in case the list is empty all combinations are allowed.

RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

This combination model applies only in case the fare is combined with another carrier

In case of CLUSTERING model: the cluster to which the fare belongs

In case of CLUSTERING model: the other clusters that allow a combination

List of allowed allocators. If the list is empty all allocators with a sales contract are permitted.

RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

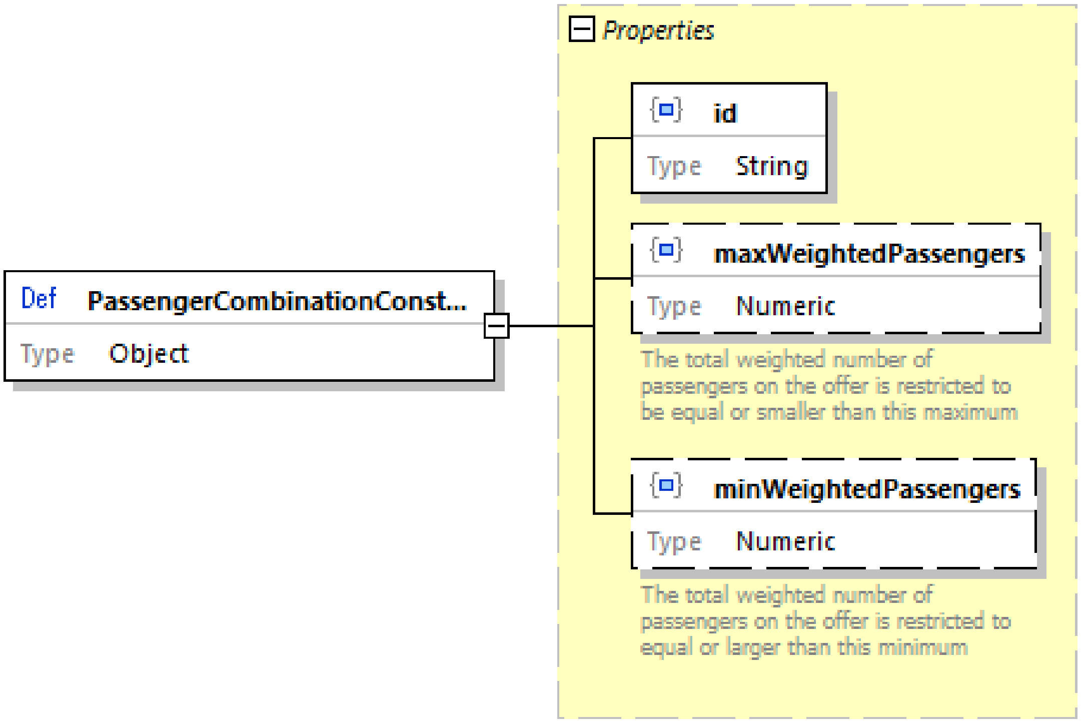
List of carriers where a common contract with separate fulfillments are allowed.

RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

Fare Structure

Fare Passenger Combination Constraint

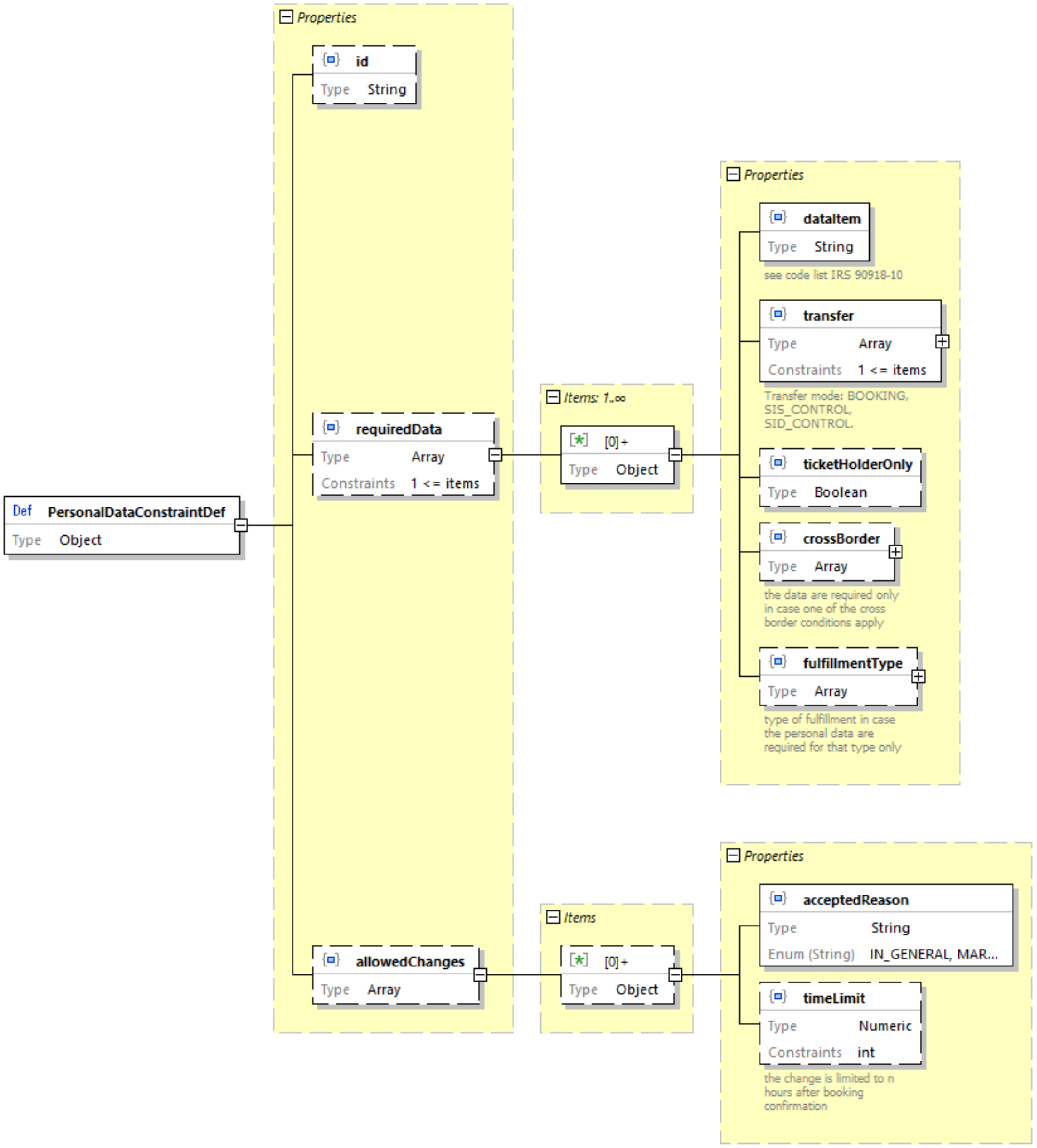
Constraints ruling the possible combinations of passengers for combination on a ticket.


Properties
Def PassengerCombinationConst...
Type Object
id
Type String
maxWeightedPassengers
Type Numeric
The total weighted number of passengers on the offer is restricted to be equal or smaller than this maximum
minWeightedPassengers
Type Numeric
The total weighted number of passengers on the offer is restricted to equal or larger than this minimum

Fare Structure

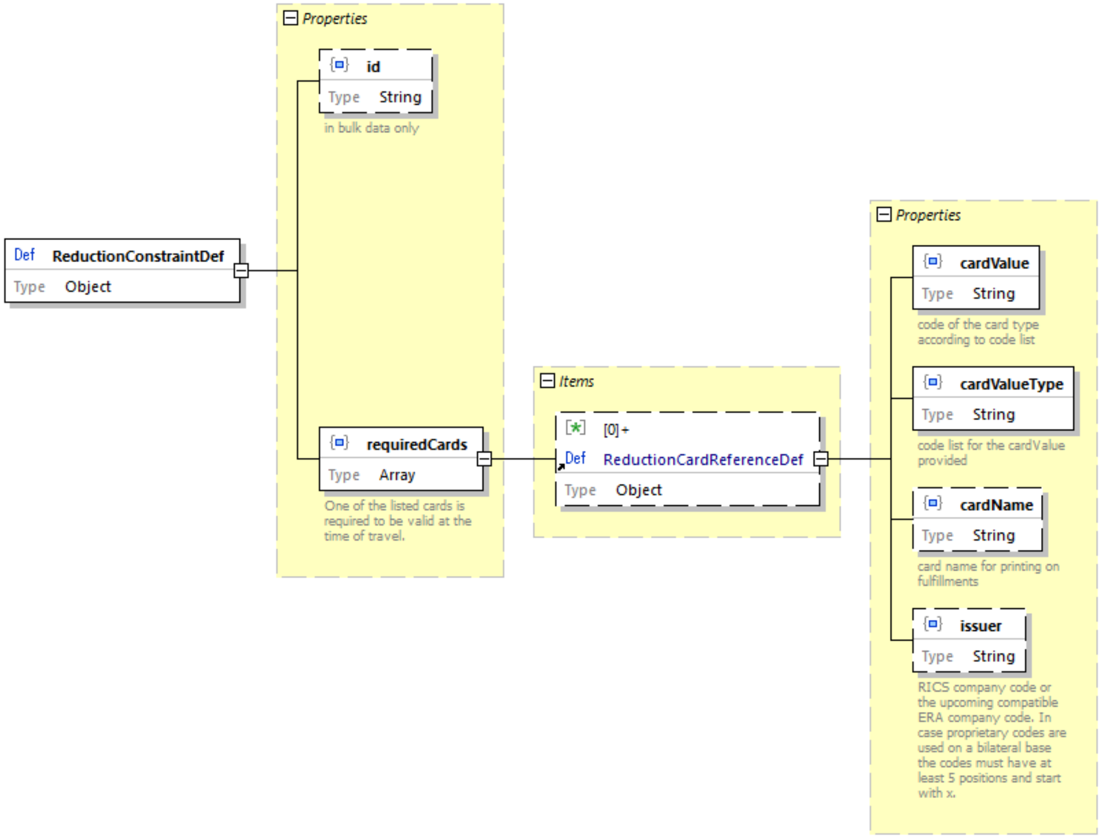
Fare Personal Data Constraint

Definition of the personal data required e.g. in a bar code of via online ticket control.



Fare Structure

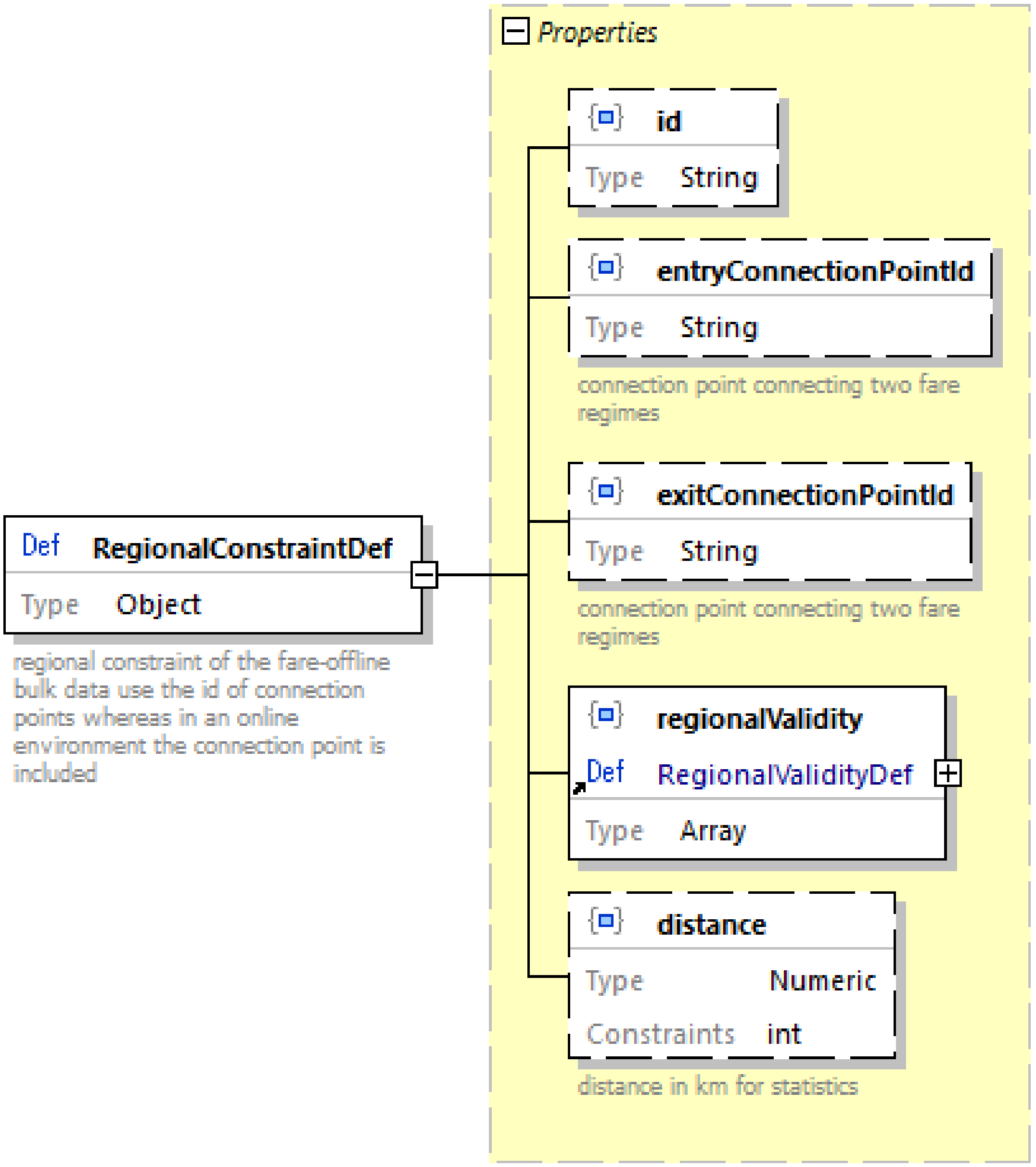
Fare Reduction Constraint


Def ReductionConstraintDef
Type Object
Properties
id
Type String
in bulk data only
requiredCards
Type Array
One of the listed cards is required to be valid at the time of travel.
Items
[0]+
Def ReductionCardReferenceDef
Type Object
Properties
cardValue
Type String
code of the card type according to code list
cardValueType
Type String
code list for the cardValue provided
cardName
Type String
card name for printing on fulfillments
issuer
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x.

Fare Structure

Fare Regional Constraint

The regional constraint defines the options to combine the fare at the start and end point via connection points.



regional constraint of the fare-offline bulk data use the id of connection points whereas in an online environment the connection point is included

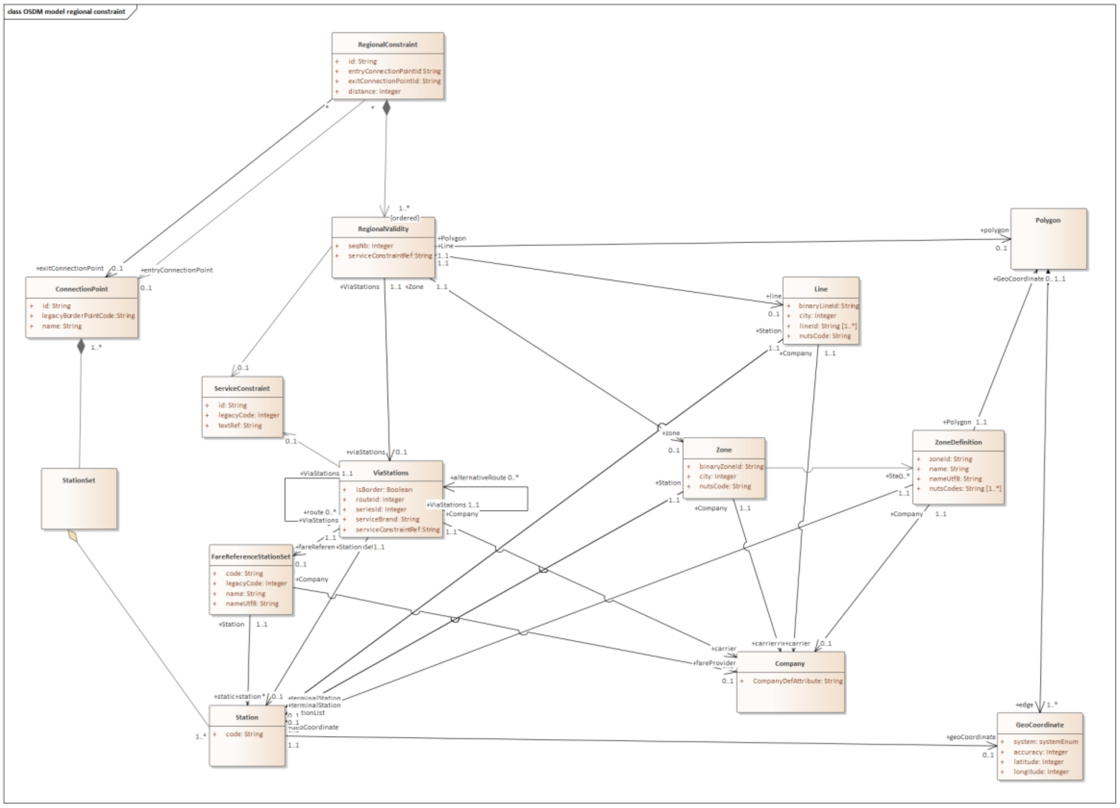
connection point connecting two fare regimes

connection point connecting two fare regimes

distance in km for statistics

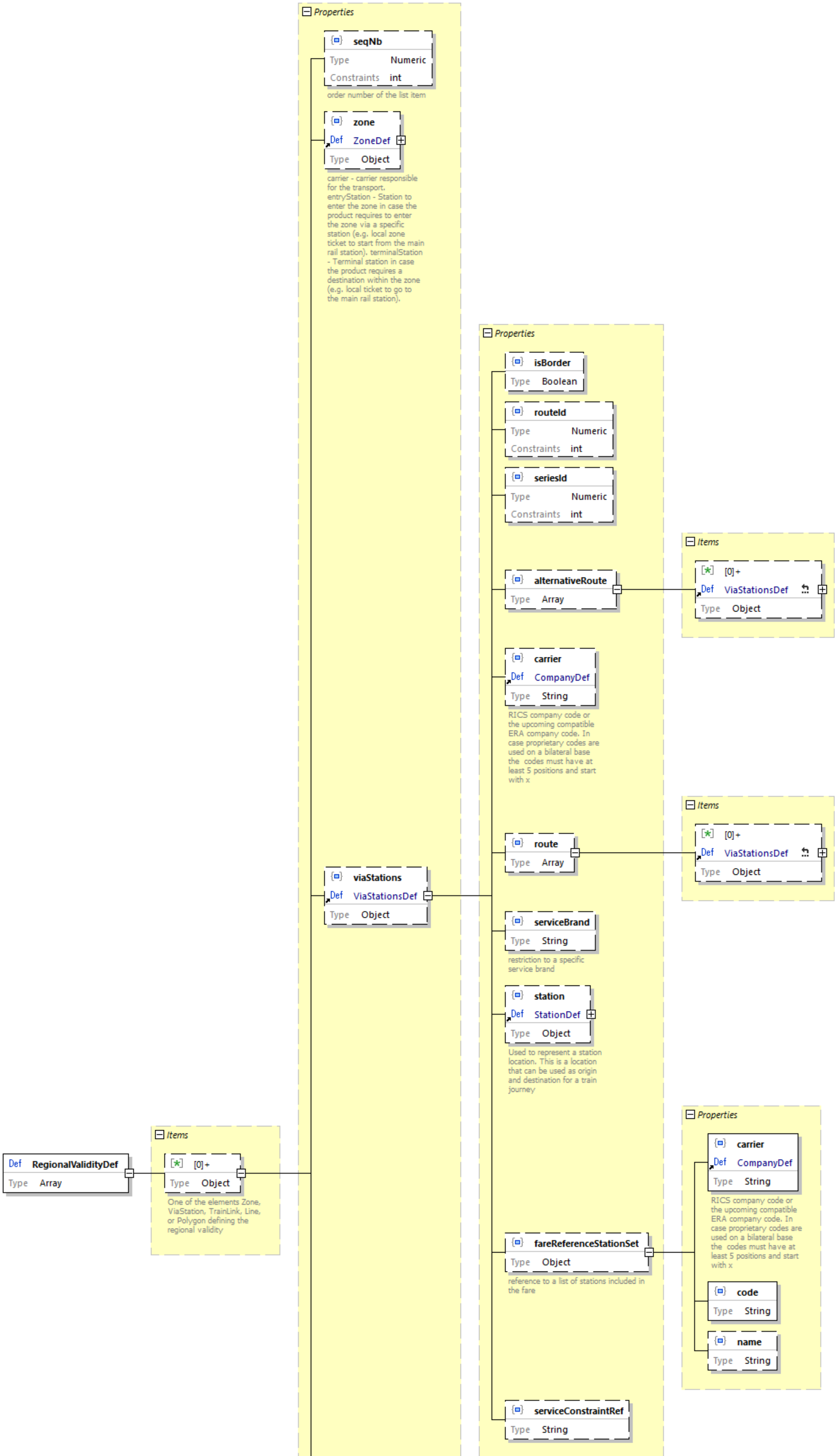
Fare Structure

Data model regional constraints:



Fare Structure

The regional validity defines the geographical validity of the fare. It is defined as an ordered list along the route. Options to define a part of the route include Zones (regional areas), Lines, route descriptions (viaStations) that can define sequences of stations, alternative routes and fare reference sets. Areas can also be defined by geo-coordinates.



One of the elements Zone, ViaStation, TrainLink, Line, or Polygon defining the regional validity

order number of the list item

carrier - carrier responsible for the transport. entryStation - Station to enter the zone in case the product requires to enter the zone via a specific station (e.g. local zone ticket to start from the main rail station). terminalStation - Terminal station in case the product requires a destination within the zone (e.g. local ticket to go to the main rail station).

RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

restriction to a specific service brand

Used to represent a station location. This is a location that can be used as origin and destination for a train journey

reference to a list of stations included in the fare

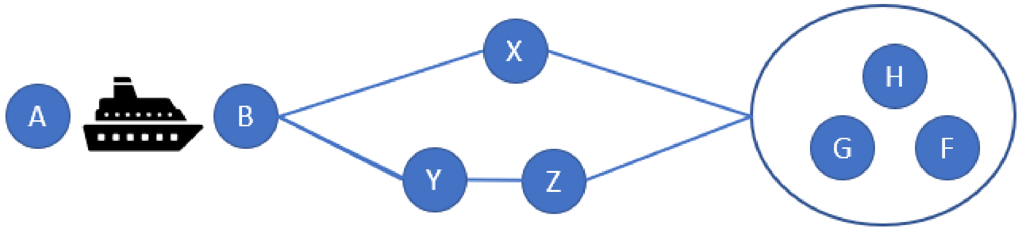
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x

Fare Structure

Route Description example in the regional validity

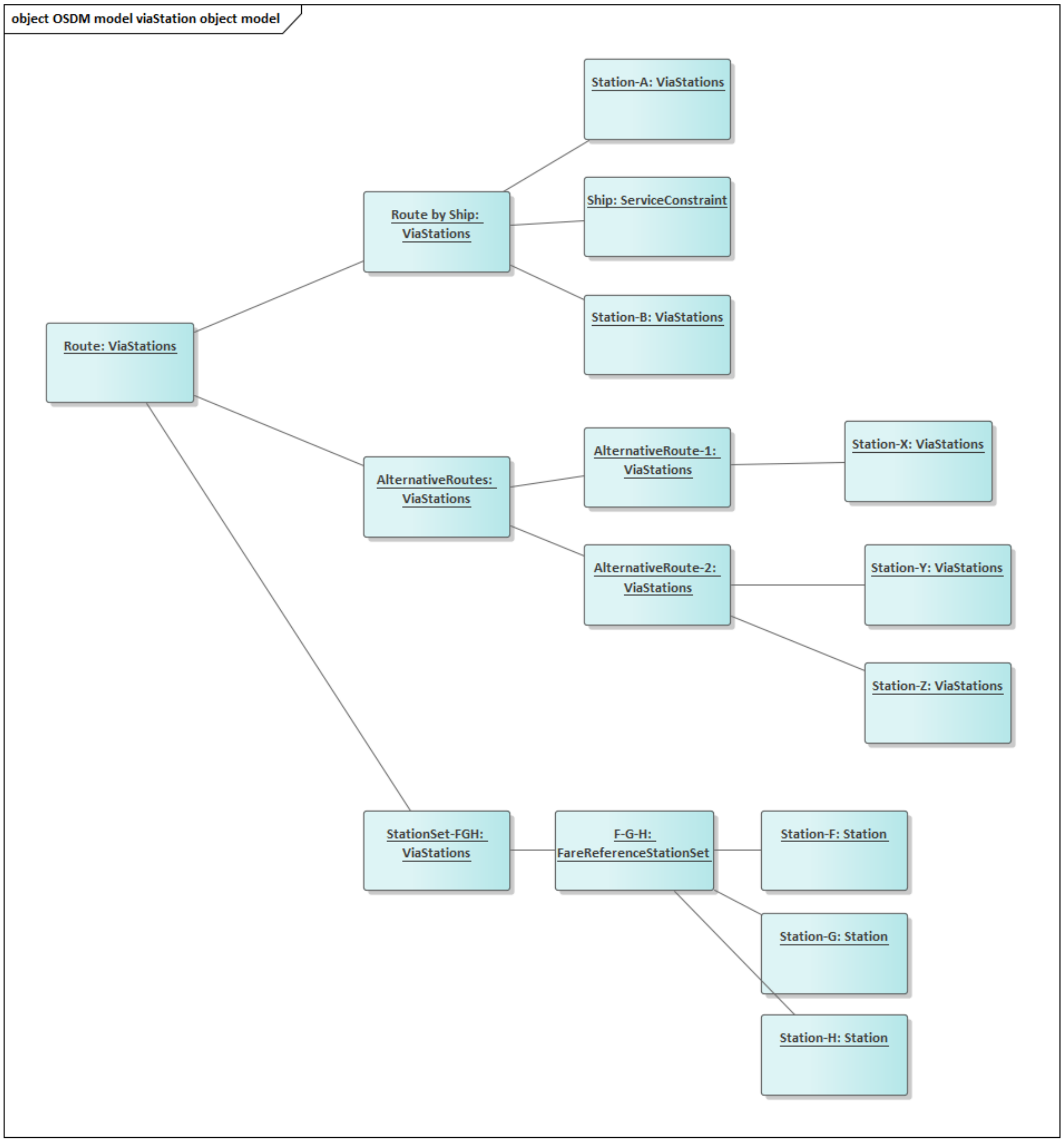
Route example:

A*Ship*B*(X/Y*Z)*FGH-Region



Fare Structure

Object model:


object OSDM model viaStation object model
Station-A: ViaStations
Route by Ship: ViaStations
Ship: ServiceConstraint
Station-B: ViaStations
Route: ViaStations
AlternativeRoute-1: ViaStations
Station-X: ViaStations
AlternativeRoutes: ViaStations
AlternativeRoute-2: ViaStations
Station-Y: ViaStations
Station-Z: ViaStations
StationSet-FGH: ViaStations
F-G-H: FareReferenceStationSet
Station-F: Station
Station-G: Station
Station-H: Station

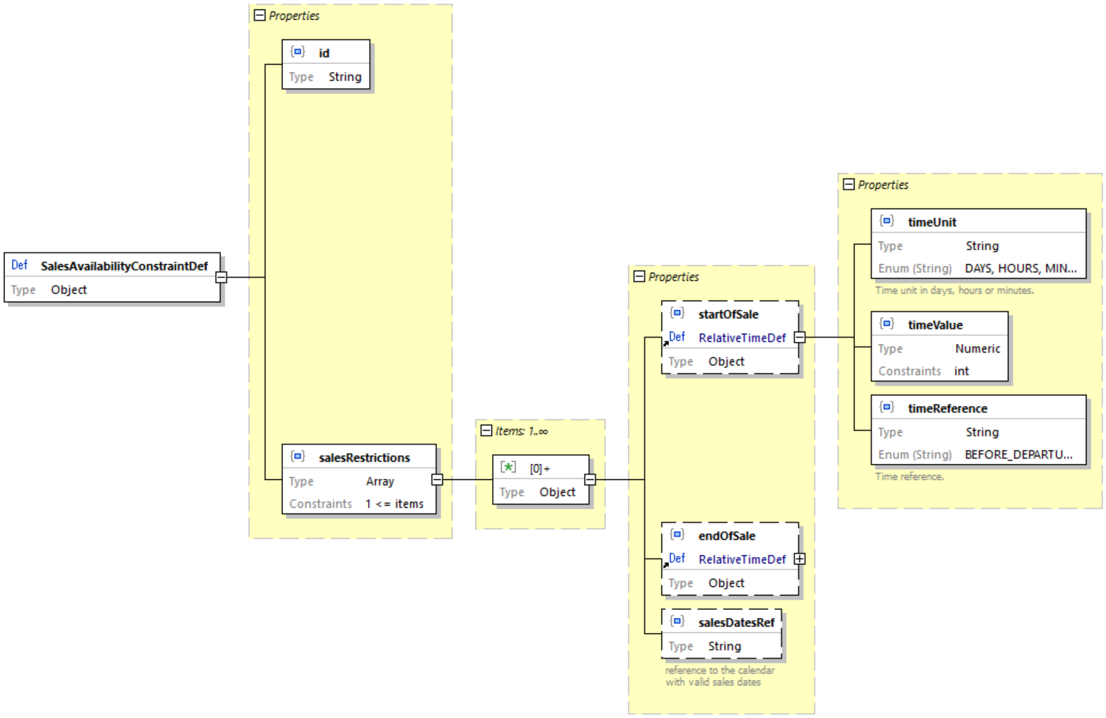
Fare Structure

Fare Sales Availability Constraint

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 provides as a list of `salesRestrictions` that have to be applied.

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.



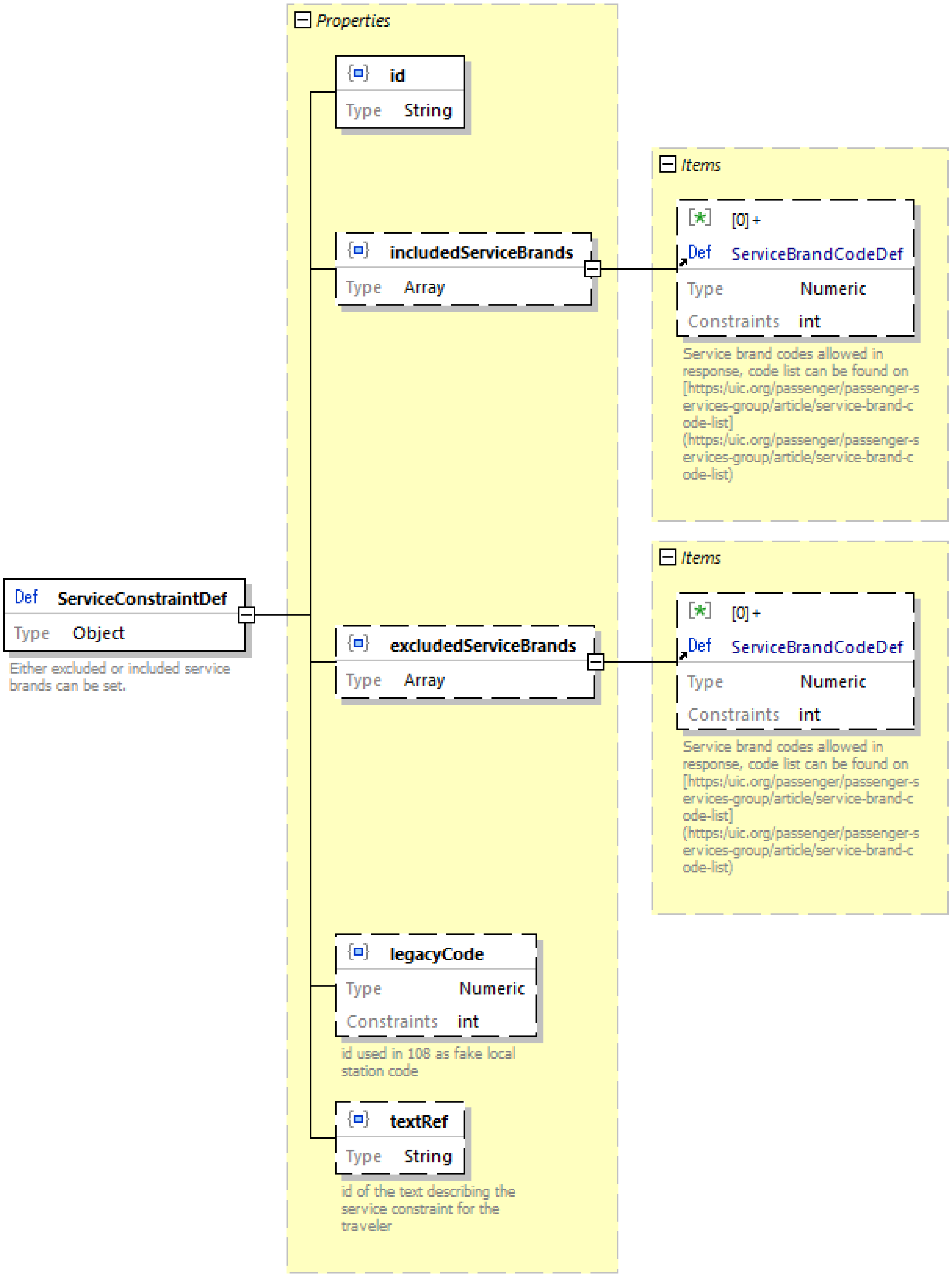
Fare Structure

Data Constraint on SalesAvailability

Code	Description
`startOfSale`, `endOfSale`	`startOfSale` < `endOfSale`

Fare Service Constraint

The service constraint defines restrictions to specific service brands. Either a list of service brands of a list of excluded service brands can be provided.


Properties
id
Type String
includedServiceBrands
Type Array
Items
[0]+
Def ServiceBrandCodeDef
Type Numeric
Constraints int
Service brand codes allowed in response, code list can be found on https:/uic.org/passenger/passenger-services-group/article/service-brand-code-list
Def ServiceConstraintDef
Type Object
Either excluded or included service brands can be set.
excludedServiceBrands
Type Array
Items
[0]+
Def ServiceBrandCodeDef
Type Numeric
Constraints int
Service brand codes allowed in response, code list can be found on https:/uic.org/passenger/passenger-services-group/article/service-brand-code-list
legacyCode
Type Numeric
Constraints int
id used in 108 as fake local station code
textRef
Type String
id of the text describing the service constraint for the traveler

Fare Structure

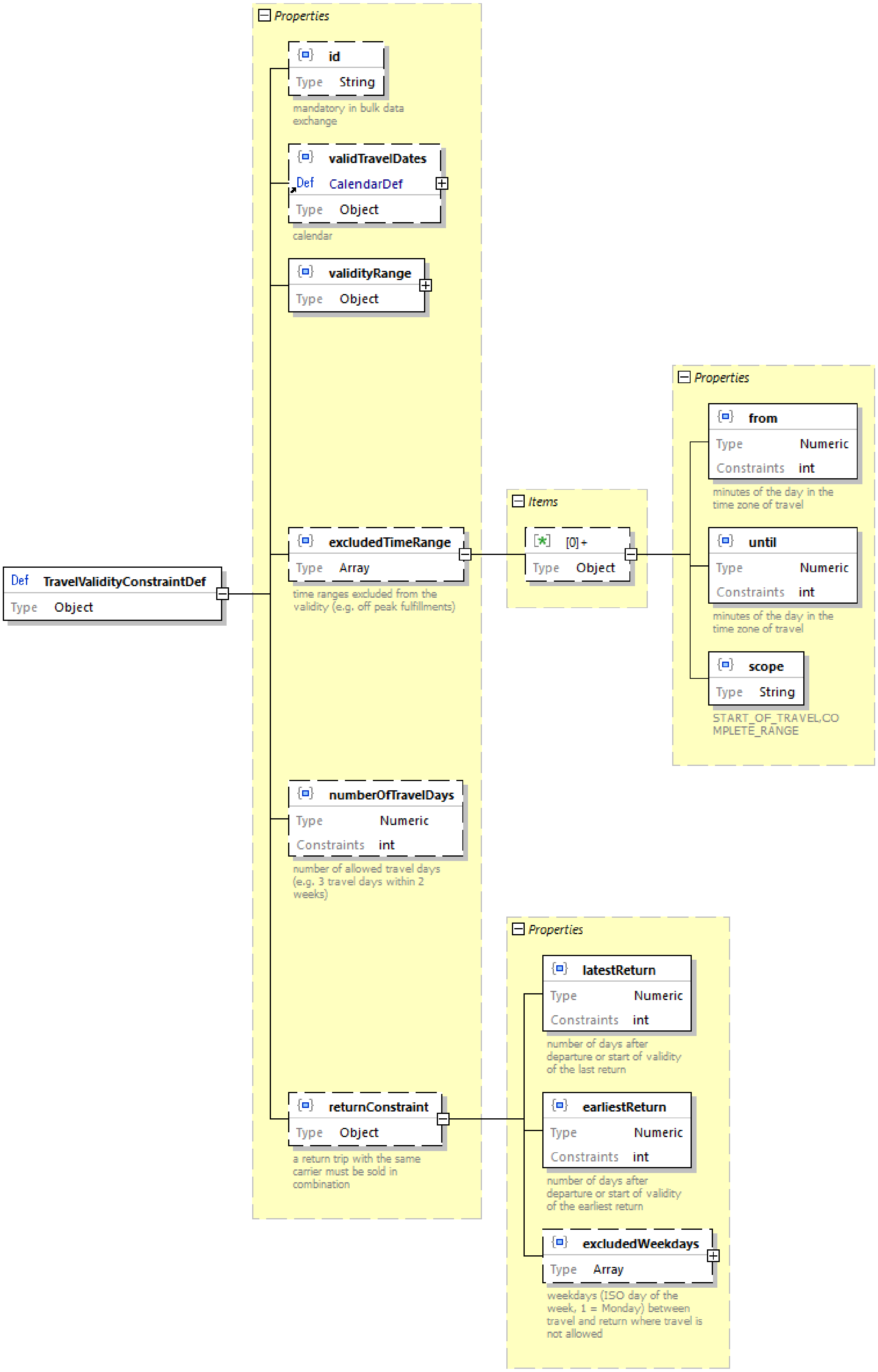
Data Constraint on ServiceConstraint

Code	Description
`includedServiceBrands,`	Only one of the lists can be used. Using both lists is

Code	Description
`excludedServiceBrands`	forbidden.

Fare Travel Validity Constraint

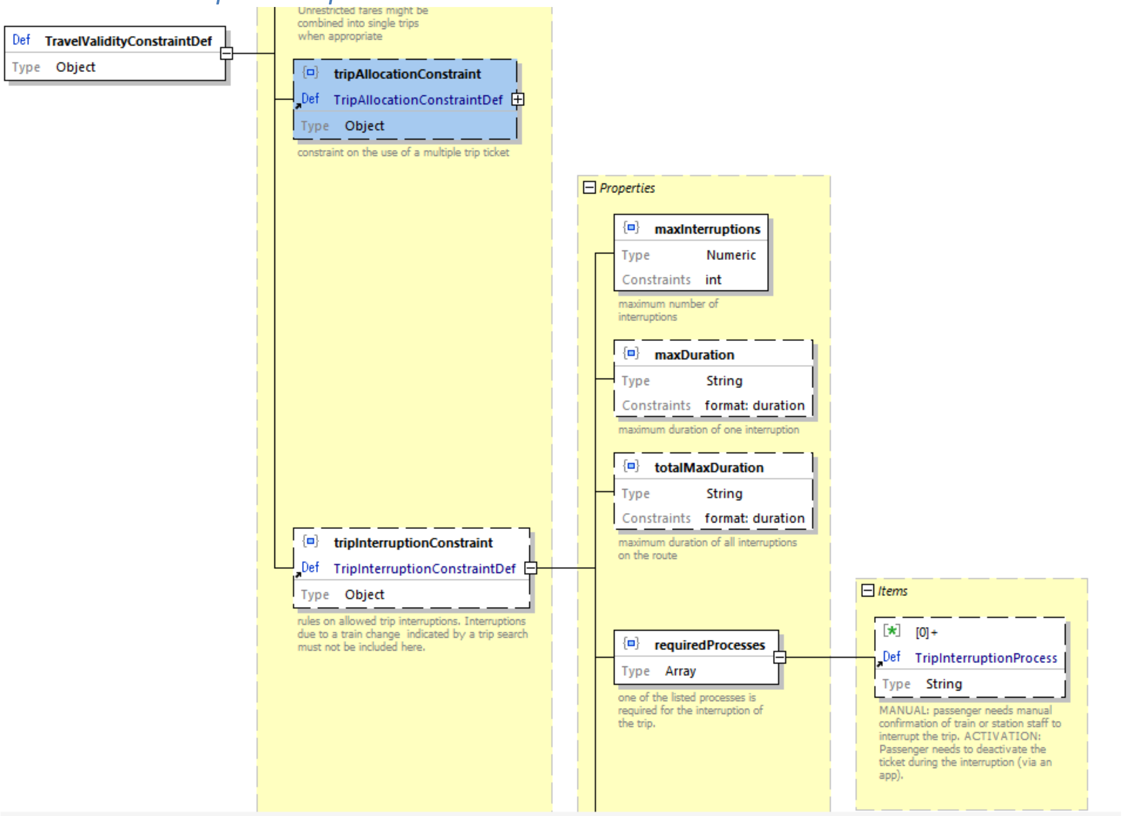
The travel validity defines the duration the passenger has to make his travel. Optional time slots (e.g. peak hours) can be excluded.


Def TravelValidityConstraintDef
Type Object
Properties
id
Type String
mandatory in bulk data exchange
validTravelDates
Def CalendarDef
Type Object
calendar
validityRange
Type Object
excludedTimeRange
Type Array
time ranges excluded from the validity (e.g. off peak fulfillments)
Items
[0] +
Type Object
Properties
from
Type Numeric
Constraints int
minutes of the day in the time zone of travel
until
Type Numeric
Constraints int
minutes of the day in the time zone of travel
scope
Type String
START_OF_TRAVEL,COMPLETE_RANGE
numberOfTravelDays
Type Numeric
Constraints int
number of allowed travel days (e.g. 3 travel days within 2 weeks)
returnConstraint
Type Object
a return trip with the same carrier must be sold in combination
Properties
latestReturn
Type Numeric
Constraints int
number of days after departure or start of validity of the last return
earliestReturn
Type Numeric
Constraints int
number of days after departure or start of validity of the earliest return
excludedWeekdays
Type Array
weekdays (ISO day of the week, 1 = Monday) between travel and return where travel is not allowed

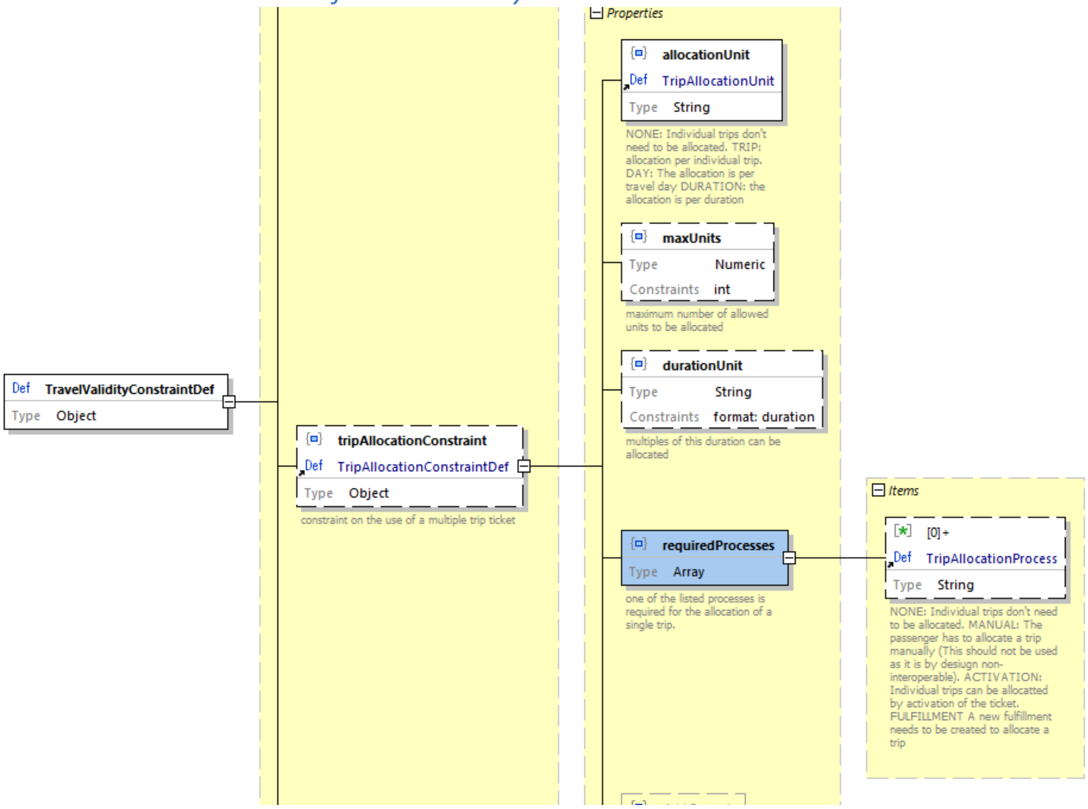
Data Constraint on TravelValidity

Code	Description
`excludedTimeRange`	`from` time < `until` time
`numberOfTravelDays`	A duration must be provided
`returnConstraint`	`earliestReturn` < `latestReturn`

Constraint on Trip Interruptions


Def TravelValidityConstraintDef
Type Object
Unrestricted fares might be combined into single trips when appropriate
tripAllocationConstraint
Def TripAllocationConstraintDef
Type Object
constraint on the use of a multiple trip ticket
tripInterruptionConstraint
Def TripInterruptionConstraintDef
Type Object
rules on allowed trip interruptions. Interruptions due to a train change indicated by a trip search must not be included here.
Properties
maxInterruptions
Type Numeric
Constraints int
maximum number of interruptions
maxDuration
Type String
Constraints format: duration
maximum duration of one interruption
totalMaxDuration
Type String
Constraints format: duration
maximum duration of all interruptions on the route
requiredProcesses
Type Array
one of the listed processes is required for the interruption of the trip.
Items
[0]+
Def TripInterruptionProcess
Type String
MANUAL: passenger needs manual confirmation of train or station staff to interrupt the trip. ACTIVATION: Passenger needs to deactivate the ticket during the interruption (via an app).

Constraint on Activation of Multi Journey Fares


Def TravelValidityConstraintDef
Type Object
tripAllocationConstraint
Def TripAllocationConstraintDef
Type Object
constraint on the use of a multiple trip ticket
Properties
allocationUnit
Def TripAllocationUnit
Type String
NONE: Individual trips don't need to be allocated. TRIP: allocation per individual trip. DAY: The allocation is per travel day DURATION: the allocation is per duration
maxUnits
Type Numeric
Constraints int
maximum number of allowed units to be allocated
durationUnit
Type String
Constraints format: duration
multiples of this duration can be allocated
requiredProcesses
Type Array
one of the listed processes is required for the allocation of a single trip.
Items
[0]+
Def TripAllocationProcess
Type String
NONE: Individual trips don't need to be allocated. MANUAL: The passenger has to allocate a trip manually (This should not be used as it is by desiugn non-interoperable). ACTIVATION: Individual trips can be allocatted by activation of the ticket. FULFILLMENT A new fulfillment needs to be created to allocate a trip

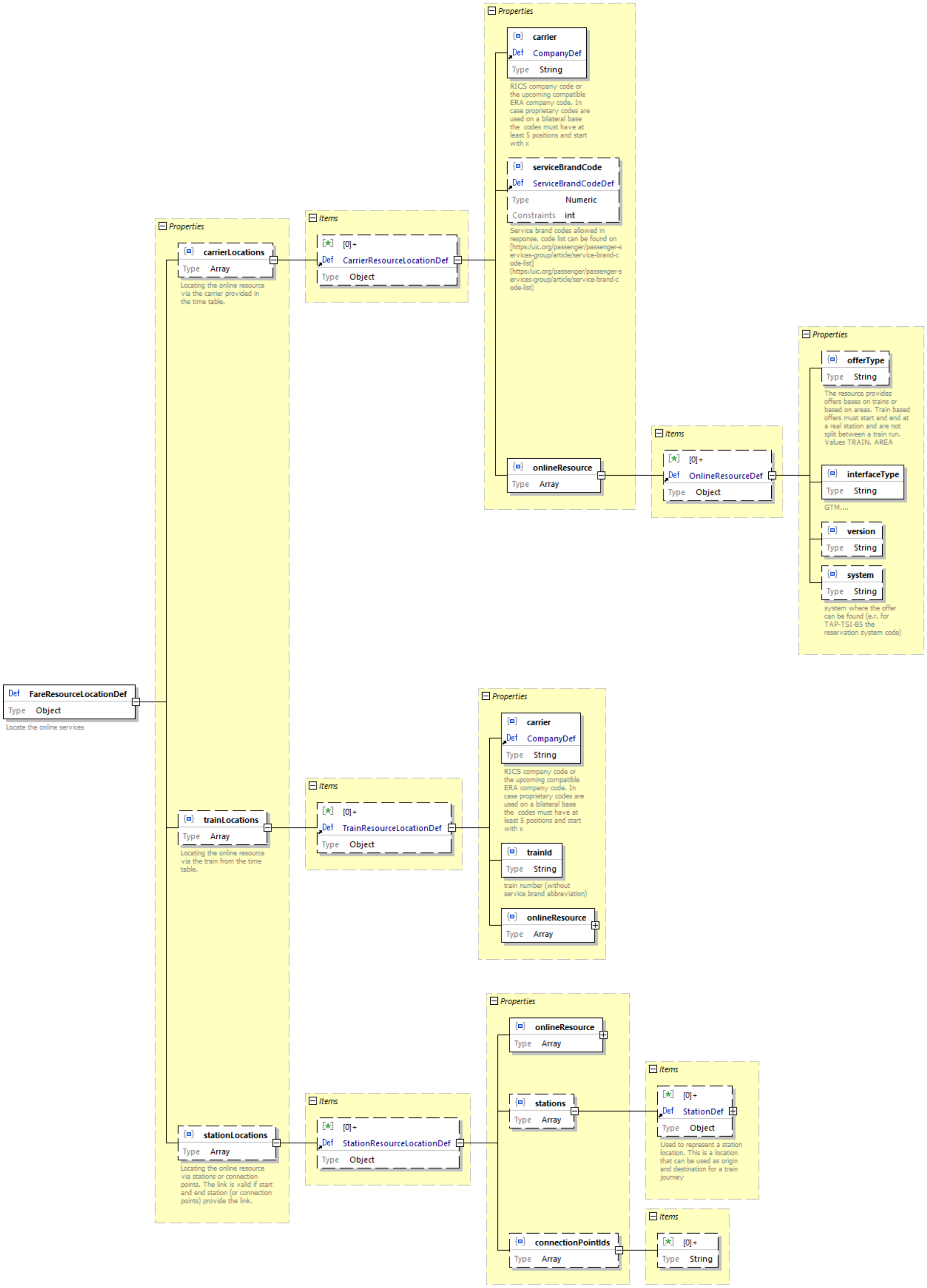
Fare Structure

Data Supporting Online Services

Additional data to support the online sales services can be exchanged.

Fare Resource Places

Information on where to fine resources for online services of OSDM online or via 90918-1 APIs


Def FareResourceLocationDef
Type Object
Locate the online services
Properties
carrierLocations
Type Array
Locating the online resource via the carrier provided in the time table.
Items
[0]+
Def CarrierResourceLocationDef
Type Object
Properties
carrier
Def CompanyDef
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
serviceBrandCode
Def ServiceBrandCodeDef
Type Numeric
Constraints int
Service brand codes allowed in response, code list can be found on [https:/uic.org/passenger/passenger-services-group/article/service-brand-code-list] (https:/uic.org/passenger/passenger-services-group/article/service-brand-code-list)
onlineResource
Type Array
Items
[0]+
Def OnlineResourceDef
Type Object
Properties
offerType
Type String
The resource provides offers bases on trains or based on areas. Train based offers must start and end at a real station and are not split between a train run. Values TRAIN, AREA
interfaceType
Type String
GTM,...
version
Type String
system
Type String
system where the offer can be found (e.r. for TAP-TSI-B5 the reservation system code)
trainLocations
Type Array
Locating the online resource via the train from the time table.
Items
[0]+
Def TrainResourceLocationDef
Type Object
Properties
carrier
Def CompanyDef
Type String
RICS company code or the upcoming compatible ERA company code. In case proprietary codes are used on a bilateral base the codes must have at least 5 positions and start with x
trainId
Type String
train number (without service brand abbreviation)
onlineResource
Type Array
stationLocations
Type Array
Locating the online resource via stations or connection points. The link is valid if start and end station (or connection points) provide the link.
Items
[0]+
Def StationResourceLocationDef
Type Object
Properties
onlineResource
Type Array
stations
Type Array
Items
[0]+
Def StationDef
Type Object
Used to represent a station location. This is a location that can be used as origin and destination for a train journey
connectionPointIds
Type Array
Items
[0]+
Type String

Fare Reservation Parameters

Data to define reservation requirements related to a fare.

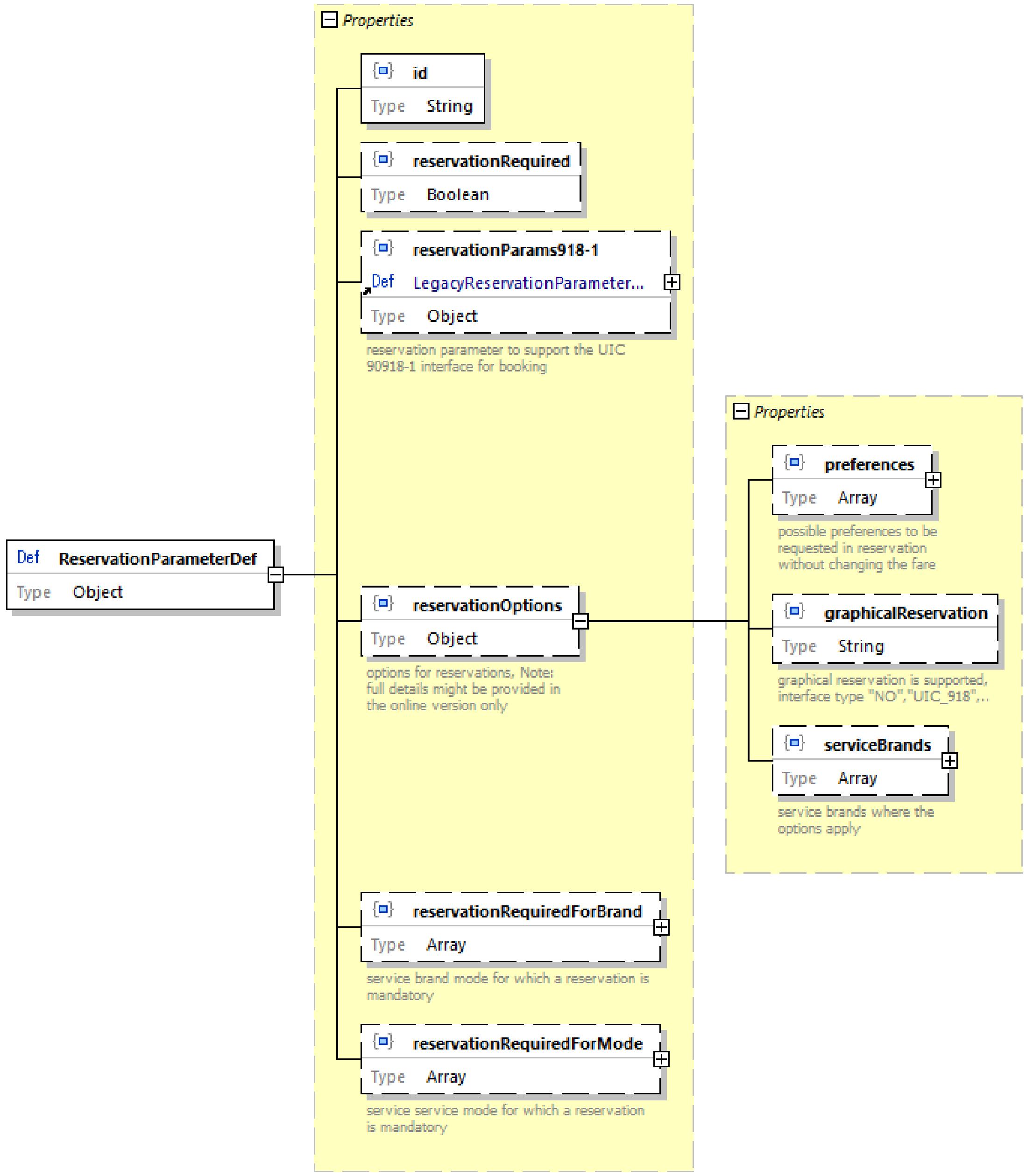

Properties
id
Type String
reservationRequired
Type Boolean
reservationParams918-1
Def LegacyReservationParameter...
Type Object
reservation parameter to support the UIC 90918-1 interface for booking
Def ReservationParameterDef
Type Object
reservationOptions
Type Object
options for reservations, Note: full details might be provided in the online version only
Properties
preferences
Type Array
possible preferences to be requested in reservation without changing the fare
graphicalReservation
Type String
graphical reservation is supported, interface type "NO","UIC_918",...
serviceBrands
Type Array
service brands where the options apply
reservationRequiredForBrand
Type Array
service brand mode for which a reservation is mandatory
reservationRequiredForMode
Type Array
service service mode for which a reservation is mandatory

Table of contents

2. Trips and Places
3. Complaint
4. Reimbursement
5. Release
6. Putting Prebooking on Hold
7. State Models
8. Ids and References

Online Sales and Distribution

Roles: Fare Provider, Distributor & Retailer

A participant can play the following roles:

- **Fare Provider** – defining the fare and combination rules and providing them to distributors offline or online.
- **Distributor** – combining fares, defining after sales rules within the frames et 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.
- **Retailer** – selling tickets from one or more distributors to the customer. Selecting the distributors 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 distributor 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 distributor. Data supporting the selection of the carriers/fare providers to be requested are part of this specification and can be transferred beforehand.

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 change 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 distributor 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 distributor to the carrier. This is the trigger for the accounting processes between the distributor and the carrier.

Fulfillment

For combined tickets the distributor 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 distributor.

The distributor 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 distributor.

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 distributor 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 distributor and carrier.

In case of a delayed payment of refunds the distributor 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 distributor 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 distributor to the carrier the customer can ask at the issuer or 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.

Distributor Role

Distribution Scenario – Sales

Timetable search

The distributor makes a time table search himself or retrieves the trip from a retailer. The distributor must 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 distributor. Data supporting the selection of the carriers/fare providers to be requested are part of this specification and can be transferred beforehand.

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 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 fare for a period defined by the carrier. Technically it is implemented as a post /booking using the offerId. The distributor 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 distributor 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 distributor to the carrier. This is the trigger for the accounting processes between the distributor and the carrier.

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

For combined tickets the distributor 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 distributor.

The distributor 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 distributor 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 distributor 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 distributor and carrier. The distributor hast to pass on information on delayed payment to the retailer.

In case of a delayed payment of refunds the distributor 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 distributor 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 distributor to the carrier the customer can ask at the issuer or 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.

Retailer Role

Distribution Scenario – sales

Timetable search

The retailer makes a timetable search himself or retrieves the trip from a distributor.

Offer

Requesting offers of a distributor 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 distributor delivers a changed trip the retailer 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 distributor. Technically it is implemented as a post /booking using the offerId. The distributor is responsible to manage the transactions including multiple carriers/fare providers.

Confirm Booking

The booking is confirmed by the retailer to the distributor.

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

The retailer 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 retailer requests fulfillments from the distributor and hands them to the passenger.

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 distributor is not involved in this step.

After sales scenario – cancellation

Refund Offer

The retailer needs to get a refund offer from the distributor 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 distributor.

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 retailer needs to obey the delayed payment instruction provided by the distributor.

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 retailer needs to obey the delayed payment instruction provided by the distributor.

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

To retrieve the stored personal data the booking must be requested.

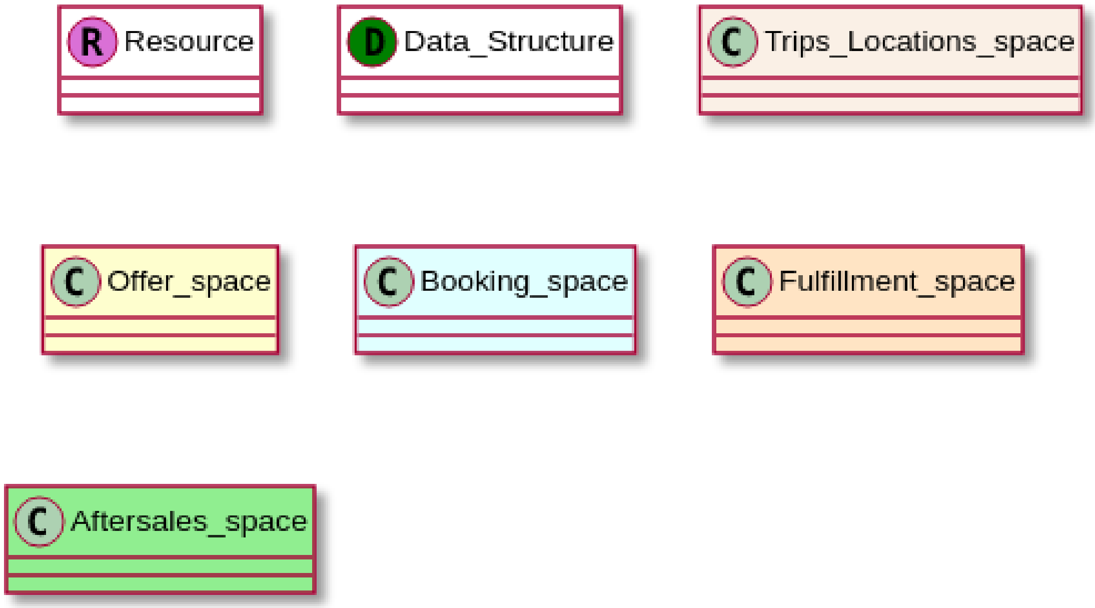
Data Structure – Online Structure

Introduction

This page shows a representation of 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.

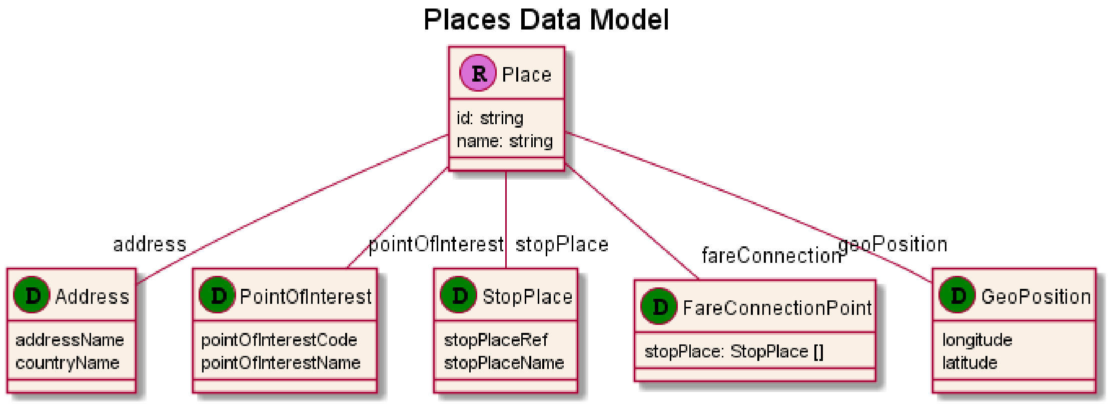
Legend



Legend

Trips and Places

Places Data Model



Place Data Model

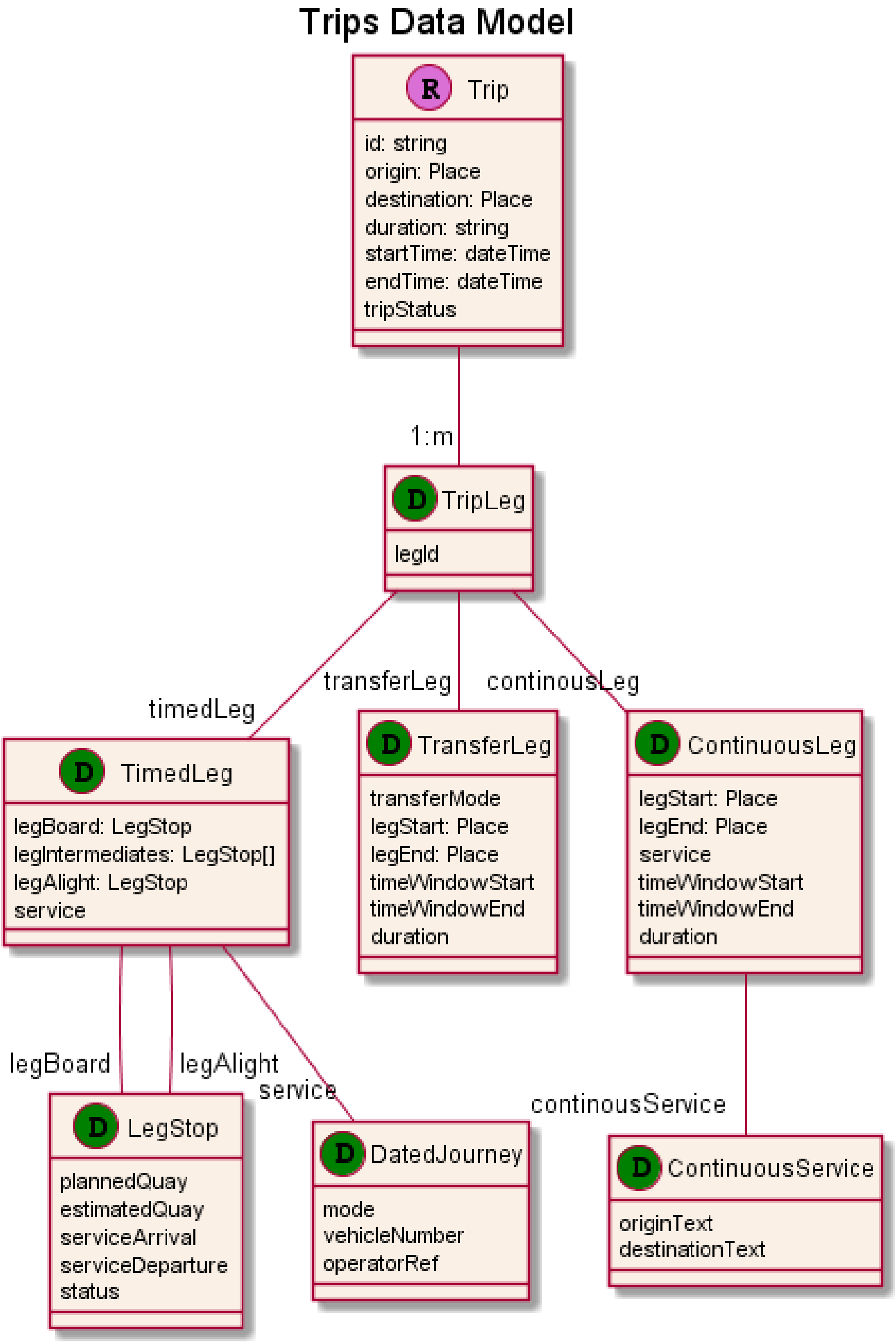
Places

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

- **Address**: any street address can be represented here. Is included to foresee extension towards multi modality and first/last miles solutions
- **PointOfInterest**: used to represent a specific point of interest
- **StopPlace**: represent a place where a train or a bus stops. It is obviously the most relevant type for OSDM. StopPlaces 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 and mandatory for train stations.
- **GeoCoordinate**: allows providing any location on the globe using its geographical coordinates.
- **FareConnectionPoint**: allows to model virtual border points by defining stations within the connection point lies.

Places 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.

Trip Data Model



Trip Data Model

Trips

Trips represent the concrete realization of a trip going from departure station to destination station.

- **origin**: departure location of the tripLeg
- **destination**: arrival location of the tripLeg

A trip is composed of one or more `tripLegs` and can be of one the following type:

- **TimedLeg**: A type of leg with a timetable schedule such a provided by public transport
- **TransferLeg**: A type of leg that links two legs such as walking from one stop to another
- **ContinousLeg**: A type of leg that is not bound to a timetable. This leg is mainly aimed at new modes such as scooter, taxis,..

Each `tripLeg` (also sometimes called leg) represents a connection between two places where the traveller will either step in a transport or step out of a transport (most likely a train).

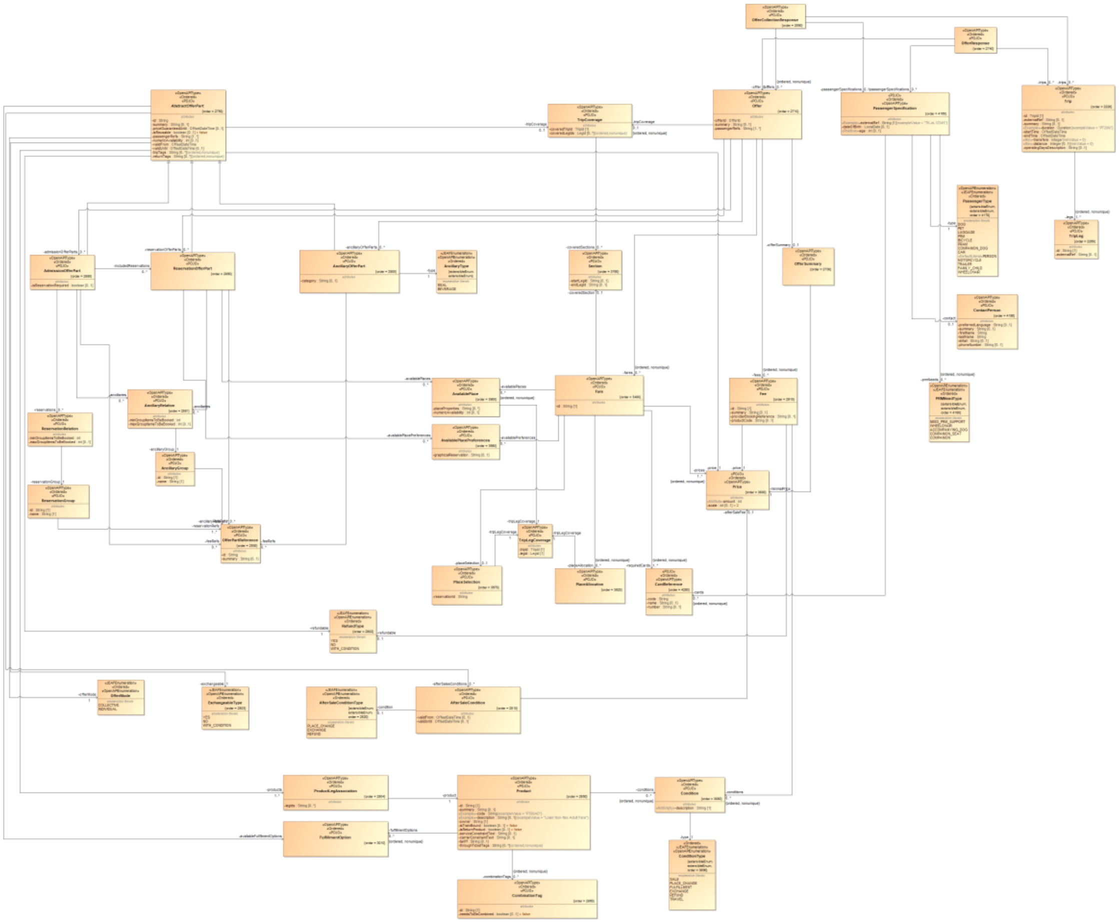
Regardless of whether the products to travel these tripLegs are train-bound, or based on with a validity period of any duration, tripLegs (and by extension trips) are always train-bound and represent the realization of the travel wish using specific trains at a specific moment in time.

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

A trip can be in states *planned*, *confirmed*, *changed* or *cancelled*.

Offers

Offers Data Model



Offers Data Model

Main Resources

Offers

In **Retailer Mode**, offers represent a collection of `OfferParts`, representing bookable elements that covers exactly one a specific trip (or the requested section of a trip in case of request to a FareProvider). 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

Some of these elements can be optional (see further).

In **Distributor Mode**, an offer will not contain offer parts but it will contain a fare element, that the distributor can use to build the final product to be sold 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 `minimalPrice` (= the price of the offer without any of the optional offer parts), a global service class and flexibility. Although the calculation rules for these global values are up to the OSDM provider, the following rules are recommended:

- `ServiceClass`: the lowest class of a significant offer part should be the service class of the offer (1st class + 2nd class = 2nd class)
- `Flexibility`: the lowest flexibility of a significant offer part should be the service class of the offer (full flex + mid flex = mid flex)

Offers usually have a validity period, that is the period over which, from the fare 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 sub resources) should have a limited lifetime (recommended value 30 minutes) and be discarded when expired or at booking time.

TripOffers

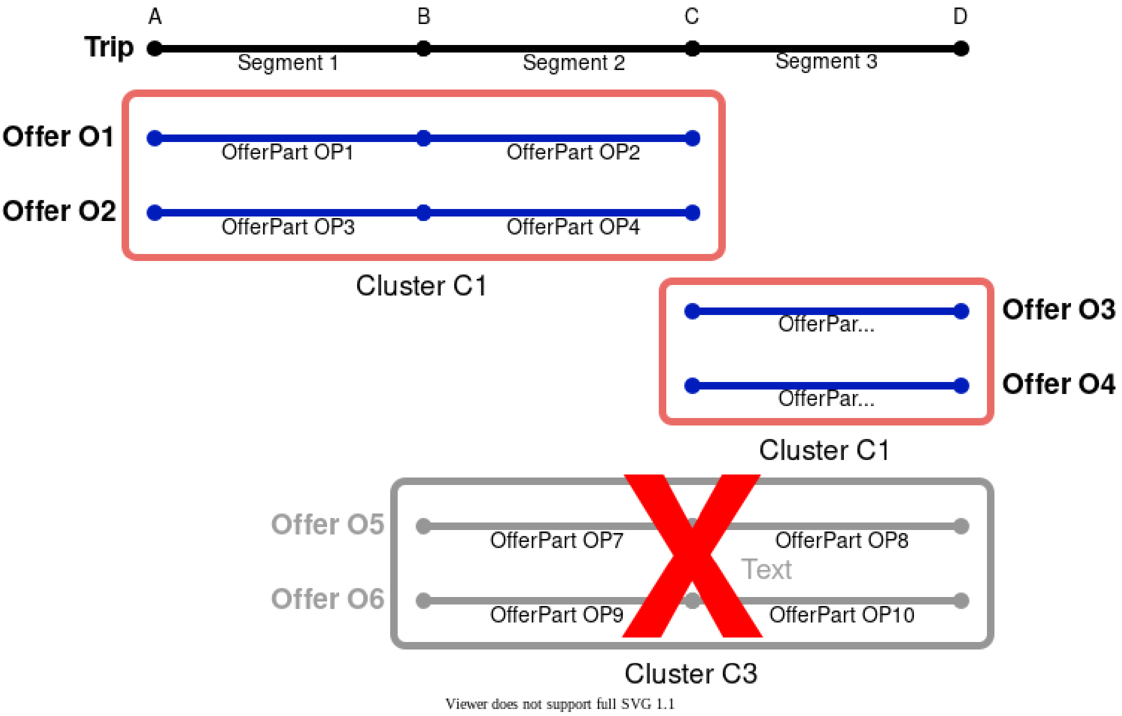
`TripOffers` are the resources grouping all the offers relating to one same trip. Indeed, in most cases the **Distributor** provider will propose several offers of different comfort and flexibility levels. In this resources, the trip resource representing the trip the offers are for and the passengers 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:

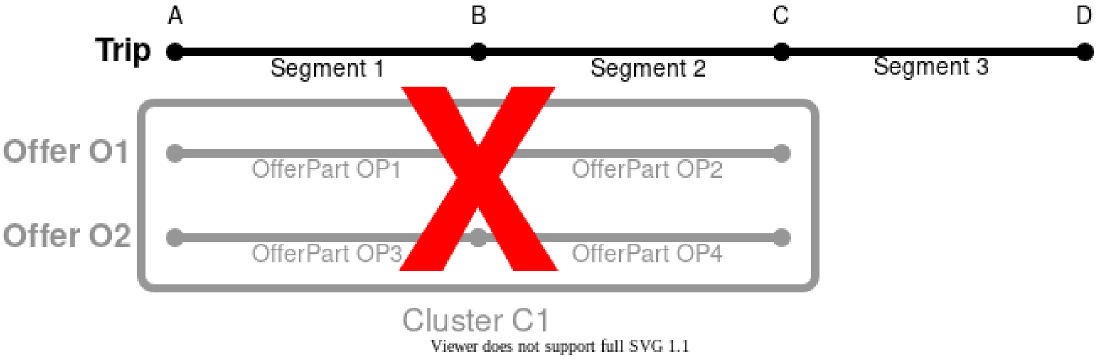
- the tripLegs covered by a given offer are indicated through the `coveredTripLegIndexes` property
- all offers covering the same set of tripLegs belong to the same `offerCluster`. All offers related to the same `offerCluster` therefore have an identical set of `coveredTripLegIndexes`
- a `tripLeg` can only be covered in one `offerCluster` within a `tripOffer` (no overlap)
- each `tripLeg` of the `trip` must be covered by at least one `offer` in each `TripOffer` (no gap)

Example with no overlap



Clusters-no-overlap

Example with no gap



Clusters-no-gap

OfferParts

Although `OfferParts` are by themselves not a resources, they deserve a separate section in order to for the reader to clearly understand the data model design. The idea is that `OfferParts` 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**
- **Ancillaries**
- **Fees**

However, all 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

In OSDM, 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 traveling together happen to get exactly the same product (because their profile is identical in terms of age, reductions etc), will still get two 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)

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 vehicles, 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 tripLegs. 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 a 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.

Offer Parts - Reservations

Reservation offer parts represent seat or other accommodation type reservation on the transportation. It might contain multiple seats/places. In contrast with admissions, a reservation is in essence bound to a specific train, although it does not include the entitlement to board the train. Passengers therefore typically need an a 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 in.

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

- Reservation Details provide additional information on the accommodation type 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:
  - `placeSelection/ReservationOptions` show, at offer retrieval stage which options are available for this reservation.
  - `SelectedOptions` allows the API consumer to specify desired options.
  - `SelectedPlaces` indicates selection of specific places (probably only relevant in case of graphical place selection.

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, two 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 tripLeg, while still offering the offer for the complete trip with reservations on all tripLegs 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 WIFI 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.

Offer Parts - Fees

Fees are used to represent additional costs for services or products. Offer parts of type "fee" can be applied to the booking process (e.g. a service fee), the trip (e.g. a reservation fee which is applied to all reservations in trains running in the same direction, namely outward or inward travel) or other offer parts. In contrast to other offer parts in OSDM, the customer is not free whether to choose a fee or not: fees are generated and applied to other services or products by the provider system.

Products

Products are the products actually offered by the distributor. 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 & after sales 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 want to do that.

Fares

Fares should be seen as the counter part of `OfferParts` in case of interactions between an distributor and a fare provider. 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 distributor to build the distributable product (that he could then sell as an offer part to a retailer), 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 distributor as well.

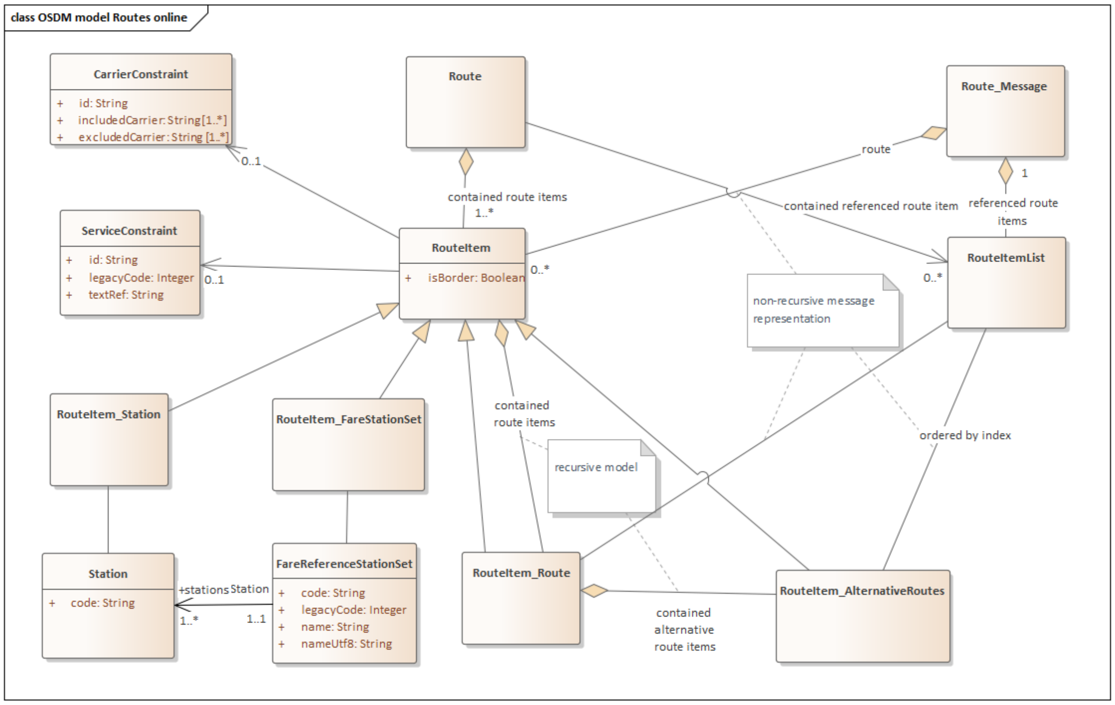
Distribution systems which also act as retailers might encapsulating both fares and offer parts in offers allowing to have a similar flow of interactions regardless of the type of provider.

Fares provide all information needed to be combined by an distributor and to allow an distributor to create fulfillments and support of control processes (e.g. providing combined control data to the TCOs). This especially includes the temporal and regional validity for travel. The rules on how to combine the fares (combination constraints) must also be included in the fare.

Regional Validity Route model

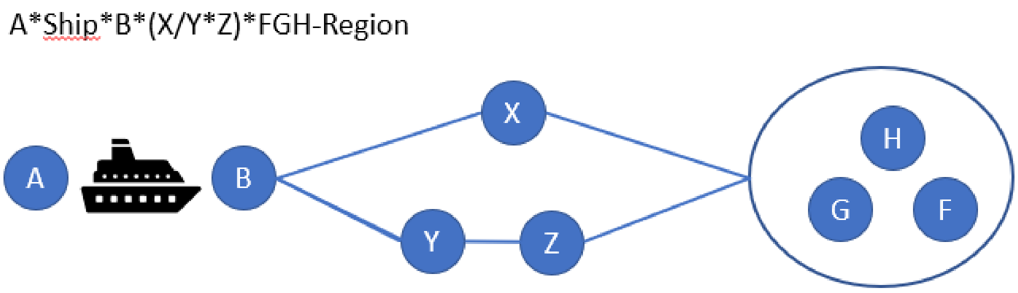
The route in fares is modeled as a structural model to allow additional functionalities including automated ticket control or validation of new trips with the described route. The model is compliant with the route model used in the FCB bar code data and the IRS 90918-4 ticket data exchange for control.

The data model makes use of recursive definitions to simplify implementations but the message contains a non-recursive representation of the data where recursive links have been replaced by a reference to an index in a list.

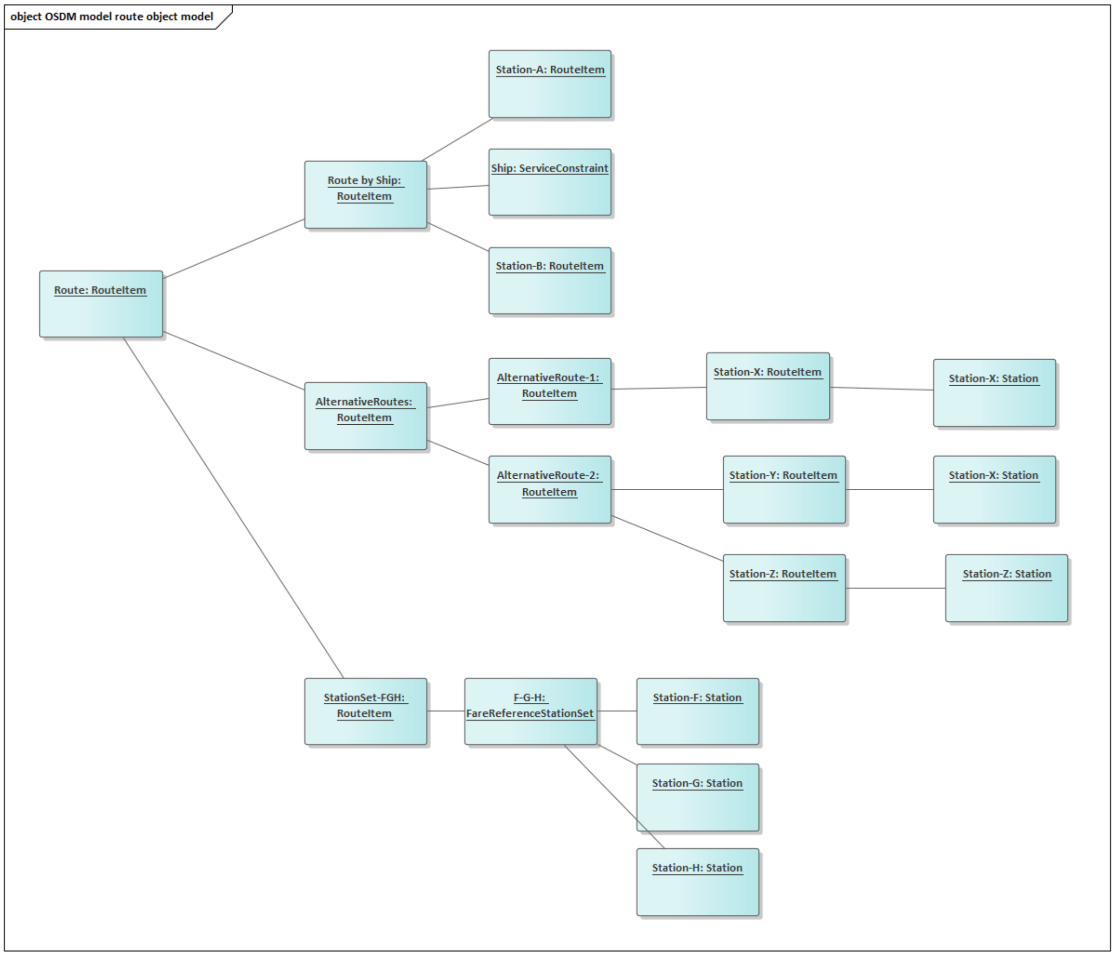


Route data model

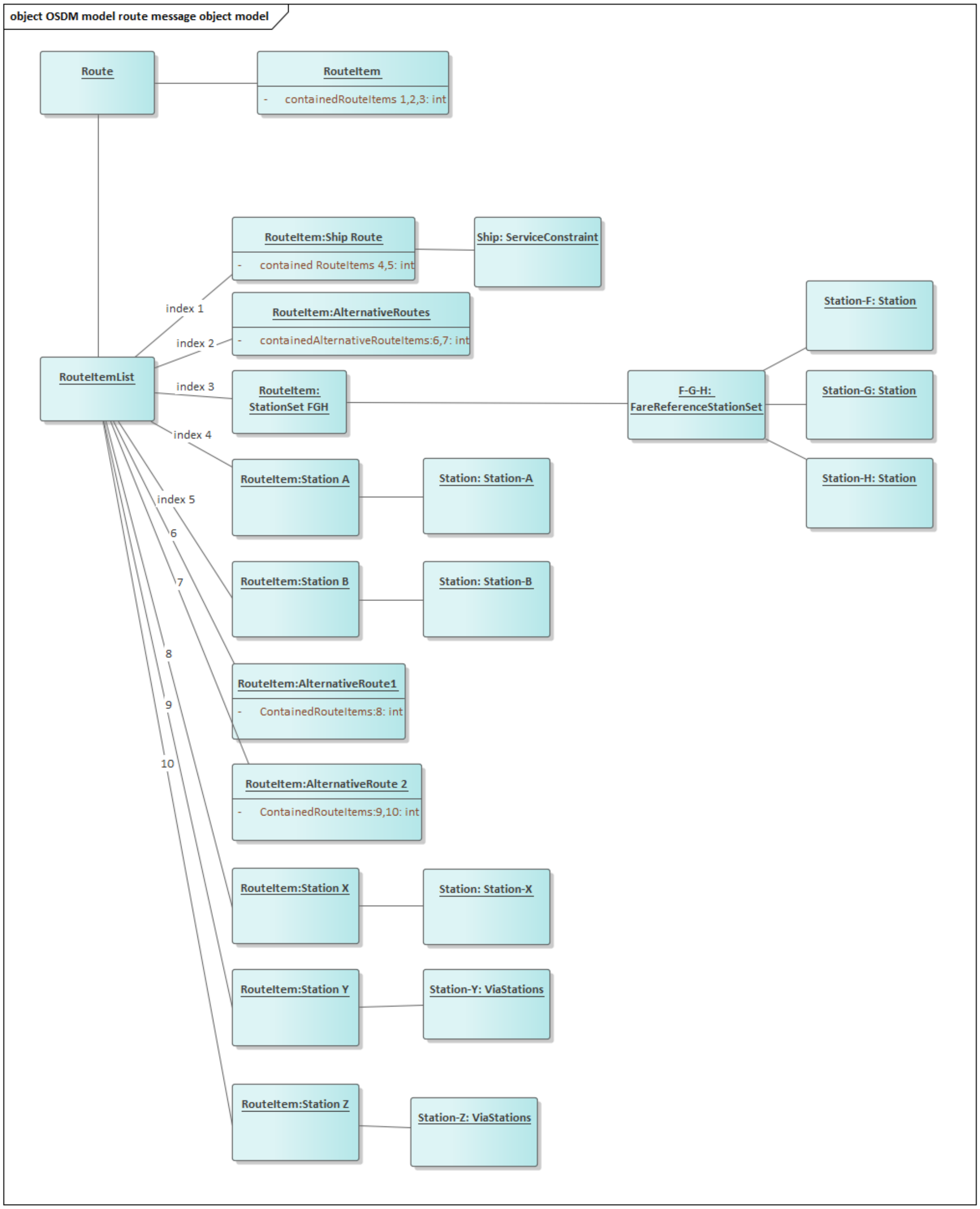
For the following example object models are shown for the message representation and the recursive model:



Route example



Route example recursive



Route example non-recursive

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 allowed to travel some tripLegs without actually needing a travel right or reservation. It is for example usually the case for infants traveling 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.

Passengers Representation

In the railway world, several elements are used to define a passenger profile (mostly in order to define the products it is entitled to):

- the passenger's age
- the reduction cards the passenger owns
- whether the passenger is a reduced-mobility or otherwise disabled passenger
- other specific status entitling to specific fares (military, senator, journalist…)

While in some systems, all the notions above are mixed into one "passenger type" notion, this approach is much more difficult, and cumbersome, to apply when multiple providers are involved, which is highly likely with OSDM. Indeed, different systems often have different age limits for types, and different ways to represent the other elements. For this reason, in OSDM we decide to map the elements above to two kinds of attributes:

- Passenger birth date, modelled as-is in the API. Note it could be a dummy birth date. Each implementer is then free to map this value to the age-related passenger types he is using internally

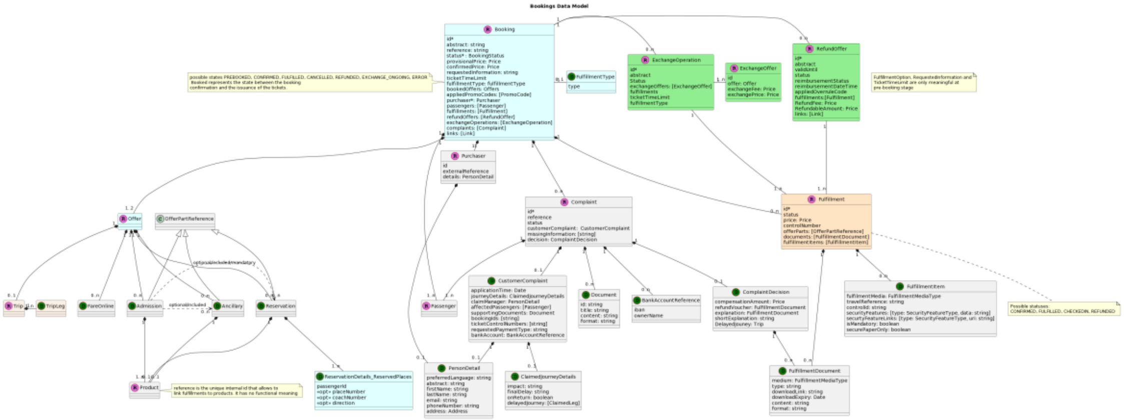
- Some attributes related to passengers disabilities (for accommodation purposes mainly)

- All other notions are modelled as reductions. Again, each implementer can map internally this clearly defined notion to the internal representation.

The presentation hereunder provides some additional examples of high-level offer modelings for pure-OSDM offers.

Booking

Booking Data Model



Bookings Data Model

Main Resources

Booking

The booking represents the offers that have been selected and turned into a booking on request to the provider of the offers.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.

At the root of the booking structure, two balance elements are provided to clarify the state of the financial exchange between a Retailer and the Distributor:

- 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.
- confirmed balance: is the balance of the booking that is confirmed. Unless after sales takes place on one or more fulfillments in the booking, this amount now must be paid to the provider.

Also located at the root of the booking structure is the ticket time limit. 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 pre-booked state. A commonly accepted value would be around 30 minutes, which is normally sufficient to allow finalizing 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.

`FulfillmentOptions` allows the API consumer to specify the format desired for the fulfillment. Only electronic fulfillment is considered in the MVP scope.

Purchaser

A purchaser represent the entity which has booked an offer. It is the person to contact in case of changes to the booking primary.

A purchaser does not need to travel thus is not necessary a passenger.

Common attributes of a passenger and a passenger are factored in the person entity.

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 `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 in OSDM only the sales 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 after sales 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 (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.

RefundOffers

Refund offers represent a provisional refund request that is made on all or a subset of the fulfillments contained in a booking.

ExchangeOperations

An exchange operations represent an ongoing exchange process, either in provisional state of in confirmed state (depending on its status). Much like a provisional booking, a provisional exchange contains the provisionally selected (new) Exchange Offer, a status and a ticket-time-limit. In addition, it also contains a reference to the fulfillments that are involved in the exchange, and will be cancelled as a result of its confirmation. Confirmed exchange operations are very similar, except for their status that will change, obviously, and the fact that the exchangeOffer is then transformed into a booked Offer in the booking and only referenced in the exchangeOperation

Exchange Offers

The exchange offers (and related models such as exchangeTripOffers) are totally similar to their offer counterpart, with the difference that ExchangeOffers also have a link to the fulfillments involved in the exchange operation, and also have 2 additional attributes for the exchange fee and exchange balance (= the difference between the value that can be returned form the fulfillment and the value of the current offers + the exchange fees = the total amount to be paid or refunded if/when confirming the exchange)

Complaint

A complaint can be filed by a passenger in case of delay/disruption or service derogation for a booking or parts of it (e.g. in case of a return trip). A passenger can provide documentation to support the claim (e.g. scanned manual remark of train staff) and it must able to lookup the status of the claim made.

If a claim is accepted or rejected the system that made the request must be informed. The accepted or rejected claim must provide an explanation for the decision especially if the amount covers the ticket price only partially.

Reimbursement

A reimbursement can be filed by a passenger in case his booked tariffs allows for a refund on unused or partially unused tickets. A passenger can provide documentation to support the request (e.g. scanned manual remark of train staff) and he must able to lookup the status of the request made.

If a request is accepted or rejected the system that made the request must be informed. The accepted or rejected request must provide an explanation for the decision especially if the amount covers the ticket price only partially.

Release

A release of a booking is an intermediate step toward a refund. It invalidates tickets and frees booked resources (e.g. reserved places on a train). The release does not refund money to the customer and thus can be made by another party involved. The refund is then completed by the original retailer. The benefit for the customer is that the time of the release is used to calculate the refund amount.

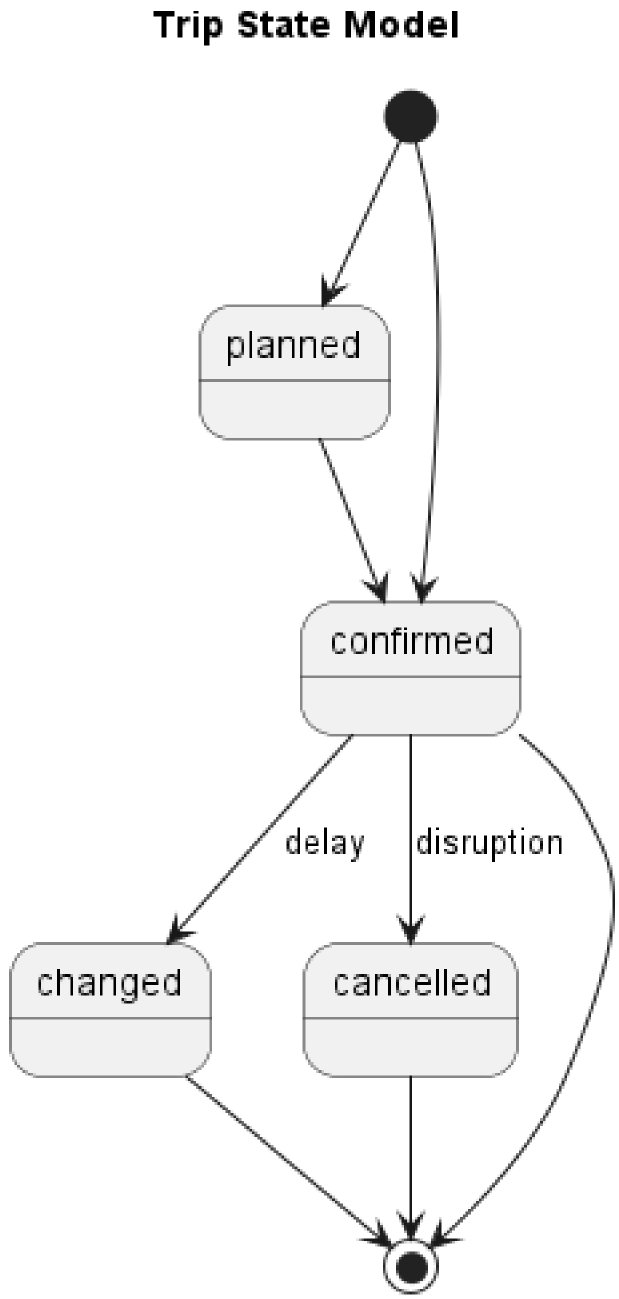
Putting bookings on Hold

An unconfirmed booking expires after the time limit of the pre-booking. It is possible to ask for an extension of the time limit and the provider might grant the extension. He has the option to add a fee for this extension.

State Models

Trip State Model

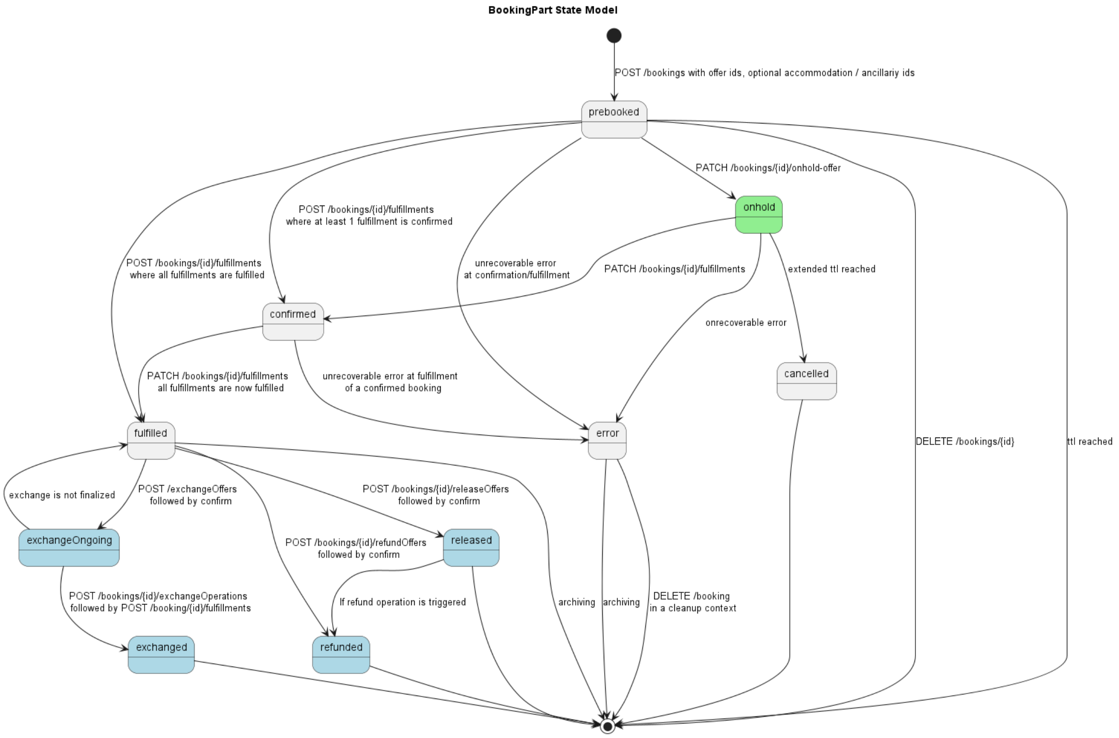
The state `PLANNED` is relevant for trips on touristic trains or if trains only run if a certain amount of bookings have been made. If the the trip is confirmed, the purchaser can be informed using the webhook API.



Trip State Model

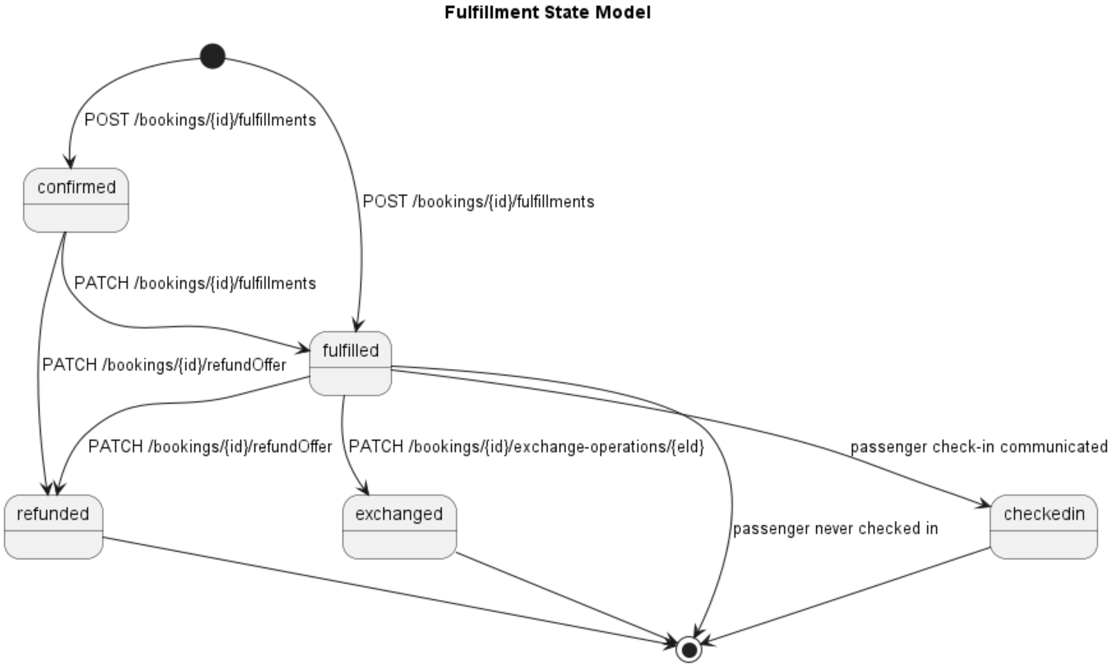
Booking State Model

The status applies to the booking parts, no overall booking state exists.


BookingPart State Model
POST /bookings with offer ids, optional accommodation / ancillariy ids
prebooked
PATCH /bookings/{id}/onhold-offer
onhold
POST /bookings/{id}/fulfillments
where at least 1 fulfillment is confirmed
POST /bookings/{id}/fulfillments
where all fulfillments are fulfilled
unrecoverable error
at confirmation/fulfillment
PATCH /bookings/{id}/fulfillments
extended ttl reached
confirmed
onrecoverable error
cancelled
PATCH /bookings/{id}/fulfillments
all fulfillments are now fulfilled
unrecoverable error at fulfillment
of a confirmed booking
fulfilled
error
DELETE /bookings/{id}
ttl reached
exchange is not finalized
POST /exchangeOffers
followed by confirm
POST /bookings/{id}/releaseOffers
followed by confirm
exchangeOngoing
POST /bookings/{id}/refundOffers
followed by confirm
released
POST /bookings/{id}/exchangeOperations
followed by POST /booking/{id}/fulfillments
If refund operation is triggered
archiving
archiving
DELETE /booking
in a cleanup context
exchanged
refunded

Booking State Model

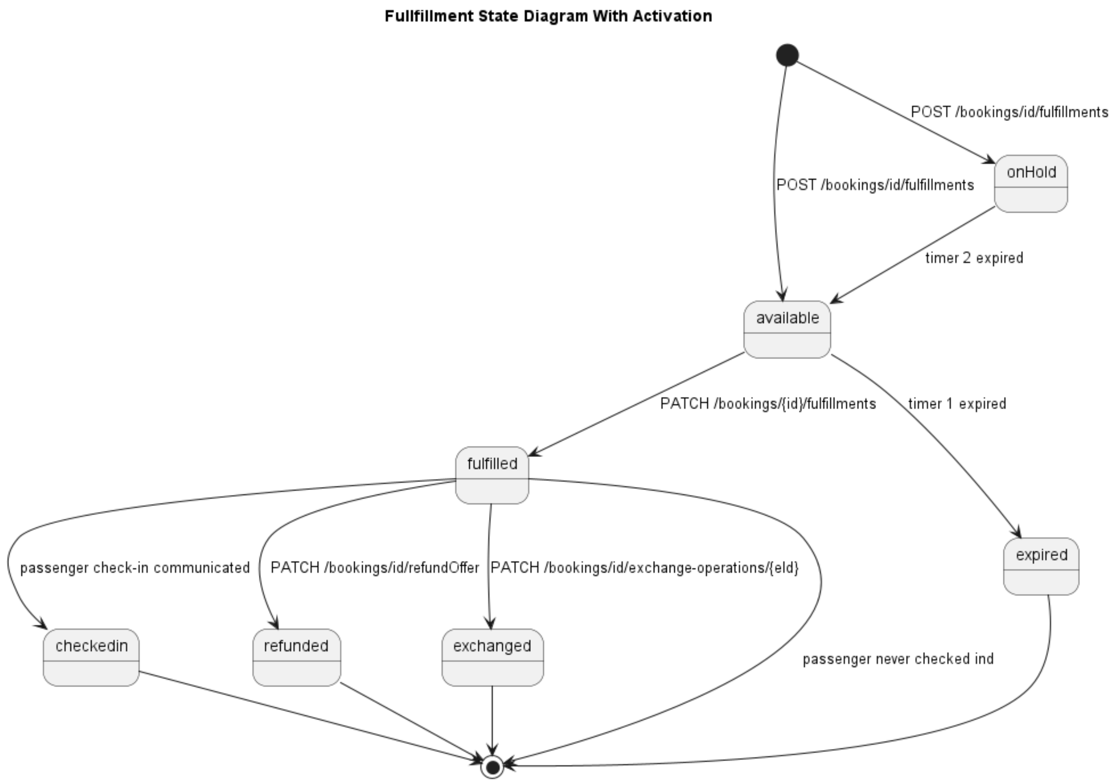
Fulfillment State Model


Fulfillment State Model
POST /bookings/{id}/fulfillments
POST /bookings/{id}/fulfillments
confirmed
PATCH /bookings/{id}/fulfillments
fulfilled
PATCH /bookings/{id}/refundOffer
PATCH /bookings/{id}/refundOffer
PATCH /bookings/{id}/exchange-operations/{eId}
passenger check-in communicated
refunded
exchanged
passenger never checked in
checkedin

Fulfillment State Model

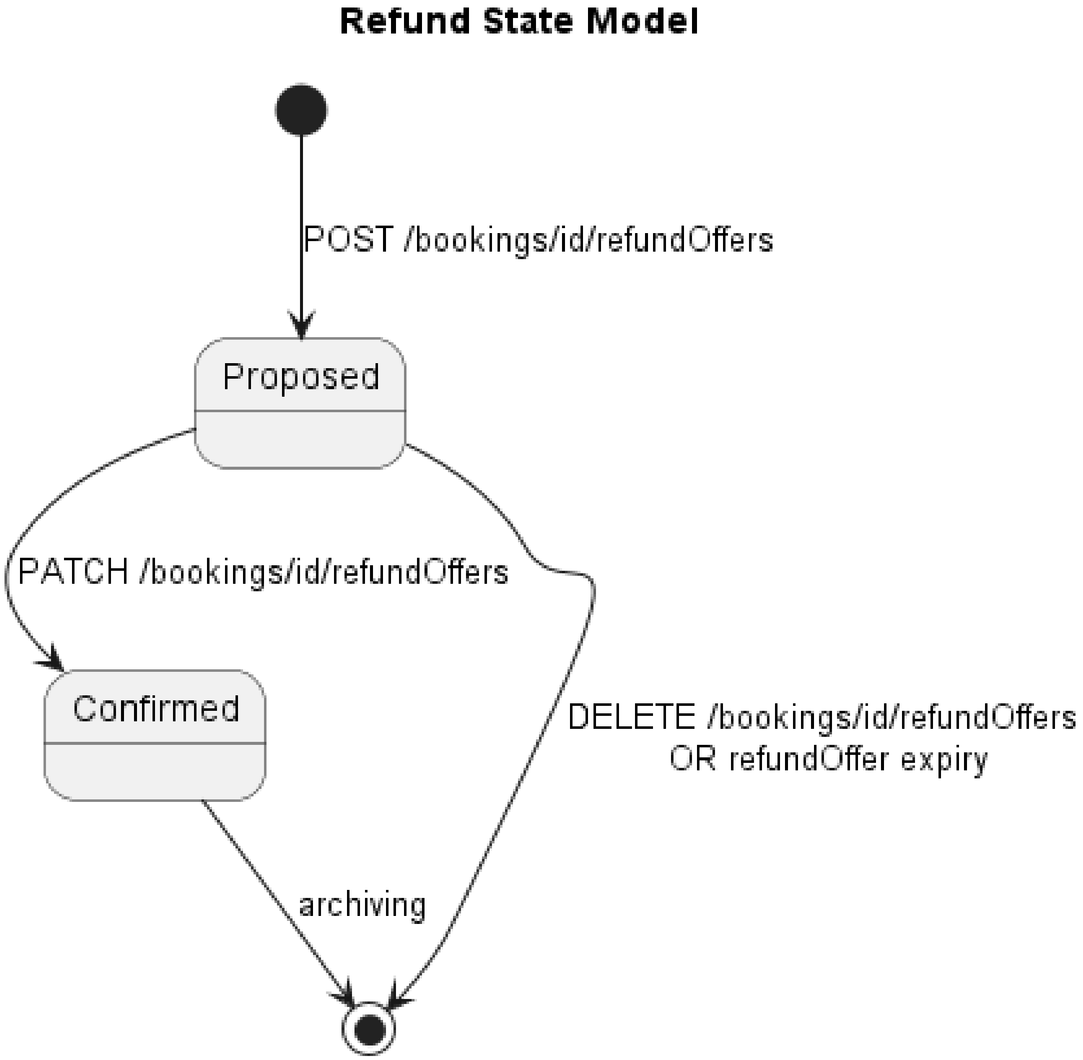
Fulfillment State Model with Activation

In case of multi-journey products, a fulfillment needs to be activated before, thus it changes from `AVAILABLE` to `FULFILLED`.


Fullfillment State Diagram With Activation
POST /bookings/id/fulfillments
POST /bookings/id/fulfillments
onHold
timer 2 expired
available
PATCH /bookings/{id}/fulfillments
timer 1 expired
fulfilled
expired
passenger check-in communicated
PATCH /bookings/id/refundOffer
PATCH /bookings/id/exchange-operations/{eId}
passenger never checked ind
checkedin
refunded
exchanged

Fulfillment State Model

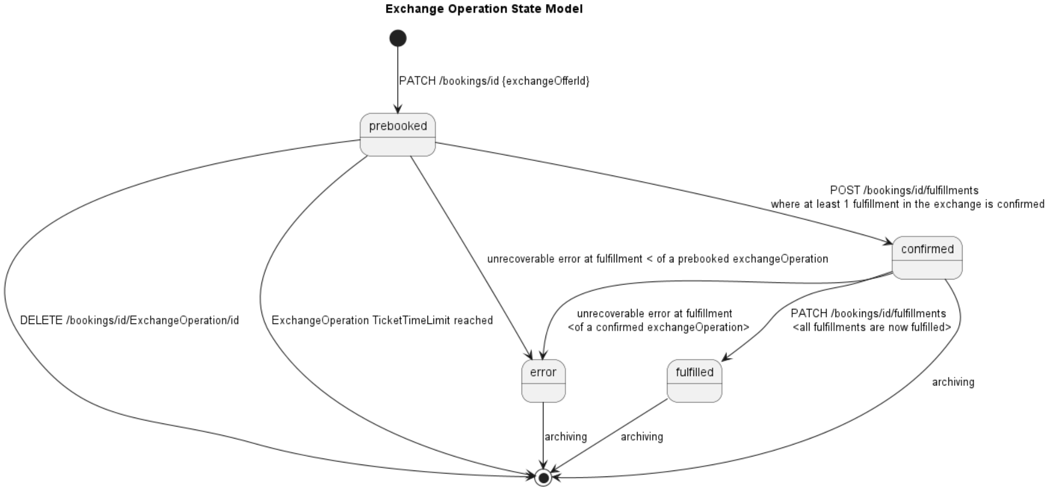
Refund State Model


Refund State Model
POST /bookings/id/refundOffers
Proposed
PATCH /bookings/id/refundOffers
Confirmed
DELETE /bookings/id/refundOffers
OR refundOffer expiry
archiving

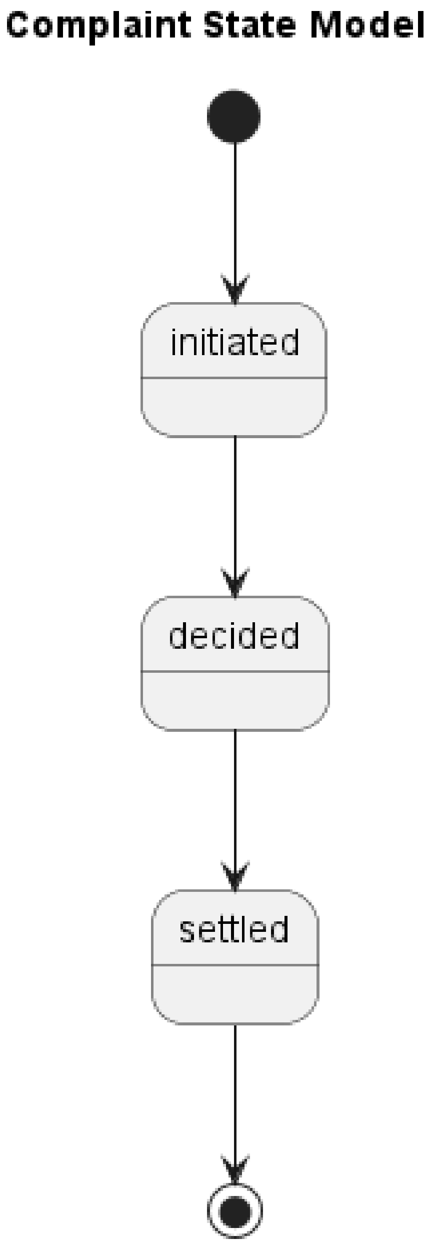
Refund State Data Model

Exchange State Model

Values are a subset of the booking status values (see before).


Exchange Operation State Model
PATCH /bookings/id {exchangeOfferId}
prebooked
POST /bookings/id/fulfillments
where at least 1 fulfillment in the exchange is confirmed
confirmed
unrecoverable error at fulfillment < of a prebooked exchangeOperation
DELETE /bookings/id/ExchangeOperation/id
ExchangeOperation TicketTimeLimit reached
unrecoverable error at fulfillment
<of a confirmed exchangeOperation>
PATCH /bookings/id/fulfillments
<all fulfillments are now fulfilled>
error
fulfilled
archiving
archiving
archiving
archiving

Complaint State Model



Complaint State Model

Ids and References

Object	ids	description
Place	id	id defining the place. The code is provided as URN, relative URNs are allowed with base path urn:uic:stn ‘8500001’

Object	ids	description
Place	alternativeIds	For a place with ids in different reference systems, the alternative ids can be returned. The reference system is encoded in the string. E.g.: “urn:uic:std:8000001”, “x_swe:stn:10000”, “ch:1:sloid:343434”
TripCollectionResponse	id	included to be compliant with OJP, no use
Trip	id	id of the trip
Trip	externalRef	external reference of the trip, e.g. provided by a time table provider / engine
Trip	situationFullRefs	references to situation messages (outside of OSDM)
TripSummary	id	id of the trip that is summerized
TripSummary	externalRef	external reference of the trip summarized, e.g. provided by a time table provider / engine
TripLeg	id	id of a leg in the trip. Unique within the trip only.
TripSpecification	externalRef	Referencing a Trip via the Trips id
TripLegSpecification	externalRef	Referencing a TripLeg via the TripLegs id
Section	externalTripRef	Referencing a Trip via the Trips id
Section	startLegId	Referencing a TripLeg via the TripLegs id
Section	endLegId	Referencing a TripLeg via the TripLegs id

Object	ids	description
AnonymousPassengerSpecification	externalRef	Reference to a passenger provided by the API consumer. Unique within the context of one Offer/Booking. Must not reference a passenger in a broader context due to GDPR regulation
Offer	offerId	Id of the offer given by the provider.
Offer	passengerRefs	reference to the passengers via the externalRef of the AbstractPassengerSpecification
Product	id	Id of the product
TripCoverage	coveredTripId	reference to the trip via the id of the trip.
TripCoverage	coveredTripLegIds	reference to the covered TripLegs within the referenced Trip via the TripLeg id
AbstractOfferPart (AdmissionOfferPart, ReservationfferPart, AncillaryOfferPart)	id	id of the offer part
AbstractOfferPart (AdmissionOfferPart, ReservationfferPart, AncillaryOfferPart)	passengerRefs	reference to the passengers via the externalRef of the AbstractPassengerSpecification
Fee	id	id of the fee
Fee	distributorBookingRef	useless
Fee	productRef	reference to the product via the produt id
Fare	id	id of the fare
Product	id	id of the product
Coach	layoutId	reference to the coach

Object	ids	description
		layout via the id of the CoachLayout
BookedOfferReservationRequest	offerId	reference to the offer from which the reservation offers parts are taken via the Offer id
BookedOfferReservationRequest	reservationOfferId	reference to the reservation offer part via the ReservationOfferPart id
BookedOfferReservationRequest	passengerRefs	reference to the passengers via the externalRef of the AbstractPassengerSpecification
Booking	id	id of the Booking
Booking	externalRef	reference to the booking provided by the consumer
Booking	bookingCode	short booking identifier by the provider of the booking. Human readable and unique within the provider context and for a limited time only
BookedOffer	offerId	id of the BookedOffer. Note: The BookedOffer id is different from the Offer id
BookingRequest	externalRef	reference of the consumer to be included in Booking as externalRef
AbstractBookingPart (Admission, Reservation, Ancillary)	id	id of the BookedOfferPart. Note: The BookedOfferPart id is different from the

Object	ids	description
		OfferPart id
AbstractBookingPart (Admission, Reservation, Ancillary)	passengerIds	reference to the pssengers via the Passenger id.
AbstractBookingPart (Admission, Reservation, Ancillary)	accountingRef	reference to accounting company e.g. in case the accounting company differs from the provider.
Passenger	id	id of the passenger created by the provider
Passenger	externalRef	external passenger ref btained from the consumer via the AbstractPassengerSpecification
Fulfillment	id	id of the fulfillment
Fulfillment	bookingRef	reference to the booking via th Booking id
Fulfillment	controlNumber	control number given by the provider and unique within the provider context and for a short time only. The control number is used as identifier within bar codes.

Table of contents

Service Specification

Introduction

This page shows a representation of 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.)

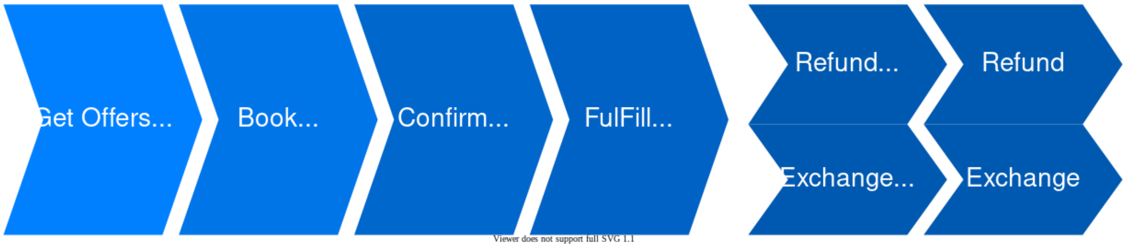
The main purpose of this document is therefore to help a quicker understanding of the API and its underlying concepts. As such, some of the details of how the information is structured in the API are not represented or simplified in the data models.

pagOverview of Services

Resources	Description
`/places`	Resources to search for places
`/trips`	Resources to search for trips
`/offers`	Resources to get bookable offers
`/availabilities`	resources to retrieve availability information on places (seats,..)
`/bookings`	Resources to manipulate bookings
`/bookings/{bookingId}/passengers/`	Resources to change passengers
`/bookings/{bookingId}/purchaser/`	Resources to change purchaser
`/bookings/{bookingId}/booked-`	resources to change pre-booked bookings,

Resources	Description
`offers/`	e.g. provide place selections
`/bookings/{bookingId}/reimbursements`	resources to reimburse unused tickets
`/bookings/{bookingId}/release-offers`	resources to release tickets
`/bookings/{bookingId}/cancel-fulfillments-offers`	resources to cancel fulfillments
`/products`	retrieve products information on one or more products
`/bookings/{bookingId}/fulfillments`	retrieve fulfillments, e.g. tickets
`/fulfillments`	confirm a booking and retrieve fulfillments
`/bookings/{bookingId}/refund-offers`	Resources to get and accept a refund offer
`/bookings/{bookingId}/exchange-operations`	Resources to get and accept a exchange offer
`/bookings/{bookingId}/exchange-offers`	*dito*
`/bookings/{bookingId}/release-offers`	Resources to get, accept or delete a release offer
`/bookings/{bookingId}/cancel-fulfillment-offers`	Resources to get, accept or delete a cancelFulfillment offer
`/complaints`	resources to create and manipulate complaints
`/coach-layouts`	Returns all coach layouts.
`/reduction-cards`	retrieve reduction card types
`/products`	retrieve product information
`/zones`	retrieve zone information

Process Flow



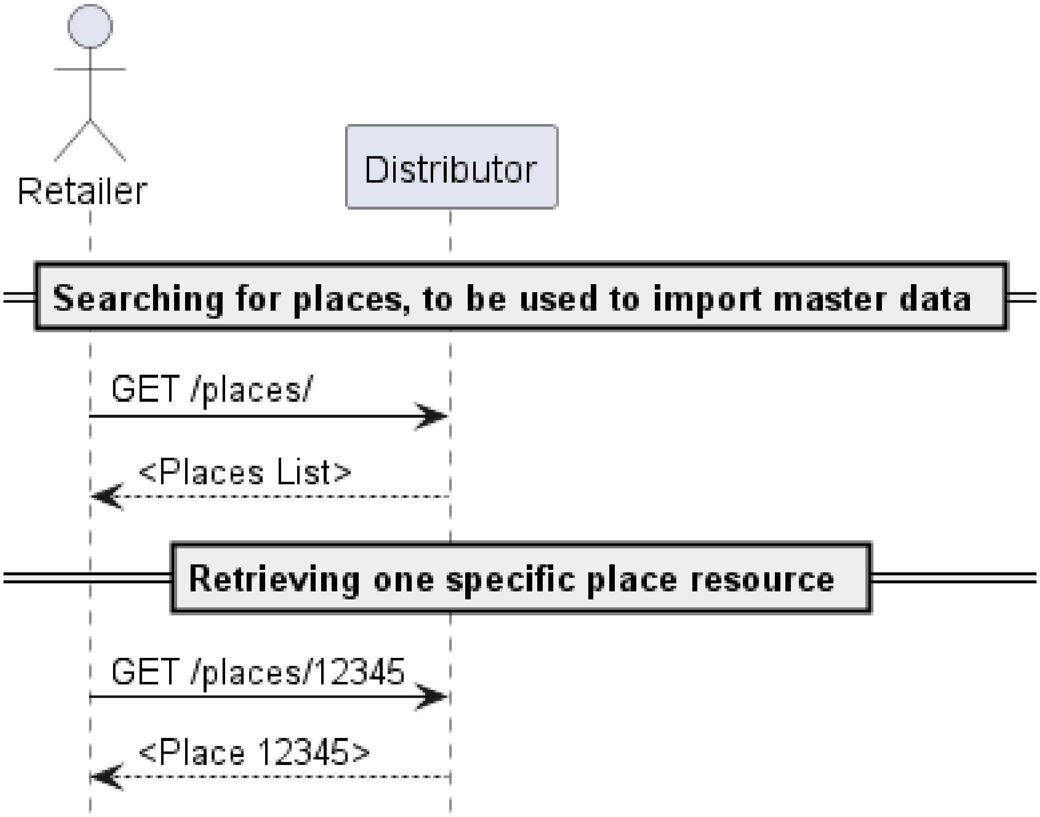
Process Flow

The process flow starts with getting offers which can be chosen by the customer. Once selected they can be pre-booked 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 Places Processes

Looking Up Places


Retailer
Distributor
Searching for places, to be used to import master data
GET /places/
<Places List>
Retrieving one specific place resource
GET /places/12345
<Place 12345>

Looking Up Places

The `\places` Lookup can be used by an API provider in order to search for places. Two typical uses cases would be

- getting a set of places (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 places list or to build a full “browsing” functionality, hence the lack of pagination features here.

Given the high stability of this information, places 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 **Retailer** remains basic here as a handful of cases have to be handled:

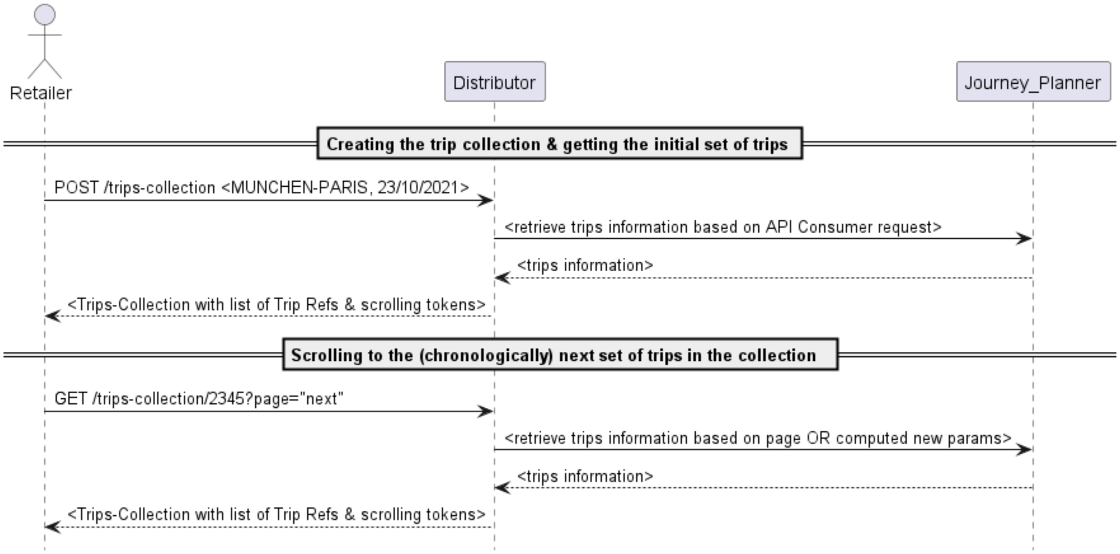
- invalid characters in the search string

- 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 **Retailer** 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


Retailer
Distributor
Journey_Planner
Creating the trip collection & getting the initial set of trips
POST /trips-collection <MUNCHEN-PARIS, 23/10/2021>
<retrieve trips information based on API Consumer request>
<trips information>
<Trips-Collection with list of Trip Refs & scrolling tokens>
Scrolling to the (chronologically) next set of trips in the collection
GET /trips-collection/2345?page="next"
<retrieve trips information based on page OR computed new params>
<trips information>
<Trips-Collection with list of Trip Refs & scrolling tokens>

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 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 distributor 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 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 to ensure that once a trip has been generated, its id can be re-used in possible subsequent operations within a reasonable time-frame:

- 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).
- It could be used in a subsequent offer request, so that offers are now built for that specific trip

Error Handling

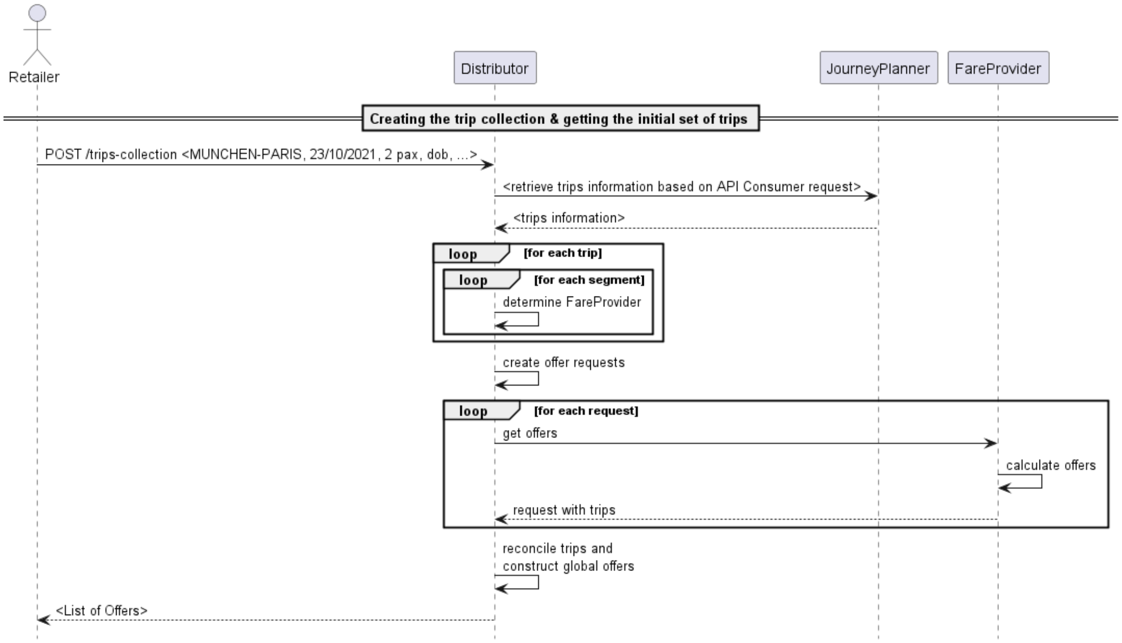
Since requesting trips still does not involve any transactional operation, the error handling is also limited to returning a JSON problem element. The following cases are to be considered:

- A search criteria value contains invalid value or invalid characters
- A search criteria 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

Getting and Browsing Offers



Getting and Browsing Offers

Requesting and browsing offers works a lot like the 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 is an alternative way to search offers if a set of specific trips is already known: provide the complete trip structure for one or several trips. This the only way to go for a request to a fare provider working according to fare rules. The trips 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. When this method is used, the API consumer can provide a "tripkey" together with the trip specification, which will be echoed in the tripoffer element matching that specific trip, for an explicit reconciliation between requested trips and provided offers.

An offer request to an **distributor** or **fare 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 **Retailer** must follow the following rules:

- While the combination logic on fares is left to the **Distributor**, it is recommended to only build and retain offers that are *homogeneous* (as much as possible) in terms of flexibility and comfort.
- As with the trips, it must remain possible to scroll forward or backwards over offers.

As described further on, any additional information required for the provisional booking can be provided in the booking operation itself

The resources used at offer steps optionally offer various levels of embedding (returning complete structure is the only mechanism mandatory to be implemented) 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 **distributor** or **fare provider** can 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 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 tag. 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 `offerTag` for the selected outward offer, or the set of potential offers (as the `offerTag` does not have to be unique. E.g. all offers for a given date might have the same if the constraint is only on date) can or must be provided. Whether the `offerTag` is mandatory in the inward offer request is indicated by the "mandatory flag" that is provided in the outward offer response next to each offerTag. If the tag 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 `offerTag` 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 returnTags

Besides the `offerTag` discussed above, some offers may have one or more `returnTag(s)` as well. As the name suggests, these can be used in order to determine how to combine offers in a return trip.

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

Hereunder an example illustrating this concept:

Outward Offers

- Offer1: -
- Offer2: #123

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

Inward Offers

- Offer5: -
- Offer6: #123
- Offer7: #345
- Offer8: #123, #234

Valid Combinations

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

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

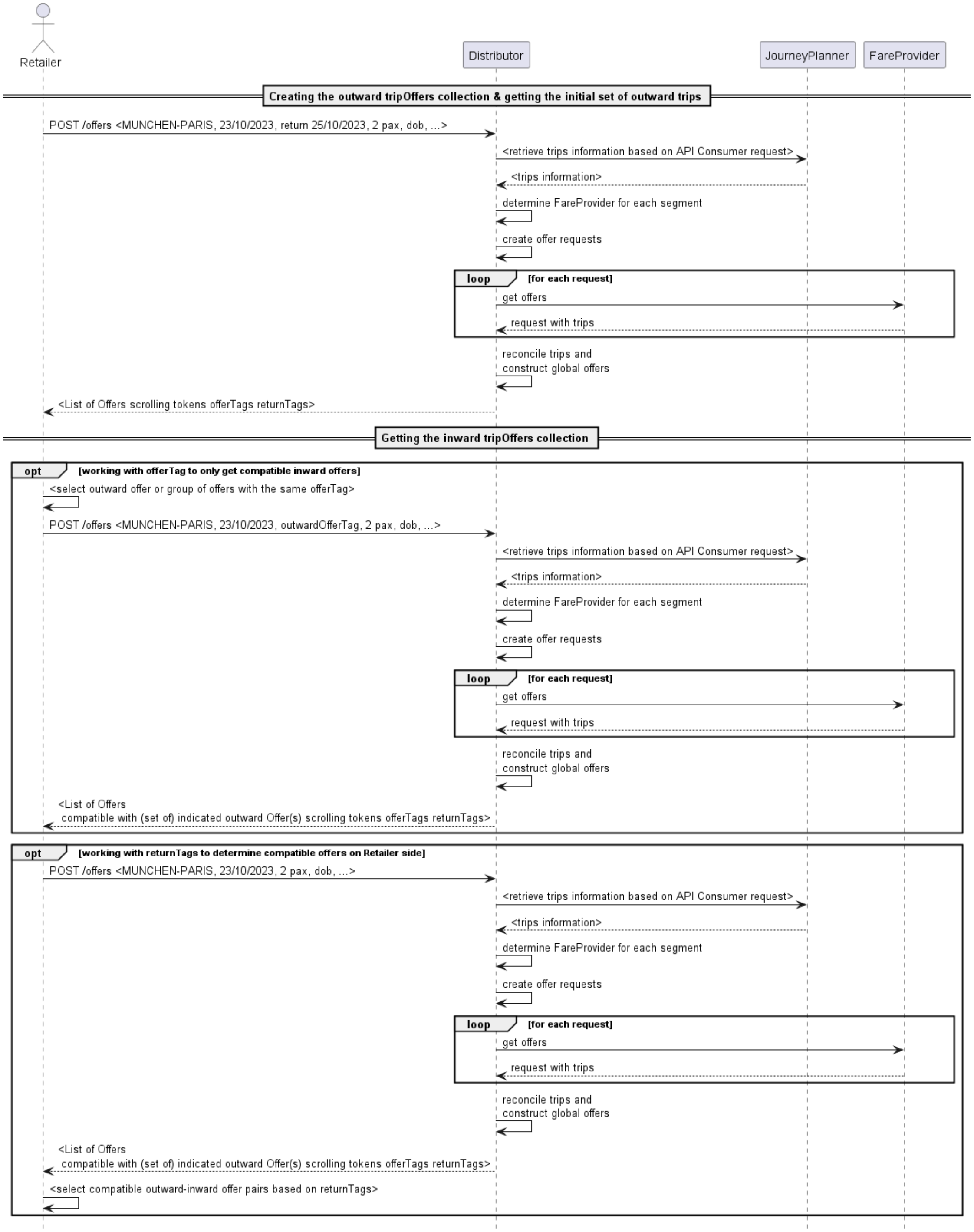
Products Covering Both Directions

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 approach:

- 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.


Retailer
Distributor
JourneyPlanner
FareProvider
Creating the outward tripOffers collection & getting the initial set of outward trips
POST /offers <MUNCHEN-PARIS, 23/10/2023, return 25/10/2023, 2 pax, dob, ...>
<retrieve trips information based on API Consumer request>
<trips information>
determine FareProvider for each segment
create offer requests
loop
[for each request]
get offers
request with trips
reconcile trips and
construct global offers
<List of Offers scrolling tokens offerTags returnTags>
Getting the inward tripOffers collection
opt
[working with offerTag to only get compatible inward offers]
<select outward offer or group of offers with the same offerTag>
POST /offers <MUNCHEN-PARIS, 23/10/2023, outwardOfferTag, 2 pax, dob, ...>
<retrieve trips information based on API Consumer request>
<trips information>
determine FareProvider for each segment
create offer requests
loop
[for each request]
get offers
request with trips
reconcile trips and
construct global offers
<List of Offers
compatible with (set of) indicated outward Offer(s) scrolling tokens offerTags returnTags>
opt
[working with returnTags to determine compatible offers on Retailer side]
POST /offers <MUNCHEN-PARIS, 23/10/2023, 2 pax, dob, ...>
<retrieve trips information based on API Consumer request>
<trips information>
determine FareProvider for each segment
create offer requests
loop
[for each request]
get offers
request with trips
reconcile trips and
construct global offers
<List of Offers
compatible with (set of) indicated outward Offer(s) scrolling tokens offerTags returnTags>
<select compatible outward-inward offer pairs based on returnTags>

Products Covering Both Directions

Error Handling

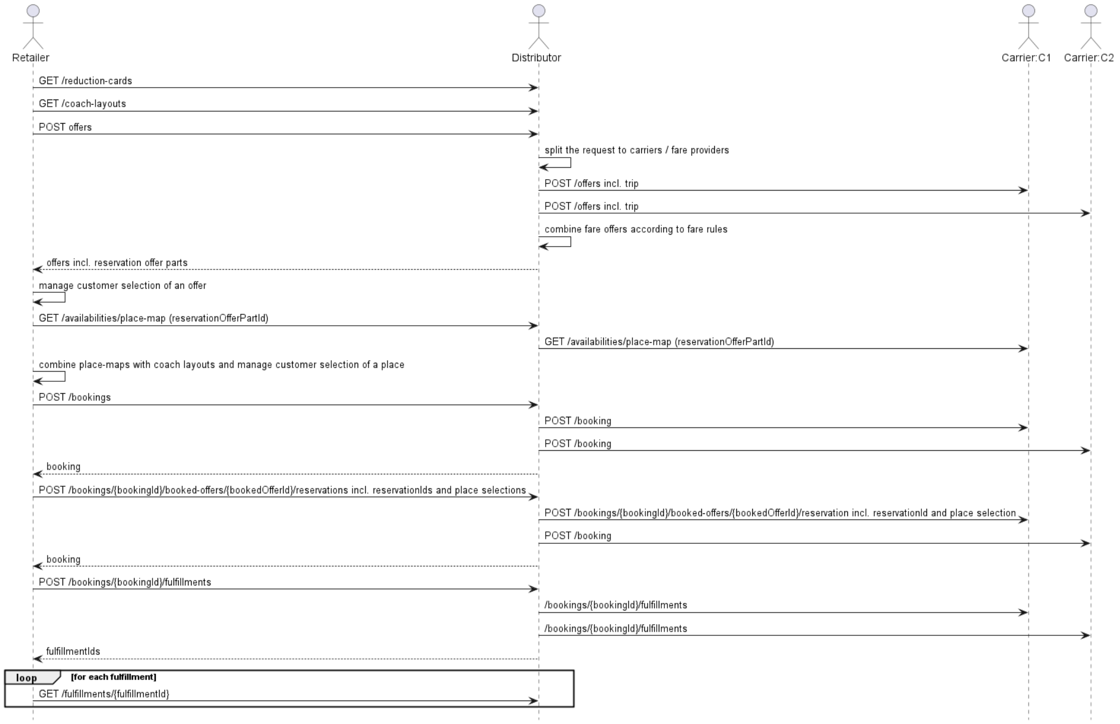
- the referenced trip cannot be found
- A search criteria value contains invalid value or invalid characters
- A search criteria lies outside accepted boundaries: it could be the date in the past, or too far in the future, or value outside bounds for the max number of changes

- 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
- Unknown error on server side

Reservation

Reservation offers are part of the offer but the selection of places is an additionl intermediate step after the prebooking of offers. The selection of places can be made via a graphical display of available places or via the specifcation of customer requirements (at a table, at the window, etc..).

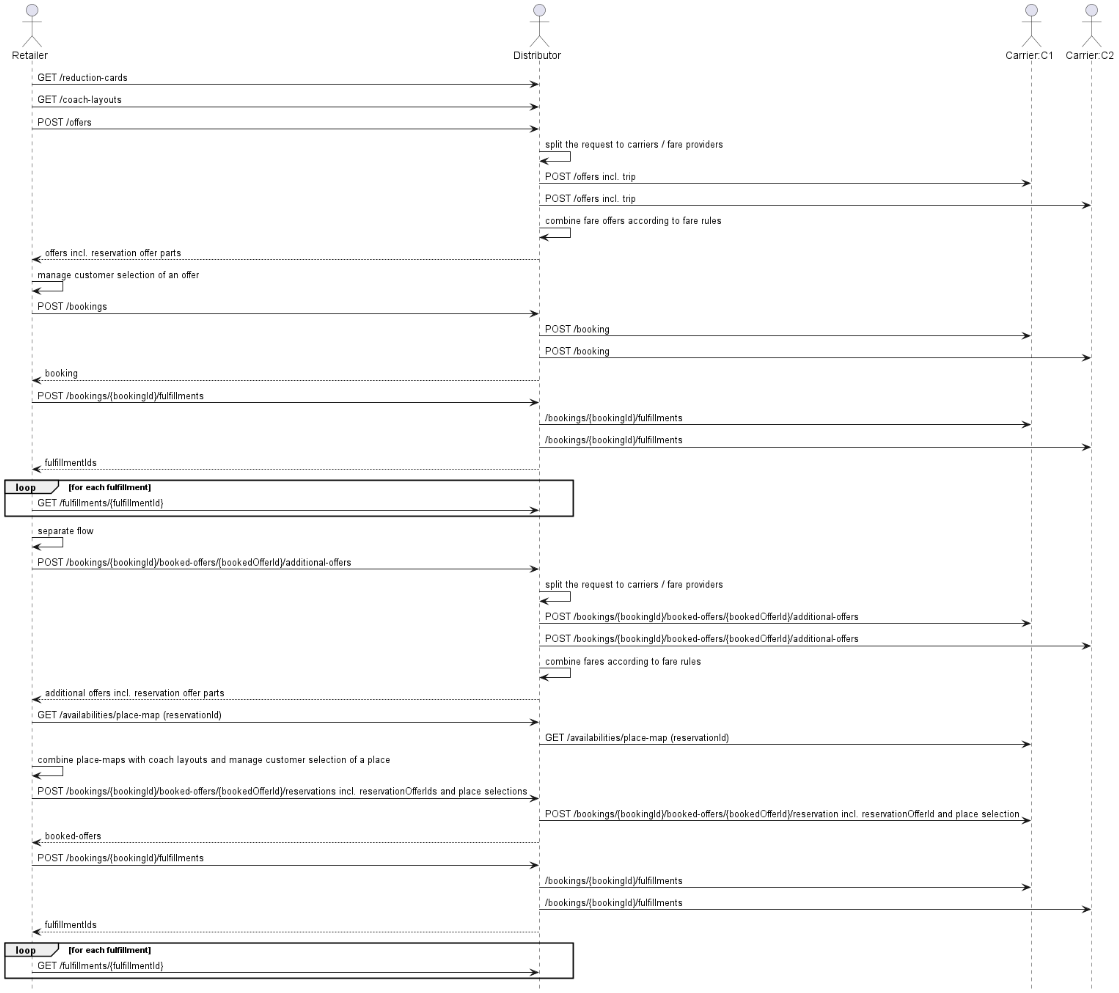
The basic flow is shown here:



Retailer
Distributor
Carrier:C1
Carrier:C2
GET /reduction-cards
GET /coach-layouts
POST offers
split the request to carriers / fare providers
POST /offers incl. trip
POST /offers incl. trip
combine fare offers according to fare rules
offers incl. reservation offer parts
manage customer selection of an offer
GET /availabilities/place-map (reservationOfferPartId)
GET /availabilities/place-map (reservationOfferPartId)
combine place-maps with coach layouts and manage customer selection of a place
POST /bookings
POST /booking
POST /booking
booking
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/reservations incl. reservationIds and place selections
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/reservation incl. reservationId and place selection
POST /booking
booking
POST /bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
fulfillmentIds
loop
[for each fulfillment]
GET /fulfillments/{fulfillmentId}

Reservations on a pre-booked Booking

Reservations can aso be added to an already confirmed booking:


Retailer
Distributor
Carrier:C1
Carrier:C2
GET /reduction-cards
GET /coach-layouts
POST /offers
split the request to carriers / fare providers
POST /offers incl. trip
POST /offers incl. trip
combine fare offers according to fare rules
offers incl. reservation offer parts
manage customer selection of an offer
POST /bookings
POST /booking
POST /booking
booking
POST /bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
fulfillmentIds
loop
[for each fulfillment]
GET /fulfillments/{fulfillmentId}
separate flow
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/additional-offers
split the request to carriers / fare providers
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/additional-offers
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/additional-offers
combine fares according to fare rules
additional offers incl. reservation offer parts
GET /availabilities/place-map (reservationId)
GET /availabilities/place-map (reservationId)
combine place-maps with coach layouts and manage customer selection of a place
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/reservations incl. reservationOfferIds and place selections
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/reservation incl. reservationOfferId and place selection
booked-offers
POST /bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
fulfillmentIds
loop
[for each fulfillment]
GET /fulfillments/{fulfillmentId}

Adding Reservations on a confirmed booking

Place Availability of Offers for Reservation

Each reservation item may provide availability information in `availablePlaces`. The total number of places is specified when availability of accommodation sub type `ANY_SEAT` is presented, optionally with all possible place properties. Sum of all available places object does not build the total number, some of these availabilities might be subset of the total. The fare provider is free to provide either total capacity only, or detail on other accommodation types and sub types.

```
{
  "availablePlaces": [
    {
      "accommodationType": "SEAT",
      "accommodationSubType": "ANY_SEAT",
      "placeProperties": [
        {
          "code": "SILENCE"
        },
        {
          "code": "WINDOW"
        },
        {
          "code": "AISLE"
        },
        {
          "code": "TABLE"
        },
        {
          "code": "OPEN_SPACE"
        }
      ],
      "numericAvailability": 123,
      "tripLegCoverage": {
        "tripId": 1,
        "legId": 2
      }
    },
    {
      "accommodationType": "SEAT",
      "accommodationSubType": "WITH_ANIMALS",
      "placeProperties": [
        {
          "code": "WITH_ANIMALS"
        }
      ],
      "numericAvailability": 5,
      "tripLegCoverage": {
        "tripId": 1,
        "legId": 2
      }
    }
  ]
}
```

Partial Reservation

A provider might offer a reservation even in case the seat is not available on the whole leg. In this casethe provider should include multiple reservation in the offer that will provide a reservation on a part of the leg.

A proprietary implemnetation is included in the specification where the split is indicated only in the reservation lateron using the splitSection data in the ReservedPlace object. A provider is allowed to use the splitSection only in case the client has allowed this by sending a ProductTag SPLITT_RESERVATION in the offer request.

Getting Coach Layouts

Graphical seat reservation allows a customer to conveniently choose its preferred place. Therefore two resources are added: First, `GET /coachLayouts` to import all coach-layouts of an distributor or fare provider. This service can be used periodically as master data service .

Second, `GET /coachLayouts/{layoutId}` returns the information for a given `layoutId` and can be used during the on-line offering and booking process.

A Complex Example Mixing Offers and Fares

Request From Front-end

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

Request Submitted to SNCB

Proposed trip by timetable system:

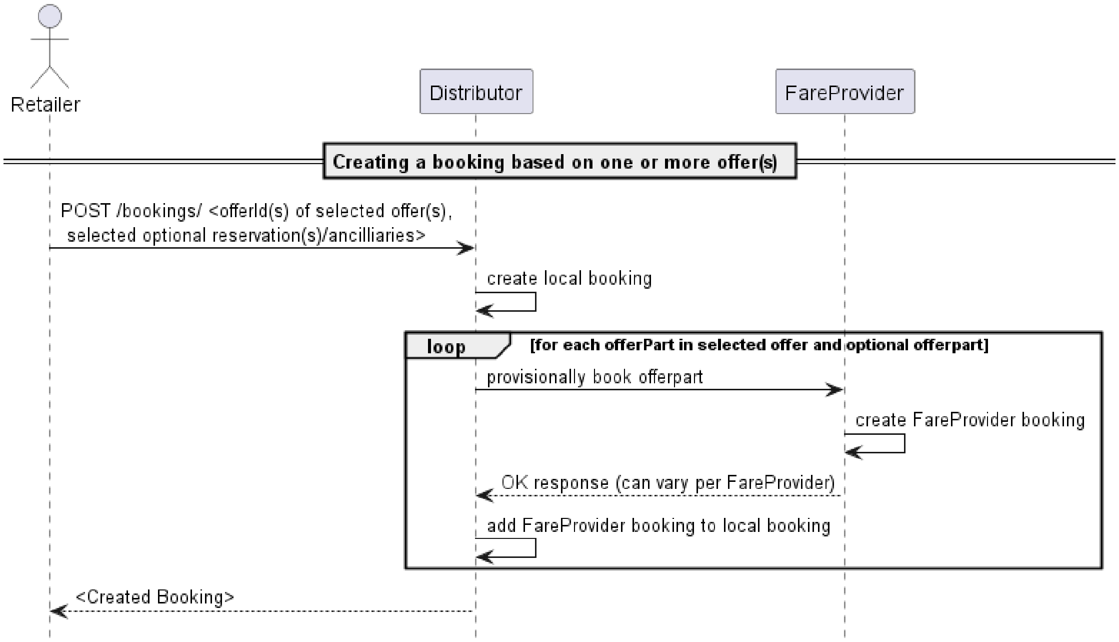
Origin - Destination	Train Number
Rotterdam → Antwerp	Thalys 9324 (mandatory reservation)
Antwerp → Liège	IC 2345 + IR 5567
Liège → Frankfurt	ICE 122 (mandatory reservation)
Frankfurt → Wien Hbf	RailJet RJ 23 (optional reservation)
Wien Hbf → Wien Stephansplatz	Metro

Fare Provider Resolution returns

Origin - Destination	Train Number	Fare Provider	Consolidated
Rotterdam → Antwerp	Thalys 9324 (mandatory reservation)	PAO	PAO
Antwerp → Liège	IC 2345 + IR 5567	Fare SNCB	Fare SNCB
Liège → Frankfurt	ICE 122 (mandatory reservation)	GUS	GUS
Frankfurt → Wien Hbf	RailJet RJ 23 (optional reservation)	Frankfurt → Salzburg (Border)	Fare DB
		Salzburg (Border) → Wien Hbf	Fare ÖBB
		Frankfurt → Wien Hbf (reservation)	Fare ÖBB
Wien Hbf → Wien Stephansplatz	Metro	Fare ÖBB	

Booking Processes

Creating a Booking Based on Offers


Retailer
Distributor
FareProvider
Creating a booking based on one or more offer(s)
POST /bookings/ <offerId(s) of selected offer(s),
selected optional reservation(s)/ancilliaries>
create local booking
loop
[for each offerPart in selected offer and optional offerpart]
provisionally book offerpart
create FareProvider booking
OK response (can vary per FareProvider)
add FareProvider booking to local booking
<Created Booking>

Creating a Booking Based on Offers

Once the offer has been selected, the API consumer can proceed to the booking of that offer. Along with the offer, optional or mandatory reservations or ancillaries can be booked as well. The optional offer parts can be easily identified 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 or ancillary is included (in which case it is not needed to explicitly add it in the booking request), mandatory (the reservation or ancillary must be added in the booking request) or optional (the reservation or ancillary may be added in the booking request).

Adding optional or mandatory elements is simply done by adding the respective offer part in the booking request (cf YAML specifications) `POST /bookings`.

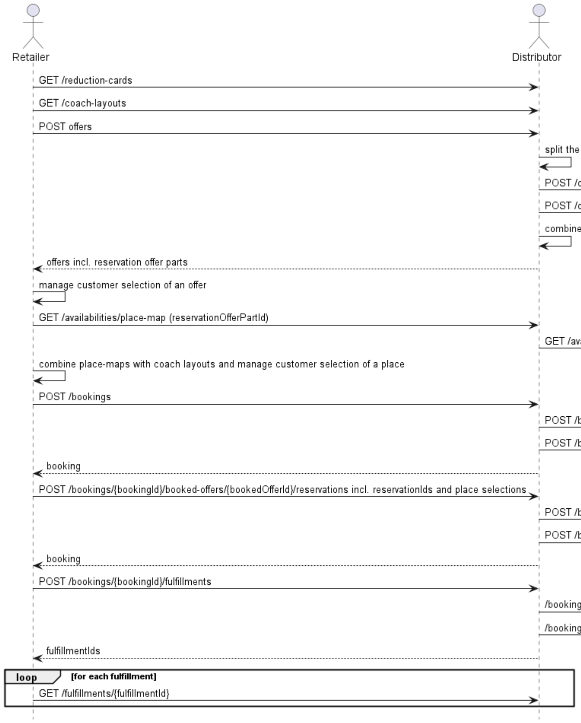
It is also possible to book several offers in one operation to the same booking. This is especially relevant to support return trips, where in most times it will be mandatory. If this is the case, a collection of offer ids (and associated reservations and ancillaries) is given instead of just one. However, 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 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.

Additional information in provisional booking step

In most cases the offer will not contain information on specific place properties for reservations. The reservation resource in the offer provides information on the availability of places with the selected offer:

- Places with specific properties; please refer to the section *Place Availability of Offers* on the semantics of provided Place Availabilities.
- Places nearby another place.
- A graphical display of available places.



Retailer
Distributor
Carrier:C1
Carrier:C2
GET /reduction-cards
GET /coach-layouts
POST offers
split the request to carriers / fare providers
POST /offers incl. trip
POST /offers incl. trip
combine fare offers according to fare rules
offers incl. reservation offer parts
manage customer selection of an offer
GET /availabilities/place-map (reservationOfferPartId)
GET /availabilities/place-map (reservationOfferPartId)
combine place-maps with coach layouts and manage customer selection of a place
POST /bookings
POST /booking
POST /booking
booking
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/reservations incl. reservationIds and place selections
POST /bookings/{bookingId}/booked-offers/{bookedOfferId}/reservation incl. reservationId and place selection
POST /booking
booking
POST /bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
/bookings/{bookingId}/fulfillments
fulfillmentIds
loop
[for each fulfillment]
GET /fulfillments/{fulfillmentId}

Graphical reservation

In some cases, additional information must be provided before or at the time of provisional booking in order to be taken into account, such as:

- Additional passenger identity information;
- Additional accommodation preferences regarding the accommodation, or its exact location.

While providing accommodation preferences is often 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 under the form of a boolean expression, referring to the passenger 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 :

```
passenger[<uuid>].details.firstName AND passenger[<uuid>].details.name
```

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

```
(passenger[0].details.firstName AND passenger[0].details.name AND
(passenger[0].details.eMail OR passenger[0].details.phone))
```

By parsing this structure, the API consumer is able to identify the elements that need to be filled-in to proceed. An initial version the grammar for required information is available for review.

The two types of information (accommodation preferences and passenger data updates) are both to be added in the POST /booking body:

- passenger information can be specified in the passengers array: `bookingRequest.selectedOffers[].passengers`
- seating preferences can be provided in `bookingRequest.selectedOffers[].placeSelections`

Reusable offers

A reusable offer is an offer that can be booked several times, as long as there is sufficient availability, in distinct bookings and for different, but equivalent, sets of passengers from those of the initial offer request.

The new set is considered equivalent when composed of passengers with a similar profile (same reductions and birth date entitling to the same product(s)).

In case of:

- booking a reusable offer for a new set of passengers, all passengers' attributes need to be be fully specified in the `POST /booking` body, except the `id` that is always generated by the server.
- booking an offer, reusable or not, for the same set of passengers of the offer request, all properties are updatable except:
  - `id`
  - `externalReference`

Note however that updating a property can influence the eligible product in the offer (such as date of birth or reduction cards) may lead to the booking being rejected in case of incoherence.

It is the choice of the OSDM provider to declare offers as reusable, or not, in the reply to `POST /offers`. Reusable offers however should be favored whenever possible: while the OSDM provider retains control, it allows implementation of powerful business use-cases (e.g. a ticket machine at the station that could continue selling reusable offers while experiencing network issues and would then synchronize the sales upon connection re-establishement).

Reminder: the accommodation preferences can be found in the `reservationOptions` elements

```
(offer.fare|integratedReservation|reservation.placeSelection.reservationOptio
ns)
```

The passengers in the booking resources are also the same type of resources as the ones present in offers. However, they could be different resources, with different ids (the passengers references do remain unchanged).

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

Error Handling

- The requested reservation option is not available on this transport
- An invalid value is provided for a passenger property
- Referenced offer or offer part not found (offer 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

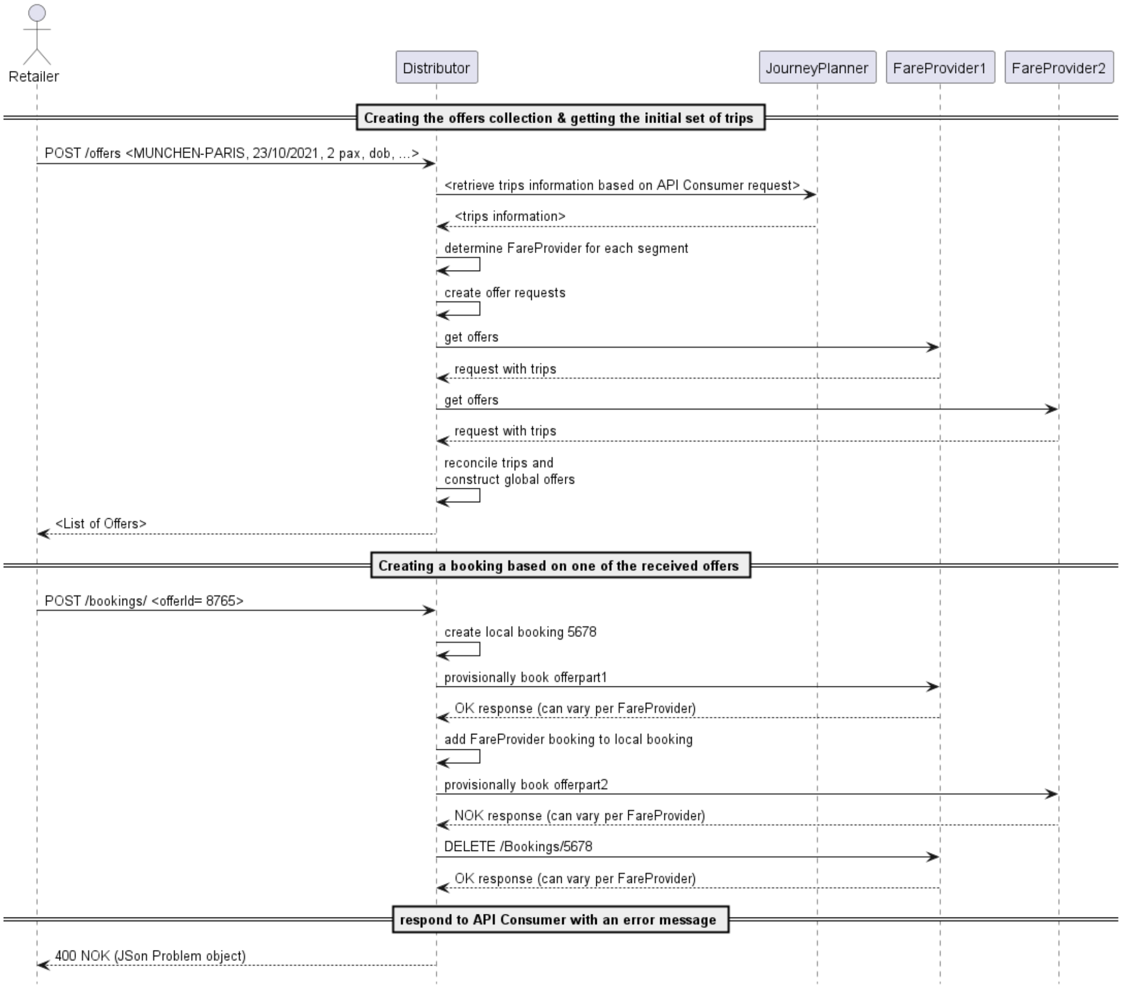
Provisionally Booking a Return Trip

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


Retailer
Distributor
JourneyPlanner
FareProvider1
FareProvider2
Creating the offers collection & getting the initial set of trips
POST /offers <MUNCHEN-PARIS, 23/10/2021, 2 pax, dob, ...>
<retrieve trips information based on API Consumer request>
<trips information>
determine FareProvider for each segment
create offer requests
get offers
request with trips
get offers
request with trips
reconcile trips and
construct global offers
<List of Offers>
Creating a booking based on one of the received offers
POST /bookings/ <offerId= 8765>
create local booking 5678
provisionally book offerpart1
OK response (can vary per FareProvider)
add FareProvider booking to local booking
provisionally book offerpart2
NOK response (can vary per FareProvider)
DELETE /Bookings/5678
OK response (can vary per FareProvider)
respond to API Consumer with an error message
400 NOK (JSon Problem object)

Handling Partial Success of Pre-Booking

As a **Retailer**, partial pre-booking is not expected. As a consequence all pre-booking operations are either fully successful or not executed at all.

However, a **Distributor** system may be configured in such ways that it is able to combine offers from different fare providers (via an OSDM-compliant API or not) and propose them in turn as one offer to its API to consumers, as one non dividable product or as a bundled pack.

Unfortunately, when the booking is attempted, the process may encounter errors leading to the booking failing with some of the sub-providers, 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 specifications, this situation needs to be rolled back. This can be done by cancelling the pre-bookings that were successful (on an OSDM 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.

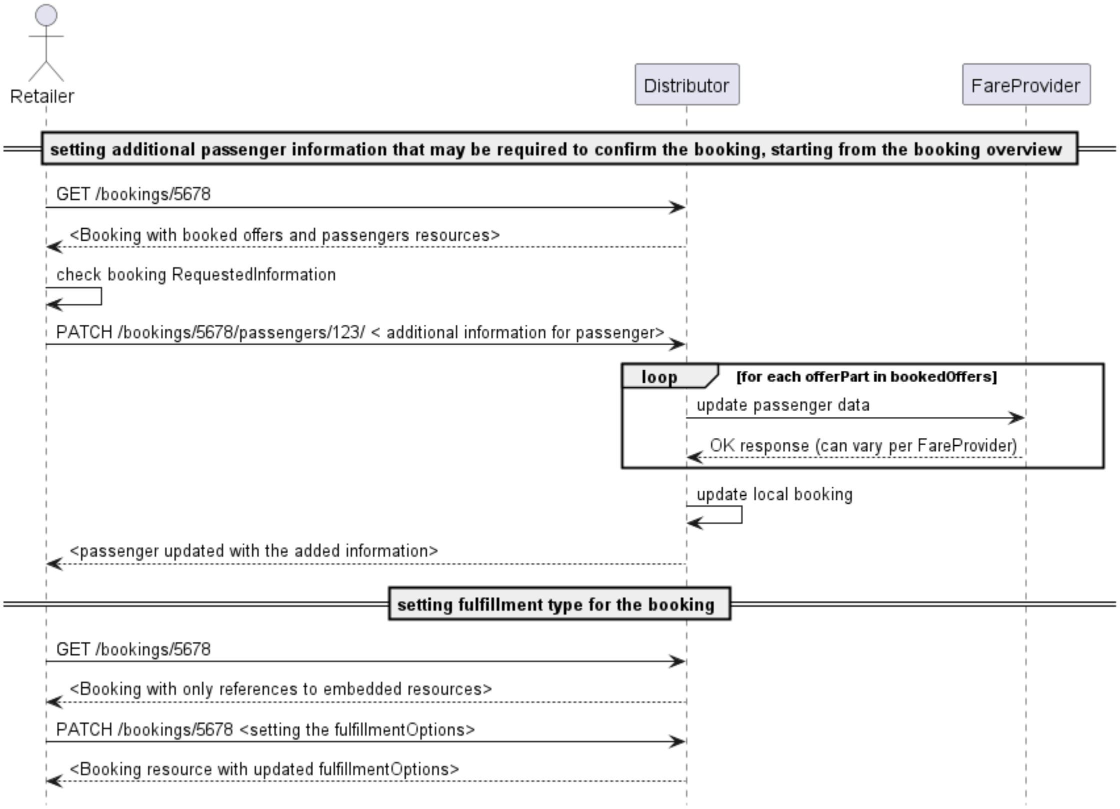
As it was the case with offers, during the booking process, some events may occur that are worth communicating to the API consumer, while they do not really constitute an error nor should interrupt the booking process. These events and situations can be communicated through the `Warning` messages:

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

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 OSDM 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 Fulfillment


Retailer
Distributor
FareProvider
setting additional passenger information that may be required to confirm the booking, starting from the booking overview
GET /bookings/5678
<Booking with booked offers and passengers resources>
check booking RequestedInformation
PATCH /bookings/5678/passengers/123/ < additional information for passenger>
loop
[for each offerPart in bookedOffers]
update passenger data
OK response (can vary per FareProvider)
update local booking
<passenger updated with the added information>
setting fulfillment type for the booking
GET /bookings/5678
<Booking with only references to embedded resources>
PATCH /bookings/5678 <setting the fulfillmentOptions>
<Booking resource with updated fulfillmentOptions>

Completing Booking for Confirmation and Fulfillment

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 sub-resources 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 fulfillment 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 fulfillment type and options. It is however recommended to the **Retailer** implementers to set a default value for these properties (especially if only one value is possible). Note that the choice of the fulfillment type & options may impact the requestedInformation. This property should therefore be re-evaluated whenever the fulfillment type is modified (both on the provider and on the consumer side).

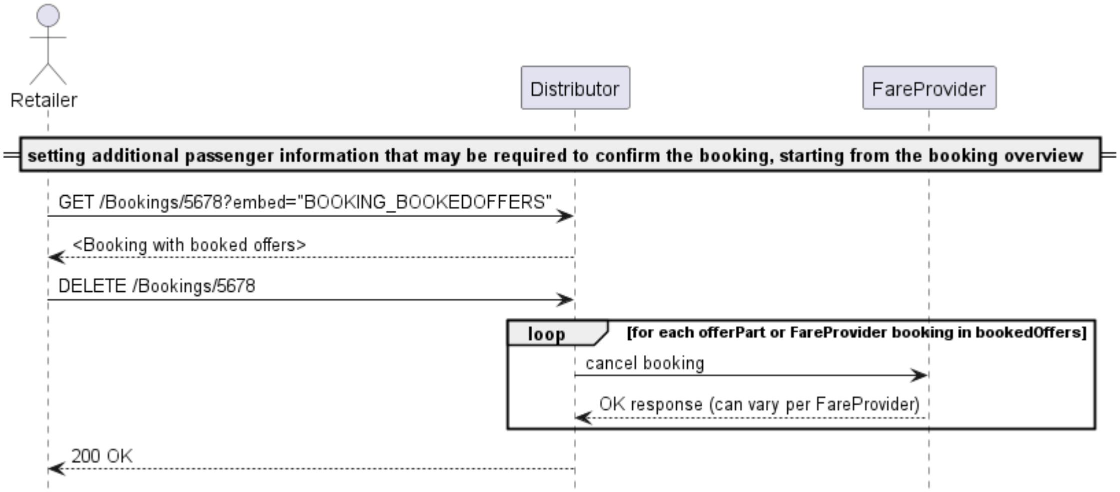
Notes

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

- An 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


Retailer
Distributor
FareProvider
setting additional passenger information that may be required to confirm the booking, starting from the booking overview
GET /Bookings/5678?embed="BOOKING_BOOKEDOFFERS"
<Booking with booked offers>
DELETE /Bookings/5678
loop
[for each offerPart or FareProvider booking in bookedOffers]
cancel booking
OK response (can vary per FareProvider)
200 OK

Cancel a Not Confirmed Booking

In case a pre-booked 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 **Retailer** 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 /bookings` for a given booking, the **Retailer** 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 /bookings` can also be used to clean-up the bookings on sub-providers where the pre-booking succeeded and who support the OSDM protocol.

Regardless of whether the cancel occurred through an explicit `DELETE /bookings` 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 an explicit info on the cancelled status of the booking. The choice of the duration of that grace period is left to the implementor.

Error handling

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

Payment information and Payment Vouchers

OSDM does not currently handle the payment process directly, which means that payment from the customer has to be taken by the distributor outside of this API.

It is necessary for the distributor or fare provider to know about certain aspects of the payment, e.g. the method of payment (e.g. Invoice, Cash, Non-Cash methods like Credit/Debit cards or Direct Debit).

It is also necessary to be able to use Payment Vouchers, such as those that are given by distributors or fare providers in response to Passenger Rights Claims, or that are used as means of refund on certain tariffs (e.g. Deutsche Bahn Sparpreis).

As a Payment Voucher is a kind of payment, the handling of these goes together in the API.

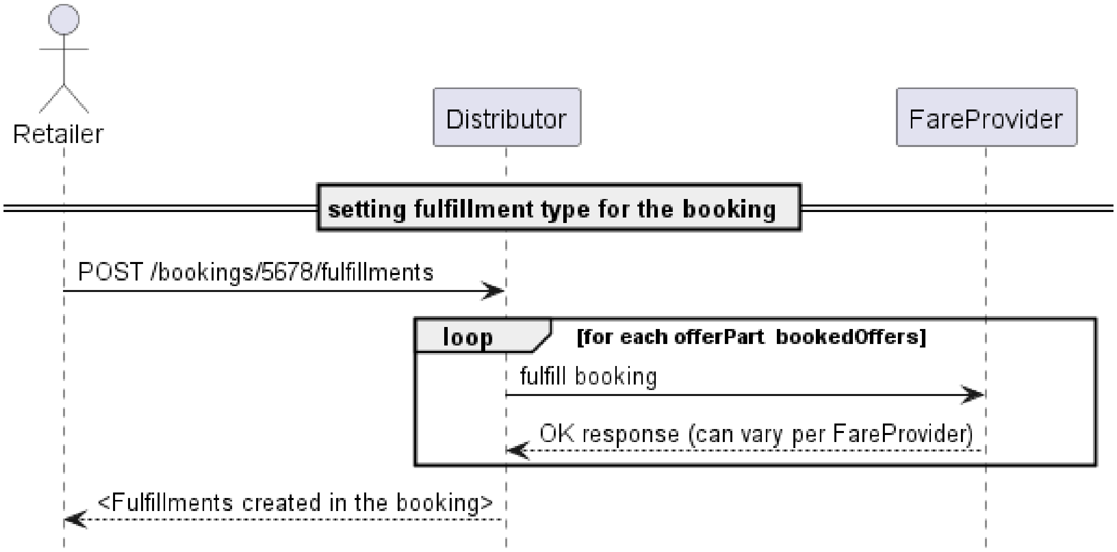
The process for a booking which uses one or more Payment Vouchers, and that specifies the means of payment, is as follows:

1. Create the preliminary booking by using the `POST /bookings` endpoint
2. Add any Payment Vouchers by using the `PATCH /bookings` endpoint and filling out the payments information only for the Payment Vouchers presented
3. This will return a preliminary booking structure which has the payment information for these vouchers added, including the value of the vouchers. Should the value of the vouchers exceed the value of the booking, an Ancillary Offer will have been added to the booking which represents a new voucher covering the overpayment.

4. Add the payment information for the balance of the booking (should there be any) in another `PATCH /bookings` call
5. The booking is now "balanced", i.e. the sum of all payments equals the sum of all offers

Confirmation and Fulfillment Processes

Fulfillment Process


Retailer
Distributor
FareProvider
setting fulfillment type for the booking
POST /bookings/5678/fulfillments
loop
[for each offerPart bookedOffers]
fulfill booking
OK response (can vary per FareProvider)
<Fulfillments created in the booking>

Fulfillment Process

The fulfillment 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:

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

The **Distributor** will have to

- confirm or fulfill the bookings towards all his providers
- retrieve the fulfillment details to populate its own booking responses (and databases, most likely)
- build the fulfillments
- update relevant booking properties as described above.

A fare provider will rarely provide real fulfillments as the distributors is responsible to create the fulfillments of the combined offers. However the fare provider has the option to provide fulfillment parts (e.g. visual security elements,..) to be integrated by the distributor in the combined fulfillment.

Fulfillment Process of Multi-Journey Products

Multi-journey products provide the right to travel on multiple journeys. These can be defined in the product as a specific number of travel days or of journeys. In order to use the travel right the fulfillment has to be enhanced with additional information on the intended use. e.g.:

Travel right	Required information
Pass for a number of travel days	The day of travel is provided and as a result the fulfillment for that selected day becomes fulfilled.
Travel right a number of journeys	The day or start time is provided and as result the fulfillment for one journey becomes fulfilled.
Travel right for a number of trips of a limited range	The time when the journey starts and the starting place needs to be added
Tickets for a selectable number of zones	the selected zones need to be provided

After the booking confirmation the fulfillment will be in state of `available` in case the fulfillment just requires additional selections to be come fulfilled. To fulfill it the patch endpoint of the API on the fulfillment needs to be used to select more options like travel date or zones. The fulfillment in the state available provides information on the selections to be made.

A fulfillment can be in state `on hold` if it is not yet available to be fulfilled. This corner case exists tickets providing a limited number of trips per time (e.g. 5 trips per week) and the number has already been used for the current time range.

Fulfillment can be in state expired if they had been available but were not requested in time.

Error handling

In the confirmation and fulfillment process, the following issues can arise:

- Unknown error on provider side
- Missing information in the booking
- No fulfillment 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 **Retailer** (combining offers of its sub-providers on request of its API consumer) has several options to handle the situation:

In all cases, the middle **Retailer** obviously has the option of proactively retrying to confirm on OSDM sub-providers 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 **Retailer** 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 **Retailer**, it should first ensure that the operation is indeed supported for all offer parts in the booking (whether the **Retailer** is hosting those or they are coming from sub-providers). Indeed, OSDM does 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)

Interlude: Requested Information per Process Step

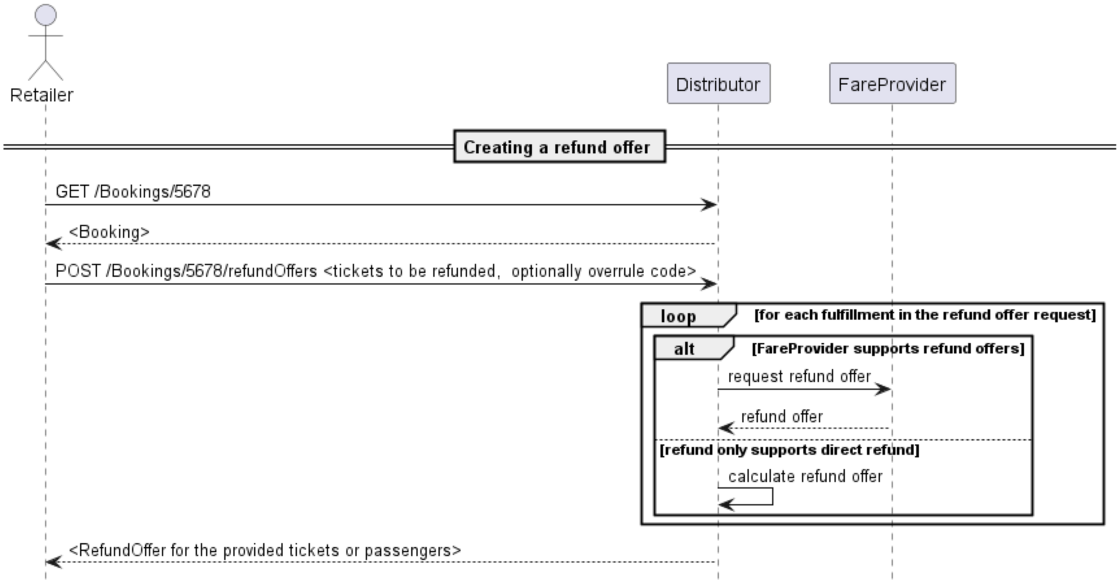
These are the required information needed per process step for major parties

<table>
<tr><th>Distributor</th><th>Pre-booking Step</th><th>Booking Step</th></tr>
<tr><td><b>Bene</b></td><td></td><td>firstName and lastName</td></tr>
<tr><td><b>DB</b></td><td>In general one firstName and name, regardless of the number of passengers. In case of regional trains, however, all names and sur names are needed, unless printed on security paper.</td><td></td></tr>
<tr><td><b>öBB</b></td><td>Both firstName and lastName are needed. dateOfBirth date may be needed. Some reduction cards require the number to be provided at pre-booking time, in order to be pre-checked. In other cases, the cards are simply checked on-board phoneNumber or eMail (once per order - as contact information)</td><td>phoneNumber or eMail (once per order - as contact information)</td></tr>
<tr><td><b>RENFE</b></td><td>Per passenger: firstName, lastName, surname document type and identity document (DNI, NIE or passport). A phoneNumber or eMail.</td><td>Per passenger: firstName, lastName, surname document type and Identity document. (DNI, NIE or passport) A phoneNumber or eMail.</td></tr>
<tr><td><b>SBB</b></td><td>Per passenger: name and first name and dateOfBirth. Additional sales parameters for some products such asphoneNumberoreMailfor reservations.
|
| | **SJ** | Todo
|
| | **SNCF** |dateOfBirthis mandatory, a fake date can be used at offer time, but the real one must be provided at pre-booking time
|
| | **Eurostar/Thalys** |firstNameandlastName`</td><td>Thalys loyalty card number</td></tr>
</table>

After Sales Processes

Refund

Request a Refund Offer

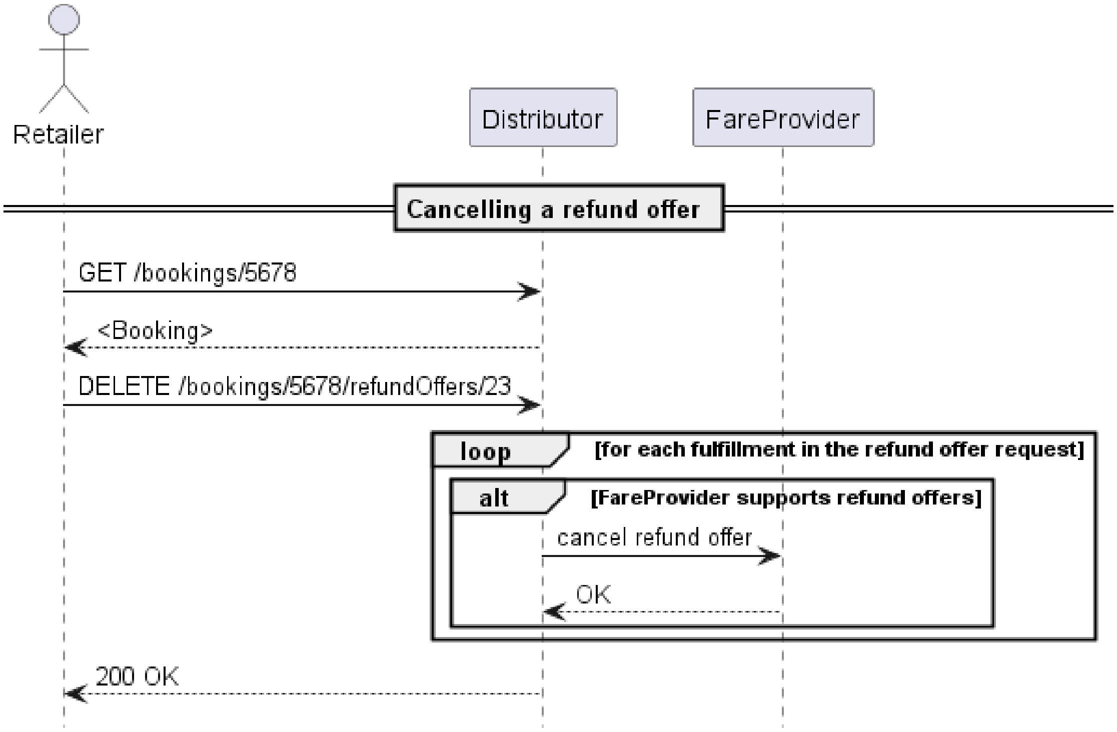

Retailer
Distributor
FareProvider
Creating a refund offer
GET /Bookings/5678
<Booking>
POST /Bookings/5678/refundOffers <tickets to be refunded, optionally overrule code>
loop
[for each fulfillment in the refund offer request]
alt
[FareProvider supports refund offers]
request refund offer
refund offer
[refund only supports direct refund]
calculate refund offer
<RefundOffer for the provided tickets or passengers>

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.

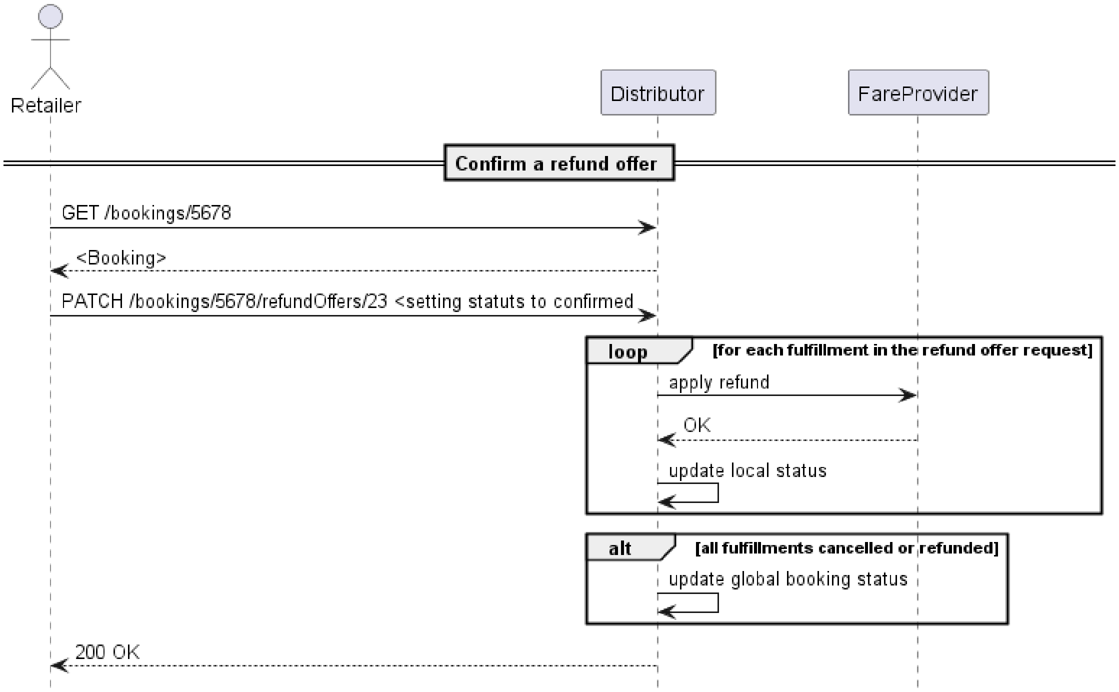
In order to perform a refund, the API consumer first has to create a refundOffer in the booking where the fulfillments to refund are located with a `POST refundOffer`. If the set of fulfillments provided is a valid set for refund, the operation creates a refundOffer that contains the information that 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 Offer


Retailer
Distributor
FareProvider
Cancelling a refund offer
GET /bookings/5678
<Booking>
DELETE /bookings/5678/refundOffers/23
loop
[for each fulfillment in the refund offer request]
alt
[FareProvider supports refund offers]
cancel refund offer
OK
200 OK

Cancel a Refund Offer

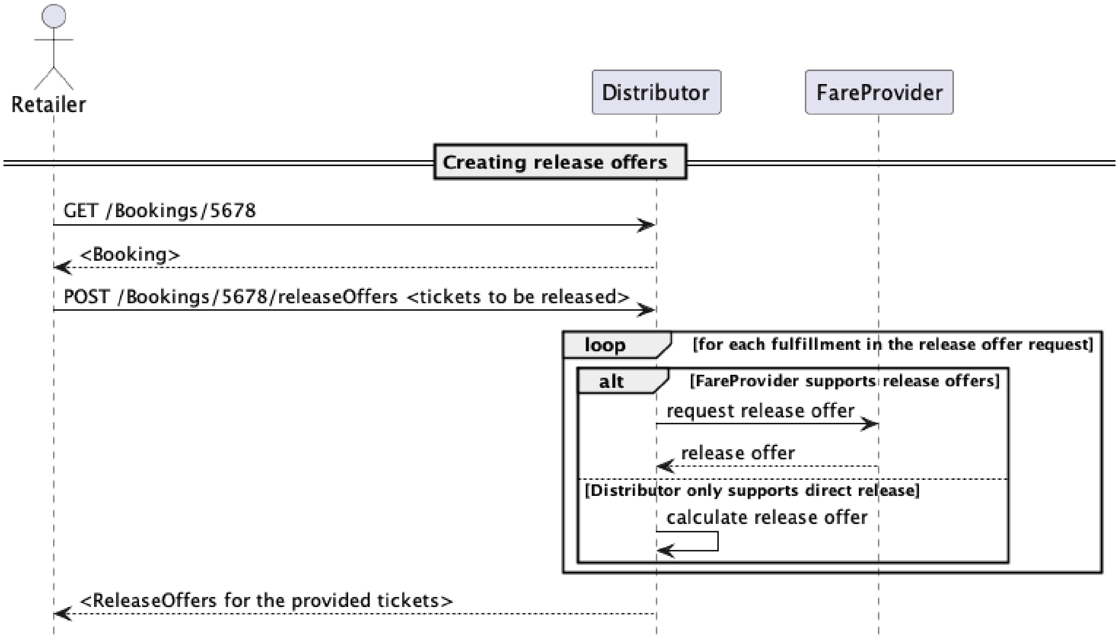
Confirm a Refund Offer


Retailer
Distributor
FareProvider
Confirm a refund offer
GET /bookings/5678
<Booking>
PATCH /bookings/5678/refundOffers/23 <setting statuts to confirmed
loop
[for each fulfillment in the refund offer request]
apply refund
OK
update local status
alt
[all fulfillments cancelled or refunded]
update global booking status
200 OK

Confirm a Refund Offer

Release a Booking

Request a release Offer

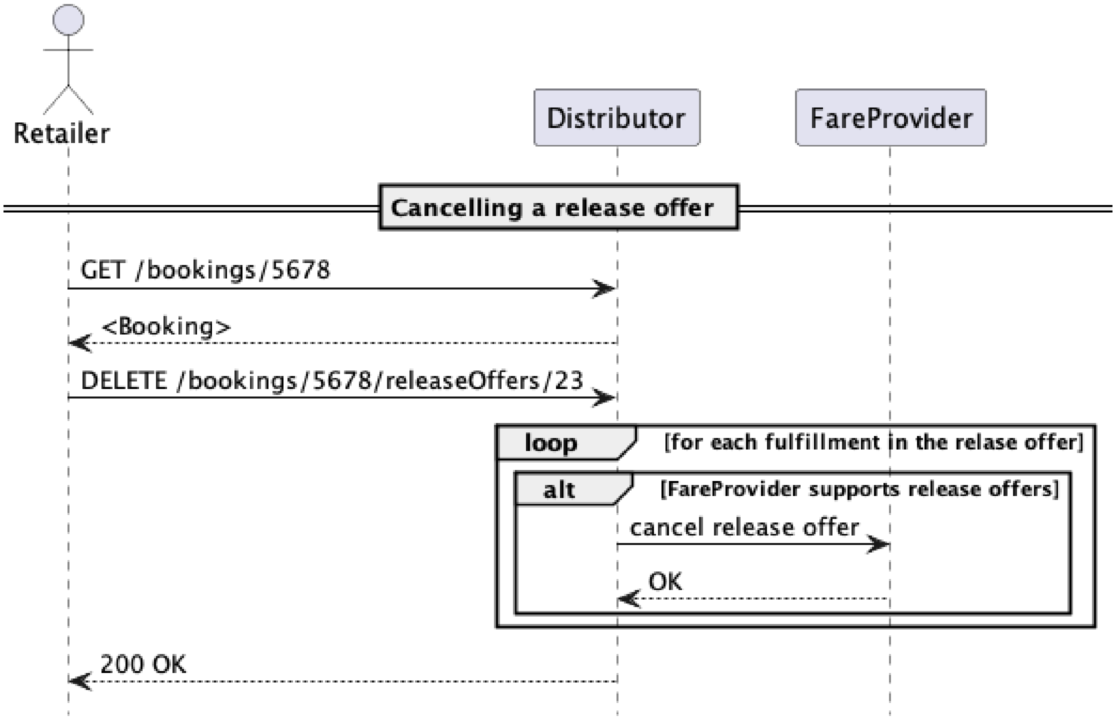

Retailer
Distributor
FareProvider
Creating release offers
GET /Bookings/5678
<Booking>
POST /Bookings/5678/releaseOffers <tickets to be released>
loop
[for each fulfillment in the release offer request]
alt
[FareProvider supports release offers]
request release offer
release offer
[Distributor only supports direct release]
calculate release offer
<ReleaseOffers for the provided tickets>

Request a Release Offer

The release of a ticket is an intermediate step towards a refund. The release invalidates the admission and frees resources such as reserved seats. It does not refund the money to the customer as this is done later on at the retailer. This intermediate step allows other parties to initiate the refund process (e.g. one of the involved carriers) and to reuse the resources. The customer benefits as the time for calculating the refund amount wll be the time of the release.

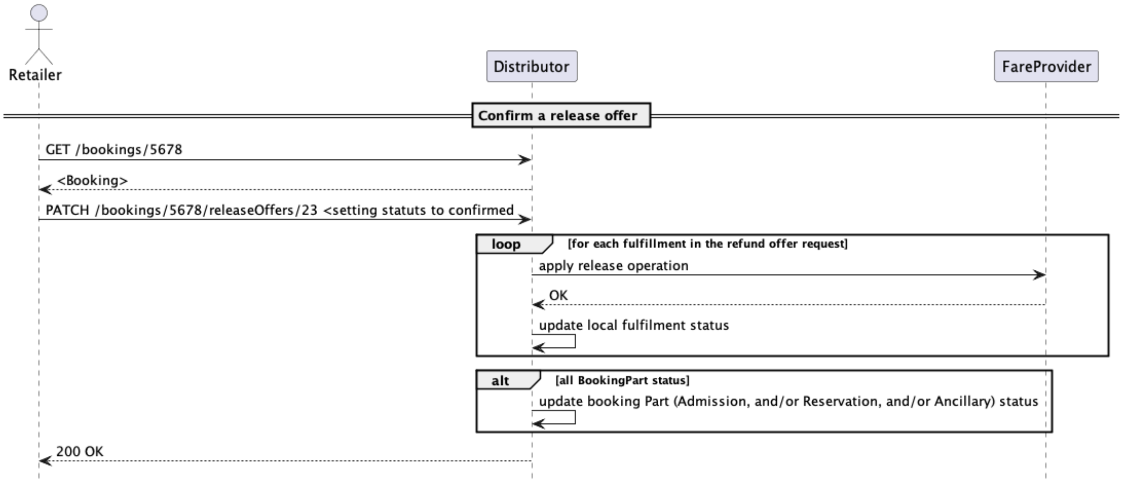
The process of releasing a ticket is similar to the refund process. A release offer is requested and needs to be confirmed to be applied.

Cancel a Release Offer


Retailer
Distributor
FareProvider
Cancelling a release offer
GET /bookings/5678
<Booking>
DELETE /bookings/5678/releaseOffers/23
loop
[for each fulfillment in the relase offer]
alt
[FareProvider supports release offers]
cancel release offer
OK
200 OK

Cancel a Release Offer

Confirm a Release Offer


Retailer
Distributor
FareProvider
Confirm a release offer
GET /bookings/5678
<Booking>
PATCH /bookings/5678/releaseOffers/23 <setting statuts to confirmed
loop
[for each fulfillment in the refund offer request]
apply release operation
OK
update local fulfilment status
alt
[all BookingPart status]
update booking Part (Admission, and/or Reservation, and/or Ancillary) status
200 OK

Confirm a Release Offer

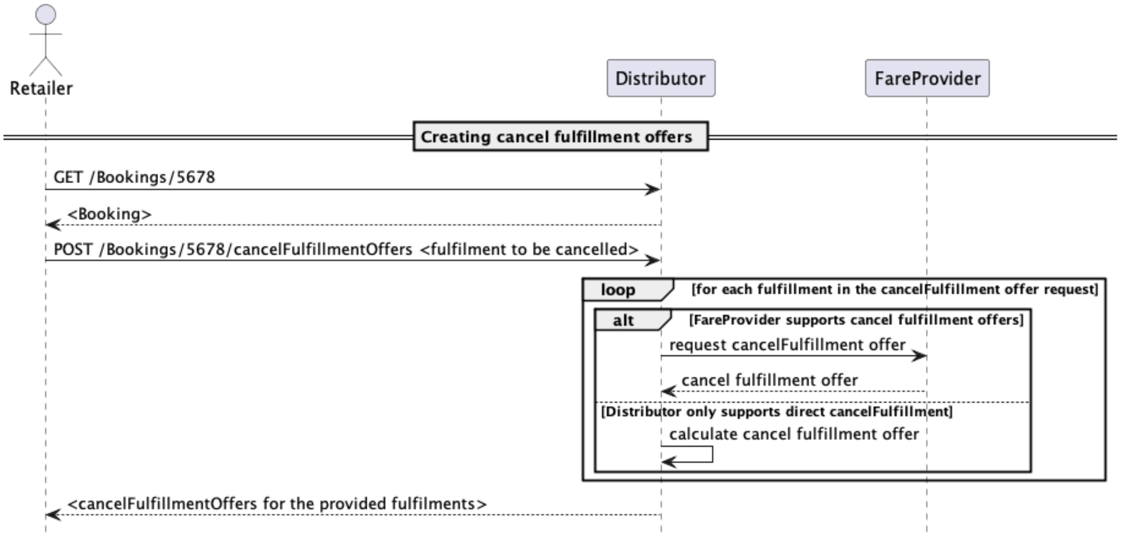
Partial Refund

Partial refunds of passengers and booking parts included in one fulfllment (Collective ticketing) are possible with version 3.2 onwards. The parts to be refunded need to be specified in the RefundSpecification.

A partial refund will result in new fulfilments after the confirmation of the refund offer and booking.

Cancel Fulfillment

Cancel Fulfillment request

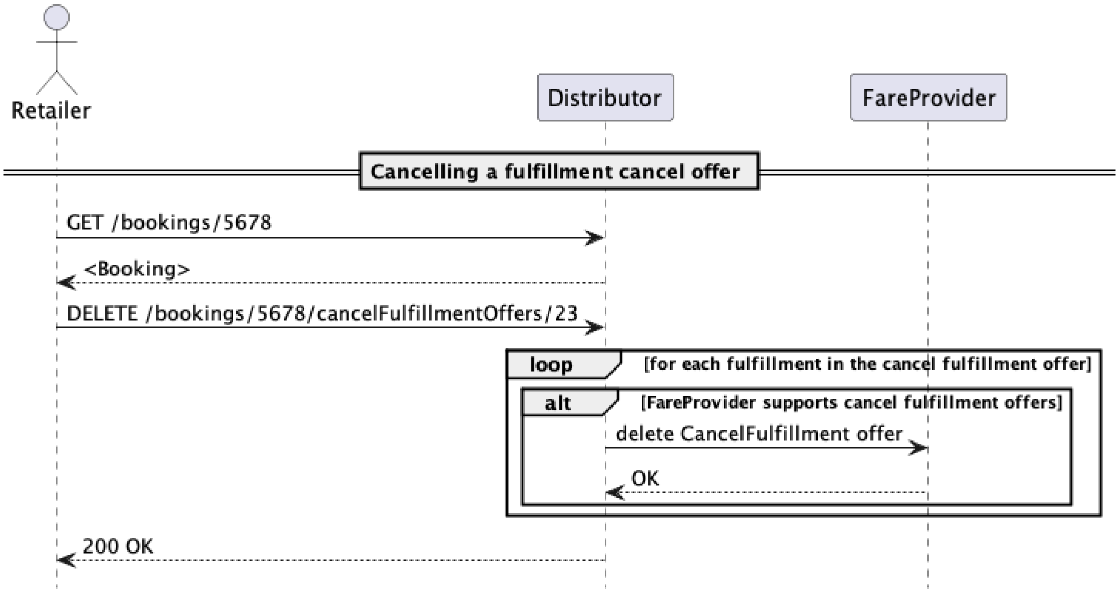

Retailer
Distributor
FareProvider
Creating cancel fulfillment offers
GET /Bookings/5678
<Booking>
POST /Bookings/5678/cancelFulfillmentOffers <fulfilment to be cancelled>
loop
[for each fulfillment in the cancelFulfillment offer request]
alt
[FareProvider supports cancel fulfillment offers]
request cancelFulfillment offer
cancel fulfillment offer
[Distributor only supports direct cancelFulfillment]
calculate cancel fulfillment offer
<cancelFulfillmentOffers for the provided fulfilments>

Request a cancelFulfillment Offer

A fulfillment can be cancelled and regenerated. This might be necessary in case of fulfillments linked to physical items (secure paper, phones or cards). The cancelled fulfillment can be recreated later on.

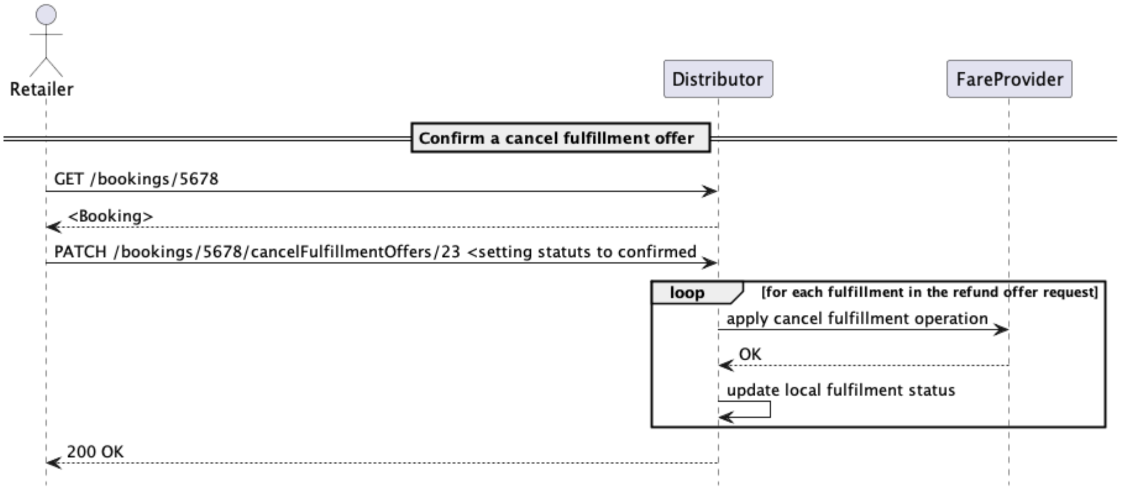
To cancel a fulfillment a cancel fulfillment offer needs to be requested. This offer can be confirmed to delete the fulfillment.

Cancel a CancelFulfillment Offer


Retailer
Distributor
FareProvider
Cancelling a fulfillment cancel offer
GET /bookings/5678
<Booking>
DELETE /bookings/5678/cancelFulfillmentOffers/23
loop
[for each fulfillment in the cancel fulfillment offer]
alt
[FareProvider supports cancel fulfillment offers]
delete CancelFulfillment offer
OK
200 OK

Cancel a Cancel a cancelFulfillment Offer

Confirm a CancelFulfillment Offer

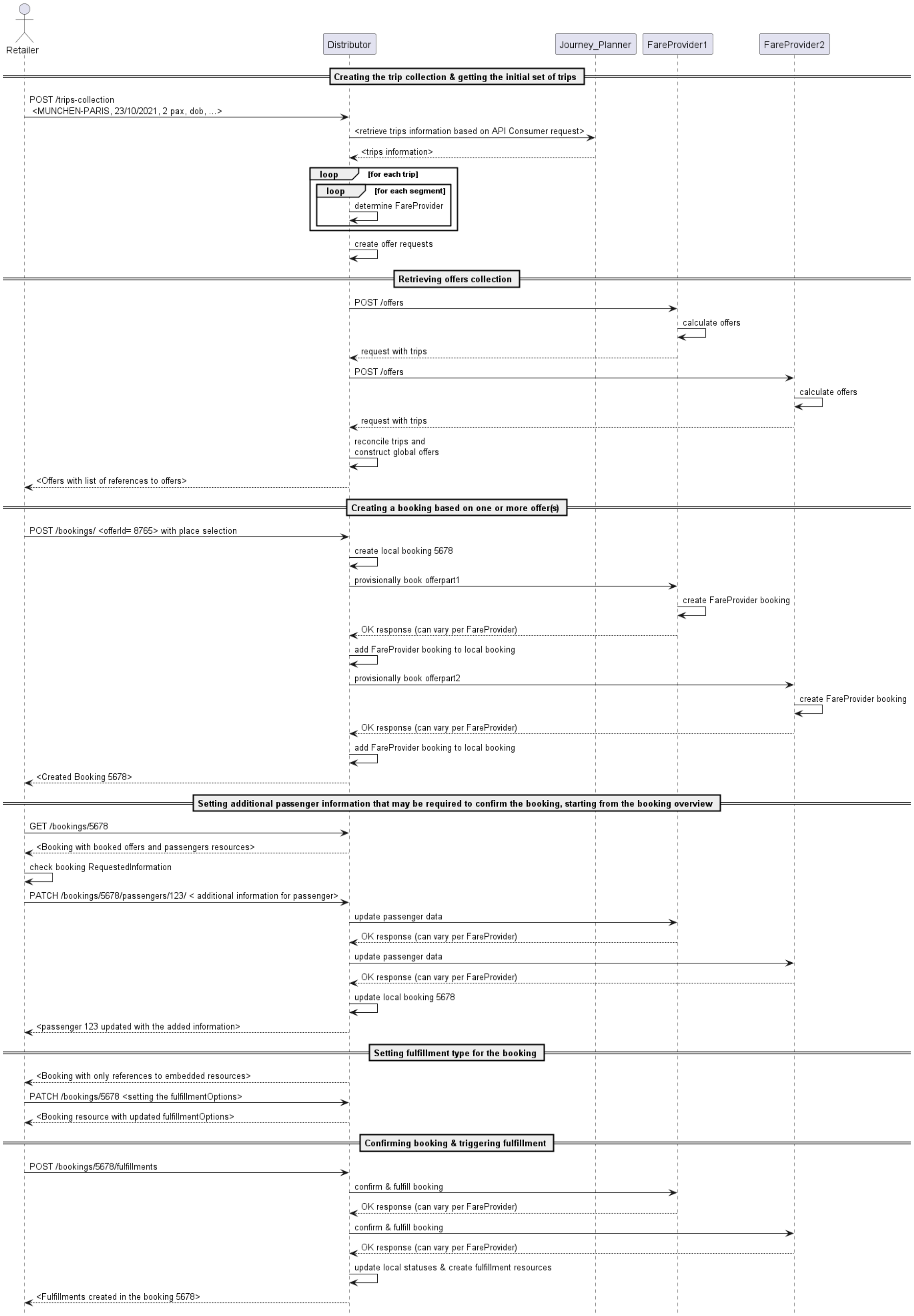

Retailer
Distributor
FareProvider
Confirm a cancel fulfillment offer
GET /bookings/5678
<Booking>
PATCH /bookings/5678/cancelFulfillmentOffers/23 <setting statuts to confirmed
loop
[for each fulfillment in the refund offer request]
apply cancel fulfillment operation
OK
update local fulfilment status
200 OK

Confirm a CancelFulfillment Offer

On Hold Bookings

An unconfirmed booking will expire after the time limit of the booking. An extension of the time limit can be requested as a OnHold-Offer. The offer can be requested and needs to be confirmed to extend the time limit. The OnHold offer might be subject to a fee.

Example End-to-end Interaction


Retailer
Distributor
Journey_Planner
FareProvider1
FareProvider2
Creating the trip collection & getting the initial set of trips
POST /trips-collection
<MUNCHEN-PARIS, 23/10/2021, 2 pax, dob, ...>
<retrieve trips information based on API Consumer request>
<trips information>
loop [for each trip]
loop [for each segment]
determine FareProvider
create offer requests
Retrieving offers collection
POST /offers
calculate offers
request with trips
POST /offers
calculate offers
request with trips
reconcile trips and
construct global offers
<Offers with list of references to offers>
Creating a booking based on one or more offer(s)
POST /bookings/ <offerId= 8765> with place selection
create local booking 5678
provisionally book offerpart1
create FareProvider booking
OK response (can vary per FareProvider)
add FareProvider booking to local booking
provisionally book offerpart2
create FareProvider booking
OK response (can vary per FareProvider)
add FareProvider booking to local booking
<Created Booking 5678>
Setting additional passenger information that may be required to confirm the booking, starting from the booking overview
GET /bookings/5678
<Booking with booked offers and passengers resources>
check booking RequestedInformation
PATCH /bookings/5678/passengers/123/ < additional information for passenger>
update passenger data
OK response (can vary per FareProvider)
update passenger data
OK response (can vary per FareProvider)
update local booking 5678
<passenger 123 updated with the added information>
Setting fulfillment type for the booking
<Booking with only references to embedded resources>
PATCH /bookings/5678 <setting the fulfillmentOptions>
<Booking resource with updated fulfillmentOptions>
Confirming booking & triggering fulfillment
POST /bookings/5678/fulfillments
confirm & fulfill booking
OK response (can vary per FareProvider)
confirm & fulfill booking
OK response (can vary per FareProvider)
update local statuses & create fulfillment resources
<Fulfillments created in the booking 5678>

Example End to End Interaction

Exchange

Requesting an exchange offer

Requesting an exchange offer is almost identical to requesting a standard offer. The only difference in the request is that the fulfillment that the API consumer wants to exchange, and an overrule code if relevant, are also provided.

Replacement of lost tickets and cards

The replacement is used to replace physical cards and tickets. There is no replacement for electronic tickets or anonymous tickets.

Requesting a replacement for a lost ticket

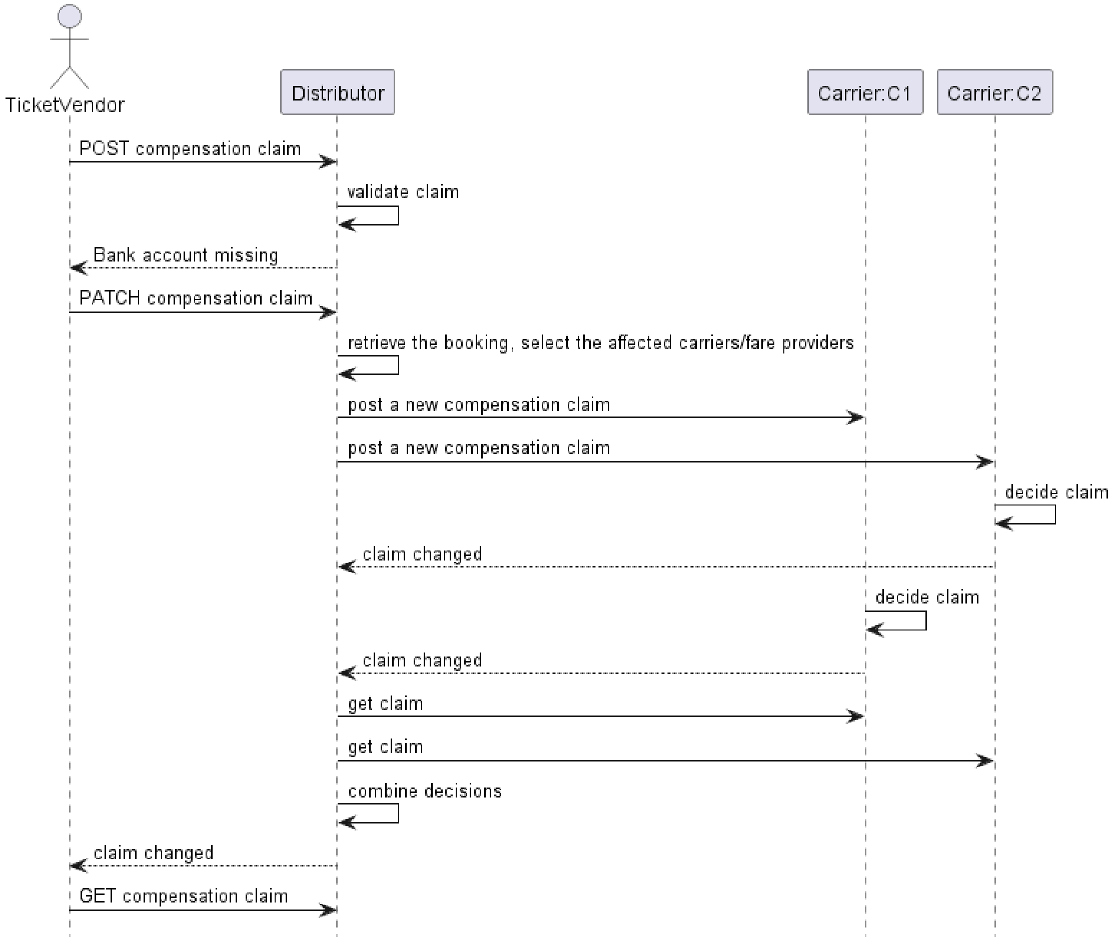
The replacement is requested similar to the request for a non-trip based offer. The search tags must include the key word CARD_LOST or TICKET_LOST. The provider will ask for the required data of the lost card or ticket to be provided with the passengers card data (card number).

The offer for a replacement returned might include a fee. The replacement offer needs to be accepted and booked the same way as a usual offer.

Complaints

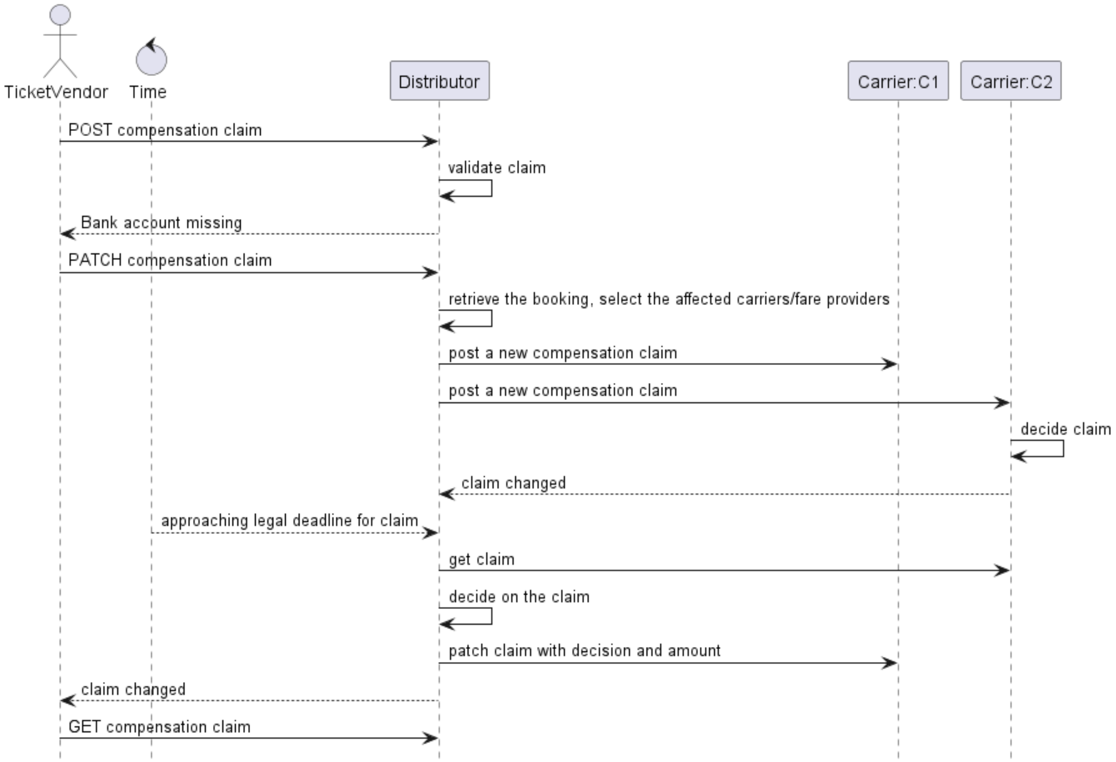
Complaints can be provided on behalf of a passenger. Complaints might concern a delay of a train or a service degradation on the journey. The handling of complaints is subject to the EU PRR and COTIV where minimal compensation amounts and time lines for the decision of a claim are defined. According to PRR the customer can decide whether he wants to be compensated by money or would accept vouchers.

The handling of a claim is an asynchronous process, where the claim is placed and decided by the carriers/fare provides involved later-on.



Complaint

As the distributor is usually also involved as a carrier and then responsible to keep the legal time lines he can decide to compensate and inform the fare provides/carriers on his decision if the time line would otherwise can not be held.



Complaint

Reimbursement

Customers who have bought a ticket which allows reimbursement and which have not traveled or traveled partially only can claim to be reimbursed. The customer must prove that he has not or only partially used the ticket. A partial use might be a only a part of the trip was traveled or only some of the travelers were traveling or a combination of both.

The non use of a ticket might be proven by documents that were provided to the passenger on a train or at a station. This prove can also be provided in electronic form by a carrier or TCO (UIC IRS 909181-4 / TAP-TSI B.14).

The customer must be able to make the reimbursement claim via a retailer to the distributor which needs to request the reimbursement from the involved fare providers / carriers.

The reimbursement process is very similar to the complaint process where instead of a complaint a reimbursement request is created. If the reimbursement request is valid the special overrule code `TICKET_UNUSED` can be used in the refund process to refund otherwise non-refundable bookings.

A Gentle Introduction to OSDM

Introduction

OSDM (Open Sales and Distribution Model) defines an API to enable and simplify the sale of transport products. The API allows Retailers to access transport products provided by distributors. It also allows distributors to access transport product bricks provided by carriers or fare providers to build combined transport products. The aim of OSDM is to provide a simple API to access required information online, however OSDM also rovides an offline data exchange of fares.

The API covers the full sales process including time table / offer serch, prebooking and booking, refund, exchange including special processes to handle delays, change of material and compensation.

OSDM covers scheduled multimodal transport services (trains, busses, trams, ...). On Demand services are currently covered with some functional limitation.

The OSDM API is rich and can be overwhelming at start. Reason for its feature richness (leading to its complexity) is the fact that it aims to cover all possible public transportation products and distribution processes within Europe and beyond.

The products can be split roughly split into three categories:

- *admissions*: the right to ride on a train, resulting in a ticket.
- *reservations*: seat or couchette or bed in case of night trains
- *ancillary*: ancillary services such WIFI, 3-Menu plate

Between these categories, relationships are modelled to express whether it is optional or mandatory to have a certain seat reservation or ancillary on a given vehicle. In an open system (e.g. Switzerland) having a reservation is optional. In contrast on a closed system (e.g. France) a seat reservation is a mandatory.

Booking a Ticket

We will start with booking a ticket, i.e. admission on a very simple trip: *Basel* - *Chur*. We assume there's no changing of trains, thus one segment only. We assume that the distributor is a Swiss website.

A segment has all the stops as well as information on the vehicle running on this segment. On this segment this is a ICE train operated by SBB.

- Step 1: Look up location codes

  To uniquely identify a location, stations etc. internally a code is used. Thus a helper service is provided that allows you to look up codes:

  ```
  POST /places
  ```

  with a body

```
{
  "placeInput": {
    "name": "Basel"
  }
}
```

As response, you get information on the location, e.g. its local name or its geo coordinates.

```
{
  "places": [
    {
      "id": "place-1",
      "name": "Basel SBB",
      "stopPlace": {
        "ref": "urn:uic:stn:8503000",
        "name": "Basel SBB"
      },
      "geoPosition": {
        "latitude": 47.547408,
        "longitude": 7.589548
      }
    }
  ]
}
```

Thus by calling the service you get the following UIC codes: *Basel*: 8500010 and *Chur*: 8509000.

- Step 2: Request Offers

Next, the simplest way to receive offers is to pass in *origin*, *destination*, *departureTime* as well as the *passenger's* date of birth by calling. The birth date is necessary to return e.g. senior fares where sensible.

```
POST /offers
```

with a body

```
{
  "tripSearchCriteria": {
    "origin": "urn:uic:stn:8500010",
    "destination": "urn:uic:stn:8503000",
    "departureTime": "<departure_timestamp>"
  },
  "anonymousPassengerSpecifications": [
    {
      "externalRef": "<external-passenger-ref>",
      "type": "PERSON",
      "dateOfBirth": "1970-01-01"
    }
  ],
  "offerSearchCriteria": {
    "currency": "CHF"
  }
}
```

As a result you get a set of offers, with exact information about the trip covered for the given set of passengers.

```
{
    "id": "C_0KK..",
    "offers": [..
    ],
    "trips": [..
    ],
    "anonymousPassengerSpecifications": [
        {
            "id": "passenger_1",
            "dateOfBirth": "1970-01-01",
        }
    ]
}
```

An offer is structured starting with an offer summary, followed by admission, reservations and ancillaries if available.

On this train a valid offer consists of an admission with a optional reservation. As no reduction card has been submitted, the second class is full price, thus 68.00 CHF. Other valid offers for this trip, e.g. for super safer fares (in CH: “Sparbillett”) or for first class are returned.

```
  {
    "id": "O_72g..",
    "offerType": "Offer",
    "offerSummary": {
        "minimalPrice": {
            "currency": "CHF",
            "amount": 6800,
            "scale": 2
        },
        "overallServiceClass": {
            "type": "BASIC"
        },
        "overallFlexibility": "FULLFLEX"
    },
    "isReusable": true,
    "coveredLegIds": [
        "leg-1"
    ],
    "availableFulfillmentTypes": [
        "ETICKET"
    ],
    "validFrom": "2021-08-30T00:00:00+00:00",
    "validUntil": "2021-08-31T00:00:00+00:00",
    "admissions": [
        {
            "id": "P_oLA..",
            "price": {
                "currency": "CHF",
                "amount": 6800,
                "scale": 2
             },
             "passengerSpecification": [
              {
                 "id": "<passenger_id>"
              }
              ],
              "validFrom": "2021-08-30T00:00:00+00:00",
              "validUntil": "2021-08-31T00:00:00+00:00",
              "refundable": "YES",
              "exchangeable": "YES",
              "products": [
                    {
                        "id": "SBB_POINT_TO_POINT",
                        "summary": "Point-to-point Ticket, Second
Class",
                        "code": "125",
                        "isTrainBound": false
                    }
               ]
            "isReservationRequired": false
        },..
    ],
    "reservations": [
        {
            "id": "P_WD1..",
            "price": {
                "currency": "CHF",
                "amount": 500,
                "scale": 2
             },
             "passengers": [
                 {
                    "id": "<passenger_id>"
                 }
              ],
              "validFrom": "2021-08-30T00:00:00+00:00",
              "validUntil": "2021-08-31T00:00:00+00:00",
              "numericAvailability": 20,
              "refundable": "NO",
              "exchangeable": "NO",
              "products": [
                    {
                        "id": "SBB_SEAT_RESERVATION",
                        "summary": "Seat Reservation",
                        "code": "PRODUCT_10000",
                        "isTrainBound": true
                    }
                ]
                "optionality": "OPTIONAL"
        },
        {
            "id": "P_WD1..",
            "price": {
                    "currency": "CHF",
                    "amount": 500,
                    "scale": 2
             },
             "passengers": [
                    {
                        "id": "<passenger_id>"
                    }
             ],
             "validFrom": "2021-08-30T00:00:00+00:00",
             "validUntil": "2021-08-31T00:00:00+00:00",
             "numericAvailability": 4,
             "refundable": "NO",
             "exchangeable": "NO",
             "products": [
                    {
                        "id": "SBB_BIKE_RESERVATION",
                        "summary": "Bike Reservation",
                        "code": "PRODUCT_10001",
                        "isTrainBound": true
                    }
            ]
            "optionality": "OPTIONAL"
        }
    ]
}
```

The number of available seat or bike place can be included in offers and expressed by the "numericAvailability" attribute.

If your overwhelmed by the numbers of offers you are getting, you can filter them by setting flexibilities, comfortClasses, offerPartType.

- Step 3: Booking Offers.

  Next your task is to display this offers in a meaningful way to your customer. Therefore every offer has a lot information about the underlying product, its comfort class, its flexibility etc. so that the customer can choose the offer best suited for its needs.

Once the customer has chosen an offer, you book it by calling

```
POST /bookings
```

with a body of

```
{
  "offers": [
    {
      "offerId": "<selected_offer_id>",
      "passengerRefs": ["<passengerRefs>"]
    }
  ],
  "passengerSpecifications": [
    {
      "externalRef": "<ext-person-ref>",
      "dateOfBirth": "1945-11-02",
      "type": "PERSON"
    }
  ]
}
```

The system now creates a booking for you, containing all the information about the trip, the passenger and the offers booked. The created booking is returned in state `PREBOOKED` with the offer structure being available as part of the `bookedOffer`.

```
{
    "booking": {
        "id": "28OD7DVM",
        "summary": "Booking number 28OD7DVM",
        "provisionalPrice": {..
        },
        "confirmedPrice": {..
        },
        "ticketTimeLimit": "2021-08-23T12:21:15+00:00",
        "fulfillmentType": "ETICKET",
        "bookedOffers": [
          {
            "id": "B_WyJ...",
            "offerSummary": {
            "minimalPrice": {
                "currency": "CHF",
                "amount": 500,
                "scale": 2
              }
            },
            "isReusable": false,
            "availableFulfillmentTypes": [
            "ETICKET"
            ],
            "trip": {..
            },
            "validFrom": "2021-09-02T00:00:00+00:00",
            "validUntil": "2021-09-02T00:00:00+00:00",
            "admissions": [..
            ],
            "reservations": [],
            "ancillaries": []
          }
        ],
        "fulfillments": [
          {
            "id": "28OD7DVM-001",
            "status": "CONFIRMED",
            "controlNumber": "28OD7DVM-001",
            "offerParts": [

"P_jekG6PnHWpW6dL8GPXLICWURJT9nh5ch5kRAjpMElgCvT8LTTNkWM6lOJsTaAUsq"
            ]
          }
        ]
    }
}
```

- Step 4: Finalize the booking and get the fulfillments

  To finalize the booking and handout a ticket the last call is to get fulfillments of this ticket. Fulfillments are a general term to describe physical tickets as well as non-digital one such as Apple’s passbook or a simple code only.

  `POST /bookings/{booking_id}/fulfillments`

  The fulfillments can be simply accessed by calling:

  `GET /bookings/{booking_id}/fulfillments`

  For example, you can use the included URL to download the PDF.

```
{
  "fulfillment": [
    {
      "id": "4ES36OIU-001",
      "status": "FULFILLED",
      "controlNumber": "4ES36OIU-001",
      "offerParts": ["P_jekG6..."],
      "fulfillmentDocuments": [
        {
          "medium": "PDF_A4",
          "type": "TICKET",
          "downloadLink": "https://ticket.osdm-demo-
test.cloud.sqills.com/ticket/4ES36OIU-001_210823145615241.png"
        },
        {
          "medium": "PDF_A4",
          "type": "TICKET",
          "downloadLink": "https://ticket.osdm-demo-
test.cloud.sqills.com/ticket/4ES36OIU_PASSENGERS1_210823145615245.pdf"
        }
      ]
    }
  ]
}
```

That's it.

What is in a Booking?

Once a booking has been created, you can get detailed information about it calling the booking resource:

```
GET /bookings/{bookingId}
```

The response gives you detailed information about the state and the price of the booking as well as the trip, the passengers and - if in state fulfilled - its fulfillments. Spend some time understanding it structure.

Reserving a Seat

In the offer response of Step 2 you will see, that additionally to a admission, there are also seat reservations returned on this trip. The seat reservation has an `"optionality": OPTIONAL`, thus you don't need to book it. Thus you call

```
POST /bookings
```

with a body of

```
{
    "offers": [
        {
            "offerId": "<selected_offer_id>",
            "optionalReservationIds": [
                "<selected_reservation_id>"
            ],
        }
    ],
    "passengerSpecifications": {...}
}
```

As you can see, in the most simple case you just have to add the id of the selected reservation offer in the booking request. The inventory system will then choose a seat for you. In this case, the place 25 in coach 3 was booked.

```
{
    "booking": {
        "bookedOffers": {
            "offerId": "offer-1",
            "reservations": [
                {
                    ..
                    "status": "BOOKED",
                    "placeAllocation": {
                        "reservedPlaces": [
                            {
                                "id": "S_7H5nJ5IEwipH4dvV2UNBRQ==",
                                "passengerId": "<passenger-id>",
                                "vehicleNumber": "IC 565",
                                "coachNumber": "3",
                                "placeNumbers": "25"
                            }
                        ]
                    }
                }
            ]
            ..
        }
    }
}
```

Traveling a Bit Further

Assuming you are hungry and want to see a scenic landscape, let's change the trip slightly: We travel from *Basel* to *Chur* and then switch train from *Chur* to *Brig* on a touristic train running through the wonderful Swiss alps. To find the UIC code of *Brig* you can again use `GET /places?matchValue=Brig` and set the UIC code of *Chur* as a via station.

In the offer request the trip now consists of two segments. The second segment is a Panorama express operated by RhB. As there's high demand on these touristic trains, seat reservation is a must. This is indicated by a reservation offer with an mandatory reservation (`"optionality": "MANDATORY"`).

Additionally, on this train there are ancillary services available. You can choose between a 3-menu plate or a plate of the day. To book one of them, you just pass in the id in the booking step.

```
{
    "offers": [
        {
            "offerId": "<selected_offer_id>",
            "optionalReservationIds": [
                "<selected_reservation_id>"
            ],
            "optionalAncillaryIds": [
                "<selected_ancillary_id>"
            ]
        }
    ],
    "passengerSpecifications": {...}
}
```

Refunding an Offer

If a customer wants to refund its ticket, the flow is a two step process analogously to the booking flow. If a booking (i.e. its underlying product) is fully refundable you will get the full amount refunded. On the other hand, if the booking is non refundable an offer of 0 CHF is returned.

- Step 1: Request a refund offer.

  The get a refund offer you POST `/bookings/{booking_id}/refundOffers`

  with ids of the fulfillments to be cancelled (You can look up the fulfillmentIds of a booking by GET `/bookings/{bookingId}`)

  ```
  {
    "fulfillmentIds": ["<fulfillment_id>"],
    "overruleCode": "PAYMENT_FAILURE",
    "refundDate": "<refund_timestamp>"
  }
  ```

  In the response you get a collection of refund offers including refund fees and the refundable amount.

- Step 2: Accept the refund offer.

  To accept the refund offer you patch the PATCH `/bookings/{booking_id}/refundOffers/{refund_id}`

  with a body of:

  ```
  {
    "status": "CONFIRMED"
  }
  ```

  Again, that's it

Where to Go From Here

This short introduction should help you getting started and assure you that OSDM is simple to use. While it's simple to use it's powerful to handle all kind of night trains, thru fares, passes and complex exchange processes on yielded products.

To fully understand OSDM we recommend you to have a look at the specification, especially the model and the processes pages, especially state model of a `booking` and a `fulfillment`.

Another good start is to study the **API** itself. Or you can ask the OSDM technical group for an introduction.

FAQ

Why is there no pre-booking resource?

In some countries, super saver fares are loaded as promotions into the system at a given date, which can lead to millions of request for offers within a short period as everybody tries to get the cheapest tickets. Storing hundreds of millions of offers or pre-bookings server side becomes a challenge.

To deal with such scenarios, the protocol is stateless between offer and booking. Stateless in the sense that the state is held on the client and not on the server side. How can this be achieved? The trick is to encode all necessary information about an offer into the offerId and thus implicitly stored on the server side.

Thus once an offerId is chosen, the offer is generated on the fly on the server side, checked for consistency and then a booking in the state `PREBOOKED`is created.

If your system thus not support this magic, you probably don't need it and can of course work cache the offers on the server side for a given time and return this information as part of the offer information.

Why are you using POST when there should be a GET?

It would be in the spirit of REST to search for `GET /bookings?firstName=John&lastName=Doe` to return all bookings of John Doe. As such a call would be logged by any involved system, this collection of data violates GDPR regulations. We have reviewed all our services and decided to us POST in such cases and thus support privacy by design.

How are IRTs modeled?

IRT stands for *integrated rail ticket* and stands for a group of products where the admission includes a seat as reservation. For example given, IRT are available on Eurostar or Thalys trains which basically act as any airplane where you can only enter if you have a reserved seat.

The way they are modelled in OSDM are as *admission* with an *included reservation*, i.e., the seat reservation have an `"optionality": INCLUDED`. The admission with the included reservation have to be treated atomically by any implementation.

How many seats are available on the train?

The availability on a given train is bound to the products available on the train. I.e. the number of available bike reservations on a train is expressed on the offers of type “Bike Reservation” by the attribute `"numericAvailability": 23`. If no bike reservation places are available, no offer of this type is returned.

This feature is optional to support by implementors, some railways decide not give insight into the numeric availability of product (especially super saver fares) on their trains.

When to pass in which passenger attributes?

We take special care not to violate passenger personal rights and build in privacy by design. Thus we collect as little information as possible at every step, i.e. only the attributes absolutely necessary to fulfill the operation are gathered. To indicate which information is needed there is a `requestedInformation` attribute which express the information needed expressed in small DSL for requested information.

Advanced Topics

Choosing Your Seat

The easiest option is to book a place near to a given place:

```
{
    "offers": [
        {
            "offerId": "<selected_offer_id>",
            "optionalReservationSelections": [
                {
                    "reservationId": "<selected_reservation_id>"
                }
            ],
            "placeSelections": [
                {
                    "reservationId": "<selected_reservation_id>",
                    "referencePlace": {
                        "coachNumber": "string",
                        "placeNumber": "string"
                    }
                }
            ]
        }
        ..
    ],
    "passengerSpecifications": [
        {
            "externalRef": "<passenger_id>",
            "type": "PERSON"
        }
    ],
}
```

Another option is to express seating wishes of a passenger such as at the window etc.

```
{
    "offers": [
        {
            "offerId": "<selected_offer_id>",
            "optionalReservationSelections": [
                {
                    "reservationId": "<selected_reservation_id>"
                }
            ],
            "placeSelections": [
                {
                    "reservationId": "<selected_reservation_id>",
                    "accommodations": [
                        {
                            "passengerRefs": "string",
                            "placeProperties": [
                                "WINDOW"
                            ]
                        }
                    ]
                }
            ]
        }
        ..
    ],
    "passengerSpecifications": [
        {
            "externalRef": "<passenger_id>",
            "type": "PERSON"
        }
    ],
}
```

Or if you have unique seat chosen via a graphical seat reservation (see below) you just pass in a coach and seat number in the `POST /bookings` request:

```
{
    "offers": [
        {
            "offerId": "<selected_offer_id>",
            "optionalReservationSelections": [
                {
                    "reservationId": "<selected_reservation_id>"
                }
            ],
            "placeSelections": [
                {
                    "reservationId": "<selected_reservation_id>",
                    "places": [
                        {
                            "coachNumber": "5",
                            "placeNumber": "21",
                            "passengerRef": "<passenger-id>"
                        }
                    ]
                }
            ]
        }
        ..
    ],
    "passengerSpecifications": [
        {
            "externalRef": "<passenger_id>",
            "type": "PERSON"
        }
    ],
}
```

To be able to build a graphical seat reservation, you first need information of the layout of the vehicles. The service `GET /coachLayouts/{layoutId}` allows you to download the layout. As a distributor you can now use this information to build a slick reservation UI that allows your customer to choose the seat or bed.

How to book a pass?

Offers for a pass for a certain origin and destination can be requested by using the `productTags` filter:

```
{
  "tripSearchCriteria": {
    "origin": "urn:uic:stn:8500010",
    "destination": "urn:uic:stn:8503000"
  },
  "offerSearchCriteria": {
    "productTags": ["Interrail"]
  }
}
```

What are fares?

Fares are priced segments which can be used to create offers for a given trips constrained by fare combination constraints. In the role as a retailer you don’t need to care about them.

Pagination

OSDM uses cursor based pagination and the `_links` concept of **HATEOAS** for pagination. Thus, for responses where pagination is relevant, pagination links will be added, following URL to `next` and `previous` pages are provided if pagination shall be supported.

```
{
  "id": "trip-1",
  "trips": [],
  "_links": [
    {
      "rel": "self",
      "href": "https://api.osdm.com/bookings/123124"
    },
    {
      "rel": "next",
      "href": "https://api.osdm.com/bookings/123124?page=next"
    },
    {
      "rel": "previous",
      "href": "https://api.osdm.com/bookings/123124?page=previous"
    }
    ...
  ]
}
```

The nature of the link is indicated by the `rel` attribute.

Where semantically valid, additional links to the `firstand` `last` pages can be provided, i.e., the`/bookings` resource.

Resources Supporting Pagination

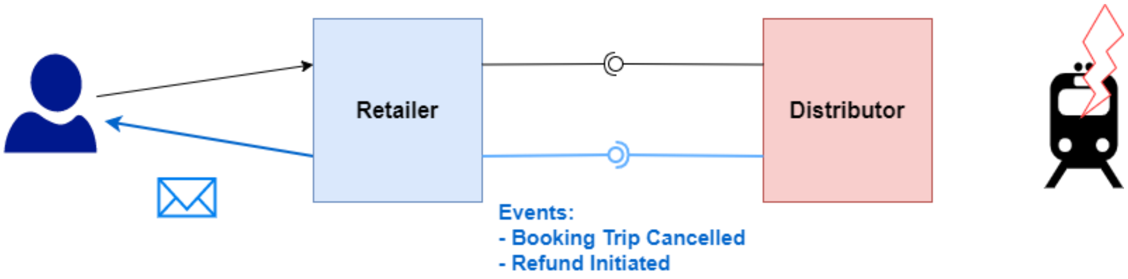
With this version of the specification, the following resources should support pagination:

- **GET** `/trip-collections/`
- **GET** `/exchange-offers/`
- **GET** `/bookings/`
- **GET** `/products/`
- **GET** `/coachLayouts/`
- **GET** `/places/`
- **GET** `/zones/`

Note that while a `page` query parameter is provided for verbs supporting pagination, it is not mandatory to use it: One implementor might use the parameter to scroll with a fixed collectionId, while another could prefer consider scrolling the retrieval of previous and next collections, thus ignoring the parameter and linking to collections with a different id.

Synchronization

For carriers or distributors to inform retailers about changes in a trip or a booking, we have added an API that a carrier or distributor system can inform a retailer about that fact, that a trip or a booking has changed. The retailer can then itself inform a passenger or the purchaser e.g. a travel agency about changes to the trip, the booking or the complaint.



Synchronization

The API informs authorized listener of the fact that changes have occurred using Webhook.

Event Types and Their Semantics

Booking

- `BOOKING_TRIP_CONFIRMED`

  A trip related to the booking which was uncertain has been confirmed. For large groups it is possible to book before the train is actually available.

  The recommended action is to reload the related booking and to bubble the event up. If final partner, inform the client. You may want to re-confirm the client is traveling and if relevant trigger confirmation the payment process.

- `BOOKING_TRIP_CHANGED`

  A trip which was booked (but has not departed yet) has had its schedule updated. Thus, it does not cover real-time updates. The trip on the level of that provider is still feasible.

  The recommended action is to reload the booking from this provider. If part of a multi-provider trip, check the total trip is still feasible and bubble the event up. If final partner, inform the client.

- `BOOKING_REAL_TIME_EVENT_OCCURRED`

  A train involved in a trip of the linked booking is experiencing some unexpected change in the actual schedule.

  The recommended action is to notify the client if it is subscribed to some kind of real time update. If relevant, start some contingency flow.

- `FULFILLMENT_AVAILABLE`

A new fulfillment has been created in the booking. Note that the creation of a fulfillment can be an asynchronous process.

The recommended action is to retrieve the booking and update the local representation.

- `FULFILLMENT_REFUNDED`

  A fulfillment has been refunded. In case of deferred refund, it would be at the end of the « waiting period ». In case of seat release, it will take place at the moment of the actual refund.

  The recommended action is to retrieve the booking and update the local representation. If relevant trigger matching payment transactions.

- `FULFILLMENT_EXCHANGED`

  A fulfillment has been exchanged. In case of deferred refund, it would be at the end of the « waiting period ». In case of seat release, it will take place at the moment of the actual refund.

  The recommended action is to retrieve the booking and update the local representation. If relevant trigger matching payment transactions (only in case of price decrease).

- `FULFILLMENT_CONTROLLED`

  A fulfillment in the booking has been checked in.

  The recommended action is to retrieve the booking and update the local representation.

- `ACCOMMODATION_RELEASED`

  An accommodation in the booking has been released, meaning it is made available again in the inventory.

  The recommended action is to retrieve the booking and inform the customer.

- `REFUND_INITIATED`

  A refund process has been initiated for one or several fulfillments in the booking using `PATCH /bookings/{id}/refunds`.

  The recommended action is to retrieve the booking and update the local representation.

- `EXCHANGE_INITIATED`

  An exchange process has been initiated for one or several fulfillments in the booking using `PATCH /bookings/{id}/exchangeOperations`.

The recommended action is to retrieve the booking and update the local representation.

- `PURCHASER_CHANGED`

  A passenger has been updated by `PATCH /bookings/{id}/purchaser`.

  The recommended action is to retrieve the booking and update the local representation.

- `PASSENGER_CHANGED`

  A passenger has been updated by `PATCH /bookings/{id}/passenger`.

  The recommended action is to retrieve the booking and update the local representation.

Complaints

- `INITIATED`

  A complaint has been submitted using `POST /complaints`.

  The recommended action is to retrieve the complaint and inform the customer.

- `DECIDED`

  A decision has been taken on the complaint (CONFIRMED or REJECTED) using `PATCH /complaints/{complaintId}`.

- `SETTLED`

  A previously CONFIRMED complaint has been settled using `PATCH /complaints/{complaintId}`.

  The recommended action is to retrieve the complaint and inform the customer.

- `INFORMATION_MISSING`

  The complaint has been examined and some information is missing `PATCH /complaints/{complaintId}`.

  The recommended action is to retrieve the complaint and inform the customer.

Subscribing to Events

A receiver has to provide the following information:

- The events it is interested in. E.g. `BOOKING` or `COMPLAINT`
- The end point the events need to be sent to. E.g. `https://trainline.com/api/webhooks/db`
- Sign message to guarantee that the sender is in fact who it claims to be. E.g. `HMAC`
- IP whitelist

Best Practices for Non-Functional Requirements

From a customer point of view, to book a trip he or she expects a seamless and fluid user experience. Fundamental to achieving this goal are fast responses for all requests triggered by the customer.

From a business point of view, the response time of a site directly correlates with the conversion rate. This means if a site takes too long to load, the customer will leave the site with out actually buying a ticket.

In order to achieve a seamless and swift user experience, all parties involved most play their part and provide fast adn predictable response times. Formally, by `95% Response Time` we denote the response time for 95% of all request. By `Max Response Time` we denote the maximum time a response can take before a time out must be expected.

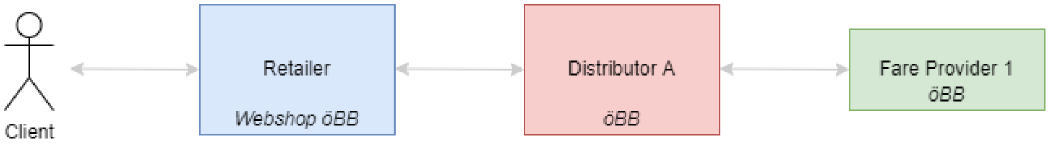
For achieve a good customer experience, we need to define some response time for a each parties and we assume the fare providers and pricing engines can be called in parallel.

Response time	Description
Fare Provider Response Time	Time to calculate/fares
Distributor Response Time	Time to calculate an offer from (remote) fare provider(s)
Retailer Response Time	Time to combine offers from
Pricing Engine Response Time	`Distributor Response Time + max(Fare Provider Response Times 1..m) + Communication Time 1`
Channel Response Time	`Retailer Response Time + max(Pricing Engine Response Time 1..n) + Communication Time 2`

The following illustration highlights the different response times:

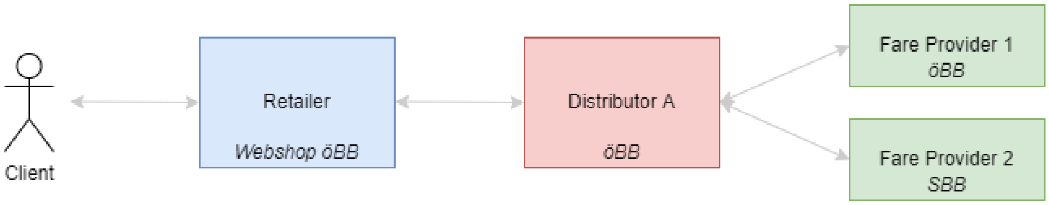
Scenario One Distributor - One Fare Provider

E.g:. öBB sells Garnisch-Partenkirchen - Innsbruck



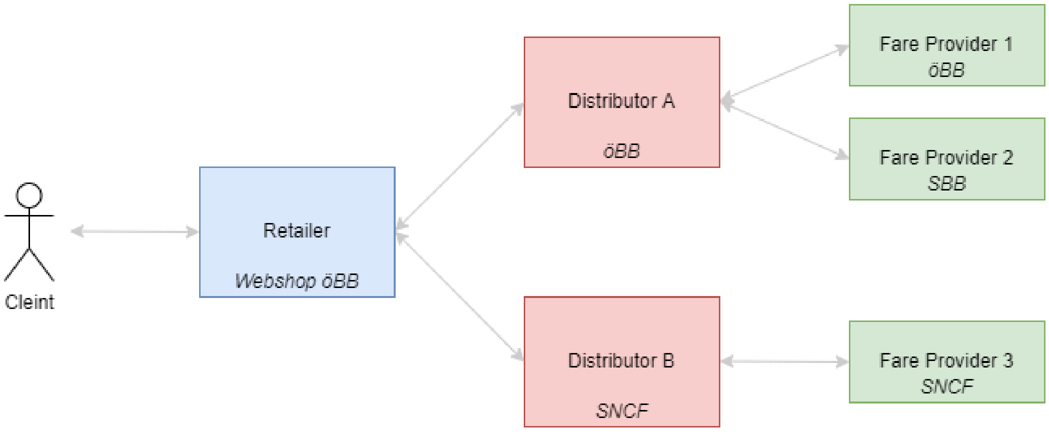
Scenario One Distributor - Multiple Fare Providers

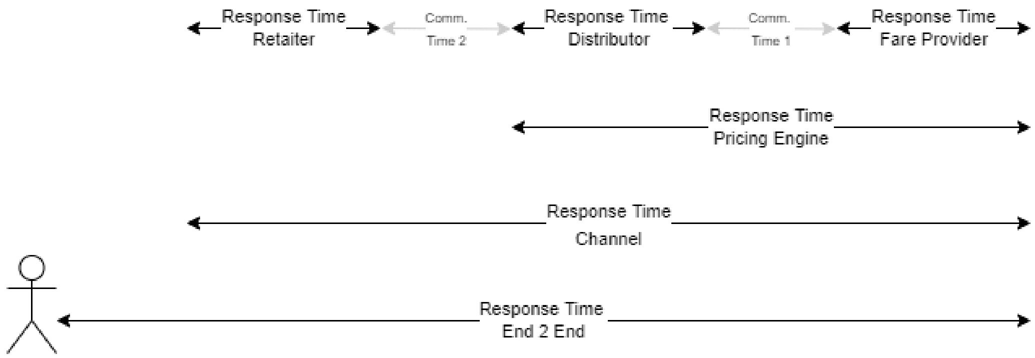
E.g:. öBB sells Garnisch-Partenkirchen - Innsbruck - Basel



Scenario Multiple Distributors - Multiple Fare Providers

E.g:. öBB sells Garnisch-Partenkirchen - Innsbruck - Basel - Colmar





Response Times

To achieve a good user experience with fast response time the following the three response times are crucial.

1. The longest response of any fare provider involved.
2. The longest response of any distributor involved.
3. The time it takes the retailer to render the content.

This observation leads to the following non-functional requirements for a given role.

Non-Functional Requirements for a “Channel”

Mandatory service level requirements to be fulfilled by a channel.

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

Resources	Look to Book Rate	95% Response Times (msec)	Max. Response Time (msec)	Required Error Handling
`/places`	-	60	80	
`/trips`	-	400	600	
`/offers`	10000:1	1000	2000	
`/bookings`	1:1	400	600	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 three days but no

Resources	Look to Book Rate	95% Response Times (msec)	Max. Response Time (msec)	Required Error Handling
				further than the train departure or until an appropriate reply was received indicating that the booking was not made.
`/bookings/{id}/...`	5:1	800	1600	
`/bookings/{id}/passengers`	0.01:1	600	900	retry
`/bookings/{id}/fulfillments`	1:1	600	1200	
`/fulfillments`	1.1:1	600	1000	
`/bookings/{id}/refundOffers`	0,5:1	1000	2000	
`/boo`	0.	10	20	

Resources	Look to Book Rate	95% Response Times (msec)	Max. Response Time (msec)	Required Error Handling
`kings/{id}/exchangeOffers`	01:1	00	00	

Non-Functional Requirements for the role “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 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.

Resources	Look to Book Rate	95% Response Times (msec)	Max. Response Time (msec)	Required Error Handling
`/places`	-	50	75	
`/off`	1	400		

Resources	Look to Book Rate	95% Response Times (msec)	Max. Response Time (msec)	Required Error Handling
`ers`	00:1			
`/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 three days but no further than the train departure or until an appropriate reply was received indicating that the booking was not made.
`/bookings/{id}/...`	5:1	400		
`/bookings/{id}/passengers`	0.01:1	200		retry
`/bookings/{id}/fulfillments`	1:1	200		
`/fulfillments`	1.1:1	400		
`/bookings/{i`	0.5:	400	800	

Resources	Look to Book Rate	95% Response Times (msec)	Max. Response Time (msec)	Required Error Handling
`d}/refundOffers`	1			
`/bookings/{id}/exchangeOffers`	0.01:1	400	800	
`/coachLayouts`	Once per day	8000	10000	
`/coachLayouts/{layoutId}`	2:1	200	300	

The mandatory code lists on enums and extensible enums are provided within the schema files.

Code Lists

URNs for code lists

Code List	Name Space and domain	CodeList	Description	example	base path for relative references
stations	`urn:uic`	`stn`	UIC station codes (= TAP-TSI retail station codes as defined in TAP-TSI Technical Document B.9). The codes are defined in TAP-TSI as numerical codes starting with the TAP-TSI country code without leading zeroes. Temporary codes not jet part of TAP-TSI reference data might be added with an additional number before the country code as aggreed with ERA.	`urn:uic:stn:8512345`	`urn:uic:stn:`
metastations	`urn:uic`	`metastn`	UIC meta station codes to e used for Points of Interrest	`urn:uic:metastn:198`	`urn:uic:metastn:`
service brands, products	`urn:uic`	`sbc`	UIC service brand code (TAP-TSI B.4.7009 / https://uic.org/passenger/passenger-services-group/article/service-brand-code-list)	`urn:uic:sbc:17`	`urn:uic:sbc:`
companies	`urn:uic`	`rics:ac`	company code (TAP-TSI https://www.era.europa.eu/registers/ocr_en / https://uic.org/support-activities/it/rics) and optional administration code (AC)	`urn:uic:rics:1080:000011`	`urn:uic:rics:`
countries	`urn:iso`	`std:iso:3166`	ISO Country Codes	`urn:iso:std:iso:3166:CH`	`urn:iso:std:iso:3166:`
currencies	`urn:iso`	`std:iso:4217`	ISO Currency Codes	`urn:iso:std:iso:4217:CFR`	`urn:iso:std:iso:4217:`

Proprietary values

Be it to represent specific places, or any other type of reference data, some OSDM providers may need to extend a code set with proprietary values in order to support their use cases, while these values are unlikely to be relevant to the rest of the OSDM community. To do so, an OSDM provider should then use the following format for the proprietary values it wishes to extend the code set with :

urn: X_<3 letters code for the provider>::

For example: urn: x_zoo:paxtype:OLIPHANT

Accommodation Type

Code	Description
SEAT	
COUCHETTE	Night trains only
BERTH	Night trains only
VEHICLE	Place for vehicles, e.g. in car carriage trains
STORAGE	Storage place for items, e.g. a hook for bicyles or space for a pram

Accommodation Sub Type, Place Property, Reservation Place Preference

The Accommodation Sub Type defines the characteristics of a place for an offer. It can not be changed within the offer.

(Reservation) Place Preferences define the options that can be selected within the offer when requesting a reservation. Place Properties define the place characteristics for the places in a reservation.

Depending on the commercial decisions some codes might be used in accommodation sub type with one carrier whereas another carrier might alow them in the place preferences (e.g. Bicycle and Tandem Bicycle might have the same price (Place Preference) of different prices (Accommodation Sub Type). There might be place properties that indicate characteristics of a place to the traveller which are not selectable (e.g. Place without Window)). Therefore the shown usage for the different types are an indication but not mandatory.

Place Preference Group collects Place preferences where only one can be chosen of by the customer.

Code	Accommodation Sub Type	Place Preference	Place Property	Description	Place Preference Group
AISLE	N	Y	Y	Place at the aisle	PLACE_LOCATION

Code	Accommodation Sub Type	Place Preference	Place Property	Description	Place Preference Group
AIR-CONDITIONED	N	Y	Y	Place in air conditioned area	
ANY_SEAT	Y	N	Y	Indicates within a place preference group explicitly that the selection is optional	
BISTRO	N	Y	Y	Places in a coach with self-service bistro	
BICYCLE	Y	Y	Y	Bicycle hook	
BUSINESS	Y	Y	Y	Manager compartment/business	
BUSINESS_COMFORT	Y	Y	N	Business Comfort – business seat with reduced service	
CABIN8	N	Y	Y	Special place group in TGV	
CAR_SMALL	Y	N	Y	Vehicle parking place category 1-3	
CAR_LARGE	Y	N	Y	Vehicle parking place category 6-8	
CARRE	N	Y	Y	Carré (4 seats facing normally 2nd Class)	
CHILDREN_AREA	Y	Y	Y	Places in children area	FAMILY
CLUB	Y	N	Y	Club Category (RENFE)	
CLUB_2	N	Y	Y	Club Duo (2 seats facing in a separate compartment)	
CLUB_4	N	Y	Y	Club 4 (4 seats facing)	
COMPARTMENT	Y	N	Y	places in a compartment	PLACE_GROUPING
COMPLETE	Y	N	Y	all places in a compartment are included, no other passengers will be located in the compartment	
CONFERENCE	Y	N	Y	Conference compartment	
CONNECTING_DOOR	N	Y	Y	Compartments with connecting Door (in Sleepers)	
COUCHETTE_2	Y	N	Y	Two person couchette cabin	
COUCHETTE_4	Y	N	Y	Couchette Four-berth	

Code	Accommodation Sub Type	Place Preference	Place Property	Description	Place Preference Group
COUCHETTE_5	Y	N	Y	Couchette Five-berth	
COUCHETTE_6	Y	N	Y	Couchette Six-berth	
COUCHETTE_COMFORT_4	Y	N	Y	Couchette higher quality Four-berth	
COUCHETTE_COMFORT_5	Y	N	Y	Couchette higher quality Five-berth	
COUCHETTE_COMFORT_6	Y	N	Y	Couchette higher quality Six-berth	
COUCHETTE_PRM_2	Y	N	Y	Couchette suitable for PRMs Two-berth	
COUCHETTE_PRM_3	Y	N	Y	Couchette suitable for PRMs Three-berth	
COUCHETTE_PRM_4	Y	N	Y	Couchette suitable for PRMs Four-berth	
DOUBLE	Y	N	Y	Two person sleeper compartment	
DOUBLE_SWC	Y	N	Y	Double sleeper compartment with shower & WC	
DOUBLE_SWC_DB	Y	N	Y	Double sleeper compartment with shower & WC & double bed	
DOUBLE_S	Y	N	Y	Double sleeper compartment with shower	
EASY_ACCESS	Y	Y	Y	Place with easy access for PRMs	
FACE_2_FACE	Y	Y	Y	places face to face (2 seats facing)	PLACE_ALIGNMENT
EXCELLENCE	Y	Y	Y	Special Excellence Places (RhB)	
FAMILY	N	Y	Y	Places in family area	FAMILY
FRONT_VIEW	N	Y	Y	Seat with front-view	

Code	Accommodation Sub Type	Place Preference	Place Property	Description	Place Preference Group
HISTORIC_COACH	Y	Y	Y	Seat in historic coach	
INCLUDING_MEAL	Y	Y	Y	Meal at the place is included	
INCLUDING_DRINK	Y	Y	Y	A drink is included at the place	
KIOSQUE	N	Y	Y	Kiosque (special seats in edge area of a TGV)	
LADIES	N	Y	Y	Ladies compartment	
LOWER_BED	N	Y	Y	Lower bed or couchette	PLACE_LOCATION
LOWER_DECK	N	Y	Y	Lower deck in a double deck train	DECK
MEN	N	Y	Y	Men compartment in night train	
MIDDLE_BED	N	Y	Y	Middle bed or couchette	PLACE_LOCATION
MIDDLE_SEAT	N	Y	Y	Middle seat	PLACE_LOCATION
MINI_SUITE	Y	N	Y	Mini Suite - single person couchette compartment (Capsule)	
MIXED	N	Y	Y	Mixed compartment in night train	
MOTOR_CYCLE	Y	N	Y	Motorcycle	
MOTOR_CYCLE_SC	Y	N	Y	Motorcycle with sidecar	
NEAR_ANIMALS	N	N	Y	Places close to place with animals	
NEAR_DINING	N	Y	Y	Places near the dining car	
NEAR_PLAY_AREA	N	Y	Y	Places near a child play area	FAMILY
NEAR_BICYCLE_AREA	Y	Y	Y	Places near the bicycle storage space	
NEAR_WHEELCHAIR	Y	Y	N	used to indicate places near the wheelchair when booked by an accompagning person	
OPEN_SPACE	Y	Y	Y	Places in open space area	PLACE_GROUPING

Code	Accommodation Sub Type	Place Preference	Place Property	Description	Place Preference Group
PANORAMA	Y	Y	Y	Places in a panorama coach	
PHONE	N	Y	Y	Places in an area with mobile phone amplifier	
POWER	N	Y	Y	Place with power socket	
PRAM	Y	N	Y	Place for a Pram	
PRAM_WITH_SEAT	Y	N	Y	Seat with space for a pram	
RESTAURANT	Y	Y	Y	Restaurant (places in a dining car)	
SILENCE	Y	Y	Y	Quiet Compartment (Seat)	
SINGLE	Y	N	Y	Single sleeper compartment	
SINGLE_SWC	Y	N	Y	Single sleeper compartment with shower & WC	
SINGLE_SWC_DOUBLE	Y	N	Y	Single compartment with shower & WC & double bed	
SIDE_BY_SIDE	N	Y	Y	places side by side (2 seats side by side)	PLACE_ALIGNMENT
SALON	N	Y	Y	Salon (6 seats facing in a separate compartment)	
SLEEPERETTE	Y	N	Y	Sleeperette (reclining seat)	
SLEEPER_DELUXE	Y	N	Y	berth deluxe	
SOLO	Y	Y	Y	Separate place without neighbor seat	PLACE_LOCATION
SOLO_COM	N	Y	Y	Special separate place without neighbor seat (e.g. in TGV)	PLACE_LOCATION
SPECIAL_SLEEPER	Y	N	Y	Special Sleeper Compartment, one Person sleeper compartment smaller than a Single	
TABLE	N	Y	Y	Places at a table	
TANDEM	Y	Y	Y	Tandem Bicycle	
TOURIST_SLEEPER_2	Y	N	Y	T2 sleeper compartment	

Code	Accommodation Sub Type	Place Preference	Place Property	Description	Place Preference Group
TOURIST_SLEEPER_3	Y	N	Y	T3 sleeper compartment	
TOURIST_SLEEPER_4	Y	N	Y	T4 sleeper compartment	
TOURIST_SLEEPER_3_SWC	Y	N	Y	T3 sleeper compartment with shower & WC	
UPPER_BED	N	Y	Y	Upper bed or couchette	PLACE_LOCATION
UPPER_DECK	N	Y	Y	Upper deck in a double deck train	DECK
VIDEO	N	Y	Y	Place with video entertainment	
WHEELCHAIR	Y	Y	Y	Wheel chair place	
WHEELCHAIR_AND_SEAT	Y	Y	Y	Wheelchair place with additional seat	
WHEELCHAIR_NO_SEAT	Y	Y	Y	Wheelchair space without additional seat	
WIFI	N	Y	Y	Places with WiFi access point	
WINDOW	N	Y	Y	place at the window	PLACE_LOCATION
WITH_ANIMALS	Y	Y	Y	Place with animals (animals allowed)	
WITH_SMALL_CHILDREN	N	Y	Y	Places for passengers with small children	FAMILY
WITHOUT_ANIMALS	N	Y	Y	Place in an area where animals are not allowed	

Ancillary category

Code	Description
PAYMENT_VOUCHER	Payment voucher to be used to pay
PRODUCT_ACCESS	Access to retrieve a product (e.g. electronic newspaper,..)
MERCHANDISE_PRODUCT	

Code	Description
LUGGAGE	Allowance for extra luggage
LUGGAGE_TRANSFER	Service for luggage transport
ON_BOARD_SERVICE	Access to service on board
STATION_SERVICE	Access to some station service, Lounge access,…
FOOD_ON_BOARD	
DRINKS_ON_BOARD	
WIFI	Access to WiFi
PARKING	parking of cars

BarcodeType

Code	Description
FCB	Flexible Content Barcode - See IRS 90918-9 / TAP TSI B.12
TLB	Ticket Layout Barcode - See IRS 90918-9 / TAP TSI B.12
SSB	Small Structured Barcode - See IRS 90918-9 / TAP TSI B.12
DOSIPAS	Double Signed Dynamic Barcode (static part (FCB) and dynamic part created at the control time) - See IRS 90918-9 / TAP TSI B.12

CardType

Code	Description
LOYALTY_CARD	Loyalty card
REDUCTION_CARD	Card providing reduction
CHIP_CARD	Chip card to load bookings/references onto it
PASS	A pass providing full or partial reduction within the region where the pass is valid

Company Code

See code TAP-TSI Company code list.

Country	Carrier Code	Short Name	Full Name
AT	1281	ÖBB	ÖBB- Personenverkehr AG
AT	2133	CAT	City Air Terminal Betriebsgesellschaft m.b.H
AT	3039	WLC	Wiener Lokalbahnen Cargo GmbH
AT	3063	RTS	RTS Rail Transport Service GmbH
AT	3153	WIENER	WIENER LINIEN GmbH & Co KG

Country	Carrier Code	Short Name	Full Name
		LINIEN	
AT	3178	RNE	RailNetEurope
AT	3226	LINZ LINIEN GmbH	LINZ LINIEN GmbH für öffentlichen Personenverkehr
AT	3236	WESTbahn	WESTbahn Management GmbH
AT	3268	Holding Graz	Holding Graz - Kommunale Dienstleistungen GmbH
AT	3282	IVB	Innsbrucker Verkehrsbetriebe und Stubaitalbahn GmbH
AT	3306	VOR	Verkehrsverbund Ost-Region (VOR) GmbH
AT	3427	VAL	ÖBB-Postbus GmbH - Vienna Airport Lines
AT	3465	OÖVG	OÖ Verkehrsverbund-Organisations GmbH Nfg. & Co KG
AT	3602	VVV	Verkehrsverbund Vorarlberg GmbH
AT	3608	WLB	WIENER LOKALBAHNEN GmbH
AT	3613	STB	Steiermarkbahn und Bus GmbH
AT	3786	ROeEE PV	Raab Odenburg Ebenfurter Eisenbahn AG Personenverkehr
BE	18	THI Factory	THI Factory SA
BE	1088	SNCB/NMBS	Société nationale des chemins de fer Belges / Nationale Maatschappij der Belgische Spoorwegen
BE	3018	THI	Thalys International
BE	3914	Moonlight Express	Moonlight Express Travel BV
BE	7187	VSB	Voyages SNCF Benelux SPRL
BG	1052	HOLDING BDZ	HOLDING BULGARIAN STATE RAILWAYS
BG	1152	BDZ PASSENGER	BULGARIAN STATE RAILWAYS PASSENGER
CH	1085	SBB AG	Schweizerische Bundesbahnen
CH	1185	SBB-Passengers	Swiss Federal Railways-Passenger subsidiary
CH	1285	SBB-EPA	Schweizerische Bundesbahnen SBB Division Voyageur
CH	1385	SBB2	Schweizerische Bundesbahnen Personenverkehr
CH	3017	CBB	Centralbahn AG
CH	3342	VöV	Verband öffentlicher Verkehr
CH	3917	THU	Regionalbahn Thurbo AG
CH	7387	VSS	Voyages SNCF Suisse SARL
CZ	3189	ARV	ARRIVA vlaky, s.r.o.
CZ	3244	LE	LEO Express Global a.s.

Country	Carrier Code	Short Name	Full Name
CZ	3246	RJ	Regiojet a.s.
CZ	3288	GWTR	GW Train Regio a.s.
CZ	3641	LG	Lokalka Group, spolek
CZ	3725	LET	Leo Express Tenders s.r.o
CZ	3752	RJUK	Regiojet UK a.s.
CZ	3825	Gepard Express	Gepard Express, SE
DE	1080	DB AG	Deutsche Bahn AG
DE	1180	DB Vertrieb GmbH	DB Vertrieb GmbH
DE	3012	BSB	Bodensee-Schiffsbetriebe GmbH
DE	3016	evb	EVB Elbe-Weser GmbH
DE	3030	GVG	Georg Verkehrsorganisation GmbH
DE	3071	EVG	Eifelbahn Verkehrsgesellschaft mbH
DE	3076	Transdev	Transdev GmbH
DE	3118	AG EMS	Aktien-Gesellschaft 'EMS'
DE	3155	RVF	Regio-Verkehrsverbund Freiburg GmbH
DE	3164	agilis	agilis Verkehrsgesellschaft mbH & Co. KG
DE	3165	ODEG	ODEG Ostdeutsche Eisenbahn GmbH
DE	3202	SSB AG	Stuttgarter Straßenbahnen AG
DE	3213	HVV	Hamburger Verkehrsverbund GmbH
DE	3229	RNV GmbH	Rhein-Neckar-Verkehr GmbH
DE	3230	Die Länderbahn	Die Länderbahn GmbH DLB
DE	3243	üstra AG	üstra Hannoversche Verkehrsbetriebe AG
DE	3249	NBE	NBE nordbahn Eisenbahngesellschaft mbH
DE	3254	HKX	Hamburg-Köln-Express GmbH
DE	3258	metronom	metronom Eisenbahngesellschaft mbH
DE	3274	S-Bahn Berlin	S-Bahn Berlin GmbH
DE	3275	HOCHBAHN	Hamburger Hochbahn AG
DE	3279	RMV	Rhein-Main-Verkehrsverbund GmbH
DE	3284	VBK GmbH	Verkehrsbetriebe Karlsruhe GmbH
DE	3295	FlixMobility	FlixMobility GmbH
DE	3312	Ticex GmbH	Ticex GmbH
DE	3314	BVG	Berliner Verkehrsbetriebe (AöR)

Country	Carrier Code	Short Name	Full Name
DE	3316	AVG	AVG Augsburger Verkehrsgesellschaft mbH
DE	3318	MVV	Münchner Verkehrs- und Tarifverbund GmbH
DE	3321	MVG	Münchner Verkehrsgesellschaft mbH
DE	3392	highQ	highQ Computerlösungen GmbH
DE	3393	SVDE	SNCF Voyages Deutschland GmbH
DE	3396	DB Fernverkehr	DB Fernverkehr AG
DE	3426	BVGHT	B.Verkehrsbetriebe HT
DE	3453	MVB	Magdeburger Verkehrsbetriebe GmbH & Co. KG
DE	3469	Naldo	Verkehrsverbund Neckar-Alb-Donau GmbH (Naldo)
DE	3493	WEG	Werra-Eisenbahnverkehrsgesellschaft mbH
DE	3496	VIASP	VIAS Passenger
DE	3510	BTE	BTE BahnTouristikExpress GmbH
DE	3540	INVG	Ingolstädter Verkehrsgesellschaft mbH
DE	3541	SP SE	SPITZKE SE
DE	3546	VVW	Verkehrsverbund Warnow GmbH
DE	3549	ERB	KEOLIS Deutschland GmbH & Co. KG
DE	3591	AKN	AKN Eisenbahn AG
DE	3594	Abellio	Abellio GmbH
DE	3607	AVG	Albtal-Verkehrs-Gesellschaft mbH
DE	3684	ABRM	Abellio Rail Mitteldeutschland GmbH
DE	3719	RFG	Regiobahn Fahrbetriebsgesellschaft mbH
DE	3722	erixx	erixx GmbH
DE	3729	RDC Deutschland	RDC Deutschland GmbH
DE	3817	DVB	Dresdner Verkehrsbetriebe AG
DE	3834	VRB	Verkehrsverbund Region Braunschweig GmbH
DE	3837	SVG	Schienenverkehrsgesellschaft mbH
DE	3841	HANS	Hanseatische Eisenbahn GmbH
DE	3848	Go Ahead BW	Tricon
DE	3849	NXG	National Express Rail GmbH
DE	3864	DB Regio	DB Regio AG
DE	3873	RTB	Rurtalbahn GmbH
DE	3885	BEG	Brohltal-Schmalspureisenbahn Betriebs-GmbH
DE	3925	VIASR	VIAS Rail GmbH

Country	Carrier Code	Short Name	Full Name
DE	7287	VSD	Voyages SNCF Deutschland GmbH
DK	1186	DSB-Passenger	Danske Statsbaner-Passenger subsidiary
DK	1286	DSB-EPA	DSB - Passenger - Reservations issued via EPA
ES	1071	Renfe	Renfe Operadora
ES	1171	RENFE VIAJEROS	RENFE VIAJEROS, S.A
ES	3256	ELIPSOS	ELIPSOS INTERNACIONAL
ES	3780	Rielsfera	RIELSFERA SAU
ES	3860	ILSA	ILSA Intermodalidad de Levante
ES	7587	VSE	Voyages SNCF Espana SRLU
FI	3191	HMVY	Haapamäen museoveturiyhdistys ry
FI	3204	PMR	Porvoon Museorautatie ry
FR	14	CWL	Compagnie des Wagons-Lits
FR	1187	SNCF Voyages	SNCF Voyages
FR	3216	THELLO	THELLO
GB	15	RMF	Rail Manche Finance
GB	17	ENS	European Night Services Ltd
GB	19	Eurostar I	Eurostar International Limited
GB	1170	ATOC Ltd	Association of Train Operating Companies Limited
GB	3008	StL UK	Stena Line UK Ltd
GB	3048	STA Travel	STA Travel International Ltd
GB	3448	STC	Switzerland Travel Center Ltd
GB	7087	VSUK	Voyages SNCF UK Limited
HR	1178	HŽ Putnički prijevoz d.o.o.	HŽ Putnički prijevoz d.o.o. za prijevoz putnika
HU	1143	GYSEV Passenger	Győr-Sopron-Ebenfurti Vasút Zrt. Személyszállítás
HU	1155	MÁV-Start	MÁV-Start Vasúti Személyszállító Zrt.
HU	1355	MAV START Zrt	MAV START Railway Passenger Transport Co.
IT	77	FS EPA	Ferrovie dello Stato Italiane SpA EPA
IT	1183	TI	TRENITALIA
IT	1280	DB B I	DB Bahn Italia Srl
IT	3270	TN	Trenord Srl

Country	Carrier Code	Short Name	Full Name
IT	3280	SAD	SAD Trasporto Locale SpA
IT	3451	FT SpA	Ferrotramviaria SpA
IT	3454	TT	Trentino Trasporti SPA
IT	3455	TFT S.p.A.	Trasporto Ferroviario Toscano S.p.A.
IT	3458	GTT	Gruppo Torinese Trasporti
IT	3460	Fergargano	Ferrovie del Gargano S.r.l.
IT	3462	BI	Busitalia Sita Nord s.r.l.
IT	3463	AIR	ARRIVA Italia Rail S.r.l.
IT	3470	NTV SPA	Nuovo Trasporto Viaggiatori SpA
IT	3827	OBB IT	OBB Italia S.r.l.
IT	7487	VSI	Voyages SNCF Italia SRL
LT	24	LTG	AB 'Lietuvos geležinkeliai'
LT	1124	LTG Link	UAB LTG Link
LU	1182	CFL Voyageurs	Société Nationale des Chemins de Fer Luxembourgeois.
NL	6	StL Holland	Stena Line Holland BV
NL	1084	NS holding	N.V. Nederlandse Spoorwegen
NL	1184	NS Reizigers	NS Reizigers BV
NL	3095	Arriva	Arriva Nederland
NL	3553	Keolis	Keolis Nederland
NL	3564	HTM	HTM Personenvervoer N.V.
NL	3604	GoAbout	GoAbout B.V.
NL	3606	Qbuzz	Qbuzz B.V.
NL	3626	GVB	GVB Exploitatie BV
NL	3888	European Sleeper	European Sleeper Cooperatie U.A.
NL	8999	Connexxion	Connexxion Openbaar Vervoer N.V.
NL	9901	Eurail B.V.	Eurail B.V.
NL	9902	Eurail Group - management	Eurail Group G.I.E. management
NO	3028	Narvik	Visit Narvik
NO	3403	FT	Flytoget AS
NO	3822	Vy Tog	Vy Tog AS
PL	1051	PKP	Polskie Koleje Państwowe Spółka Akcyjna
PL	1151	PR	Przewozy Regionalne Spółka z o.o.

Country	Carrier Code	Short Name	Full Name
PL	1251	PKP Intercity	PKP Intercity S.A.
PL	1351	SKMT	PKP SKM w Trójmieście Spółka z o.o.
PL	3151	KM	‘Koleje Mazowieckie-KM’ Sp. z o.o.
PL	3238	Arriva RP Sp. z o.o.	Arriva RP Spółka z ograniczoną opdowiedzialnością
PL	3252	KD	Koleje Dolnośląskie S.A.
PL	3253	KW	Koleje Wielkopolskie Sp. z o.o.
PL	3397	Eurotrans	Eurotrans Sp.z o.o.
PL	3495	WISKOL	WISKOL Sp. z o.o.k spolka komandytowa
PT	1094	CP	CP - Comboios de Portugal, E.P.E.
PT	3390	FERTAGUS	FERTAGUS, Travessia do Tejo, Transportes, S.A.
RO	1153	CFR Călători	Societatea Naţională de Transport Feroviar de Călători
RO	3322	REGIOTRANS	SC Regiotrans SRL Brasov
RO	3570	REGIO CALATORI	REGIO CALATORI S.R.L.
RO	3942	IRC	SC Interregional Calatori SRL
RS	1172	SV	Srbija Voz a.d.
RU	1120	JSC ‘FPC’	Joint Stock Company ‘FEDERAL PASSENGER COMPANY’
SE	1174	SJ	SJ AB
SE	3027	Länstrafiken	Länstrafiken i Norrbotten AB
SE	3051	Snälltåget	Snälltåget i Sverige
SE	3126	Öresundtåg	Skånetrafiken, Region Skåne
SE	3197	VT	Västtrafik AB
SE	3251	Scandlines AB	Scandlines AB
SE	3294	Botniatåg	Botniatåg AB
SE	3350	ITAB	Inlandsbanan Turism AB
SE	3550	Tagab	Tågåkeriet i Bergslagen AB
SE	3578	SL	AB Storstockholms Lokaltrafik
SI	1079	SŽ, d.o.o.	Slovenske železnice, d.o.o.
SI	1179	SŽ - Potniški promet, d.o.o.	Slovenske železnice - Potniški promet d.o.o.
SI	1279	SŽ - Potniški promet	Slovenske železnice / Potniški promet d.o.o.
SK	1156	ZSSK	Železničná spoločnosť Slovensko, a.s.
SK	3425	VIALTE	VIALTE s.r.o.

Country	Carrier Code	Short Name	Full Name
SK	3524	Arriva SK	Arriva Service s.r.o.
US	3287	Amtrak	National Railroad Passenger Corporation

ControlDataExchangeType

Code	Description
REGISTRY	See IRS 90918-4 / TAP TSI B.14, data are delivered to the UIC registry
PEER2PEER	See IRS 90918-4 / TAP TSI B.14, data are exchange peer 2 peer

ControlSecurityType

Code	Description

SIP Security in Paper -CIT secure background See IRS 90918-9
| |SID Security in Data (e.g. signed barcode) See IRS 90918-9

`SIS Security in System (Online control vs. Ticket registry) See IRS 90918-9/90918-4

Currency

Currency codes used in OSDM are based on the ISO 4217 standard.

FareType

Code	Description
ADMISSION	transport contract
RESERVATION	seat allocation might include a supplement
ANCILLARY	ancillary

Fare CoachType

Legacy reservation code defined in UIC 90918-1 / TAP TSI B.5.51.1 .

Fare CompartementTypeCode

Legacy reservation code defined in UIC 90918-1 / TAP TSI B.5.51.2.

FulfillmentType

Code	Description
ETICKET	Electronic ticket
CIT_PAPER	Value paper
PASS_CHIP	Fulfillment loaded on pass
PASS_REFERENCE	Reference to booking loaded on pass

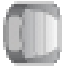
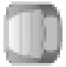
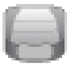
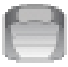
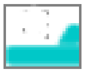
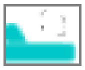
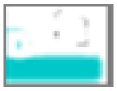
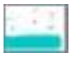
FulfillmentMediaType

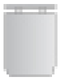
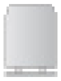
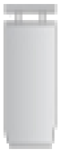
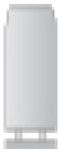
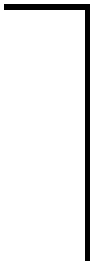
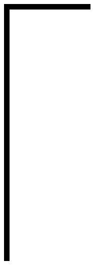
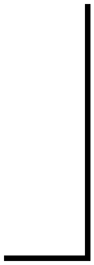
Code	Description
RCT2	RCT2 secure paper format (including compressed format)
RCCST	Credit Card Size format
UIC_PDF	PDF according to UIC Standard
PDF_A4	proprietary A4 pdf format
PKPASS	pkpass file format
RETAILER_APP	mobile ticket in retailer specific format to be loaded into mobile app
TICKETLESS	no ticket
SMS	ticket information of link to the fulfillment send by SMS

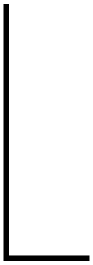
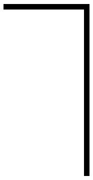
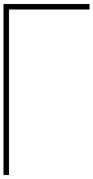
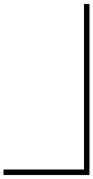
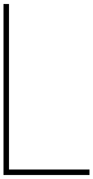
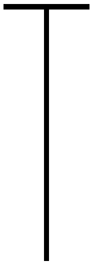
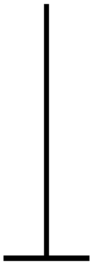
Gender

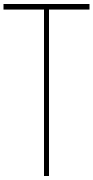
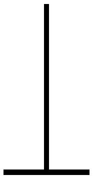
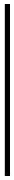
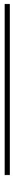
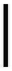
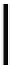
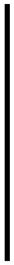
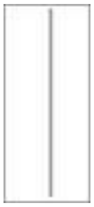
Code	Description
MALE	Male
FEMALE	Female
X	Diverse

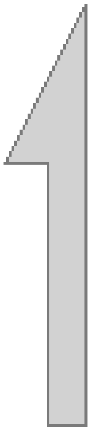
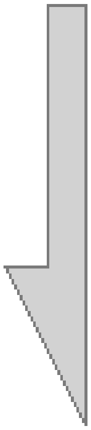
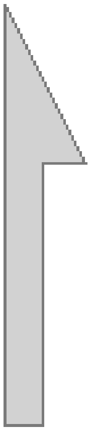
Graphics Items

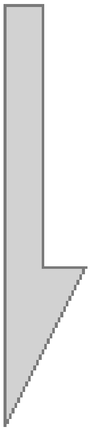
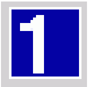
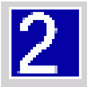
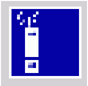
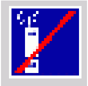
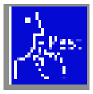
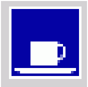
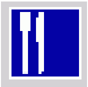
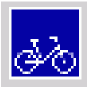
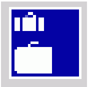
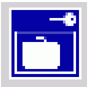
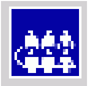
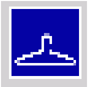
icon	description	icon code	orientation used	mounting used
	seat	1	to right	-
	seat	1	to left	-
	seat	1	up	-
	seat	1	bottom	-
	berth	2	to left	-
	berth	2	to right	-
	couchette	3	to right	-
	couchette	3	to left	-

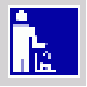
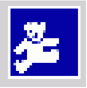
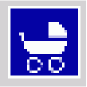
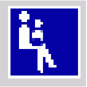
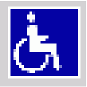
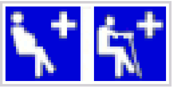
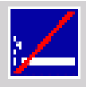
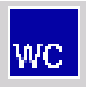
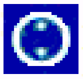
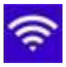
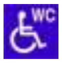
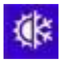
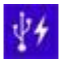
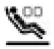
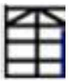
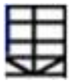
icon	description	icon code	orientation used	mounting used
	small table	20	-	top
	small table	20	-	bottom
	big table	21	-	top
	big table	21	-	bottom
	wall with angle to left, height covers three places	24	-	bottom
	wall with angle to right, height covers three places	25	-	bottom
	wall with angle to left, height covers three places	24	-	top

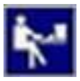
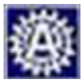
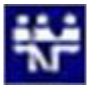
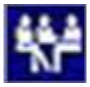
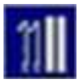
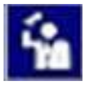
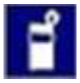
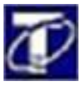
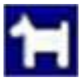
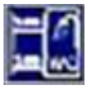
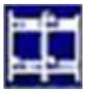
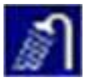
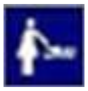
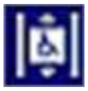
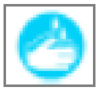
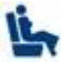
icon	description	icon code	orientation used	mounting used
	wall with angle to right, height covers three places	25	-	top
	small wall with angle to left, height covers two places	27	-	bottom
	small wall with angle to right, height covers two places	28	-	bottom
	small wall with angle to left, height covers 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

icon	description	icon code	orientation used	mounting used
	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	-	

icon	description	icon code	orientation used	mounting used
	arrow indicating upper level	51	-	top-to-bottom
	arrow indicating lower level	52	-	top-to-bottom
	arrow indicating upper level	53	-	top-to-bottom

icon	description	icon code	orientation used	mounting used
	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	-	-

icon	description	icon code	orientation used	mounting used
	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	-	-

icon	description	icon code	orientation used	mounting used
	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

icon	description	icon code	orientation used	mounting used
1	1st class coach	121	-	-
2	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 coach	173	-	-
🛏	sleeping car	174	-	-
🚗	car-carrier	175	-	-

Interface Type

Code	Description
IRS90918_1_RESERVATION_BINARY	Reservation interface according to IRS90918-1 binary message format / TAP TSI B.1 binary
IRS90918_1_RESERVATION_XML	Reservation interface according to IRS90918-1 XML message format / TAP TSI B.5 soap
OSDM	OSDM API

Language

Language codes used in OSDM are based on the ISO 639-1 standard.

Leg Attributes

Service Facilities

Nationality

Language codes used in OSDM are based on the ISO 639-1 standard.

NUTS codes

The common classification of territorial units in the EU (COMMISSION REGULATION (EU) 2016/2066) provides codes for different levels of geographical/political units. E.g.:

Code	Description
BE	BELGIUM
BE2	VLAAMS GEWEST
BE21	Prov. (2) Antwerpen
BE211	Arr. Antwerpen

Overrule Code

Code	Description
CONNECTION_BROKEN	The next leg can not be reached by passenger as previous leg failed.
EQUIPMENT_FAILURE	Vehicle can no longer be operated.
PAYMENT_FAILURE	Payment failure.
PRM_SUPPORT_UNAVAILABLE	A PRM has booked a trip which is not available to him/her.
SALES_STAFF_ERROR	Sale staff error.
STOP_NO_SERVED	Vehicle did not serve the stop place.
STRIKE	Strike.
TECHNICAL_FAILURE	Technical failure on the distribution channel.
TICKET_UNUSED	Ticket not (fully) used by passenger.

Passenger Type (aka. Traveler Type)

Code	Description	Transportable	Bulk (Offline)	Online Request	Online Reply	918-9 FCB Version 3
YOUNG_CHILD	Young child defined by the carrier depending on the age		X		X	freeAddonChild
CHILD	Child defined by the carrier depending on the age		X		X	child
YOUTH	Youth defined by the carrier depending on the age		X		X	youth
ADULT	Adult defined by the carrier depending on the age		X		X	adult
SENIOR	Senior defined by the carrier depending on the age		X		X	senior
FAMILY_CHILD	Child associated with a family traveling together		X	X	X	freeAddonChild
ACCOMP_PRM	Accompanying Person for PRM		X		X	adult
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	child + flag passengerWithReducedMo

Code	Description	Transportable	Bulk (Offline)	Online Request	Online Reply	918-9 FCB Version 3
						bility
WHEELCHAIR	Passenger with wheel-chair			X		wheelchair indication in the reservation data
PERSON	Used in requests together with date of birth and in replies in case of products that might include a change of passenger type along the route			X	X	
PRM	Person with reduced mobility - to be used in case of accompanying person or dog, date of birth must be provided additionally			X		flag passengerWithReducedMobility
DOG	A dog	X	X	X	X	dog
PET	A pet but not a dog	X	X	X	X	
LUGGAGE	Over-size luggage	X	X	X	X	special luggage data
BICYCLE	A bicycle	X	X	X	X	bicycle

Code	Description	Transportable	Bulk (Offline)	Online Request	Online Reply	918-9 FCB Version 3
PRAM	A pram	X	X	X	X	special code in the reservation data
ACCOMP_DOG	An accompanying dog for a PRM	X	X	X	X	dog
CAR	A car for car-carriage trains	X	X	X	X	special car carriage data
MOTOCYCLE	A motorcycle for car carriage trains	X	X	X	X	special car carriage data
TRAILER	A trailer for car carriage trains	X	X	X	X	special car carriage data

Passport

Code	Description
ID_CARD	
PASSPORT	

Code	Description
DIPLOMATIC_PASSPORT	
REFUGEE_TRAVEL_DOCUMENT	
DRIVING_LICENCE	

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 documen
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

Code	Description
BOOKING	The data will be transfered via the booking services
PRE_BOOOKING	The data will be transfered via the pre-booking (offer) services
SIS_CONTROL	The data will be transfered via security in system control data exchange (IRS 90918-4)
SID	The data will be transfered 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 enterd wronlgy by the sales agent

Product Tags

Code	Description
SPLIT_RESERVATION	Indicate for the provider that the distributor is prepared to get reservations including a split inside the reservation Without this indication the provider is not allowed to use the SplitSection Object inside a reservation. The implementation of SPLIT_RESERVATION is not recommended. To provide reservations that do not cover a full leg multiple reservations as offerParts should be created.
Product Types	All product types can be used as Product Tags in the search

Product Types

Code	Description
ADMISSION_MULTI_RIDE	admission that can be used for multiple journeys
ADMISSION_POINT2POINT	admission for a single (point to point) journey
ADMISSION_PASS	admission for a pass to be used unlimitedly within
UPGRADE_PASS	upgrade for a pass to be used unlimitedly within
UPGRADE_POINT2POINT	upgrade for a single (point to point) journey
UPGRADE_MULTI_RIDE	upgrade that can be used for multiple journeys
RESERVATION	reservation
ANCILLARY_SERVICE	ancillary service
ANCILLARY_ITEM	ancillary item (e.g. merchandising item)
REDUCTION_CARD	reduction card

Public Transportation Modes (ptMode)

see Transport Modes

Point of Interest (POI)

POICodeList: By default the code list is set to UIC.

POICode: Values are depending on code list and set is too large to be reproduced

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	Included Cards
UIC_EURAIL	Eurail	Eurail Pass	TRAVEL_PASS	
UIC_INTERRAIL	Eurail	Interrail Pass	TRAVEL_PASS	
UIC_FIP_LEASURE_RED	FIP	FIP reduction (50%)	REDUCTION_CARD	
UIC_FIP_DUTY	FIP	FIP duty	TRAVEL_PASS	
UIC_FIP_LEASURE_FREE	FIP	FIP free personal use	TRAVEL_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	TRAVEL_PASS	
UIC_INTERRAIL_1	Eurail	Interrail Pass First Class	TRAVEL_PASS	

Predefined Card-Ids	Issuer	Description	Type	Included Cards
UIC_FIP_LEASURE_RED_1	FIP	FIP reduction (50%) First Class	REDUCTION_CARD	
UIC_FIP_DUTY_1	FIP	FIP duty First Class	TRAVEL_PASS	
UIC_FIP_LEASURE_FREE_1	FIP	FIP free personal use First Class	TRAVEL_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	
UIC_EURAIL_2	Eurail	Eurail Pass Second Class	TRAVEL_PASS	
UIC_INTERRAIL_2	Eurail	Interrail Pass Second Class	TRAVEL_PASS	
UIC_FIP_LEASURE_RED_2	FIP	FIP reduction (50%) Second Class	REDUCTION_CARD	
UIC_FIP_DUTY_2	FIP	FIP duty Second Class	TRAVEL_PASS	
UIC_FIP_LEASURE_FREE_2	FIP	FIP free personal use Second Class	TRAVEL_PASS	
UIC_RAILPLUS_2	*	A pure rail plus card Second Class	REDUCTION_CARD	
UIC_RIT_1_2	*	RIT reduction for RIT 1 members Second Class	REDUCTION_CARD	
UIC_RIT_2_2	*	RIT reduction for RIT 2 members	REDUCTI	

Predefined Card-Ids	Issuer	Description	Type	Included Cards
		Second Class	ON_CARD	
UIC_RIT_3_2	*	RIT reduction for RIT 3 members Second Class	REDUCTION_CARD	
UIC_EU_DISABILITY_CARD	*	EU Disabled Card		
UIC_INT_DISABILITY_CARD	*	International Disabled Card	REDUCTION_CARD	UIC_EU_DISABILITY_CARD

Other cards should start with the RICS code of the issuer.

Reservation Preference Group

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

Service Brands

Snapshot version, for the actual code list see UIC Service Brand Code List.

Service Brand Code	Abbreviation	Reservation Ticket Text	MERITS Description
33		Ferry	Ship
37		normal train	Train
46	TAJ	TAJ	Day car train
47	TAC	TAC	Car sleeper train, motor rail (CST)
48	SAE	SAE	Unaccompanied car service, motor rail
49	EIC	ExpressIC	Fast and Comfortable Interregional trains
50	EC	EuroCity	EuroCity

Service Brand Code	Abbreviation	Reservation Ticket Text	MERITS Description
51	ICE	ICE	ICE
52	AVE	AVE	AVE
53	EIL	EUROSTAR	Eurostar
54			Talgo
55	OTU	Oresundstog	Oresundstog
56	TGV	TGV	TGV Bruxelles à Lille / Province
58	TRN	Intercités	Intercités
59	AE	ALLEGRO	Allegro
60	ECB	EuroCityBrenner	EuroCityBrenner
62			Suburban service
63	IC	Intercity	Intercity
64			Hotel Train
65		Ferry	hydrofoil
66	IC	Intercity	Inter City Lyn
67			TRN
68			International
69			Express
70	EN	EuroNight	Euro Night
71	HST	High-speed train	High-speed train
72	TRN	TRAIN	Train SNCF
73	TGV	TGV	TGV Sud-Est
74	TGV	TGV	TGV Atlantique
75	TGV	TGV	TGV Nord
76	TGV	TGV	TGV Lyria
77	TGV	TGV	TGV Duplex
79	TGV	TGV	TGV Est
80	TGV	TGV	TGV Interconnexion
82		THALYS	Thalys
83		Ferry	Hovercraft
84	RE	regional train	Regional
85	GPE	Gotthard Panorama Express	Gotthard Panorama Express
87		PENDOLINO	Pendolino
88			Suburban
89	ALV	Alvia	Alvia
90	AVN	Avant	Avant

Service Brand Code	Abbreviation	Reservation Ticket Text	MERITS Description
91	TER	TRAIN	Regional TER
92	REG	Regiontog	Regiontog
93	FB	FRECCIABIANCA	FRECCIABIANCA
94	SC	SuperCity	Supercity
95	CNL	City Night Line (D)	DB Nachtzug
96	INI	InterCityNotte Italia	InterCityNotte
97	GB	ATOC MEMBER OPERATED SERVICE	ATOC MEMBER OPERATED SERVICE
98	ESI	ES Italia	Eurostar Italia
99			Funicular
100			Airport train
101			Night train
102			Touristic train
107			Historical train, steam engine train
108	IRE	IRE	Interregio-Express
109	RB	RB	Regionalbahn
110	RE	RE	Regional-Express
111	RT	RT	RegioTram
112			Shinkansen
113	THT	TrainHotel Talgo	Train hotel talgo
114	EUR	Euromed	Euromed
115	ALR	Alaris	Alaris
116	ALT	Altaria	Altaria
117	ARC	Arco	Arco
119			S-Bahn
121		Night Train	Night Train
122	IR	Interregional	Interregional
123	IRN	Interregional Night Train	Interregional Night Train
124	NLT	TOLSTOI	Tolstoi
126			ARZ
128	AVE	RENFE SNCF EN COOPERATION	Renfe SNCF
129	TGV	TGV INOUI	Renfe SNCF
130	BUS	IC Bus	Bus
131	BUS	IC Bus International	Bus

Service Brand Code	Abbreviation	Reservation Ticket Text	MERITS Description
153		special train	Sonderzug
154			InterCityRapid
155			InterPici
157			Fast train
158			Euregio
159		Bus	IC Ersatzbus
160		Bus	IP Ersatzbus
162		Bus	Replacement Bus
163	TGV	TGV	TGV Duplex Lyria
166	TGV	TGV INOUI	TGV Duplex France Allemagne
170	YHT	YHT	High speed train in Turkey
171	FA	FRECCIARGENTO	FRECCIARGENTO
172	FR	FRECCIAROSSA	FRECCIAROSSA
173	AP	Albula Panorama	Albula Panorama (Panoramic Car)
174	BEX	Bernina Express	Bernina Express (Panorama Train)
175	GEX	Glacier Express	Glacier Express (Panorama Train)
176	GP	Golden Pass	Golden Pass (Panorama Train)
177	BNI	Bernina Panorama	Bernina Panorama (Panoramic Car)
178	zb	zb Zentralbahn AG	Luzern-Interlaken Express (Panorama Train)
179	BXB	Bernina Express Bus	Bernina Express (Panorama Bus)
200	GGB	Gornergrat Bahn	Mountain train
202	ICE	ICE-Allemagne France	ICE Allemagne-France
203		ÖBB-NIGHTLINE	ÖBB Night Line
205	ICP	Intercity Plus	Intercity Plus
206	RID	Riviera Day	Riviera day
207	RIN	Riviera Night	Riviera night
209	RJ	RAIL JET	Rail Jet
213	AZ	DB Autozug	DB Autozug
214		Berlin-Warszawa-Expresas	Berlin-Warszawa-Express
215		Railpromo Austria Express/Treski	Austria Express/Treski
216	PRECIOS	MERCADO	Precios Mercado
219	TGV	TGV	TGV

Service Brand Code	Abbreviation	Reservation Ticket Text	MERITS Description
223	FB	FB	FernBus
224	ICB	Intercitybus	ÖBB-Intercitybus
225	TLK	TLK train	Yours Rail Lines
226	A	RailBus	RailBus
227	BUS	Replacement bus for Regional Train	Replacement bus for Regional Train
228	IR	InterREGIO train	InterREGIO train
229	IRB	Replacement bus for InterRegio train	Replacement bus for InterRegio train
230	MP	Fast International Train	Fast International Train
231	MR	musicREGIO train	musicREGIO train
232	OS	Stopping Train	Stopping Train
233	P	Fast Train	Fast Train
234	R	REGIO train	REGIO train
235	RE	REGIOekspres train	REGIOekspres train
236	VR	viaREGIO train	viaREGIO train
237	TK	TurKol	TurKol
238	EIP	EIC Premium	High-speed train
239	SKM	PKP SKM w Trojmiescie	PKP SKM w Trojmiescie
240	SA	SAPSAN	High speed train
242	STR	STRIZH	Strizh night train
243	STR	STRIZH	Strizh interregional
244	NJ	NJ	NJ Night Jet
245	CAR	AUTOCAR	French regional buses (not sold via Hermes)
246	RJX	RJX	RJX railjet xpress
247	CJX	CJX	CJX cityjet xpress
248		Night train BC	Night train BC
249	TGV	TGV INOUI	TGV INOUI
250	TGV	TGV INOUI	TGV INOUI DUPLEX (double decker TGV)
251	ALI	Aare Linth	Aare Linth (Panorama Train)
252	TGO	Treno Gottardo	Treno Gottardo (Panorama Train)
253	VAE	Voralpen-Express	Voralpen-Express (Panorama Train)
254	LK	FRECCIALINK	FRECCIALINK

Service Class

Code	Description
BEST	
HIGH	Standard first class
STANDARD	
BASIC	Standard second class

Service Facilities / Leg Attributes

OSDM code	Description	TAP-TSI code list	TAP-TSI code
	- classes -		
FIRST_CLASS	First class accommodation	B.4.7037	6
SECOND_CLASS	Second class accommodation	B.4.7037	7
NO_CLASS	Unified class For services with no class differentiation. A service with only one class UIC	B.4.9039	70
PREMIUM	Premium class A class with comfort level higher than first class Business, etc Includes Comfort, Club, Pullman	B.4.9039	54
PREFERENTE	Preferente Spanish first class on long distance trains	B.4.9039	55
TURISTA	Turista Spanish second class on long distance trains	B.4.9039	56
EXECUTIVE	Executive class A class with comfort level higher than first class Trenitalia	B.4.9039	66
BUSINESS	Business class First class Trenitalia	B.4.9039	67
PREMIUM	Premium class A second class with comfort level higher than normal second class Trenitalia	B.4.9039	68
STANDARD	Standard class Trenitalia’s normal second class Trenitalia	B.4.9039	69
	- boarding hints -		
LOADING_ADDITIONAL	Additional loading of vehicles on a train of accompanied motorcars	B.4.7161	3
UNLOADING_ADDITIONAL	Additional unloading of vehicles from a train of accompanied motorcars	B.4.7161	4
LOADING	Loading of vehicles on a train of accompanied motorcars	B.4.7161	7
UNLOADING	Unloading of vehicles from a train of accompanied	B.4.71	9

OSDM code	Description	TAP-TSI code list	TAP-TSI code
	motorcars	61	
BOARDING_POSSIBLE_2H	Boarding is possible 2 hours before departure	B.4.7161	33
ALIGHTING_POSSIBLE_2H	Alighting is possible up to 2 hours after arrival	B.4.7161	34
BOARDING_POSSIBLE_30M	Boarding is possible 30 minutes before departure	B.4.7161	35
ALIGHTING_POSSIBLE_30M	Alighting is possible up to 30 minutes after arrival	B.4.7161	36
OVERNIGHT_STAY_ALLOWED	Overnight stay possible on board	B.4.7161	40
N/A	Time at which the traveller checks in	B.4.7161	44
N/A	Time at which the traveller checks out	B.4.7161	45
	- special passenger facilities -		
BABY_ROOM	Baby care facilities available	B.4.9039	24
NURSERY_SERVICE	Nursery service	B.4.7161	30
CHILDREN_SERVICE	Services for children	B.4.7161	11
FAMILY_COMPARTMENTS	Family compartment(s) available	B.4.9039	48
ARMY_FAMILY_SERVICES	Special services for army families	B.4.7161	32
BUSINESS_SERVICES	Services for business people	B.4.7161	29
DISABLED_SERVICES	Services for disabled persons	B.4.7161	27
DISABLED_FACILITIES	Facilities for heavily disabled persons available	B.4.9039	21
WHEELCHAIR_ACCESS	Access for wheelchairs possible	B.4.9039	28
WHEELCHAIR_TOILET	Dedicated toilet together with dedicated space for wheelchair	B.4.9039	107
POSTAL_SERVI	Postal services available	B.4.71	37

OSDM code	Description	TAP-TSI code list	TAP-TSI code
CES		61	
LUGGAGE_TRANSFER_OFFERED	Luggage transport offered	B.4.7161	41
LUGGAGE_TRANSFER_NOT_OFFERED	Luggage transport is not offered	B.4.7161	42
	- entertainment -		
VIDEO	Video coach available	B.4.9039	33
MOVIES	Movies	B.4.7161	28
MUSIC	Music	B.4.7161	43
WIFI_FREE	Free WiFi service on board available	B.4.7161	46
WIFI_OFFERED	WiFi service on board available	B.4.7161	47
	- place categories -		
SEAT	seats		
N/A	First-class seats	B.4.9039	4
N/A	Second-class seats Also: standard, etc.	B.4.9039	5
N/A	First-class sleepers	B.4.9039	2
N/A	Standard class (second class) sleepers	B.4.9039	3
SLEEPER	sleepers		
SLEEPERETTE	Sleeperette Sleeperette or reclining seat	B.4.9039	8
SLEEPER_SINGLE	sleeper, single		
SLEEPER_SPECIAL	sleeper, special		
SLEEPER_DOUBLE	sleeper, double		
SLEEPER_T2	Sleeper, T2 (two-bed compartment) 0		

OSDM code	Description	TAP-TSI code list	TAP-TSI code
SLEEPER_T3	Sleeper, T3 (three bedcompartment) 0		
SLEEPER_T4	Sleeper, T4 (four-bed compartment) 0		
N/A	First-class sleeper, single	B.4.90 39	11
N/A	First-class sleeper, special	B.4.90 39	12
N/A	First-class sleeper, double	B.4.90 39	13
N/A	Sleeper, T3 Second-class sleeper, T3 (three-bed compartment) 0	B.4.90 39	16
N/A	Sleeper T4 Second-class sleeper, T4 (four-bed compartment) 0	B.4.90 39	17
N/A	Second class sleeper for two	B.4.90 39	60
N/A	Second-class sleeper, T2 (two-bed compartment) 0	B.4.90 39	15
N/A	First-class sleeper, single with shower 0	B.4.90 39	18
N/A	First-class sleeper, double with shower 0	B.4.90 39	19
SLEEPER_SINGLE_SHOWER_WC	sleeper, single with shower and WC		
SLEEPER_DOUBLE_SHOWER_WC	sleeper, double with shower and WC		
SLEEPER_T2_SHOWER_WC	sleeper for two with shower and toilets		
SLEEPER_T3_SHOWER_WC	sleeper, T3, shower, WC		
N/A	First-class sleeper, single with shower and WC	B.4.90 39	57
N/A	First-class sleeper, double with shower and WC	B.4.90 39	58
N/A	Second class sleeper for two with shower and toilets	B.4.90 39	61
N/A	Second class sleeper, T3, shower, WC	B.4.90 39	59
COUCHETTE			

OSDM code	Description	TAP-TSI code list	TAP-TSI code
N/A	First-class couchettes	B.4.9039	6
N/A	Second-class couchettes	B.4.9039	7
COUCHETTE_2	compartment with 2 couchettes		62
COUCHETTE_4	compartment with 4 couchettes		63
COUCHETTE_6	compartment with 6 couchettes		64
N/A	Second class compartment with 2 couchettes	B.4.9039	62
N/A	Second class compartment with 4 couchettes	B.4.9039	63
N/A	Second class compartment with 6 couchettes	B.4.9039	64
COUCHETTE_WHEELCHAIR	Couchette with wheelchair space		
N/A	Second class couchette with wheelchair space	B.4.9039	65
VEHICLE_TRAINSPORT	Facilities for vehicle transport etc. Not bicycles. Cars, boats, trailers, motorbikes		
	- food -		
RESTAURANT	Restaurant facility		
BAR	A bar is available	B.4.9039	47
N/A	Restaurant service in 1st class only	B.4.7161	22
COLD_BUFFET	Cold buffet service	B.4.7161	21
HOT_BUFFET	Hot buffet service	B.4.7161	23
MEAL_INCLUDED	Meal service included		
BUFFET	Buffet	B.4.7161	31
MEAL_AT_SEAT	Meal is provided at the seat	B.4.7161	38
MEAL_SELF_SERVICE	Self service meals	B.4.7161	39

OSDM code	Description	TAP-TSI code list	TAP-TSI code
TROLLEY_FOOD_SERVICE	Trolley service (beverage and food cart)	B.4.7161	25
SNACK	Snack	B.4.7161	26
BREAKFAST	Breakfast	B.4.7161	5
DINNER	Dinner	B.4.7161	6
LUNCH	Lunch	B.4.7161	8
BUFFET_MACHINE	Buffet machine available	B.4.9039	50
MINI_BAR	Mini-bar available	B.4.9039	34
N/A	Meal service included for 1st class passengers	B.4.7161	24
	- other -		
MASK_MEDI_GRADE	Medical-grade masks must be worn	B.4.9039	71
MASK	Mask obligation according to legal regulation Mask obligation according to legal regulation	B.4.9039	72
VACCINATION	Vaccinated, tested, recovered rule applies on trains, valid proof must be presented	B.4.9039	106
N/A	Metro connection Metro, subway, underground connection Only used in TSDUPD	B.4.9039	101
N/A	Taxi connection Taxi connection Only used in TSDUPD	B.4.9039	102
N/A	Bus connection Bus connections available Used only in TSDUPD	B.4.9039	103
N/A	Tram connection Tram connections available Used only in TSDUPD	B.4.9039	104
N/A	2nd Class couchette five beds C5 2nd Class couchette five beds C5 DB Nachtzug	B.4.9039	105
BICYCLE	Facility for bicycle transport available	B.4.9039	26
PANORAMA	Panorama coach available	B.4.9039	36

OSDM code	Description	TAP-TSI code list	TAP-TSI code
TELEPHONE	Telephone service is available	B.4.9039	44
POWER	Service provides power supply sockets	B.4.9039	45
NON_SMOKER	The travel service is non-smoking	B.4.9039	20
PULLMAN	Pullman car seats	B.4.9039	46
WARNING_STRIKE	Warning! Service may be affected by strike action	B.4.7161	48
	- sales hints -		
NO_BOOKING	The product is not bookable	B.4.7037	25
BOOKING_ON_BOARD_ONLY	Booking is possible on board of the vehicle only		
BOOKING_ON_LOCAL_STATION_ONLY	Booking is possible at the local station / stop only		
BOOKING_ON_LOCALLY_ONLY	Booking is possible locally only		
RESERVATION	Reservation possible	B.4.7037	11
SUPPLEMENT	A supplement is charged	B.4.7037	12
RESERVATION_MANDATORY	Reservation is compulsory	B.4.7037	13
RESERVATION_RECOMMENDED	Reservation is recommended	B.4.7037	14
RESERVATION_MANDATORY_PAR	Reservation is compulsory for parties	B.4.7037	15
N/A	Reservation is compulsory in first-class	B.4.7037	16
GROUPS_RESTRICTED	Group booking restricted	B.4.7037	17
NO_GROUPS	Groups of travellers are not allowed	B.4.7037	18
NO_RESERVATI	Reservation is not possible	B.4.70	19

OSDM code	Description	TAP-TSI code list	TAP-TSI code
ON		37	
N/A	Reservation in first class only	B.4.7037	20
RESERVATION_MANDATORY_ORIGIN	Reservation compulsory from origin station	B.4.7037	21
RESERVATION_RESTRICTED	Reservation allowed for some classes or products	B.4.7037	22
N/A	Reservation in second class only	B.4.7037	23
NOT_AVAILABLE	Not available	B.4.7037	24
BOOKABLE_INT	Bookable through an international reservation system	B.4.7037	26
BOOKABLE_NAT	Bookable through a national reservation system	B.4.7037	27
BOOKABLE_LOCAL	Bookable manually at Rus selling point	B.4.7037	28

Stations

codeList = UIC, ERA

Additional codelists can be defined by implementers. The codelist name should then be prefixed by “X_<3 letters code for the provider>”. Example: “X_PAO_STATIONS”

StationCode: Values are depending on codelist and set is too large to be reproduced.

Supported Online Services

Code	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

Code	Description
EXCHANGE	Exchange with the same number of passengers
FULFILLMENT	Provides 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 depature). The time zone of the departure station applies.
AFTER_SALE	Time value calculated relative to the sates 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	

Transfer Type

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

Transport Mode

Corresponding to numerical codes in TAP-TSI (B.2.3) / UIC (https://uic.org/passenger/passenger-services-group/article/service-brand-code-list)

Code	Description	TAP-TSI Code B.2.3
HIGH_SPEED_TRAIN		8
HISTORIC_TRAIN		16
INTERCITY		9
REGIONAL		11
INTERREGIONAL		10
TRAIN		37
URBAN		12
TRAM		35
UNDERGROUND		36
NIGHT_TRAIN		13
SHARED_TAXI		34
MOTOR_RAIL	Car carriage trains	13
MOUNTAIN_TRAIN		15
PLANE		3
COACH_GROUP	Group of coaches included in multiple trains (through coaches)	31
SHIP		33
BUS		32

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_SAME_CARRIER	Exchange with a new fare of the same carrier
UPGRADE	Exchange with a fare of higher value
GROUP_DOWN_SIZE	Exchange with a group of less passengers

Travel Account Types

Code	Description
TRAVEL_PASS	A travel account or card that also serves as ticket in some area
REDUCTION_CARD	A customer card providing reductions
MULTI_RIDE	A travel account that provides a limited number if trips or item
LOYALTY_CARD	A customer/traveler card that only serves to collect loyality incentives

TravelDirection

Trave direction of a coach. To get the direction of travel for a place the direcion information of the coah needs to be combined with the direction of the seat in the coach layout.

Code	Description
UNSPECIFIED	The direction of travel is undefined
IN_DIRECTION	In direction of travel
OPPOSITE_DIRECTION	Opposite to direction
CHANGING	direction is changing
STARTING_IN_DIRECTION	starting in direction but changing later
STARTING_OPPOSITE_TO_DIRECTION	starting oposite to direction but changing later

TravelValidityType

Type of the travel validity of a fare.

Code	Description
SINGLE_TRIP	Fare is for a single trip.
MULTIPLE_TRIPS	Fare is for multiple trips but with restrictions on the usage.
UNRESTRICTED	The fare is for multiple unrestricted use within the provided travel validity period.

TripAllocationProcess

Trip allocation process indicated the type of process a passenger needs to follow to make use of a multi journey ticket with restrictions.

Code	Description

Code	Description
`NONE`	No action is required, the usage might be tracked automatoally or via on-board controls.
`MANUAL`	The passenger has to allocate a trip manually e.g. by stamping the ticket. (outdated).
`ACTIVATION`	The fulfillment needs to be activated for the trip.
`FULFILLMENT`	A separate fulfillment needs to be retrieved for the trip.

TripAllocationUnit

Trip allocation unit indicates the unit that can be allocatoed on a multi journey ticket with restrictions.

Code	Description
`NONE`	Individual trips don't need to be allocated.
`MANUAL`	allocation per individual trip.
`ACTIVATION`	The allocation is per travel day for multiple trips on the day.
`FULFILLMENT`	The allocation is per duration.

TripInterruptionProcess

Trip interruption process indicates the type of process to be followed by the passenger to interrupt a trip.

Code	Description
`NONE`	No action is required from the passenger in case he is interrupting his trip.
`MANUAL`	In case a passenger is interrupting his trip he needs to receive a confirmation from train or station staff
`ACTIVATION`	The electronic ticket needs to be deactivated during the interruption.

To be released with version 3.1.

TODO: add a list of standardized warnings and needs attributes `Warning.code` and `Warning.details` added

Open Question: Do we need a namespace for the errors and warnings?

Errors and Warnings

General HTTP error codes and generic situations

The following standard HTTP error codes are used in the specification:

Error Code	Description
400	Bad request
401	Unauthorized
403	Forbidden
404	Not found
409	Conflict
500	Internal server error
501	Not implemented
503	Service unavailable

Functional Errors

The OSDM API makes use of the JSON Problem structure to return information about functional errors in the handling of a request. The problem structure is defined in RfC-7807 which defines a way to carry machine-readable details of errors in a HTTP response to avoid the need to define new error response formats for HTTP APIs.

For OSDM, the `code` property should begin with the code of the error type. This code should be the unique identifier for the functional situation in the absolute URI in the `type` property as well.

The `title` is a short, human-readable summary of the problem type which **should not** change from occurrence to occurrence of the problem. The `detail` is a human-readable explanation specific to this occurrence of the problem. The `status` is the HTTP status code generated by the origin server for this occurrence of the problem.

```
"problem": {
  "code": "PLACE_NO_RESULTS",
  "type": "https://osdm.io/errors/place-no-results",
  "title": "The search did not return any result",
  "detail": "The place `Duckburg` could not be found",
  "status": 404,
}
```

The list of all error codes that can be returned by the API, needs to exposed on an URI:

`https://<host>/errors/<error-code>`

where host denotes the host of the API.

Standardized Functional Errors

In order that OSDM implementations behave consistently in error situations, the following error codes must be supported in case of functional errors by all implementations:

Functional area	Error Code	Title
Places	PLACE_INVALID_CHARACTERS	Invalid characters in the search string
Places	PLACE_NO_RESULTS	The search did not return any result
Trips	TRIP_INVALID_CHARACTERS	A search criteria value contains invalid value or invalid characters
Trips	TRIP_SEARCH_CRITERIA_OUTSIDE_BOUNDARY	A search criteria lies outside accepted boundaries
Trips	TRIP_PLACE_UNKNOWN	A provided place is not known
Trips	TRIP_NO_SEARCH_RESULT	The search did not return any result
Offers	OFFER_TRIP_NOT_FOUND	The referenced trip cannot be found (expired ?)
Offers	OFFER_INVALID_CHARACTERS	A search criteria value contains invalid value or invalid characters
Offers	OFFER_SEARCH_CRITERIA_OUT_OF_BOUNDS	A search criteria lies outside accepted boundaries
Offers	OFFER_PLACE_UNKNOWN	A provided place is not known
Offers	OFFER_NO_RESULTS	The trip search did not return any result
Offers	OFFER_SCHEDULE_MISMATCH	Schedule mismatch between systems
Offers	BOOKING_RESERVATION_OPTION_NOT_AVAILABLE	The requested reservation option is not available on this vehicle
Offers	BOOKING_PASSENGER_PROPERTY_NOT_MODIFIABLE	Attempted to modify a read-only property (passenger)
Offers	BOOKING_OFFERPART_PROPERTY_NOT_MODIFIABLE	Attempted to modify a read-only property (reservation, ancillary or fare)
Booking	BOOKING_OFFER_NOT_FOUND	Referenced Offer or offer part not found (offer expired ?)
Booking	BOOKING_INCOMPATIBLE_OFFER_PART	Incompatible offer part with the offer
Booking	BOOKING_INFORMATION_MISSING	Missing information
Booking	BOOKING_INSUFFICIENT_AVAILABILITY	Insufficient availability for one of the requested products
Booking	BOOKING_PLACE_NOT_AVAILABLE	The requested place is not available
Booking	BOOKING_MODIFY_READ_ONLY_PROPERTY	Attempted to modify a read-only property

Functional area	Error Code	Title
Booking	BOOKING_BOOKING_ALREADY_CONFIRMED	The booking is already confirmed
Booking	BOOKING_BOOKING_ALREADY_CANCELLED	The booking is already cancelled
Booking	BOOKING_MODIFICATION_NOT_ALLOWED	The booking and does not allow modifications
Booking	BOOKING_VEHICLE_TOO_HEAVY	A vehicle is too heavy to be transported by car carriage. This relates to the current load of the train or coach, so booking might be possible at another
Booking	BOOKING_VEHICLE_WEIGHT_MISSING	A vehicle is not possible without providing the weight of the vehicle
Confirm	CONFIRMATION_PARTIAL_SUCCESS	Partial success
Confirm	CONFIRMATION_OPERATION_NOT_SUPPORTED	Operation not supported on one of the offer parts
Confirm	CONFIRMATION_UNKNOWN_ERROR	Unknown error on provider side
Confirm	CONFIRMATION_INFORMATION_MISSING	Missing information in the booking
Confirm	CONFIRMATION_FULFILLMENT_TYPE_NOT_SELECTED	Fulfillment type not selected
Confirm	CONFIRMATION_BOOKING_ALREADY_CONFIRMED	Booking already confirmed
Confirm	CONFIRMATION_BOOKING_ALREADY_FULFILLED	Booking already fulfilled
Confirm	CONFIRMATION_BOOKING_ALREADY_CANCELLED	Booking already cancelled

The list can be extended by an implementor but at least these errors must be captured and they must be presented with the codes listed here above.

In case they wish to pass additional warnings specific to their situation and not covered by any of the case below, they can do so by replacing the OSDM prefix with an `X_` followed by an unique identifier for the provider (ex: `X_NVS_NOMEAL`).

Functional Warnings

In addition to errors, a generic warnings mechanism is used to pass non-blocking information or events, such as a price difference with the initially offered price at booking time to the caller.

A warning message is typically made of a code and a detail. The details are not translated and the expected language there should be in English. It is up to the implementers to foresee a translation based on the code if relevant. Optionally, a warning message can contain reference to resources in the model.

```
"warning": {
  "code": "PLACE_PROPERTY_NOT_RESPECTED",
  "type": "https://osdm.io/warnings/place-property-not-respected",
  "title": "Place property not respected",
  "detail": "Place property `WINDOW` is not respected in offer with id
`123456`",
}
```

Standardized Functional Warnings

In order that OSDM implementations behave consistently in warning situations, the following warning codes must be supported in case of functional warnings by all implementations:

Functional area	Warning Code	Title
Reservation	`PLACE_PROPERTY_NOT_RESPECTED`	Place property not respected
todo	`todo`	todo

As for the warnings, an implementor is required to use these warning codes and descriptions for the situations identified in order to be compliant.

In case they wish to pass additional warnings specific to their situation and not covered by any of the case below, they can do so by replacing the OSDM prefix with an `X_` followed by an unique identifier for the provider (ex: `X_NVS_NOMEAL`).

Dealing with Unsupported Parameters in Requests

Some OSDM requests potentially support a large number of parameters (e.g. filters). Supporting all of them is not always possible, e.g. due restrictions of the underlying systems. Thus the following rule applies:

- If the parameter is required, return an *error* “NOT SUPPORTED”.
- If the parameter is optional, return a *response* including a *warning* indicating that the optional request parameter was ignored.

Pagination

OSDM uses cursor based pagination and the `_links` concept of **HATEOAS** for pagination. Thus, for responses where pagination is relevant, pagination links will be added, following URL to `next` and `previous` pages are provided if pagination shall be supported.

```
{
  "id": "trip-1",
  "trips": [],
  "_links": [
    {
      "rel": "self",
      "href": "https://api.osdm.com/bookings/123124"
    },
    {
      "rel": "next",
      "href": "https://api.osdm.com/bookings/123124?page=next"
    },
    {
      "rel": "previous",
      "href": "https://api.osdm.com/bookings/123124?page=previous"
    }
    ...
  ]
}
```

The nature of the link is indicated by the `rel` attribute.

Where semantically valid, additional links to the `first`and `last` pages can be provided, i.e., the`/bookings` resource.

Resources Supporting Pagination

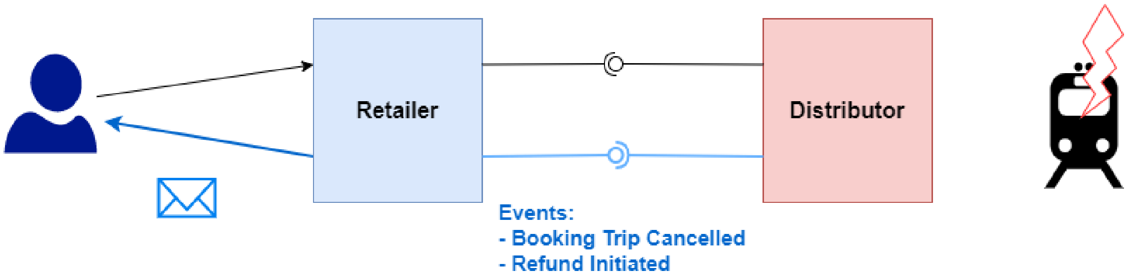
With this version of the specification, the following resources should support pagination:

- **GET** `/trip-collections/`
- **GET** `/exchange-offers/`
- **GET** `/bookings/`
- **GET** `/products/`
- **GET** `/coachLayouts/`
- **GET** `/places/`
- **GET** `/zones/`

Note that while a page query parameter is provided for verbs supporting pagination, it is not mandatory to use it: One implementor might use the parameter to scroll with a fixed collectionId, while another could prefer consider scrolling the retrieval of previous and next collections, thus ignoring the parameter and linking to collections with a different id.

Synchronization

For carriers or distributors to inform retailers about changes in a trip or a booking, we have added an API that a carrier or distributor system can inform a retailer about that fact, that a trip or a booking has changed. The retailer can then itself inform a passenger or the purchaser e.g. a travel agency about changes to the trip, the booking or the complaint.



Synchronization

The API informs authorized listener of the fact that changes have occurred using Webhook.

Event Types and Their Semantics

Booking

- `BOOKING_TRIP_CONFIRMED`

  A trip related to the booking which was uncertain has been confirmed. For large groups it is possible to book before the train is actually available.

  The recommended action is to reload the related booking and to bubble the event up. If final partner, inform the client. You may want to re-confirm the client is traveling and if relevant trigger confirmation the payment process.

- `BOOKING_TRIP_CHANGED`

  A trip which was booked (but has not departed yet) has had its schedule updated. Thus, it does not cover real-time updates. The trip on the level of that provider is still feasible.

  The recommended action is to reload the booking from this provider. If part of a multi-provider trip, check the total trip is still feasible and bubble the event up. If final partner, inform the client.

- `BOOKING_REAL_TIME_EVENT_OCCURRED`

  A train involved in a trip of the linked booking is experiencing some unexpected change in the actual schedule.

  The recommended action is to notify the client if it is subscribed to some kind of real time update. If relevant, start some contingency flow.

- `FULFILLMENT_AVAILABLE`

  A new fulfillment has been created in the booking. Note that the creation of a fulfillment can be an asynchronous process.

  The recommended action is to retrieve the booking and update the local representation.

- `FULFILLMENT_REFUNDED`

A fulfillment has been refunded. In case of deferred refund, it would be at the end of the « waiting period ». In case of seat release, it will take place at the moment of the actual refund.

The recommended action is to retrieve the booking and update the local representation. If relevant trigger matching payment transactions.

- `FULFILLMENT_EXCHANGED`

  A fulfillment has been exchanged. In case of deferred refund, it would be at the end of the « waiting period ». In case of seat release, it will take place at the moment of the actual refund.

  The recommended action is to retrieve the booking and update the local representation. If relevant trigger matching payment transactions (only in case of price decrease).

- `FULFILLMENT_CONTROLLED`

  A fulfillment in the booking has been checked in.

  The recommended action is to retrieve the booking and update the local representation.

- `ACCOMMODATION_RELEASED`

  An accommodation in the booking has been released, meaning it is made available again in the inventory.

  The recommended action is to retrieve the booking and inform the customer.

- `REFUND_INITIATED`

  A refund process has been initiated for one or several fulfillments in the booking using `PATCH /bookings/{id}/refunds`.

  The recommended action is to retrieve the booking and update the local representation.

- `EXCHANGE_INITIATED`

  An exchange process has been initiated for one or several fulfillments in the booking using `PATCH /bookings/{id}/exchangeOperations`.

  The recommended action is to retrieve the booking and update the local representation.

- `PURCHASER_CHANGED`

  A passenger has been updated by `PATCH /bookings/{id}/purchaser`.

The recommended action is to retrieve the booking and update the local representation.

- `PASSENGER_CHANGED`

  A passenger has been updated by `PATCH /bookings/{id}/passenger`.

  The recommended action is to retrieve the booking and update the local representation.

Complaints

- `INITIATED`

  A complaint has been submitted using `POST /complaints`.

  The recommended action is to retrieve the complaint and inform the customer.

- `DECIDED`

  A decision has been taken on the complaint (CONFIRMED or REJECTED) using `PATCH /complaints/{complaintId}`.

- `SETTLED`

  A previously CONFIRMED complaint has been settled using `PATCH /complaints/{complaintId}`.

  The recommended action is to retrieve the complaint and inform the customer.

- `INFORMATION_MISSING`

  The complaint has been examined and some information is missing `PATCH /complaints/{complaintId}`.

  The recommended action is to retrieve the complaint and inform the customer.

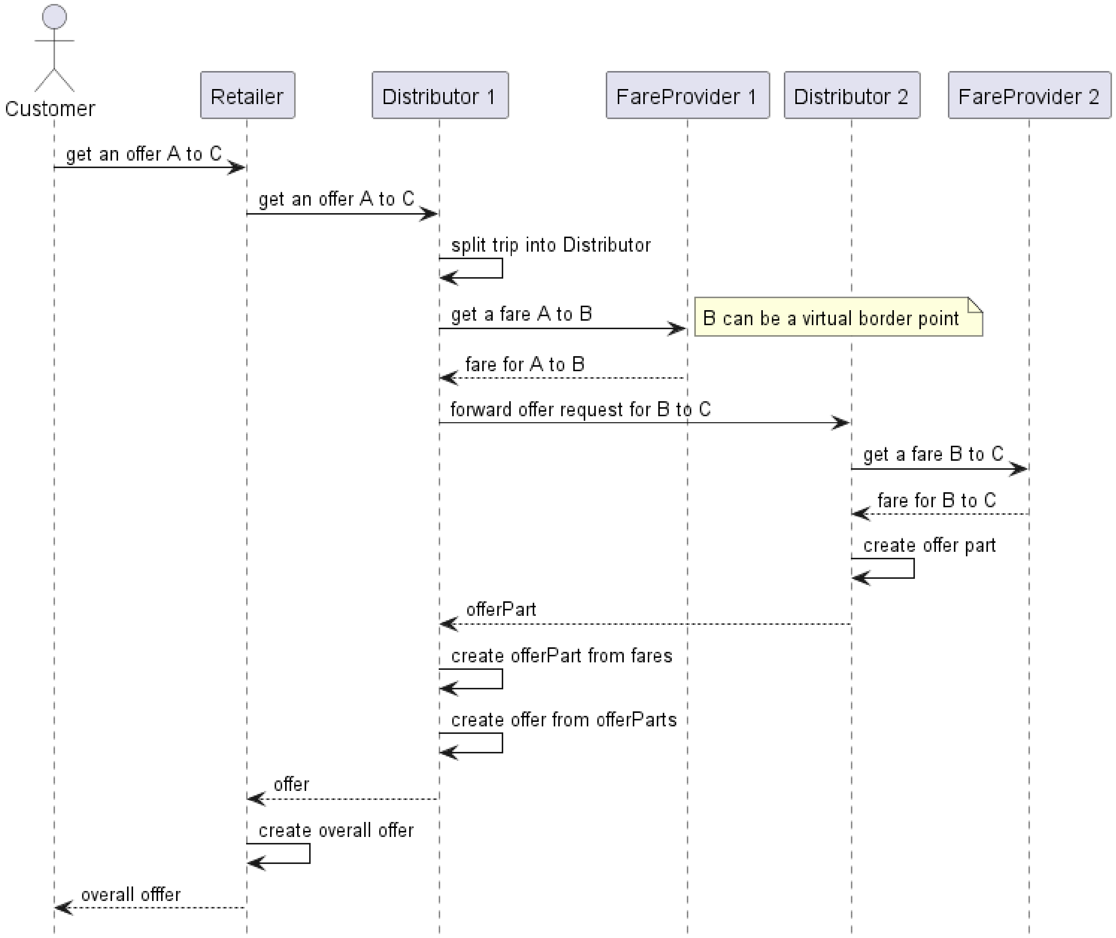
Subscribing to Events

A receiver has to provide the following information:

- The events it is interested in. E.g. `BOOKING` or `COMPLAINT`
- The end point the events need to be sent to. E.g. `https://trainline.com/api/webhooks/db`
- Sign message to guarantee that the sender is in fact who it claims to be. E.g. `HMAC`
- IP whitelist

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 sales process.



Compliance

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 a Distributor

The following services/features are mandatory/optional to implement:

Resources	Description	Need to Support
`/offers`	Resources to get bookable offers	**Mandatory** is to provide at least admission offers.
`/bookings`	Resources to manipulate bookings	**Mandatory**
`/bookings/{bookingId}/passengers`	Resources to manipulate passenger information at every stage of the flow	**Mandatory**
`/bookings/{bookingId}/purchaser`	Resources to manipulate a purchasers information at every stage of the flow	**Mandatory**
`/bookings/{bookingId}/fulfillments`	Resources to retrieve fulfillments, e.g. tickets	**Mandatory** is to support A4 PDF tickets.
`/bookings/{bookingId}/refund-offers`	Resources to get and accept a refund offer	**Mandatory** is to support full refund.
`/bookings-search/`	Resources to search for bookings	**Mandatory**
`/fulfillments`	Resources to retrieve fulfillments, e.g. tickets	**Mandatory** is to support A4 PDF tickets.

Trip planning Resources	Description	Need to Support
`/places`	Resources to search for a place	**Optional**
`/trips`	Resources to search for trips	**Optional**

Booking Modifications Resources	Description	Need to Support
`/bookings/{bookingId}/booked-offers/{bookedOfferId}/ancillaries`	Resources to manipulate parts of a booking consisting of ancillaries	*Conditional*
`/bookings/{bookingId}/booked-offers/{bookedOfferId}/reservations`	Resources to manipulate parts of a booking consisting of reservations	*Conditional*; mandatory to be supported in case seat assignment is provided.

Reservation Resources	Description	Need to Support
`/availabilities`	Resources to get information on available seats for a booking	*Conditional*; mandatory to be supported in case seat assignment is provided.

Advanced After Sales Resources	Description	Need to Support
`/bookings/{bookingId}/documents`	Resources to add documents to a booking	*Optional*
`/bookings/{bookingId}/exchange-operations`	Resources to manage exchange operations	*Optional*
`/bookings/{bookingId}/release-offers`	Resources to get and accept a release offer to return a ticket as a preliminary step for a refund	*Optional*
`/bookings/{bookingId}/reimbursements`	Resources to manage reimbursements of unused tickets	*Optional*
`/bookings/{bookingId}/onHoldOffers`	Resources to manage on hold offers (extensions of the booking time limit)	*Optional*
`/bookings/{bookingId}/cancelFulFillmentOffers`	Resources to cancel a fulfillment	*Optional*

Complaint Resources	Description	Need to Support
`/complaints`	Resources to manage complaints	*Optional*

Master Data Resources	Description	Need to Support
`/coach-layouts`	Resources to get layouts of coaches	*Optional*
`/places`	Resources to get all places	*Optional*
`/products`	Resources to retrieve products information on one or more products	*Optional*
`/reduction-cards`	Resources to retrieve reduction	*Optional*
`/zones`	Resources to retrieve zone information	*Optional*

All non-functional requirements defined in the services must be fulfilled.

Especially all implementations most 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 Carrier aka. Fare Provider

The following services/features are mandatory/optional to implement:

Resources	Description	Need to Support
`/offers`	Resources to get bookable offers	**Mandatory** is to provide fare offers
`/bookings`	Resources to manipulate bookings	**Mandatory**
`/bookings/{bookingId}/passengers`	Resources to manipulate passenger information at every stage of the flow	**Mandatory**
`/bookings/{bookingId}/purchaser`	Resources to manipulate a purchasers information at every stage of the flow	**Mandatory**
`/bookings/{bookingId}/fulfillments`	Resources to retrieve fulfillments, e.g. tickets	**Mandatory** is to support A4 PDF tickets.
`/bookings/{bookingId}/refundOffers`	Resources to get and accept a refund offer	**Mandatory** is to support full refund.
`/booking-search/`	Resources to search for bookings	**Mandatory**
`/fulfillments`	Resources to retrieve fulfillments, e.g. tickets	**Mandatory** is to support A4 PDF tickets.

Trip planning Resources	Description	Need to Support
`/places`	Resources to search for a place	**Optional**
`/trips`	Resources to search for trips	**Optional**

Booking Modifications Resources	Description	Need to Support
`/bookings/{bookingId}/booked-offers/{bookedOfferId}/ancillaries`	Resources to manipulate parts of a booking consisting of ancillaries	*Conditional*
`/bookings/{bookingId}/booked-offers/{bookedOfferId}/reservations`	Resources to manipulate parts of a booking consisting of reservations	*Conditional*; mandatory to be supported in case seat assignment is provided.

Reservation Resources	Description	Need to Support
`/availabilities`	Resources to get information on available seats for a booking	*Conditional*; mandatory to be supported in case seat assignment is provided.

Advanced After Sales Resources	Description	Need to Support
`/bookings/{bookingId}/documents`	Resources to add documents to a booking	*Optional*
`/bookings/{bookingId}/exchange-operations`	Resources to manage exchange operations	*Optional*
`/bookings/{bookingId}/release-offers`	Resources to get and accept a release offer to return a ticket as a preliminary step for a refund	*Optional*
`/bookings/{bookingId}/reimbursements`	Resources to manage reimbursements of unused tickets	*Optional*
`/bookings/{bookingId}/on-hold-offers`	Resources to manage on hold offers (extensions of the booking time limit)	*Optional*
`/bookings/{bookingId}/cancel-fulfillment-offers`	Resources to cancel a fulfillment	*Optional*

Complaint Resources	Description	Need to Support
`/complaints`	Resources to manage complaints	*Optional*

Master Data Resources	Description	Need to Support
`/coach-layouts`	Resources to get layouts of coaches	*Optional*
`/places`	Resources to get all places	*Optional*
`/products`	Resources to retrieve products information on one or more products	*Optional*
`/reduction-cards`	Resources to retrieve reduction	*Optional*
`/zones`	Resources to retrieve zone information	*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.