INFORMATION TECHNOLOGIES FOR SHIFT TO RAIL

D7.10 Development Readiness Review Pack

Conceptual Interface Specifications (CIS) iteration 2

Due date of deliverable: 30/11/2017

Actual submission date: 14/11/2018

Leader of this Deliverable: Thales

Reviewed: Y

Document status		
Revision	Date	Description
1	07/05/2018	First draft
2	29/06/2018	Second version
3	01/07/2017	New version based on iteration 1
4	25/10/2017	Update data models and interfaces with FREL content
5	23/05/2018	WP1 interfaces added THP revision
6	21/06/2018	Updated version
7	18/07/2018	Updated version
8	23/07/2018	Final Version after TMC approval and Quality check
9	14/11/2018	Final version after Final IT2Rai review meeting

Project funded from the European Union's Horizon 2020 research and innovation program		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/05/2015 Duration: 36 months

EXECUTIVE SUMMARY

This document aims at providing a high level, simple and comprehensive idea of the exchanges between the IT2Rail services. Describing the data models used by each functional area to perform these exchanges and as well providing detailed information about the interfaces used by those functional areas.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF INTERFACES

LIST OF ABBREVIATIONS

UML	**U**nified **M**odeling **L**anguage

1. PREFACE

1.1 PURPOSE OF THIS DOCUMENT

This document provides the “Conceptual Interface Specification” (CIS) for the project. It summarises all interfaces identified and documented for each service provided by the IT2Rail Pilot. This document is a foundation to ensure the overall consistency of the development and shall guarantee the interoperability and openness of the design. It will be documented iteratively and shall be used as the reference specification of interfaces.

1.2 ASSOCIATED DOCUMENTS

Document Name	Reference
D7.2 - Development Readiness Review Pack	ITR-T7.2-E-THA-003

1.3 DOCUMENT CONVENTIONS

In order to model an abstract view of the tool, this document uses the UML notation (use case, sequence and activity diagrams).

2. OVERVIEW

This document is part of an update of the deliverable “D7.10 - Development Readiness Review Pack (FREL)”, that was prepared to be delivered on FREL phase. On this new version it includes all the updates that were designed during AREL and FREL phases.

The document is focused on the description of exchanges between different functional areas of the project and is divided into two main parts:

1. Description of the data models;
2. Description of the Interfaces.

Within the whole system several interfaces have to be defined as base for any kind of interaction between the different functional areas of the system as there are:

a. Interoperability Framework;
b. Travel Shopper;
c. Booking & Ticketing;
d. Travel Companion;
e. Trip Tracker;
f. Business Analytics.

On the diagrams below (Figure 1, Figure 2 and Figure 3) it’s possible to have a high-level overview of the interaction between different components and it’s visible the high number of exchanges between the different areas.

Certainly, there are interfaces in both directions of a link between functionalities. Therefore, each of the six functional areas has its own chapter, the containing the details of the functionalities that can be called.

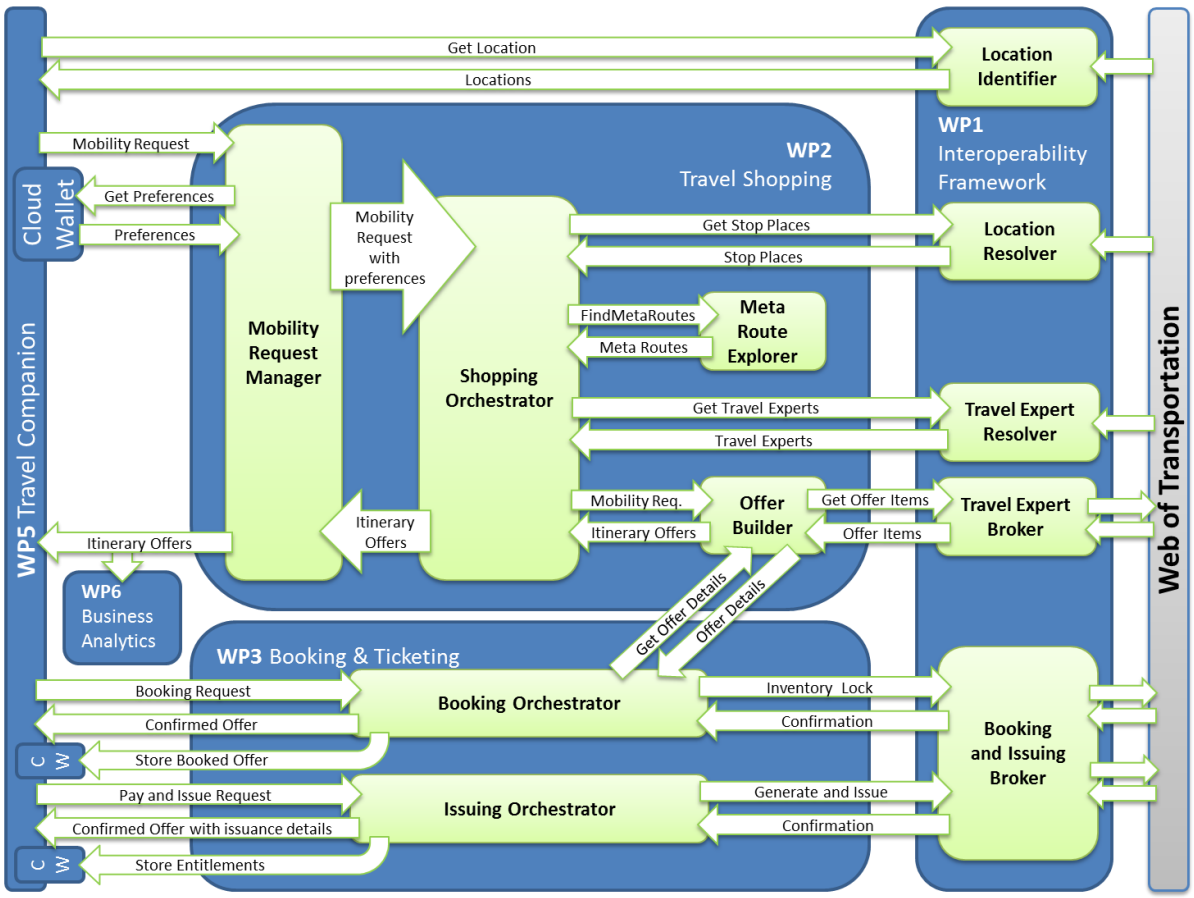


Figure 1: Shopping, booking and ticketing exchanges

This figure represent the entire flow to acquired entitlements to perform trip, starting from the traveller interaction with the traveller companion to search for location, till the last step of getting the entitlements stored on the cloud wallet.

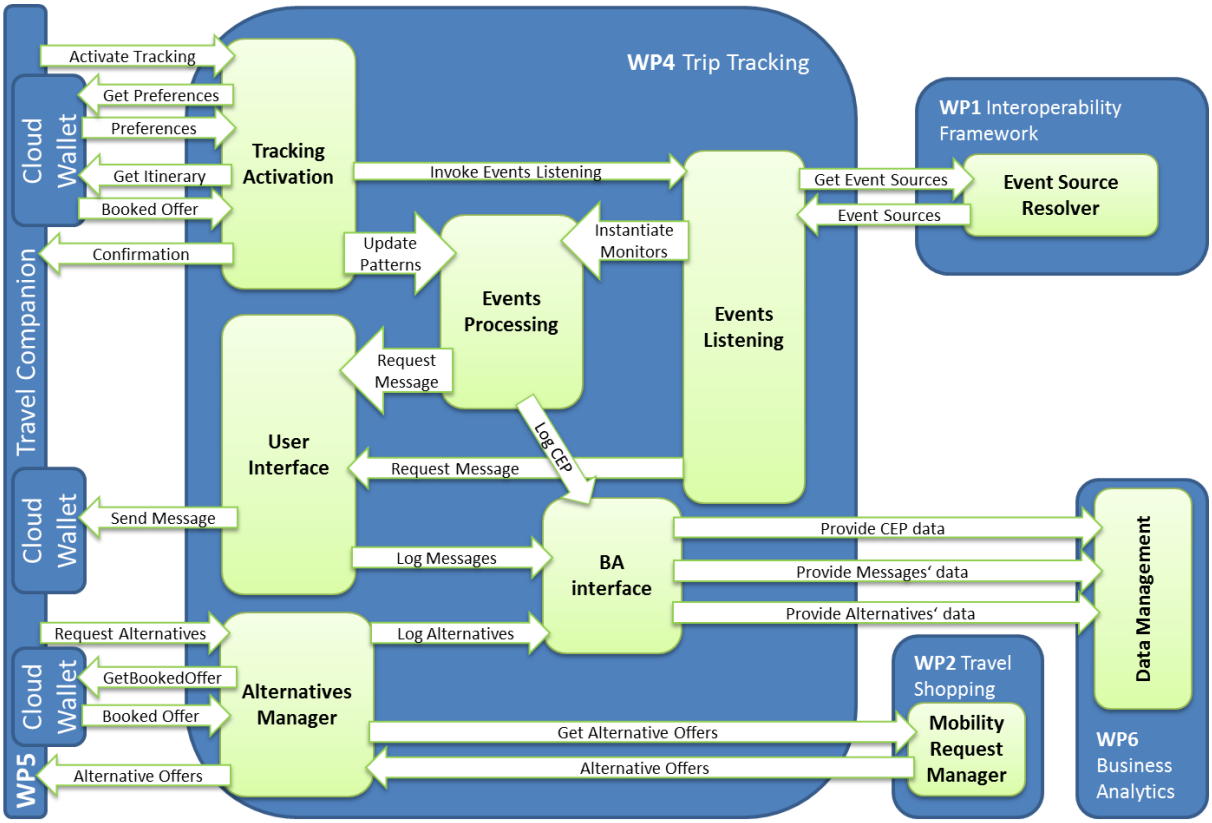


Figure 2: Trip tracking exchanges

This figure represents different flows concerning the trip tracking functionalities:

- The tracking activation triggered by the traveller at the travel companion;
- The events listening performed by monitors when invoked;
- The processing of events that triggers messages to the traveller;
- The request for alternatives raised by the traveller upon trip disruption and handled by the alternatives manager;
- Logging data for business analytics data management

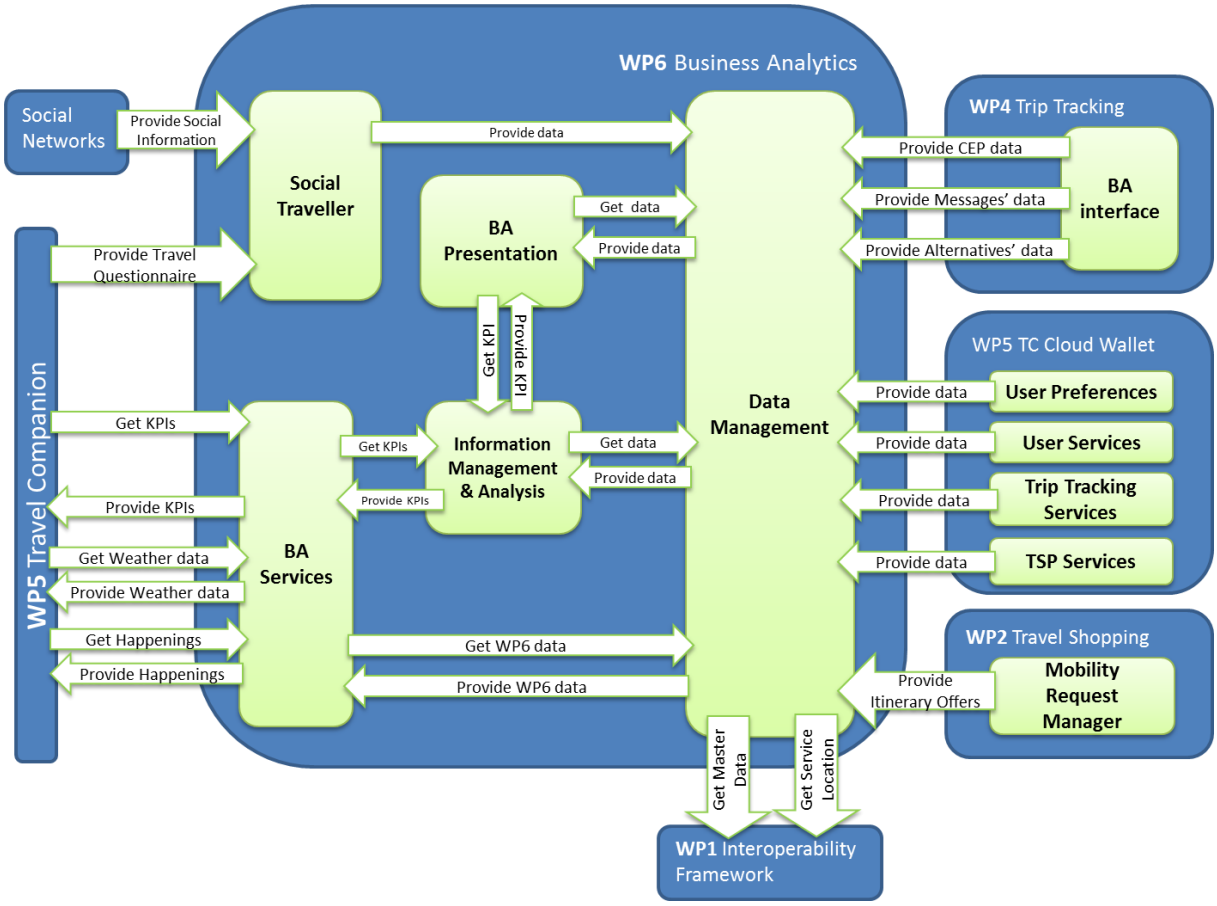


Figure 3: Business analytics exchanges

On Figure 3 it is represented all the flows to collect and made available the data handled by the Business Analytics functional area.

3. DATA MODELS

3.1 INTEROPERABILITY FRAMEWORK

3.1.1 Description

The Interoperability Framework provides a set of web services, collectively known as “Packaged Resolvers”, to the Travel Companion, the Shopping, Booking and Ticketing, and the Trip Tracker applications. Their purpose is to insulate those applications from the “mechanics” of interoperability across heterogeneous and distributed resources available at Travel Service Providers and over the world wide web, i.e. the “web of transport data”. The “Packaged Resolvers” use a common underlying framework that handles semantic interoperability and is controlled by inference rules and configuration information stored in the Interoperability Framework’s Asset Manager. In this section the high level model of input and output data entities associated with the Packaged Resolvers web services are described, as they constitute the exchange items that implement functional exchanges with the Travel Companion, the Shopping, Booking and Ticketing, and the Trip Tracker applications. A full description of the Interoperability Framework, including the Asset Manager, is provided in other Work Package 1 deliverables of the IT2Rail project.

3.1.2 Data Model

3.1.2.1 Service Descriptor

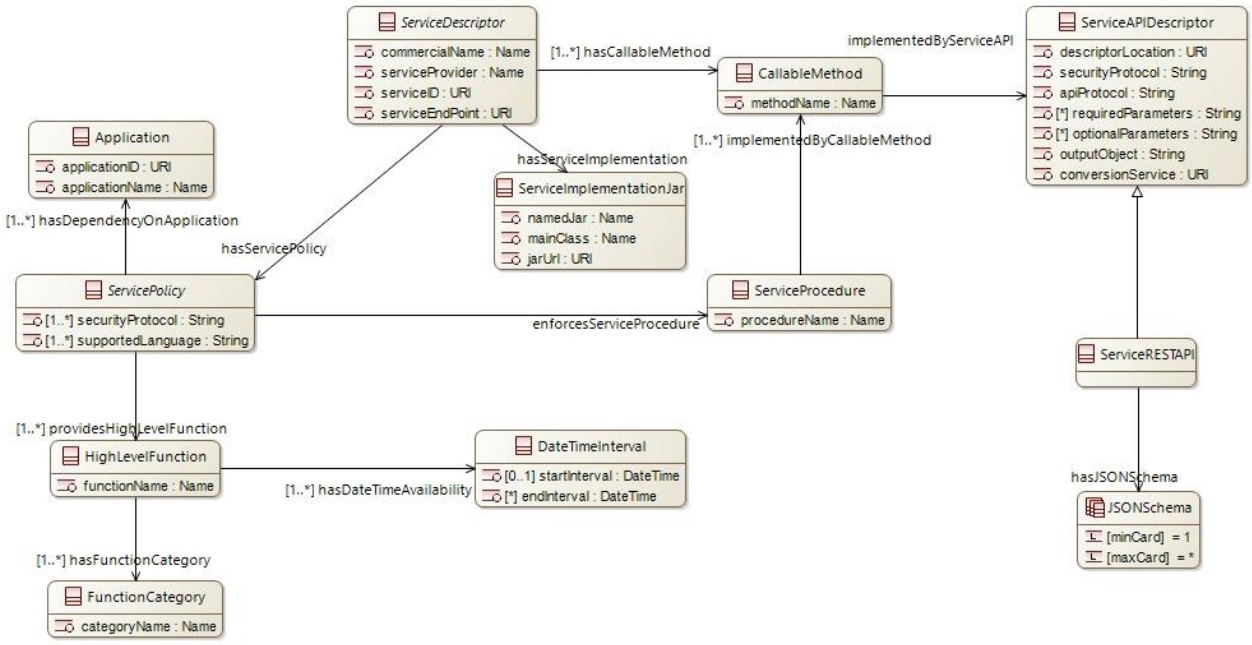


Figure 4: [CDB]L IF Semantic Web Service registry

The Service Descriptor is an abstract class that represents a generic external service provided by a Travel Service Provider. This entity is “published” in the IF Semantic Web Service manager by a Travel Service Provider advertising the service it provides to the ecosystem

3.1.2.2 Travel Expert

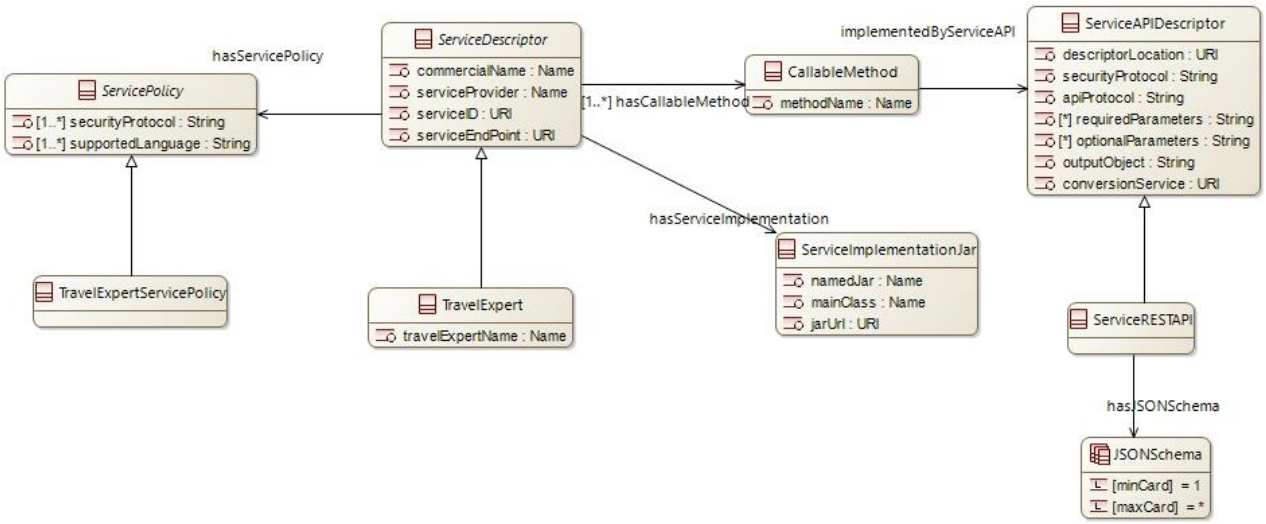


Figure 5: [CBD]L IF Travel Expert

The Travel Expert is a concrete implementation of the Service Descriptor class that represents a Travel Expert or a Booking Engine service in the IT2Rail project.

3.1.2.3 Location Identification

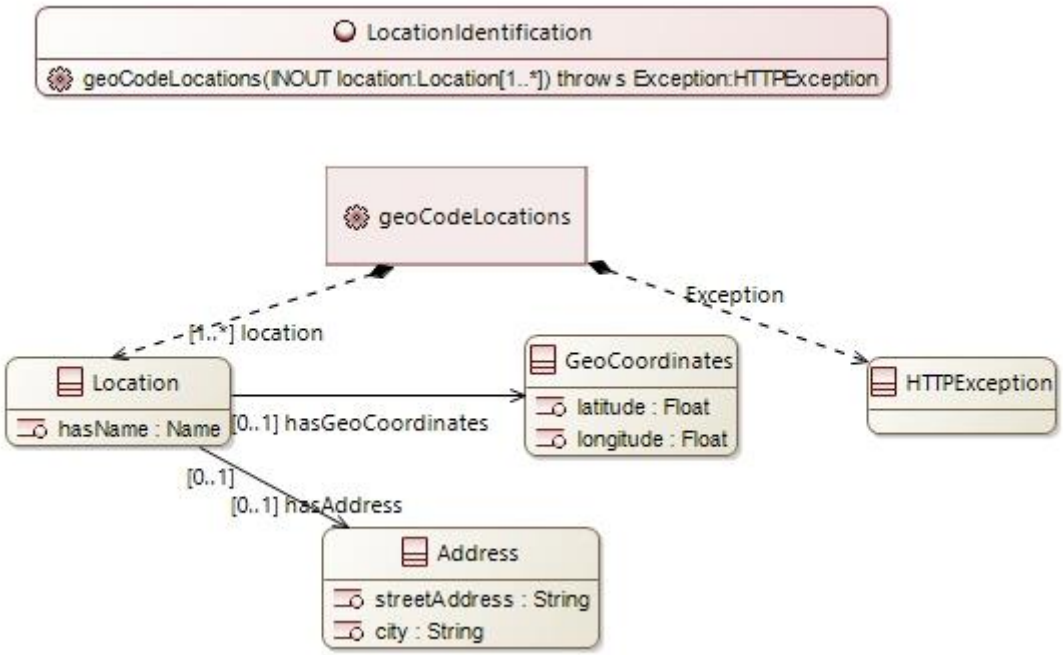


Figure 6: [CDB]L IF Location Identification EI

The geoCodeLocation operation of the Location Identification web service has one or more Location entities as input and returns a list of Locations with their associated Address and GeoCoordinates

3.1.2.4 Location Resolver

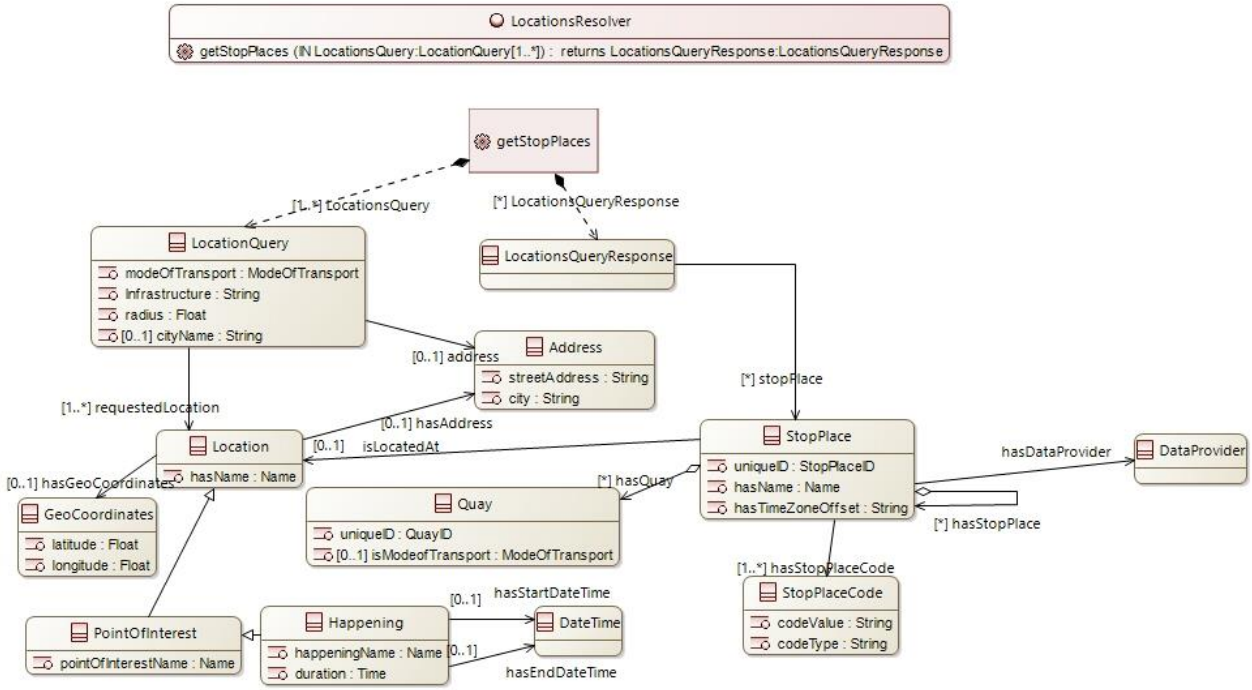


Figure 7: [CDB]L IF Location Query EI

The getStopPlaces operation of the Location Resolver web service has one or more LocationQuery entities as input and returns a LocationsQueryResponse containing a list of StopPlace

3.1.2.5 Travel Expert Resolver

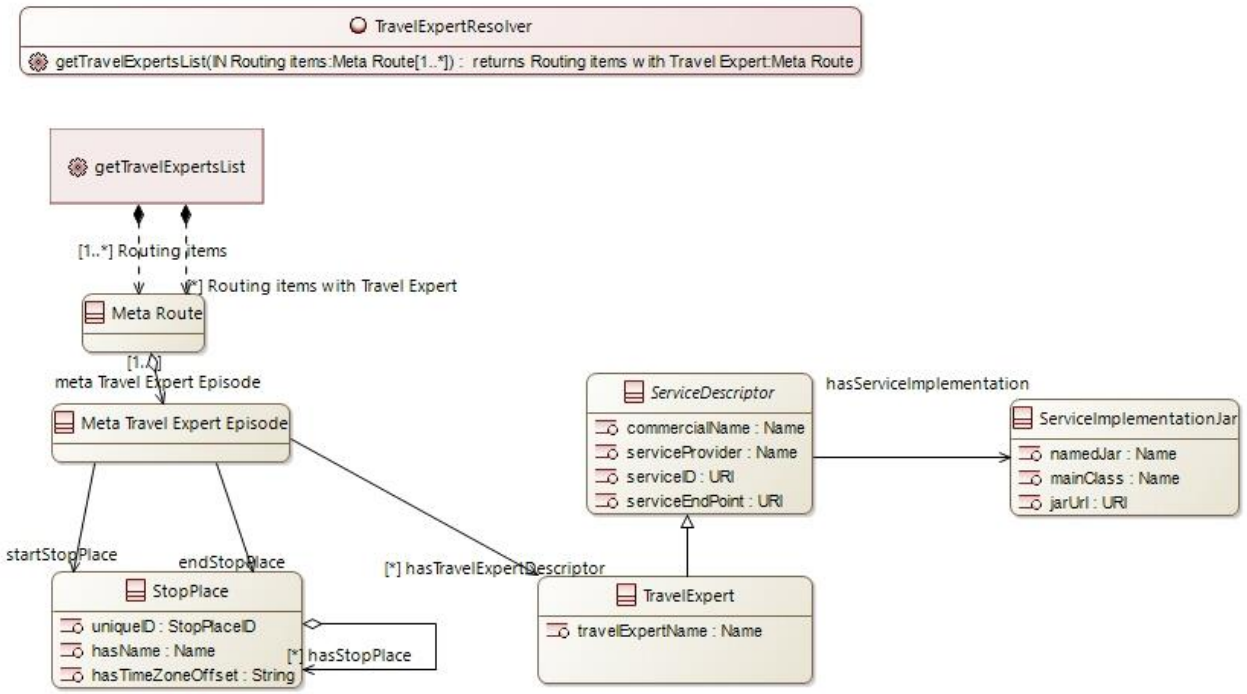


Figure 8: [CBD]L IF Travel Expert Query EI

The getTravelExpertsList operation of the Travel Expert Resolver web service has one or more MetaRoute entities as input and returns a Meta Travel Expert Episodes associated with their respective Travel Expert service descriptor

3.1.2.6 Network Statistics Provider

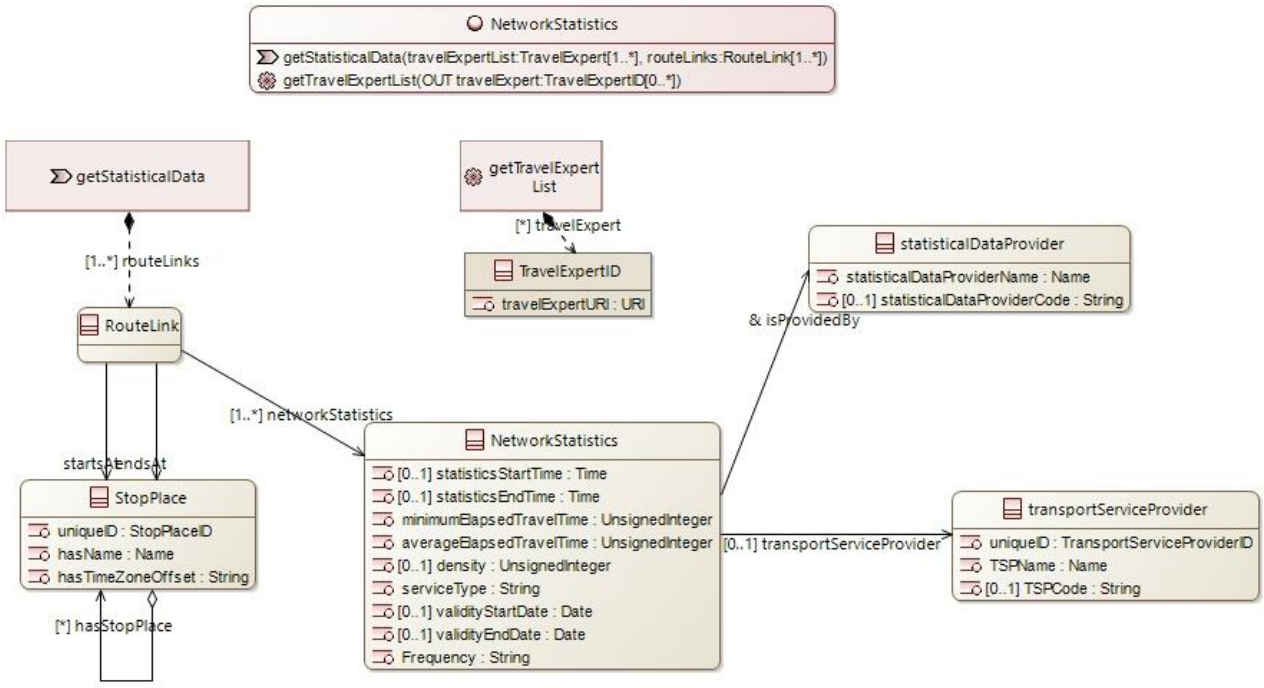


Figure 9: [CDB]L Shopping - Statistical Data For Meta Route Explorer

The getStatisticalData operation of the NetworkStatistic web service has one or more RouteLink entities as input and returns a list of RouteLink associated with their respective NetworkStatistics. The getTravelExpertsList operation of the NetworkStatistic web service has no input and returns a list of all Travel Expert ids available in the Semantic Web Services registry.

3.1.2.7 Navitia Decoder

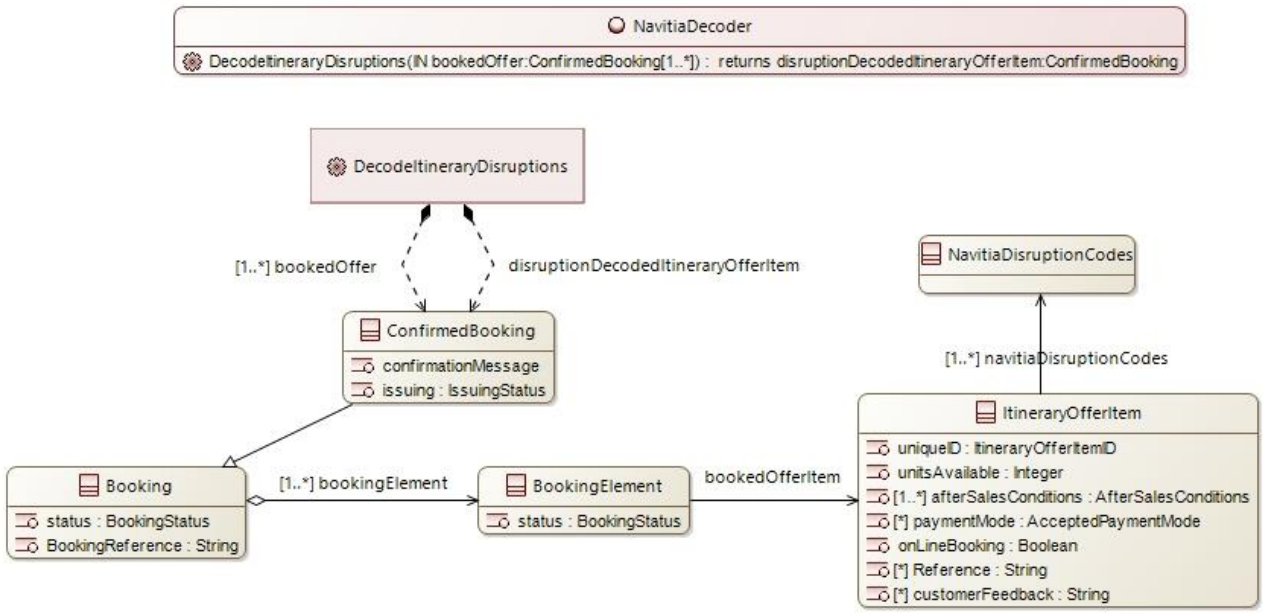


Figure 10: [CDB]L IF Disruption Navitia Decode EI

The DecodeItineraryDisruptions operation of the NavitiaDecoder web service has one or more ConfirmedBooking entities as input and returns a list of ConfirmedBooking associated with the NavitiaDisruptionCodes of the ItineraryOfferItems in the ConfirmedBooking's BookingElements

3.1.2.8 Travel Expert Broker

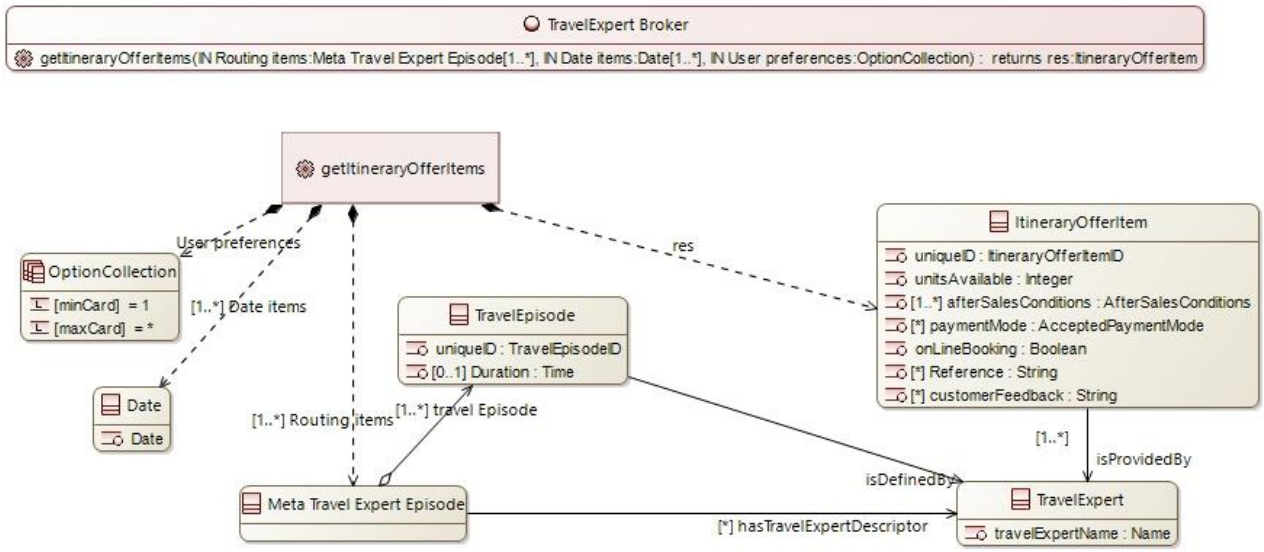


Figure 11: [CDB]L IF Travel Expert Broker EI

The getItineraryOfferItems operation of the Travel Expert Broker web service has one or more User Preferences, Meta Travel Episode and Date Items entities as input and returns a list of ItineraryOfferItems as output

3.1.2.9 Booking Engine Broker

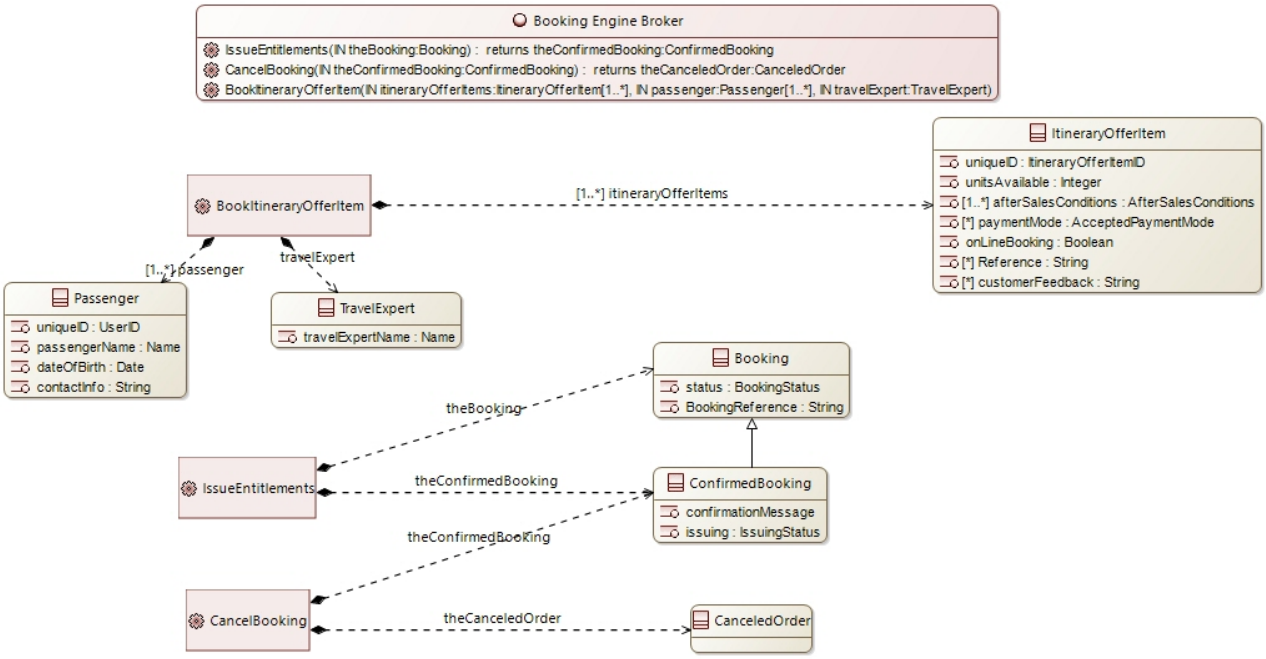


Figure 12: [CDB]L IF Booking Broker EI

The BookItineraryOfferItem operation of the Booking Engine Broker web service has one Travel Expert, or more Passengers and one or more ItineraryOfferItems entities as input and returns a Booking.

The IssueEntitlement operation of the Booking Engine Broker web service has one Booking entity as input and returns a ConfirmedBooking

The CancelBooking operation of the Booking Engine Broker web service has one ConfirmedBooking entity as input and returns a CanceledOrder

3.2.1 Description

The following chapter is describing the data organization and structure from the Shopping point of view. These elements are used in the interface definition of the Shopping components. Such elements may be reused and details may be added from other functional area point of view.

3.2.2 Data Model

3.2.2.1 Overview of the Mobility Request data model

This high-level class diagram shows the main classes and their relationships concerning the Mobility Request class model.

Some class attributes aim at clarifying the contents and use of the class.

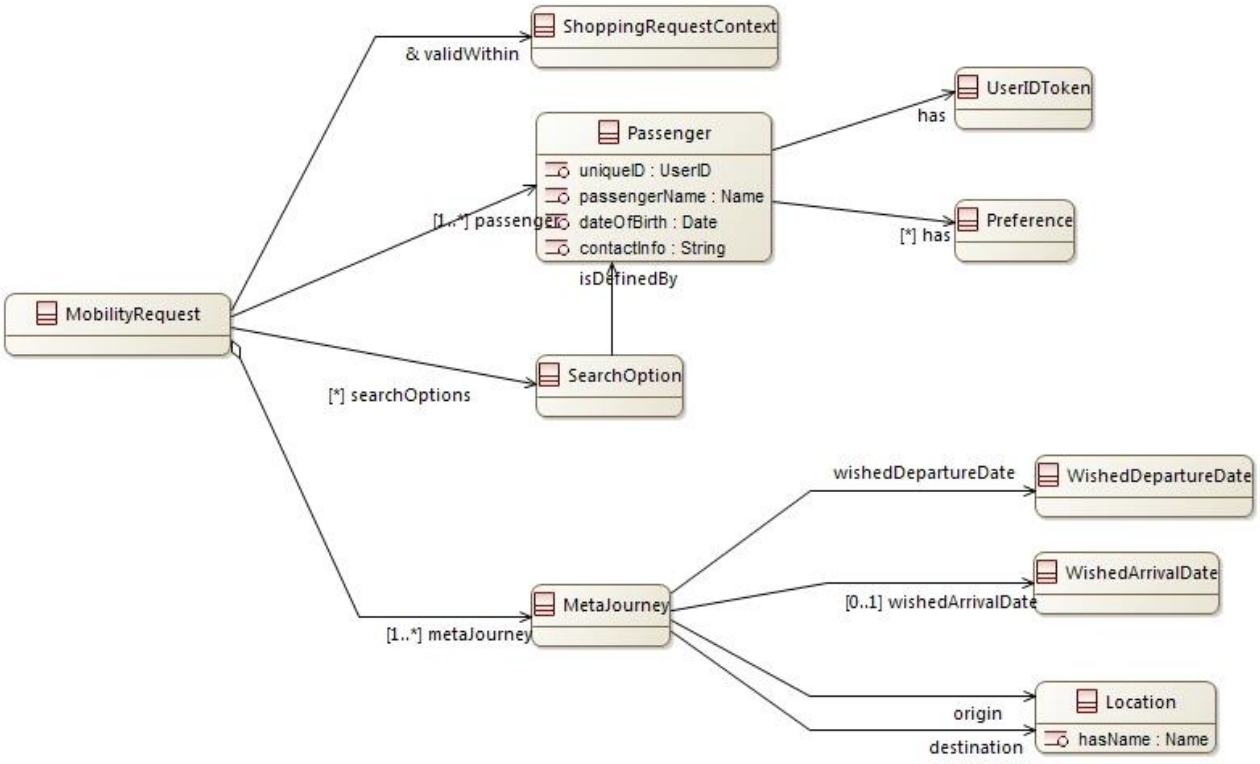


Figure 13: Mobility Request data model

This data model shows the organization of the Mobility Request in IT2Rail.

The Mobility Request contains the following items:

- Some location items: these elements correspond to the origin and the destinations of the Meta Journey, which are specified by the user in the Mobility Request;
- Some dates: the dates correspond to the moment when the user wants to start and/or finish his trips;
- Some preferences: the Mobility Request contains the set of one-time preferences, which are specified by the user. It contains also the user ID token to retrieve the user preferences, which are stored in the Travel Companion Cloud.

3.2.2.2 Overview of the Itinerary Offer data model

This high-level class diagram shows the main classes and their relationships concerning the Itinerary Offer class model.

Some class attributes aim at clarifying the contents and use of the class.

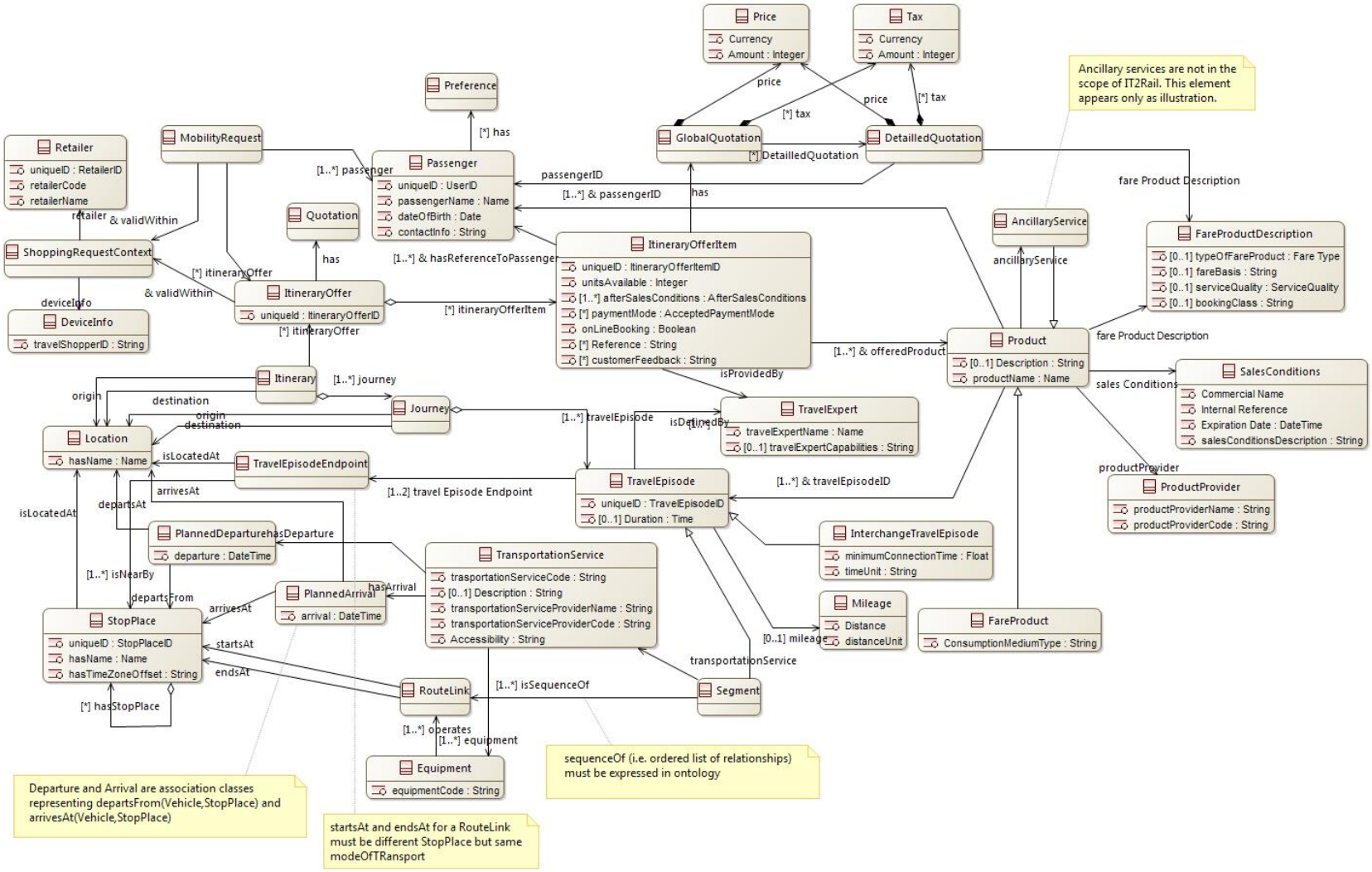


Figure 14: Itinerary Offer data model - Model reference: *[CDB]L Shopping - Itinerary Offer*

This model shows the organization of the Itinerary Offers in IT2Rail.

The Itinerary Offer is a combination of Itinerary Offer items.

An Itinerary Offer item is computed by a travel expert and corresponds to a unique combination of the following elements:

- Passenger(s). A passenger is defined by a set of preferences and his passenger ID.
- Travel episode(s). The travel episode is a part of the itinerary;
- Product(s). A product can be associated to the following elements: a reservation, a fare product description and ancillary services.

Note: Ancillary services are not in IT2Rail scope.

The Itinerary Offer contains also the description of the itineraries, which join origins and destinations specified in the Mobility Request.

3.2.2.3 Overview of the Network Data model

This high-level class diagram shows the main classes and their relationships concerning the Network Data class model.

Some class attributes aim at clarifying the contents and use of the class.

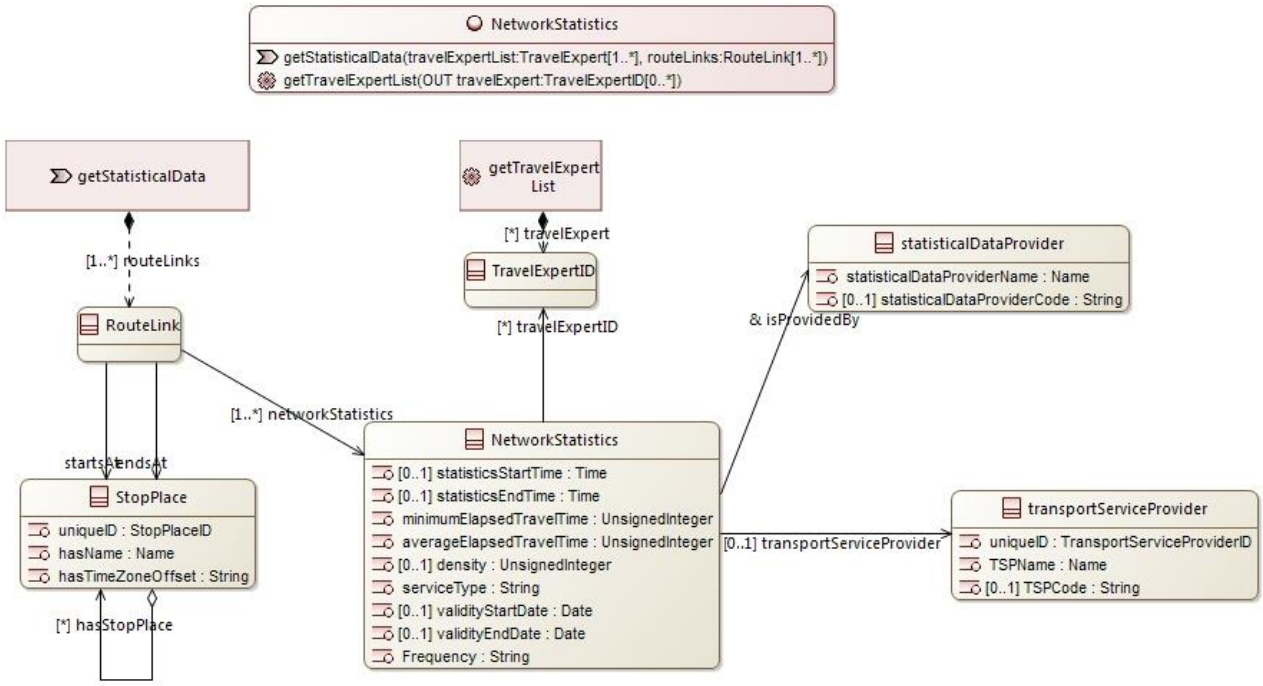


Figure 15: Statistical Data for the Meta Route Explorer model

This model shows the organization of the Network Data in IT2Rail.

The Network Data is a combination of Network daily statistics for a given list of RouteLinks.

These statistics are defined by a given Travel Expert.

3.2.2.4 Overview of the Itinerary model

This high-level class diagram shows the main classes and their relationships concerning the Itinerary class model.

Some class attributes aim at clarifying the contents and use of the class.

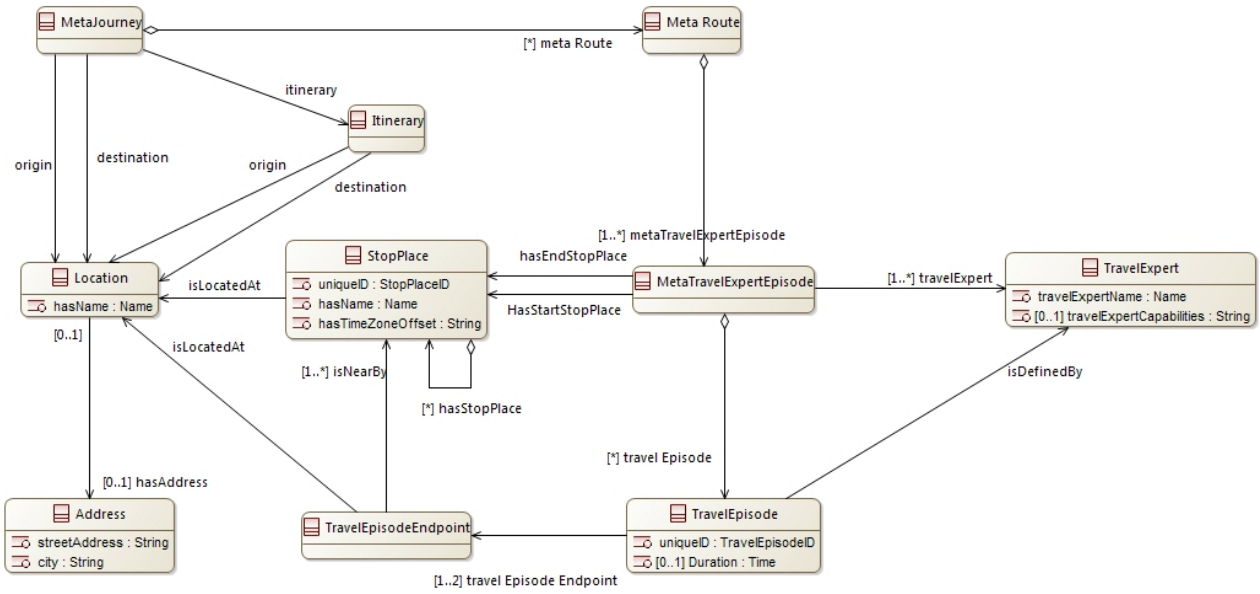


Figure 16: Itinerary model

This model shows the relationships between the itinerary and other routing items like journey, metaroute, travel episode, etc.

3.3.1 Description

This chapter contains the functional / conceptual data model of the Booking and Ticketing interfaces. It must be linked with the Interoperability Framework Ontology. Only functional data should appear: no message format, error code or envelop are described.

3.3.2 Data Model

3.3.2.1 Booking data model

This data model is used by BT Orchestrate Booking and BT Book Inventory functions (Figure 17).

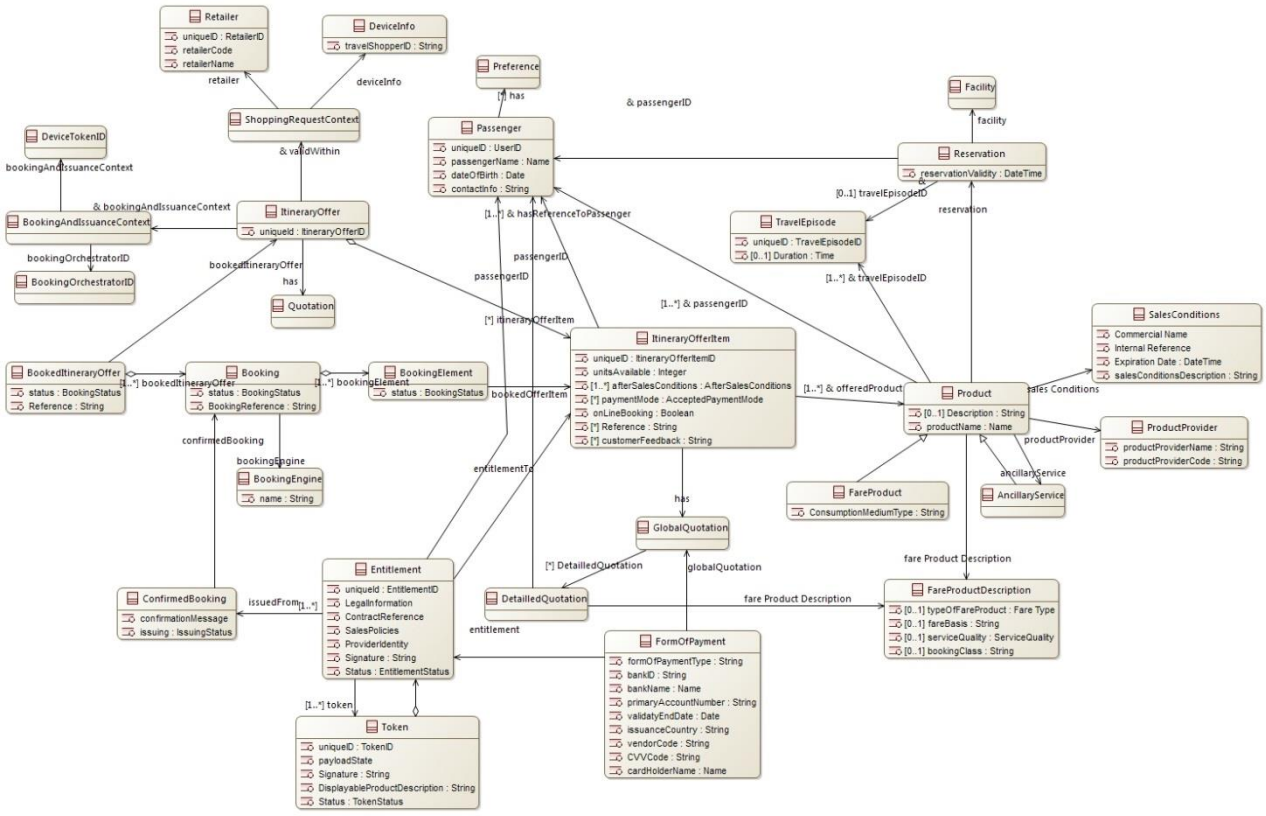


Figure 17: Booking engines data model

Model reference: *[CDB]L Booking Classes - Class Diagram Blank - FREL version: Diagram updated (*SVN Version: 814)

This model shows that a booking groups together multiple items: an itinerary offer and the booking elements. These booking elements provide information about the itinerary offer items. URI (Uniform

Resource Identifier) are used as ID to provide reference to different services (Booking orchestrator for instance).

3.3.2.2 Shopping data model

The high-level class diagram on Figure 14 shows the main classes and their relationships.

Some class attributes are given to clarify the contents and use of the class. This data model shows the organization of the itinerary offers in the IT2Rail eco-system. The key point is that an itinerary offer is composed of itinerary offer items. These items may be priced and refer to a fare product description, fare rules and a fare product dependent customization parameters. One itinerary offer item could always be linked to a travel episode (refer to the overview of Itinerary model in Travel Shopping specifications) within an itinerary. The ancillary services are considered products.

The passenger preferences must include the capacities of the Travel Companion smart device for validation.

This model describes in detail the ItineraryOfferItem object and the Fare Product Description and term and conditions included in the Fare Rules Description. Such information is mandatory for the IT2Rail scheme as it be used for the entitlement and further Shift2Rail IP4 projects.

3.3.2.3 Fulfilment data model

This high-level class diagram shows the main classes and their relationships (Figure 18).

Some class attributes are given to clarify the contents and use of the class.

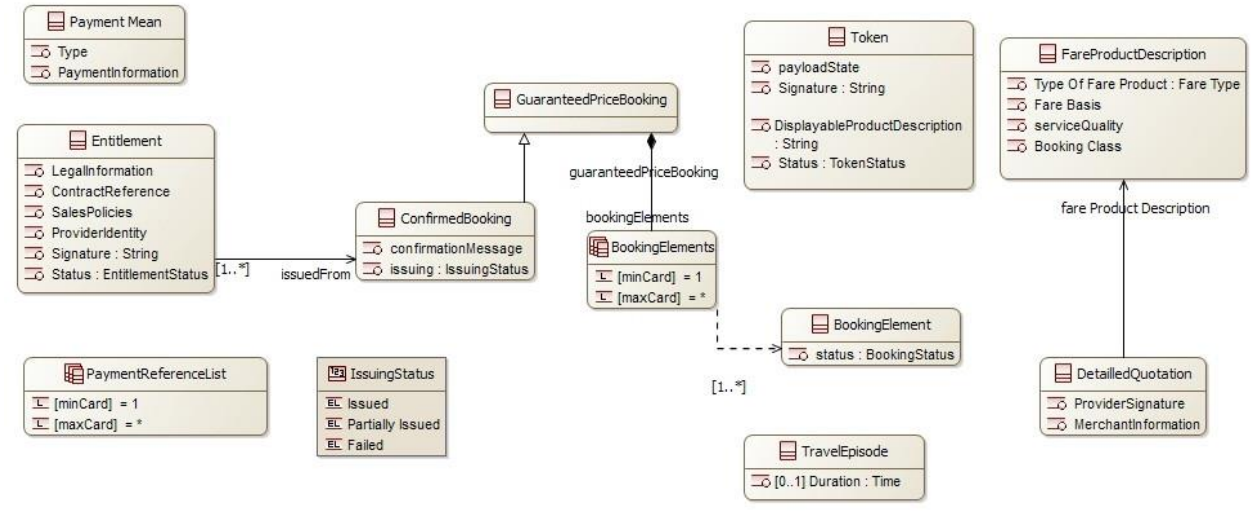


Figure 18: FulfillmentEngine data model

Model reference: *[CDB]L Fulfillment - Orchestrate & issuing - Classes* (SVN Version: 814)

This data model adds some detail regarding the payment means data and the status of multiple bookings (BookingElements and ConfirmedBooking).

3.3.2.4 Payment data model

This high-level class diagram shows the main classes and their relationships (Figure 19).

Some class attributes are given to clarify the contents and use of the class.

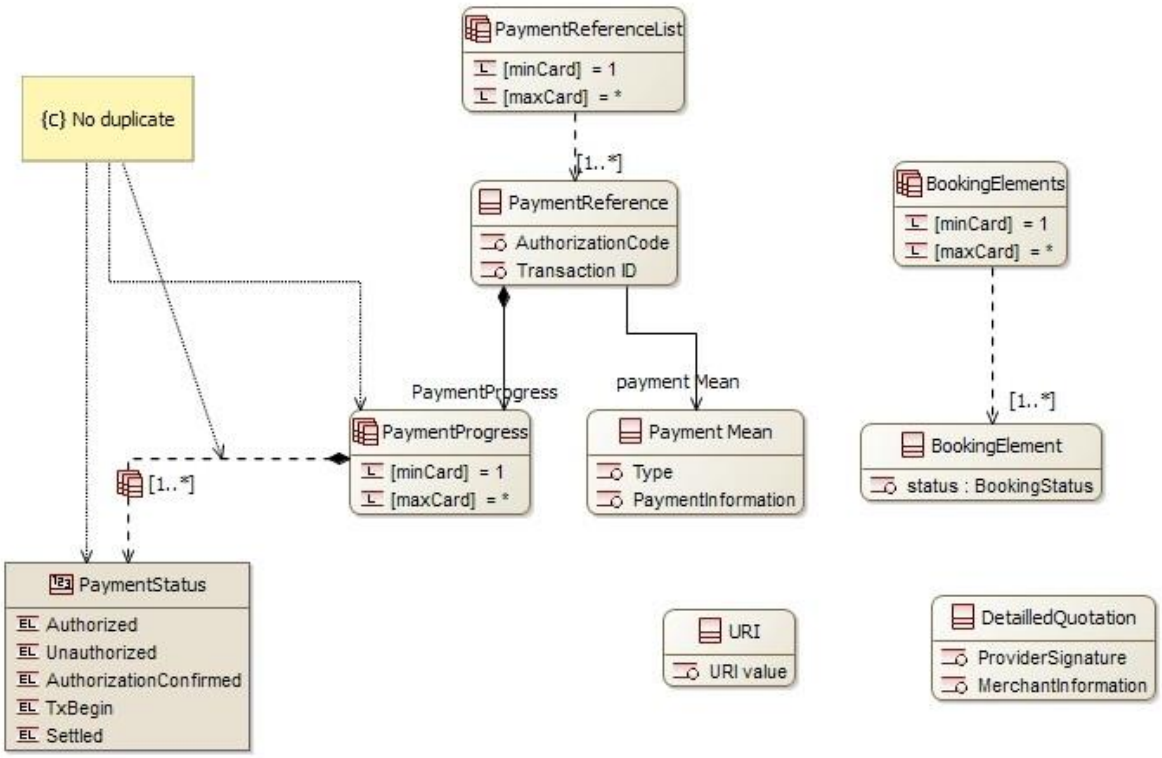


Figure 19: Payment data model

Model reference: *[CDB]L Payment - Orchestrator - Classes (*SVN Version: 814)

This data model is provided to detail the parameters of the payment and issuing interfaces. The PaymentProgress element is an ordered list with no duplicate of payment status that is used to track the progress of the two-step payment process for each payment reference.

3.3.2.5 Validation data model

The following schema details the model used by Booking and Ticketing for the Entitlement and Token that Transport Service Provider issue (Figure 20).

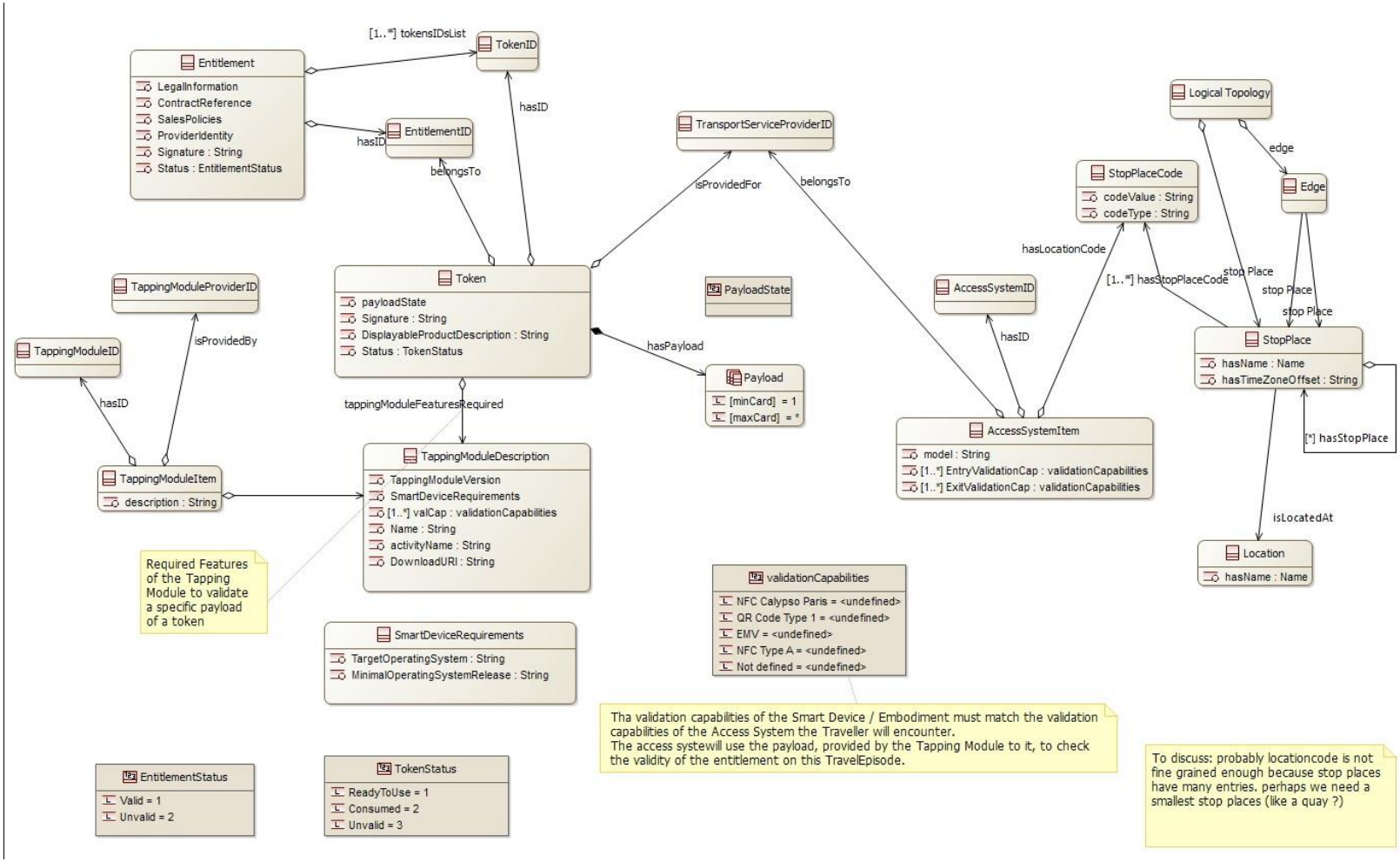


Figure 20: Entitlement & Token data model

Model reference: *[CDB]L Validation Capabilities management (*SVN Version: 814)

In this schema it is important to notice that:

- Payload is a collection of Bytes. This is the information that a particular Tapping Module is able to use for validation;
- The Tapping Module ID: must allow the Travel Companion application to successfully download and execute the Tapping Module compatible for this particular Token.

3.3.2.6 Fare product data model

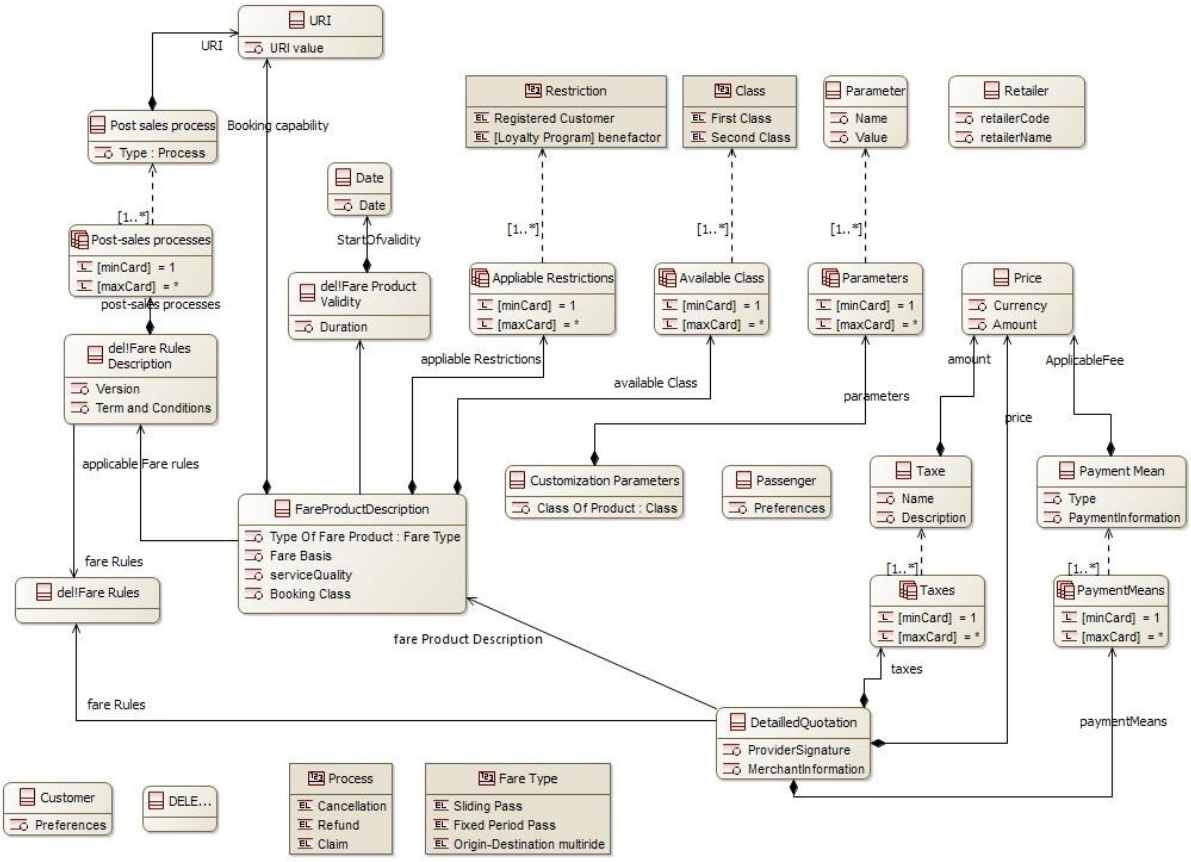


Figure 21: Fare product data model

Model reference: *[CDB]L TSP Fare Product Engine - Classes (*SVN Version: 814)

Warning: This is an example of a possible fare product data model. It may differ for each TSP.

The important point in this fare product data model (Figure 21) is that it models a description of the fare product rather than the actual fare structure of every fare product owner. As a result, the central object in this diagram is the FareProductDescription.

3.4.1 Description

This chapter contains the functional / conceptual data model of the Trip Tracker interfaces. It must be linked with the ontology.

3.4.2 Data Model

3.4.2.1 Activate Tracking data model

This diagram shows the main exchange items, classes and their relationships concerning the request of the activate tracking data model.

Some class attributes aim at clarifying the contents and use of the class.

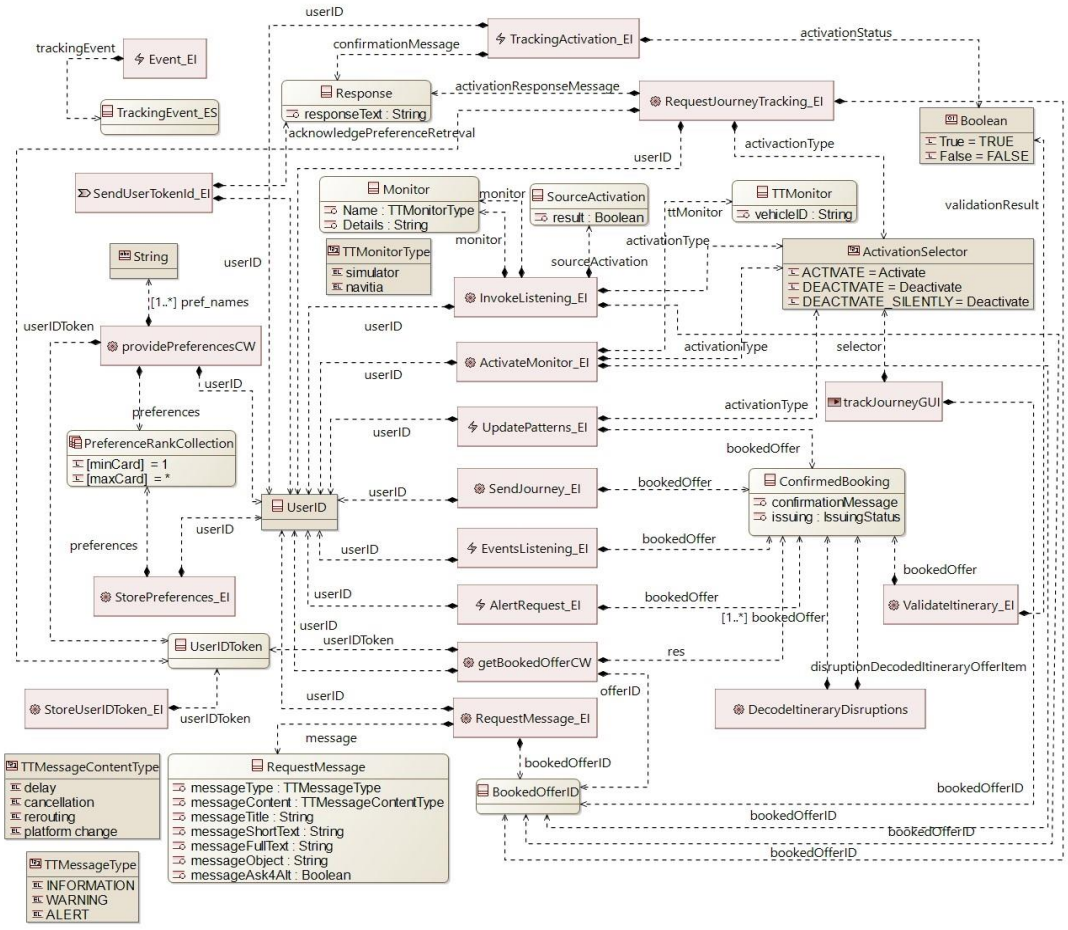


Figure 22: Activate Tracking data model

Model reference: *[CDB]L TT Activate Tracking*

This data model specifies the organization of the Activate Tracking with the following classes:

- ActivationSelector – is an Enumeration type of values: activate, deactivate, deactivate_silently;
- Boolean – is a Boolean type with the meaning of the activation or the deactivation of the tracking or the itinerary validation;
- BookedOfferID – is used to uniquely distinguish each BookedOffer;
- ConfirmedBooking – is a sequence of consecutive TravelEpisode(s). Each journey is based on a couple of origin and destination;
- Monitor – identifies the event source;
- PreferenceRankCollection – is a collection of user preferences within the tracking activation process and the aim is to receive those preferences, which are relevant for trip tracking; This enables to customize Trip Tracker's behaviour according to Traveller's preferences;
- RequestMessage – enables to request the Trip Tracker for sending the message to Travel Companion;
- Response – informs about the success/failure of taking the request into account;
- SourceActivation – is a Boolean type with the meaning of the activation or the deactivation of the listening of the event´s source;
- String – is enumerating of the preferences in request;
- TrackingEvent_ES – The event when the validation of the itinerary is not valid;
- TTMessageContentType – corresponds to „Message content" in TT Preferences;
- TTMessageType – is also defined to easily distinguish the impact of the message
- TTMonitor – the activation of the particular event source based on vehicleID;
- TTMonitorType – the determination of the particular event source;
- UserID – is used to uniquely distinguish each user;
- UserIDtoken – is used to uniquely distinguish each user.

3.4.2.2 Perform Event Processing data model

This diagram shows the main exchange items, classes and their relationships concerning the display of the Perform Event Processing data model.

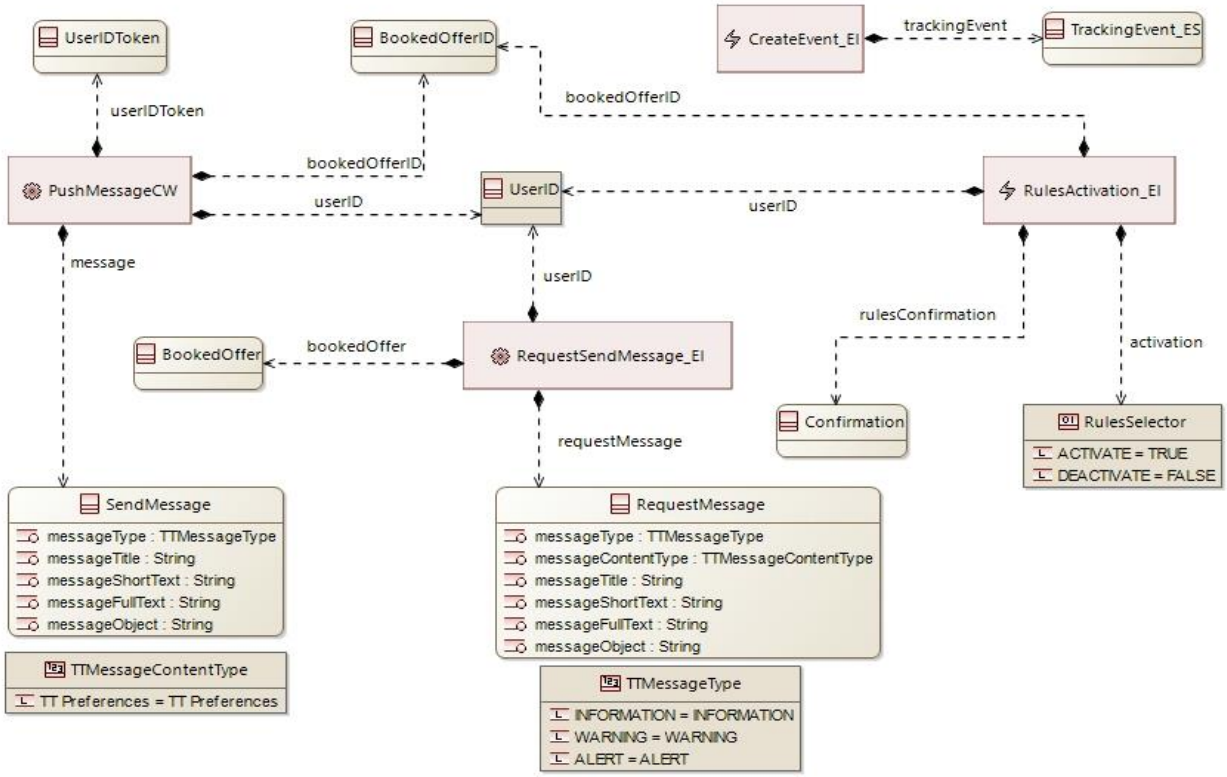


Figure 23: Perform Event Processing data model

Model reference: *[CDB]L TT Perform Event Processing*

This data model specifies the organization of the Perform Event Processing with the following classes:

- ArrivalDelay – the event related to the delay of the arrival;
- ArrivalDelayEvent – the structure of the event related to the delay of the arrival which is logged for further processing in the Business Analytics;
- ArrivalDelayEventRules – the activation of the rules related to ArrivalDelayEvent. When the rules are matched, they will be used as a threshold for determining which type of message;
- BookedOfferID – is used to uniquely distinguish each BookedOffer;
- DepartureDelay – the event related to the delay of the departure;
- DepartureDelayEvent – the structure the event related to the delay of the departure which is logged for further processing in the Business Analytics;
- DepartureDelayEventRules – the activation of the rules related to DepartureDelayEvent; When the rules are matched, they will be used as a threshold for determining which type of message;
- Disruption – the disruption identified by event source which may have the impact on a journey;
- DisruptionEffect – the type of the disruption's effect which Trip Tracker is able to recognize;

- DisruptionStatus – determines the disruption's status, whether the disruption is active, whether it happened in the past or is expected/scheduled in the future;
- RequestDate – the timestamp identifying when the request for the activation/deactivation of the rules was sent;
- RequestMessage – enables to request the Trip Tracker for sending the message to Travel Companion;
- RulesConfirmation – confirmation of the rule's activation or deactivation;
- RulesDeactivationRequest – the request related to the deactivation of the rules;
- StopPointStatus – the determination of the departure´s status: added, deleted, delayed, and unchanged;
- TrackingEvent_ES – is the detected event from the events sources;
- TTMessageContentType – corresponds to „Message content" in TT Preferences;
- TTMessageType – is also defined to easily distinguish the impact of the message;
- TTStopPlace – determines the place where the delay of the arrival/departure occurred;
- UserID – is used to uniquely distinguish each user;
- UserIDtoken – is used to uniquely distinguish each user.

3.4.2.3 User Interface data model

This diagram shows the main exchange items, classes and their relationships concerning the request of the User Interface data model.

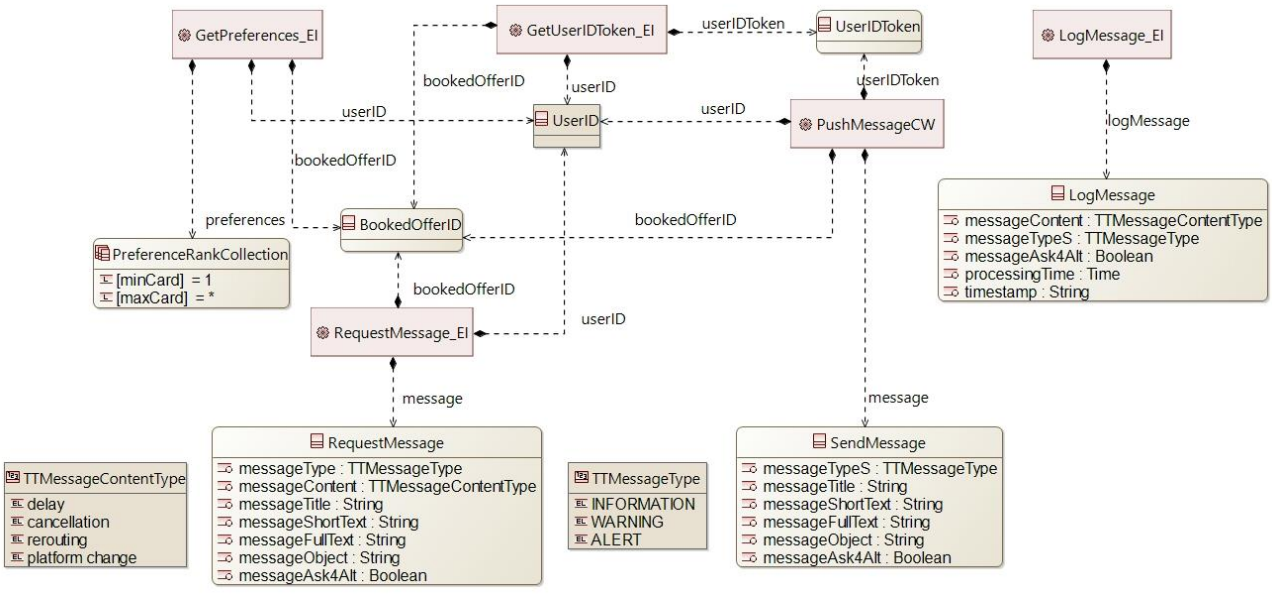


Figure 24: User Interface data model

Model reference: *[CDB]L TT User Interface*

This data model specifies the organization of the User Interface with the following classes:

- BookedOfferID – is used to uniquely distinguish each BookedOffer;
- LogMessage – the data structure related to the message which is logged for further processing in the Business Analytics;
- PreferenceRankCollection – is a collection of user preferences within the tracking activation process and the aim is to receive those preferences, which are relevant for trip tracking; This enables to customize Trip Tracker's behaviour according to Traveller's preferences;
- RequestMessage – enables to request the Trip Tracker for sending the message to Travel Companion;
- SendMessage – aims at sending a message to the Traveller through the Travel Companion;
- TTMessageContentType – corresponds to Message content in TT Preferences;
- TTMessageType – is also defined to easily distinguish the impact of the message;
- UserID – is used to uniquely distinguish each user;
- UserIDtoken – is used to uniquely distinguish each user.

3.4.2.4 Trip Tracker's Preferences data model

This diagram shows the main exchange items, classes and their relationships related to the Trip Tracker´s preferences. This set of Trip Tracker´s preferences is only a sample for the project purposes. These preferences can be modified and extended in the follow-up projects.

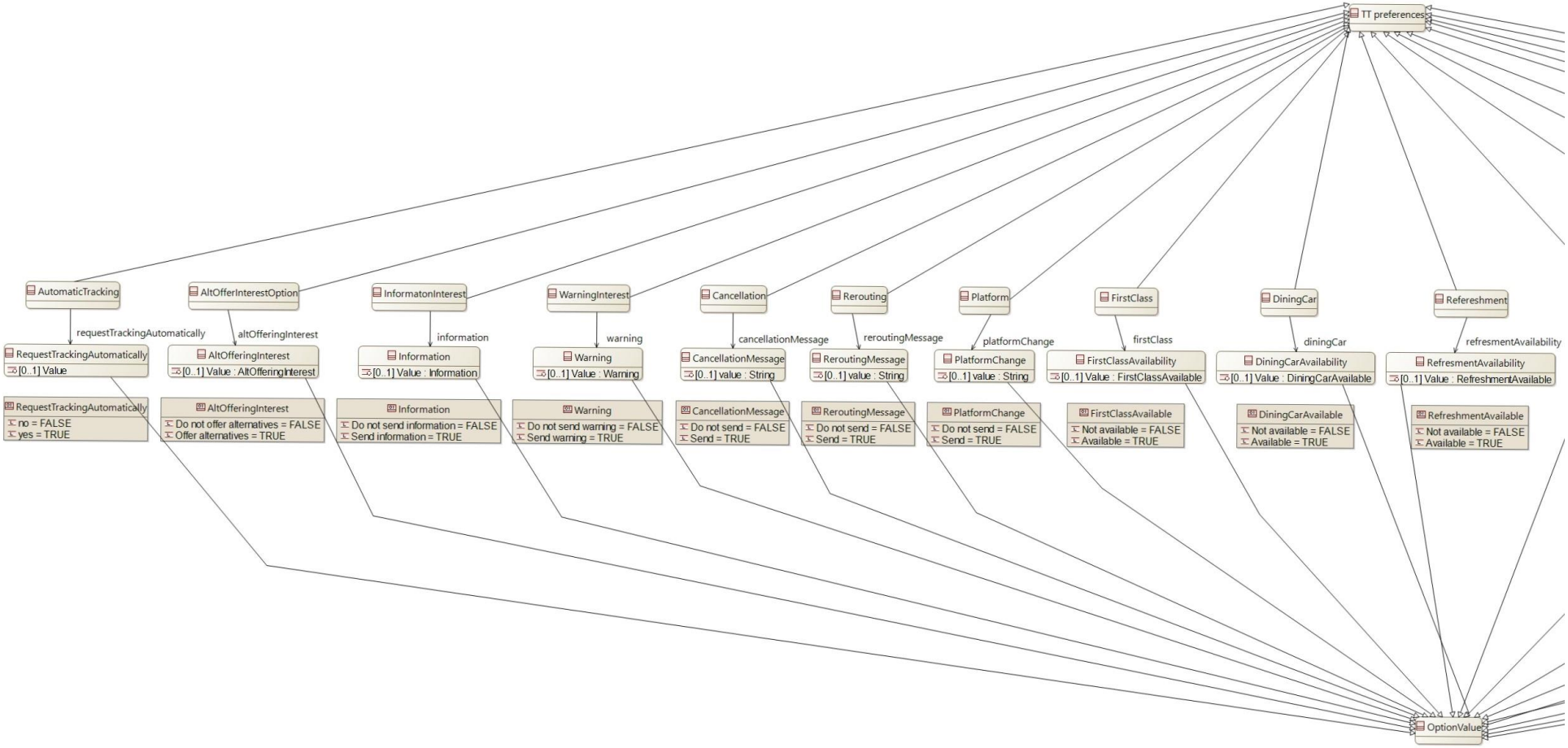


Figure 25: Trip Tracker's Preferences data model (part 1)

Model reference: *[CDB]L TT Preferences*

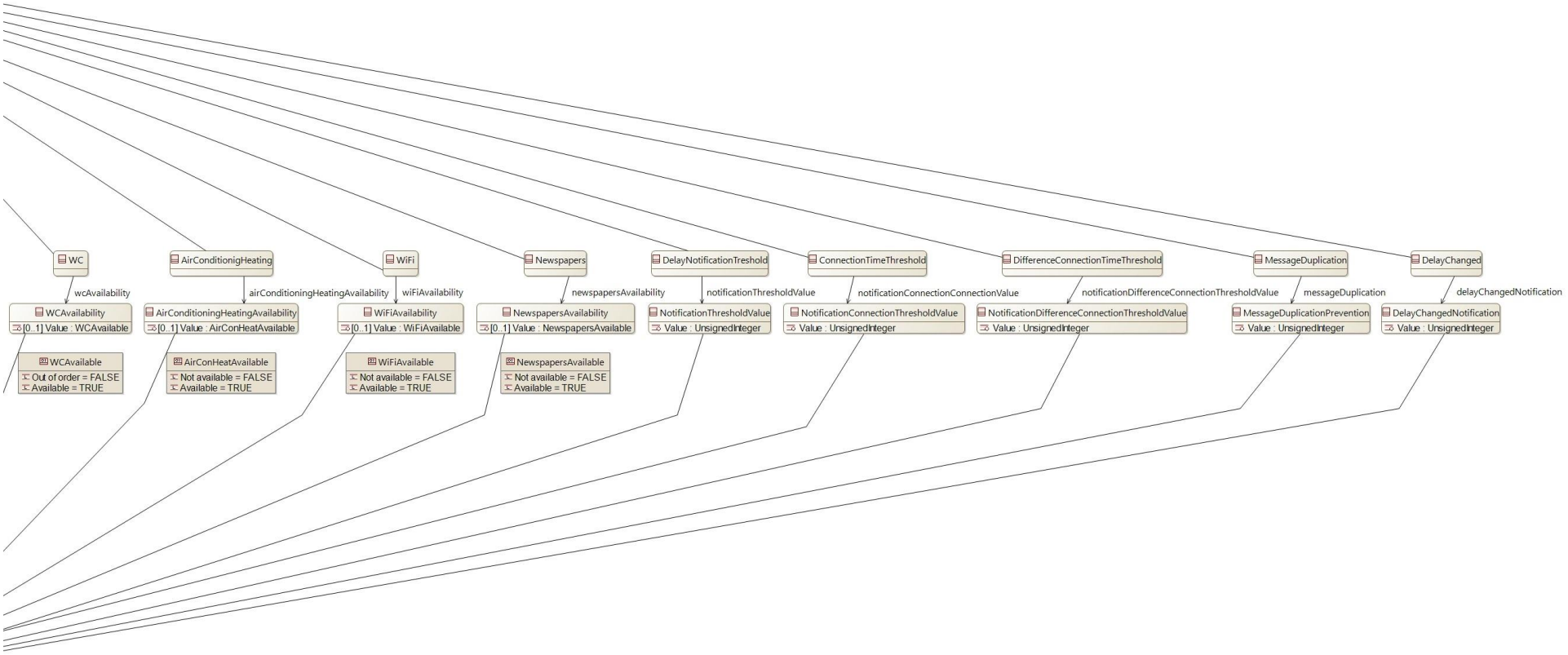


Figure 26: Trip Tracker's Preferences data model (part 2)

Model reference: *[CDB]L TT Preferences*

This data model specifies the organization of Trip Tracker's preferences with the following classes:

- AutomaticTracking – Request journey tracking automatically, when a new itinerary is stored in the CW;
- AltOfferInterestOption – Ask about interest in alternatives offering, where applicable;
- InformationInterest; WarningInterest – Enable/disable receiving messages of selected type;
- Cancellation; Rerouting; Platform; FirstClass; DiningCar; Refreshment; WC; AirConditioningHeating; WiFi; Newspapers – display only messages selected by the user, i.e. avoid displaying unwanted messages;
- DelayNotificationTreshold – Don't display a warning, when the delay is less than given time;
- ConnectionTimeTreshold – Don't display a warning, when remaining connection time is less than given value;
- DifferenceConnectionTimeTreshold – Don't display a warning, when the difference between remaining and minimum connection time is less than given value;
- MessageDuplication – Don't display the same message again, if less than given time since the last display elapsed;
- DelayChanged – Consider the delay unchanged (so the delay message still the same), if it has changed by less than given value.

3.4.2.5 Manage Alternatives data model

This diagram shows the main exchange items, classes and their relationships concerning the request of the Manage Alternatives data model.

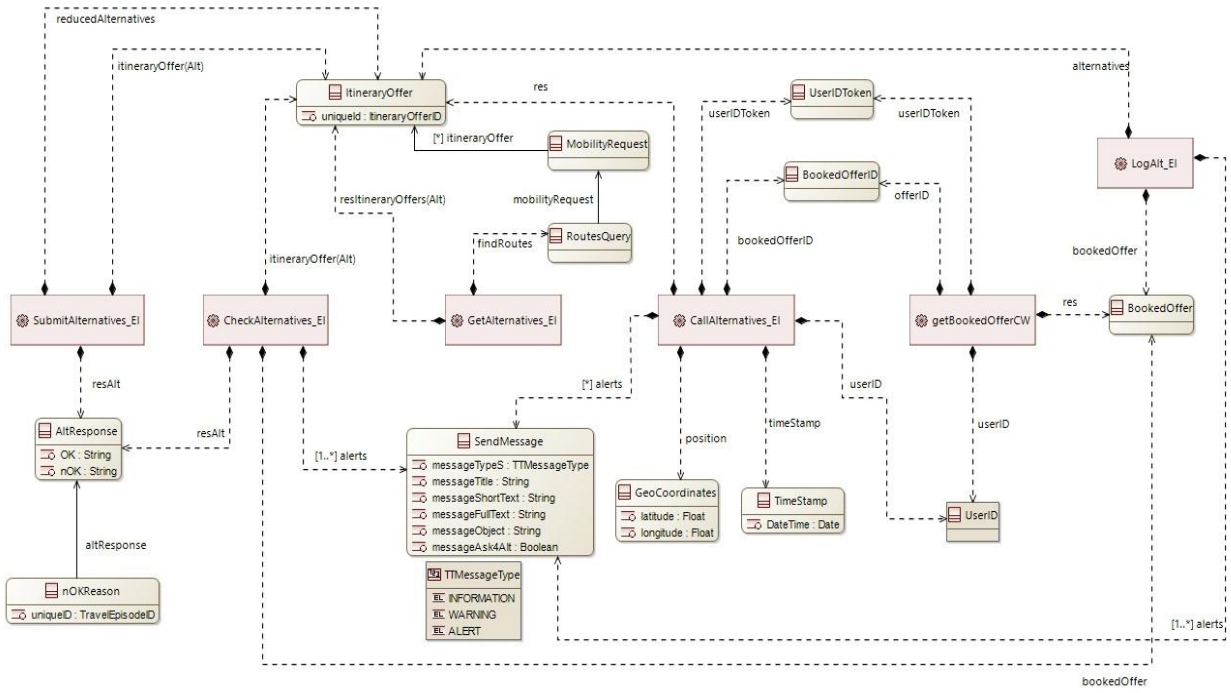


Figure 27: Manage Alternatives data model

Model reference: *[CDB]L TT Manage Alternatives*

This data model specifies the organization of the Manage Alternatives with the following classes:

- AltResponse – is the response whether the check of alternatives was successful or not;
- BookedOffer – is a sequence of consecutive TravelEpisode(s). Each journey is based on a couple of origin and destination;
- BookedOfferID – is used to uniquely distinguish each BookedOffer;
- GeoCoordinates – Coordinates of a point expressed in decimal format (WS84);
- ItineraryOffer – is the response in the form of the offer of the alternatives to the request of the alternatives;
- MobilityRequest – is the Traveller's query for travel information about a specific itinerary;
- nOKReason – identifies travel episodes for which the checking was not successful;
- RoutesQuery – the selected route by the Traveller – the origin, the destination and the transport mode;
- SendMessage – aims at sending a message to the Traveller through the Travel Companion;
- TimeStamp – the moment when the Trip Tracker received the request for the calculation of alternatives;
- TTMessageType – is also defined to easily distinguish the impact of the message;
- UserID – is used to uniquely distinguish each user;
- UserIDToken – is used to uniquely distinguish each user.

3.4.2.6 BA Cooperation data model

The following diagram shows the main exchange items, classes and their relationships concerning the request of the BA Cooperation - CEP data model.

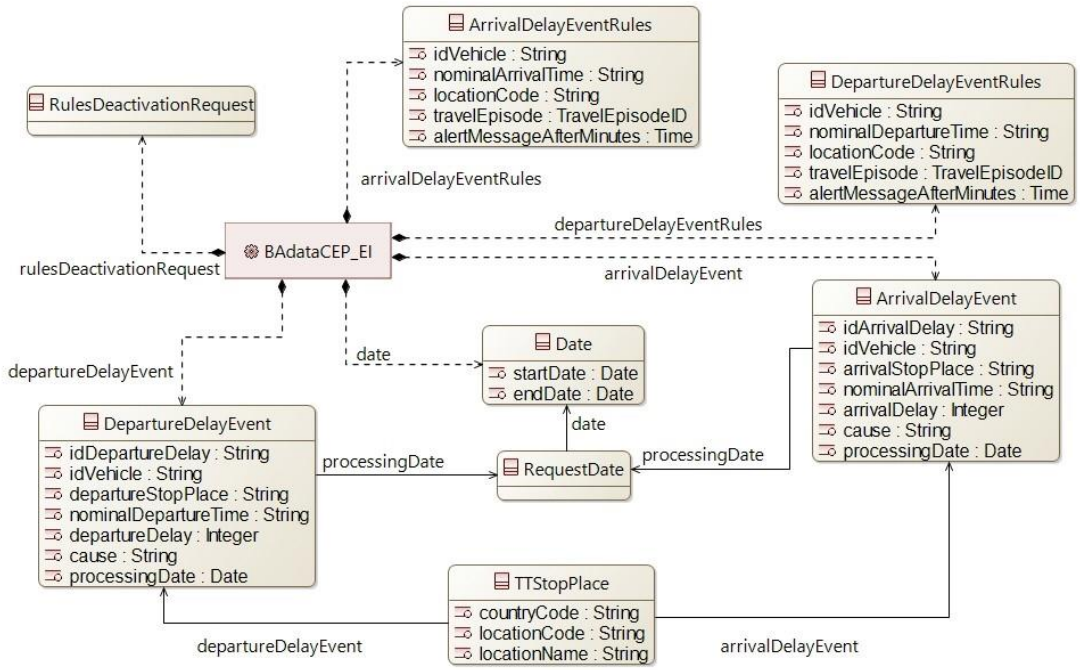


Figure 28: BA Cooperation - CEP data model

Model reference: *[CDB]L TT BA Cooperation CEP*

This data model specifies the organization of the BA Cooperation - CEP with the following classes:

- ArrivalDelayEvent – the event related to the delay of the arrival;
- ArrivalDelayEventRules – the activation of the rules related to ArrivalDelayEvent. When the rules are matched, they will be used as a threshold for determining which type of message;
- DepartureDelayEvent – the event related to the delay of the departure;
- Date – date on the basis of which all the required information about the events shall be sent;
- DepartureDelayEventRules – the activation of the rules related to DepartureDelayEvent. When the rules are matched, they will be used as a threshold for determining which type of message;
- RequestDate – identifies the date when the request for the activation/deactivation of the rules was sent;
- RulesDeactivationRequest – the request related to the deactivation of the rules;
- TTStopPlace – the determination of the place where the delay of the arrival/departure occurred.

The following diagram shows the main exchange items, classes and their relationships concerning the request of the BA Cooperation - alternatives data model.

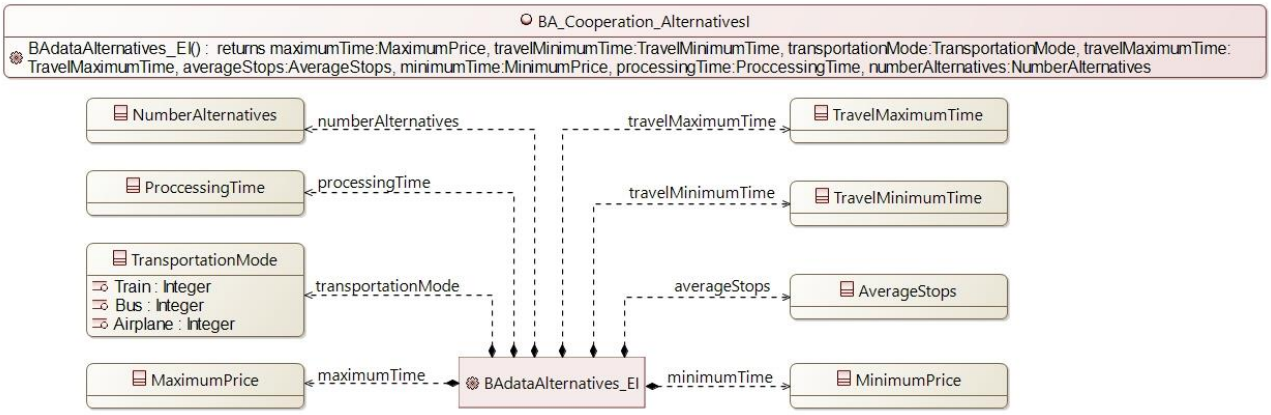


Figure 29: BA Cooperation - alternatives data model

Model reference: *[CDB]L TT BA Cooperation Alternatives*

This data model specifies the organization of the BA Cooperation - alternatives with the following classes:

- AverageStops – the average of the AverageAlternatives field;
- MaximumPrice – the maximum of the AveragePrice field;
- MinimumPrice – the minimum of the AveragePrice field;
- NumberAlternatives – number of alternatives submitted in given period;
- ProccessingTime – average of the ProcessingTime field;
- TransportationMode – count of the TransportStops field by type;
- TravelMaximumTime – the average of the TravelMaximumTime field;
- TravelMinimumTime – the average of the TravelMinimumTime field.

3.5.1 Description

The following chapter is describing the data organization and structure from Travel Companion point of view; These elements are used in the interface definition of the Travel Companion components; Such elements may be reused and detail may be added from other functional area point of view;

3.5.2 Data Model

Within the Travel Companion application, the Tapping Modules are manipulating the data described in the following schema;

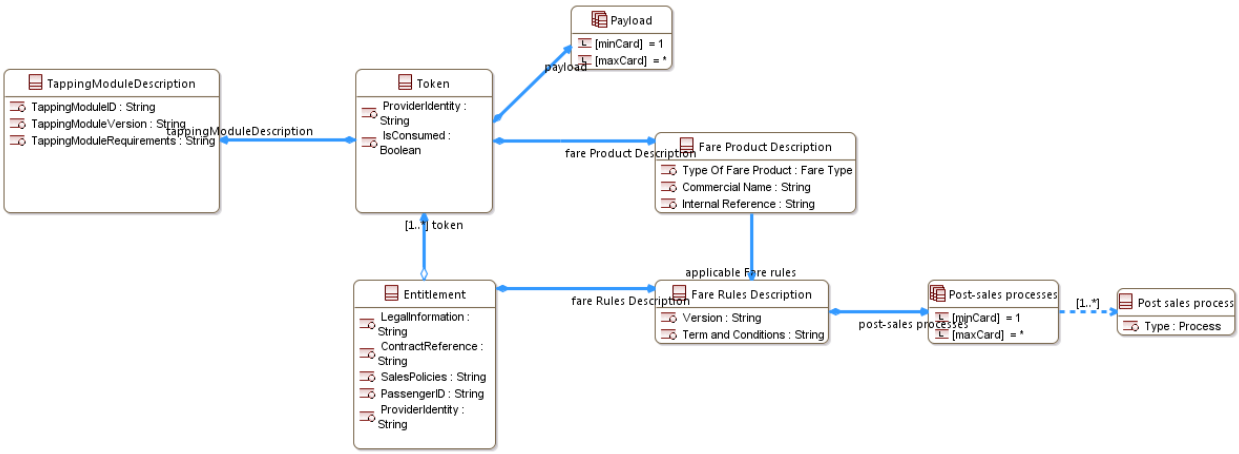


Figure 30: Tapping module data model

This diagram shows the relationship between the Token and the TappingModule; The Token is compatible with an identified version of a TappingModule that must be used to perform the validation process; How does the TappingModule uses the Token is implementation dependent.

Figure 31 shows some of the types of data that are referenced in the operations invoked and called on TC; It does not show all types of data which TC handles, but only those that are more specific to it, which are essentially, though not only, the ones that TC generates.

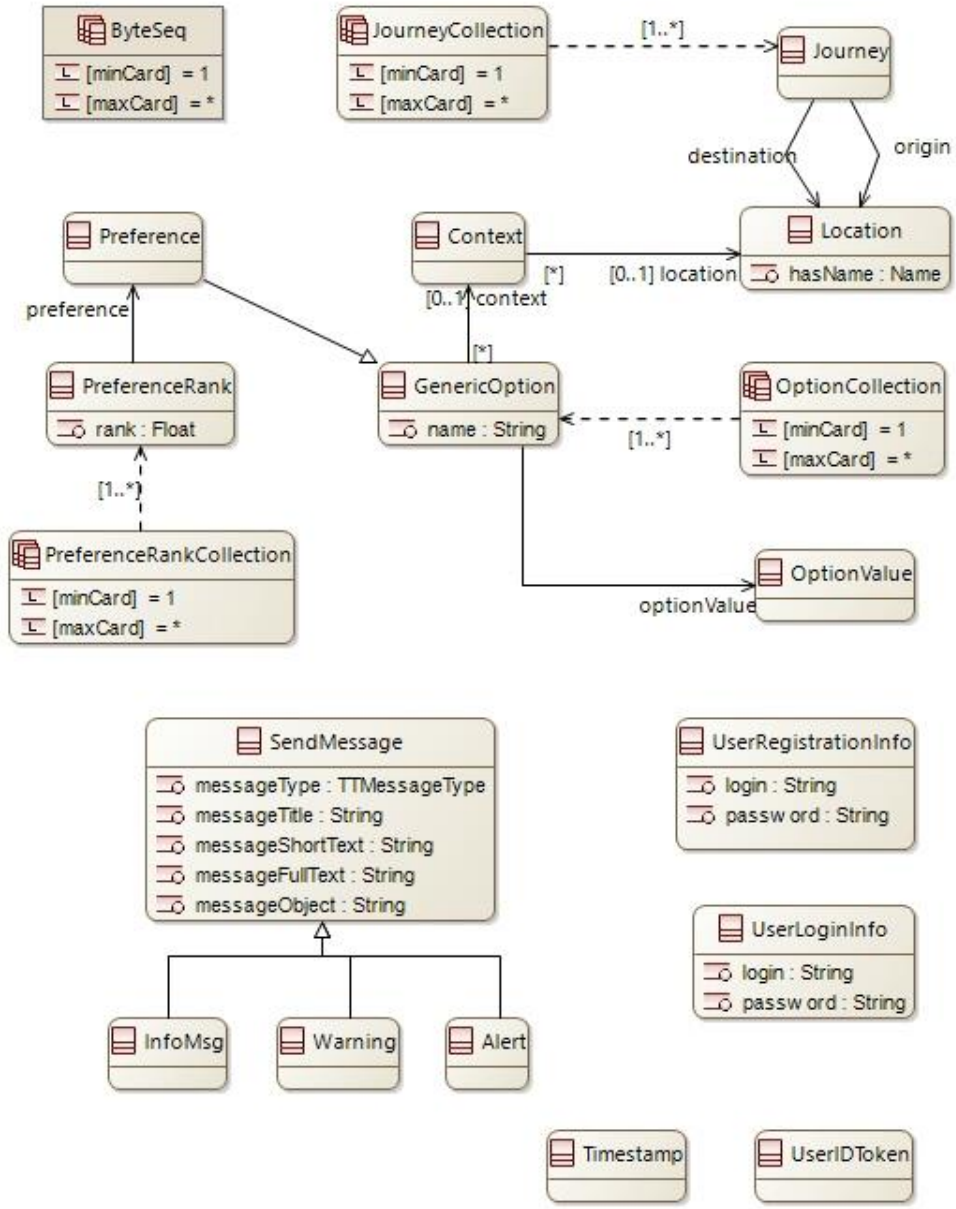


Figure 31: Classes defining the types of data specific to TC

More precisely, the information that TC generates is in particular related to the traveller: her identity, her preferences, etc;

Naturally, TC handles many kinds of data related to the traveller's trips, such as itineraries, offers, bookings, entitlements, and so on; However, for simplicity these data are not described in this document, but they are left for companion specifications of other IT2Rail modules.

3.6.1 Description

The next section contains the class diagram designed by using the Capella tool as well as a short description of each involved class;

3.6.2 Data Model

3;6;2;1 Overview of the BA internal data model

The following figure (Model reference: "***[CDB]L BA Internal Data***"diagram, SVN version 794) describes the data needed by WP6 to compute its KPIs:

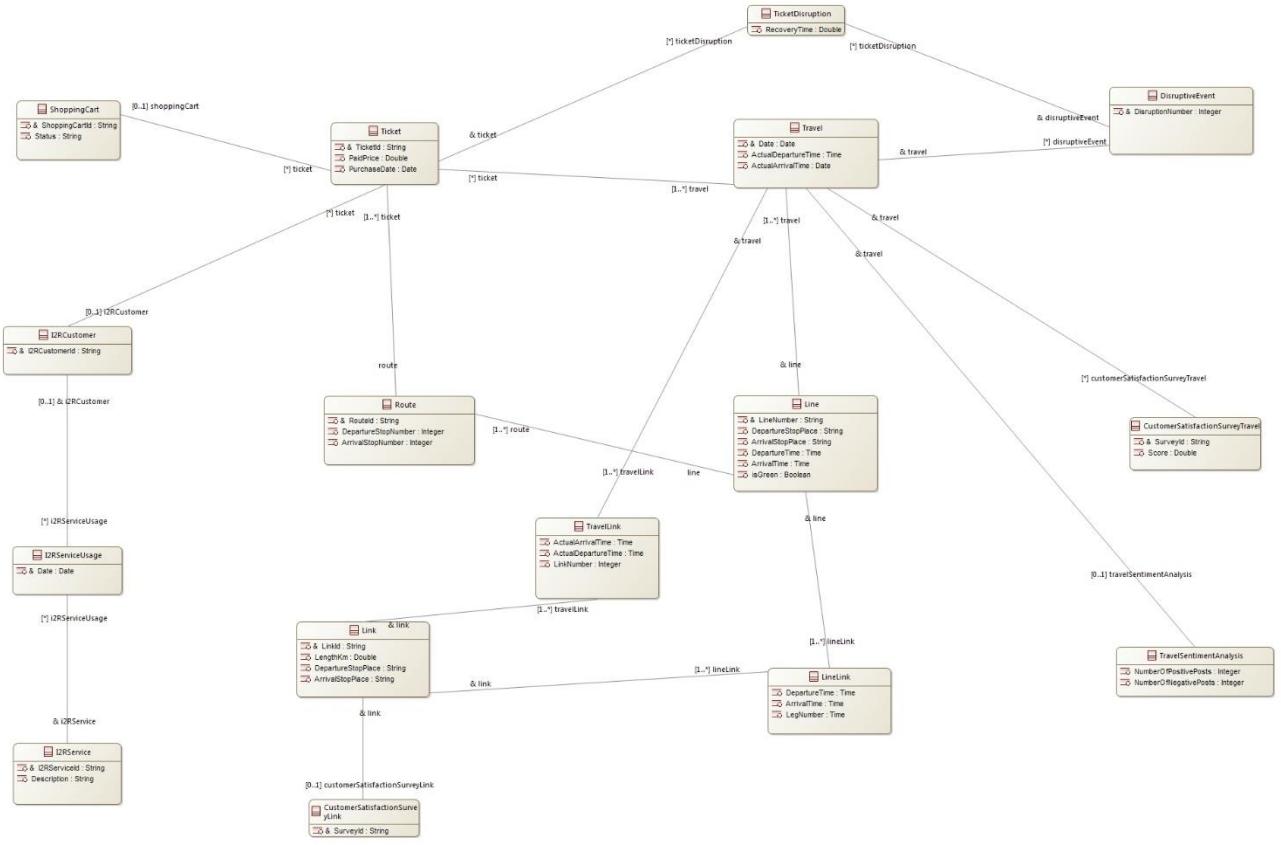


Figure 32: The Business Analytics internal class diagram

A brief description of the employed classes follows:

Link: It is a pair of consecutive stopping places; It is characterized by departure and arrival stopping places, and length;

Line: Represents the key element of the transportation network, e;g; a train line or a flight; It is characterized by departure and arrival stopping places, scheduled departure and arrival times, and a flag indicating whether this line is green;

LineLink: Connects a line with the links which it encompasses; It is characterized by the scheduled arrival and departure times, and by the number of the link within the line;

Route: Fragment of the line, delimited by two stops; It is associated with its line, and characterized by the numbers of the delimiting stops;

Travel: Travel of a line, executed in a specific date; It is associated with its line, and characterized by the actual departure and arrival times;

TravelLink: Connects a travel with the links which it encompasses; It is characterized by the actual arrival and departure times, and by the number of the link within the travel;

Entitlement-WP6: Entitlement for traveling; It is associated with a route and a travel; It might be associated with a shopping cart, if it was bought online, and with a customer, if the customer is a registered IT2Rail user; It is characterized by purchase date and paid price;

DisruptiveEvent: Represents an event which has an impact on a travel; It is associated with the disrupted travel;

EntitlementDisruption: Connects the disruptive events with the entitlements they affected; It is characterized by the recovery time;

CustomerSatisfactionSurveyTravel: Represents a survey conducted to assess the satisfaction of a user about a specific travel; It is associated with the surveyed travel, and characterized by the score provided by the user;

CustomerSatisfactionSurveyLink: Represents a survey conducted to assess the satisfaction of a user about a specific link; It is associated with the specific link, and characterized by the score provided by the user;

TravelSentimentAnalysis: Represents the sentiment analysis executed on the data derived from social networks regarding a specific travel; It is associated with the analyzed travel and characterized by the number of positive and negative retrieved posts;

ShoppingCart: Represents a shopping cart for entitlement purchase; It is characterized by a status, i;e; “completed”, “abandoned” or “in use”;

I2RCustomer: Represents a customer registered within IT2Rail;

I2RService: Represents an ancillary service provided by IT2Rail;

I2RServiceUsage: Represents the usage of an IT2Rail service by an IT2Rail user, in a specific date;

3;6;2;2 Overview of the BA KPI data model

The following figure (Model reference: "***[CDB]L BA KPI***" diagram, SVN version 794) shows the main exchange items, classes and their relationships concerning KPIs:

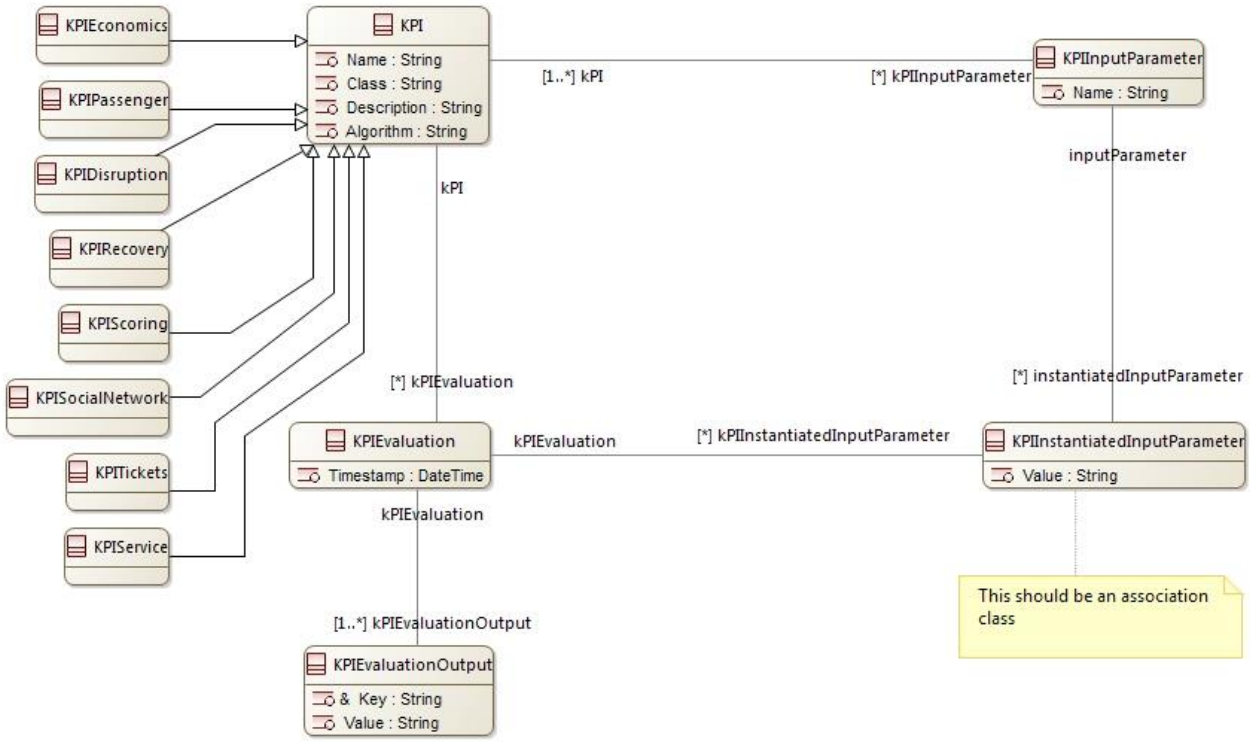


Figure 33: The Business Analytics KPI class diagram

4. INTERFACES

4.1 GENERAL OVERVIEW

On this section it will be described the operations available per interface; This description will be done using the following interface description structure:

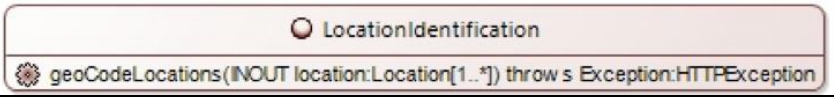
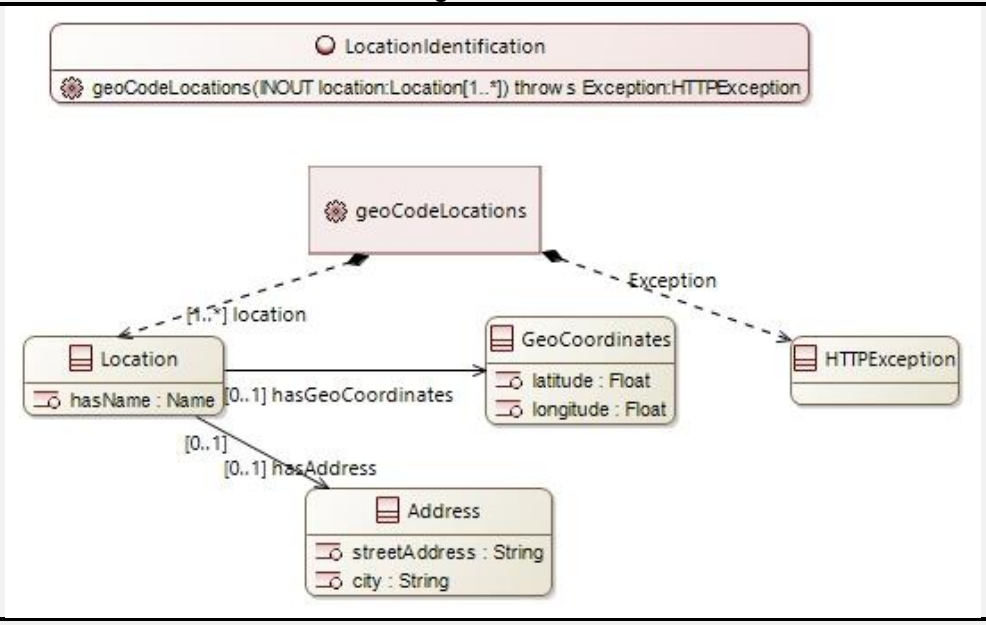
Interface ID:	The ID of this interface		
Interface Name:	The Name of this Interface		
Purpose of the Interface	The purpose of the interface		
Requestor:	Who uses the interface		
Provider	Who provides the interface		
Description:	Description of this interface		
Preconditions:	Which conditions must be fulfilled to use this interface		
Postconditions:	Which conditions are fulfilled after the use of this interface		
Request / Input	Name	M / O / C	Description / Example
	Name	M / O / C	Description / Example
	Name	M / O / C	Description / Example
Response / Output	Name	M / O / C	Description / Example
	Name	M / O / C	Description / Example
	Name	M / O / C	Description / Example
Exceptions:	Which exceptions could appear		
Notes and Issues:	Additional notes, issues and comments		

4.2.1 Interoperable Framework interfaces

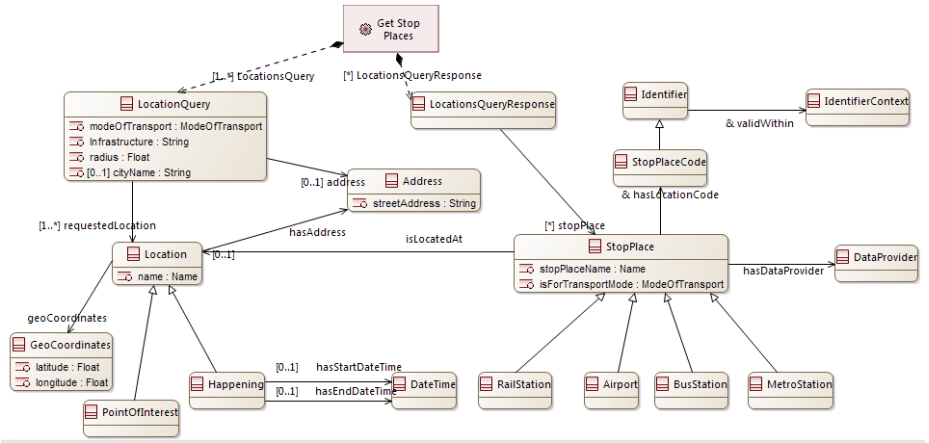
The list of interfaces provided by the Interoperability Framework:

Name	Purpose
Interface 1 – Location Identification	Provide list of Locations with geo coordinates
Interface 2 – Location Resolver	Provide list of Stop Place within a given radius of a Location
Interface 3 – Travel Expert Resolver	Associate Meta Travel Expert Episode with corresponding Travel Expert service descriptors
Interface 4 – Network Statistics	Provide Network data statistic as a list of RouteLinks with associated statistics information Provide list of Travel Experts registered in Semantic Web Service Registry;
Interface 5 – Navitia Decoder	Return Decode Confirmed Booking ItineraryOfferItem Stop Place and Transportation Vehicle data elements in the Navitia coding convention
Interface 6 – Travel Expert Broker	Provide a proxy to external, heterogeneous Travel Experts for shopping
Interface 7 – booking Engine Broker	Provide a proxy to external, heterogeneous Travel Experts for booking

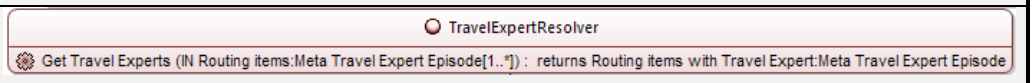
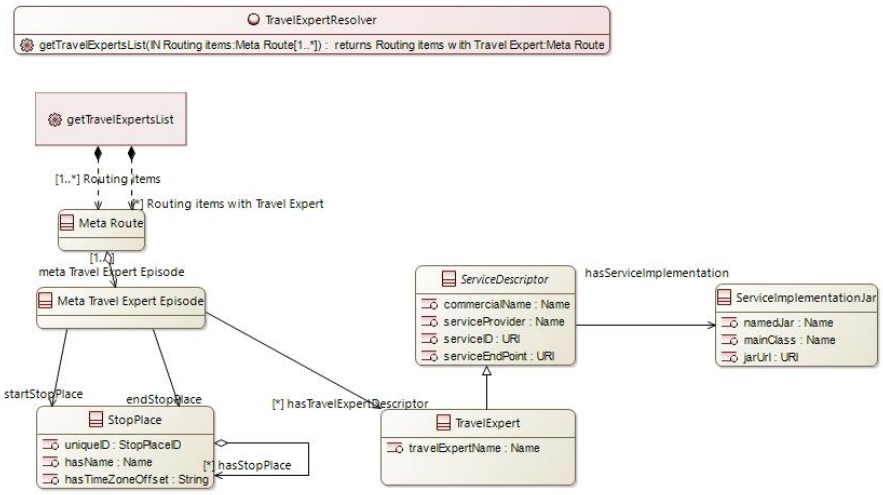
Interface 1 - Location Identification

Interface ID:	1
Interface Name:	
Purpose of the Interface	Provide list of Locations with geo coordinates
Requestor:	Travel Companion (Travel Companion component)
Provider	GeoCoding service
Description:	Provide list of Locations with geo coordinates
Preconditions:	Url encoded string for location name
Postconditions:	Zero, one or more locations with geo coordinates returned
Exchange Items	
Exceptions:	Http exception for invalid request or no results
Notes and Issues:	

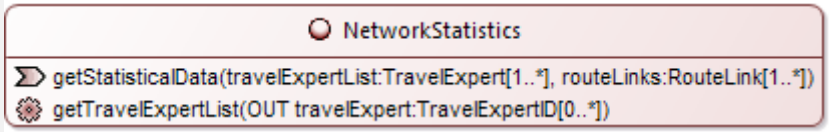
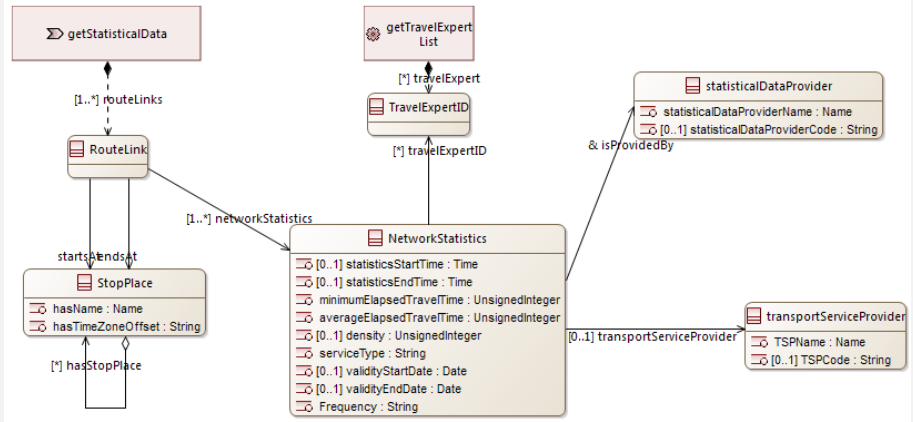
Interface 2 - Location Resolver

Interface ID:	2
Interface Name:	LocationsResolver Get Stop Places (IN LocationsQuery:LocationQuery[1..*]) : returns LocationsQueryResponse:LocationsQueryR
Purpose of the Interface	Provide list of Stop Place within a given radius of a Location
Requestor:	Travel Shopping (Travel Shopping component)
Provider	Location Resolver
Description:	Provide list of Stop Place within a given radius of a Location
Preconditions:	Valid LocationQuery
Postconditions:	Zero, one or more StopPlace returned
Exchange Items	
Exceptions:	Invalid request
Notes and Issues:	

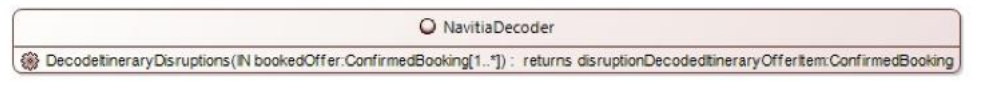
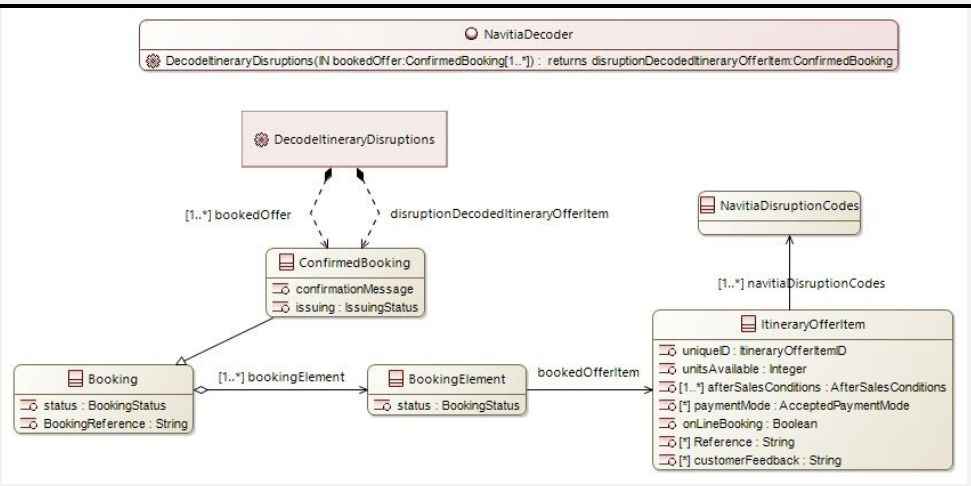
Interface 3 - Travel Expert Resolver

Interface ID:	3
Interface Name:	
Purpose of the Interface	Associate Meta Travel Expert Episode with corresponding Travel Expert service descriptors
Requestor:	Travel Shopping (Travel Shopping Components)
Provider	Travel Expert Resolver
Description:	Associate Meta Travel Expert Episode with corresponding Travel Expert service descriptors
Preconditions:	Valid Meta Travel Expert request
Postconditions:	Zero, one or more Travel Expert service descriptors are associated with input Meta Travel Expert Episode
Exchange Items	
Exceptions:	Invalid request
Notes and Issues:	Additional notes, issues and comments

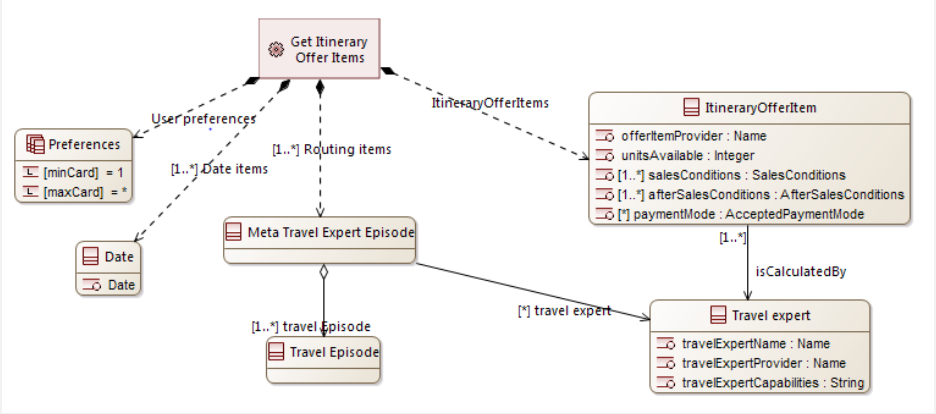
Interface 4 - Network Statistics

Interface ID:	4
Interface Name:	
Purpose of the Interface	Provide Network data statistic as a list RouteLinks with associated statistics information Provide list of Travel Experts registered in Semantic Web Service Registry
Requestor:	Travel Shopping (Travel Shopping component)
Provider	Network Graph Manager
Description:	getStatisticalData operation: Provide Network data statistic as a list RouteLinks with associated statistics information getTravelExpertList operation: Provide list of Travel Experts registered in Semantic Web Service Registry
Preconditions:	getStatisticalData operation: Travel Expert Id getTravelExpertList: none
Postconditions:	getStatisticalData operation: zero or more RouteLinks with associated statistics getTravelExpertList operation: zero or more TravelExpertID
Exchange Items	
Exceptions:	Invalid request
Notes and Issues:	

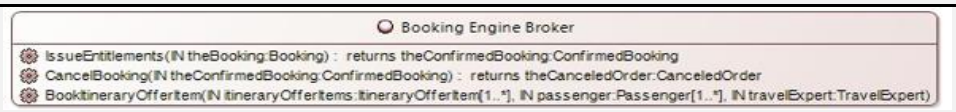
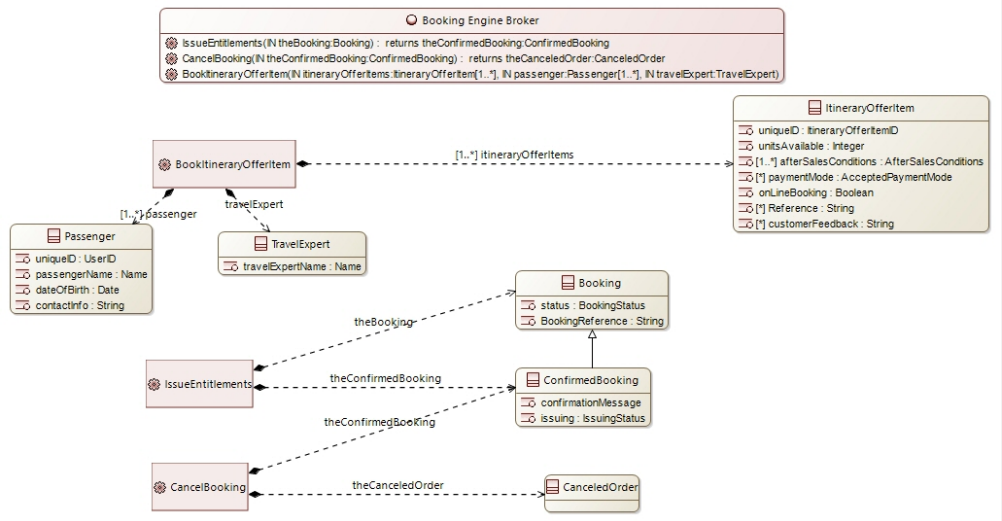
Interface 5 - Navitia Decoder

Interface ID:	5
Interface Name:	
Purpose of the Interface	Return Decode Confirmed Booking ItineraryOfferItem Stop Place and Transportation Vehicle data elements in the Navitia coding convention
Requestor:	Trip Trackler
Provider	Disruption Navitia Decoder
Description:	Provide codes in the Navitia convention for elements of the itineraries of a Confirmed Booking
Preconditions:	Valid Confirmed Booking
Postconditions:	Confirmed Booking itinerary associated with Navitia codes
Exchange Items	
Exceptions:	Invalid Confirmed Booking
Notes and Issues:	

Interface 6 - Travel Expert Broker

Interface ID:	6
Interface Name:	TravelExpert Broker Get Itinerary Offer Items(IN Routing items:Meta Travel Expert Episode[1..*], IN Date items:Date[1..*], IN User preferences:Preferences) : returns ItineraryOfferItems:ItineraryOfferItem
Purpose of the Interface	Provide a proxy to external, heterogeneous Travel Experts for shopping
Requestor:	Offer Builder (Travel Shopping Component)
Provider	Semantic Broker
Description:	Return ItinerayOfferItems for Meta Travel Expert Episodes from external, heterogeneous Travel Experts
Preconditions:	Travel Expert ontology and mappings loaded in Broker Ontology Manager
Postconditions:	Zero, one or more ItineraryOfferItem returned
Exchange Items	
Exceptions:	Missing mappings or schemas, unreachable or unavailable external Travel Experts
Notes and Issues:	

Interface 7 - Booking Engine Broker

Interface ID:	7
Interface Name:	
Purpose of the Interface	Provide a proxy to external, heterogeneous Travel Experts for booking
Requestor:	Booking Orchestrator (Booking and Ticketing Component)
Provider	Semantic Broker
Description:	Return Booking (inventory lock), Confirmed Booking with Entitlements or Booking Cancelation
Preconditions:	Travel Expert ontology and mappings loaded in Broker Ontology Manager
Postconditions:	Zero, one or more Booking, Confirmed Booking or Cancelled Booking returned
Exchange Items	
Exceptions:	Missing mappings or schemas, unreachable or unavailable external Travel Experts
Notes and Issues:	

4.2.2 Travel Shopping interfaces

This is the list of functional exchanges provided by the shopping:

Name	Purpose
Mobility Request Manager	
Interface 8 – Send Mobility Request	This functional exchange provides the mobility request
Shopping Orchestrator	
Interface 9 – Send Mobility Request with Traveller Preferences	This functional exchange provides the mobility request with the customer preferences
Interface 10 – Decode Mobility Request	This functional exchange retrieves the list of stop places
Meta Route Explorer	
Interface 11 – Send Mobility Request with Stop Places	This functional exchange provides the mobility request with the list of associated stop places
Offer Builder	
Interface 12 – Send Mobility Request with Traveller Preferences, Smartest Routes and Travel Experts List	This functional exchange provides the mobility request with the customer preferences, the list of associated stop places and the list of associated travel experts
Interface 13 – Get Offer Item List	This functional exchange is used to Get Offers from a specific Travel Expert, for the Meta Travel Expert Episodes relevant to this Travel Expert;

Mobility Request Manager

This Component provides one Interface to IT2Rail compliant systems;

Interface 8 - Send Mobility Request

Interface ID:	8
Interface Name:	Send Mobility Request
Purpose of the Interface	This Interface receives a mobility request from the Travel Companion
Requestor:	TC Shopping (PA)
Provider	TS Manage Mobility Request
Description:	This Interface receives a mobility request from a Traveller who is using the Travel Companion mobile app;
Preconditions:	A traveller wants to know more information regarding specific itineraries; A mobility request is issued by the Travel Companion mobile app;
Postconditions:	The Mobility Request Manager retrieves user preferences from Travel Companion Cloud then a mobility request enriched with Traveller Preferences is sent to the Travel Shopping Orchestrator;
Request / Input	Mobility Request (see Capella Model)
Response / Output	ItinerayOffer (see Capella Model)
Exceptions:	
Notes and Issues:	

Shopping Orchestrator

This function provides two Interfaces to IT2Rail compliant systems;

Interface 9 - Send Mobility Request with Traveller Preferences

Interface ID:	9
Interface Name:	Send Mobility Request with Traveller Preferences
Purpose of the Interface	This Interface sends a mobility request enriched with user preferences;
Requestor:	TS Manage Mobility Request
Provider	TS Orchestrate Shopping
Description:	This Interface sends a mobility request along with user preferences to the Shopping Orchestrator;
Preconditions:	The Mobility Request Manager collected user preferences from Travel Companion;
Postconditions:	A mobility request with traveller preferences is forwarded to the Travel Shopping Orchestrator in order to decode the mobility query;
Request / Input	Date, MetaJourney, PreferenceCollection (see Capella Model)
Response / Output	ItineryOffer (see Capella Model)
Exceptions:	
Notes and Issues:	

Interface 10 - Decode Mobility Request

Interface ID:	10
Interface Name:	Decode Mobility Request
Purpose of the Interface	Decode locations
Requestor:	TS Orchestrate Shopping
Provider	TS Decode Mobility Request
Description:	Provide list of Stop Place within a given radius of an identified Location
Preconditions:	Valid LocationQuery with Identified Location
Postconditions:	Zero, one or more StopPlace returned
Request / Input	**Location Query** (see Capella Model)
Response / Output	**LocationsQueryResponse** (see Capella Model)
Exceptions:	Invalid request
Notes and Issues:	Additional notes, issues and comments

Meta Route Explorer

This function provides one Interface to IT2Rail compliant systems;

Interface 11 - Send Mobility Request with Stop Places

Interface ID:	11
Interface Name:	Send Mobility Request with Stop Places
Purpose of the Interface	To provide a list of meta routes
Requestor:	TS Orchestrate Shopping
Provider	TS Select Smartest Routes
Description:	The service provides a list of meta routes to reach an end point from a start point for each meta journey requested;
Preconditions:	The travel shopping orchestrator provides a mobility query containing: • A list of Meta journeys; For each Meta journey, the user needs to precise the origin and destination points with the nearest Stop Places; • Possibly travellers' information and search options;
Postconditions:	The process provides the list of smartest routes covering the meta journey requested;
Request / Input	**Date, StopPlaceCode, MetaJourney, PreferenceCollection** (see Capella Model)
Response / Output	**Date, StopPlaceCode, MetaJourney, MetaRoutes** (see Capella Model)
Exceptions:	
Notes and Issues:	

Offer Builder

This function provides three Interfaces to IT2Rail compliant systems;

Interface 12 - Send Mobility Request with Traveller Preferences, Smartest Routes and Travel Experts List

Interface ID:	12
Interface Name:	Send Mobility Request with Traveller Preferences, Smartest Routes and Travel Experts List
Purpose of the Interface	To return a list of aggregated itinerary offers
Requestor:	TS Orchestrate Shopping
Provider	TS Build Offer
Description:	To return a list of aggregated itinerary offers with transport number, schedule and price
Preconditions:	The travel shopping orchestrator provides a request containing a list of smartest routes and the travel experts to call for each route;
Postconditions:	The process returns a list of aggregated itinerary offers;
Request / Input	**Meta Route, TravelExpert, Date, PreferenceCollection** (see Capella Model)
Response / Output	**ItineraryOffer** (see Capella Model)
Exceptions:	
Notes and Issues:	

Interface 13 - Provide Offer Detail For Offer Display

Interface ID:	13
Interface Name:	Provide Itinerary Offer Details
Purpose of the Interface	This Interface provides itinerary offer details
Requestor:	Ticketing & Booking: BT Orchestrate Booking
Provider	TS Build Offer
Description:	The Offer Builder temporary stores itinerary offer details; These details are used by other WP;
Preconditions:	An application requests details related to a given itinerary offer;
Postconditions:	Itinerary offer details are provided to the requestor
Request / Input	**ItineraryOfferRefID** (see Capella Model)
Response / Output	**ItineraryOffer** (see Capella Model)
Exceptions:	
Notes and Issues:	

4.2.3 Booking & Ticketing interfaces

The following table displays the list of operations available per interface; The “Name” column is the name of the interface followed by “:” and the name of the operation; This is the list of interfaces provided by the Booking & Ticketing;

Name	Purpose
Manage Entitlement Token	
Interface 14 ManageBookingEntitlementTokenI:putEntitlementTokenCW	This interface should be invoked by the Travel Companion to store the entitlement token in the CW
Interface 15 – ManageBookingEntitlementTokenI:updateTokenCW	This interface should be invoked by the Travel Companion to update the stored token in the CW
Interface 16 – ManageBookingEntitlementTokenI:putBookingCW	This interface should be invoked by the Travel Companion to store the booking in the CW
Manage Booking	
Interface 17 – ManageBookingI:BookItineraryOffer	This interface should be invoked to book an itinerary offer;
Interface 18 – ManageBookingI:LockInventory	This interface should be invoked to book a capacity constrained offer item
Interface 19 – PayBookingAndIssueEntitlementI:Pay&IssueItineraryOffer	This interface should be invoked to pay and issue an itinerary offer;
Interface 20 – PayBookingAndIssueEntitlementI:Generate Entitlements Tokens	This interface should be invoked to issue the Entitlements and Tokens of a GuaranteedPriceBooking;
Interface 21 – PayBookingAndIssueEntitlementI: Cancel Entitlements Tokens	This interface should be invoked to cancel an entitlement, and therefore all associated tokens;
Validate Token B&T	
Interface 22 – ManagePayloadI:GetPayload	The interface implements the interaction between the Travel Companion and the BT Validation Engine
Interface 23 – ManagePayloadI:UpdatePayload	The interface implements the interaction between the Travel Companion and the BT Validation Engine
Interface 24 – ConsumePayloadI:ValidateAndConsumeToken	The interface implements the interaction between the BT Validation Engine and the BT Consumption Engine;

Name	Purpose
Manage Entitlement Token	
Interface 25 – AlterTokenI:AlterToken	The interface implements the interaction between the BT Validation Engine and the BT Update Token;
Publish Ticketing Data	
Interface 26 – TicketingPublishingI:PublishTopologies	This interface provides the current ticketing logical and physical topologies;
Interface 27 – TicketingPublishingI:PublishFareProductsAndRulesDescriptions	This interface should be invoked to provide the current description of fare products and fare rules;
Interface 28 – TicketingPublishingI:PublishFareProductsAndRules	This interface should be invoked to provide the current list of fare products and fare rules;
Manage Offer Item	
Interface 29 – ManageOfferItemI:GetOfferItemList	This interface should be invoked to retrieve the list of available offer items;
Interface 30 – ManageOfferItemI:UpdateOfferItem	This interface should be invoked to update an OfferItem with an altered CustomizationParameters;
Interface 31 – ManageOfferItemI:GetPrice	This interface should be invoked to price an available itinerary offer item;
Manage Payment	
Interface 32 – ManagePaymentI:BeginPaymentTransaction	This interface should be invoked to start the two-step payment process;
Interface 33 – ManagePaymentI:VerifyAuthZCode	This interface updates the PaymentReferenceList with information from a trusted PaymentEngine;
Interface 34 – ManagePaymentI:AuthorizePayment	This interface should be invoked to request the authorization of the payment in the two-step payment process;
Interface 35 – ManagePaymentI:SettlePayment	This interface is to settle the payment;

Manage Entitlement Token

This Component provides three interfaces to IT2Rail compliant systems;

Interface 14 - ManageBookingEntitlementTokenI:putEntitlementTokenCW

Interface ID:	14		
Interface Name:	putEntitlementTokenCW		
Purpose of the Interface	This interface should be invoked by the Travel Companion to store the entitlement token in the CW		
Requestor:	Travel Companion		
Provider	Booking & Ticketing		
Description:	This interface should be invoked by the Travel Companion to store the entitlement token in the CW		
Preconditions:	None		
Postconditions:	None		
Request / Input	GuaranteedPriceBooking	M	A booking where all booked ItineraryOfferItem's prices are guaranteed
	Entitlement	M	The contract stipulated between the Offer Provider and the Customer for the associated ItineraryOfferItem
	Token List	M	The list (array) of token (ID);
	UserIDToken	M	The UserIDToken is the pointer identifying a unique token associated to a customer
Response / Output	ReturnStatus		OK/KO/Access Denied/Technical Error
Exceptions:	None		
Notes and Issues:	None		

Interface 15 - ManageBookingEntitlementTokenI:updateTokenCW

Interface ID:	15		
Interface Name:	updateTokenCW		
Purpose of the Interface	This interface should be invoked by the Travel Companion to update the stored token in the CW		
Requestor:	Travel Companion		
Provider	Booking & Ticketing		
Description:	This interface should be invoked by the Travel Companion to update the stored token in the CW		
Preconditions:	None		
Postconditions:	None		
Request / Input	Token	M	The expression of an entitlement in an involved organization used to perform the trip
	ValidationContext	M	The context of the validation
	TokenID	M	The TokenID is the pointer identifying a unique token
	EventDateTime	M	The time stamp of the occurred event
	UserIDToken	M	The UserIDToken is the pointer identifying a unique token associated to a customer
Response / Output	ReturnStatus		OK/KO/Access Denied/Technical Error
Exceptions:	None		
Notes and Issues:	None		

Interface 16 - ManageBookingEntitlementTokenI:putBookingCW

Interface ID:	16		
Interface Name:	putBookingCW		
Purpose of the Interface	This interface should be invoked by the Travel Companion to store the booking in the CW		
Requestor:	Travel Companion		
Provider	Booking & Ticketing		
Description:	This interface should be invoked by the Travel Companion to store the booking in the CW		
Preconditions:	None		
Postconditions:	None		
Request / Input	BookingID	M	The BookingID is the pointer identifying a unique booking
	UserIDToken	M	The UserIDToken is the pointer identifying a unique token associated to a customer
Response / Output			
Exceptions:	None		
Notes and Issues:	None		

Manage Booking

This Component provides five interfaces to IT2Rail compliant systems;

Interface 17 - ManageBookingI:BookItineraryOffer

Interface ID:	17		
Interface Name:	BookItineraryOffer		
Purpose of the Interface	This interface should be invoked to book an itinerary offer;		
Requestor:	Travel Companion		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to book an itinerary offer;		
Preconditions:	An itinerary offer has been chosen by the passenger (traveller);		
Postconditions:	None		
Request / Input	ItineraryOfferRefID	M	URL pointing to the Valued Itinerary Offer that the customer chose to book and pay for; The offer to be booked;
	Passenger List	M	The list (array) of passenger (ID);
	DeviceTokenID	M	The DeviceTokenID is the pointer identifying a unique device
Response / Output	BookingID	M	The BookingID is the pointer identifying a unique booking
Exceptions:	None		
Notes and Issues:	None		

Interface 18 - ManageBookingI:LockInventory

Interface ID:	18		
Interface Name:	LockInventory		
Purpose of the Interface	This interface should be invoked to book a capacity constrained offer item		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to book a capacity constrained offer item;		
Preconditions:	None		
Postconditions:	None		
Request / Input	ItineraryOfferItem List	M	The list (array) of itinerary offer item
	Passenger List	M	The list (array) of passenger (ID)
	DeviceTokenID	M	The DeviceTokenID is the pointer identifying a unique device
	BookingEngine	M	An URL to the booking engine
Response / Output	BookingElements	M	BookingElement linked to the initial OfferItem
Exceptions:	None		
Notes and Issues:	The output of this interface should not be cached		

Interface 19 - PayBookingAndIssueEntitlementI:Pay&IssueItineraryOffer

Interface ID:	19		
Interface Name:	Pay&IssueItineraryOffer		
Purpose of the Interface	This interface should be invoked to pay and issue an itinerary offer;		
Requestor:	Travel Companion		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to pay and issue an itinerary offer;		
Preconditions:	None		
Postconditions:	None		
Request / Input	FormOfPayment	M	The type of the payment process
	BookingID	M	The BookingID is the pointer identifying a unique booking
	UserIDToken	M	The UserIDToken is the pointer identifying a unique token associated to a customer
Response / Output	Boolean	M	OK/KO
Exceptions:	None		
Notes and Issues:	None		

Interface 20 - PayBookingAndIssueEntitlementI:Generate Entitlements Tokens

Interface ID:	20		
Interface Name:	Generate Entitlements Tokens		
Purpose of the Interface	This interface should be invoked to issue the Entitlements and Tokens of a GuaranteedPriceBooking;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to issue the Entitlements of a GuaranteedPriceBooking; The Entitlement should be issued only if the PaymentReferenceList complies with the business rules (there should be no blocking payment); The issuing of the entitlement is contractually binding the customer with his/her providers;		
Preconditions:	None		
Postconditions:	None		
Request / Input	Passenger List	M	The list (array) of passenger (ID)
	ItineraryOfferItem List	M	The list (array) of itinerary offer item
	BookingElements	M	ID of the BookingElement being paid;
	ShoppingRequestContext	M	Information specific to the current shopping request
Response / Output	Entitlement List	M	The list (array) of entitlement; The contract stipulated between the Offer Provider and the Customer for the associated ItineraryOfferItem
Exceptions:	None		
Notes and Issues:	This interface has been merged with the Generate Entitlement as the issuing of the entitlement may occurs before the Token issuing;		

Interface 21 - PayBookingAndIssueEntitlementI: Cancel Entitlements Tokens

Interface ID:	21		
Interface Name:	Cancel Entitlements Tokens		
Purpose of the Interface	This interface should be invoked to cancel an entitlement, and therefore all associated tokens;		
Requestor:	Travel Companion		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to cancel an entitlement, and therefore all associated tokens;		
Preconditions:	None		
Postconditions:	None		
Request / Input	EntitlementID List	M	The list (array) of entitlement (ID); The EntitlementID is the pointer identifying a unique Entitlement;
Response / Output			
Exceptions:	None		
Notes and Issues:	None		

Validate Token B&T

This Component provides four interfaces to IT2Rail compliant systems;

Interface 22 - ManagePayloadI:GetPayload

Interface ID:	22		
Interface Name:	GetPayload		
Purpose of the Interface	The interface implements the interaction between the Travel Companion and the BT Validation Engine		
Requestor:	TC Tapping Module (PA) (Travel Companion Component)		
Provider	BT Validation Engine		
Description:	Retrieve Token and its Payload from Travel Companion		
Preconditions:	Validate Token (Preparation) is complete, tapping module identified and loaded, Traveller presents appropriate embodiment		
Postconditions:	Token's payload is retrieved		
Request / Input			
Response / Output	Payload	M	Is a collection of bytes meaning that a token is able to be used for validation;
Exceptions:	None		
Notes and Issues:	None		

Interface 23 - ManagePayloadI:UpdatePayload

Interface ID:	23		
Interface Name:	UpdatePayload		
Purpose of the Interface	The interface implements the interaction between the Travel Companion and the BT Validation Engine		
Requestor:	TC Tapping Module (PA) (Travel Companion Component)		
Provider	BT Validation Engine		
Description:	Retrieve Token and its Payload from Travel Companion		
Preconditions:	Validate Token (Preparation) is complete, tapping module identified and loaded, Traveller presents appropriate embodiment		
Postconditions:	Token's payload is retrieved		
Request / Input	Payload	M	Is a collection of bytes meaning that a token is able to be used for validation;
Response / Output			
Exceptions:	None		
Notes and Issues:	None		

Interface 24 - ConsumePayloadI:ValidateAndConsumeToken

Interface ID:	24		
Interface Name:	ValidateAndConsumeToken		
Purpose of the Interface	The interface implements the interaction between the BT Validation Engine and the BT Consumption Engine		
Requestor:	BT Validation Engine		
Provider	BT Consumption Engine		
Description:	Submit token's payload and validation context to validation		
Preconditions:	Token payload retrieved and Validation Context created		
Postconditions:	Validation Status and updated Payload are returned;		
Request / Input	ValidationContext	M	Complementary information needed to analyse the Payload
	Payload	M	Is a collection of bytes meaning that a token is able to be used for validation;
Response / Output	ValidationStatus	M	The status of the validation of the token
	Payload	M	Is a collection of bytes meaning that a token is able to be used for validation;
Exceptions:	None		
Notes and Issues:	None		

Interface 25 - AlterTokenI:AlterToken

Interface ID:	25		
Interface Name:	AlterToken		
Purpose of the Interface	The interface implements the interaction between the BT Validation Engine and the BT Update Token		
Requestor:	BT Validation Engine		
Provider	BT Update Token		
Description:	Alter token with updated payload		
Preconditions:	ValidationStatus indicates passed validation		
Postconditions:	Token successfully retrieved and altered with updated payload;		
Request / Input	ValidationTransaction	M	The ValidationTransaction knows if the token is effective
	TokenID	M	The TokenID is the pointer identifying a unique token
Response / Output	ReturnStatus		OK/KO/Access Denied/Technical Error
Exceptions:	None		
Notes and Issues:	None		

Publish Ticketing Data

This Component provides three interfaces to IT2Rail compliant systems;

Interface 26 - TicketingPublishingI:PublishTopologies

Interface ID:	26		
Interface Name:	PublishTopologies		
Purpose of the Interface	This interface provides the current ticketing logical and physical topologies;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to provide the current ticketing logical and physical topologies after a change of topology elements in ticketing systems;		
Preconditions:	A Topology must have been versioned;		
Postconditions:	None		
Request / Input	Logical topology	M	Format dependant logical topology;
	Physical topology	M	Format dependant physical topology;
Exceptions:	None		
Notes and Issues:	None		

Interface 27 - TicketingPublishingI:PublishFareProductsAndRulesDescriptions

Interface ID:	27		
Interface Name:	PublishFareProductsAndRulesDescriptions		
Purpose of the Interface	This interface should be invoked to provide the current description of fare products and fare rules;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to provide the current description of fare products and fare rules, after a change in fare product or rules in ticketing systems;		
Preconditions:	Fare sets must have been versioned and validated for publishing;		
Postconditions:	None		
Request / Input	Fare Rules Description	M	Description of the current fare rules
	FareProductDescription	M	Description of the current fare products
Exceptions:	None		
Notes and Issues:	None		

Interface 28 - TicketingPublishingI:PublishFareProductsAndRules

Interface ID:	28		
Interface Name:	PublishFareProductsAndRules		
Purpose of the Interface	This interface should be invoked to provide the current list of fare products and fare rules;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to provide the current list of fare products and fare, after a change in fare product or rules in ticketing systems;		
Preconditions:	Fare sets must have been versioned and validated for publishing;		
Postconditions:	None		
Request / Input	Fare Rules	M	Fare rules in the internal ticketing system format
	FareProduct	M	Fare products rules in the internal ticketing system format
Exceptions:	None		
Notes and Issues:	None		

Manage Offer Item

This Component provides three interfaces to IT2Rail compliant systems;

Interface 29 - ManageOfferItemI:GetOfferItemList

Interface ID:	29		
Interface Name:	GetOfferItemList		
Purpose of the Interface	This interface should be invoked to retrieve the list of available offer items;		
Requestor:	Shopping		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to retrieve the list of available ItineraryOfferItems that may be used to build an itinerary offer for the customer to provide transportation solution for the passengers on the considered itinerary;		
Preconditions:	None		
Postconditions:	None		
Request / Input	Itinerary	M	The itinerary considered
	Customer	M	The customer for which the offer is constructed;
	UserID List	O	The list (array) of user (ID);
	Retailer	O	The retailer that will sell the Fare product;
Response / Output	ItineraryOfferItem	M	The initial ItineraryOfferItem;
Exceptions:	None		
Notes and Issues:	The output of this interface may be cached for further use until new operational parameters are published;		

Interface 30 - ManageOfferItemI:UpdateOfferItem

Interface ID:	30		
Interface Name:	UpdateOfferItem		
Purpose of the Interface	This interface should be invoked to update an OfferItem with an altered CustomizationParameters;		
Requestor:	Shopping		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to update an ItineraryOfferItem with an altered CustomizationParameters; No other alteration of an ItineraryOfferItem should be allowed;		
Preconditions:	None		
Postconditions:	None		
Request / Input	Itinerary	M	The itinerary considered;
	Customer	M	The customer for which the offer is constructed;
	UserID	M	The UserID is the pointer identifying a unique customer
	Retailer	M	The retailer that will sell the Fare product;
	ItineraryOfferItem	M	The itinerary offer item to update;
Response / Output	ItineraryOfferItem	M	The updated ItineraryOfferItem;
Exceptions:	None		
Notes and Issues:	The output of this interface should not be cached;		

Interface 31 - ManageOfferItemI:GetPrice

Interface ID:	31		
Interface Name:	GetPrice		
Purpose of the Interface	This interface should be invoked to price an available itinerary offer item;		
Requestor:	Shopping		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to price an available itinerary offer item		
Preconditions:	None		
Postconditions:	None		
Request / Input	ItineraryOfferItem	M	The itinerary offer item to be priced
Response / Output	GlobalQuotation	M	The price of the itinerary order item that may be proposed to the customer for the passenger on this itinerary;
Exceptions:	None		
Notes and Issues:	The output of this interface should not be cached;		

Manage Payment

This Component provides four interface to IT2Rail compliant systems;

Interface 32 - ManagePaymentI:BeginPaymentTransaction

Interface ID:	32		
Interface Name:	BeginPaymentTransaction		
Purpose of the Interface	This interface should be invoked to start the two-step payment process;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to start the two-step payment process; The provider of the interface must check the availability if needed of the Payment Module;		
Preconditions:	None		
Postconditions:	None		
Request / Input	FormsOfPayment	M	Payment means considered
Response / Output	TransactionID	M	ID of the transaction, created from the BeginPaymentTransaction interface;
Exceptions:	None		
Notes and Issues:	None		

Interface 33 - ManagePaymentI:VerifyAuthZCode

Interface ID:	33		
Interface Name:	VerifyAuthZCode		
Purpose of the Interface	This interface updates the PaymentReferenceList with information from a trusted PaymentEngine;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface updates the PaymentReferenceList with information from a trusted PaymentEngine; It should be used to verify that the authorization code provided to issue the entitlement or token is legitimate and still valid;		
Preconditions:	None		
Postconditions:	None		
Request / Input	FormsOfPayment	M	Payment means considered
	PaymentReferenceList	O	Existing payment reference list
Response / Output	PaymentReferenceList	M	Updated payment reference list
Exceptions:	None		
Notes and Issues:	None		

Interface 34 - ManagePaymentI:AuthorizePayment

Interface ID:	34		
Interface Name:	AuthorizePayment		
Purpose of the Interface	This interface should be invoked to request the authorization of the payment in the two-step payment process;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to request the authorization of the payment in the two-step payment process; The client of the interface must use the Payment Progress element of the PaymentReferenceList to monitor the progress of the payment process;		
Preconditions:	None		
Postconditions:	None		
Request / Input	FormsOfPayment	M	Payment means considered
	TransactionID	M	ID of the transaction, created from the BeginPaymentTransaction interface;
	BookingElementID	M	ID of the BookingElement being paid;
Response / Output			
Exceptions:	None		
Notes and Issues:	None		

Interface 35 - ManagePaymentI:SettlePayment

Interface ID:	35		
Interface Name:	SettlePayment		
Purpose of the Interface	This interface is to settle the payment;		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface settles the payment (transferring the amount charged from the customer account to the merchant account) and updates the PaymentReferenceList to reflect the status of the settlement;		
Preconditions:	None		
Postconditions:	None		
Request / Input	FormsOfPayment	M	Payment means considered
	PaymentReferenceList	M	Payment reference list
Response / Output	PaymentReferenceList	M	Updated PaymentReference List
Exceptions:	None		
Notes and Issues:	None		

4.2.4 Trip Tracker interfaces

The list of interfaces

Name	Purpose
Interface 36 – BAdataAlternatives	Exchange of logged alternatives data between the Trip Tracker and the Business Analytics;
Interface 37 - BAdataCEP	Exchange of logged CEP data between the Trip Tracker and the Business Analytics;
Interface 38 - CallAlternatives	Triggering journey recalculation
Interface 39 - DecodeItineraryDisruptions	Decoding of the event structure provided by the Navitia platform
Interface 40 - GetAlternatives	The request for alternatives calculation
Interface 41 - RequestJourneyTracking	Triggering request for activation of given journey tracking

Interface 36 - BAdataAlternatives

Interface name	**BAdataAlternatives**		
Description	Exchange of logged alternatives data between the Trip Tracker and the Business Analytics;		
Requestor	BA Data Collection		
Provider	TT OrchestrateBA		
Release	**CREL**	**AREL**	**FREL**
			Complete
Pre-conditions	Logging of alternatives data		
Post-conditions	Further processing in the Business Analytics		
Request / Input			
Response / Output	AverageStops		
	MaximumPrice		
	MinimumPrice		
	NumberAlternatives		
	ProcessingTime		
	TransportationMode		

	TravelMaximumTime
	TravelMinimumTime
Exceptions	
Notes	

Interface 37 - BAdataCEP

Interface name	**BAdataCEP**		
Description	Exchange of logged CEP data between the Trip Tracker and the Business Analytics;		
Requestor	BA Data Collection		
Provider	TT OrchestrateBA		
Release	**CREL**	**AREL**	**FREL**
			Complete
Pre-conditions	Logging of CEP data		
Post-conditions	Further processing in the Business Analytics		
Request / Input	Date		
Response / Output	ArrivalDelayEvent		
	ArrivalDelayEventRules		
	DepartureDelayEvent		
	DepartureDelayEventRules		
	RulesDeactivationRequest		
Exceptions			
Notes			

Interface 38 - CallAlternatives

Interface name	**CallAlternatives**		
Description	Triggering journey recalculation		
Requestor	TC AlertManagement (PA)		
Provider	TT GetAlternatives		
Release	**CREL**	**AREL**	**FREL**
			Complete
Pre-conditions	Request alternatives		
Post-conditions	Get booked offer		
Request / Input	UserID		
	UserIDToken		
	SendMessage		
	GeoCoordinates		
	BookedOfferID		
Response / Output	ItineraryOffer		
Exceptions			
Notes			

Interface 39 - DecodeItineraryDisruptions

Interface name	**DecodeItineraryDisruptions**		
Description	Decoding of the event structure provided by the Navitia platform		
Requestor	TT EventsListening		
Provider	IF Decode Navitia Disruptions		
Release	**CREL**	**AREL**	**FREL**
			Complete
Pre-conditions	Identify available sources of events		
Post-conditions	Invoke listening of Navitia event´s source		
Request / Input	ConfirmedBooking		
Response / Output	ConfirmedBooking		
Exceptions			
Notes			

Interface 40 - GetAlternatives

Interface name	**GetAlternatives**		
Description	The request for alternatives calculation		
Requestor	TT GetAlternatives		
Provider	TS Compute Itinerary Offer Items		
Release	**CREL**	**AREL**	**FREL**
			Complete
Pre-conditions	Get booked offer		
Post-conditions	Check alternatives		
Request / Input	RoutesQuery		
Response / Output	ItineraryOffer		
Exceptions			
Notes			

Interface 41 - RequestJourneyTracking

Interface name	**RequestJourneyTracking**		
Description	Triggering request for activation of given journey tracking		
Requestor	TC AlertManagement (PA)		
Provider	TT OrchestrateActivation		
Release	**CREL**	**AREL**	**FREL**
	Partial	Partial	Complete
Pre-conditions	Tracking requested by the Traveller		
Post-conditions	Request preferences retrieval		
Request / Input	ActivationSelector		
	UserID		
	UserIDToken		
	BookedOfferD		
Response / Output	Response		
Exceptions			
Notes			

4.2.5 Travel Companion interfaces

The list of interfaces from the Travel Companion;

Name	Purpose
Interface 42 - Manage Preferences	To provide and group operations concerning the management of user preferences (retrieve and update) by other IT2Rail modules
Interface 43 – Manage Booking Entitlement Token	To provide and group operations concerning the handling (storage, retrieval, and update) of booked offers, entitlements and tokens stored on the TC CW;
Interface 44 - TC2TTrack	To provide and group operations concerning the handling of disruptions (including operations needed for the manging of alternatives in case of disruptions);
Interface 45 – Manage Payment Data	The user token was generated by TC and corresponds to the provided user ID; the creditCardType corresponds to one of the accepted by the system;
Interface 46 – Personal App GUI	This interface is actually a modelling device to capture the operations that the user can perform on the TC PA; As such, the items in this interface are not intended to be implemented methods (though they ultimately correspond to some code), but, rather, actions performed by the user on the PA GUI;

TC CloudWallet

The interfaces that the TC CW exports towards other IT2Rail modules (TS, BT, TT) are ExportPreferencesI, TC2TtrackI, and ManageBookingEntitlementTokenI; The tables presented in the rest of this section provide some further details concerning these interfaces, and in particular of the operations they export;

Interface 42 - Manage Preferences

Interface ID:	42		
Interface Name:	ManagePreferencesI		
Purpose of the Interface	To provide and group operations concerning the managemente of user preferences (retrieve and update) by other IT2Rail modules		
Requestors:	TS, TT		
ProvideAllPreferencesCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID;		
Postconditions:	The preferences returned are all those of the user, with their associated score;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
Response / Output	preferences	M	List of returned preferences
Exceptions:			
Notes and Issues:			
UpdatePreferencesCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; The preferences which its values were not sending, will be update to the minimum possible value;		
Postconditions:	The new preference value was being stored in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token

	preferencesList	M	The new preferences that will replace the existing ones;
Response / Output	res	M	List of returned preferences
Exceptions:			
Notes and Issues:			

Interface 43 - Manage Booking Entitlement Token

Interface ID:	43		
Interface Name:	ManageBookingEntitlementTokenI		
Purpose of the Interface	To provide and group operations concerning the handling (storage, retrieval, and update) of booked offers, entitlements and tokens stored on the TC CW;		
Requestors:	BT		
putBookingCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID;		
Postconditions:	The received booking ID is correctly stored in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	booking	M	ID of the booking to be stored
Exceptions:			
Notes and Issues:			
getBookingCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the ID received corresponds to a booking stored in the TC CW		
Postconditions:	The booking returned is the one corresponding to the ID received as input		

Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	booking	M	ID of the booking to be retrieved
Response / Output	res	M	Booking returned
Exceptions:			
Notes and Issues:			
putEntitlementTokenCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID;		
Postconditions:	The received entitlement and tokens are correctly stored in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	booking	M	Booking to which the entitlement and tokens correspond
	entitlement	M	Entitlement to be stored
	tokenList	M	List of tokens to be stored
Response / Output	res	M	Outcome of the storage, whether successful or not
Exceptions:			
Notes and Issues:			
updateTokenCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the token to be updated was correctly stored in the TC CW		
Postconditions:	The token is updated in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	token	M	Token to be updated
	tokid	M	ID of the token to be updated

	valContext	M	Context in which the validation leading to the update of the token is performed
	currentValues	M	Timestamp of the validation
Response / Output	res	M	Outcome of the update, whether successful or not
Exceptions:			
Notes and Issues:			

Interface 44 - TC2TTrack

Interface ID:	44		
Interface Name:	TC2TTrackI		
Purpose of the Interface	To provide and group operations concerning the handling of disruptions (including operations needed for the manging of alternatives in case of disruptions);		
Requestors:	TT		
PushMessageCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the ID of the booked offer corresponds to an offer stored on the TC CW;		
Postconditions:	The received message is correctly stored in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	message	M	Message with the alert for the user
	bookedOfferID	M	ID of the booked offer to which the message is related
Exceptions:			
Notes and Issues:			
getBookedOfferCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the ID received corresponds to a booked offer stored in the TC CW		

Postconditions:	The booked offer returned is the one corresponding to the ID received as input		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	offerID	M	ID of the booked offer to be retrieved
Response / Output	res	M	Booked offer returned
Exceptions:			
Notes and Issues:			

Interface 45 - Manage Payment Data

Interface ID:	45		
Interface Name:	ManagePaymentDataI		
Purpose of the Interface	To provide and group operations concerning the handling of the payment data (including operations such as create, update, remove and retrieve);		
Requestors:	TC PA		
AddUserCreditCardCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the creditCardType corresponds to one of the accepted by the system;		
Postconditions:	The credit card information was stored in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	creditCardType	M	The type of the credit card
	creditCardData	M	All the data related to the Credit card such as number, CVV, expiration date and display name

Exceptions:			
Notes and Issues:			
UpdateUserCreditCardCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the creditCardType corresponds to one of the accepted by the system and the creditCardData must be related to one that was stored in the TC CW;		
Postconditions:	The credit card information was updated in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	creditCardId	M	ID of the credit card
	creditCardType	M	The type of the credit card
	creditCardData	M	All the data related to the Credit card such as number, CVV, expiration date and display name
Exceptions:			
Notes and Issues:			
removeUserCreditCardCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID; the creditCardData must be related to one that was stored in the TC CW;		
Postconditions:	The credit card information was deleted in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
	creditCardId	M	ID of the credit card
	creditCardType	M	The type of the credit card
	creditCardData	M	All the data related to the Credit card such as number, CVV, expiration date and display name
Exceptions:			

Notes and Issues:			
removeUserCreditCardCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID;		
Postconditions:	The credit card information related to this user was deleted in the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
Exceptions:			
Notes and Issues:			
getAllUserCreditCardCW			
Preconditions:	The user token was generated by TC and corresponds to the provided user ID;		
Postconditions:	All credit card information related to this user was retrieve from the TC CW;		
Request / Input	userID	M	ID of the user
	userIDToken	M	Authorization token
Exceptions:			
Notes and Issues:			
getAllCardTypesCW			
Preconditions:			
Postconditions:	The credit card types which are accepted in the system;		
Request / Input			
Exceptions:			
Notes and Issues:			

TC PersonalApplication

As mentioned above, the interfaces that the TC PA exports towards other IT2Rail modules are only two: *PersonalAppGUI*, and *ManagePayloadI* (*AlertMagmtPA_IntI* is an internal interface used by TC CW, and not by an external service or actor); The tables presented in the rest of this section provide some further details concerning these interfaces, and in particular of the operations they export;

Interface 46 - PersonalAppGUI

Interface ID:	46		
Interface Name:	PersonalAppGUI		
Purpose of the Interface	This interface is actually a modelling device to capture the operations that the user can perform on the TC PA; As such, the items in this interface are not intended to be implemented methods (though they ultimately correspond to some code), but, rather, actions performed by the user on the PA GUI;		
Requestors:	TC PA user (traveller)		
Preconditions:	All actions grouped in this interface require that the user is logged in the application, aside from *createUserGUI* and *loginUserGUI*;		
Postconditions:	The result of each action is the start of the corresponding interaction as described in Deliverable D5;2;		
Interface Name:	ManagePayloadI		
Purpose of the Interface	This interface collects the low-level operations, *GetPayload* and *UpdatePayload*, which are used to retrieve and update the payload on the embodiment of a token during the token validation process, as depicted in Deliverable D5;2;		
Requestors:	BT		
GetPayload			
Preconditions:	The embodiment has been presented to the BT validation module;		
Postconditions:	The payload of the token is returned;		
Request / Input			
Exceptions:			
Notes and Issues:			
UpdatePayload			

Preconditions:	The embodiment can be updated		
Postconditions:	The embodiment is updated		
Request / Input	updatedPayload	M	The new payload for the token
Exceptions:			
Notes and Issues:			

4.2.6 Business Analytics interfaces

Interfaces for data collection

Name	Purpose
Interface 47 – Get_Master_DataI	This interface retrieves master data regarding transport operators from the Interoperability Framework;
Interface 48 – Get_Services_LocationI	This interface retrieves information concerning the services location;
Interface 49 – Get_Travel_DataI	This interface retrieves information for analysing travel data;
Interface 50 – BA_Cooperation_CEPI	This interface provides information on events' evaluation from Complex Event Processing engine;
Interface 51 –BA_cooperation_messagesI	This interface provides information on all messages that were sent to the Travel Companion to be displayed to a customer, with a set of related information
Interface 52 – BA_cooperation_alternativesI	In case of travel plan reallocation, this interface retrieves information for at least any indication, whether some alternatives were offered to the customer or not;
Interface 53 – Get_Travel_InformationI	This interface retrieves travel information from external data providers
Interface 54 – Get_Social_InformationI	This interface retrieves messages from social network platforms

Interface 47 - Get_Master_DataI

Interface ID:	47		
Interface Name:	Get_Master_DataI		
Purpose of the Interface	This interface retrieves master data regarding transport operators from the Interoperability Framework;		
Requestor:	Business Analytics		
Provider	Interoperability Framework		
Description:	This interface retrieves master data regarding transport operators from the Interoperability Framework;		
Preconditions:	No preconditions		
Postconditions:	The Business Analytics has collected master data for analytics purpose		
Request / Input	Start StopPlace global ID	M	Id of the first Stop Place of a MetaTravelExpertEpisode
	End StopPlace global ID	M	Id of the last Stop Place of a MetaTravelExpertEpisode
Response / Output	TravelExpert descriptor	M	Descriptor that supplies offer items on the metaroute
Exceptions:	No exceptions		
Notes and Issues:	No specified		

Interface 48 - Get_Services_LocationI

Interface ID:	48		
Interface Name:	Get_Services_LocationI		
Purpose of the Interface	This interface retrieves information concerning the services location;		
Requestor:	Business Analytics		
Provider	Interoperability Framework		
Description:	This interface retrieves information concerning the services location;		
Preconditions:	No preconditions		
Postconditions:	The Business Analytics can invoke the interfaces exposed by the other IT2Rail work packages;		
Request / Input	serviceName	1	The service name to invoke
Response / Output	serviceLocation	1	Current service URI
Exceptions:	No exceptions		
Notes and Issues:	No specified		

Interface 49 - Get_Travel_DataI

Interface ID:	49		
Interface Name:	Get_Travel_DataI		
Purpose of the Interface	This interface retrieves information for analysing travel data;		
Requestor:	Business Analytics		
Provider	Booking & Ticketing		
Description:	This interface retrieves information for analysing travel data		
Preconditions:	No preconditions		
Postconditions:	The Business Analytics platform can correlate this information with other data related to the travel and transport domains; Finally a series of KPIs can be computed on the basis of user's requests;		
Request / Input	startPeriod endPeriod travelFilter	1 1 1	Initial date of the time window Final date of the time window Filter travels of interest (city, transport mode, etc;)
Response / Output	travelData	0;;n	List of data concerning travels taking place during the input time window
Exceptions:	No exceptions		
Notes and Issues:	No specified		

Interface 50 - BA_Cooperation_CEPI

Interface ID:	50		
Interface Name:	BA_Cooperation_CEPI		
Purpose of the Interface	This interface provides information on events' evaluation from Complex Event Processing engine; One event may have different impacts to different customers; For each couple event-customer, three different impact levels and appropriate message types are expected to be distinguished in WP4;		
Requestor:	Business Analytics		
Provider	Trip Tracking		
Description:	This interface provides information on events' evaluation from Complex Event Processing engine; One event may have different impacts to different customers; For each couple event-customer, three different impact levels and appropriate message types are expected to be distinguished in WP4;		
Preconditions:	No preconditions		
Postconditions:	Information on events is provided in an aggregated format and indicates the number of travellers affected by a specific event; The total number of events evaluated in the queried period shall be available;		
Request / Input	No input	-	-
Response / Output	Event ID	M	Unique event identifier inside IT2Rail domain;
	Impact level	M	This parameter specifies the impact level linked to the event;
	Number of customers	M	Customers that have been affected by the event with the appropriate impact;
Exceptions:	No exceptions;		
Notes and Issues:	No specified;		

Interface 51 - BA_cooperation_messagesI

Interface ID:	51		
Interface Name:	BA_cooperation_messagesI		
Purpose of the Interface	This interface provides information on all messages that were sent to the Travel Companion to be displayed to a customer, with a set of related information, including a link to the causal event (where appropriate);		
Requestor:	Business Analytics		
Provider	Trip Tracking		
Description:	This interface provides information on all messages that were sent to the Travel Companion to be displayed to a customer, with a set of related information, including a link to the causal event (where appropriate);		
Preconditions:	No preconditions		
Postconditions:	Information on events is provided in an aggregated format and indicate the number of travellers affected by a specific event; The total number of events evaluated in the queried period shall be available;		
Request / Input	No input	-	-
Response / Output	Message ID	M	Unique message identifier inside the Trip Tracker
	Event ID	M	Event, which was the cause of the message
	Travel ID	M	Traveller, who receives the message (unique identifier inside IT2Rail domain)
	Message Type	M	This field in fact indicates the severity of an event impacting on traveller's journey;
	Processing Time	M	The interval between capturing the event and sending the message
Exceptions:	No exceptions;		
Notes and Issues:	No specified;		

Interface 52 - BA_cooperation_alternativesI

Interface ID:	52		
Interface Name:	BA_cooperation_alternativesI		
Purpose of the Interface	In order to compute statistics related to proposed travel plan reallocation for each customer and all required aggregations (by customer, by transport/service provider, by time, and so on), the IT2Rail Business Analytics requires to get at least any indication, whether some alternatives were offered to the customer or not;		
Requestor:	Business Analytics		
Provider	Trip Tracking		
Description:	In order to compute statistics related to proposed travel plan reallocation for each customer and all required aggregations (by customer, by transport/service provider, by time, and so on), the IT2Rail Business Analytics requires to get at least any indication, whether some alternatives were offered to the customer or not		
Preconditions:	No preconditions		
Postconditions:	Business Analytics collects information on alternatives offered upon customer's request		
Request / Input	No input	-	-
Response / Output	Message ID	M	Message, which invoked the request for alternatives;
	Travel ID	M	Traveller requesting alternatives;
	Number of proposed alternatives:	M	This field specifies the amount of alternatives proposed to the traveller
	Processing Time	M	The interval between the request is received and alternatives are proposed to the traveller;
Exceptions:	No exceptions;		
Notes and Issues:	No specified;		

Interface 53 - Get_Travel_InformationI

Interface ID:	53		
Interface Name:	Get_Travel_InformationI		
Purpose of the Interface	This interface retrieves travel information from external data providers		
Requestor:	Business Analytics		
Provider	Data Provider		
Description:	This interface retrieves travel information from external data providers		
Preconditions:	No preconditions		
Postconditions:	The Business Analytics collects information on travel and transport domains from external data sources;		
Request / Input	startPeriod endPeriod travelFilter	1 1 1	Initial date of the time window Final date of the time window Filter travels of interest (city, transport mode, etc;)
Response / Output	travelInfo	0;;n	List of information concerning travels taking place during the input time window
Exceptions:	No exceptions;		
Notes and Issues:	No specified;		

Interface 54 - Get_Social_InformationI

Interface ID:	54		
Interface Name:	Get_Social_InformationI		
Purpose of the Interface	This interface retrieves messages from social network platforms		
Requestor:	Business Analytics		
Provider	Social network		
Description:	This interface retrieves messages from social network platforms; All crawled messages are related with some topics specified by experts;		
Preconditions:	Authorization key and security key in order to access to third party APIs; Specify the related topics used to crawl data from the social network;		
Postconditions:	The Business Analytics collects messages from social network platforms;		
Request / Input	startPeriod endPeriod topic	1 1 0;;1	Initial date of the time window Final date of the time window Filter messages of interest (i;e; keywords)
Response / Output	Res	0;;n	List of messages sent during the input time window
Exceptions:	No exceptions;		
Notes and Issues:	No specified;		

4.2.7 Interface provided by Business Analytics

Name	Purpose
Interface 55 - Provide_Traveller_QuestionnaireI	This interface collects answers of Traveller Questionnaires from the Travel Companion mobile app
Interface 56 - BusinessAnalyticsServiceI	This interface allows to provide information concerning specific KPIs, happenings and weather information, analysis of messages collected by social media;
Interface 57 - Get_Itinerary_OffersI	This interface receives itinerary offers computed by Travel Shopping;
Interface 58 - Provide itinerary offers for BA computations	The purpose of this interface is to provide itinerary offers to Business Analytics;

Interface 55 - Provide_Traveller_QuestionnaireI

Interface ID:	55		
Interface Name:	Provide_Traveller_QuestionnaireI		
Purpose of the Interface	This interface collects answers of Traveller Questionnaires from the Travel Companion mobile app		
Requestor:	Traveller		
Provider	Business Analytics		
Description:	This interface collects answers of Traveller Questionnaires from the Travel Companion mobile app		
Preconditions:	-		
Postconditions:	The Business Analytics collects answers of Traveller Questionnaire from Travel Companion mobile app		
Request / Input	travelDate travelLeg meanOfTransportation userID userCommnet travelQuestions • question rate	1 1 1 1 0;;1 1;;n 1 1	Date and time of travel leg Travel leg Transport mean used Traveller's User Id Optional comment by the user Travel Questionnaire Question text Vote (1-5)
Response / Output	No Output	-	-
Exceptions:	No exceptions;		
Notes and Issues:	No specified;		

Interface 56 - BusinessAnalyticsServiceI

Interface ID	56		
Interface Name	BusinessAnalyticsServiceI		
Purpose of the interface	This interface allows to provide information concerning specific KPIs, happenings and weather information, analysis of messages collected by social media;		
Requestor	Travel Companion		
Provider	Business Analytics		
Description	This interface allows to provide information concerning a specific KPI, happenings taking place in a specific city and weather information, summary information on messages collected by social media;		
Preconditions	A Traveller wants to visualize some business analytics or even useful information related to travel data		
Postconditions	The IT2Rail Business Analytics collects KPIs, happenings and weather data; then sends them to Travel Companion mobile app in order to show them to the IT2Rail user;		
GetBAKPIs			
Request / Input	kpi_names	1;;n	Names of required KPIs
Request / Output	Res	1;;n	Values of required KPIs
GetHappenings			
Request/Input	startPeriod endPeriod Location • hasName	1 1 1 1	Initial date of the time window Final date of the time window Place of interest City name
Request / Output	Res	0;;n	List of happenings taking place in the required city during the input time window
GetWeatherInfo			
Request/Input	startPeriod endPeriod Location • hasName	1 1 1 1	Initial date of the time window Final date of the time window Place of interest City name
Request / Output	weatherInfo	1;;n	List of weather information for the required city during the input time window
Exceptions	No exceptions;		
Notes and issues	No specified;		

Interface 57 - Get_Itinerary_OffersI

Interface ID	57		
Interface Name	Get_Itinerary_OffersI		
Purpose of the interface	The purpose of this interface is to receive itinerary offers computed by Travel Shopping;		
Requestor	Travel Shopping		
Provider	Business Analytics		
Description	The purpose of this interface is to receive itinerary offers computed by the IT2Rail Travel Shopping;		
Preconditions	The Travel Shopping completed all its activities related to a mobility request; Itinerary offers were sent to Travel Companion after completing TS activities;		
Postconditions	The Business Analytics receives itinerary offers and starts its own computation;		
Request/Input	Itinerary offers / itineraries: ID Amount Note Travel Episode • Departure point ▪ Label ▪ Type ▪ Reference ID ▪ Geo code ▪ Date ▪ Time • Arrival point ▪ Label ▪ Type ▪ Reference ID ▪ Geo code ▪ Date ▪ Time • Mode • Travel Expert • Operator • Transport ID • Class • Carbon footprint PRM information	 1 1 0;;n 1;;n 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0;;1 1 1	List of Itinerary offers / itineraries ID of the itinerary Price associated to the travel solution Free text to give warning associate to itinerary List of segments composing the travel solution Departure point Name of the departure point Type of the point (AIR, Rail, bus station) Code associated (IATA code, …) Latitude and longitude Departure date Departure time Arrival point Name of the arrival point Type of the point (AIR, Rail, bus station) Code associated (IATA code, …) Latitude and longitude Arrival date Arrival time Transport mode Travel Expert which provides the info Operator of the segment (carrier, rail, …) Flight number, train number, bus line Passenger class (first, business, economy, …) Estimation of the segment footprint Information about PRM capabilities
Request / Output	No output	-	-
Exceptions	No exceptions;		
Notes and issues	No specified;		