Contract No. H2020 – 636078

INFORMATION TECHNOLOGIES FOR SHIFT TO RAIL

D3.2 – Booking & Ticketing Specifications Document

Due date of deliverable: 31/10/2017

Actual submission date: 20/11/2017

Leader/Responsible of this Deliverable: THALES

Reviewed: Y

Document status		
Revision	Date	Description
1	12/12/2016	AREL Draft version for review.
2	02/02/2017	AREL internal version – This revision includes a lot of comments from WP3 partners (UITP, Amadeus, SNCF and INDRA).
3	25/07/2017	FREL Draft version for review • INDRA & Thales Review. • AMADEUS updated Capella's diagrams. • UITP Review.
4	31/07/2017	Version after final review. Awaiting TMC approval.
5	28/08/2017	Review from WP7. Awaiting TMC approval.
6	23/10/2017	Final Version after TMC approval
7	20/11/2017	Final version after dissemination level's change

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

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

EXECUTIVE SUMMARY

This document is about the Booking and Ticketing functions specification, their description, their data model and their external interfaces.

The content of other functional areas of the project is not shown in detail because they are not in the Booking and Ticketing's scope.

All terms and acronyms are defined in the IT²RAIL glossary.

TABLE OF CONTENTS

Executive Summary 3
List of Figures 6
List of Tables 7
List of Abbreviations 8
1. Introduction 9
2. Referenced documents 10
3. Operational Aspects 11
3.1 Principles 11
3.1.1 Interoperability and interoperability Framework 11
3.1.2 Distribution 11
3.2 Scope and Purposes 12
3.3 Design Drivers 13
3.3.1 Semantic Interoperability 13
3.3.2 Service Architecture 13
3.4 Actors and Use Cases 14
3.4.1 Actors 14
3.4.2 Context 15
3.4.3 Use cases / Capabilities 17
4. Functional Aspects 19
4.1 Components 19
4.2 Functions 20
4.2.1 Booking and Ticketing Functions 20
4.3 Functions of involved actors 23
5. Design decisions and constraints 24
6. Capabilities: sequence diagrams 27
6.1 Capabilities specifications 27
6.1.1 Publish fare sets 27
6.1.2 Shop – Provide itinerary offer items 29
6.1.3 Booking 31
6.1.4 Pay and deliver Entitlement Token and Embodiment 33
6.1.5 Validate Token 35
6.1.6 Inspect the validity of the ticket 39
6.1.7 Manage Business Analytics 40
7. Exchanges 41

7.1 Data model 41
7.1.1 Booking data model 41
7.1.2 Shopping data model 43
7.1.3 Fulfilment data model 45
7.1.4 Payment data model 46
7.1.5 Validation data model 46
7.1.6 Fare product Data model 48
7.2 Interfaces 49
7.2.1 Manage Entitlement Token Interface 50
7.2.2 Manage Booking Interface 55
7.2.3 Validate Token B&T Interfaces 61
7.2.4 Publish ticketing data Interface 69
7.2.5 Manage Offer Item Interface 73
7.2.6 Manage Payment Interface 77
7.2.7 Digest of interfaces provided by Booking and Ticketing 82
8. Annex 84
8.1 System-wide decisions 84

LIST OF FIGURES

Figure 1: Booking and Ticketing main interactions 16
Figure 2: BT Use cases diagram (Mission / Capabilities & Actors) 18
Figure 3: BT components breakdown 19
Figure 4: Booking and Ticketing functions breakdown 20
Figure 5: Interactions between actors 23
Figure 6: Fare set Initialisation scenario 29
Figure 7: Provide Itinerary Offer items 30
Figure 8: Booking scenario 32
Figure 9: Payment and issuing scenario 34
Figure 10: Validate Token Use Cases diagram 36
Figure 11: Validate Token B&T 37
Figure 12: B&T Validation functions 38
Figure 13: Business Analytics issuing data requesting scenario 40
Figure 14: Booking engines data model 42
Figure 15: Shopping scenario data model 44
Figure 16: FulfillmentEngine data model 45
Figure 17: Payment data model 46
Figure 18: Entitlement & Token data model 47
Figure 19: Fare product data model 48
Figure 20: Booking and Ticketing required and provided interfaces 49
Figure 21: Manage Entitlement interface 50
Figure 22: Manage Booking Interface 55
Figure 23: Validate Token Interfaces 61
Figure 24: Manage Payload Interface 62
Figure 25: Consume and alter Token Interfaces 65
Figure 26: Publish ticketing data interface 69
Figure 27: Manage Offer Item Interface 73
Figure 28: Manage Payment Interface 77

LIST OF TABLES

Table 1: Referenced documents 10
Table 2: System-wide decisions 84

LIST OF ABBREVIATIONS

AREL	**A**dditional **REL**ease
BT	**B**ooking and **T**icketing
CW	**C**loud **W**allet
CDB L	**C**lass **D**iagram **B**lank
CRB L	**C**apability **R**ealisation **B**lank
DBMS	**D**ata **B**ase **M**anagement **S**ystem
ETE	**E**ntitlement **T**oken **E**mbodiment
FREL	**F**inal **REL**ease
FS L	**F**unction **S**cenario
GUI	**G**raphical **U**ser **I**nterface
GDS	**G**lobal **D**istribution **S**ystem
ID	**ID**entification
IDB	**I**nterface **D**iagram **B**lank
LCBD	**L**ogical **C**omponent **B**reak**D**own
LFBD	**L**ogical **F**unction **B**reak**D**own
MCB	**M**issions **C**apabilities **B**lank
NA	**N**ot **A**pplicable
PA	**P**ersonal **A**ssistant
PRM	**P**erson with **R**educed **M**obility
SBP	**S**oftware **B**est **P**ractices
TBD	**T**o **B**e **D**one
TC	**T**ravel **C**ompanion
TCO	**T**icket **C**ontrolling **O**rganisation
TOC	**T**ransport **O**perating **C**arrier
TS	**T**ravel **S**hopping
TSO	**T**ravel **S**ervice **O**perator
TSP	**T**ravel **S**ervice **P**rovider
URI	**U**niform **R**esource **I**dentifier
URL	**U**niform **R**esource **L**ocator
WP	**W**ork **P**ackage

1. INTRODUCTION

This document identifies the design decisions and specifications that shall be applicable for the Booking and Ticketing (hereafter BT) process in the IT²RAIL ecosystem. It provides a functional architecture to all stakeholders. It has been defined by the project “Booking and Ticketing” technical community and it is used as a contribution for the detailed service specifications produced by the technical coordination area. This document is a separate contribution (Booking and Ticketing) and has been consolidated and aligned to form a coordinated view of architecture and shared data interfaces.

The main results of the deliverable are a set of principles for a traveller centred approach for a seamless travel experience, a single conceptual architecture, a description of functions, of their data model and a description of messages exchanged involved in the Booking and Ticketing processes.

Therefore, this document specifies the main functions of the Booking and Ticketing process, and for each of them, their functioning and external exchanges.

2. REFERENCED DOCUMENTS

This section lists the document reference number, title, revision, and date of all documents referenced in this specifications document.

Reference Number	Title	Revision	Date
Part of D7.10	IT²RAIL glossary, part of Development Readiness Review Pack – D7.10	Iteration 2	TBD
NA	IT²RAIL-Proposal_second stage_SECTION 1-3_28082014_.pdf	1	09/10/14
NA	FM Submission IP4-V4.pdf	4	08/11/14
ITR-WP1-T-TRI-014-01	WP1 specifications	1	19/04/16
ITR-T2.2-I-HAC-002	Shopping (IT²RAIL WP2) specification document.	1	27/05/16
ITR-T5.2-D-SNC-001	Travel Companion specification document (WP5)	1	02/10/15
ITR-T1.1-T-FHG-006	WP1 Ontology	2	12/12/15

Table 1: Referenced documents

3. OPERATIONAL ASPECTS

3.1 PRINCIPLES

3.1.1 Interoperability and interoperability Framework

This document (the “Booking and Ticketing specifications”) describes the elements that are mandatory for booking and ticketing systems to interoperate in the IT²RAIL ecosystem regardless of each implementation. For this reason the following specifications are functional and non-technical and this document may refer to the Interoperability Framework as defined by the Interoperability Framework.

The Booking and Ticketing functional area will rely on the Interoperability Framework to resolve interoperability issue between implementations. This document presents WP3 interfaces with their associated data. Such interfaces are mapped to functional exchanges and the provided data model is strictly compliant with the IT²RAIL ontology.

Aside the interface and data, the Interoperability Framework should be invoked to dynamically resolve resources (like BT services) when the resource location is not statically defined or part of any input. When needed, the Interoperability Framework artefacts may be used to transform data model or adapt messages to ensure technical interoperability.

3.1.2 Distribution

The Booking and Ticketing Work Package defines the elements needed for multiple booking systems and ticketing systems to interoperate. As a result, this work package will always consider the uttermost distributed scenario for any given situation. This principle will ease the identification of every business entity with one or many BT defined elements.

3.2 SCOPE AND PURPOSES

This document describes the functional scenario and interfaces of Booking and Ticketing systems to interact within the IT²RAIL global use case for F-REL[1]. This use case presents some particularities:

- This is a co-modal[2] use case, so involved BT systems are independent from each other.
- Only pre-paid products are dealt with by IT²RAIL (others shall be covered by IP4).
- Reimbursement/refund is not considered.
- The payment is done at the end of the booking process for finalising the entitlement, and the trip is made of journeys bunched as co-modal functions of the trip.
- Responsibilities related to a disruption in a transport mode are not considered.

[1] FREL: Release of Final Functions. Third and last iteration for the final release in IT²RAIL.

[2] A travel is considered as co-modal when it involves multiple modes and operators (multimodal) but there are multiple contractual agreements, all linked to the passenger, one with each Travel Service Provider. According to the IT²RAIL Newsletter (September 2016):

"Comodality is where the passenger selected travel solution consists of an aggregation of Travel Service Provider (TSP) products/ services, performed in the distribution link of the supply chain. None of the contributing TSPs are aware of the contribution of the others: multiple 'tickets' (transport contracts) are established between the passenger and each contributing TSP, each guaranteeing arrival only at the destination of the service provided by each TSP."

3.3.1 Semantic Interoperability

In order to enable the interoperability principle and interact with the Interoperability Framework, any compliant BT system will provide (by annotating the provided services) or use semantic interoperability technology (resolver, broker…) as defined in Interoperability Framework specifications (ITR-WP1-T-TRI-014-01). To be precise BT is going to use a Broker to connect to the TSPs that will be one of the Interoperability Framework assets.

3.3.2 Service Architecture

Service architecture promotes the following concepts:

- Service isolation: a provided service should be independent from the rest of the eco-system.
- Loose coupling: a modification of a service should not impact (or very little) on the other services.
- Contractual interface: a service commits on its interfaces.
- Implementation agnostic: a service definition is independent from its implementation.

Using a service architecture, the IT²RAIL specifications do not describe a single operational system but the way multiple systems/services should cooperate to offer the best service possible to the end-user.

IT²RAIL being by nature distributed, the use of an architecture based on services provides multiple advantages:

- The first one is the possibility to interface legacy systems to the new eco-system with minimal adaptation.
- Secondly, service architecture targets seamless integration with loose coupling providing isolation of existing systems and enabling new business models.

As a result, one must always consider that there could be multiple instantiations of a service as well as multiple allocation of a service to roles depending upon the business model. As an example: a retailer may implement a ticket issuing (Entitlement fulfilment engine in IT²RAIL) service to issue entitlements[3] or the Retailer can rely on Travel Service Providers to issue multiple entitlements needed for a co-modal trip.

[3] If appropriate commercial agreements have been signed with relevant Travel Service Providers.

3.4 ACTORS AND USE CASES

3.4.1 Actors

The main actors involved with Booking and Ticketing systems are the following:

- The **Customer**: Using his / her smart device, he / she does a mobility request, selects one or several segments to create his / her trip and pays his / her booking(s) (in IT²RAIL the payment is done at the end of the booking process for finalising the entitlement, and the trip is made of journeys bunched as co-modal functions of the trip[4]). The Customer may not be the Traveller.
- The **Traveller**: using his / her smart device and, possibly one or several physical media, "taps" one of them to access to the travel service or to the transport network. In case of manual access check, he/she presents the relevant medium to the staff (presenting as well in some cases an official ID document).

In a general way, the Traveller may be different than the customer, but in the scope of IT²RAIL, it is simplified, the Customer and the Traveller are considered as the same person (the user).

In addition to the Traveller / Customer actors, a number of other actors play a role in the IT²RAIL environment. These are typically based in service providers (e.g. the providers of travel services).

- **Travel Service Provider** (TSP): This actor is offering the customer its fare products for purchase (Travel Shopping and Booking and Ticketing). It also provides the travel service corresponding to the fare product (Booking and Ticketing, Trip Tracker & Business Analytics).
- **Travel Service Operator**[5] (TSO): This actor operates the travel service.
- **Merchant**: This actor is receiving the payment in a transportation sale from the Customer. It is in charge of the payment apportionment if needed.
- **Distributor**: This actor is providing the Travel Service Provider fare products to retailers for sales. This mediation is performed through Travel Shopping activity.
- **Retailer**: This actor is managing the sales, providing the customer with the product it seeks.
- **Payment Service Provider**: This actor is a banking service selected and used by the Customer to pay his / her journeys.
- **Ticket Controlling Organisation** (TCO): A Ticket Controlling Organisation is any organisation charged with inspection of passengers' tickets.
- **Issuer:** an organisation issuing Entitlement, Token and Embodiment.

[4] Only pre-paid products are dealt with by IT²RAIL, others shall be covered by IP4.

[5] The TSO could be named also TOC: "Transport Operating Carrier", the company running the vehicles used for transport.

Alternatively, the BT functional scenarios involve IT²RAIL internal actors:

- The **Travel Companion Application** (or Personal Application): Travel Companion – Hosted by the Smart Device, it provides a Graphical User Interface (GUI) to the Customer / Traveller for interacting with the systems. The Smart Device also hosts a tapping module, software which enables the smart devices to interact with the access systems of the TSO. Each TSO has its own tapping module. The tapping module uses the relevant private data of the TSO located in the token and sends them to the access system. These private data are written by the TSO in the token during the entitlement stage.
- The **Travel Companion Wallet** (or Cloud Wallet): Travel Companion – This system in the Cloud provides persistence to the wallets of the Customer / Traveller.
- **Interoperability Framework**: This system provides tools for interoperability. It also requests TSP to get their topologies, ItinerariesOfferItems, availability, prices, bookings and payloads. It is involved in all the exchanges between systems.
- **Travel Shopper:** –Travel Shopping – This element of the Shopping requests TSPs through the Interoperability Framework to get a valuated ItineraryOfferItem.
- **Business Analytics**: This system, using Booking and Ticketing usage data (amongst several other data sources) and big data technologies, analyses how the transportation ecosystem works and helps managers to render it better within the frame of IT²RAIL (co-modality).
- **OfferProvider:** An organisation also BusinessUser providing travellers with travel offers.
- **OfferItemProvider:** An organisation providing offer items to OfferProvider.
- **Travel Expert:** A system providing travel expert broker and/or Travel expert broker services

3.4.2 Context

The Booking and Ticketing functional area defines functions and interfaces. These are involved in exchanges with other IT²RAIL functions. The following schema (Figure 1) shows an overview of the main interactions in the IT²RAIL eco-system.

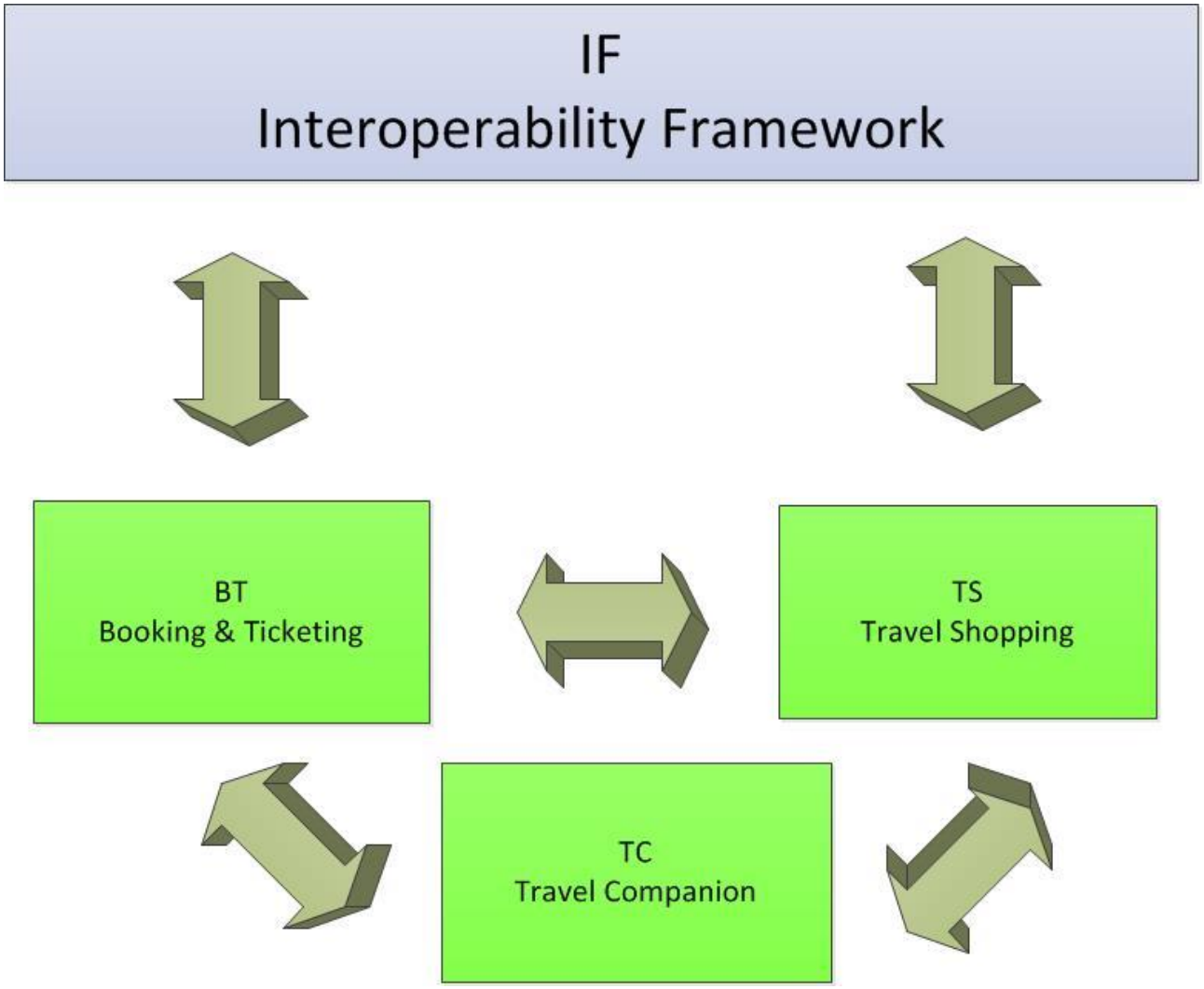


Figure 1: Booking and Ticketing main interactions

This diagram illustrates the relationship between Booking and Ticketing, Travel Companion and Travel Shopping from a functional point of view.

This diagram highlights the fact that the Shopping compliant functions may have no pre-established relationship with ticketing systems emphasising the need of a distributed environment.

3.4.3 Use cases / Capabilities

IT²RAIL Booking and Ticketing functional area is contributing to the IT²RAIL Mission and Capabilities.

The main capabilities of the Booking and Ticketing are the following:

- "Booking": Booking Travel Episodes (each Travel Episode is associated to a given TSP)
- "Pay and deliver Entitlement Token and Embodiment":
    - Paying the Travel Episodes booked(not developed in IT²RAIL because it's out of scope)
    - Issuing entitlements to travelling, with the associated tokens – with the payloads to be used by the Tapping Module
    - Issuing embodiments on physical media, in case the smart device hardware is not compatible with the access systems of a TSP for this journey. In IT²RAIL, this is usually a proprietary function.
- *Managing fare set (Fare products, Fare Rules and Topologies) and logical and physical topologies (Legacy perimeter, out of scope in IT²RAIL).*
- *Computing pricing according to the fare set (Fare products, Fare Rules and Topologies) and for the requested itinerary (Legacy perimeter, out of scope in IT²RAIL).*
- "Perform After Sales": Performing after sales processes (not developed in IT²RAIL). After sales processes are usually the operations on a travel contract: information, and when possible, modifications, refund, renew, compensation...
- "Validate token": Enabling validation using the Smart Device (in fact: its Tapping Module) or a physical medium owned by/provided to the traveller. Validation: consumption of the token
- "Inspect the validity of the ticket": Enabling inspection using the Smart Device or the physical medium of the Traveller (not developed in IT²RAIL). Inspection: Control of the validity of the token
- "Publish Fare sets": Publishing ticketing data to actors that need it (not developed in IT²RAIL)
- All capabilities: Providing data for Business Analytics.

The following diagram shows the Missions[6] (M) and their associated Use Cases / Capabilities (C) where the Booking and Ticketing functions are involved.

For example, the Mission "Manage mobility request and travel rights delivery" and its Capabilities (like Booking, Paying...) allows the Traveller (e.g.: Jane) to do a mobility request, to book and pay it and to get his / her entitlement(s) and token(s) to travel (Figure 2). Please note that "Shop" is not in

[6] Note: A Mission (labelled "M" in the following diagram) is a set of inter-related Use Cases / Capabilities.

the scope of Booking and Ticketing but this capability is let here to show the whole process a Traveller can do.

Some others actors like Issuer or Merchant have been involved in that mobility request.

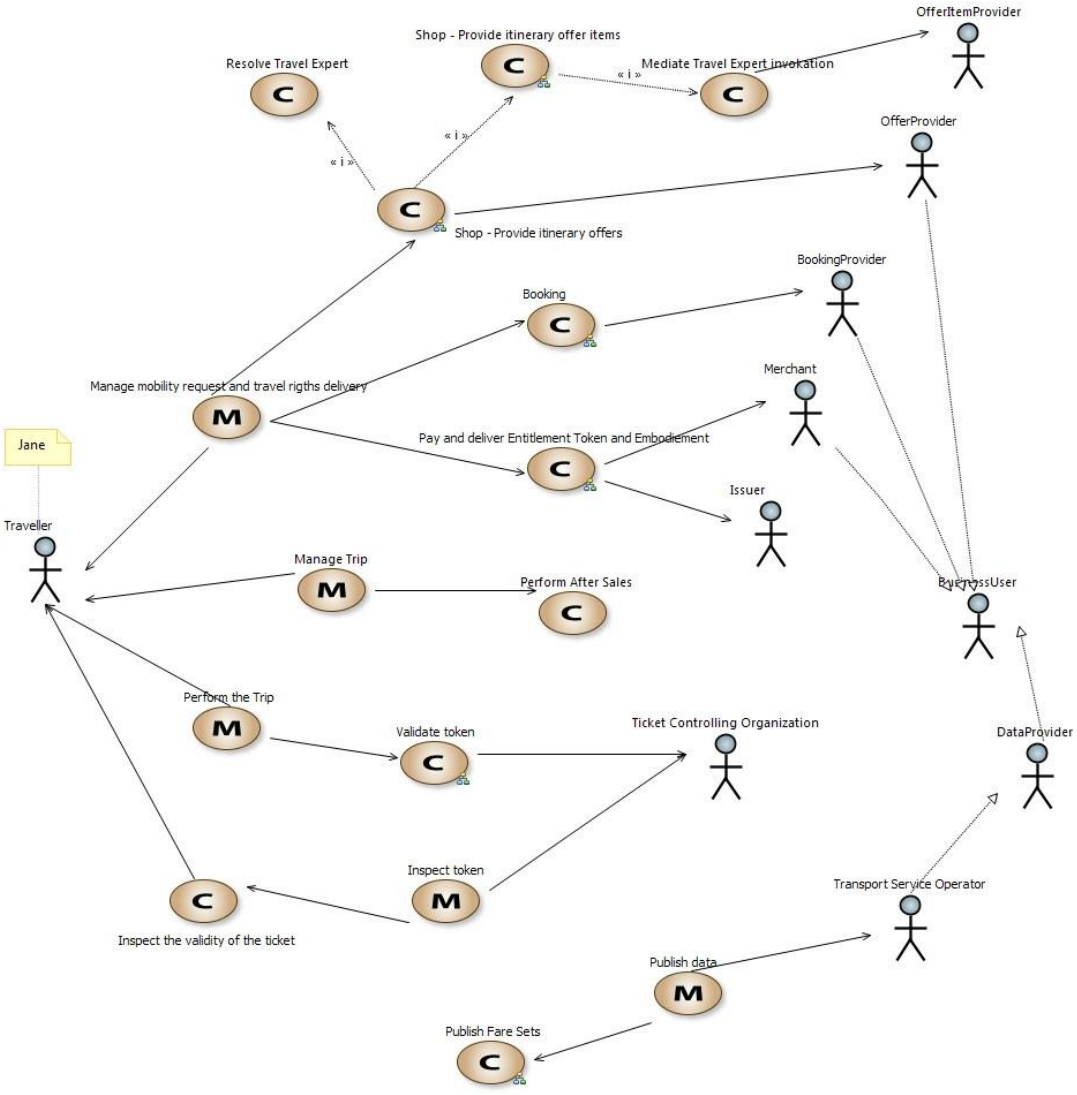


Figure 2: BT Use cases diagram (Mission / Capabilities & Actors)

Model reference: *[MCB] Capabilities - BT Only (*SVN Version: 814)

4. FUNCTIONAL ASPECTS

4.1 COMPONENTS

The Booking and Ticketing is decomposed into several components (Figure 3):

- BT Validation Engine
- BT Payment Engine
- BT Inspection Engine
- BT Asset Management
- BT Fare Product Engine
- BT Consumption Engine
- BT Booking Inventory
- BT Booking orchestrator
- BT ETE[7] Issuing
- BT Update Token
- BT Manage Fare sets

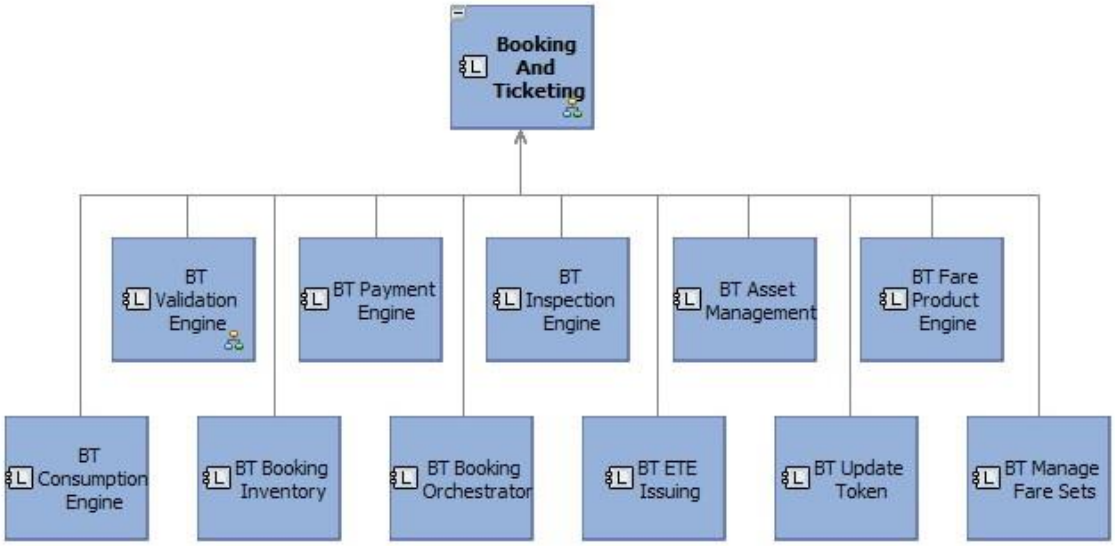


Figure 3: BT components breakdown

Model reference: *[LCBD] BT - Booking and Ticketing Component Breakdown (*SVN Version: 814)

The diagrams of the next sections show the functions and sub-functions involved in the Booking and Ticketing domain (breakdown of functions).

[7] BT ETE : Booking and Ticketing Entitlement Token Embodiment

These functions will then be used in the specification scenarios diagrams.

4.2 FUNCTIONS

4.2.1 Booking and Ticketing Functions

The Booking and Ticketing functions are matching ticketing system capacities. As a result there are no IT²RAIL specific functions. For a given functionality all further described functions should already exist under a form or another in legacy ticketing systems. In other words, they are based on different and specific structures and software.

The Booking and Ticketing main functions are organised as follow (Figure 4):

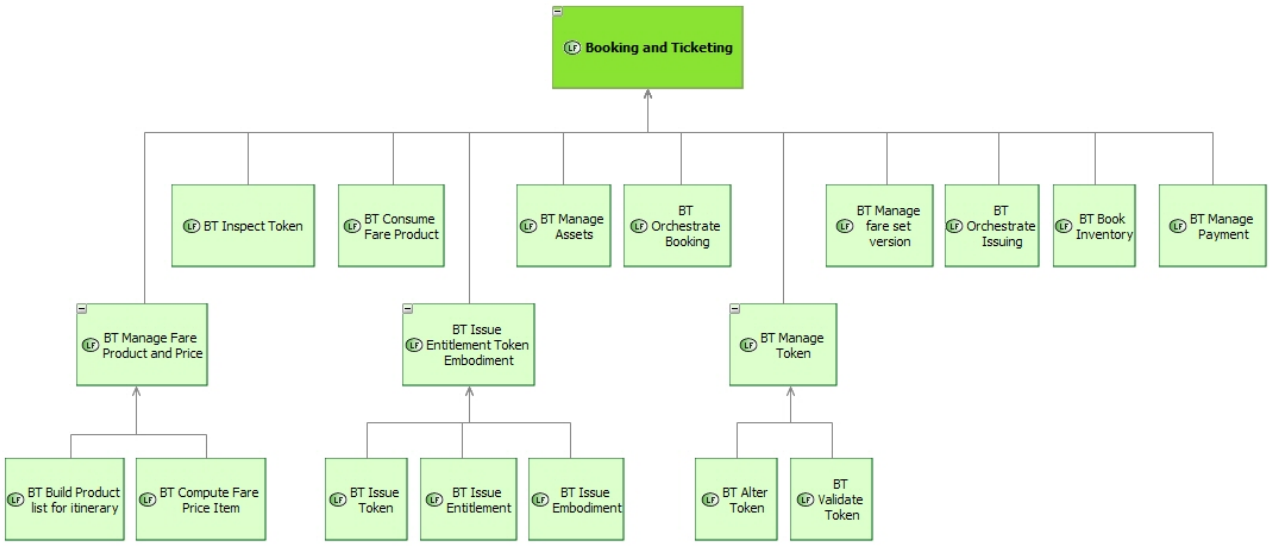


Figure 4: Booking and Ticketing functions breakdown

Model reference: *[LFBD] BT - Booking and Ticketing Function Breakdown (*SVN Version: 814)

The Booking and Ticketing functions perform the following processes in a ticketing system:

From the **BT ETE issuing** (Fig 3) component:

- **BT Orchestrate Issuing:** In the IT²RAIL ecosystem, a travel may require multiple complementary travel rights. In order to take this capability into account, the fulfilment of such travel must have orchestration capabilities to manage the multiple travel rights issuing. This function is provided by a Booking Provider. The responsible of this function is a broker.
- **BT Issue Entitlement Token Embodiment**: This function is in charge of issuing the travel rights under the form of an Entitlement and/or a Token and/or an Embodiment. All these three capabilities are allocated to three separate functions: **BT Issue Entitlement, BT Issue Token** and **BT Issue Embodiment.** This function is provided by one or multiple Issuers. The responsible of this function is a TSP.

From the **BT Booking orchestrator** (Fig 3) component:

- **BT Orchestrate Booking**: This function applies business rules to orchestrate the booking with multiple booking engines for a travel. This function is provided by a Booking Provider. The responsible of this function may be a TSP or another actor (like distributor).

From the **BT Booking Inventory** (Fig 3) component:

- **BT Book Inventory**: This function manages the availability of capacity controlled products. This function is provided by a Booking Provider. The responsible of this function is a TSP.

From the **BT Update Token** (Fig 3) component:

- **BT Manage Token**:
    - **BT Alter Token:** This function is also responsible for the proper synchronisation of the Token between the Travel companion cloud and application cache.

From the **BT Fare Product Engine** (Fig 3) component:

- **BT Manage Fare Product and Price:** This function is split into the following functions:
    - **BT Compute Fare Price Item**: This function is able to apply the Fare Rules to Fare products. It holds the currently applicable version of a TSP fare products and ensures that at shopping time and consumption time, the fare rules are identical. This function is provided by an OfferItemProvider. The responsible of this function may be a TSP or another actor (like distributor or broker).
    - **BT Build Product List For Itinerary**: This function returns a list of fare products according to the parameters of the request. This function is needed only if the journey planner is distinct from the OfferItemProvider (e.g. in the urban environment). This function is provided by an OfferItemProvider. The responsible of this function may be a TSP or another actor (like distributor or broker).

From the **BT Payment Engine** (Fig 3) component:

- **BT Manage Payment**: This function is responsible for the payment of a travel solution and manages the authentication, the creation of the financial transaction, the settlement and the cancel. This function implementation is strongly linked to the Travel Companion Payment module. This function is broken down into several functions to allow a two-step payment[8]. This function is provided by a Merchant.

[8] The "two-step payment" terminology is mainly used by payment providers and refers to a payment involving more than one step.

From the **BT Inspection Engine** (Fig 3) component:

- **BT Inspect Token**: This function is similar to the **BT Manage Token** function except that it deals with inspection mechanisms. It, for example, may consider additional sources of information (Ticketing system back office) to validate the travel rights of a passenger. This function is provided by a Ticket Controlling Organisation. The responsible of this function may be a TSP or another actor (like third party).

From the **BT Consumption Engine** (Fig 3) component:

- **BT Consume Fare Product**: This function is part of the ticketing validation process. It is responsible of the consumption of the fare product. For example, given a 9 trip multi-rides ticket, the consumption of a trip will generate an update of the fare product instance with an 8 trip value. This function may rely on the **BT Alter Token** function to ensure the product consumption is correctly updated to all involved parties. This function is provided by a Ticket Controlling Organisation. The responsible of this function may be a TSP. The responsible to synchronise the TC CW with the updated payload token must be the BT Booking Orchestrator. As the TSP cannot communicate with the TC CW directly.

From the **BT Asset Management** (Fig 3) component:

- **BT Manage Assets**: This function deals with all operational processes for a given TSP. It includes asset management. This function is provided by a Travel Service Provider or a Travel Service Operator.

From the **BT Manage Fare Sets** (Fig 3) component:

- **BT Manage Fare set version**: This function deals with all operational processes for a given TSP. It includes both fare product validity and versioning. This function is provided by a TSP.

From the **BT Validation Engine** (Fig 3) component:

- **BT Manage Token**:
    - **BT Validate Token**: This function is interacting with the Tapping Module function in order to exchange Token information. This function is able to interpret Token information and validate the fare products (example of validation functions: apply of deny-lists, verification of security protocols …).

4.3 FUNCTIONS OF INVOLVED ACTORS

The following are the functions performed by external actors involved in the Booking and Ticketing functional scenarios.

- **Ask for Itinerary Offer (A)** [9]: This function is performed by the Customer and triggers the shopping process.
- **TS Travel Expert Offer Builder**: This function is defined in the Travel Shopping specification.
- **BookOffer (A)**: This function is performed by the Customer and triggers the booking process.
- **PayOffer (A)**: This function is performed by the Customer and triggers the payment and issuing process.
- **ValidateToken (A)**: This function is performed by the Traveller and triggers the Validation BT process.

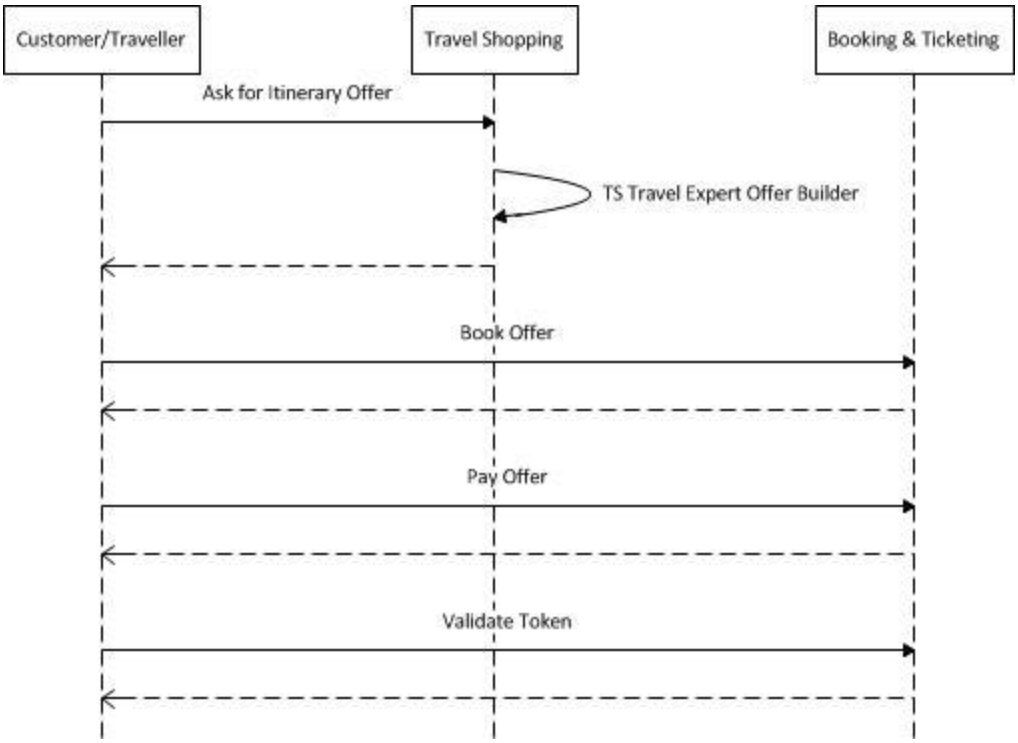


Figure 5: Interactions between actors

Note: The previous figure shows the main interactions between main actors. It's an overview of the Booking and Ticketing process. It's a simplified view of the BT.

[9] Each actor function name is suffixed with a "(A)" pattern. It allows in the Capella models and diagrams to easily distinguish a system function from an actor function.

5. DESIGN DECISIONS AND CONSTRAINTS

This chapter contains all the design decisions that have arisen from the Booking and Ticketing specifications development that may impact on the rest of the IT²RAIL ecosystem:

- Design decisions related to **itinerary offers**:
  - An itinerary offer (ItineraryOffer) is identified by a unique identifier (ItineraryOfferRefId).
  - The itinerary offers are temporarily stored during the Travel Shopping process.
  - The itinerary offers are time-limited (after the end of the validity, the offer is no longer valid and cannot be booked and purchased).
  - The details of an itinerary offer can be retrieved (from TS and CW in IT²RAIL context) using the ItineraryOfferRefId as reference.
  - An itinerary offer is composed of one or several valid itinerary offer items (ItineraryOfferItem).
  - An IT²RAIL compliant system will answer to a Traveller's mobility request with one or more itinerary offers.
  - An IT²RAIL Travel shopper must provide the details of the itinerary offers already created and still valid.
  - An itinerary offer item is linked to one and only one TSP (through its TravelExpert class).
  - The itinerary offer item details contain information regarding the state of the price, if it is guaranteed or not.
  - The entitlement is always related to an itinerary offer.
  - The Interoperability Framework is able to identify which TSP has to be called to book an itinerary offer item.

- Design decisions related to **Booking**:
  - The Booking Orchestrator is able to call the Travel Shopper to get the itinerary offer details. It should be the CW but in IT²RAIL context, the BT orchestrator is linked to the TS.
  - The Interoperability Framework may be called by the Booking Orchestrator for any exchange with external TSP (if the TSP offering the booking function is different from the booking provider providing access to another booking function such as BT Book Inventory). For instance, Air TSP as Booking Orchestrator and Booking Provider for Air GDS content will call the Interoperability Framework for the interaction with external booking providers but not for his own content.
  - The Interoperability Framework is called at booking step when the TSP expects to have a lock of the itinerary offer item (through the LockInventory Interface).
  - Booking should be able to resynchronise in case the customer loses its connection. If the customer loses his connection during the payment loop, when he/she reconnects, he/she should be able to carry on the payment process. If Booking is not able to resynchronise, the payment will be automatically

cancelled when the customer doesn't complete the payment process after a meaningful time.

- The Booking Orchestration function is able to identify which Travel episodes need a call to TSP booking functions based on the information provided in the itinerary offer item details.
- The Booking Orchestration function is able to orchestrate different booking calls to different TSP booking engines through the Interoperability Framework.
- The Booking Orchestration function assembles all booked ItineraryOfferItems in a single booked ItineraryOffer which price is guaranteed (GuaranteedPriceBooking).
- An itinerary offer item needs a new call at booking step to obtain the guaranteed price of this item.
- A GuaranteedPriceBooking is identified by a BookingRefID and it contains Bookings with a confirmed prices element.
- Booking elements (BookingElement) are itinerary offer items with status; this status comes from the lock step of the booking process (LockInventory call).

- Design decisions related to **Validation** / Tapping Module:
  - There is only one Token type (e.g. EMV, barcode, NFC…) to use per travel episode.
  - The validation is done by a Tapping Module, external to Travel Companion application, provided by the Ticket Controlling Organisation, or its subsidiary.
  - The existing validation structures and processes should not require major adaptation to plug into the IT²RAIL ecosystem. The only required adaptation for IT²RAIL should concern Tapping Module and Token issuing.
  - A compatible Tapping Module must be installed in the Smart Device as soon as possible to be sure that the Token will be available for the validation step. It is possible to install it prior to the travel. The TC calls the plugin application. If it's not yet installed, it will be then.
  - The Travel Companion may install many Tapping Modules, to be able to validate them considering different TSP.
  - An IT²RAIL compliant ticketing system must be able to provide the Traveller with its validation status if provided with the Token, eventually through the use of the corresponding Tapping Module.

- Design decisions related to **Issuing**:
  - The issuing processes are handled by each Issuer.
  - The booking and issuing orchestration functions will only orchestrate the Issuers. Each Issuer will produce its own entitlement / token / embodiment once it has got a proof of payment.

- The issuer orchestration function is able to orchestrate different issue entitlement calls to different issuers/TSPs thought the Interoperability Framework.
- One Entitlement can be related to one or multiple Tokens.
- One Token is related to at least one Travel Episode.
- All Entitlements in IT²RAIL are co-modal. As a consequence, each issuer (TSP) will produce its own entitlement.
- Each issuing must be cancellable, as the next issuing step may trigger a global rollback.

- Design decisions related to **Payment** (not in IT²RAIL scope):
  - Each payment must be cancellable as the next payment step may trigger a global rollback.
  - In IT²RAIL scope, an issuing must not occur if its payment is not confirmed.

6. CAPABILITIES: SEQUENCE DIAGRAMS

This section synthesises all the capabilities of the Booking and Ticketing system. These specifications are written following the use cases methodology. Each use case is described with text and sequences diagram(s). The data exchanged across the actors and systems are then described in detail in the Interface chapter.

The “nominal / normal” scenarios are described in this specification document. In IT²RAIL scope, the error management (consequently to failure cases) is not described.

6.1 CAPABILITIES SPECIFICATIONS

6.1.1 Publish fare sets

Fare sets policies address all the different types of fares that exist in a single transport network, or in a group of transport networks (if integrated), related potentially to different parameters like:

- Market segmentation, such as customer/traveller profiles (PRM…) and peak users demand period, date and/or time of travel,
- Network physical and logical topologies,
- Complementarity between modes (e.g. parking + public transport at given locations),
- Customer management policies, such as luggage management, incentives towards loyalty programs and special tickets (return ticket and others),
- Budgetary policies: objectives and resources issues (network capacity and offer),
- Economic, social or geographical policies set up by public authorities (especially in case of public service contracts).

The fare policy is a subject that is very important to be kept under control for TSP. It has to be pointed out that some fares depend only upon commercial practices when the services are operated in an open market, and that other categories of fares are decided under the control of public authorities when the services are not commercially viable and subject to public service requirements (see Regulation (EU) 1370/2007 recently amended by Regulation (EU) 2016/2338 as part of the 4th Railway Package legislation approved in December 2016). Each organisation (TSP/TSO and/or public authority) has its own fare policy and strategy. However some fares can be common between some transport organisations when a ticketing interoperability and commercial agreements are put in place either on a voluntary basis or through decisions taken by public authorities when public service requirements are at stake.

The Fare Policies can be based on customised pricing strategies (e.g. for products or services based on current market demands evolving over time, which allow to get two different prices for two customers travelling in the same conditions) or non-customised pricing strategies (e.g. when same



prices are applied for a product for a given category of customers, not depending on the external or internal parameters).

In IT²RAIL context the fare sets are very different from one mode to another and therefore it's not yet possible to define a common fare set for all actors.

Usually, for transport modes that are under public authority management (especially urban and regional) the fare setting mechanisms are based on non-customised pricing strategies but can be very complex as they could be used as strategic tools for political reasons.

On the contrary, for long distance train and air, the fare sets are often simpler, but the price may change drastically from one date/time to another (businesses set flexible prices based on customised pricing strategies).

The objective for IT²RAIL is to identify the simplest fare set for each mode that matches with the type of the trip, and the possibility offered by the entitlement and validation processes for each segment.

This scenario describes a use case that may occur multiple times over time.

This use case is the publication of new operational parameters. For example, a Travel Service Provider (TSP) defines new Fare Products or alters existing Fare Rules. Following this a new version of operational parameters is issued. Such mechanism allows the management of fare product lifecycle and guarantees the homogeneity of the fare products and rules within the system.

Each time new release becomes available, managed by the Travel Service Provider, it updates the different systems involved in the eco-system. At minimum it must update:

- The *TS Publish Statistical data* logical function with the following information:
    - Version information
    - Logical topology
    - Physical topology
- The Booking and Ticketing *BT Fare Product Engine* component implementations with the following information:
    - Version information
    - Logical topology
    - Fare product
    - Fare rules
- The *TS Compute Itinerary Offer Item* logical function, if needed, with the following information:
    - Version information
    - Logical topology
    - Fare product description
    - Fare rules description

The initialisation scenario shows where the Booking and Ticketing functions are used (see Figure 6):

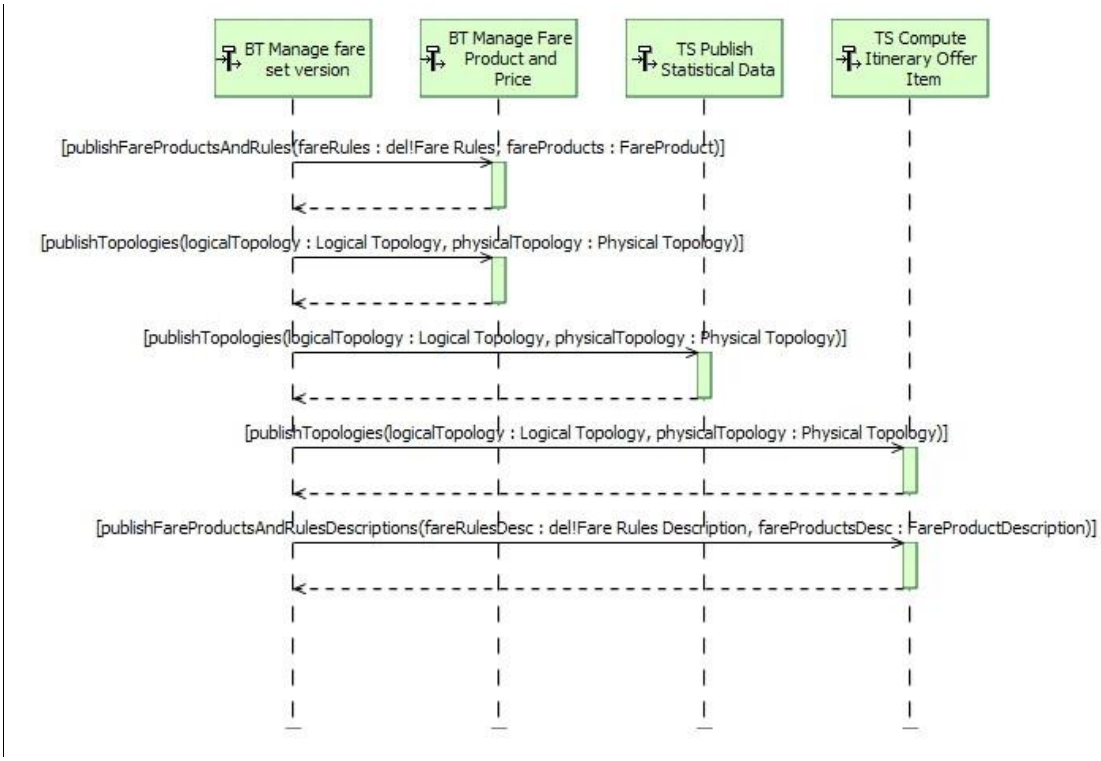


Figure 6: Fare set Initialisation scenario

Model reference: *[FS]L Publish Ticketing Data - BT (*SVN Version: 814)

In IT²RAIL, all the involved functions are provided by a single actor, mainly a TSP. So, the consistency of the fare set is guaranteed (by the legacy system). The implementation of such interfaces becomes irrelevant.

6.1.2 Shop – Provide itinerary offer items

This scenario describes a part of the shopping process. The shopping process can be summed up to the following: a customer launches its Travel Companion App and does a mobility request for a travel, to get a list of all available itinerary offers providing a solution to its mobility request. Then the customer may choose a particular valued itinerary offer to pursue the scenario with the booking process. Within this complex process, the Booking and Ticketing functions are involved when the shopping functions are requesting itinerary offer items from travel experts.

The Booking and Ticketing logical functions involved in this scenario are the following:

- *BT Build Product list for itinerary*: This function is used by some *TS Compute itinerary Offer Item* logical functions to build the itinerary offer item if the journey planner is not integrated to the offer builder.
- *BT Compute Fare Price Item*: This function is used by some *TS Compute itinerary Offer Item* logical functions to compute the fare price of an itinerary offer item when the considered product can be sold at shopping time.

The following system shopping scenario shows where the Booking and Ticketing functions are used (Figure 7):

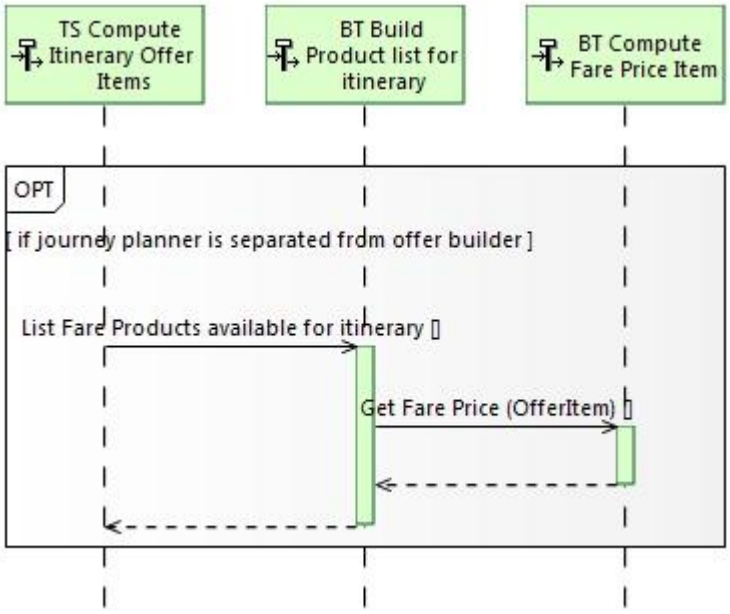


Figure 7: Provide Itinerary Offer items

Model reference: *[FS]L SBPI - Shop - Provide itinerary offer items - BT (step 1.1.1.1) - FREL version: Diagram updated (*SVN Version: 814)

For an overview of the shopping process, please refer to the Travel Shopping specification document.

This scenario requires a minimum of two exchanges from the logical functions of the *BT Fare Product Engine* component of the TSP:

- getOfferItemList: This interface returns to the travel shopper the list of available itinerary offer items to further constitute an itinerary offer.
- getFarePrice: This interface allows to obtain the anticipated price of a specific itinerary offer item. It is important to notice that the Fare Price is not the amount finally charged but a

complex structure, composed of several attributes. This price should be specified as confirmed or as informative.

These interfaces will be defined in chapter 7.2 and the data used will be defined in chapter 7.1.

6.1.3 Booking

This scenario describes the booking process: a Customer wants to book one valued itinerary offer. The aim of the Manage Booking is to convert an itinerary offer into a booking with a guaranteed price (GuaranteedPriceBooking).

The Booking and Ticketing logical functions involved in this scenario are:

- *BT Compute Fare Price Item*: This function is involved in the booking scenario as, depending on the TSP considered, the price computed before the booking may evolve between shopping time and booking time. In IT²RAIL scope, if the price is different, the reservation is cancelled.
- *BT Orchestrate Booking*: This function can be provided by a retailer, a distributor or a TSP. It is in charge of the booking of the complete itinerary offer by orchestrating the different booking engines involved in the trip.
- *BT Book Inventory*: This function is in charge of locking the capacity requested in the Offer for a particular travel episode. This is not a mandatory function: it depends on the business model of each TSP. For example, an itinerary offer item involving Air France will need a Book Inventory function but not in general for the urban modes.

The system booking scenario (Figure 8) shows where the Booking and Ticketing functions are used:

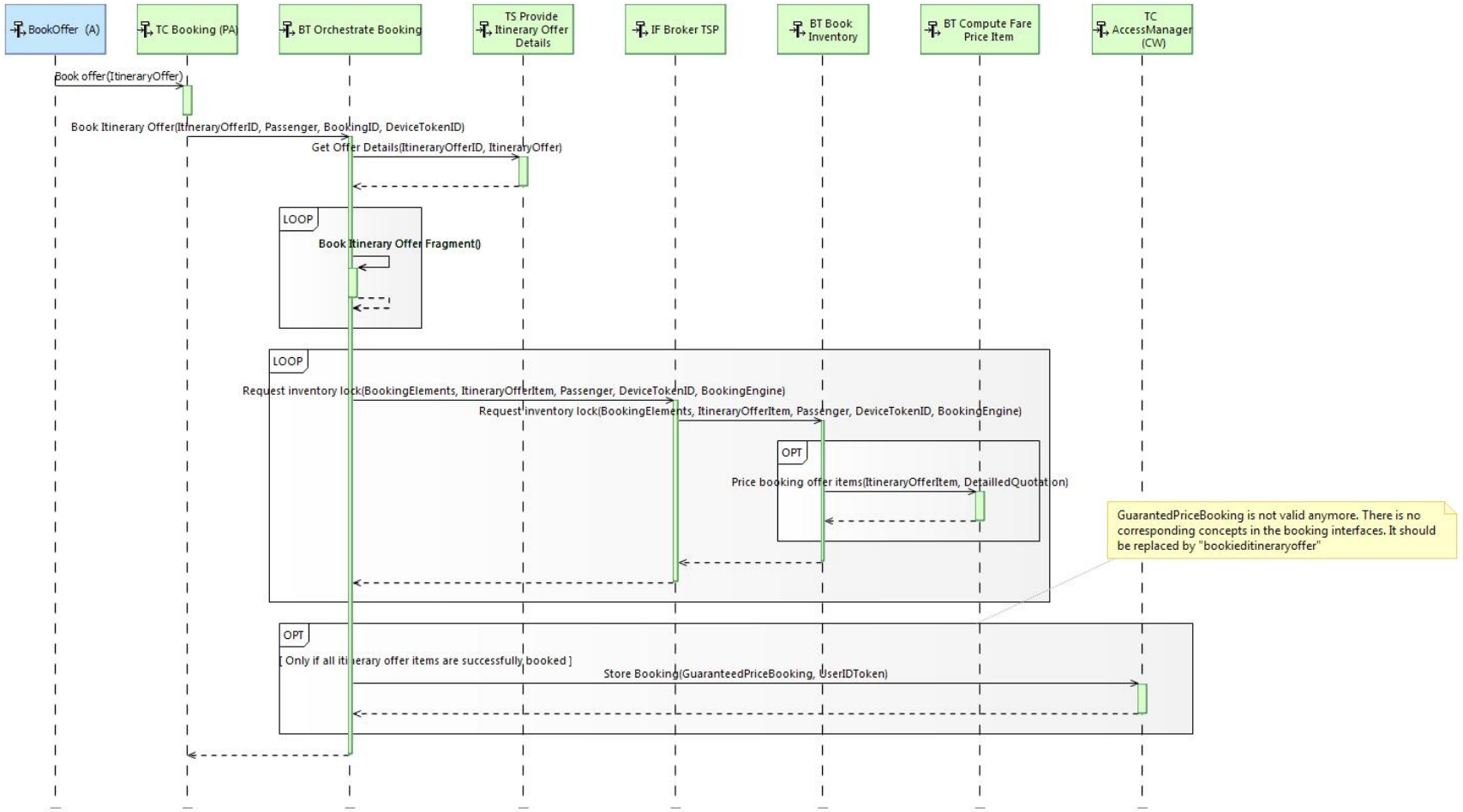


Figure 8: Booking scenario

Model reference: *[FS]L SBPI - Booking - BT (step 2) - FREL version: Diagram updated (*SVN Version: 814)

This scenario exposes the following interfaces of the Booking and Ticketing functions:

- The *BT Compute Fare Price Item* logical function exposes the following interfaces:
  - Price Booking offer items: In the scope of IT²RAIL, this interface is identical to the GetFarePrice of the shopping scenario.
  - Identify Critical Products Booking Failure: This interface identifies if at least one critical product exists.
- The *BT Orchestrate Booking* logical function exposes the following interfaces:
  - Book Valued Itinerary Offer: This interface books a whole valued itinerary Offer, detects if a critical product is missing, and cancels the booking if needed. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - Book Valued Itinerary Offer fragment: This interface is identical to the previously defined Book Valued Itinerary Offer interface with the notable exception that the Valued Itinerary Offer fragment only contains part of the itinerary Offer Items that may be handled by a particular instance of the BT Orchestrate Booking function.
  - Identify Critical Products Booking Failure: This interface identifies if at least one critical product exists among the list of failed elements of the itinerary offer.
- The *BT Book Inventory* logical function exposes the following interfaces:
  - LockInventory: This interface books the capacity corresponding to the travel episode and its associated offer items. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - ReleaseInventory: This interface allows releasing a booked offer item. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.

6.1.4 Pay and deliver Entitlement Token and Embodiment

This scenario aims at describing both the payment (Not in scope in IT²RAIL) and delivering sub-processes as a part of a one-stop shop. This process will only take place if the booking part has been performed and the Customer has selected an itinerary offer to book and pay for. The sub-process describes the steps to pay for the itinerary the customer has booked, and the entitlement and token issuing.

The Booking and Ticketing logical functions involved in this scenario are (Figure 9):

- *BT Orchestrate issuing*: This function is mandatory in this process. It can be provided by a retailer, a distributor or a TSP. It is in charge of orchestrating the different TSPs issuers involved in the Issue Entitlements Token and Embodiment for a GuaranteedPriceBooking.
- *BT Issue Entitlement Token Embodiment*: This function is in charge of issuing a valid Entitlement and/or Token for the customer. This function is part of the TSP business.
- The sub-functions of *BT Manage Payment*: These functions are performing the financial process with the merchant bank with regards to the selected payment means.

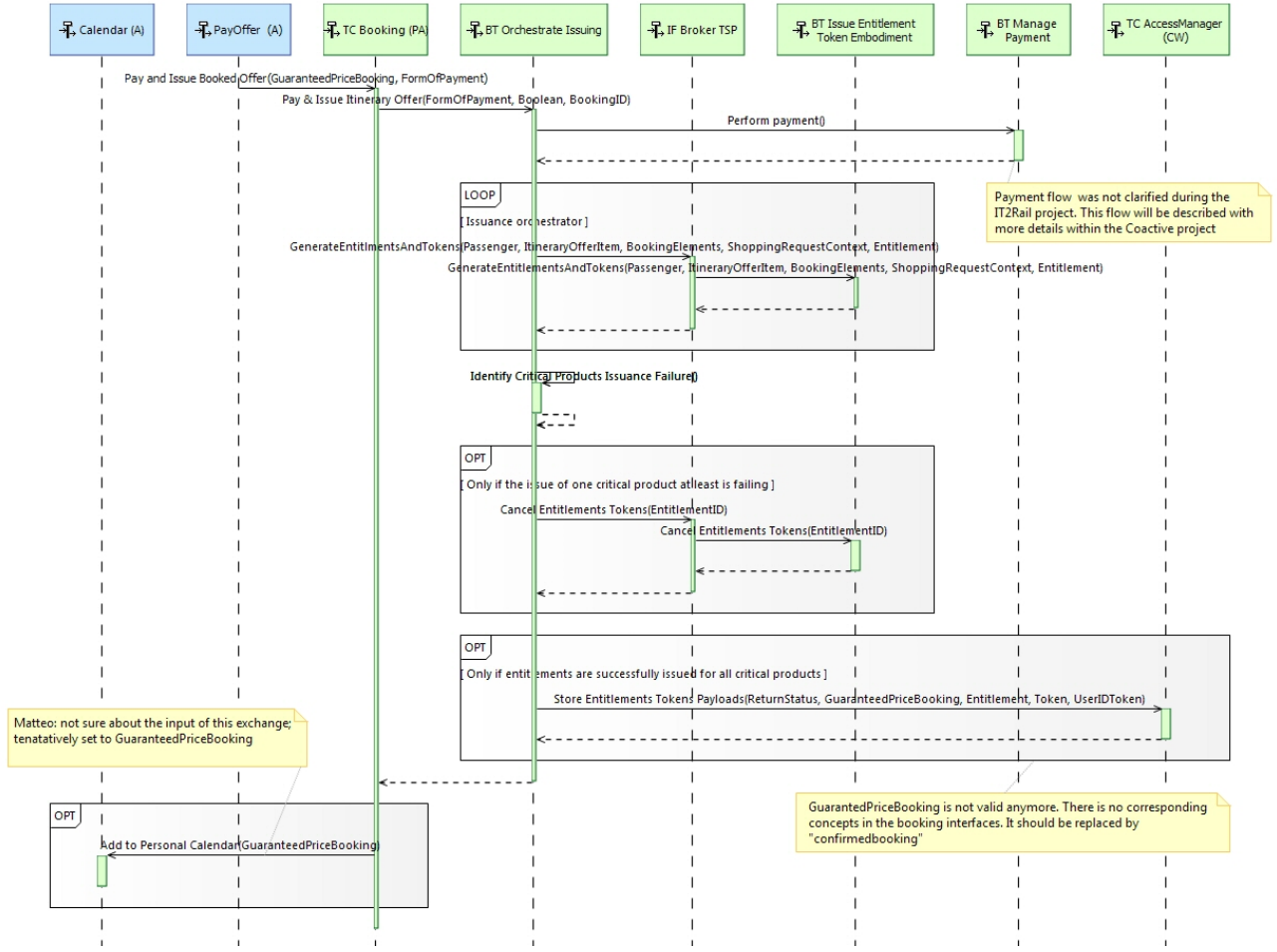


Figure 9: Payment and issuing scenario

Model reference: *[FS]L SBPI - Pay and deliver Entitlement, Token and Embodiment - BT (step 3) - FREL version: Diagram updated (*SVN Version: 814)

This scenario exposes the following interfaces of the Booking and Ticketing functions.

- The *BT Orchestrate Issuing* logical function exposes the following interfaces:
  - Pay & Issue Valued Itinerary Offer: This interface orchestrates the two-step Payment process with the issuing process, and optionally if needed the cancel process. This interface is specified in chapter 1.1 and the data used are described in chapter 7.1.
  - Identify Critical Products Issuance Failure: This interface identifies if at least one critical product exists among the list of failures during payment and issuance. The response of this interface will allow the orchestrator to trigger the cancellation of the already issued products.
- The *BT Issue Entitlement Token Embodiment* logical function exposes the following interfaces:
  - Generate Entitlements Tokens: This interface generates the requested travels rights (entitlements and tokens). This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - Issue Entitlements: This interface issues the entitlements. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - Issue Tokens: This interface generates then issues the Tokens. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - Cancel Entitlements Tokens: This interface cancels a complete issued entitlement, in case a critical product issuance failure has been previously identified. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - Cancel Entitlements: This interface cancels entitlements. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.
  - Cancel Tokens: This interface cancels Tokens. This interface is specified in chapter 7.2 and the data used are described in chapter 7.1.

6.1.5 Validate Token

Validate Token is a capability provided to Travellers and Ticket Controlling Organisations enabling the Traveller to demonstrate - and Ticketing Controlling Organisation to verify - that Travellers are indeed in possession of the right to use the mobility services on one or more travel episodes that compose their Travel.

This right is represented by a token stored with the Traveller's Travel Companion wallet, which is processed electronically.

The Validate Token capability includes specialised realisations as described in the following diagram (Figure 10):



Figure 10: Validate Token Use Cases diagram

Model reference: *[CRB]L Validate Token - detail (*SVN Version: 814)

The “Validate Token” capability realisation includes “Validate Token (Preparation) TC”, itself including “Validate Token – Load Tapping Module TC”, detailed in the Travel Companion specification document (WP5). The following specifications will concern the “Validated Token –B&T” capability realisation.

The following sequence of activities is performed in the realisation of the Validate Token – B&T capability

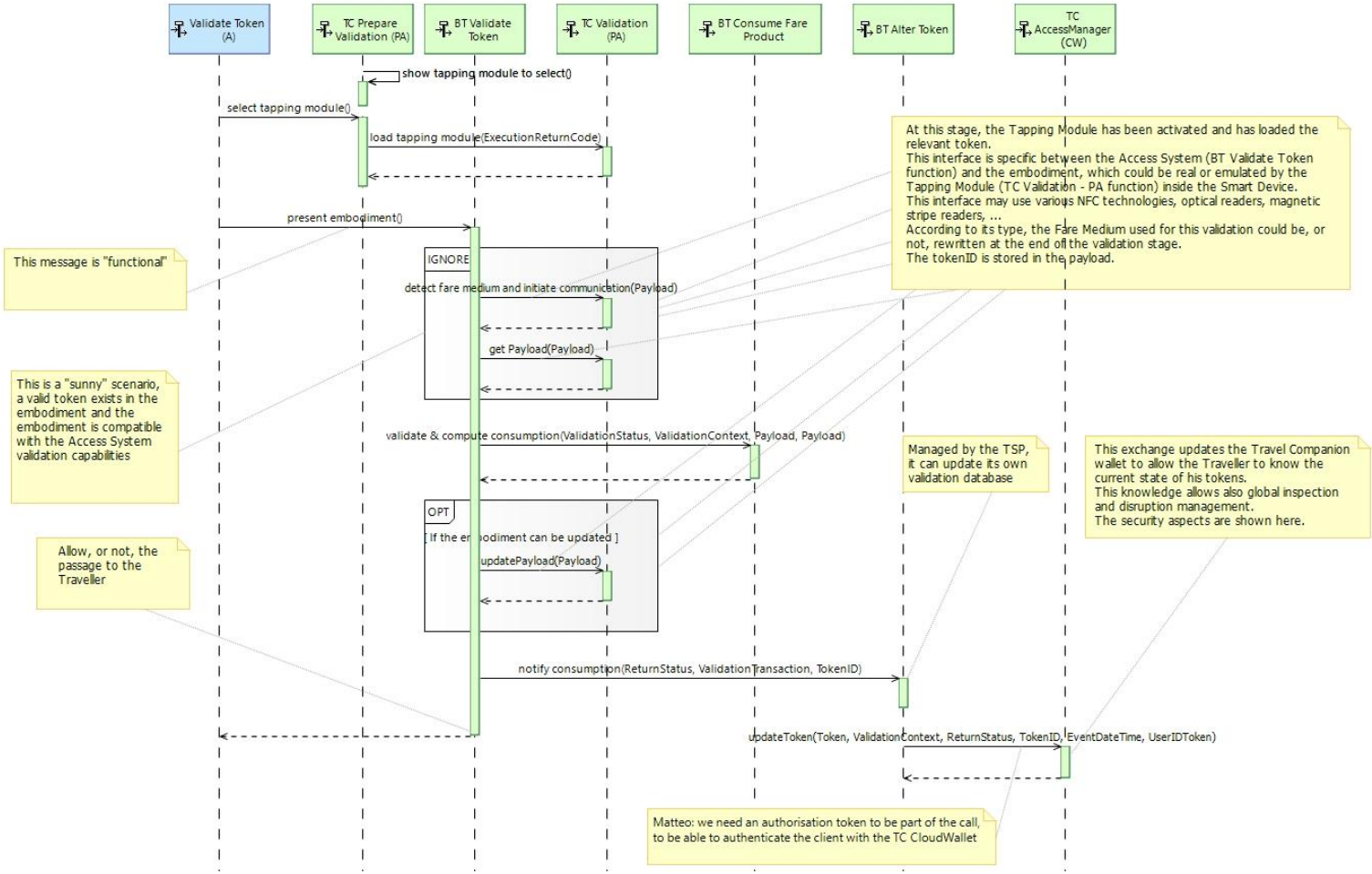


Figure 11: Validate Token B&T

Model reference: *[FS]L Validate Token with an access system - BT - FREL version: Diagram updated (*SVN Version: 814)

Booking and Ticketing functions and components involved in Validate Token B&T are described in the following figure (Figure 12):

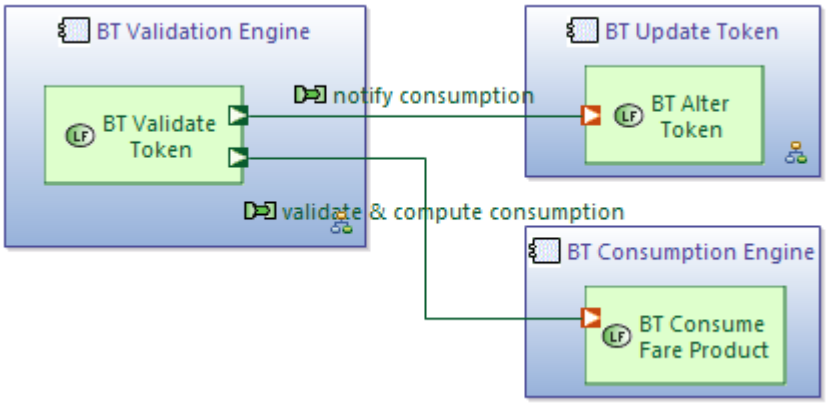


Figure 12: B&T Validation functions

Booking and Ticketing Functions involved in Validate Token B&T capability realisation			
Logical function	**Description**	**Functional Exchange**	**Component**
BT Validate Token	The function identifies the Token and processes its associated payload based on the Token embodiment presented by the Traveller. Different embodiments of a Token are possible to support interaction with heterogeneous, existing Token and payload reading equipment and technologies. The function creates an instance of the specific validation context associated with the validation equipment (e.g. timestamp, location) and submits it with the Token's payload for product consumption.	Validate and compute consumption to BT Consume Fare Product Notify Consumption to BT Alter Token	BT Validation Engine This component is typically an existing equipment specific to a Ticketing Controlling Organisation

Booking and Ticketing Functions involved in Validate Token B&T capability realisation			
Logical function	**Description**	**Functional Exchange**	**Component**
BT Consume Fare Product	This function uses the Token payload and the validation context to apply specific validation rules and criteria, returning an updated payload and the validation result. Implementation of this function is company specific and not part of the current specification.	Validate and compute consumption from BT Validate Token	BT Consumption Engine
BT Alter Token	This function updates the Token with the updated payload and interacts with the Travel Companion to propagate the Token update to the Travel Companion's tokens wallet.	Notify Consumption from BT ValidateToken	BT Update Token

6.1.6 Inspect the validity of the ticket

This capacity is not in the scope of IT²RAIL.

6.1.7 Manage Business Analytics

The Business Analytics system requests the TSP issuing data in order to do statistics on the habits and feedbacks of the Travellers (see Figure 13).

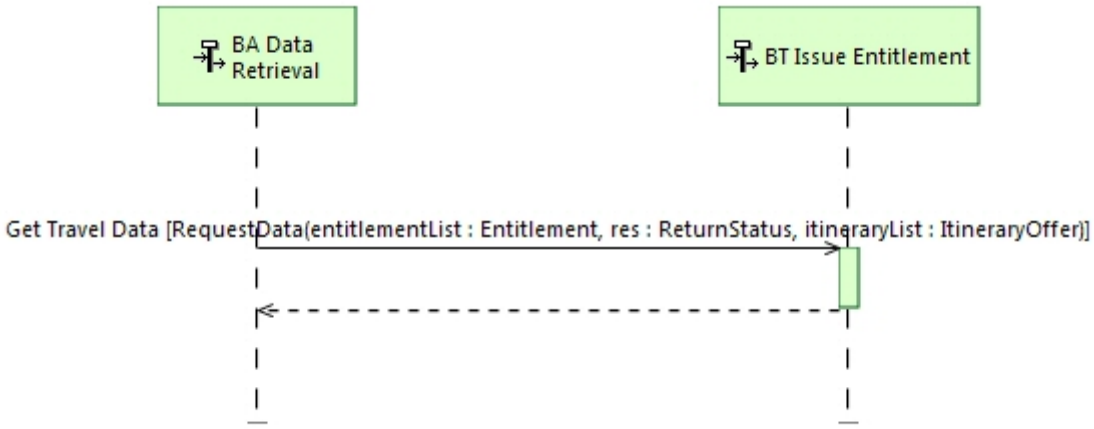


Figure 13: Business Analytics issuing data requesting scenario

Model reference: *[FS]L Analyse Travel Data - B&T (*SVN Version: 814)

7. EXCHANGES

This chapter presents the interfaces used in the previously described functional scenarios.

This chapter consists of two sub-chapters:

- The first one will allow mutualisation and consistency of the data model between all the interfaces. The used data model must be integrated in the IT²RAIL ontology.
- The second one will describe all the interfaces details.

7.1 DATA MODEL

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.

7.1.1 Booking data model

This data model is used by BT Orchestrate Booking and BT Book Inventory functions (Figure 14).

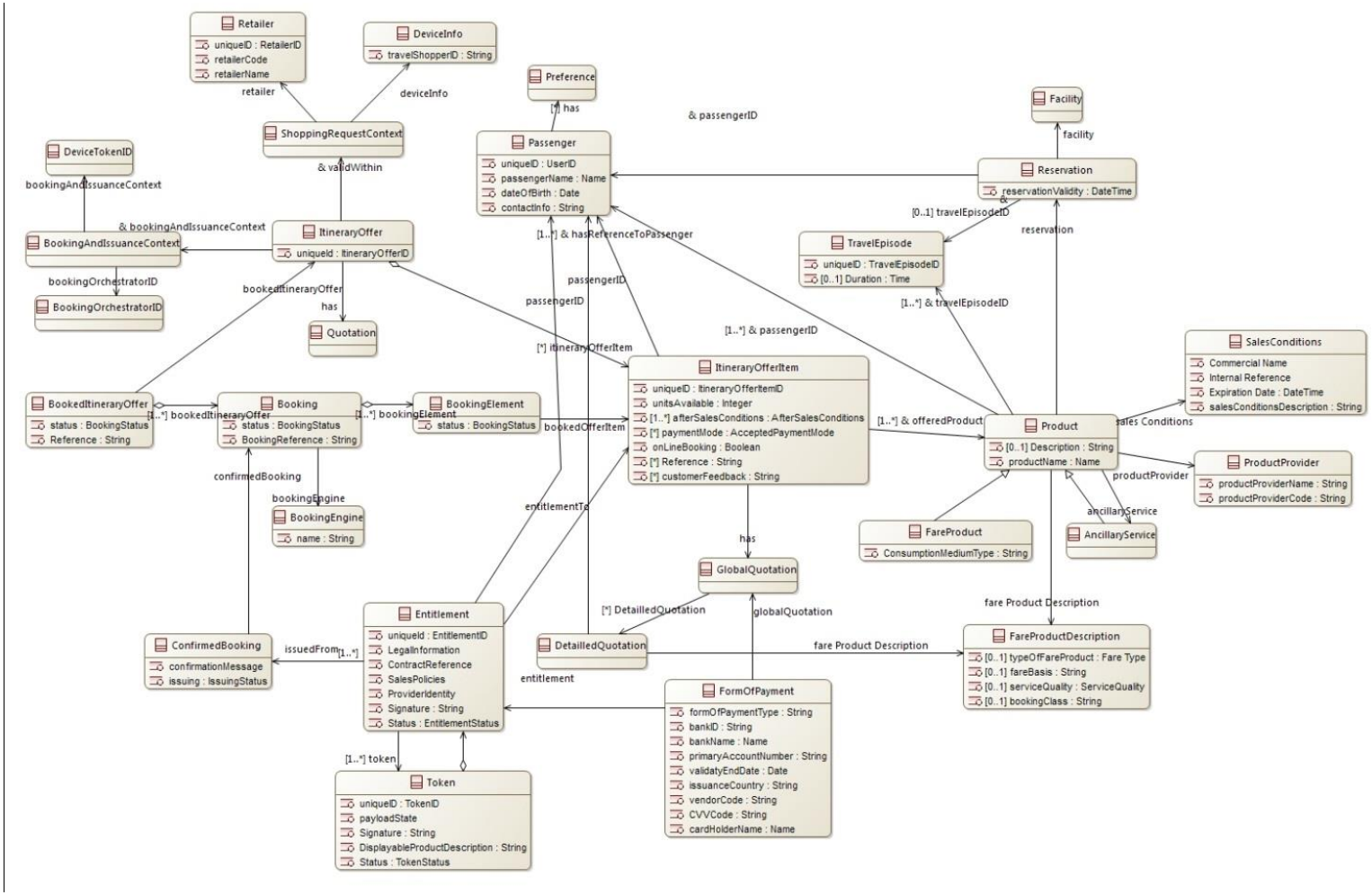


Figure 14: Booking engines data model

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).

7.1.2 Shopping data model

This high level class diagram shows the main classes and their relationships (Figure 15).

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

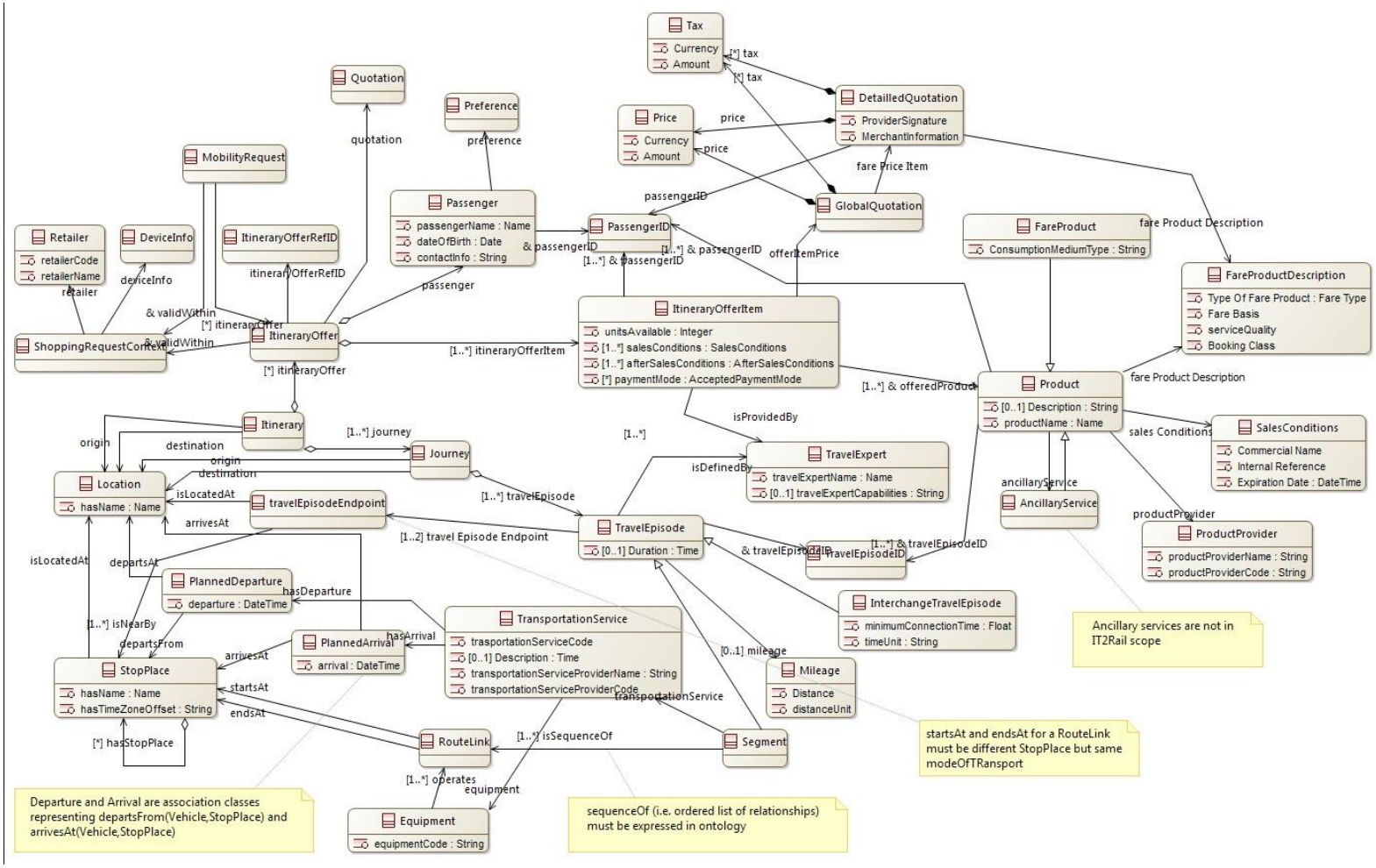


Figure 15: Shopping scenario data model

Model reference: *[CDB]L Shopping - Itinerary Offer (*SVN Version: 814)

This data model shows the organisation 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 customisation 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.

7.1.3 Fulfilment data model

This high level class diagram shows the main classes and their relationships (Figure 16).

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

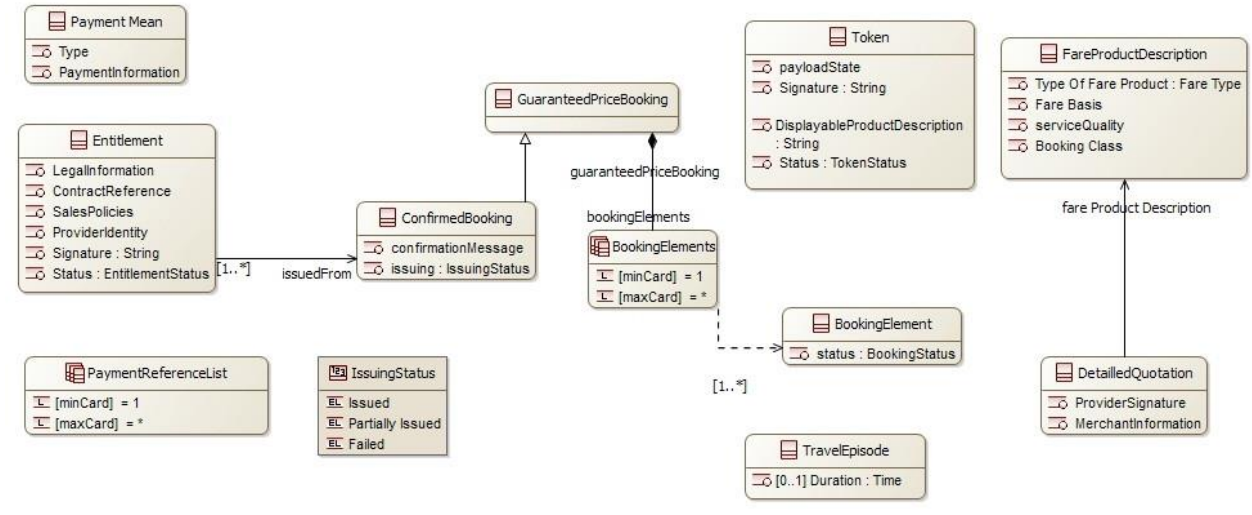


Figure 16: FulfillmentEngine data model

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).

7.1.4 Payment data model

This high level class diagram shows the main classes and their relationships (Figure 17).

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

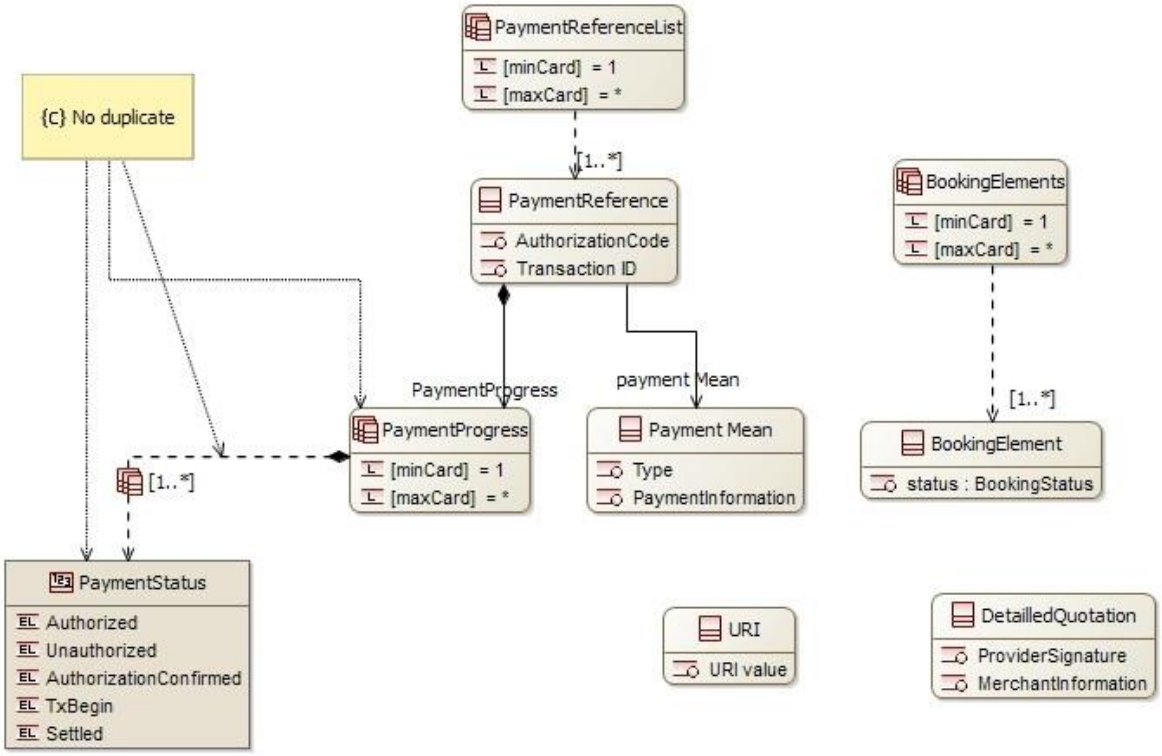


Figure 17: Payment data model

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.

7.1.5 Validation data model

The following schema details the model used by Booking and Ticketing for the Entitlement and Token that Travel Service Provider issue (Figure 18).

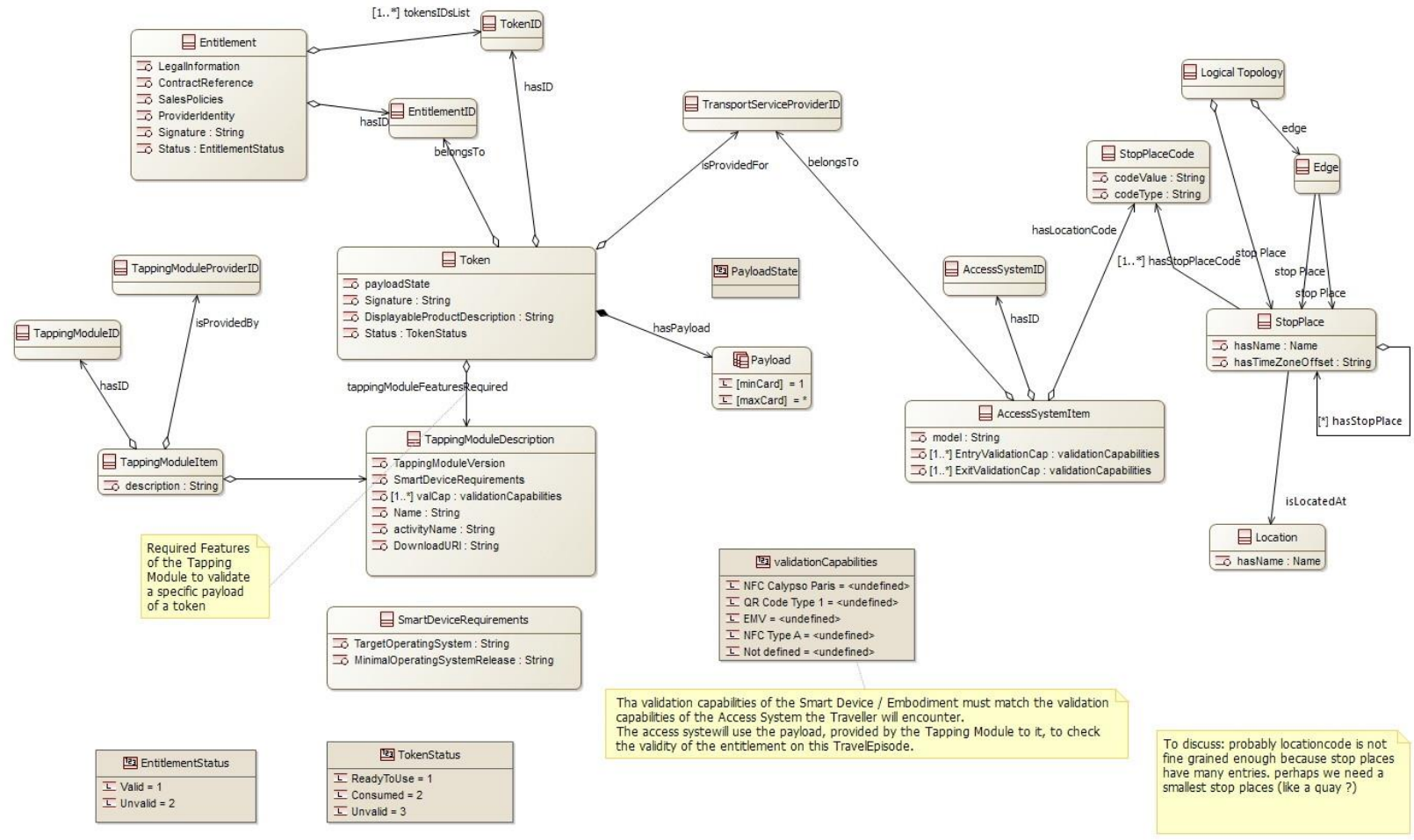


Figure 18: Entitlement & Token data model

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.

7.1.6 Fare product Data model

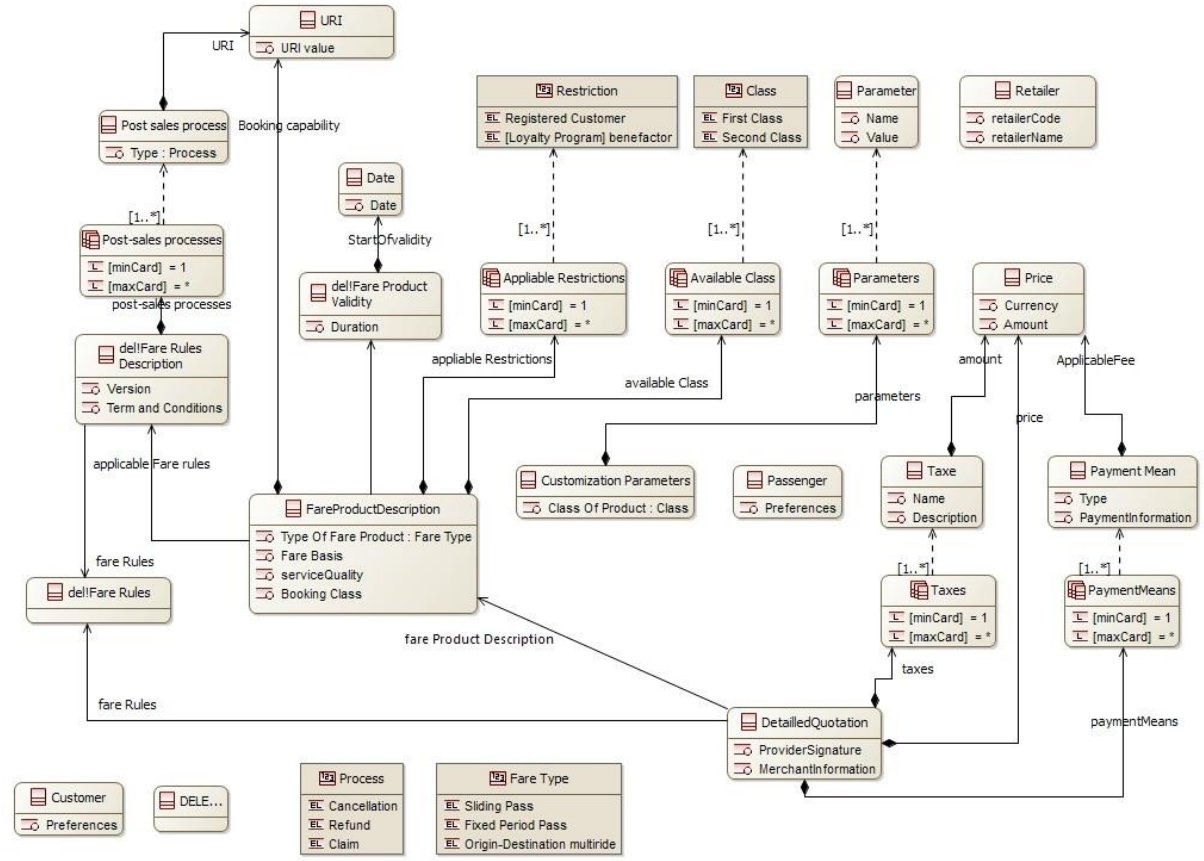


Figure 19: Fare product data model

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 19) 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.

7.2 INTERFACES

This chapter specifies the interfaces of the functions of the Booking and Ticketing, with their inputs and outputs.

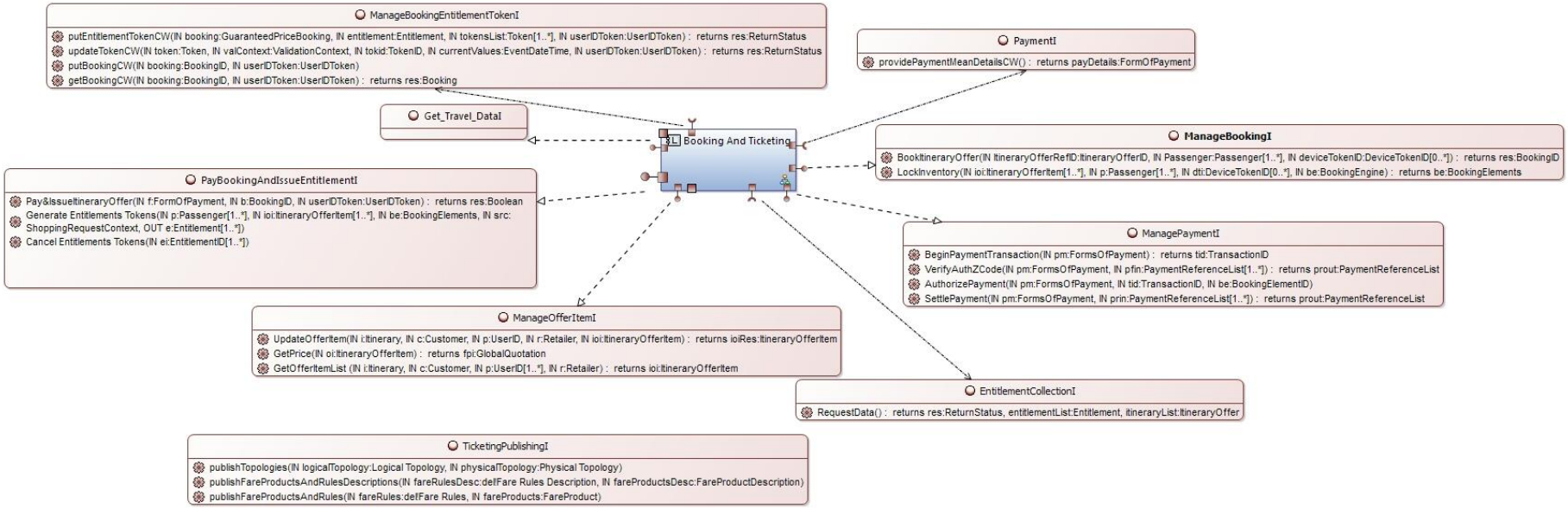


Figure 20: Booking and Ticketing required and provided interfaces

Model reference: *[CDI] BT - Contextual Component Detailed Interfaces (*SVN Version: 814)

The previous diagram (Figure 20) presents the external interfaces provided by Booking and Ticketing systems.

7.2.1 Manage Entitlement Token Interface

This interface is exposed by the BT Manage Token functions (Figure 21).

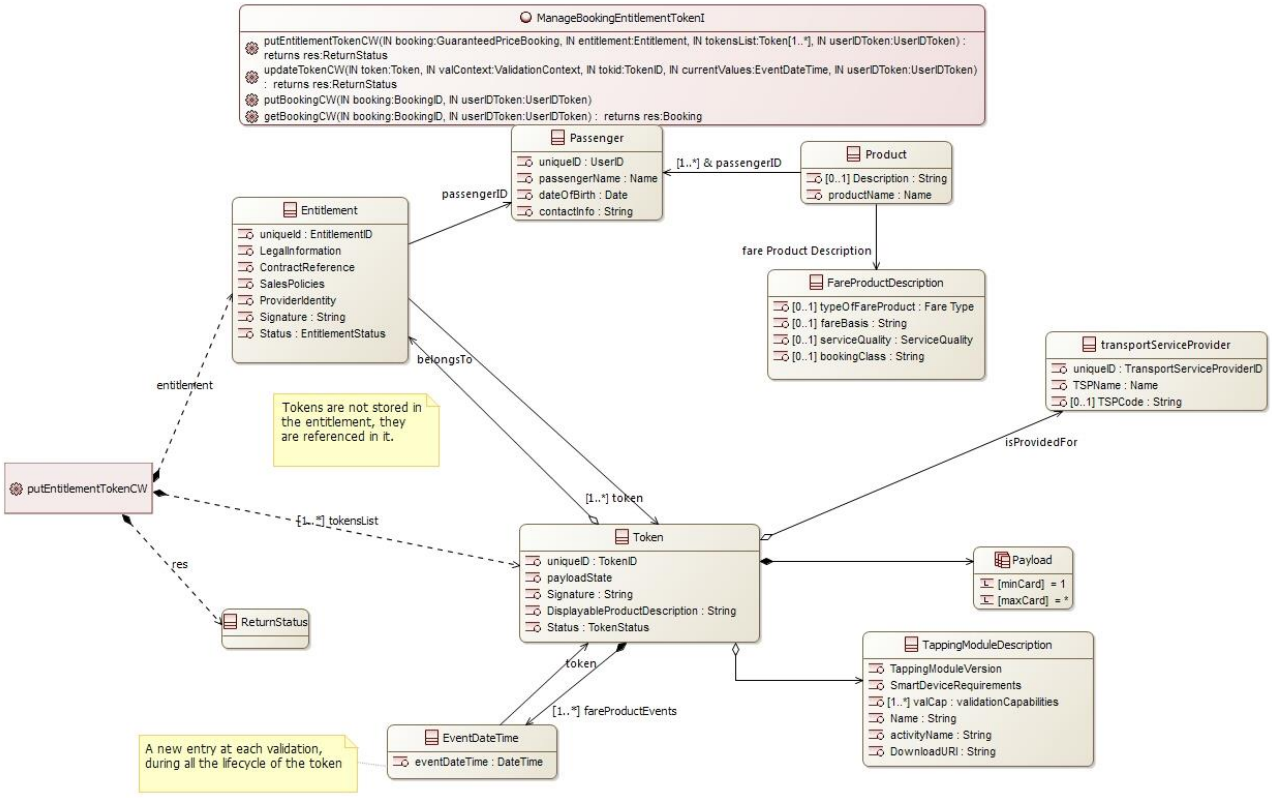


Figure 21: Manage Entitlement interface

Model reference: *[CDB]L Entitlement Token - EI Data and Interface* (SVN Version: 814)

ManageBookingEntitlementTokenI:putEntitlementTokenCW

Interface ID:	3.01		
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		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
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		

ManageBookingEntitlementTokenI:updateTokenCW

Interface ID:	3.02		
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		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
Preconditions:	None		
Postconditions:	None		
Request / Input	Token	M	The expression of an entitlement in an involved organisation 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		

ManageBookingEntitlementTokenI:putBookingCW

Interface ID:	3.03		
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		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
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		

ManageBookingEntitlementTokenI:getBookingCW

Interface ID:	3.04		
Interface Name:	getBookingCW		
Purpose of the Interface	This interface should be invoked by the Travel Companion to retrieve the booking from the CW		
Requestor:	Booking & Ticketing		
Provider	Travel Companion		
Description:	This interface should be invoked by the Travel Companion to retrieve the booking in the CW		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
Preconditions:	A booking stored before in the CW		
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	Booking	M	The booking element
Exceptions:	None		
Notes and Issues:	None		

7.2.2 Manage Booking Interface

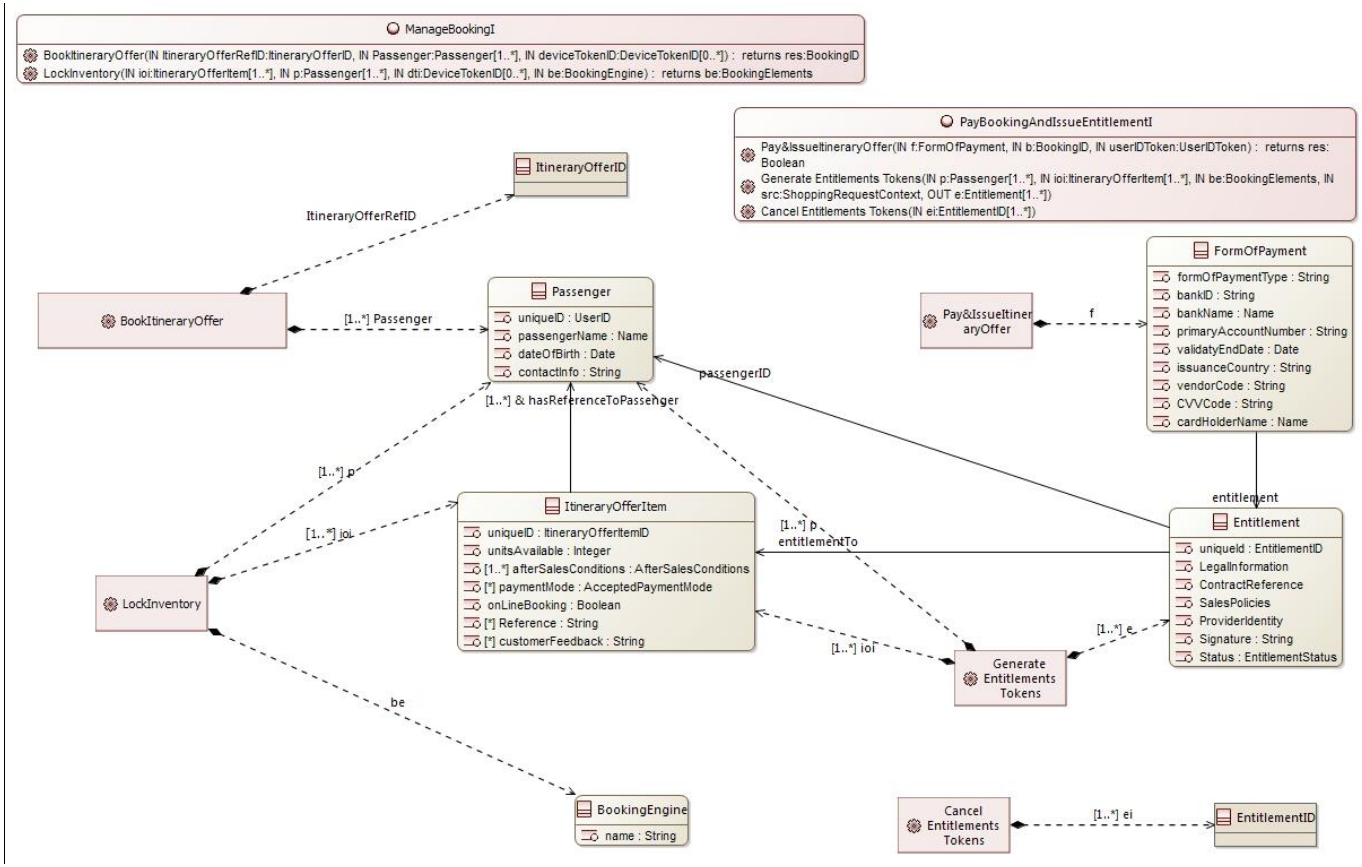


Figure 22: Manage Booking Interface

Model reference: *[CDB]L BT Booking - EI Data Interface - FREL version: Diagram updated (*SVN Version: 814)

ManageBookingI:BookItineraryOffer

Interface ID:	3.05		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
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		

ManageBookingI:LockInventory

Interface ID:	3.06		
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.		
Impact to CREL	Partial		
Impact to AREL	Partial		
Impact to FREL	Complete		
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		

PayBookingAndIssueEntitlementI:Pay&IssueItineraryOffer

Interface ID:	3.07		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
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		

PayBookingAndIssueEntitlementI:Generate Entitlements Tokens

Interface ID:	3.08		
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.		
Impact to CREL	Partial		
Impact to AREL	Partial		
Impact to FREL	Complete		
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.		

PayBookingAndIssueEntitlementI: Cancel Entitlements Tokens

Interface ID:	3.09		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA		
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		

7.2.3 Validate Token B&T Interfaces

The following interfaces are requested or provided for the realisation of the functional exchanges involved in Validate Token B&T (Figure 23).

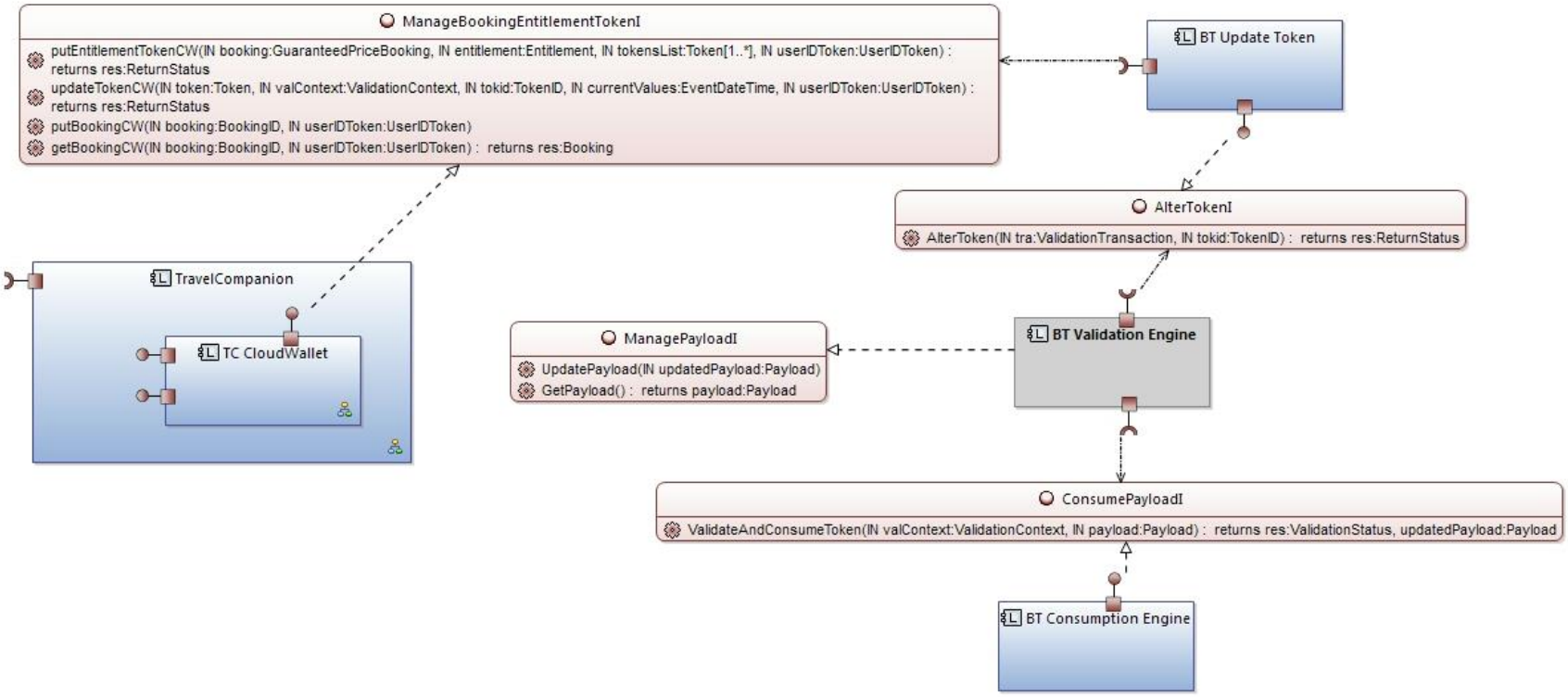


Figure 23: Validate Token Interfaces

Model reference: *[IDB] BT Validation Engine - Interfaces Diagram Blank] (*SVN Version: 814)

A. ManagePayloadI

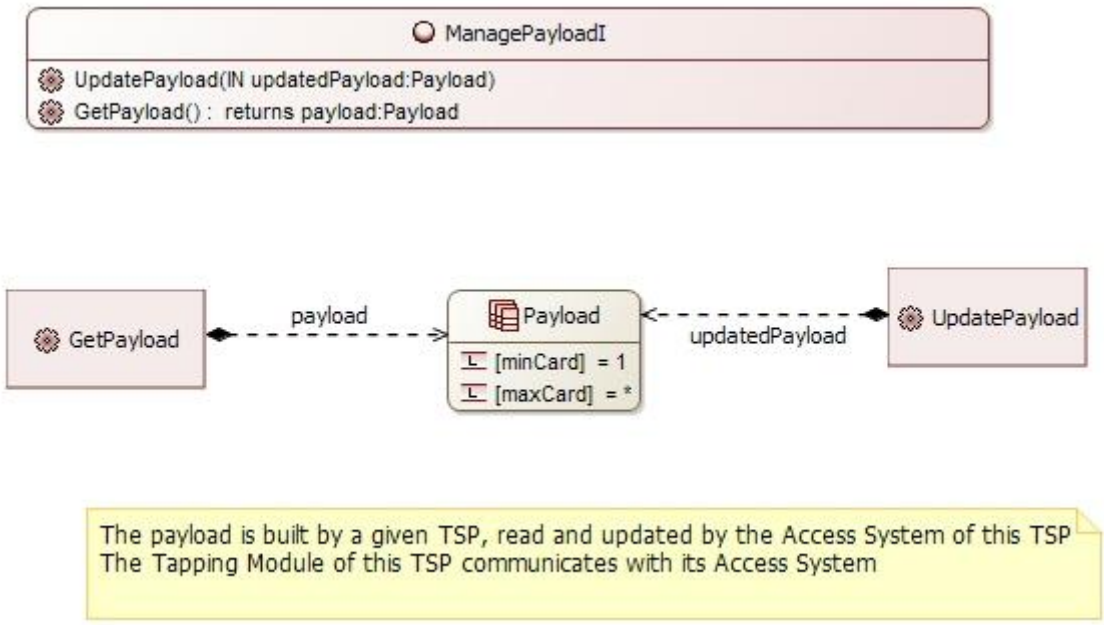


Figure 24: Manage Payload Interface

Model reference: *[CDB]L Payload - EI Data Interface (*SVN Version: 814)

ManagePayloadI:GetPayload

Interface ID:	3.10		
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		
Impact to CREL	Partial		
Impact to AREL	Complete		
Impact to FREL	Complete		
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		

ManagePayloadI:UpdatePayload

Interface ID:	3.11		
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		
Impact to CREL	Partial		
Impact to AREL	Complete		
Impact to FREL	Complete		
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		

B. ConsumePayloadI

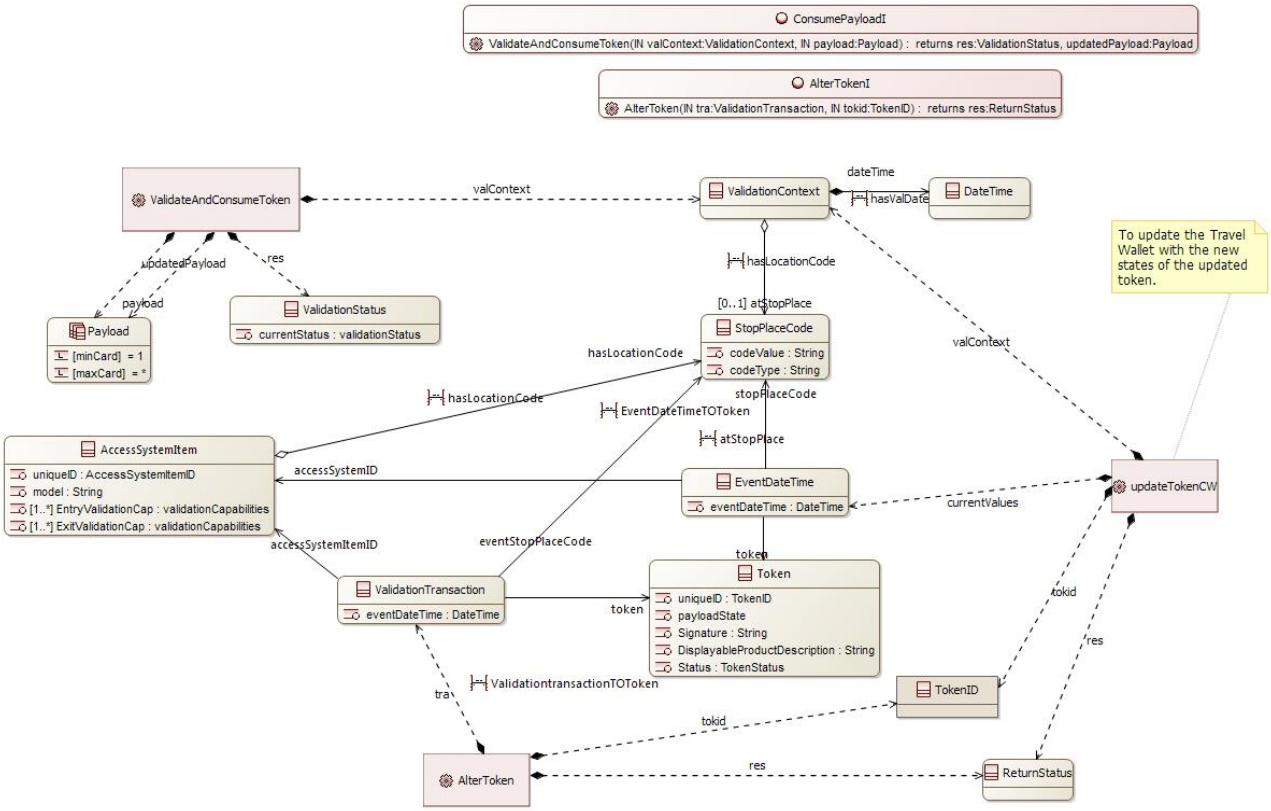


Figure 25: Consume and alter Token Interfaces

Model reference: *[CDB]L BT Validation - EI Data Interface (*SVN Version: 814)

ConsumePayloadI:ValidateAndConsumeToken

Interface ID:	3.12		
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		
Impact to CREL	Partial		
Impact to AREL	Complete		
Impact to FREL	Complete		
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		

AlterTokenI:AlterToken

Interface ID:	3.13		
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		
Impact to CREL	Partial		
Impact to AREL	Complete		
Impact to FREL	Complete		
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		

C. ManageBookingEntitlementTokenI

Same interface as described in §7.2.1

7.2.4 Publish ticketing data Interface

It is provided as an example.

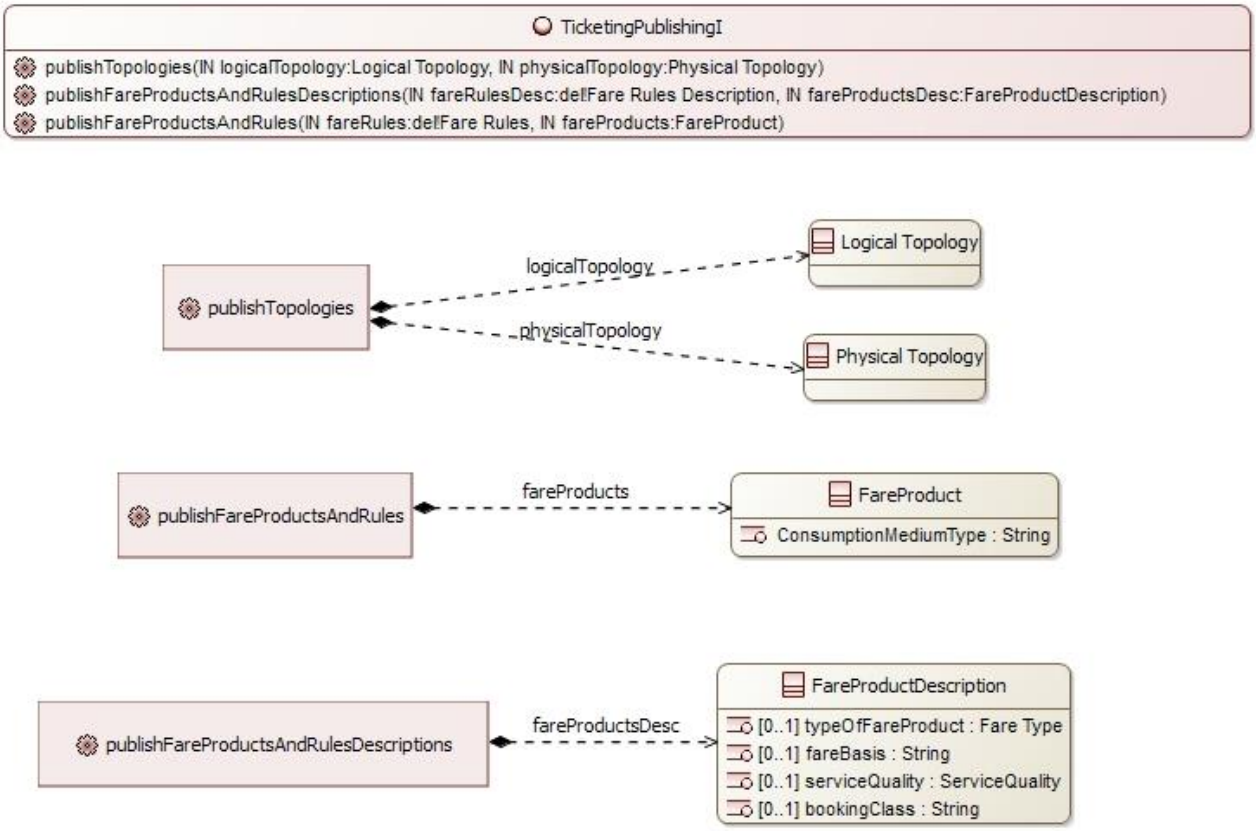


Figure 26: Publish ticketing data interface

Model reference: *[CDB]L BT Publish ticketing data - EI Data and interface (*SVN Version: 814)

TicketingPublishingI:PublishTopologies

It is provided for information purposes.

Interface ID:	3.14		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

TicketingPublishingI:PublishFareProductsAndRulesDescriptions

It is provided for information purposes.

Interface ID:	3.15		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

TicketingPublishingI:PublishFareProductsAndRules

It is provided for information purposes.

Interface ID:	3.16		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

7.2.5 Manage Offer Item Interface

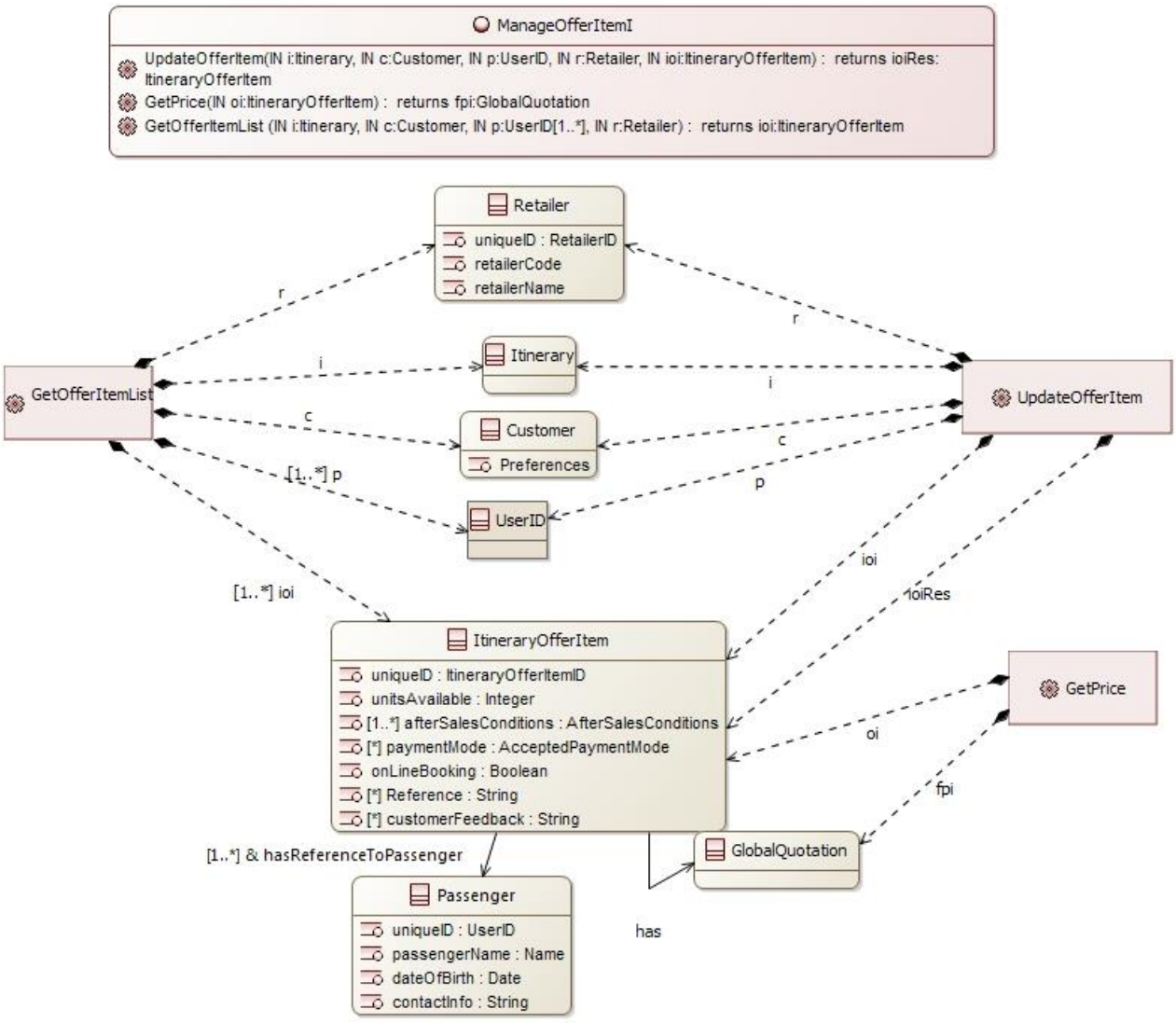


Figure 27: Manage Offer Item Interface

Model reference: *[CDB]L SBPI - Classes - EI Data and interfaces (*SVN Version: 814)

ManageOfferItemI:GetOfferItemList

Interface ID:	3.17		
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.		
Impact to CREL	Partial		
Impact to AREL	Partial		
Impact to FREL	Complete		
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.		

ManageOfferItemI:UpdateOfferItem

Interface ID:	3.18		
Interface Name:	UpdateOfferItem		
Purpose of the Interface	This interface should be invoked to update an OfferItem with an altered CustomisationParameters.		
Requestor:	Shopping		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to update an ItineraryOfferItem with an altered CustomisationParameters. No other alteration of an ItineraryOfferItem should be allowed.		
Impact to CREL	Partial		
Impact to AREL	Partial		
Impact to FREL	Complete		
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.		

ManageOfferItemI:GetPrice

Interface ID:	3.19		
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		
Impact to CREL	Partial		
Impact to AREL	Partial		
Impact to FREL	Complete		
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.		

7.2.6 Manage Payment Interface

It is provided as an example for the entitlement issuing implementation.

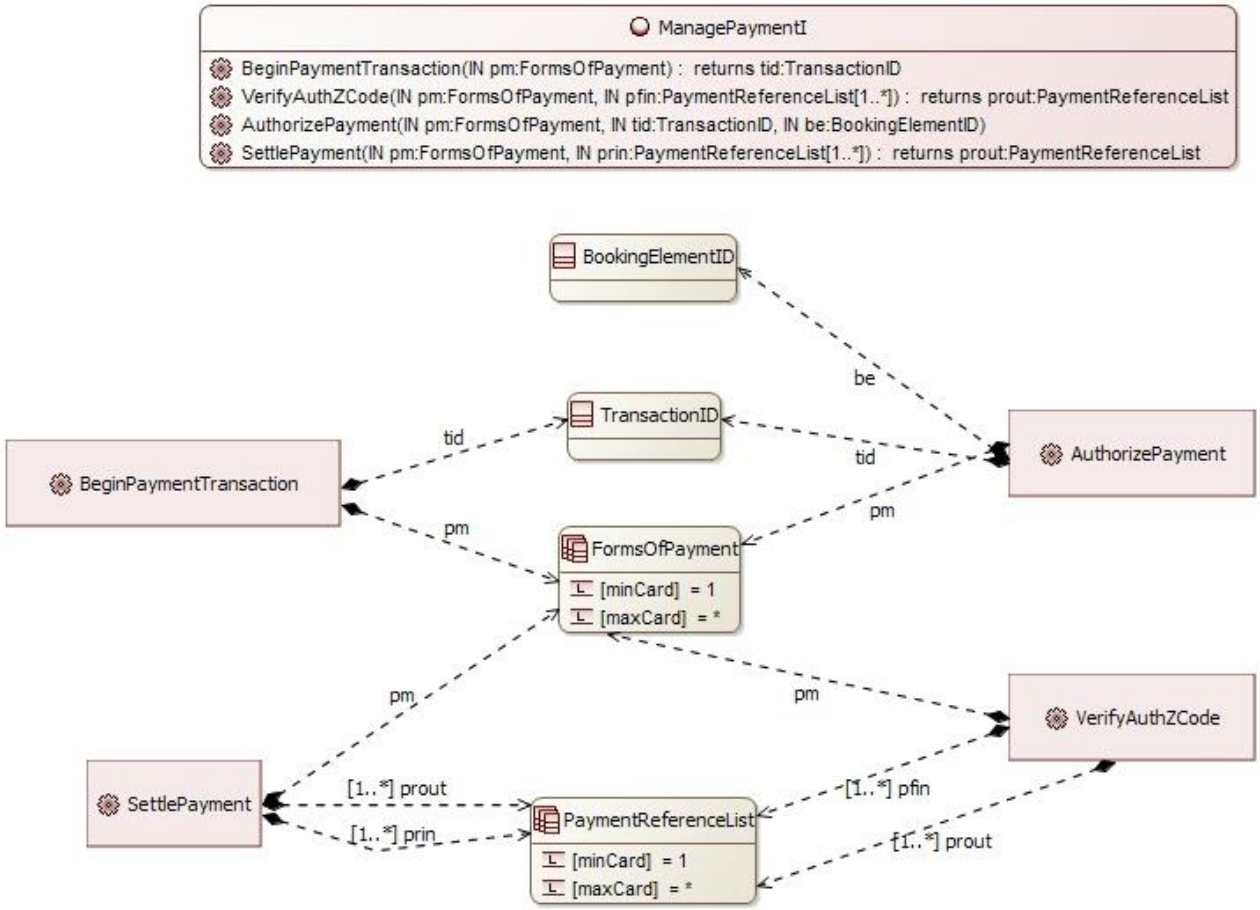


Figure 28: Manage Payment Interface

Model reference: *[CDB]L BT Payment Transactions - EI Data Interfaces (*SVN Version: 814)

ManagePaymentI:BeginPaymentTransaction

It is provided for information purposes.

Interface ID:	3.20		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

ManagePaymentI:VerifyAuthZCode

It is provided for information purposes.

Interface ID:	3.22		
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 authorisation code provided to issue the entitlement or token is legitimate and still valid.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

ManagePaymentI:AuthorizePayment

It is provided for information purposes.

Interface ID:	3.21		
Interface Name:	AuthorizePayment		
Purpose of the Interface	This interface should be invoked to request the authorisation of the payment in the two-step payment process.		
Requestor:	Booking & Ticketing		
Provider	Booking & Ticketing		
Description:	This interface should be invoked to request the authorisation 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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

ManagePaymentI:SettlePayment

It is provided for information purposes.

Interface ID:	3.23		
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.		
Impact to CREL	NA		
Impact to AREL	NA		
Impact to FREL	NA (Information purpose)		
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		

7.2.7 Digest of interfaces provided by Booking and Ticketing

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. The purpose column is briefly describing the purpose of the operation.

Name	Purpose
ManageBookingEntitlementTokenI:putEntitlementTokenCW	This interface should be invoked by the Travel Companion to store the entitlement token in the CW
ManageBookingEntitlementTokenI:updateTokenCW	This interface should be invoked by the Travel Companion to update the stored token in the CW
ManageBookingEntitlementTokenI:putBookingCW	This interface should be invoked by the Travel Companion to store the booking in the CW
ManageBookingEntitlementTokenI:getBookingCW	This interface should be invoked by the Travel Companion to retrieve the booking from the CW
ManageBookingI:BooktineraryOffer	This interface should be invoked to book an itinerary offer.
ManageBookingI:LockInventory	This interface should be invoked to book a capacity constrained offer item
PayBookingAndIssueEntitlementI:Pay&IssueItinerary Offer	This interface should be invoked to pay and issue an itinerary offer.
PayBookingAndIssueEntitlementI:Generate Entitlements Tokens	This interface should be invoked to issue the Entitlements and Tokens of a GuaranteedPriceBooking.
PayBookingAndIssueEntitlementI: Cancel Entitlements Tokens	This interface should be invoked to cancel an entitlement, and therefore all associated tokens.
ManagePayloadI:GetPayload	The interface implements the interaction between the Travel Companion and the BT Validation Engine
ManagePayloadI:UpdatePayload	The interface implements the interaction between the Travel Companion and the BT Validation Engine
ConsumePayloadI:ValidateAndConsumeToken	The interface implements the interaction between the BT Validation Engine and the BT Consumption Engine

Name	Purpose
AlterTokenI:AlterToken	The interface implements the interaction between the BT Validation Engine and the BT Update Token
TicketingPublishingI:PublishTopologies	This interface provides the current ticketing logical and physical topologies.
TicketingPublishingI:PublishFareProductsAndRulesDescriptions	This interface should be invoked to provide the current description of fare products and fare rules.
TicketingPublishingI:PublishFareProductsAndRules	This interface should be invoked to provide the current list of fare products and fare rules.
ManageOfferItemI:GetOfferItemList	This interface should be invoked to retrieve the list of available offer items.
ManageOfferItemI:UpdateOfferItem	This interface should be invoked to update an OfferItem with an altered CustomisationParameters.
ManageOfferItemI:GetPrice	This interface should be invoked to price an available itinerary offer item.
ManagePaymentI:BeginPaymentTransaction	This interface should be invoked to start the two-step payment process.
ManagePaymentI:VerifyAuthZCode	This interface updates the PaymentReferenceList with information from a trusted PaymentEngine.
ManagePaymentI:AuthorizePayment	This interface should be invoked to request the authorisation of the payment in the two-step payment process.
ManagePaymentI:SettlePayment	This interface is to settle the payment.

8. ANNEX

8.1 SYSTEM-WIDE DECISIONS

This chapter recalls all Booking and Ticketing decisions taken by the Booking and Ticketing partners.

N°	Decisions	Status
1	Group of Passengers are not considered in the scope of IT²RAIL Booking and Ticketing.	Done
2	Booking and Ticketing PaymentEngine demonstration should only include Orchestration, actual settlement should be simulated.	Done
3	Valued Itinerary Offer objects should contain retailer information for Booking/issuing: • Booking Orchestrator Reference. • PaymentEngine Reference (Orchestrator). • Retailer fee (Fare Price).	Done
4	All functions that will only deploy in a Travel Service Provider environment should be renamed to reflect this for clarity.	To do IP4
5	ItineraryOfferItem objects must include Merchant Payment Engine reference. Such input is provided by TSP Travel Expert Offer builder	To do IP4
6	Tokens are directly issued to the Travel Wallet by the TSP FulfillmentEngine-Issuer.	Done
7	ID of customer and authentication token for wallet access must be passed through the sequence of calls.	Done
8	At issuing time: the token payload delivery should be handle by the TSP FulfillmentEngine-Issuer and accepted by the corresponding Tapping Module.	Done
9	Tapping module may be downloaded when selecting an offer to book & pay. Tapping module and payment module must appear within the offer so the Travel Companion App can download the proper modules.	Done
10	Capability of the smart device should be included in the mobility request.	Done

Table 2: System-wide decisions

End-of-document