INTERNATIONAL UNION
OF RAILWAYS

unity, solidarity, universality

OSDM - An Introduction

December 2020

Clemens Gantert, Chairman OSDM
Andreas Schlapbach, Tech Lead FSM & Chairman OSDM

1. Introduction

The aims of the **Open Sales and Distribution Model (OSDM)** are twofold:

1. to substantially **simplify the booking process for customers** of rail trips and,
2. to **lower complexity and distribution costs** for distributors and railway carriers.

Thus, distributors and railways joint forces and developed a state-of-the-art distribution API and fare exchange model.

Specification is Open Source and available to everybody:

https://unioninternationalcheminsdefer.github.io/OSDM/

1. OSDM – Vision

OSDM
Open Sale & Distribution Model

Powerful combination

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

Easy distribution

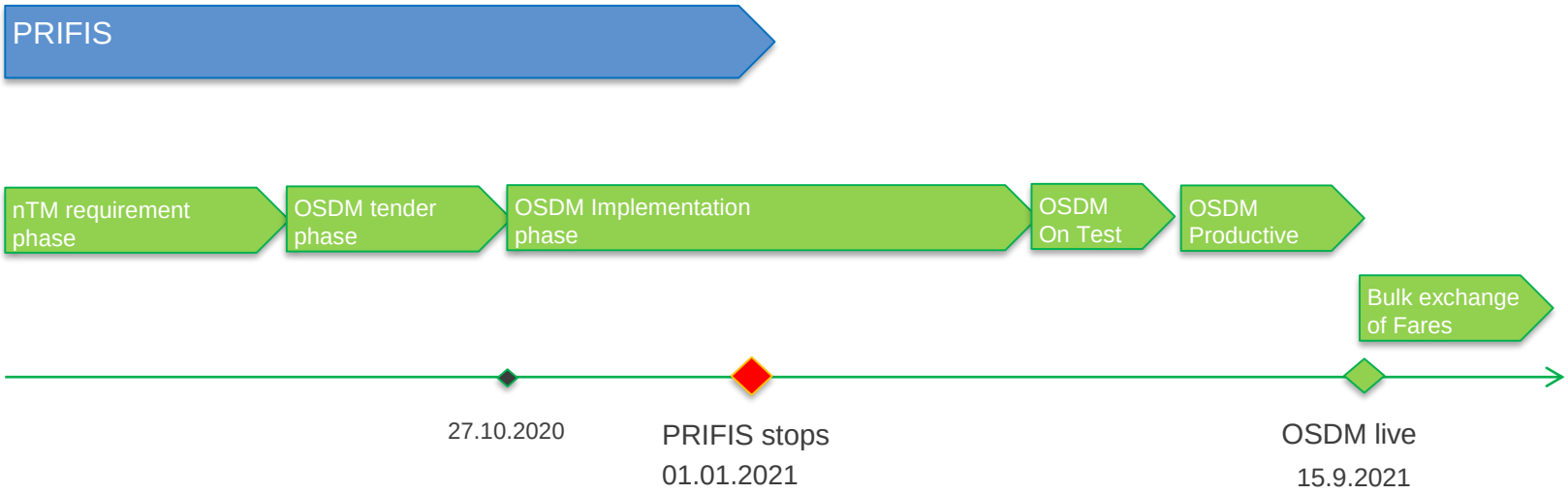
- It must be **easily possible to distribute** existing and new products.
- Easily possible means two things:

  1. Firstly, for a **customer** it must be easily possible to find and book and – if needed – refund a booking.

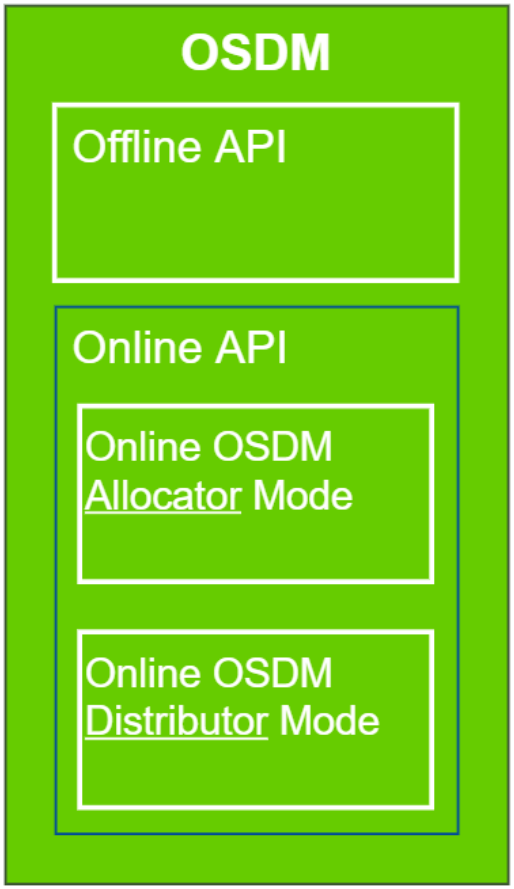
  2. Secondly, for the **rail sector as a whole** the complexity of distribution must be reduced to save costs both for development as well as distribution.

1. PRIFIS is End of Live

Replacement: OSDM-Offline



2. Overview of OSDM



Replaces PRIFIS

New Online Distribution and Carrier Interface

2. OSDM Online API

OSDM
Open Sale & Distribution Model

- The OSDM is a **light-weight API** consisting of a set of **micro services**.
- It is modelled in YAML, fully supporting the **REST** paradigm.
- The specification is completely **Open Source** and developed in an **agile manner**.

2. Supported Processes

OSDM
Open Sale & Distribution Model

Sales and Distribution processes:

- Searching for trips
- Getting offers
- Booking an offer
- Confirmation of the booking
- Fulfillment of the booking

After-sale processes:

- Getting a refund/exchange offers
- Booking a refund/exchange offer
- Fulfillment of the booking

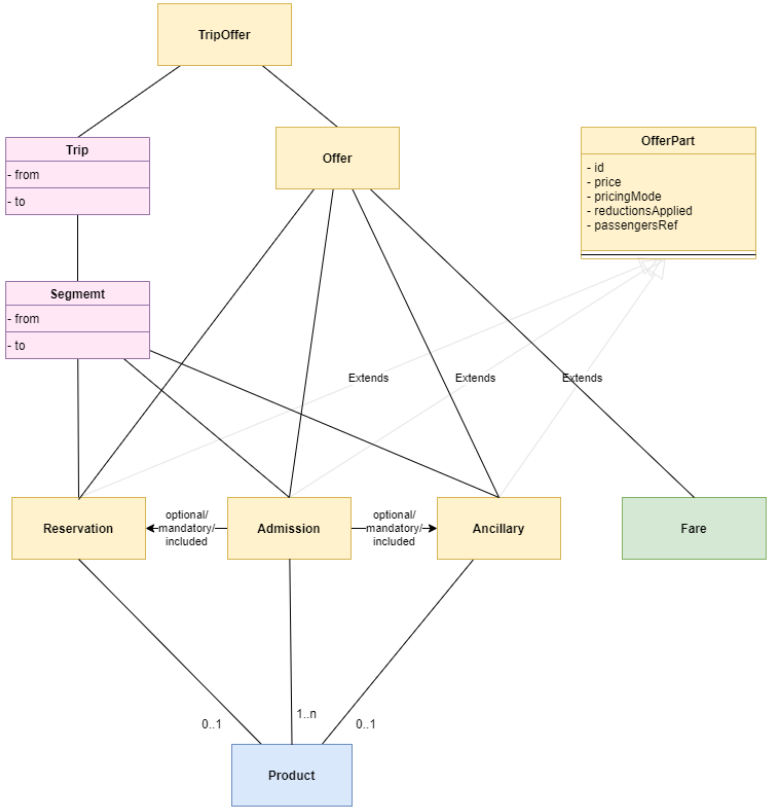
2. Supported Products

OSDM
Open Sale & Distribution Model

Offers consist of the following parts:

- **Admissions** (aka. tickets),
- **Reservations**,
- **Ancillaries**
- **Fares** (priced segments)

The offers are linked by **optional / mandatory / included relations** to an **overall offer.**



UIC 90918-10 - OSDM 0.9.7 OAS3

Specifications for the OSDM API standard. The OSDM specification supports two modes of operation: Distributor Mode and Allocator Mode. The API works the same in both mode, except that in allocator mode the API also returns fare information. The OSDM specification combines the nTM as well as the FSM initiative into one aligned API interface.

Nicolas Selleslagh - Website
Send email to Nicolas Selleslagh
Apache 2.0

Servers
https://virtserver.swaggerhub.com/Real-IT/FSM_MVP/1.0.0 - fsm dummy server

Filter by tag

Trips resources to search for trips and locations

GET	/locations	The GET locations service returns matching locations for a given match value based on MatchType.
GET	/locations/{locationId}	The GET locations service retrieves a location element.
GET	/trips/{tripId}	The GET trips service retrieves a trip element representing a train travel made of one or more segments. Depending on the embed either references or full location definitions will be returned
GET	/trips-collection /{tripsCollectionId}	The GET trips-collection service returns a collection of different trips for a specified origin and destination (and via). The unique codes of the origin and destination can be resolved using the locations service.
POST	/trips-collection	The GET tripsCollection service returns different trips for a specified origin and destination (and via). The unique codes of the origin and destination can be resolved using the locations service.

Offers resources to get bookable offers

GET	/trip-offers-collection /{tripOffersCollectionId}	The GET tripOffersCollection service returns different offers and trips for a specified origin and destination (and via). The unique codes of the origin and destination can be resolved using the locations service.
POST	/trip-offers-collection	The POST tripsOffersCollection service returns different trip offers for a specified origin and destination (and via) locations.
GET	/trip-offers/{tripOfferId}	The GET trip-offers service returns different offers for a specified trip offer id.
GET	/offers/{offerId}	The GET offers service returns the offer with the requested included sub resources in a given state.
POST	/offers	Get offers not linked to trips based on key words or regional information (e.g. for passes or seasonal tickets)

Offer Parts resources to manipulate parts of an offer consisting of, e.g., admissions, reservations, or ancillaries

GET	/offers/{offerId}/admissions/{admissionId}	The GET admission service returns the information on the admission with the id provided.
GET	/offers/{offerId}/reservations/{reservationId}	The GET reservation service returns the information on the reservation with the id provided.
PATCH	/offers/{offerId}/reservations /{reservationId}	The PATCH of the place selections service allows updating reservation elements, to update the desired selection of places and to retrieve selectable places for a graphical display
GET	/offers/{offerId}/ancillaries/{ancillaryId}	The GET ancillaries service returns the information on the ancillary with the id provided.
GET	/offers/{offerId}/fares/{fareId}	The GET fares service returns the information on the ancillary with the id provided.
PATCH	/offers/{offerId}/fares /{fareId}	The PATCH of the place selections service allows updating reservation elements, to update the desired selection of places and to retrieve selectable places for a graphical display

Bookings
resources to manipulate bookings

- POST /bookings/search — The POST bookings search service allows to search for and find bookings based on a set of criteria. It returns a list of booking ids.
- POST /bookings — The POST bookings create service allows to create a booking based on a previously requested offer. The offer id is provided in the query.
- GET /bookings/{bookingId} — The GET booking service returns the information on the booking with the id provided.
- DELETE /bookings/{bookingId} — The DELETE booking service deletes the booking with the id provided. It is only possible before the booking is confirmed.
- PATCH /bookings/{bookingId} — The PATCH bookings service triggers the confirmation of the booking

Passengers
resources to manipulate a passenger's information at every stage of the flow

- GET /offers/{offerId}/passengers/{passengerId} — The GET passenger service returns the passenger's information.
- PATCH /offers/{offerId}/passengers/{passengerId} — The PATCH passenger service allows updating a passenger's information.
- GET /bookings/{bookingId}/passengers/{passengerId} — The GET passenger service returns the passenger's information.
- PATCH /bookings/{bookingId}/passengers/{passengerId} — The PATCH passenger service allows updating a passenger's information.

Products
resources to retrieve products information on one or more products

- GET /products/{productId} — The GET product service returns the information on products with the id provided.

Fulfillments
resources to retrieve fulfillments, e.g. tickets

- POST /bookings/{bookingId}/fulfillments — The POST fulfillments service triggers the fulfillment of the booking based on the information currently contained in the provided booking.
- PATCH /bookings/{bookingId}/fulfillments — The PATCH fulfillments service allows the effective fulfillment of one ore more confirmed fulfillment.
- GET /fulfillments/{fulfillmentId} — The GET ticket service returns the information on the ticket with the id provided.

RefundOffers
resources to get and accept a refund offer

- POST /bookings/{bookingId}/refundOffers — The POST refundOffers initiates a refund process by creating a RefundOffer resource containing the needed information on the potential operation. One refund offer can then be accepted via a PATCH, deleted or left to die at the end of its lifetime.
- GET /bookings/{bookingId}/refundOffers/{refundOfferId} — The GET refundOffer service returns the refund offer with the id provided.
- PATCH /bookings/{bookingId}/refundOffers/{refundOfferId} — The PATCH refundOffers action allows to accept and confirm a refund operation.
- DELETE /bookings/{bookingId}/refundOffers/{refundOfferId} — The DELETE refundOffers action allows to cancel a refundOffer without waiting for expiry.

ExchangeOffers
resources to get and accept an exchange offer

- POST /bookings/{bookingId}/exchangeOffers — *** NOT-MVP *** The POST exchangeOffers initiates an exchange process by creating a ExchangeOffer resource containing the needed information on the potential operation. The proposition can then be accepted via a PATCH, deleted or left to die at the end of its lifetime.
- PATCH /bookings/{bookingId}/exchangeOffers/{exchangeOfferId} — *** NOT-MVP *** The PATCH exchangeOffers action allows to accept and confirm an exchange operation.

CoachLayouts
resources to get the layout of a coach

- GET /coachLayouts — *** NOT-MVP ***Retrieve the coach layout description needed for graphical reservation

2. OSDM – Roles
Definitions

The **OSDM Online API** is split between roles participants can take:

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

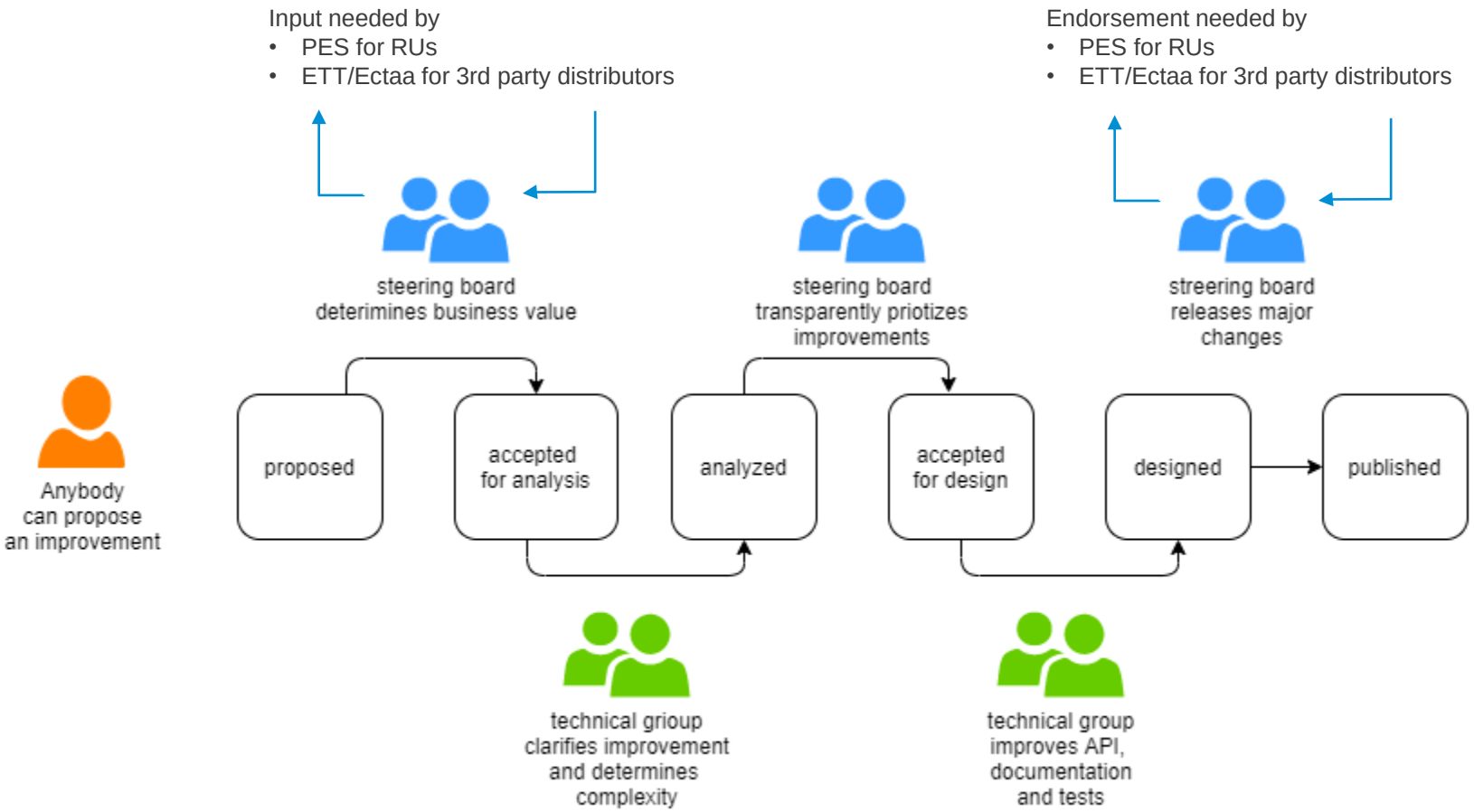
A RU or system provider can support more than one role.

3. Organization of OSDM Working Group
Evolution process

Basic Ideas

- A specification is **product**.
- We have a **backlog of improvements** where all parties can contribute.
- An improvement addresses a need and solves a problem, thus producing **business value.**
- The **OSDM executive committee** prioritizes the backlog based on the value for the railway customer and the railway sector as a whole.
- The OSDM executive committee is a **fair representation of the parties** involved, thus railways and distributors and others.
- The OSDM technical working group **designs** improvements to the standard. To stay focused, the work in progress should not be larger than 3.
- The OSDM technical working group taking special care not to break existing implementations, thus **securing investments** made by all the parties.

3. Organization of OSDM Working Group
Evolution process



3. Organization of OSDM Working Group

Aim: incremental and steady improvement

Technical Aspect

- Versioning major/minor versions following semantic versioning.

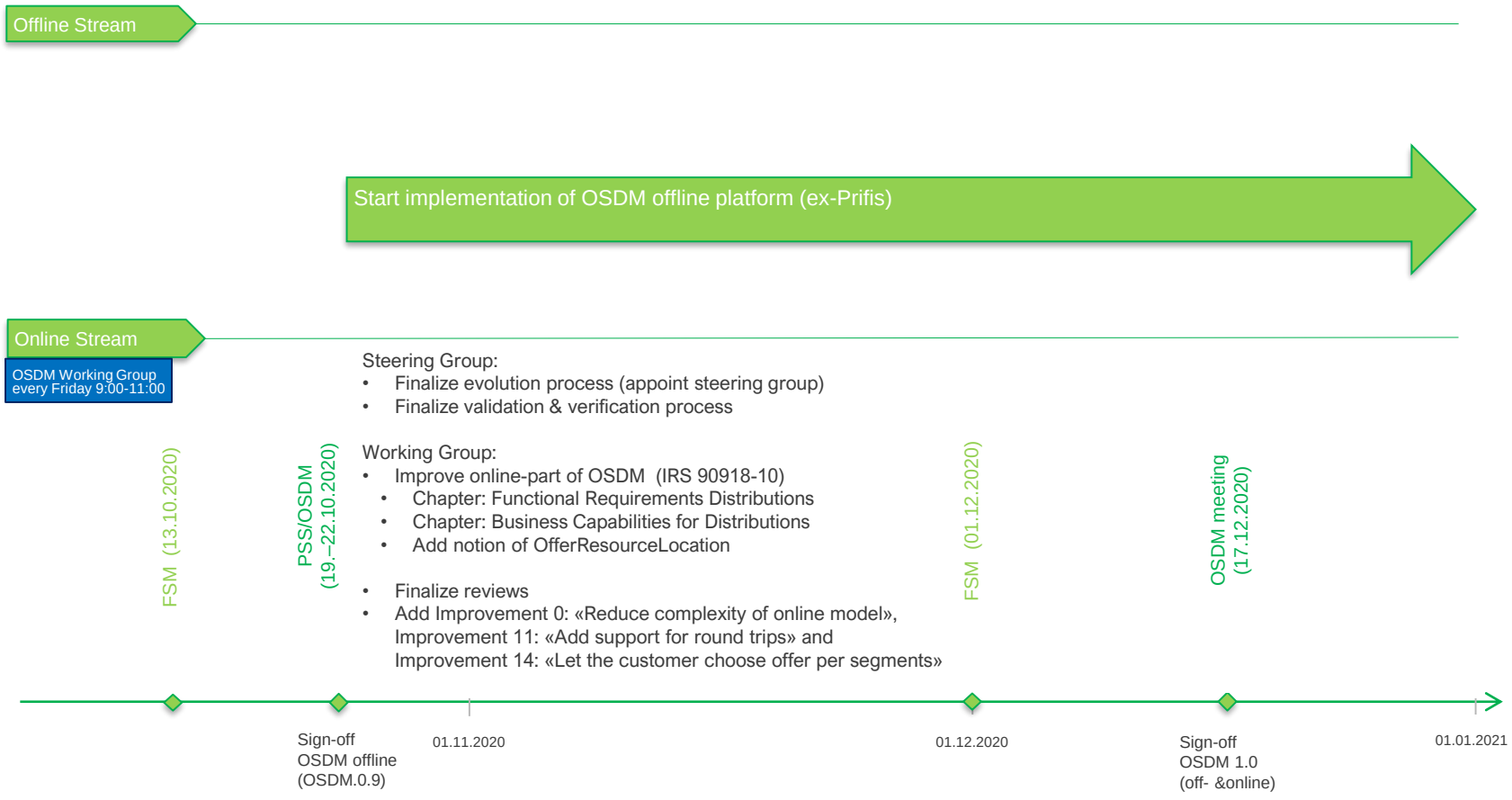
Organizational Aspect

- annual publication of an approved IRS 90918-10 version by UIC
- Intermediary minor changes published within the work group

Maintenance Process

- New requirements can be added by all participants in OSDM

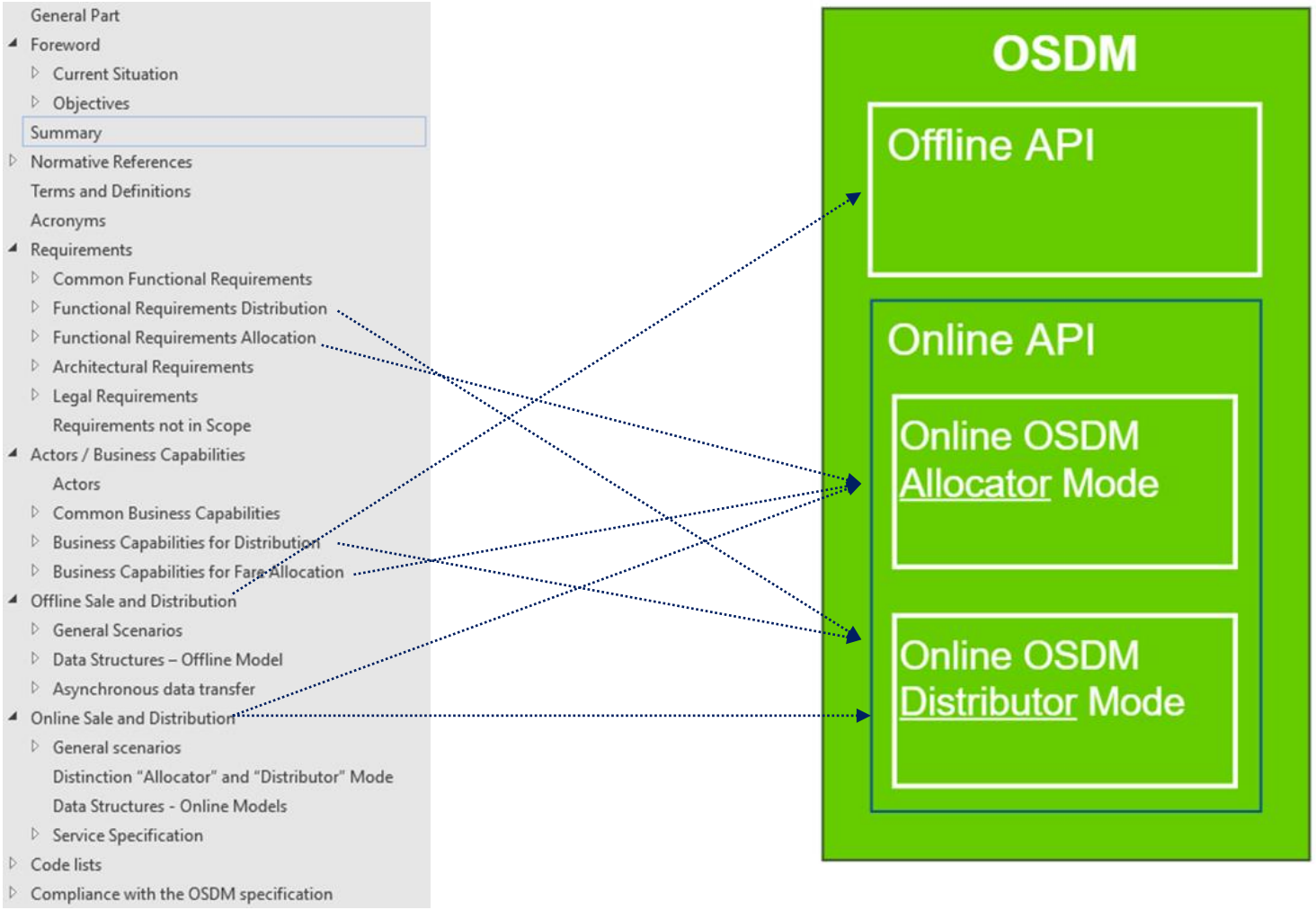
4. Schedule for Specification Work



4. Mapping Business Capabilities to Specification

- The specification needs to layout **the business requirements/capabilities** first before documenting the technical implementation
- In order to reflect the functional requirements leading to API they need be made explicit and documented.
- This gives the **business representatives** the possibility to be ascertained that their needs a reflected and addressed.
- Additionally it helps preventing repeating discussions already had.

4. Mapping Capabilites to API



5. List of Enhancements

The API is continuously enhanced

OSDM
Open Sale & Distribution Model

#	Improvements name	Owner	State
0	Reduce complexity of online model	ETT	Published in Version 1.0
11	Add support for round trips	DB / Sqills Bayer Ralf (IT-SCP-MVD-ETS - Extern) Nieuwerth Johan (IT-SCP-MVD-ETS - Extern)	Published in Version 1.0
1	Offer-based combination	Amadeus Parmigiani Elisa (IT-SCP-MVD-ETS - Extern)	Accepted for analysis
2	Need for stateless processes	Sqills Nieuwerth Johan (IT-SCP-MVD-ETS - Extern)	Accepted for analysis
3	Support for partial refund / exchange	Trenitalia Stefano Turri	proposed
4	Align /locations and /trips to Transmodel (OJP)	SBB Hirzel Peter (IT-SCP-BP-VTP)	Proposed
5	Add a notion of a fee	DB Bayer Ralf (IT-SCP-MVD-ETS - Extern	Accepted for analysis
6	Support for new mobility services based on pay-after-use	UIC:Door2Door	proposed
7	Add full support for persons with reduced mobility	CIT Jan Vavra	proposed
8	Add real-time support	DB Bayer Ralf (IT-SCP-MVD-ETS - Extern)	proposed
9	Add pdf binary in fulfillment messages	DB Dorsch Jessica (IT-SCP-MVD-ETS - Extern)	Accepted for analysis
10	Add support for offers only spanning part of a journey	SBB Schlapbach Andreas (IT-SCP-MVD-ETS)	proposed
12	Add support for URT ticket	Jan Vavra	Published in Version 1.0
13	Add support for mixed forms of fulfillment types (mix value paper and non-value paper)	Jan Vavra	proposed
14	Let the customer choose offers per segments independently	Trenitalia (Luca/Sandara)	Published in Version 1.0