Blueprint for an

Application Programming Interface (API)

from Transport Operator to MaaS Provider

a first technical milestone towards Mobility as a Service

Version 1.0
16-05-2019

Table of Contents

1 Introduction

Mobility as a Service (MaaS) offers the opportunity for seamless travel using any available transport mode. For MaaS to be successful, transport operators are required to share their transport services and availability of their assets in a digital form. To facilitate MaaS providers and thus enable the deployment of MaaS services, transport operators are also required to standardize the digital form to facilitate access to their information.

2 Goal of this document

In this Blueprint for an Application Programming Interface (API) from Transport Operators (TO) to MaaS Providers (MP) we look into the necessary functional requirements for the interoperability between transport operators. The goal of this document is to:
- Define the necessary scope for full interoperability between transport operators for the deployment of MaaS services, always keeping the customer journey in mind to determine which calls are needed between MaaS-providers and Transport Operators.
- Define the necessary parameters and values to fulfill this scope
- Define the available parameters in various already available APIs and propose amendments where applicable.
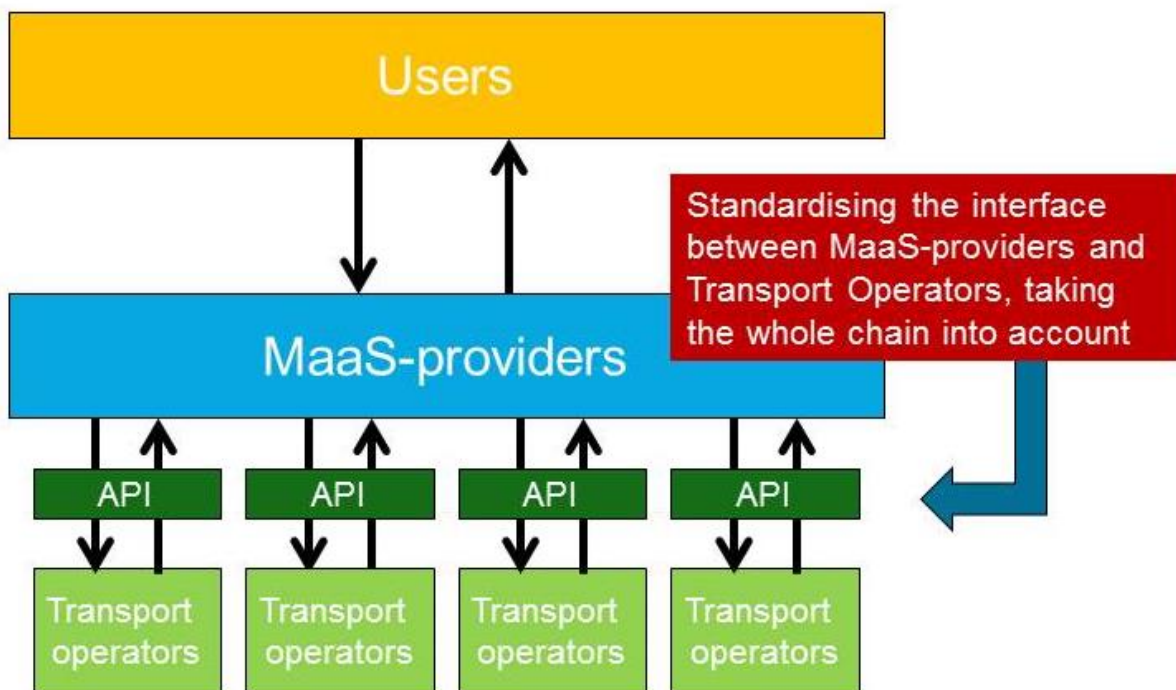
3 Who is involved

This document has been written to consolidate the work of:
- The technical working group of the 12 bikes haring operators involved in the 'deelfietsconvenant', the arrangement that requires interoperability between these transport operators to operate bike sharing schemes in the main municipalities of the Netherlands.

- The technical working group of 20+ on-demand car operators involved in deploying interoperable services to support MaaS-development in the Netherlands, including the eMaaS project team at the University of Twente.
- The 24 consortia that have signed the framework agreement of the Ministry of Infrastructure and Water Management in the Netherlands, who have reviewed the work starting from v0.7.
- The 65 participants in the precompetitive dialogue phase of the MaaS-Programme of the Ministry of Infrastructure and Water Management in the Netherlands, who have reviewed the work starting from v0.8.

4   What is in this version

- Version 1.0 has been updated after consultation with the precompetitive dialogue phase participants and comments from the bike sharing- and on-demand car operators.

A digital version of the necessary API calls is available for consultation in the link below (see also Table 1 in section 7).

https://app.swaggerhub.com/apis-docs/efel85/GTOAS/1.0.6

5   To be added after this version

In no particular order, the following developments are envisioned for future versions:
- Further (continuous) refinement after testing and implementation
- Specification of payment/ticketing-API from ETC
- Dynamic navigation to appropriate resources for API-calls listing asset characteristics, using HATEOAS or other.
- Standardization of support calls (e.g. flat tire is 001)
- Addition of infrastructure assets, e.g. EV charging stations
- Determine pagination and rate limiting requirements
- Define and describe OAuth scopes for flows
- Calls for feedback and ratings between users and transport operators

6 User Stories

To facilitate the definition of parameters and values that are required for full interoperability in MaaS, user stories have been defined from three different perspectives: the User, Transport Operator (TO) and MaaS Provider (MP).

By using these three perspectives, the chances are increased that all necessary functionalities for MaaS are taken into account. These functionalities can then be related to the necessary interface specifications between the Transport Operator and MaaS Provider. This document does not set up any requirements for the human-machine-interface (HMI) between Users and MaaS Providers.

Definitions within the User Stories

Definition	Meaning
API	Application Programming Interface, using REST-APIs as architectural style
User	Customer wanting to make a journey
Maas Provider	Provider of travel advice, information, booking and invoicing
Transport Operator	Owner of (any) transport assets. This can be a bike sharing or car sharing platform, public transport operators, taxi companies, ferry operators etc.
Required for MaaS	Yes = mandatory Conditional = mandatory for some operators Optional = mandatory for no operators
User	Competence = is the user able Conditions = is the user compliant Authentication = confirmation of identity/profile/token

Part 1: from a USER perspective

Item	1.1
Who	As a USER
What	I want to depart from STARTLOCATION and arrive at DESTINATION
Why	To define from where to where I need mobility services for my trip
Required For MaaS	STARTLOCATION=yes DESTINATION=conditional
Comments	*Some transport operators require the asset to be brought back to a specific station or zone. This requires knowledge about the desired destination or trip (single, return, multi-leg).*

Item	1.2
Who	As a USER
What	I want to know the PRICING of my trip
Why	To determine how expensive my trip will be
Required for MaaS	PRICING=yes
Comments	

Item	1.3
Who	As a USER
What	I want to receive a single INVOICE for my entire trip
Why	To simplify my cost overview
Required for MaaS	INVOICE=yes
Comments	

Item	1.4
Who	As a USER
What	I want to give a RATING and see other ratings of a transport operator
Why	To leave my feedback or determine if I want to use a certain transport operator
Required for MaaS	RATING=optional
Comments	

Item	1.5
Who	As a USER
What	I want to be able to REPORT an issue
Why	In case the asset I want to use has a problem/damage/issue
Required for MaaS	REPORT=yes
Comments	*Maybe this doesn't have to be available in an API, but needs to be covered by B2B arrangements. A User want the MaaS Provider to solve any issues, as this is their travel interface. A booking should only be made if an asset has no known technical issues, a transport operator should facilitate this.*

Item	1.6
Who	As a USER
What	I want to be able to select an asset based on COMPETENCES of the vehicle
Why	To fit with the criteria for my trip
Required for MaaS	COMPETENCES=yes

Comments	*E.g. selection of number of seats, type of vehicle, range, fuel type etc.*
	• *Proposals:*
	   o *No of passengers*
	   o *Propulsion (e.g. hydrogen)*
	   o *Vehicle class*
	   o *Brand*
	   o *Type*
	   o *Bicycle type (men, women, tandem)*
	   o *Steering wheel on left or right*
	   o *Colour*
	   o *State of charge (%)*
	   o *Exclusive yes/no (in case of ridesharing)*
	   o *Type of access/key*
	   o *Towing hook*
	   o *Airconditioning*
	   o *Cabrio*
	   o *Child's seat*
	   o *Winter tires*
	   o *Allowed to travel abroad*
	   o *Pets allowed*
	   o *Smoking allowed*
	   o *Underground parking allowed*
	   o *Easy accessibility to location (lift, escalator)*

Item	1.7
Who	As a USER
What	I want to receive SUPPORT during my trip
Why	In case I want to be guided along my travel, get additional suggestions or need any kind of support.

Required for MaaS	SUPPORT=yes
Comments	*Added in v0.9*

Part 2: from a MAAS PROVIDER perspective

Item	2.1
Who	As a MAAS PROVIDER
What	I want to know which travel means are available around STARTLOCATION which allow to reach DESTINATION
Why	To give travel advice to the USER
Required for MaaS	STARTLOCATION=yes DESTINATION=conditional
Comments	*The destination is not always relevant, but some assets need to be brought back to their specific station or zone or even if a one way trip is possible, to a specific zone or station at destination location*

Item	2.2
Who	As a MAAS PROVIDER
What	I want to know if the trip starts at STARTLOCATION and ends at DESTINATION Or will end at the STARTLOCATION
Why	To define my travel options to the USER
Required for MaaS	STARTLOCATION=yes DESTINATION=conditional
Comments	Covered by user story 2.1 *The destination is not always relevant, but some shared bikes need to be brought back to their specific station or zone or even if a one way trip is possible, to a specific zone or station at destination location*

Item	2.3
Who	As a MAAS PROVIDER
What	I want to know the ACCEPTABLE DISTANCE for the USER from LOCATION X to STARTLOCATION
Why	To define the travel options to the USER
Required for MaaS	ACCEPTABLE DISTANCE=optional LOCATION X=optional
Comments	*A user can have a preference for maximum distance he/she wants to walk to reach a bicycle. Proposed standard value = 500 meters*

Item	2.4
Who	As a MAAS PROVIDER
What	I want to know the CONDITIONS of a transport operator
Why	To define the travel options to the USER
Required for MaaS	CONDITIONS=yes (but can be periodical)
Comments	*E.g. business conditions, user conditions for the rental of the asset etc. These can be updated every week or month (t.b.d.), and do not necessarily have to be requested with each query*

Item	2.5
Who	As a MAAS PROVIDER
What	I want to be able to place a BOOKING with a TRANSPORT OPERATOR
Why	To book an asset beforehand
Required for MaaS	BOOKING=conditional
Comments	*This could also be done without a USER requesting a booking. In this case the booking risk lies with the MAAS PROVIDER instead of the TRANSPORT OPERATOR. In this case, the TO's own clients might not have access to the assets if the MP books everything in advance.*

Item	2.6
Who	As a MAAS PROVIDER
What	I want the USER to be able to OPEN/CLOSE/PAUSE the asset through my interface
Why	To make the use of the asset as easy as possible
Required for MaaS	OPEN=conditional CLOSE=conditional PAUSE=optional
Comments	*Requires information on the locking systems of operators. Pausing is an optional function to allow different pricing models when asset is temporarily parked by user*

Item	2.7
Who	As a MAAS PROVIDER
What	I want to give my USER on-the-fly USAGE INFORMATION about the asset usage and the booking from the TRANSPORT OPERATOR

Item	2.7
Why	To avoid having to keep and update all the information myself
Required for MaaS	USAGE INFORMATION=conditional
Comments	*A transport operator could like to send real-time usage instructions (e.g. "please unlock the bike now using the QR-code") to the User through the MaaS-provider interface.*

Item	2.8
Who	As a MAAS PROVIDER
What	I want to patch my USER through to the HELPDESK of the TRANSPORT OPERATOR in case of issues
Why	To deliver the best support possible
Required for MaaS	HELPDESK=yes
Comments	*A Transport Operator can give specific support about the asset in case of issues. A direct link between User and Transport Operator is required, the MaaS Provider can facilitate this link through their service. As a reference, insurance companies offer similar assistance, where a neutral helpdesk can take on the 'image' of the insurance provider that manages the specific contract of the User.*

Item	2.9
Who	As a MAAS PROVIDER
What	I want to be able to CANCEL/MODIFY a transaction or booking
Why	To inform the TRANSPORT OPERATOR about any changes
Required for MaaS	CANCEL=yes MODIFY=yes
Comments	*MaaS providers need to be able to cancel or modify transactions or bookings on behalf of their users.*

Item	2.10
Who	As a MAAS PROVIDER
What	I want to know if my USER can share a journey or booking with a USER from another MAAS PROVIDER
Why	To efficiently make use of available transportation through carpooling or ridesharing
Required for MaaS	no
Comments	*This allows higher occupancy of available assets through ridesharing and*

	carpooling.

Item	2.11
Who	As a MAAS PROVIDER
What	I want to receive information on public transport USER stops and line information
Why	To plan an efficient route for my USER and give the necessary SUPPORT along the journey.
Required for MaaS	no
Comments	*For planning purposes, e.g. information on kerbs, ramps, lights, displays, linetype and transport operator.*

Part 3: from a TRANSPORT OPERATOR perspective

Item	3.1
Who	As a TRANSPORT OPERATOR
What	I want to know from when to when (TIME T1 to TIME T2) the USER wants to use my assets
Why	To define if this fits my offer of assets
Required for MaaS	TIME T1(START TIME/DAY)=conditional TIME T2(END TIME/DAY)=conditional
Comments	*This is optional, only required in case of usage restrictions of the Transport Operator or to implement the option to book an asset beforehand (long-term).*

Item	3.2
Who	As a TRANSPORT OPERATOR
What	I want to know the DESTINATION of the USER
Why	To determine if my assets are suitable or available
Required for MaaS	DESTINATION=conditional
Comments	*The destination is not always relevant, but some shared bikes need to be brought back to their specific station or zone or even if a one way trip is possible, to a specific zone or station at destination location*

Item	3.3
Who	As a TRANSPORT OPERATOR

What	I want to know if the USER has the right USER COMPETENCE
Why	To determine if the USER is allowed to use my assets
Required for MaaS	USER COMPETENCE=yes
Existing API's	Not available/necessary in GBFS, use other MaaS-API specs
Comments	*E.g. the user should have a driving license, correct contact details, a membership etc. This could be a liability issue that needs to be covered with insurance providers.*

Item	3.4
Who	As a TRANSPORT OPERATOR
What	I want to know if the USER complies with my USER CONDITIONS before starting a trip
Why	To determine if the USER is allowed to use my assets
Required for MaaS	USER CONDITIONS=yes
Comments	*E.g. user is not on a blacklist, registered member*

Item	3.5
Who	As a TRANSPORT OPERATOR
What	I want to give a RATING and see other ratings of a USER
Why	To leave my feedback about and determine if USER can use my asset
Required for MaaS	RATING=optional
Comments	*A transport operator might want to rate a user or determine if a user is allowed to use an asset based on their rating*

Item	3.6
Who	As a TRANSPORT OPERATOR
What	I want to be able to receive USER AUTHENTICATION
Why	To confirm the identity of the USER using my asset
Required for MaaS	USER AUTHENTICATION=yes
Comments	*Authentication provides the transport operator with a confirmation of a user's identity, profile or token.*

Item	3.7
Who	As a TRANSPORT OPERATOR
What	I want to be able to notify the MaaS provider to CONTACT the USER
Why	In case of problems, emergencies or other issues
Required for MaaS	CONTACT=yes
Comments	*A transport operator can give specific support about the asset in case of issues. A direct link between user and transport operator is required, the MaaS Provider can facilitate this link through their service (see also item 2.8).*

Item	3.8
Who	As a TRANSPORT OPERATOR
What	I want to be able to CANCEL/MODIFY a transaction or booking
Why	To inform the MAAS PROVIDER about any changes
Required for MaaS	CANCEL=yes MODIFY=yes
Comments	*Transport operators need to be able to cancel or modify transactions or bookings in case an asset is unavailable or delayed.*

7 Process Flows

Together with the eMaaS project team from the University of Twente, process flows for the customer journey have been defined. This helps to scope the necessary functions required in the API building blocks.

The goal is to accommodate different business models within these functional flows. For example, both asset information from free-floating systems (bike sharing, car sharing, ride sharing, taxi) and (virtual) station/fixed-route based systems (public transport, (virtual)mobility hub or station-dependent transportation) can be shared through the functional descriptions provided in this chapter, as the focus lies on sharing asset information.

Functional Blocks

Figure 1 below depicts the main functional and information blocks of the MaaS API. This visualization aims at giving a general overview of the different functional modules within the API. The Transport Operator to MaaS Provider (TO-MP) API is composed of 7 functional blocks. The 2 orange functional blocks (Operator Information & Asset Information) contain fixed information about the transport operator and their assets based on the General Bikeshare Feed Specification (GBFS).
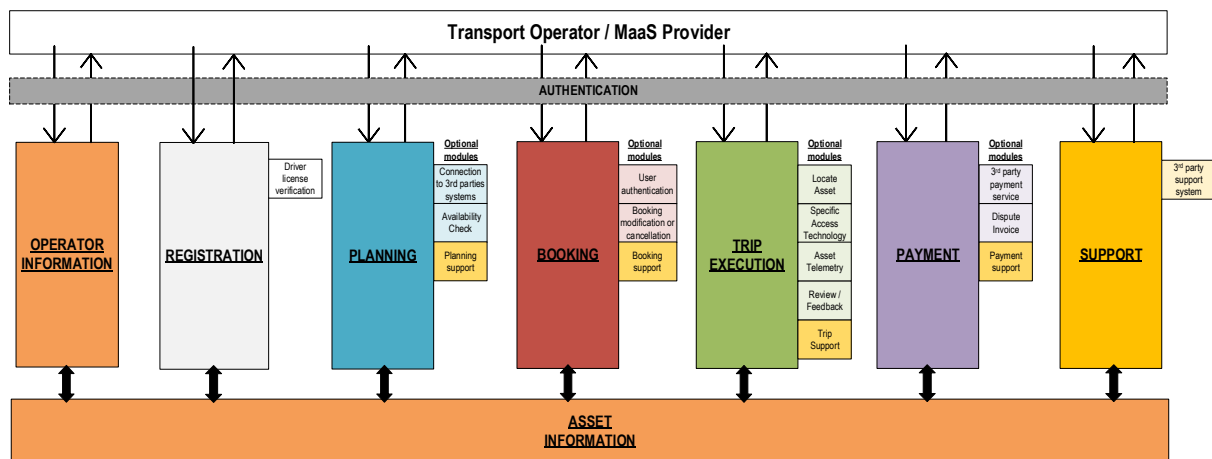

Fig. 1: MaaS Provider to/from Transport Operator (TO-MP) API – Functional Blocks visualization

Fig. 1 shows the different functions in the interface between MaaS Providers and Transport Operators for the functions of (de)registration, planning, booking, trip execution, payment and providing operator information & asset information respectively.

- ➢ Operator Information/General Information: Gives static information on the operator according to the GBFS(+) standard.
- ➢ Registration: Offers the possibility for users to delete, sign-up or log-in into their account with the MaaS provider
- ➢ Planning: Gives information about availability, estimated travel time and costs.
- ➢ Booking: Allows booking of a specific asset for a specific place, time and date.
- ➢ Trip Execution: Allows access to asset and travel during booked period.
- ➢ Payment: Allows payment of the service. Supports different business models (i.e. pay-as-you-go or subscription-based).
- ➢ Support: Assists users in the solution of operational troubles encountered during any part of the process. Connects with optional support modules.
- ➢ Asset Information: Is defined as a separate module that can be used by other modules to supplement API calls with specific asset information where applicable. Assets can be vehicles or for example infrastructural assets.
- ➢ Optional modules: The more dynamic functional blocks have additional optional modules which are used for execution of sub-processes derived from the main

functions which might not be desired or required depending on scope of the MaaS implementation and Business Models.

API Authentication
Figure 2 below shows that the API features authentication for each call to allow secure communication and exchange of information between Maas Provider and Transport Operators.
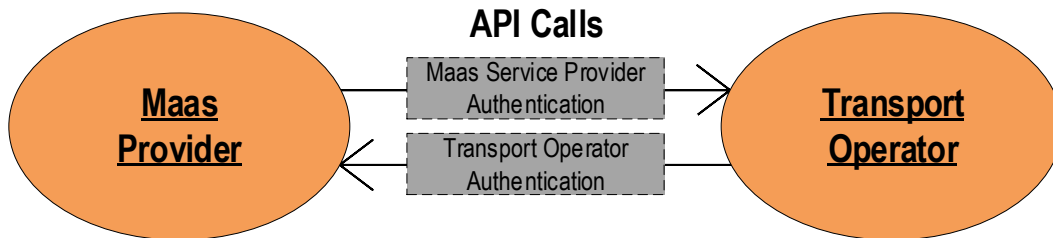

Fig. 2: API calls and authentication

MaaS Provider authentication and authorization should take place following OAuth 2.0 standards using either Authorized Code or Resource Owner password credentials flows/grant types. The flows are scenarios an API client performs to get an access token from the authorization server.

OAuth 2.0 provides the below mentioned flows suitable for the types of API clients we are looking for in MaaS. The relevant security schemes have been also included in the OpenAPI 3.0 description on Swaggerhub.

> *Authorization code* – The most common flow, mostly used for server-side and mobile web applications. This flow is similar to how users sign up into a web application using their Facebook or Google account.

> *Resource owner password credentials* (or just *password*) – Requires logging in with a username and password. Since in that case the credentials will be a part of the request, this flow is suitable only for trusted clients (for example, official applications released by the API provider).

A Transport Operator might require authentication to communicate with a MaaS Provider, for example to manage (update/cancel) a booking or to send a callback request. That makes bidirectional authentication necessary.

Operational view of the API

Figures 4 to 8 on the following pages show an operational view of the API. These are separated in a technical process flow and a user journey. The figures are accompanied by a description of how the necessary functions between MaaS Provider and Transport Operator are fulfilled in the current API specification.

The API calls, as described in Swaggerhub (see Table 1), are also indicated in the process flow. Moreover, Optional Modules are shown throughout the operational flows to point out (external) procedures that can happen within the current process.

Table 1. Overview of API calls as described in Swaggerhub

Reference #	API call	Description
❌		Not defined yet
1. Operator information		Gives information about systems, stations, operating hours
1.1	GET/system-information	describes the system (incl. TO conditions)

(1.2)		GET/station-information	describes all available stations
(1.3)		GET/system-hours	describes the system hours of operation
(1.4)		GET/system-calendar	describes the operating calendar for a system. An array of year objects defined as follows (if start/end year are omitted, then assume the start and end months do not change from year to year). [from GFBS]
(1.5)		GET/system-regions	describes regions for a system that is broken up by geographic or political region. It is defined as a separate feed to allow for additional region metadata (such as shape definitions). [from GBFS]
(1.6)		GET/system-alerts	informs customers about changes to the system outside of normal operations. This feed is intended to inform customers about changes to the system that do not fall within the normal system operations. For example, system closures due to weather would be listed here, but a system that only operated for part of the year would have that schedule listed in the system-calendar.json feed. This file is an array of alert objects defined as below. Obsolete alerts should be removed so the client application can safely present to the end user everything present in the feed. The consumer could use the start/end information to determine if this is a past, ongoing or future alert and adjust the presentation accordingly. [from GBFS]
(1.7)		GET/system-pricing-plans	gives pricing information
(2.) Planning			gives information about transport asset availability
(2.1)		GET/free-asset-status	describes asset availability and asset options
(3.) Booking			a booking is the main object exchanged between MaaS and a TSP. before a booking can be made via a TSP, available options at a given location can be listed as follows.
(3.1)		GET/bookings/options	Returns available transport options for given coordinate. Start time can be defined, but it is optional. If startTime is not provided, but required by the third party API, a default value of "Date.now()" is used. [from MaaS-API]. Including querying options for ridesharing / carpooling.
(3.2)		GET/bookings/	Returns an array of all the bookings that have been created in an earlier timeframe
(3.3)		POST/bookings/	Creates a new Booking for the TSP in booked state. The returned object will be a reference that is passed back & forth throughout the booking life cycle. The Booking may be modified in the response, e.g. location being adjusted for a more suitable pick-up location. In addition, the service may contain a meta attribute for arbitrary TSP metadata that the TSP needs later, and token attribute depicting how long the current state is valid.
(3.4)		GET/bookings/{id}	Return a specific booking that has been created through the system.
(3.4.1)		GET/bookings/state	The life-cycle state of the booking (from NEW to FINISHED)
(3.4.2)		GET/bookings/leg	
(3.4.3)		GET/bookings/customer	
(3.4.4)		GET/bookings/id	The identifier MaaS will be using to referring to the booking
(3.4.5)		GET/bookings/token	The validity token (such as booking ID, travel ticket etc.) that MaaS clients will display to validate the trip when starting the leg.
(3.5)		PUT/bookings/{id}	Modifies the state of a Booking, e.g. cancels, pays or reschedules it. The previous booking information is passed forward as-is for reference
(3.5.1)		PUT/bookings/state	The life-cycle state of the booking (from NEW to FINISHED)
(3.5.2)		PUT/bookings/leg	

(3.5.3)	PUT/bookings/customer	
(3.5.4)	PUT/bookings/id	The identifier MaaS will be using to referring to the booking
(3.5.5)	PUT/bookings/token	The validity token (such as booking ID, travel ticket etc.) that MaaS clients will display to validate the trip when starting the leg.
(4.) Trip Execution		Provides asset location, grants access or exit from an asset or transport system, offers trip monitoring
(4.1)	PUT/notification/{id}	Notification from MaaS provider to transport operator in case of user no-show or other usage-related issues.
(4.2)	GET/location/{id}	provides location of specific asset related to a booking
(4.3)	PUT/access/{id}	provides access to an asset or transport system
(4.4)	PUT/pause/{id}	provides option to temporarily pause/park asset rental (in case lower fares are given for parked assets)
(4.5)	PUT/exit/{id}	provides exit from an asset or transport system
(4.6)	POST/tripmonitoring/{id}	monitors the current location of the asset and duration & distance of the trip
(4.7)	GET/leg-summary/{id}	generates a summary of the specific leg
(5.) Payment		t.b.d.
(6.) Support		Support for the user while the trip is being executed
(6.1)	POST/request/{id}	Identifies a request for support from user via MP and modifies a support request
(6.2)	GET/status/{id}	Identifies the current status of the support request
(6.3)	POST/callback/{id}	Subscribes to a webhook for future support notifications from TO
(6.4)	POST/unsubscribe/{id}	Unsubscribes to a webhook for callback notifications

As a summary of the data exchanged provided by the API calls in all the operational blocks of the TO-MP API, Figure 3 shows an overview of the data exchange between the Transport Operators and MaaS providers within the TO-MP API.
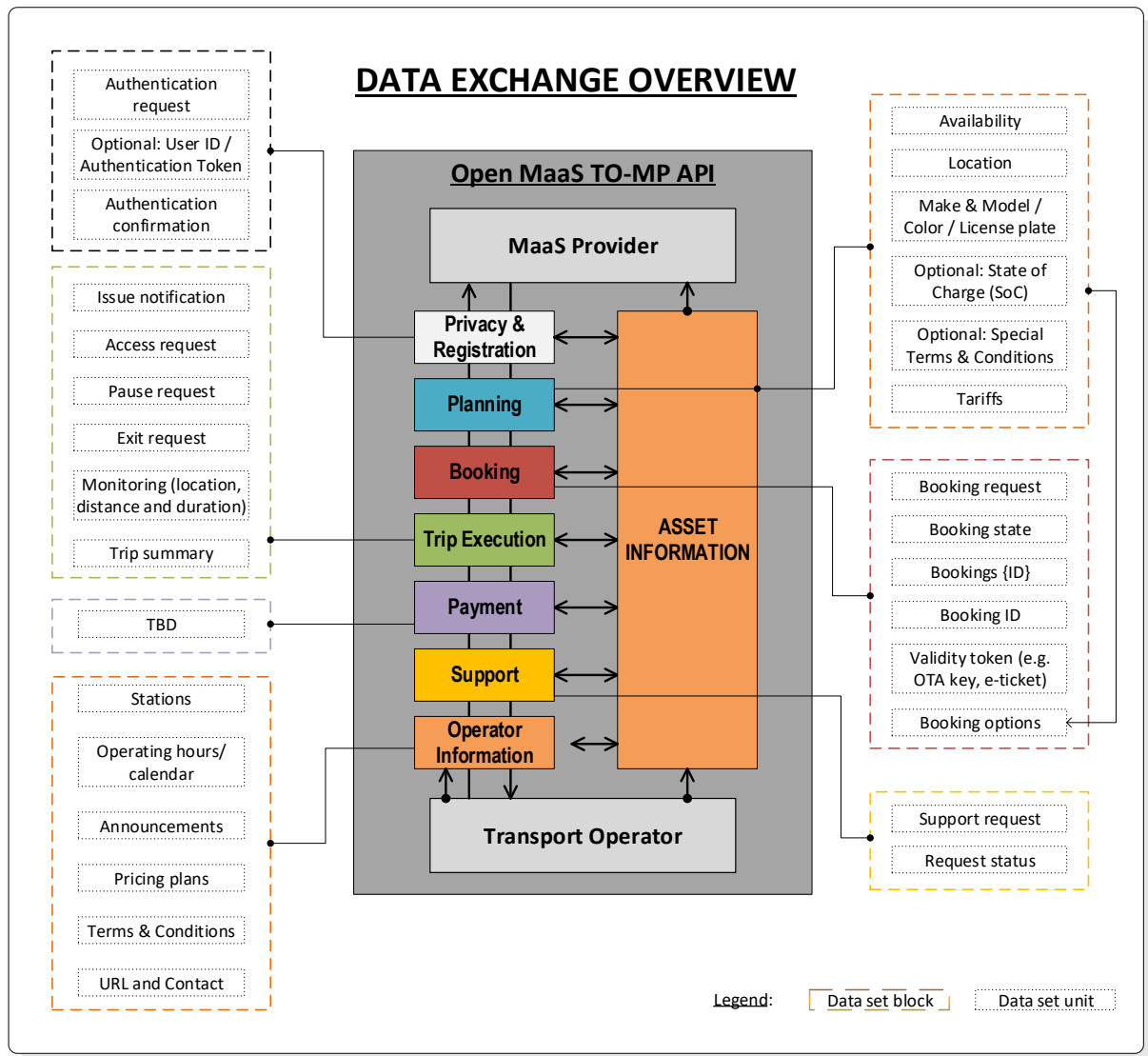
DATA EXCHANGE OVERVIEW

Fig. 3: Data exchange overview of TO-MP API

Registration and Planning

The first operational block in the TO-MP API is the registration or deregistration block. This block offers the possibility for users to either delete, sign-up or log-in into their account with the MaaS provider. The TO-MP API would enable the possibility to use the costumer account with a specific Transport Operator to log-in into the MaaS Provider system.

Planning forms the exploration phase of a trip, where options are explored by the User through the MaaS Provider. The MaaS Provider has an archive of (semi-)static general information/operator information which is periodically retrieved from the Transport Operator. The MaaS Provider can check real-time availability of assets to give different travel options to the User.

The following table describes the functions between MaaS Provider and Transport Operators and refers to the available API functions that are already available.

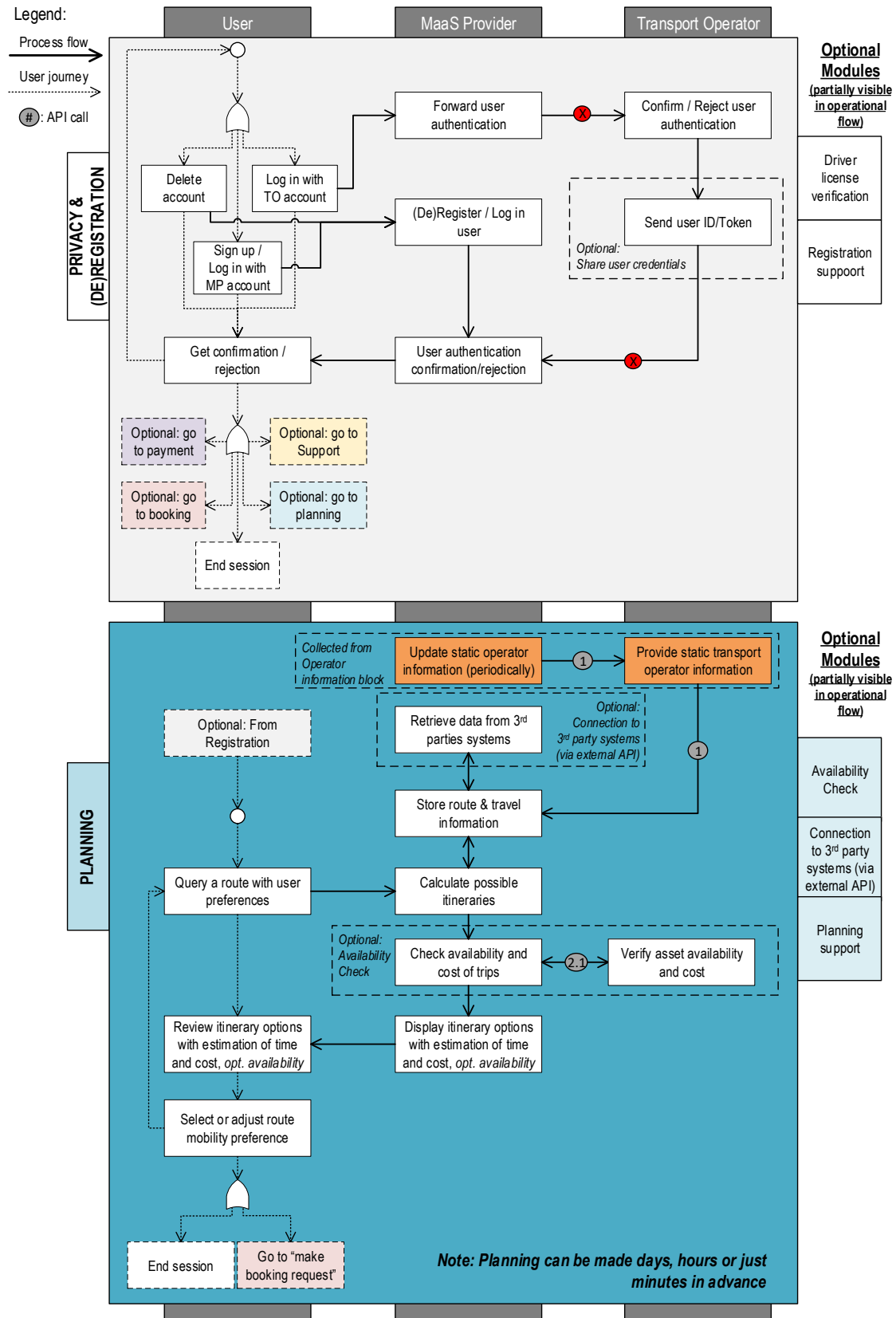| Functions between MaaS-provider and Transport Operator			
Category	*Function*	*User Story*	*Reference*
Registration	User authentication	t.b.d	t.b.d
Planning	Update static operator information > provide static operator information	1.2; 1.6; 2.1; 2.2; 2.3; 2.4; 3.4	General Information [from GBFS]
Planning	Check availability of trips > Verify availability and temporarily reserve asset	1.1; 1.2; 2.1; 2.2; 2.3; 3.2	Asset availability and competences [from GBFS and amended]

Legend:
- Process flow
- User journey
- (#): API call

PRIVACY & (DE)REGISTRATION

User	MaaS Provider	Transport Operator

- Delete account
- Log in with TO account
- Sign up / Log in with MP account
- Forward user authentication → ✕ → Confirm / Reject user authentication
- (De)Register / Log in user
- *Optional: Share user credentials* — Send user ID/Token
- Get confirmation / rejection ← User authentication confirmation/rejection ← ✕
- Optional: go to payment
- Optional: go to Support
- Optional: go to booking
- Optional: go to planning
- End session

Optional Modules (partially visible in operational flow)
- Driver license verification
- Registration suppoort

PLANNING

- *Collected from Operator information block* — Update static operator information (periodically) — (1) — Provide static transport operator information
- Optional: From Registration
- *Optional: Connection to 3rd party systems (via external API)* — Retrieve data from 3rd parties systems
- (1)
- Store route & travel information
- Query a route with user preferences → Calculate possible itineraries
- *Optional: Availability Check* — Check availability and cost of trips — (2.1) — Verify asset availability and cost
- Review itinerary options with estimation of time and cost, *opt. availability* ← Display itinerary options with estimation of time and cost, *opt. availability*
- Select or adjust route mobility preference
- End session
- Go to "make booking request"

Note: Planning can be made days, hours or just minutes in advance

Optional Modules (partially visible in operational flow)
- Availability Check
- Connection to 3rd party systems (via external API)
- Planning support

Fig. 4: Operational view of the Registration & Planning modules

Booking

Booking is the phase where the User has committed to a certain travel option offered by the MaaS Provider. This can be a result of the Planning phase, or in case a User knows exactly which ticket or booking they want, the result of a new booking request directly.

| Functions between MaaS-provider and Transport Operator			
Category	*Function*	*User Story*	*Reference*
Booking	Make booking request > Process booking	1.6; 2.5; 3.1; 3.2	Booking > POST/bookings/ [from MaaS-API]
Booking	Provide User Authentication > Request User Authentication	3.3; 3.4; 3.6	Components/securityschemes [from MaaS-API]
Booking	Cancel / Modify Booking > Cancel / Modify Booking	1.5; 2.9; 3.8	Booking > PUT/bookings/{id} [from MaaS-API]
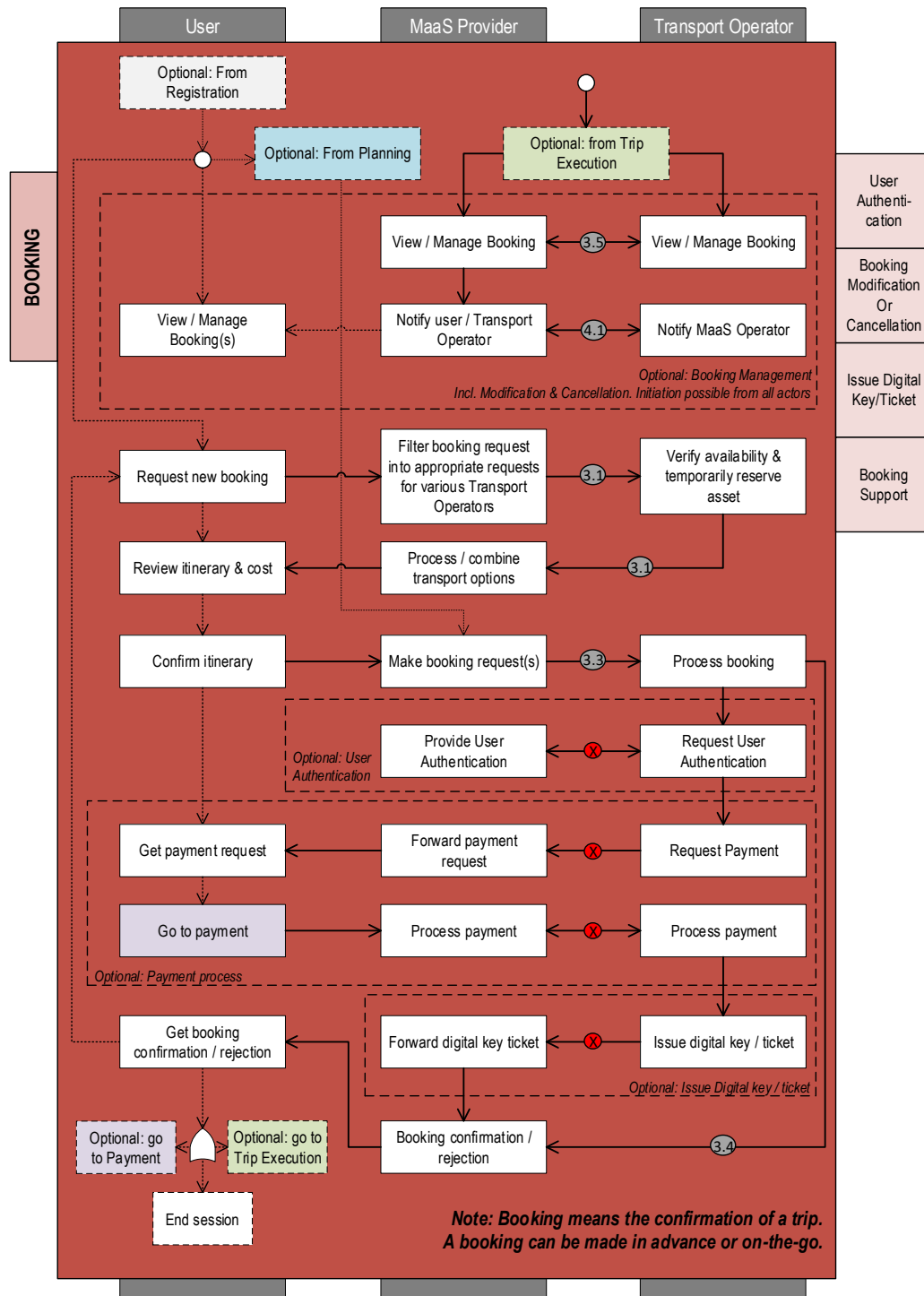
Fig. 5: Operational view of the Booking module

Trip Execution

The trip execution module offers all functionalities for the User during the trip. This includes breakdown into different legs, access to the asset, ending a leg and monitoring a trip. When all legs are concluded, summaries of the specific legs are exchanged to offer the User a complete overview of the executed trip.

| Functions between MaaS-provider and Transport Operator			
Category	*Function*	*User Story*	*Reference*
Trip Execution	Forward location request > provide location	1.1; 2.1	Asset availability and competences > free_asset_status [from GBFS] New proposal: GET/provide_location/{id}
Trip Execution	Forward access request > grant / reject access	2.6; 3.6	New proposal: PUT/request_access/{id}
Trip Execution	Monitor trip <> monitor use of asset	2.7	New proposal: GET/monitor_trip/{id}
Trip Execution	Forward exit request > grant / reject exit	2.6	New proposal: PUT/request_exit/{id}
Trip Execution	Generate Trip Summary > Provide Leg Summary	1.3	New proposal: GET/leg_summary/{id}
Trip Execution	Manage Review / Feedback <> Review / Feedback with respect to user	1.4; 2.8; 3.5	t.b.d.
Trip Execution	Trip support (optional)	2.8; 3.7	New proposal: GET/notification/{id}
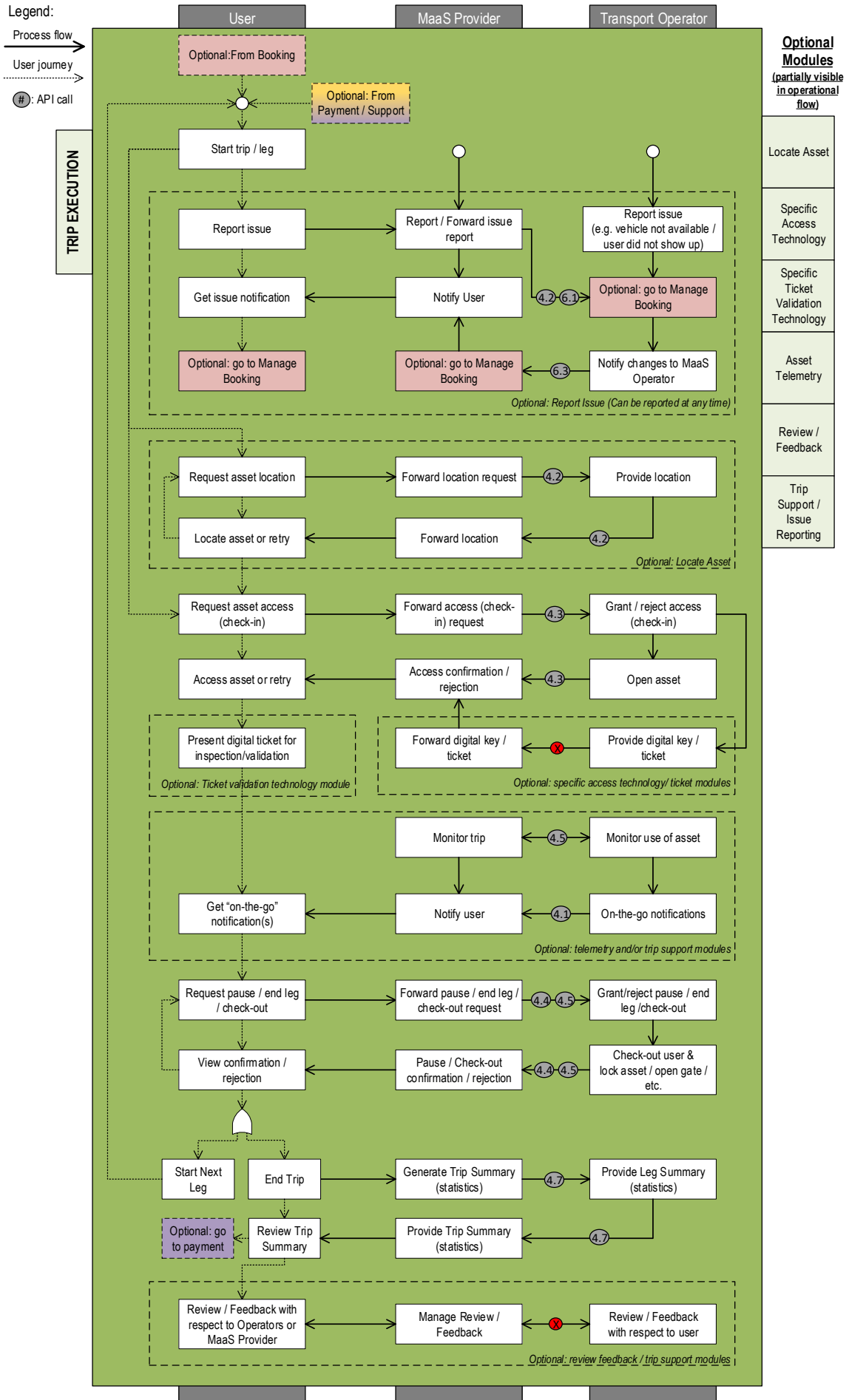
Fig. 6: Operational view of the trip execution module

Payment

The payment module offers two alternative payment models that can also be used in conjunction: a subscription model and a pay-as-you-go model. A subscription model can be used to determine how much time/distance/credits/etc. a User has spent for the trip and subtract the necessary value. A pay-as-you-go model provides the opportunity for direct invoicing and payment after a trip has been completed. Additionally, the payment module offers a submodule for special payments such as the payment of a deposit, the payment of a fine, payment for damage to the vehicles, etc.

| **Functions between MaaS-provider and Transport Operator**			
Category	*Function*	*User Story*	*Reference*
Payment/SM	Request / receive payment <> Request / receive payment	1.2; 1.3	t.b.d.
Payment/SM	Manage payment <> manage payment	1.2; 1.3	t.b.d.
Payment/PAYG	Collect total trip costs > calculate trip costs	1.2; 1.3	t.b.d.
Payment/PAYG	Manage dispute <> manage dispute	1.4; 1.5	t.b.d.
Payment/PAYG	Request / receive payment <> Request / receive payment	1.2; 1.3	t.b.d.
Payment/SM	Manage payment <> manage payment	1.2; 1.3	t.b.d.
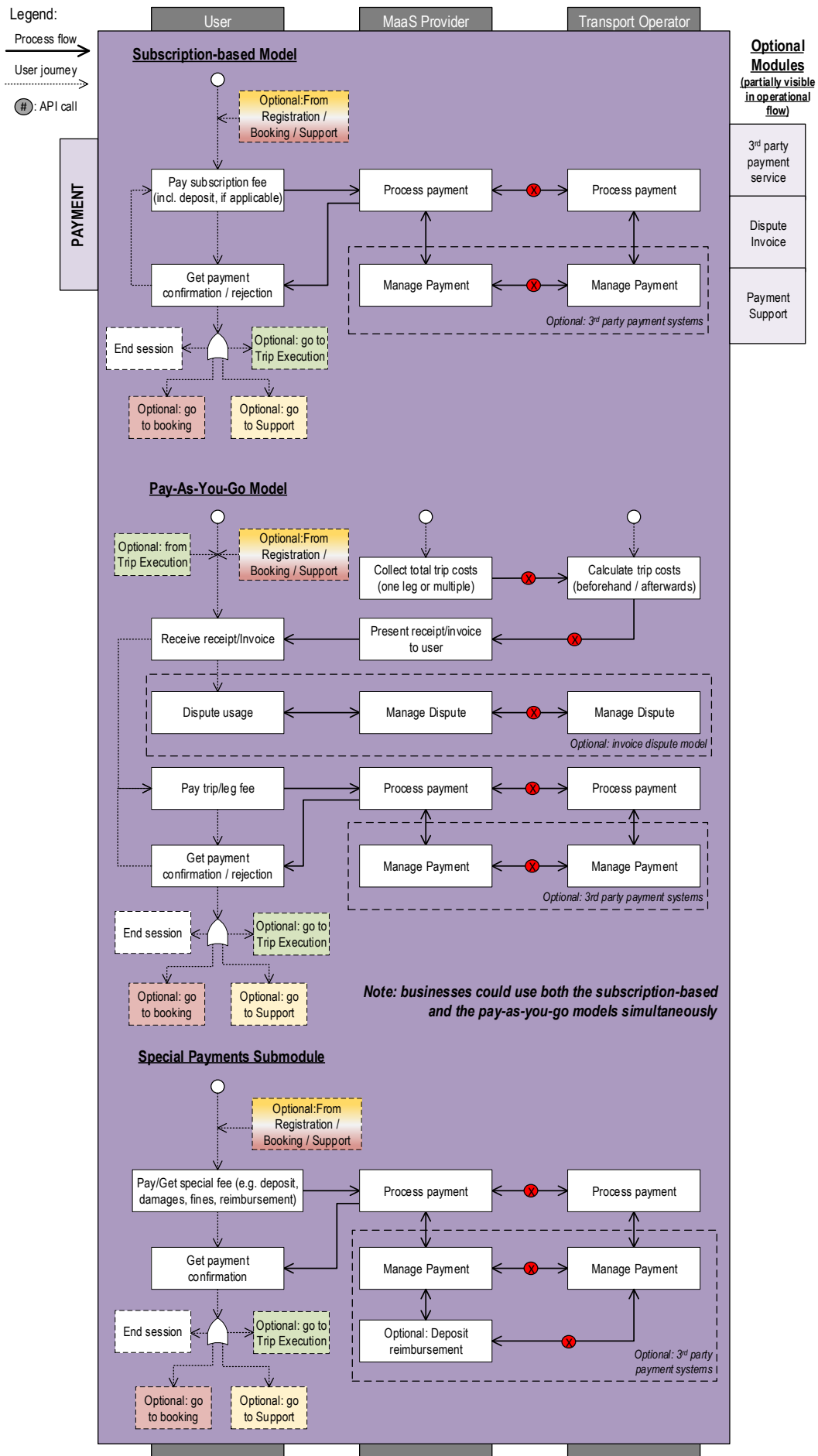
Fig. 7: Operational view of the payment module

Support

The support module offers functional blocks that refer to the technical assistance to the user in case of an issue experienced during any of the other modules. Within this module optional 3rd party systems can be used to solve the user problem.
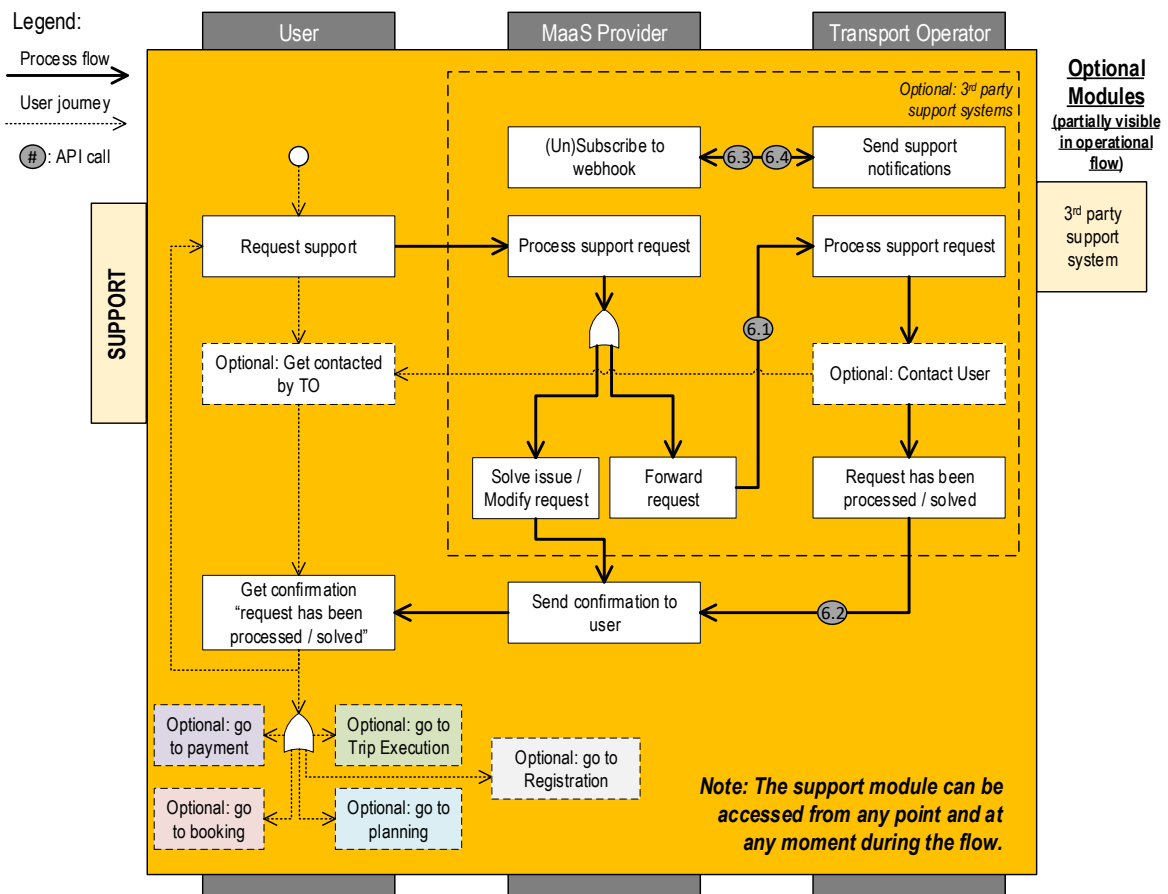

Fig. 8: Operational view of the support module

8 Overview of parameters for MaaS

The following overview summarizes the user stories in categories, GBFS(+)/MaaS-API options and specifies requirement for MaaS

Nr.	User Story	Category	Existing API description used in this document	Required for MaaS
1.1	As a USER, I want to depart from STARTLOCATION and arrive at DESTINATION, To define from where to where I need mobility services for my trip	PLANNING	GBFS MaaS-API	Yes
1.2	As a USER, I want to know the PRICING of my trip, To determine how expensive my trip will be	PRICING	GBFS	Yes
1.3	As a USER, I want to receive a single INVOICE for my entire trip, To simplify my cost overview	PRICING	No	Yes
1.4	As a USER, I want to give a RATING and see other ratings of a transport operator, To leave my feedback or determine if I want to use a certain transport operator	TRIP EXECUTION	No	No
1.5	As a USER, I want to be able to REPORT an issue, In case the asset I want to use has a problem/damage/issue	TRIP EXECUTION	No	Yes
1.6	As a USER, I want to be able to select an asset based on COMPETENCES of the vehicle, To fit with the criteria for my trip	PLANNING	GBFS+	Yes
2.1	As a MAAS PROVIDER, I want to know which travel means are available around STARTLOCATION which allow to reach DESTINATION, To give travel advice to the USER	PLANNING	GBFS MaaS-API	Yes
2.2	As a MAAS PROVIDER, I want to know if the trip starts at STARTLOCATION and ends at DESTINATION,	PLANNING	GBFS MaaS-API	Yes
2.3	As a MAAS PROVIDER, I want to know the ACCEPTABLE DISTANCE for the USER from LOCATION X to STARTLOCATION , To define the travel options to the USER	PLANNING	GBFS+?	No
2.4	As a MAAS PROVIDER, I want to know the CONDITIONS of a transport operator, To define the travel options to the USER	PLANNING	GBFS MaaS-API	Yes
2.5	As a MAAS PROVIDER, I want to be able to place a BOOKING with a TRANSPORT OPERATOR , To book an asset beforehand	BOOKING	MaaS-API	Conditional

2.6	As a MAAS PROVIDER, I want the USER to be able to OPEN/CLOSE/PAUSE the asset through my interface, To make the use of the asset as easy as possible	TRIP EXECUTION	GBFS+?	Conditional
2.7	As a MAAS PROVIDER, I want to give my USER on-the-fly USAGE INFORMATION about the asset usage and the booking from the TRANSPORT OPERATOR, To avoid having to keep and update all the information myself	TRIP EXECUTION	No	Optional
2.8	As a MAAS PROVIDER, I want to patch my USER through to the HELPDESK of the TRANSPORT OPERATOR in case of issues, To deliver the best support possible	TRIP EXECUTION	No	Yes
2.9	As a MAAS PROVIDER, I want to be able to CANCEL/MODIFY a transaction or booking , To inform the TRANSPORT OPERATOR about any changes	BOOKING	MaaS-API	Yes
2.10	I want to know if my USER can share a journey or booking with a USER from another MAAS PROVIDER	PLANNING	No	No
3.1	As a TRANSPORT OPERATOR , I want to know from when to when (TIME T1 to TIME T2) the USER,	PLANNING	GBFS MaaS-API	Conditional
3.2	As a TRANSPORT OPERATOR , I want to know the DESTINATION of the USER, To determine if my assets are suitable or available	PLANNING	GBFS MaaS-API	Conditional
3.3	As a TRANSPORT OPERATOR , I want to know if the USER has the right USER COMPETENCE, To determine if the USER is allowed to use my assets	PLANNING	No	Yes
3.4	As a TRANSPORT OPERATOR , I want to know if the USER complies with my USER CONDITIONS before ,	PLANNING	No	Yes
3.5	As a TRANSPORT OPERATOR , I want to give a RATING and see other ratings of a USER, To leave my feedback about and determine if USER can use my asset	TRIP EXECUTION	No	Optional
3.6	As a TRANSPORT OPERATOR , I want to be able to receive USER AUTHENTICATION, To determine if and how USER may use my asset	PLANNING	MaaS-API	Yes
3.7	As a TRANSPORT OPERATOR , I want to be able to CONTACT the USER, In case of problems, emergencies or other issues	TRIP EXECUTION	No	Conditional
3.8	As a TRANSPORT OPERATOR, I want to be able to CANCEL/MODIFY a transaction or booking , To inform the MAAS PROVIDER about any changes	PLANNING	MaaS-API	Yes

9 **GBFS+ recommendations**

The following additions to GBFS have been proposed by the technical working group to the GBFS community. The acceptance of these suggestions and future phasing is still to be defined. Possibly, a national GBFS+ standard can be implemented to speed up developments in the Netherlands.

1. Deep links, Add rental_url to free bikes and stations
There is already a change-requests (from others) for an extension of the standard, covering exactly our wishes. So we include request #25 in GBFS+, which enables deep links.

2. Type_of_system
We will add type_of_system in the "system info" file. Allowed values are [free_floating, station_based, virtual_station_based]

3. Type_of_bike
We add a file "Types_of_bikes" which describes the different bike types (type_id, name, gears, electric, description, img_url)
In free-bike-status file we add the field type_of_bike
(our first proposal on OpenBikeShare Github)

4. TTL
The time to live (TTL) for real-time data feeds will be at most 30s, so that traveller has always the most actual information about the availability of bicycles.

There are some other topics to cover to make an awesome bike standard in the future, but more research has to be done. Possible topics are:

- Which fields should be compulsory?
- Operation area: For a free-floating system we would like to indicate where you can return your bike (for example you are only allowed to return the bike within the city). In this https://github.com/NABSA/gbfs/issues/65 thread there is already a discussion about this idea.
- Virtual stations: We would like to introduce virtual stations (a virtual location where you allowed to park your bike) within GBFS so operators comparable with Donkey Republic are supported as well. We created a proposal. The exact location of a virtual zone should be presented as GeoJSON polygon in station_information.json.
- Option to define a radius around a bike or bikesharing station for location-specific API-calls
- Option to OPEN/CLOSE/PAUSE an asset

10 Technical Specifications

The technical working group suggests to implement this interface using REST-API's

Other quality specifications are:

Criteria	Value
Time To Live	Max. 30 seconds
Reliability	95%
API-call max radius around asset	500 meters
API-call min radius around asset	10 meters
Pagination of API-responses	t.b.d. after testing of v1.0
Rate limiting	t.b.d. after testing of v1.0

11 To-do's and risks

- Opening and closing of assets can vary greatly between different transport operators. Some regard this technology as their own intellectual property and are not willing to offer external access. This is a risk for common API development and might require further harmonization in the future.
- Which service/helpdesk functions are required for the User?
- Options for ticketing and payment of legs/trips
- Central registry of available stations/hubs/operators and assets, with unique ID's

12 Available API's

This chapter provides and overview of available commercial and non-commercial API's on the market.

Name	Website	Service	License
BoMaaS / FLOU.io	https://tapahtumat.tekes.fi/event/bomaas2310 https://app.swaggerhub.com/apis/FLOU	Ticket sales (example) Service registry catalogue	Creative commons 4.0
SUTI	http://www.suti.se/	Exchange of demand responsive traffic information between clients and providers	Membership
NeTEX	http://netex-cen.eu/	Exchange of Public Transport schedules and related data (topology and fare information)	GPL license managed by CEN standards process
GTFS	General Transit Feed Specification https://developers.google.com/transit/gtfs/	Public transportation schedules and associated geographic information	Google - Apache 2.0
GBFS	General Bikeshare Feed Specification https://github.com/NABSA/gbfs	Bikesharing system, service and status information	Open standard, community on Github
MaaS-API	http://www.maas-api.org/	Booking and listing	MIT license / Alliance Membership
Uber API	https://developer.uber.com/docs/riders/ride-requests/introduction	Uber ride requests	Developer dashboard membership
IPSI	Interoperable Product Service Interface https://oepnv.eticket-deutschland.de/en/fachpublikationen/themenportal-ipsi/	Mobile ticketing, ticket purchase, conditions for sale of tickets	License with VDV
Wiener API	http://akirk.github.io/Wiener-Linien-API/	Public transport schedules	Open government data Wien (OGD)
OTP	Open Trip Planner http://www.opentripplanner.org/	Multimodal trip planner Passenger information	Open source

Name	Website	Service	License
		and transportation network analysis	
OTM	Open Trip Model www.opentripmodel.org	Exchange real-time logistics data	Creative Commons 4.0
TripGo API	https://developer.tripgo.com/ https://developer.tripgo.com/specs/#	Plan door-to-door trips using a large variety of public and private transport. It integrates real-time information and, for selected providers, allows users to book and pay for transport.	Apache License 2.0 Free testing below a threshold of API calls