TRANSPORT OPERATOR MOBILITY-AS-A-SERVICE PROVIDER API (TOMP-API) PROFILE SWITZERLAND

Systemaufgaben Kundeninformation (SKI) – Team SKI+
https://transportdatamanagement.ch

Status	Recommended
Version	0.5
Date	2023-01-01
Authors	Markus Meier (SBB SKI+), David Rudi (SBB SKI+), Matthias Günter (SBB SKI+)
Copyright	

Document information

Description	This document contains information, assessments, and explanations on the TOMP-API standard, which is intended for use by the SKI+ team on behalf of the FOT (Federal Office of Transport) in Switzerland.
Target audience	People who use or want to use data and APIs with the TOMP-API standard in the field of mobility in Switzerland to conceive, design, develop and test business applications.
Electronic documentation	https://transportdatamanagement.ch/en/standards/

Change History

Version	Status	Changes	Authors	Date
0.1	Draft	Initial version	M. Meier	2022-06-17
0.4	Draft	Various extensions	M. Meier, D. Rudi, M. Günter	2022-11-22
0.5	Draft	Feedback from FOT	M. Meier, D. Rudi, M. Günter, D. Grögler, A. Döbeli	TBA

Table of Content

1 What is it all about? 3
2 Description and Context 3
3 Who is responsible? 4
4 Important Links 4
5 Technology stack and standards 5
6 Usage 5
7 Datasets/Services Switzerland 5
8 Assessment 5
9 Conclusions 6
10 Specification and recommendation 6
11 TOMP-API – Swiss Profile 7
11.1 Purpose and Scope 7
11.2 Security [must] 7
11.3 Service Catalogue [must] 7
11.4 Geo Referencing [must] 8
11.5 Date and Time [must] 9
11.6 HTTP parameters [must] 9
11.6.1 Accept-Language [must] 9
11.6.2 Compression [must] 9
11.6.3 API [must] 9
11.6.4 API-Version [must] 9
11.6.5 maas-id [may] 10
11.7 TOMP-API endpoints resp. operations 10
11.7.1 Planning [must] 10
11.7.2 booking [must] 13
11.7.3 trip execution [tbd] 15
11.7.4 operator information [tbd] 15
11.7.5 Payment [tbd] 15

1 What is it all about?

This document describes the TOMP-API standard, which is currently considered to be pioneering for MaaS (Mobility as a Service), specifically, regarding the booking workflow (from offers to payment) in the field of intermodal mobility.

This document includes a general description of the standard (chapters 2 ff.), an analysis of the standard for deployment/usage in Switzerland (chapter 6 ff.), and a draft profile (chapter 11) with specifications and recommendations for the deployment of the TOMP-API in Switzerland.

Depending on the outcome of current, on-going studies and the overall strategy of the Swiss Confederation and the Federal Office of Transport (FOT), the development of a booking workflow based on TOMP-API can become an important task of the future NADIM[1].

The FOT's and thus SKI+'s long-term goal is to enable an open MaaS ecosystem for Switzerland, involving the booking workflow that facilitates a high interoperability among the MaaS providers and transport operators participating in NADIM.

2 Description and Context

The TOMP-API is the quasi-standard for MaaS today. It has been developed since 2018 with the goal of enabling an open, interoperable ecosystem for MaaS. The TOMP-API is completely free and open source.

The vision for TOMP-API is "effortless mobility for everyone". In the future, every mobility customer should be able to call up, book, pay for, and use mobility offers (travel suggestions) from public transport and private providers with one single app from a MaaS provider. For this purpose, the TOMP-API defines the API interfaces along the MaaS process chain "search-book-use-pay".

The following parts are defined in the current TOMP-API version 1.3:

- operator information: basic information of the provider (operating times, price models, stops, vehicles).
- planning: inquiries (non-binding queries) and offers (binding, bookable offers).
- booking: booking of a trip based on the offer.
- trip execution and support: information during the trip, about individual legs, additional services, events, trip progress, etc.
- payment: payment information.

These parts must be implemented by the transport operators (TO), the MaaS provider (MP), or both, depending on their role. Instead of the TO, a central system (aggregator or enabler/converter) can also implement the API.

For a complete, interoperable, "roaming-ready" MaaS ecosystem, various details in the application of TOMP-API need to be precisely defined. In addition, other basics are needed that are still missing today, in particular:

---

[1] Nationale Datenvernetzungsinfrastruktur Mobilität (NADIM; en: national data-exchange infrastructure for mobility).

- A service catalogue (registry, directory) of the TO and MP, so that they and their services can be found.
- A legal or contractual framework, e.g., an association, so that the sale and procurement of services are contractually secured.
- Procedures and infrastructure for billing, clearing, and payment.
- Procedures for safeguarding (security, data protection).

These and other specifications will be provided in future versions of this document, in the form of a "profile" or "realization guide" for Switzerland.

For MaaS applications, the TOMP-API is currently the most advanced open standard. Only the OSDM standard, which originates from the public transport/railway sector, can be considered as an alternative, however OSDM has hardly been tested in intermodal MaaS scenarios to date.

3 Who is responsible?

Since its beginning, the TOMP-API is developed by a small Dutch team led by Edwin van den Belt, called the TOMP Working Group (TOMP WG). Access and membership to the group have not yet been precisely regulated. However, various international players, including MobilityData (USA) and FabMob (France), have established contacts with the TOMP WG and started a loose collaboration.

Currently (as of fall 2022), the group is striving to clarify and specify its foundations (legal form, membership, business rules, etc.). The intention of these changes is for the standard to become more widely and internationally supported in the future, and for the group to work and make decisions more effectively. Currently, Switzerland is represented by one collaborator of the SKI+ team; soon, in a more formal setup, the Swiss participation may shift to the FOT. Details have yet to be defined.

4 Important Links

No.	Description	Link
1	TOMP Working Group.	https://tomp-wg.org
2	GitHub repository used by the TOMP Working Group.	https://github.com/TOMP-WG/TOMP-API
3	TOMP-API specification based on openapi 3.0.0.	https://github.com/TOMP-WG/TOMP-API/blob/master/TOMP-API.yaml
4	TOMP-API specification in reader-friendly form on swaggerhub.com.	https://app.swaggerhub.com/apis/TOMP-API-WG/transport-operator_maas_provider_api
5	Homepage of the standards and profiles curated by the SKI and SKI+ teams, including this profile.	https://transportdatamanagement.ch/en/standards/

5 Technology stack and standards

The TOMP-API largely conforms to the REST (representational state transfer) paradigm. This includes:

- Definition of API functions (operations) using HTTP protocol with verbs (GET, POST) and URL paths,
- a valid OpenAPI 3.0.0 YAML specification [3], from which program code for client or server applications can be generated using tools such as Swagger Editor,
- definition of data formats in JSON with JSON Schema (integrated in the OpenAPI 3.0 YAML specification).

6 Usage

The TOMP-API is now receiving worldwide attention and is increasingly being used. The TOMP Working Group lists around 60 users or supporters. However, exact numbers and details are not known, and often these are likely to be only partial implementations.

We are not aware of an existing complete, interoperable MaaS system based on the TOMP-API yet.

7 Datasets/Services Switzerland

Currently, scenarios for MaaS and the booking workflow are being investigated and discussed in studies and proofs-of-concepts in Switzerland. This includes the efforts of the SKI+ team, which assessed the TOMP-API as described in the next section. These efforts resulted from an assignment by the FOT (Swiss Federal Office of Transport).

Further evaluations were done by private Swiss companies (e.g., ZüriMobil and YOURTAXI) who implemented parts of the TOMP-API, as well as international operators who use the TOMP-API in Swiss applications (e.g., Donkey Republic and TIER).

8 Assessment

The following assessment resulted from the analyses of TOMP by the SKI+ team[2]:

P1 international	+++	Originally from the Netherlands, but internationally supported.
P2 open	+++	Completely open project, open source, free license, open documentation, blueprints, etc. available. Open working group.
P3 simple	++	Comparably compact, simple formats, and intuitive workflows.
P4 established	++	Young standard, but rapidly increasing adoption.
P5 evolutionary	++	The TOMP WG applies common conventions and best practices, such as semantic versioning and deprecation.

[2] These eight principles can be found in the standardization concept of NADIM. Meaning: + / ++ / +++: low, medium, high fulfilment of the principle.

P6 of high quality	++	Parts of the documentation can be improved.
P7 compliant	++	TOMP WG seeks conformance with Transmodel and G*FS.
P8 interpretation-free	+	Experience to date shows that various points still need to be clarified and missing parts (such as registry, see above) need to be specified.

9 Conclusions

For MaaS applications, TOMP-API is currently the most advanced open standard. Only OSDM can be considered a competitor, however it has hardly been tested in MaaS scenarios yet.

10 Specification and recommendation

The long-term goal is to create an open MaaS system for Switzerland, curated by NADIM, that enables automated interoperability or roaming. The current architectural design envisions a hybrid model for this, encompassing individual providers being connected by central enabler/converter components of NADIM, while others offer their APIs themselves.

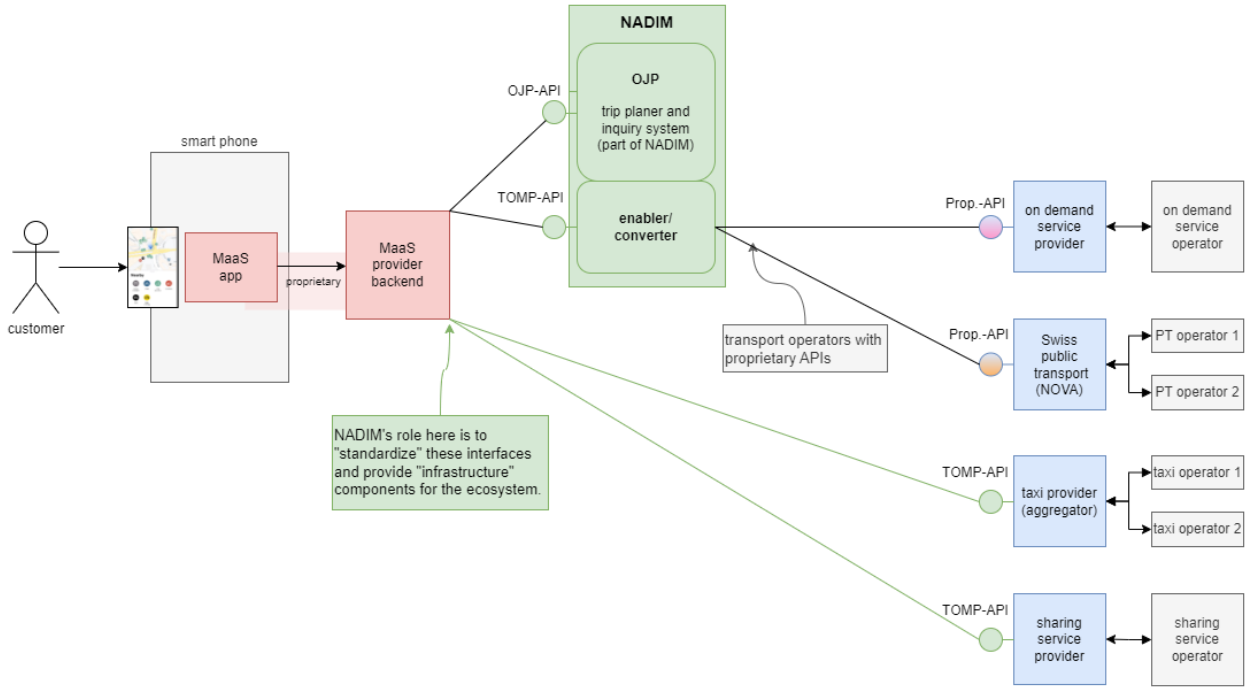

Figure 1: Hybrid model for distribution integration orchestrated by NADIM in Switzerland.

To make access to data and services as easy as possible for mobility intermediaries, all providers are to be callable via standardized APIs (green lollipops). The long-term goal is that all providers can be called automatically and without individual customization.

As a standard, we propose OJP for trip planning and TOMP-API for booking request, booking, usage and partial payment.

Further details will follow in future versions of this document and elsewhere.

11 TOMP-API – Swiss Profile

The reference and starting point of this document is the TOMP-API specification (version 1.3). This document is to be read together with the specification. It clarifies and details parts and offers restrictions to the original specification to improve interoperability.

Note that for certain functional requirements, as well as REST call and each of the calls' parameters we define whether it:

- **may** be included, i.e., it is optional to implement/user a REST call/feature or include a parameter of a REST call. If a provider implements the call and/or parameters, then the implementation must adhere to the description given in this profile
- **should** be included, i.e., it is highly recommended that a REST call/feature and/or a parameter is implemented. As above the implementation must adhere to this document
- **must** be included, i.e., the REST call and/or parameter must be included and adhere to the given document.

Note, that we can indicate a REST call as optional ("may"), but then require certain parameters as mandatory ("must") and vice versa.

11.1 Purpose and Scope

This profile supplements the existing TOMP standard with additional specifications and clarifications. The profile is intended to make the application of TOMP in Switzerland and within the future NADIM framework work as smooth as possible.

More specifically, whoever participates in NADIM will have to comply with certain requirements and obligations. These are expected to include the given profile, starting with version 1.0.

11.2 Security [must]

No.	Topic	Description	Class	Example
1	HTTPS	All TOMP-APIs must be secured with HTTPS (TLS). They must have a valid certificate issued by a trusted authority.	must	
2	Authentication/ Authorization	A common, simple method must be used for authentication / authorization (API Keys, Bearer Tokens or comparable).	must	

Future development is still unclear. Conceivable, but not yet defined, is the development of a central security or identity and access management infrastructure based on OAuth.

TODO: Details will follow in later versions of this document.

11.3 Service Catalogue [must]

We assume that a future infrastructure as described above will require a service catalogue in which the APIs of all participating providers and operators can be found. The catalogue should

hold all the necessary information such as URLs, standards and protocols so that the APIs can be called.

TODO: Details are still to be clarified and will follow in later versions of this document.

11.4 Geo Referencing [must]

No.	Topic	Description	Class	Example
1	Location information	The WGS 84 coordinate system shall be used for all location information (TOMP: "place"). lng and lat (longitude and latitude) are mandatory, alt (altitude) is optional.	must	
2	Determining coordinates	Coordinates of referenced objects must be determined and recorded using map services (e.g., by swisstopo (map.geo.admin.ch) or Open Street Map (www.openstreetmap.org) with an accuracy of 10 m or better. Coordinates of nearby or higher-level objects must not be used.	must	The coordinates of a taxi stand (stopping edge of the foremost taxi) at a train station must be precisely measured in the map service. It is not permissible to use the coordinates of the station (or its DiDok operating point) for this purpose.
3	Coordinate representation	Coordinates shall be given in decimal degrees. The degree-minutes-seconds representation is not allowed.	must	Bundesplatz Bern: correct: lat=46.947112, lng=7.444142, not allowed: lat=46°56′49.592″, lng=7°26′38.905″.
4	Accuracy / Resolution	lng and lat should be given to 5 or 6 decimal places (this corresponds to a resolution of about 1 m and 10 cm, respectively).	must	correct: lng=7.444142, not useful: lng=7.44414422034823094 (too high resolution, seemingly accurate) or lng=7.444 (too low resolution, thus inaccurate).
5	Reference (identifiers)	Where available, locations must also be referenced with ids from common directories, for example ids from DiDok or with a Swiss Location ID (SLOID).	must	
6	Calculating distances	For some applications, distances between points must be approximated. Distances allow to determine which point (location) is closest to a given point and thus could be meant.	must	

11.5 Date and Time [must]

No.	Topic	Description	Class	Example
1	Date / Time	Date-time timestamps are generally given in Zulu time (ISO 8601 formatted, pattern yyyy-MM-ddTHH:mm:ss.SSS, time zone UTC, with a "Z" for Zulu time at the end). Zulu time (UTC) is 1 hour after local time (Central European Time CET) in winter, 2 hours in summer (Central European Summer Time CEST).	must	2022-02-18T14:08:20.123Z or 2022-02-18T14:08:20Z for a local date February 18, 2022 and time 15:08:20 (CET).

11.6 HTTP parameters [must]

11.6.1 Accept-Language [must]

According to the specification, the user (client) can specify the desired language/localization here.

No.	Topic	Description	Class	Example
1	Language	The endpoint should also work and respond with "default" if an empty, invalid, or unknown value is used here.	must	
2	Language values	The values de, fr, it, en and the corresponding language can be supported.	may	

11.6.2 Compression [must]

No.	Topic	Description	Class	Example
1	Compression	The service must support compression in the transmission.	must	

11.6.3 API [must]

According to the specification, this parameter provides a hint about the content of API. The original specification specifies it as "TOMP or something else".

No.	Topic	Description	Class	Example
1	API	For the usage of this profile the value is set to "TOMP-CH".	must	

11.6.4 API-Version [must]

According to the specification, "the version of the API".

No.	Topic	Description	Class	Example
1	API version	The version is set to "0.4" (the value of the current profile).		

11.6.5 maas-id [may]

According to the specification "the Id of the sending MaaS provider".

No.	Topic	Description	Class	Example
1	Sending MaaS provider- ID	For the time being, the parameter can still be specified by the provider for its clients. In the future hybrid system (see above), this ID will be used to uniquely identify the operators. Based on this ID, the service provider can unambiguously and legally determine which client (MaaS providor) performed the call. The ID will probably be assigned by NADIM in the future and stored in a service catalogue.	may	

TODO: Details will follow in later versions of this document.

11.7 TOMP-API endpoints resp. operations

The following sections provide information on individual endpoints or operations of the TOMP-API. We follow the subdivision in the specification ([3], [4]).

11.7.1 Planning [must]

11.7.1.1 POST /planning/inquiries [may]

No.	Topic	Description	Class	Example
1		Returns "informative options for the given itinerary" according to the specification. May or may not be implemented. If not implemented, clients may use /planning/offers instead.	may	
2		If implemented, this endpoint must provide planning information, i.e., information provided by the respective system "to the best of its knowledge" ("best effort"), but	must	

No.	Topic	Description	Class	Example
		possibly inaccurate or provisional.		
3		The response times of the endpoint should be short. The system should not call any third-party APIs but work exclusively with its own trip planner algorithm and pre-loaded data.	should	• An on-demand provider could use this to show an approximate, possible, but non-binding connection with an on-demand bus, without any concrete dispatching (checking currently available vehicles). • A trip planner such as OJP could hereby perform a theoretical, hypothetical calculation based on preloaded provider operating hours and road route planning or air distance.

11.7.1.2 POST /planning/offers [must]

According to specification delivers "bookable offers for the given itinerary".

Usually, an offer contains one leg (a section with one transport company). In complex systems such as public transport, the offer may contain travel chains with multiple legs.

Every transport company participating in the system must offer this service. This is the single and central service for the "Booking Request" step.

No.	Topic	Description	Class	Example
1		The service must know and use its own control points and POIs completely with WGS 84 coordinates accurate to about 10 meters.	must	

Request:

No.	Topic	Description	Class	Example
1		The WGS-84 coordinates in the request must always be provided and primarily used to define locations (with 10 m precision or better): • **from:** the place where one enters or takes over a vehicle, or the entrance to the transportation system (e.g., a control barrier or gate).	must	

No.	Topic	Description	Class	Example
		• **to:** the place where you get out of or drop off a vehicle, or where you leave the transport system. The goal of these requirements is an accurate representation and footpath-routing to these locations in the MaaS apps.		
2		If available, IDs (e.g., StopPlaceRef or StopPointRef) in the stationId field shall be accepted and prioritized (and coordinates ignored).	must	
3		stationIds are not a required field according to the TOMP-API standard and may be omitted. The service should be able to provide the best possible response even if this is the case.	should	

Processing:

No.	Topic	Description	Class	Example
1		An operator may define control points using stationIds, but is not required to. If the operator uses stationIds, it must provide them via operator information > /operator/stations.	may	
2		If from and to are supplied with stationIds, the service must use them and calculate corresponding offers.	must	
3		If from and to are supplied with coordinates only, the operator must determine its best possible closest serving points and use them in the quote, so that no or only short additional transfers on foot are necessary. To do this, the operator must be able to approximate distances between the desired points (from, to) and its service points (see section on geo referencing). *Todo: In a later version it must be studied, if a service needs to be*	must	• An e-scooter sharing operator (free floating) should determine and offer a nearby scooter. This should be "on the right side", facing the destination, if possible. • A taxi company should identify and offer a serviceable point on the side of the road, or alternatively a well-suited boarding point, e.g., a taxi stand or similar nearby.

No.	Topic	Description	Class	Example
		provided that can map coordinates to stationId as a central service.		
4		Use the departureTime from the request as a priority. If this is missing, use arrivalTime. If this is also missing, calculate with departureTime = now.	must	

Response (201):

No.	Topic	Description	Class	Example
1		give the WGS-84 coordinates of the effective service points (from and to), not those of the request.	must	
2		fill in validUntil (validity period).	must	
3		validUntil shall be at least 5 minutes if possible.	should	
4		during the validity period (validUntil) the corresponding resources shall be blocked if possible. (The specification says "(expected to be) available", thus allowing the resources not to be blocked. However, this should be avoided if possible, as it can be annoying for users).	should	
5		If possible, only one leg should be included in the response (offer). For footpaths at the beginning/end, no legs are to be supplied. The array named “legs” should be left empty if possible and not used.	should	

11.7.2 booking [must]

No.	Topic	Description	Class	Example
1		TOMP requires a two-phase commit for the booking. These two steps/endpoints are mandatory and a core component for the booking workflow.	must	1. POST /booking: creates a binding booking based on the preceding offer with status PENDING. 2. POST/bookings/{id}/events: in this step, the binding booking is committed (with the operation “COMMIT”), cancelled (with the

No.	Topic	Description	Class	Example
				operation “CANCEL”), or another action is taken. In case of a commit, the booking changes its status to CONFIRMED.

11.7.2.1 POST /bookings [must]

Creates a booking (booking object) in "pending" status.

No.	Topic	Description	Class	Example
1		Must be implemented.	must	
2		This step must be preceded by the /planning/offers step. There, the id is delivered, which is to be used here.	must	
3		The booking id must be supplied (although not a required field by TOMP). The service must use the id to retrieve the preceding offer and create an exactly matching booking with the status PENDING.	must	
4		The other attributes (from, to, etc.) can be omitted from the request.	may	
5		The operator must accept the id from the previous step (offer from /planning/offers), if it is still valid (validUntil), and be able to convert it into a booking.	should	
6		IMPORTANT: The specification allows the booking to be different from the offer, e.g., the departure location may effectively be different than in the original offer. We consider this an unnecessary flexibility of the standard that has a high potential to cause confusion. An offer should be binding, i.e., an unmodifiable offer that becomes a legally binding contract through a commitment. We therefore stipulate: More strictly than in the specification, we require that the	must	

No.	Topic	Description	Class	Example
		booking (booking) in this step matches the offer (offer) of /planning/offers exactly, especially with respect to places and times.		

11.7.2.2 POST/bookings/{id}/events [must]

This endpoint is used to definitively confirm the booking returned in the previous step with a COMMIT or to cancel it with CANCEL, or to be otherwise used according to specification.

TODO: Not yet investigated and to be clarified in future versions:

- *Postponed commit.*
- *Callbacks, callback URLs for booking updates (or restriction to polling).*

11.7.3 trip execution [tbd]

A total of 9 services (operations) to support during the journey.

Presumably, GET /legs/{id} will have to be implemented as a minimum.

TODO: Which of them are mandatory to implement and how is still to be investigated in more detail.

11.7.4 operator information [tbd]

It is expected to be mandatory to implement:

GET /operator/meta

GET /operator/stations - if a provider uses stations (stationIds).

TODO: Further details need to be investigated and will follow in later versions of this document.

11.7.5 Payment [tbd]

TODO: Is currently still unclear, needs to be investigated further.