



UNLOCKING LARGE-SCALE ACCESS TO COMBINED MOBILITY THROUGH A EUROPEAN MAAS NETWORK.

Deliverable D2.6

Final Open API specification



This report is part of a project that has received funding by the European Union's Horizon 2020 research and innovation programme under grant agreement number 723314.

The content of this report reflects only the authors' view. The Innovation and Networks Executive Agency (INEA) is not responsible for any use that may be made of the information it contains.

Deliverable D2.6

Final Open API specification

Due date of deliverable: 31/05/2019

Actual submission date: 31/07/2019

Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/06/2017

Duration: 30 months

Document Control Sheet

Deliverable number:	D2.6
Deliverable responsible:	Softeco Sismat SRL
Work package:	WP2
Main editor:	Alessandro Barisone

Editor name	Organisation
Michal Certicky	Ceske Vysoke Uceni Technicke V Praze
Jan Hrnecir	Ceske Vysoke Uceni Technicke V Praze
Alessandro Barisone	Softeco Sismat SRL
Marco Gorini	Softeco Sismat SRL
Jose David Fernández	Mosaic Factor SL
Corrado Alesso	Urbannext SA
Vassilis Psaltopoulos	Institute of Communication and Computer Systems
Nikolaos Touser	Institute of Communication and Computer Systems

Document Revision History			
Modifications Introduced			
Version	Date	Reason	Editor
1.0	06/05/2019	First version	Marco Gorini
1.1	20/05/2019	First integration of partners contributions	Marco Gorini
1.2	28/06/2019	Integration of updates from partners	Alessandro Barisone
1.3	24/07/2019	Final revision	Alessandro Barisone
2.0	31/07/2019	Version ready for submission	Alessandro Barisone

Legal Disclaimer

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability to third parties for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law. © 2017 by IMOVE Consortium.

Executive Summary

Deliverable D.2.6 documents the IMOVE Final Open API Specification, the outcome of Task 2.3, running in Work Package 2.

The main objective of the task is to build on the design activities performed in T2.2 about the architecture and specification of the IMOVE system, outlined first in deliverable D2.2 and then enhanced in D2.6 and address with greater and proper detail a well-established and crucial area of the software enablers' framework: the interface for external mobile app backends to interact with IMOVE.

This interface is fundamental for the IMOVE ecosystem, because it has been designed to foster the integration with backends of mobile applications, provided by MaaS operators themselves or by independent 3rd party software providers (e.g. map providers, online resellers and startups). The final goal is to let end users to access MaaS services in a seamless and reliable way, interacting with the applications they already are accustomed to and opening up to third parties integrations.

Given its importance, this functionality is implemented in a dedicated software enabler named Open API itself (it was formerly referred to as B2B API in the earliest stages of the project). A collaborative document had been set up for partners to contribute, integrate and keep contents up to date, because this specification has evolved with the progress in the activities, according to the iterative methodology followed in Work package 2 and the continuous improvement of the other software enablers.

In the initial version of this specification, drafted in Deliverable D2.3, a technology baseline has been established, the subset of enablers' functionalities to be part of this API have been identified and an initial description of the interface in terms of requests and responses has been provided.

The Open API specification has then been kept updated to reflect the subsequent iterations of work package 2 tasks, reviews in the IMOVE reference data model identification (work package 3) and feedbacks from the Living Labs deployment phase. The present report documents the Final Open API specification, reflecting the overall results of Task 2.3 and providing the information reference and guidance for mobile app back-ends integration with the IMOVE software enablers framework.

Abbreviations and Acronyms

B2B	Business To Business
DoA	Description of Action
EC	European Commission
GA	Grant Agreement
LL	Living Lab
OAS	OpenAPI Specification
REST	Representative State Transfer
WP	Work Package

Table of Contents

1	Introduction.....	9
1.1	Work package 2	9
1.2	Task 2.3.....	11
1.3	Deliverable structure	11
2	Open API concept	12
3	Open API in IMOVE.....	13
3.1	IMOVE software enablers framework architecture	13
3.2	Open API design Process	15
4	IMOVE Open API specification	17
4.1	Technological guidelines.....	17
4.1.1	RESTful web services.....	17
4.1.2	HTTP Secure (HTTPS).....	17
4.1.3	JSON WEB TOKEN (JWT).....	17
4.2	IMOVE Open API endpoints	18
	Conclusions	50
	References	51

List of Figures

Figure 1 IMOVE software enablers concept.....	10
Figure 2 Work package 2 tasks	11
Figure 3 IMOVE software enablers interfaces.....	14
Figure 4. SwaggerHub online tool for OpenAPI editing and documentation.....	16

List of Tables

Table 1. Endpoint /identity	18
Table 2. Endpoint /tariff.....	19
Table 3. Endpoint /tracks	20
Table 4. Endpoint /places	21
Table 5. Endpoint /recurrentPlaces	22
Table 6. Endpoint /contextPreferences	23
Table 7. Endpoint /qualityServicePreferences	26
Table 8. Endpoint /transportTypePreferences	27
Table 9. Endpoint /userPreferences	28
Table 10. Endpoint /rankingAlternativeTrips.....	31
Table 11. Endpoint /preferredTransportModes.....	34
Table 12. Endpoint /routing	35
Table 13. Endpoint /availability	41
Table 14. Endpoint /shared	42
Table 15. Endpoint /shared/current	43
Table 16. Endpoint /ride.....	44
Table 17. Endpoint /tickets	45
Table 18. Endpoint /tickets/purchased	46
Table 19. Endpoint /notification	47
Table 20. Endpoint /gamificationData.....	48
Table 21. Endpoint /incentives	49

1 INTRODUCTION

The main goal of IMOVE project is to experiment and validate “Mobility as a service” models and operating conditions in different cities around Europe, supporting and running Living Labs. These are innovation ecosystems located in well-established geographic areas where novel services and research concepts can be tested and applied to a user community. Building such complex environments may start from green-field or evolve and reshape existing infrastructures and services with innovative business models and technologies. This process often impacts many stakeholders along specific value chains; in this context the knowledge base of academic institutions participating in IMOVE, the various local associate partners providing advanced mobility concepts and solutions (business models, guidelines, sustainability methods, etc.) along with the background expertise of industrial and ICT partners are the most important factors for a successful experimentation and validation.

IMOVE envisioned three fundamental areas of intervention to foster MaaS development, each one addressed in a specific work package and then adapted, implemented and evaluated across the Living Labs according to their peculiarities:

- **Scalability unlockers** in terms of business models and guidelines, regulatory aspects and policies to promote travel behaviour change and modal (Work Package 1)
- Specification of an information technology workflows and design and implementation of **software enabler** components targeting Mobility as a Service operations (WP 2)
- **Information exchange framework**, in terms of reference data model, management policies and analytics (WP 3)

1.1 WORK PACKAGE 2

As anticipated, activities in Work Package 2 pertain to the definition and implementation of innovative ICT components and software tools to support personal mobility needs and address the day to day challenges in effective and sustainable people transport in complex urban environments:

- Empower end users with tailored services and a better travel experience
- Develop innovative mobility schemes and foster integration among transport modes
- Smoothly operate transportation services back-office activities
- Promote sustainability in urban environments and enhance citizens quality of life

ICT technologies play a key role across the value chain of Mobility as a Service solutions not only because they offer services to the actors involved (end users, MaaS operators, transportation service providers) but also because they are essential in the implementation of the large scale interaction of these actors in their operational workflow.

At the same time ICT can be the baseline for more advanced services and business development both at the corporate and the public administration levels, leveraging on the insights gathered running the services to assist strategic planning and effective resource allocation.

IMOVE envisioned a suite of targeted ITS components, namely *software enablers*, to act as combinable building blocks to adapt to the business needs, the different technological maturity and the level of integration of the cooperating providers.

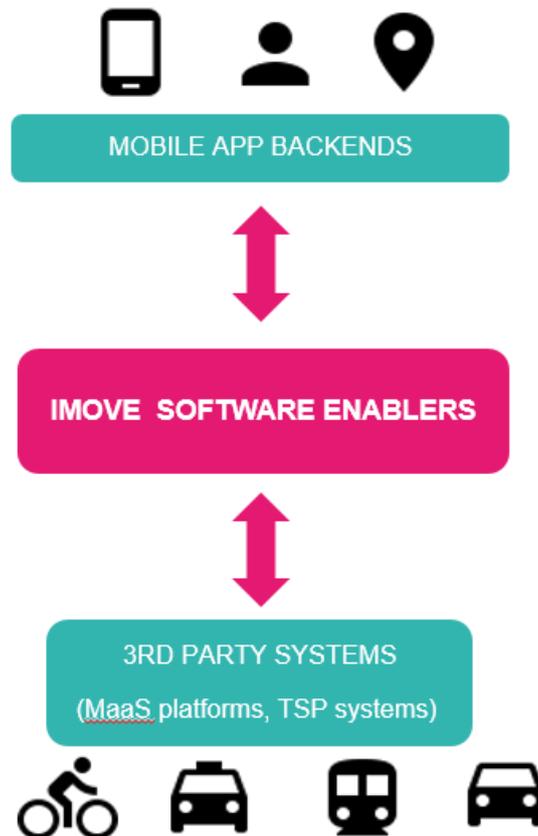


Figure 1 IMOVE software enablers concept

Work package 2 spans the complete software enablers' development process from the concept to the field operation: specific tasks have been devoted to single phases in the process:

- **Task 2.1:** Functional and technical specifications
- **Task 2.2:** Architecture and Design
- **Task 2.3:** Open API specification
- **Task 2.4:** SW Enablers Development
- **Task 2.5:** Integration & Technical support

The overall approach followed in the project and in WP2 as well is iterative, since the research field is highly innovative and the evolving nature of mobility landscape and transportation business in urban areas demands incremental adaptation of existing processes and continuous progress in the service level and user experience for customers.

Thus, task T2.3 about the Open API specification builds on the initial outcomes elaborated at the time being in task T2.1 about requirements and in T2.2 about architecture and design. These tasks will remain active and ongoing in parallel with logically subsequent tasks T2.4 and T2.5 about the enablers' implementation, deployment and Living lab testing phase.

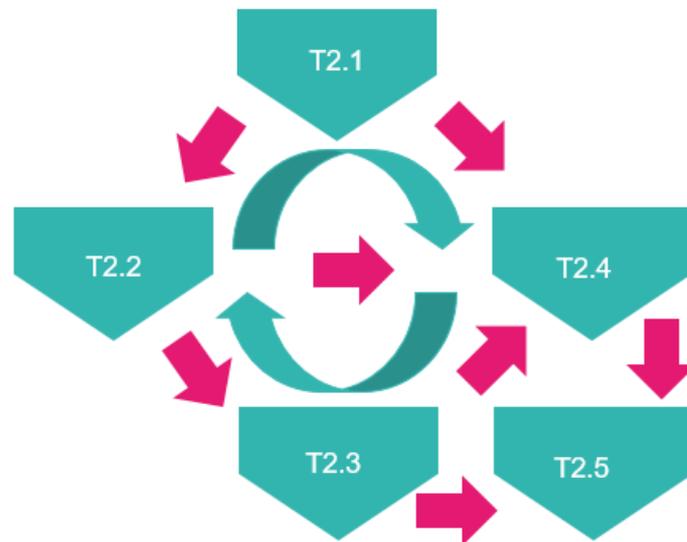


Figure 2 Work package 2 tasks

This will allow feedback and on-field expertise to become valuable sources of information for improvements and adaptations along the project lifetime, in order to refine the design and to ensure the innovative nature and effectiveness of the proposed solutions.

1.2 TASK 2.3

This document reports the initial achievements in the context of task T2.3, running from month 1 to month 28 and devoted to the IMOVE Open API specification for external systems (mobility app backends) to interact with the IMOVE framework.

This task iteratively relies on the progress of tasks 2.1 about technical specification and 2.2 about architecture and design, to define and evolve a comprehensive specification of the aforementioned software interface.

To this end, deliverables D2.1 ([1]) and D2.2 ([3]) served as baseline for T2.3 activities, but improvements in the outcomes of their corresponding tasks can be foreseen as this task advances.

1.3 DELIVERABLE STRUCTURE

This introductory chapter describes the work package and the task this deliverable refers to, and in Chapter 2 the Open API concept is summarized in general terms.

In the subsequent Chapter 3 the topic is presented in the scope of IMOVE, briefly reviewing the architectural picture envisioned so far and establishing a methodology to start the specification process and a roadmap for further enhancements.

Chapter 4 comprises the initial Open API specification.

The conclusion section starts with a review of the task and work package status at the time being and informs about aspects to be taken into account and addressed in the forthcoming work in this context. Furthermore, useful guidelines for the implementation and integration phase have been established right from the start and will be improved by running further iterations on this topic.

2 OPEN API CONCEPT

In information technology the expression **Open API** stands for the specification of a public application programming interface, to define and control the way a software system can provide services and perform integrated workflows in cooperation with other ICT platforms and tools.

In broader terms of business process analysis it can be viewed as a key instrument to promote the involvement of external stakeholders in the fruition of well-defined services, possibly involving heterogeneous operating environments and spanning across corporate boundaries. This empowers a provider in potentially reaching a wider customer base and in establishing and managing business relations in specific directions along the value chain, in particular:

- B2B services, designed to serve and interact with other businesses and companies
- B2C services, directly targeted towards end users.

Information technology plays a crucial role in service provision, leveraging on

- high speed Internet connections
- mobile communication networks
- personal mobile devices
- data storage and business intelligence tools

It benefits businesses with

- automation and / or reduced effort in transactions
- cost effectiveness
- scalability
- availability

The identification and proper development of these services can have direct impact on business turnover as they enable different revenue streams and sales channels, and at the same time it can optimize value added generation through operating cost reduction.

3 OPEN API IN IMOVE

IMOVE aims at supporting the diffusion and operation of the Mobility as a Service paradigm in different European cities, through the coordinated experimentation of the following action lines in the so called Living Labs:

- Business analysis and models
- Design and implementation of ICT tools and software services enabling and enhancing MaaS operation
- Data management and analytics

In relation to these aspects, Open API can be considered a cross-cutting concept, as this specification represents a technological interconnection protocol to exchange data and services between involved business parties.

However the following sections will explain how within IMOVE the expression Open API will refer to a specific subset of workflows and services implemented in the conceived *software enablers*.

3.1 IMOVE SOFTWARE ENABLERS FRAMEWORK ARCHITECTURE

IMOVE has not been envisioned as a platform, but as a flexible framework of components designed to easily integrate with technological services already implemented by MaaS operators and transportation providers. The aim is to enrich that environment providing specific functionalities in different areas of interest.

According to the original project description in the DoA ([1]) and the outcomes of the initial architecture specification and design reflected in deliverable D2.2 ([3]), the following software enablers have been identified within IMOVE:

1. **Identity Manager:** manages the end users IMOVE identity accounts.
2. **User Tariffs Manager:** deals with user subscriptions and transport services.
3. **Mobility Tracker:** collects user mobility-related information.
4. **Preferences Manager:** stores information about the user's preferences, habits, and needs.
5. **Notification Manager:** delivers to mobile apps backends the messages generated by other enablers.
6. **Roaming Manager:** enables the interconnection of multiple MaaS operators to provide service coverage outside user own subscription area and cross-border roaming services.
7. **Price Manager:** is responsible for retrieving information about the fares of local transport providers and apply pricing policies.
8. **Mobility Organizer:** provides means of transport choices and advanced mobility solutions for MaaS users.
9. **Booking Manager:** allows users to perform mobility service bookings and purchases.
10. **Incentives & Gamification Manager:** implements policies to reward travellers for MaaS services usage and incentivize specific sustainable mobility behaviours.
11. **Open API:** (formerly named B2B API) allows 3rd party mobile app backends that are interested in integrating the functionality provided by IMOVE into their offers. It also takes care of access control, security and communication monitoring.

Each software enabler is part of an interconnected ecosystem, as it implements complex workflows possibly involving different types of interactions:

- With other software enablers within IMOVE (*black arrows* in the following picture)
- With a MaaS operator ICT infrastructure, to gain access to transportation service providers IT platforms (*blue arrows*)

- With mobile application backends, to supply functionalities to app end users (red arrow). This interface corresponds to the Open API specification described in this document.

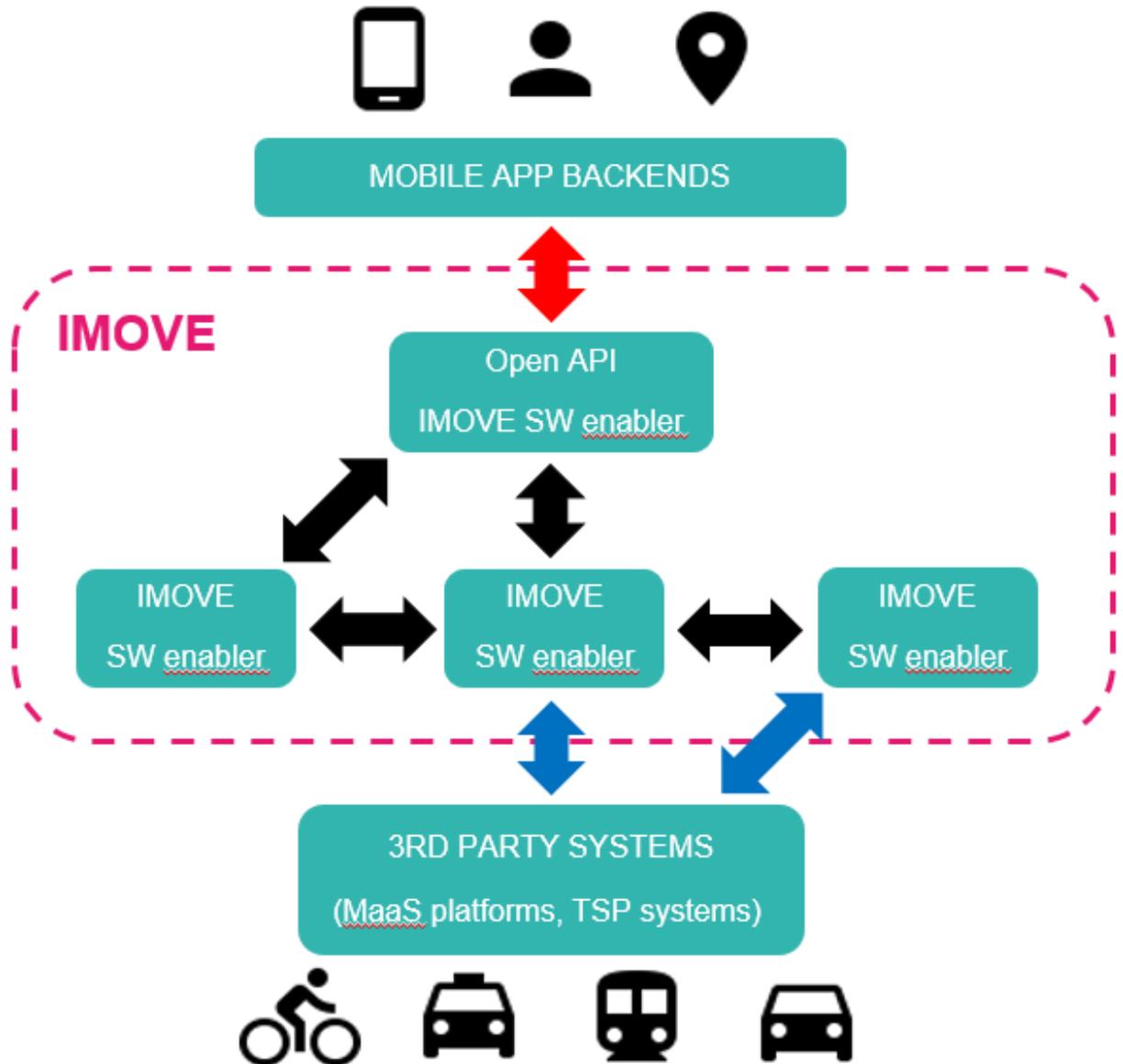


Figure 3 IMOVE software enablers interfaces

Focusing on mobile applications in the mobility sector, it should be noted that they are backed by different vendors and organizations:

- Private entities or public administration partners of IMOVE
- MaaS operators partners of IMOVE
- Commercial software or open source companies

In IMOVE a specific software enabler has been designed to target a well-established set of services to such a broad audience and it was referenced as B2B API in initial project phase and in the relevant documents.

Unfortunately, the term B2B was often misinterpreted by non-technical people, interpreting it as the business relationship between the MaaS provider and the transportation providers.

WP2 partners decided to renamed it to “Open API” for the sake of clarity and because this moniker proved to be more effective in presenting the concept to Living Labs partners and external stakeholders. It also better

conveys the main purpose of this enabler, embodying an interface layer and defining a coherent and comprehensive specification of data exchanges and procedures with mobile app backends.

The next section identifies the methodology followed in the design process, and the subsequent chapter presents the outcome that emerged from the architectural analysis and has been iteratively improved and refined leveraging also the feedbacks coming back from Living Labs experimentation.

3.2 OPEN API DESIGN PROCESS

The design of the Open API specification started downstream the initial system architecture drafted in deliverable D2.2 ([3]) which represented the initial synthesis of WP2 partners coordinated work. These activities have been carried out through several phone meetings and with the editing and progressive enhancement of a collaborative document where each participant has contributed with insights, comments and review.

This process continued in the subsequent project phases, and the document has been kept open to modifications and expansions to take advantage of further analysis, iterative revisions and feedbacks generated from the Living Labs,

In principle the Open API specification consists of:

- a selection of software enablers methods to be made available to mobile apps backends
- a harmonization pass to ensure they represent a coherent and self-contained set
- the identification of guidelines and technologies to address the scalability, security and reliability requirements

All these points fit well with the collaborative approach profitably followed in the architecture design task, so a new shared document has been set up for gathering and elaborating the information, and it constituted the main source of information to draft the Open API specification from, both in the initial and in the final version.

The technological aspects have been addressed in specific parts of the document, and have been reflected in section 4.1 of this deliverable.

After the preliminary design activities and the establishment of the technological baseline, the initial description of interface calls in terms of requests, responses and data structures definitions have been sketched out in the collaborative environment of Apiary, a web based tool for API definition and prototyping ([9]). This tool was previously used in WP2 for the specification of the IMOVE software enablers in task 2.2.

The interfaces have been expressed in the standard, language-agnostic OpenAPI format ([13]) (formerly known and still commonly referenced as swagger) which basically consist in a JSON object, which may be represented either in form of a JSON or YAML document.

The document has been edited with the SwaggerHub online tool and was versioned in a git repository.

The screenshot shows the SwaggerHub interface for editing an OpenAPI specification. On the left, a sidebar lists API resources: 'BOOKING_MANAGER' (with endpoints for GET /booking/availability, GET /booking/shared, and DELETE /booking/shared) and 'INCENTIVES_AND_GAMIFICATION MANAGER' (with endpoints for GET /incentives/gamificationData, GET /incentives/catalogue, and POST /incentives/requestIncentive). The main editor area shows a 'DELETE' endpoint for '/booking/shared' with a 405 'Invalid input' response. The right-hand panel displays a list of endpoints for 'Booking_Manager' (GET /booking/availability, GET /booking/shared, DELETE /booking/shared) and 'Incentives_And_Gamification Manager' (GET /incentives/gamificationData).

Figure 4. SwaggerHub online tool for OpenAPI editing and documentation

Details of the data entities that are part of the Open API have been refined in cooperation with work package 3 activities, where a reference data model for MaaS has been designed and iteratively improved.

4 IMOVE OPEN API SPECIFICATION

After the presentation of the Open API concept in ICT and the specification of its role in the high-level architectural view of IMOVE software enablers, this chapter comprises the specification of IMOVE Open API with the technological guidelines, the list of API endpoints provided and the textual description of their purpose and functionality.

4.1 TECHNOLOGICAL GUIDELINES

This section introduces principles and technologies underlying the Open API specification within IMOVE.

4.1.1 RESTFUL WEB SERVICES

Web services are modern software components that conform to a service-oriented architecture (SOA). They provide interoperability between information processing systems connected in a network (typically the Internet) through a communication protocol. They can be described and developed following different definition and implementation standards, built on lower level networking protocols (TCP/IP, HTTP) but introducing specific set of operations and protocols to send and receive data. Examples of these higher level messaging styles and protocols are SOAP (Simple Object Access Protocol) ([10]) and REST (REpresentational State Transfer) ([11]). SOAP uses a XML based message format transmitted over application layer protocols, usually HTTP.

According to its definition, REST is a set of architectural constraints to manage the exchange of resources and services via a request/response mechanism established on the properties and capabilities of the HTTP protocol. More in details, resources are identified by a URI (identifiers that express the protocol, address and endpoint information of a resource) and operations match HTTP request methods (e.g. GET, POST, PUT, DELETE).

Responses consist of an HTTP status code (e.g. 200 for a successful request, 404 for a resource not found) and possibly a body encoded in a well-known format such as HTML, XML but the typical choice nowadays is JSON. This is the acronym that stands for JavaScript Object Notation, an open standard text based format that allows object description through attribute/value pairs and specific notations for arrays and objects. The representation is human readable and is usually more compact than the XML provided one.

4.1.2 HTTP SECURE (HTTPS)

Secure HTTP (HTTPS) is an extension to HTTP protocol to ensure secure network data flow over the Internet. Communication is encrypted using Transport Layer Security (TLS) to protect the privacy and integrity of the data transmitted over the network. The trust between communicating peers is established by the use of digital certificates, issued by specific certificate authorities.

4.1.3 JSON WEB TOKEN (JWT)

JSON web token (JWT) ([12]) is an open standard to exchange in a compact form a set of claims between two parties. The specification determines mandatory and optional content as well as the structure and encoding of the pieces of information. On top of JWT several token encryption and exchange schemes have been defined.

Authentication and authorization features in complex web-based architectures are addressed by token inclusion in the request messages flowing in the system. An example of this is the so called bearer token authentication, a security scheme on which several authentication protocols rely on, which permits access to a resource only when attaching to request headers a valid token, issued at the successful end of an authentication workflow performed by an identity service.

4.2 IMOVE OPEN API ENDPOINTS

Table 1. Endpoint /identity

Endpoint		/identity
HTTP Method	PUT	
Description	Creates a new IMOVE user	
Software enabler	Identity manager	
Authentication	Bearer token	
Request body		
<pre>{ "phone" : "+123456789", "email" : "a@a.com", }</pre>		
Response code	200	Successful operation
Response	<pre>{ "imove_user_id": "", }</pre>	
HTTP Method	GET	
Description	Fetch the information of a user	
Software enabler	Identity Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre>{ "imove_user_id": "", "phone" : "+123456789", "email" : "" }</pre>	
HTTP Method	PATCH	
Description	Update the information of a user	
Software enabler	Identity manager	
Authentication	Bearer token	
Request body		
<pre>{ "imove_user_id": "", "phone" : "+123456789", "email" : " a@a.com" }</pre>		
Response code	204	Successful operation

Table 2. Endpoint /tariff

Endpoint		/tariff
HTTP Method	PUT	
Description	Create a new subscription. A relation to an existing user is required.	
Software enabler	User tariff manager	
Authentication	Bearer token	
Request body	<pre>{ "plan" : "Monthly", "payment" : "CreditCard", "recurring_on" : "2018-01-15T21:38:51+00:00", "created" : "2018-01-15T21:38:51+00:00", }</pre>	
Response code	200	Successful operation
Response	<pre>{ "subscriptionId": "" }</pre>	
HTTP Method	GET	
Description	Obtains info for an existing subscription.	
Software enabler	User tariff manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre>{ "plan" : "Monthly", "payment" : "CreditCard", "recurring_on" : "2018-01-15T21:38:51+00:00", "created" : "2018-01-15T21:38:51+00:00", "quota": { sharing_credit: {} pt_credit: {} taxi_credit: {} } }</pre>	

Table 3. Endpoint /tracks

Endpoint		/tracks
HTTP Method	POST	
Description	Send tracks (geocoordinates + time stamp) provided by a GPS of a user device. This tracks will be used to profile the user and provide better services within the IMOVE platform	
Software enabler	Mobility Tracker	
Authentication	Bearer token	
Request body		
<pre>{ "data": [{ "coordinates": [45.065, 7.694], "time_stamp": "2018-01-15T21:37:51+00:00" }, { "coordinates": [45.0657, 7.6943], "time_stamp": "2018-01-15T21:38:51+00:00" }] }</pre>		
Response code	200	Successful operation
Response	<pre>{ "success": true }</pre>	

Table 4. Endpoint /places

Endpoint		/places
HTTP Method	GET	
Description	Get a list of places where a given user has been	
Software enabler	Mobility Tracker	
Authentication	Bearer token	
Parameters Name	Type	Description
start	date	All records that have started before the provided timestamp will be ignored
end	date	All the records that have ended after the provided timestamp will be ignored
last_item	string	Key of the last item got from a previous call. Provide this in order to get the next items.
area	array of locations	Bounding box for filtering results
items	integer	Maximum number of items to be returned.
Response code	200	Successful operation
Response	<pre>{ "data": [{ "id": "kvhcvkdhjvakhjdvadhjvdhajkhvadkjdvhakj", "description": "Bus stop.", "location": [45.065, 7.694], "periode": ["2018-01-15T21:39:51+00:00", "2018-01-15T22:11:41+00:00"], },], "last_item": "kvhcvkdhjvakhjdvadhjvdhajkhvadkjdvhakj" }</pre>	

Table 5. Endpoint /recurrentPlaces

Endpoint		/recurrentPlaces	
HTTP Method	GET		
Description	Get a list of places that a given user visits frequently. A part of the description of the places, a schedule describing the patterns of visits by the user is returned.		
Software enabler	Mobility Tracker		
Authentication	Bearer token		
Parameters Name	Type	Description	
start	date	All records that have started before the provided timestamp will be ignored	
end	date	All the records that have ended after the provided timestamp will be ignored	
last_item	string	Key of the last item got from a previous call. Provide this in order to get the next items.	
area	string	Bounding box for filtering results	
items	integer	Maximum number of items to be returned.	
Response code	200	Successful operation	
Response	<pre> { "data": [{ "id": 1, "description": "mosaic factor", "startTime": "2018-06-01T01:00:00.000Z", "endTime": "2018-06-01T02:00:00.000Z", "latitude": 41.406285, "longitude": 2.1927889, "arrival": "10:00", "departure": "22:00", "monthDay": 1, "weekDay": 0 }, { "id": 2, "description": "mosaic factor", "startTime": "2018-06-01T02:00:00.000Z", "endTime": "2018-06-01T03:00:00.000Z", "latitude": 41.406285, "longitude": 2.1927889, "arrival": "10:00", "departure": "22:00", "monthDay": 1, "weekDay": 0 }] } </pre>		

Table 6. Endpoint /contextPreferences

Endpoint	/contextPreferences
HTTP Method	POST
Description	Set context (distance, traffic, weather) preferences
Software enabler	Preference Manager
Authentication	Bearer token
Request body	
<pre> { "distances_transport_preferences": { "LONG": [{ "distance": [0, 10], "transport_mode": "TRAIN", "ranking": 0.84 }, { "distance": [0, 10], "transport_mode": "BUS", "ranking": 0.90 }, { "distance": [0, 10], "transport_mode": "CAR", "ranking": 0.54 }, { "distance": [0, 10], "transport_mode": "BIKE", "ranking": 0.20 }], "MEDIUM": [{ "distance": [11, 20], "transport_mode": "TRAIN", "ranking": 0.05 }, { "distance": [11, 20], "transport_mode": "BUS", "ranking": 0.39 }, { "distance": [11, 20], "transport_mode": "CAR", "ranking": 0.87 }, { "distance": [11, 20], "transport_mode": "BIKE", "ranking": 0.46 }] } } </pre>	

```

],
"SHORT":
[
  {
    "distance": [21, 30],
    "transport_mode": "TRAIN",
    "ranking": 0.05
  },
  {
    "distance": [21, 30],
    "transport_mode": "BUS",
    "ranking": 0.39
  },
  {
    "distance": [21, 30],
    "transport_mode": "CAR",
    "ranking": 0.87
  },
  {
    "distance": [21, 30],
    "transport_mode": "BIKE",
    "ranking": 0.46
  }
]
},
"transport_preference_by_weather":
{
  "RAINY":
  [
    {
      "transport_mode": "TRAIN",
      "ranking": 0.92
    },
    {
      "transport_mode": "BUS",
      "ranking": 0.80
    },
    {
      "transport_mode": "CAR",
      "ranking": 0.82
    },
    {
      "transport_mode": "BIKE",
      "ranking": 0.05
    }
  ]
}

```

Response code	200	Successful operation
Response	{ "success": true }	
HTTP Method	GET	
Description	Get user context (distance, weather and traffic) preferences.	
Software enabler	Preference Manager	
Authentication	Bearer token	
Response code	200	Successful operation

Response	
	<pre> { "transport_preference_by_distance": { "LONG": [{ "distance": [0, 10], "transport_mode": "TRAIN", "ranking": 0.84 }, { "distance": [0, 10], "transport_mode": "BUS", "ranking": 0.90 }, { "distance": [0, 10], "transport_mode": "CAR", "ranking": 0.54 }, { "distance": [0, 10], "transport_mode": "BIKE", "ranking": 0.20 }] }, "transport_preference_by_traffic": { "MEDIUM": [{ "transport_mode": "TRAIN", "ranking": 0.84 }, { "transport_mode": "BUS", "ranking": 0.46 }, { "transport_mode": "CAR", "ranking": 0.87 }, { "transport_mode": "BIKE", "ranking": 0.61 }] }, "transport_preference_by_weather": { "RAINY": [{ "transport_mode": "TRAIN", "ranking": 0.92 }, { "transport_mode": "BUS", </pre>

	<pre> "ranking": 0.80 }, { "transport_mode": "CAR", "ranking": 0.82 }, { "transport_mode": "BIKE", "ranking": 0.05 }], } </pre>
--	--

Table 7. Endpoint /qualityServicePreferences

Endpoint		/qualityServicePreferences	
HTTP Method	POST		
Description	Set quality service (cheap, fast, comfortable) preferences		
Software enabler	Preference Manager		
Authentication	Bearer token		
Request body			
<pre> { "cheap_vs_fast": 0.85, "cheap_vs_comfort": 0.11, "fast_vs_comfort": 0.65 } </pre>			
Response code	200	Successful operation	
Response	<pre> { "success": true } </pre>		
HTTP Method	GET		
Description	Get user quality of service preferences.		
Software enabler	Preference Manager		
Authentication	Bearer token		
Response code	200	Successful operation	
Response	<pre> { "cheap_vs_fast": 0.85, "cheap_vs_comfort": 0.11, "fast_vs_comfort": 0.65 } </pre>		

Table 8. Endpoint /transportTypePreferences

Endpoint		/transportTypePreferences	
HTTP Method	POST		
Description	Set transport type (public, private, shared) preferences		
Software enabler	Preference Manager		
Authentication	Bearer token		
Request body			
<pre>{ "private_vs_public": 0.72, "public_vs_shared": 0.33, "shared_vs_private": 0.90 }</pre>			
Response code	200	Successful operation	
Response	<pre>{ "success": true }</pre>		
HTTP Method	GET		
Description	Get user transport type preferences.		
Software enabler	Preference Manager		
Authentication	Bearer token		
Response code	200	Successful operation	
Response	<pre>{ "private_vs_public": 0.72, "public_vs_shared": 0.33, "shared_vs_private": 0.90 }</pre>		

Table 9. Endpoint /userPreferences

Endpoint	/userPreferences	
HTTP Method	GET	
Description	Get all user preferences regarding distance, traffic, weather (contextual preferences) and (quality of service preferences) and (transport type preferences).	
Software enabler	Preference Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre> { "transport_preference_by_distance": { "LONG": [{ "distance": [0, 10], "transport_mode": "TRAIN", "ranking": 0.84 }, { "distance": [0, 10], "transport_mode": "BUS", "ranking": 0.90 }, { "distance": [0, 10], "transport_mode": "CAR", "ranking": 0.54 }, { "distance": [0, 10], "transport_mode": "BIKE", "ranking": 0.20 }], "MEDIUM": [{ "distance": [11, 20], "transport_mode": "TRAIN", "ranking": 0.05 }, { "distance": [11, 20], "transport_mode": "BUS", "ranking": 0.39 }, { "distance": [11, 20], "transport_mode": "CAR", "ranking": 0.87 }, { "distance": [11, 20], </pre>	

	<pre> "transport_mode": "BIKE", "ranking": 0.46 },], "SHORT": [{ "distance": [21, 30], "transport_mode": "TRAIN", "ranking": 0.05 }, { "distance": [21, 30], "transport_mode": "BUS", "ranking": 0.39 }, { "distance": [21, 30], "transport_mode": "CAR", "ranking": 0.87 }, { "distance": [21, 30], "transport_mode": "BIKE", "ranking": 0.46 }]], "transport_preference_by_traffic": { "HIGH": [{ "transport_mode": "TRAIN", "ranking": 0.95 }, { "transport_mode": "BUS", "ranking": 0.39 }, { "transport_mode": "CAR", "ranking": 0.42 }, { "transport_mode": "BIKE", "ranking": 0.46 }], "MEDIUM": [{ "transport_mode": "TRAIN", "ranking": 0.84 }, { "transport_mode": "BUS", "ranking": 0.46 }], } </pre>
--	--

	<pre> { "transport_mode": "CAR", "ranking": 0.87 }, { "transport_mode": "BIKE", "ranking": 0.61 }], "LOW": [{ "transport_mode": "TRAIN", "ranking": 0.69 }, { "transport_mode": "BUS", "ranking": 0.85 }, { "transport_mode": "CAR", "ranking": 0.90 }, { "transport_mode": "BIKE", "ranking": 0.70 }] }, "transport_preference_by_weather": { "RAINY": [{ "transport_mode": "TRAIN", "ranking": 0.92 }, { "transport_mode": "BUS", "ranking": 0.80 }, { "transport_mode": "CAR", "ranking": 0.82 }, { "transport_mode": "BIKE", "ranking": 0.05 }], "SUNNY": [{ "transport_mode": "TRAIN", "ranking": 0.75 }, { "transport_mode": "BUS", "ranking": 0.85 }] } </pre>
--	--

	<pre> }, { "transport_mode": "CAR", "ranking": 0.94 }, { "transport_mode": "BIKE", "ranking": 0.60 }], "COLD": [{ "transport_mode": "TRAIN", "ranking": 0.89 }, { "transport_mode": "BUS", "ranking": 0.85 }, { "transport_mode": "CAR", "ranking": 0.90 }, { "transport_mode": "BIKE", "ranking": 0.41 }] }, "cheap_vs_fast": 0.3, "cheap_vs_comfortable": 0.75, "fast_vs_comfortable": 0.9, "private_vs_public": 0.2, "public_vs_shared": 0.5, "shared_vs_private": 0.8, "budget": { "amount": 34.5, "currency": "EUR" } } </pre>
--	--

Table 10. Endpoint /rankingAlternativeTrips

Endpoint	/rankingAlternativeTrips
HTTP Method	POST
Description	Get a list of given trips sorted by ranking according to (contextual, quality of service and transport type) preferences
Software enabler	Preference Manager
Authentication	Bearer token
Request body	

```

{
  "trips" :
  [
    {
      "trip" :
      {
        "trip_origin": [45.065, 7.694],
        "trip_destination": [45.074, 7.614],
        "legs":
        [
          {
            "leg_origin" : [45.065, 7.694],
            "leg_destination" : [45.063, 7.678],
            "leg_start_time" : "2018-01-15T21:36:59+00:00",
            "leg_end_time" : "2018-01-15T21:39:51+00:00",
            "mode_of_transport" : "BIKE",
            "transport_provider_id" : null
          },
          {
            "leg_origin" : [45.063, 7.678],
            "leg_destination" : [45.074, 7.614],
            "leg_start_time" : "2018-01-15T21:41:59+00:00",
            "leg_end_time" : "2018-01-15T22:19:51+00:00",
            "mode_of_transport" : "METRO",
            "transport_provider_id" : ""
          }
        ],
        "trip_price" :
        {
          "amount" : 4.0,
          "currency" : "EUR"
        }
      }
    },
    {
      "trip" :
      {
        "trip_origin": [45.065, 7.694],
        "trip_destination": [45.074, 7.614],
        "legs":
        [
          {
            "leg_origin" : [45.065, 7.694],
            "leg_destination" : [45.074, 7.614],
            "leg_start_time" : "2018-01-15T21:36:59+00:00",
            "leg_end_time" : "2018-01-15T21:39:51+00:00",
            "mode_of_transport" : "BUS",
            "transport_provider_id" : "ALSA"
          }
        ],
        "trip_price" :
        {
          "amount" : 4.5,
          "currency" : "EUR"
        }
      }
    }
  ],
  "by_CP" : true,

```

<pre>"by_QoS" : true, "by_TT" : true } </pre>		
Response code	200	Successful operation
Response	<pre>{ "ranked_trips" : [{ "trip" : { "trip_origin": [45.065, 7.694], "trip_destination": [45.074, 7.614], "legs": [{ "leg_origin" : [45.065, 7.694], "leg_destination" : [45.063, 7.678], "leg_start_time" : "2018-01-15T21:36:59+00:00", "leg_end_time" : "2018-01-15T21:39:51+00:00", "mode_of_transport" : "BIKE", "transport_provider_id" : null }, { "leg_origin" : [45.063, 7.678], "leg_destination" : [45.074, 7.614], "leg_start_time" : "2018-01-15T21:41:59+00:00", "leg_end_time" : "2018-01-15T22:19:51+00:00", "mode_of_transport" : "METRO", "transport_provider_id" : "" }] }, "trip_price" : { "amount" : 4.0, "currency" : "EUR" } }, "rank" : 6], { "trip" : { "trip_origin": [45.065, 7.694], "trip_destination": [45.074, 7.614], "legs": [{ "leg_origin" : [45.065, 7.694], "leg_destination" : [45.074, 7.614], "leg_start_time" : "2018-01-15T21:36:59+00:00", "leg_end_time" : "2018-01-15T21:39:51+00:00", </pre>	

	<pre> "mode_of_transport" : "BUS", "transport_provider_id" : "ALSA" }], "trip_price" : { "amount" : 4.5, "currency" : "EUR" } }, "rank" : 3 }] } </pre>
--	---

Table 11. Endpoint /preferredTransportModes

Endpoint		/preferredTransportModes	
HTTP Method	GET		
Description	Get a list of transport modes sorted by ranking according to contextual values of (distance, weather, traffic) preferences		
Software enabler	Preference Manager		
Authentication	Bearer token		
Parameters Name	Type	Description	
distance_enum	string	Set a type of distance among: LONG, MEDIUM, and SHORT	
weather_enum	string	Set weather among: SUNNY, RAINY, etc	
traffic_enum	string	Set a type of traffic among: HIGH, MEDIUM, and LOW	
Response code	200	Successful operation	
Response	<pre> { "preferred_transport_mode": [{ "transport_mode": "TRAIN", "rank": 0.95 }, { "transport_mode": "BUS", "rank": 0.68 }, { "transport_mode": "BIKE", "rank": 0.46 }] } </pre>		

Table 12. Endpoint /routing

Endpoint		/routing
HTTP Method	POST	
Description	Composes and returns a list of proposed journey plans combining various transport services, taking into account routing request, user's preferences and active subscriptions. Each proposed plan consists of a list of legs, each one served by exactly one transport provider.	
Software enabler	Mobility Organizer	
Authentication	Bearer token	
Request body		
<pre>{ "origin": { "lat": 53.447332, "lon": -2.27787 }, "destination": { "lat": 53.496903, "lon": -2.245515 }, "settings": { "allowedTransportModes": ["BUS", "METRO", "SHARED_BIKE", "TROLLEYBUS", "FERRY", "WALK", "BIKE", "TAXI", "TRAM", "TRAIN", "CAR"] }, "departureDateTime": "2019-06-20T11:15:00+00:00" }</pre>		
Response code	200	Successful operation
Response body	<pre>{ "id": "b242b057bfc74de7a08b0f20dba6e706", "request": { "origin": { "lat": 53.447332, "lon": -2.27787 }, "destination": { "lat": 53.496903, "lon": -2.245515 } } }</pre>	

	<pre> }, "settings": { "allowedTransportModes": ["SHARED_BIKE", "CAR", "WALK", "TRAIN", "METRO", "BUS", "TROLLEYBUS", "FERRY", "TAXI", "BIKE", "TRAM"], "cyclingSpeedKmph": 15, "bikeType": "OTHER", "avoidTraffic": 1, "avoidClimbs": 1, "avoidStairs": false, "accessibilityOptions": [] }, "departureDateTime": "2019-06-20T11:15:00Z" }, "status": "OK", "plans": [{ "id": 7, "start": { "lat": 53.447332, "lon": -2.27787, "name": "Origin", "dateTime": "2019-03-05T12:24:30+01:00", "stopID": "ORIG" }, "end": { "lat": 53.496903, "lon": -2.245515, "name": "Destination", "dateTime": "2019-03-05T12:56:35+01:00", "stopID": "DEST" }, "stats": { "cost": { "value": 7.33556604385376, "currency": "GBP" }, "distanceMeters": 7787, "durationSeconds": 1925, "physicalEffortKj": 225, "emissionsGramsCO2": 536 } }] </pre>
--	---

	<pre> }, "legs": [{ "id": 0, "start": { "lat": 53.447332, "lon": -2.27787, "name": "Origin", "dateTime": "2019-03-05T12:24:30+01:00", "stopID": "ORIG" }, "end": { "lat": 53.45127, "lon": -2.27773, "name": "Firswood, Firswood Station (Manchester Metrolink)", "dateTime": "2019-03-05T12:38:00+01:00", "stopID": "\$null\$9400ZZMAFIR1" }, "stats": { "distanceMeters": 825, "durationSeconds": 810 }, "mode": "WALK", "steps": [{ "start": { "lat": 53.447332, "lon": -2.27787, "name": "Origin", "dateTime": "2019-03- 05T12:24:30+01:00", "stopID": "ORIG" }, "end": { "lat": 53.447311, "lon": -2.27778, "dateTime": "2019-03-05T12:24:36+01:00" }, "via": [], "distanceMeters": 6, "durationSeconds": 6 }, ...] }, { "id": 1, "start": { "lat": 53.45127, "lon": -2.27773, </pre>
--	---

	<pre> "name": "Firswood, Firswood Station (Manchester Metrolink)", "dateTime": "2019-03-05T12:38:00+01:00", "stopID": "\$null\$9400ZZMAFIR1" }, "end": { "lat": 53.47838, "lon": -2.24302, "name": "Manchester City Centre, St Peters Square (Manchester Metrolink)", "dateTime": "2019-03-05T12:50:00+01:00", "stopID": "\$null\$9400ZZMASTP2" }, "stats": { "distanceMeters": 4169, "durationSeconds": 720 }, "mode": "TRAM", "steps": [{ "start": { "lat": 53.45127, "lon": -2.27773, "name": "Firswood, Firswood Station (Manchester Metrolink)", "dateTime": "2019-03- 05T12:38:00+01:00", "stopID": "\$null\$9400ZZMAFIR1" }, "end": { "lat": 53.47838, "lon": -2.24302, "name": "Manchester City Centre, St Peters Square (Manchester Metrolink)", "dateTime": "2019-03- 05T12:50:00+01:00", "stopID": "\$null\$9400ZZMASTP2" }, "via": [{ "lat": 53.46187, "lon": -2.27711, "name": "Trafford Bar, Trafford Bar (Manchester Metrolink)", "dateTime": "2019-03- 05T12:41:00+01:00", "stopID": "\$null\$9400ZZMATRA1" }, { "lat": 53.46996, "lon": -2.26768, </pre>
--	---

	<pre> "name": "Cornbrook, Cornbrook (Manchester Metrolink)", "dateTime": "2019-03- 05T12:44:00+01:00", "stopID": "\$null\$9400ZZMACRN1" }, { "lat": 53.47476, "lon": -2.25018, "name": "Manchester City Centre, Deansgate-Castlefield (Manchester Metrolink)", "dateTime": "2019-03- 05T12:47:00+01:00", "stopID": "\$null\$9400ZZMAGMX1" }], "distanceMeters": 4169, "durationSeconds": 720, "instruction": { "manoeuvre": "BOARD_PT" } }, "ptDetails": { "lineId": "\$null\$MET: 5:I:", "lineName": "5", "tripId": "\$null\$Trip057921" } }, { "id": 2, "start": { "lat": 53.47838, "lon": -2.24302, "name": "Manchester City Centre, St Peters Square (Manchester Metrolink)", "dateTime": "2019-03-05T12:50:00+01:00", "stopID": "\$null\$9400ZZMASTP2" }, "end": { "lat": 53.496903, "lon": -2.245515, "name": "Destination", "dateTime": "2019-03-05T12:56:35+01:00", "stopID": "DEST" }, "stats": { "distanceMeters": 2793, "durationSeconds": 395 }, "mode": "TAXI", </pre>
--	--

	<pre> "steps": [{ "start": { "lat": 53.47838, "lon": -2.24302, "name": "Manchester City Centre, St Peters Square (Manchester Metrolink)", "dateTime": "2019-03- 05T12:50:00+01:00", "stopID": "\$null\$9400ZZMASTP2" }, "end": { "lat": 53.478765, "lon": -2.24252, "dateTime": "2019-03-05T12:50:07+01:00" }, "via": [], "distanceMeters": 54, "durationSeconds": 7, "instruction": { "manoeuvre": "LEAVE_PT" } }, { "start": { "lat": 53.478765, "lon": -2.24252, "dateTime": "2019-03-05T12:50:07+01:00" }, "end": { "lat": 53.478832, "lon": -2.242665, "dateTime": "2019-03-05T12:50:08+01:00" }, "via": [], "distanceMeters": 12, "durationSeconds": 1 }, ...] }], "boundingBox": { "topLeft": { "lat": 53.496903, "lon": -2.280049 }, "bottomRight": { "lat": 53.447311, "lon": -2.23595 } } </pre>
--	---

	<pre> }, "type": "WALK_PT", "nearestDepartures": ["2019-03-05T12:24:30+01:00", "2019-03-05T12:24:30+01:00"] }, ...], "creationTimestamp": "2019-06-20T13:10:50.342+01:00" } </pre>
--	---

Table 13. Endpoint /availability

Endpoint		/availability
HTTP Method	GET	
Description	Returns the various mobility options and services for a given location	
Software enabler	Booking Manager	
Authentication	Bearer token	
Parameters name	Type	Description
position	location	Coordinates to look for services around
Response code	200	Successful operation
Response	<pre> { "vehicles": [{ "id": "12345", "provider": "ShareCar", "type": "car", "plate": " XX000YY ", "location": [45.065, 7.694] }], "stations": [{ "id": "", "provider": "ShareCar", "type": "car", "freeSpots": 1, "freeVehicles": 2, "location": [45.065, 7.694] }] } </pre>	

Table 14. Endpoint /shared

Endpoint		/shared	
HTTP Method	POST		
Description	Creates a reservation for a car/scooter/bike (or, more generally, any shared vehicle). Returns the details of the reservation		
Software enabler	Booking Manager		
Authentication	Bearer token		
Request body	<pre>{ "vehicle": { "id": "12345", "provider": "ShareCar", "plate": "XX000YY" } }</pre>		
Response code	200	Successful operation	
Response	<pre>{ "reservation_id": "xyz12345", "vehicle": { "id": "12345", "provider": "ShareCar", "type": "Car", "location": [45.065, 7.694], }, "start_time": "2018-01-15T16:00:00+00:00", "end_time": "2018-01-15T18:00:00+00:00", "status": "reserved/cancelled/running/finished" }</pre>		
HTTP Method	GET		
Description	Retrieves details of a reservation		
Software enabler	Booking Manager		
Authentication	Bearer token		
Parameters Name	Type	Description	
reservationId	integer	Identifier of the reservation	
Response code	200	Successful operation	
Response	<pre>{ "reservation_id": "xyz12345", "vehicle": { "id": "12345", "provider": "ShareCar", "type": "Car", "location": [45.065, 7.694], }, "start_time": "2018-01-15T16:00:00+00:00", "end_time": "2018-01-15T18:00:00+00:00", "status": reserved/cancelled/running/finished" }</pre>		
HTTP Method	DELETE		

Description	Cancels a running reservation for a car/scooter/bike (or, more generally, any shared vehicle)	
Software enabler	Booking Manager	
Authentication	Bearer token	
Parameters Name	Type	Description
reservationId	integer	Identifier of the reservation
Response code	204	Successful operation

Table 15. Endpoint /shared/current

Endpoint	/shared/current	
HTTP Method	GET	
Description	Returns the list of running reservations for the user. There could be active reservations on multiple providers.	
Software enabler	Booking Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre>[{ "reservation_id": "xyz12345", "vehicle": { "id": "12345", "provider": "ShareCar", "type": "Car", "location": [45.065, 7.694], }, "start_time": "2018-01-15T16:00:00+00:00", "end_time": "2018-01-15T18:00:00+00:00", "status": reserved/cancelled/running/finished" }, { "reservation_id": "xyz12346", "vehicle": { "id": "54321", "provider": "BikeNBike", "type": "Bike", "location": [45.065, 7.694], }, "start_time": "2018-01-15T18:00:00+00:00", "end_time": "2018-01-15T19:00:00+00:00", "status": reserved/cancelled/running/finished" }]</pre>	

Table 16. Endpoint /ride

Endpoint		/ride
HTTP Method	POST	
Description	Requests a ride for the user at a specified location	
Software enabler	Booking manager	
Authentication	Bearer token	
Request body	<pre>{ "passenger": { "phone": "+123456789", "email": "a@a.com" }, "pickuplocation": [45.065, 7.694], "pickupAddress": "" }</pre>	
Response	<pre>{ "ride_id": "12345", }</pre>	
HTTP Method	GET	
Description	Retrieves details of a ride	
Software enabler	Booking Manager	
Authentication	Bearer token	
Parameters Name	Type	
rideId	integer	
Response code	200	Successful operation
Response	<pre>{ "ride_id": "12345", "vehicle": { "id": "12345", "provider": "TaxiComp", "type": "Taxi", "pickuplocation": [45.065, 7.694], "pickupAddress": "" }, "arrival_time": 300, "status": reserved/cancelled/running/finished" }</pre>	
HTTP Method	DELETE	
Description	Cancels a running reservation for a ride	
Software enabler	Booking Manager	
Parameters Name	Type	Description
rideId	integer	Identifier of the reservation

Table 17. Endpoint /tickets

Endpoint		/tickets
HTTP Method	POST	
Description	Purchases a specific ticket	
Software enabler	Booking Manager	
Authentication	Bearer token	
Request body		
	<pre>{ "ticket": { "id": "12345" } }</pre>	
Response code	200	Successful operation
Response	<pre>{ "tickets": [{ "id": "12345", "provider": "BusBus", "type": "single", "mode": "pt", "pnr": "xyz123zyx" "area": "center" }] }</pre>	
HTTP Method	GET	
Description	Returns the list of purchasable tickets for a given city or location	
Software enabler	Mobility Organizer	
Authentication	Bearer token	
Parameters Name	Type	Description
position	location	Location to look for available tickets
Response code	200	Successful operation
Response	<pre>{ "tickets": [{ "id": "12345", "provider": "BusBus", "type": "single", "mode": "pt" }, { "id": "12346", "provider": "BusBus", "type": "daily", "mode": "pt" }] }</pre>	

Table 18. Endpoint /tickets/purchased

Endpoint	/tickets/purchased	
HTTP Method	GET	
Description	Returns the list of purchased tickets which are currently valid.	
Software enabler	Booking Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre> { "tickets": [{ "id": "12345", "provider": "BusBus", "type": "single", "mode": "pt", "pnr": "xyz123zyx" "purchaseTime": "2018-01-15T13:10:00+00:00", "expirationTime": "2018-01-15T15:10:00+00:00" }, { "id": "12346", "provider": "BusBus", "type": "single", "mode": "pt", "pnr": "zyx321xyz", "purchaseTime": "2018-01-15T16:45:00+00:00", "expirationTime": "2018-01-15T18:45:00+00:00" }] } </pre>	

Table 19. Endpoint /notification

Endpoint	/notification	
HTTP Method	GET	
Description	Get notification messages for a given user.	
Software enabler	Notification Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre> { "messages": [{ "id": "1234", "content": "Bonus for you", "type": "Incentive ", "options": "NoReply", }, { "id": "1235", "content": "Bus delay", "type": "Warning", "options": "NoReply", },], } </pre>	

Table 20. Endpoint /gamificationData

Endpoint	/gamificationData	
HTTP Method	GET	
Description	Retrieves the user's status in the incentives and gamification module.	
Software enabler	Incentive & Gamification Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre> { "status": { "points": "123", "awardsHistory": [{ "award": { "type": "BonusPt", "points": "50", } "date": "2018-01-15T13:10:00+00:00" }, { "award": { "type": "Voucher", "points": "150", } "date": "2018-03-18T13:10:00+00:00" }] } } </pre>	

Table 21. Endpoint /incentives

Endpoint	/incentives	
HTTP Method	GET	
Description	Retrieves the awards catalogue	
Software enabler	Incentive & Gamification Manager	
Authentication	Bearer token	
Response code	200	Successful operation
Response	<pre>{ "awards": [{ "id": "123", "type": "BonusPt", "points": "50", }, { "id": "124", "type": "Voucher", "points": "150", }] }</pre>	
HTTP Method	POST	
Description	Redeems an award	
Software enabler	Incentive & Gamification Manager	
Authentication	Bearer token	
Request body	<pre>{ "id": "123" }</pre>	
Response code	200	Successful operation
Response	<pre>{ "award": [{ "id": "123", "type": "BonusPt", "points": "50", }], "status": "ok", "message": "You received a public transport ticket!" }</pre>	

CONCLUSIONS

This document showcased the Final Open API specification of the IMOVE system.

This API is intended for mapping and transportation mobile app developers to integrate in their backend implementations innovative MaaS services, leveraging the IMOVE ecosystem capabilities.

Specific MaaS business workflows have been identified, based on MaaS operators IT infrastructures and to be extended and enriched through the adoption of software developed within the IMOVE project. To this aim, a framework of self-consistent and interoperating elements, identified as *enablers*, have been envisioned, developed and made available to the Living Labs.

This specification represents a subset of the overall functionalities targeted in IMOVE, mostly regarding the interactions with backend of mobile applications targeting customers and end users.

According to the architecture drafted in D2.2 ([3]) and updated in D2.5([5]), this specification has first been drafted in D2.3 ([4]) and is implemented in a well-defined software enabler, formerly referred to as B2B API but renamed to Open API for the sake of clarity, leveraging the augmented expertise and domain knowledge acquired through meeting and talks with stakeholders and the feedbacks gathered in the presentations of the architecture, as the term B2B was often confused with the relationship between the MaaS operator and the transportation service providers.

REFERENCES

- [1] IMOVE Description of the Action (DoA), Grant Agreement No. 723314, 04/05/2017, Annex I, Part A.
- [2] Barisone, Alessandro, “D2.1 - Requirement specification and analysis”. *IMOVE project deliverables*
- [3] Certicky, Michal, “D2.2 - Initial system architecture and specification”. *IMOVE project deliverables*
- [4] Barisone, Alessandro, “D2.3 - Initial Open API specification”. *IMOVE project deliverables*
- [5] Barisone, Alessandro, “D2.4 - IMOVE System Implementation 1”. *IMOVE project deliverables*
- [6] Barisone Alessandro, “D2.5 - Final system architecture and specification”. *IMOVE project deliverables*
- [7] Fernandez, José, “D3.1 – Data management plan”. *IMOVE project deliverables*
- [8] Psaltopoulos, Vassilis, “D3.2 - IMOVE Data Translators and Reference Information Model – first version”. *IMOVE project deliverables*
- [9] Apiary: Platform for API Design, Development & Documentation, <https://apiary.io/>
- [10] World Wide Web Consortium (W3C). “SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)”. Accessed 26/06/18. <https://www.w3.org/TR/soap12/>
- [11] Roy Thomas Fielding, *Architectural Styles and the Design of Network-based Software Architectures, Chapter 5, Doctoral dissertation, University of California, Irvine, 2000.*
https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [12] Jones et al., Internet Engineering Task Force (IETF) Request for Comments: 7519, JSON Web Token (JWT) <https://tools.ietf.org/html/rfc7519>
- [13] OpenAPI Specification. <https://swagger.io/specification/>