APIs when acting on someone else’s behalf

Technical analysis, prepared by Citrus Solutions

English translation commissioned by FICORA (prepared by AAC Global)

Citrus Solutions Oy /

Marjukka Niinioja, Olli Nylander, Tessa Viitanen

20 November 2018

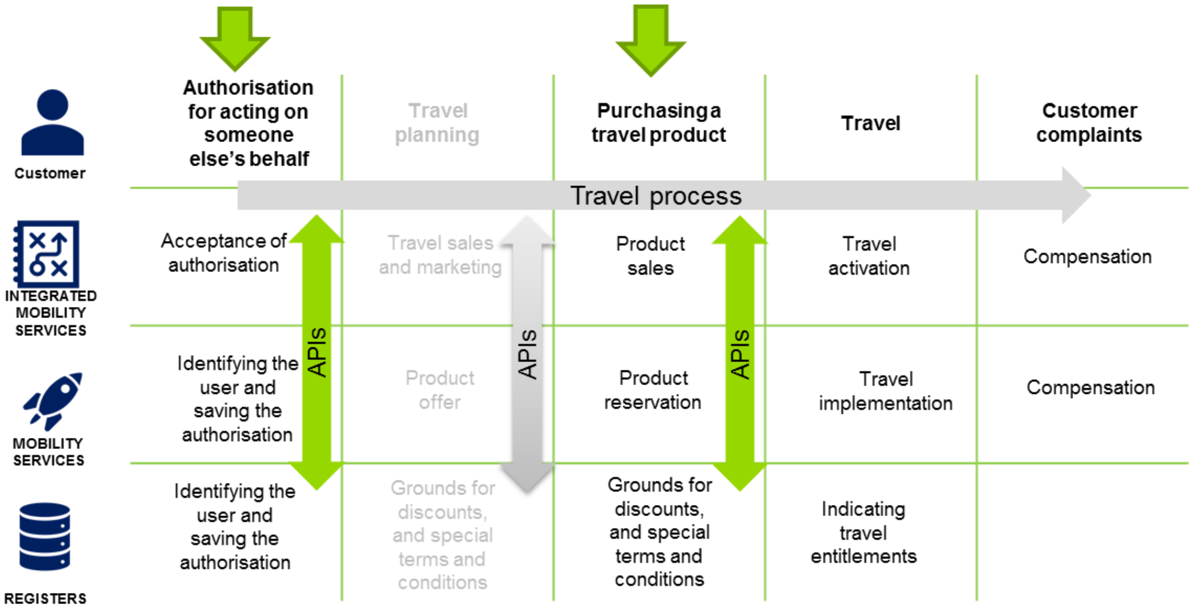
<citrus>

1. Introduction

The Citrus team, together with the project team, has prepared certain possible options for defining application programming interfaces (APIs) and data flows for travel chains when acting on someone else's behalf. As the legislation does not stipulate any detailed specifications for APIs, it is possible, in practice, to reach several different agreements between different parties. This Citrus document describes different technical solution tools.

2. Travel chains – parties and data flows

A general figure of a travel chain has been prepared to form the basis of API specifications. Parties include a customer, an integrated mobility service, mobility services and registers. The travel chain proceeds through the following stages: the customer's authorisation for acting on their behalf, travel planning (not within the scope of the legislation; illustrated in grey), purchasing a journey, travel, customer complaint.



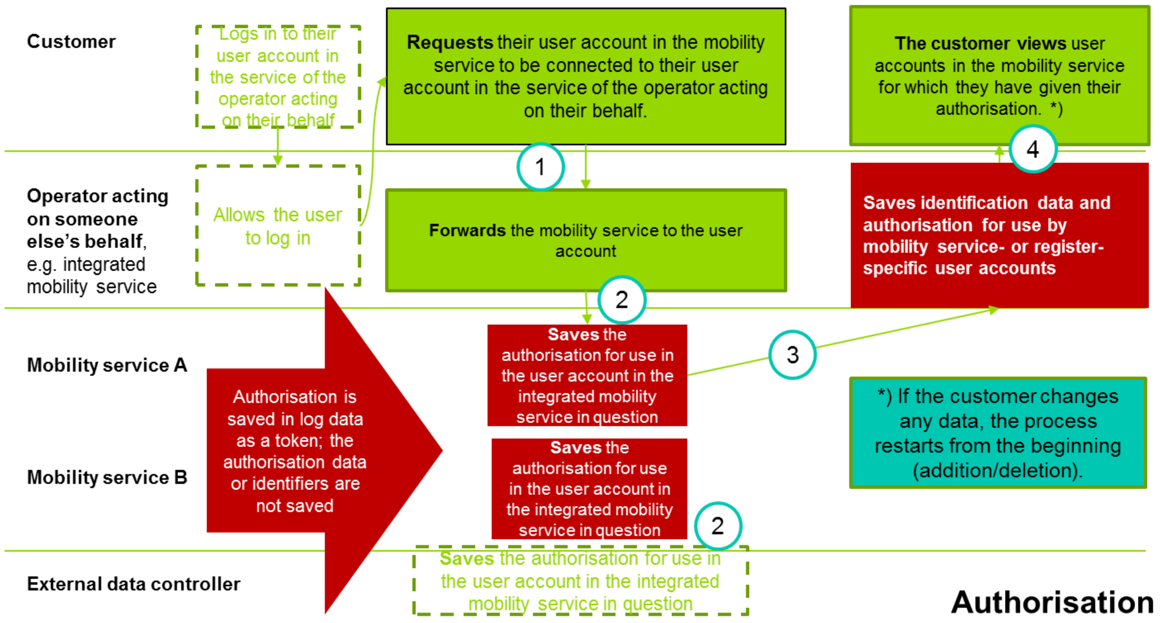
Customer	Authorisation for acting on someone else's behalf	Travel planning	Purchasing a travel product	Travel	Customer complaints
INTEGRATED MOBILITY SERVICES	Acceptance of authorisation	Travel sales and marketing	Product sales	Travel activation	Compensation
MOBILITY SERVICES	Identifying the user and saving the authorisation	Product offer	Product reservation	Travel implementation	Compensation
REGISTERS	Identifying the user and saving the authorisation	Grounds for discounts, and special terms and conditions	Grounds for discounts, and special terms and conditions	Indicating travel entitlements	

As shown in the figure, API specifications are important at the authorisation and purchase stages. When acting on someone else's behalf, parties to the travel chain have different roles, as described below:

Task/role	Customer	Operator acting on someone else's behalf (e.g. integrated mobility service)	Party that opens up their API (e.g. transport service)	External data controller
Authorisation to use user accounts	X			
Verifying authorisations		X	X	X
Data about products related to the travel chain		X	X	
Accessibility			X	
Opening up APIs		X	X	(X)
Producing travel identifiers		X	X	
(Payment)	X			

3. Authorisation

A process description has been prepared for authorisations. Parties to the process include a customer, an operator acting on someone else's behalf (e.g. an integrated mobility service), a mobility service and an external data controller. The areas highlighted in red are based on an alternative model, in which identification data about authorisation is saved for the operator acting on someone else's behalf and the mobility service provider. The final solution depends on the parties and their internal needs.


Customer
Logs in to their user account in the service of the operator acting on their behalf
Requests their user account in the mobility service to be connected to their user account in the service of the operator acting on their behalf.
The customer views user accounts in the mobility service for which they have given their authorisation. *)
Operator acting on someone else's behalf, e.g. integrated mobility service
Allows the user to log in
1
Forwards the mobility service to the user account
4
Saves identification data and authorisation for use by mobility service- or register-specific user accounts
2
Mobility service A
Authorisation is saved in log data as a token; the authorisation data or identifiers are not saved
Saves the authorisation for use in the user account in the integrated mobility service in question
3
*) If the customer changes any data, the process restarts from the beginning (addition/deletion).
Mobility service B
Saves the authorisation for use in the user account in the integrated mobility service in question
2
External data controller
Saves the authorisation for use in the user account in the integrated mobility service in question
Authorisation

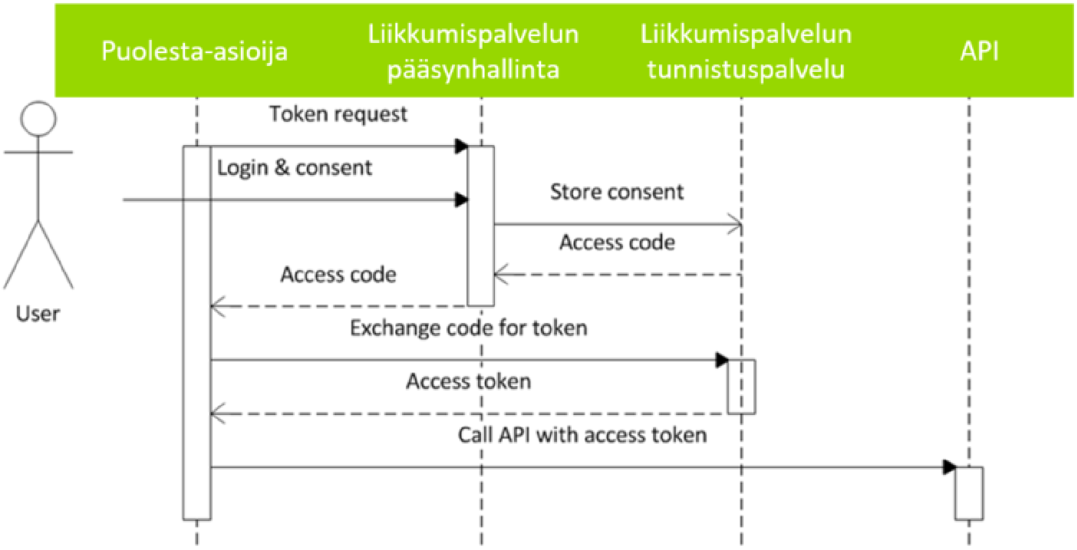
Authorisation APIs are indicated by numbers from one to four in the figure. The data content passing through APIs is presented in the following table:

Authorisation APIs	Data content
1. The customer starts the authorisation process.	• The customer's identifier in the integrated mobility service • Data about the mobility service, i.e. the service to which a connection is made
2. Identifying the user, making new authorisation, changing previous authorisation, checking the validity of authorisation or checking registered data (CALL)	• Identification data about the mobility service (e.g. data based on an agreement number) • The customer's identification data in the authorised service or, with regard to new authorisation, identification data related to the customer (in accordance with special legislation and the service's data protection level)
3. Authorisation for use by the mobility service and user account	• The customer's identification data in the authorised service (if an account or registered data exists), any validity period of the authorisation. *) ***) Of these, an option based on OpenID Connect and the OAuth 2.0 protocol, an option based on the national trust network and an option based on a server-based solution are presented.**
4. The operator acting on someone else's behalf gives notification of the user accounts authorised by the customer and of any other service providers that have not yet been authorised (FEEDBACK)	• Identifiers of authorised mobility service accounts • Available mobility service partners/parties to the travel chain or a register that has not yet been authorised

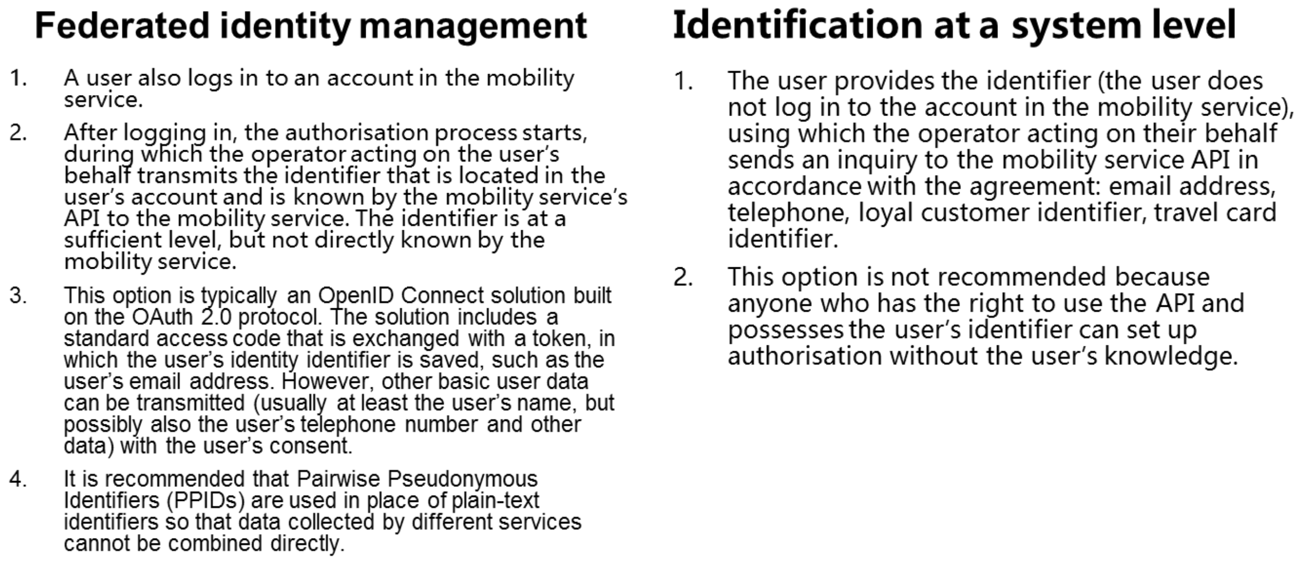
The following principles apply to using and saving authorisations:

- Data is disclosed from a mobility service to the operator acting on someone else's behalf.
- The mobility service must save information that indicates that the user has given their consent to disclosing data.
- This information must be saved for at least as long as data is disclosed, i.e. for as long as the authorisation to use user account data remains valid in this case.
- At minimum, the authorisation is an access management token, the validity period of which usually ranges from tens of seconds to a few hours, and which the service can renew using the refresh token function whenever the user uses the service, as long as the user can be found and the authorisation has not been removed. When the validity of the authorisation ends, data has been saved in log data, where it can be accessed in the case of conflicts or information security incidents within the scope of the retention period for log data.
- The mobility service is in the position of an access management service, and it is responsible for defining how long the identification and authorisation remain valid.

The example presented in the figure represents an authorisation and identification flow based on OpenID Connect. This is one possible option.



A user authorises the operator acting on their behalf to use their user account in the mobility service. We have built two optional models of this type of authorisation. They are presented in the following figure.



Federated identity management

1. A user also logs in to an account in the mobility service.
2. After logging in, the authorisation process starts, during which the operator acting on the user's behalf transmits the identifier that is located in the user's account and is known by the mobility service's API to the mobility service. The identifier is at a sufficient level, but not directly known by the mobility service.
3. This option is typically an OpenID Connect solution built on the OAuth 2.0 protocol. The solution includes a standard access code that is exchanged with a token, in which the user's identity identifier is saved, such as the user's email address. However, other basic user data can be transmitted (usually at least the user's name, but possibly also the user's telephone number and other data) with the user's consent.
4. It is recommended that Pairwise Pseudonymous Identifiers (PPIDs) are used in place of plain-text identifiers so that data collected by different services cannot be combined directly.

Identification at a system level

1. The user provides the identifier (the user does not log in to the account in the mobility service), using which the operator acting on their behalf sends an inquiry to the mobility service API in accordance with the agreement: email address, telephone, loyal customer identifier, travel card identifier.
2. This option is not recommended because anyone who has the right to use the API and possesses the user's identifier can set up authorisation without the user's knowledge.

The OpenID Connect identity layer is based on the OAuth 2.0 (RFC 6749) protocol (https://tools.ietf.org/html/rfc6749). OpenID Connect is a federated identity management (FIM) recommendation built on top of OAuth 2.0 (https://openid.net/specs/openid-connect-core-1_0.html). OpenID Connect is more secure than OAuth 2.0 or SAML alone, and it is more suitable for APIs in JSON format than SAML, which is particularly suitable for SOAP APIs. Most global identification services use OpenID Connect or SAML. The TUPAS identification system of Finnish banks will be replaced by these. The transition period for members of the TUPAS trust network ends in autumn 2019 (http://www.finanssiala.fi/uutismajakka/Sivut/Luottamusverkosto,-TUPAS-ja-tunnistamisen-muutokset.aspx). The Finnish Communications Regulatory Authority (FICORA) has prepared recommendations 212/2018 S (SAML) and 213/2018 S (OIDC) for national trust network APIs. The Finnish identification sector shifted towards the use of OpenID Connect during 2018.

For example, a solution similar to OAuth 2.0, excluding practices recommended by OpenID Connect, have been used during the first stage of the Lippu project. For example, a customised X-Authorization header has been used in the first-stage API https://github.com/finnishtransportagency/lippu-api , whereas the recommendation is to use a standard Authorization header with a Bearer token in API calls. An example of how to search for user data if a mobility service offers the /userinfo API as stated in the recommendation is presented below:

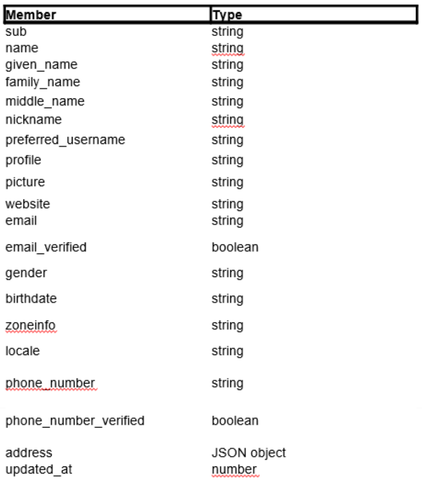
```
GET /userinfo HTTP/1.1

Host: server.example.com

Authorization: Bearer SlAV32hkKG
```

Login endpoints in the first-stage API have different functions and names than the /authorize and /token endpoints stated in the recommendation. The example solution does not include any requirement for acting on someone else's behalf, i.e. a user account or the identification of its user. The API user identified in the first-stage API is not the owner of the user account, but an application that uses the API. Passenger data is transmitted in plain text, not in pseudonymous format. Deviating from recommendations usually results in additional specifications, plans, implementations, tests and information security verifications from API users and may form an extra obstacle and generate additional costs when using the API.

Authorisations can also be given on the basis of consent[1]. Examples of authorisations based on consent as listed in the data protection report prepared by Dittmar & Ingrenius include the standard methods offered by OpenID Connect to interact with the user as listed here. However, API providers should note that consent is only one way of obtaining authorisations. When using a solution based on OpenID Connect, **a mobility service** should be implemented so that a user uses a service that offers to act on their behalf and the operator acting on the user's behalf asks for consent on the user's behalf.

- none: Can be used to check whether authorisation exists; only errors are returned.
- Login: The user is asked to log in again if their previous login is no longer valid.
- Consent: The user is asked to give their consent to using their data (login required).
- Select account: The user is asked to select the user account, for the use of which they want to authorise the operator acting on their behalf (the user may have identifiers and a valid session for several accounts, e.g. business or personal account, their child's account, etc.).

OpenID Connect defines certain standard user data that can be distributed based on consent given by the user. Note: Even if all standard data about the user was available, consent must be asked and only the data that is necessary in order to act on the user's behalf can be distributed. For example, a customer number or personal identity code require additions to the protocol.

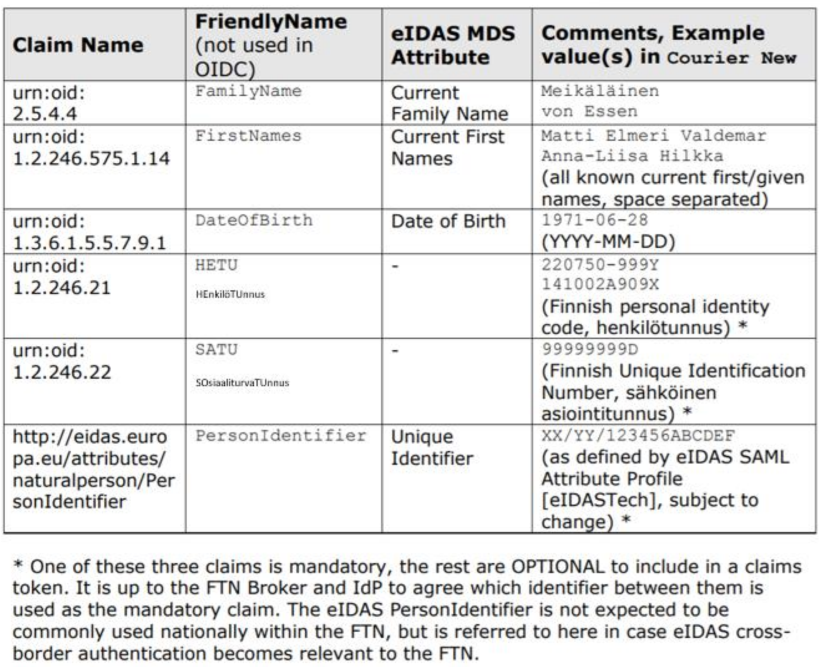


Member	Type
sub	string
name	string
given_name	string
family_name	string
middle_name	string
nickname	string
preferred_username	string
profile	string
picture	string
website	string
email	string
email_verified	boolean
gender	string
birthdate	string
zoneinfo	string
locale	string
phone_number	string
phone_number_verified	boolean
address	JSON object
updated_at	number

In the national trust network, the additional fields presented in the following figure have been defined for OpenID Connect (and SAML). (Finnish Trust Network OpenID Connect 1.0

[1] Here, consent may not mean legal consent to process personal data, but keeping the user generally informed.

Protocol Profile version 1.0. FICORA Recommendation. 213/2018 Shttps://www.viestintavirasto.fi/attachments/suositukset/ftn_oidc_profile_v1.0_ficora_rec_213_2018_s.pdf)[2]



Claim Name	**FriendlyName** (not used in OIDC)	**eIDAS MDS Attribute**	**Comments, Example value(s) in `Courier New`**

urn:oid: 2.5.4.4	`FamilyName`	Current Family Name	`Meikäläinen`
			`von Essen`

urn:oid: 1.2.246.575.1.14	`FirstNames`	Current First Names	`Matti Elmeri Valdemar`
			`Anna-Liisa Hilkka`
			(all known current first/given names, space separated)

urn:oid: 1.3.6.1.5.5.7.9.1	`DateOfBirth`	Date of Birth	`1971-06-28` (YYYY-MM-DD)

urn:oid: 1.2.246.21	`HETU`
	HEnkilöTUnnus
		-	`220750-999Y`
			`141002A909X`
			(Finnish personal identity code, henkilötunnus) *

urn:oid: 1.2.246.22	`SATU` SOsiaaliturvaTUnnus	-	`99999999D` (Finnish Unique Identification Number, sähköinen asiointitunnus) *
http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier	`PersonIdentifier`	Unique Identifier	`XX/YY/123456ABCDEF` (as defined by eIDAS SAML Attribute Profile [eIDASTech], subject to change) *

* One of these three claims is mandatory, the rest are OPTIONAL to include in a claims token. It is up to the FTN Broker and IdP to agree which identifier between them is used as the mandatory claim. The eIDAS PersonIdentifier is not expected to be commonly used nationally within the FTN, but is referred to here in case eIDAS cross-border authentication becomes relevant to the FTN.

4. Purchasing a journey, i.e. a sales interface

The following minimum requirements apply to the functionality of a sales interface:

- A mobility service offers an API at least for receiving reservations/orders (data flows in travel chains – purchasing a journey).
- So that an operator acting on someone else's behalf can place an order, it must have access to sufficient route, stop, timetable, price and digital accessibility data, as well as physical accessibility data, via APIs (in practice, the essential data listed in chapter 2, section 1, subsection 1 of the Act on Transport Services is necessary for normal-priced travel entitlements and those carrying a discount, compensation or special condition when acting on someone else's behalf as stipulated in section 2a; see data flows in travel chains – travel planning).
- A condition for acting on someone else's behalf is that "the identification and user information existing in the service user's user account" be used (section 2a, subsection 1). In the API, data in the user account used to purchase a journey and the personal data of any other passengers should be transmitted in pseudonymous

[2] FICORA's specification concerns APIs with a pre-defined data content between strong identification providers. The right to process personal data must be assessed when acting on someone else's behalf.

format and only to the extent as is necessary to register and check the travel entitlement.

The differences in using APIs at the first stage of the Lippu project and its second stage (acting on someone else's behalf) are as follows:

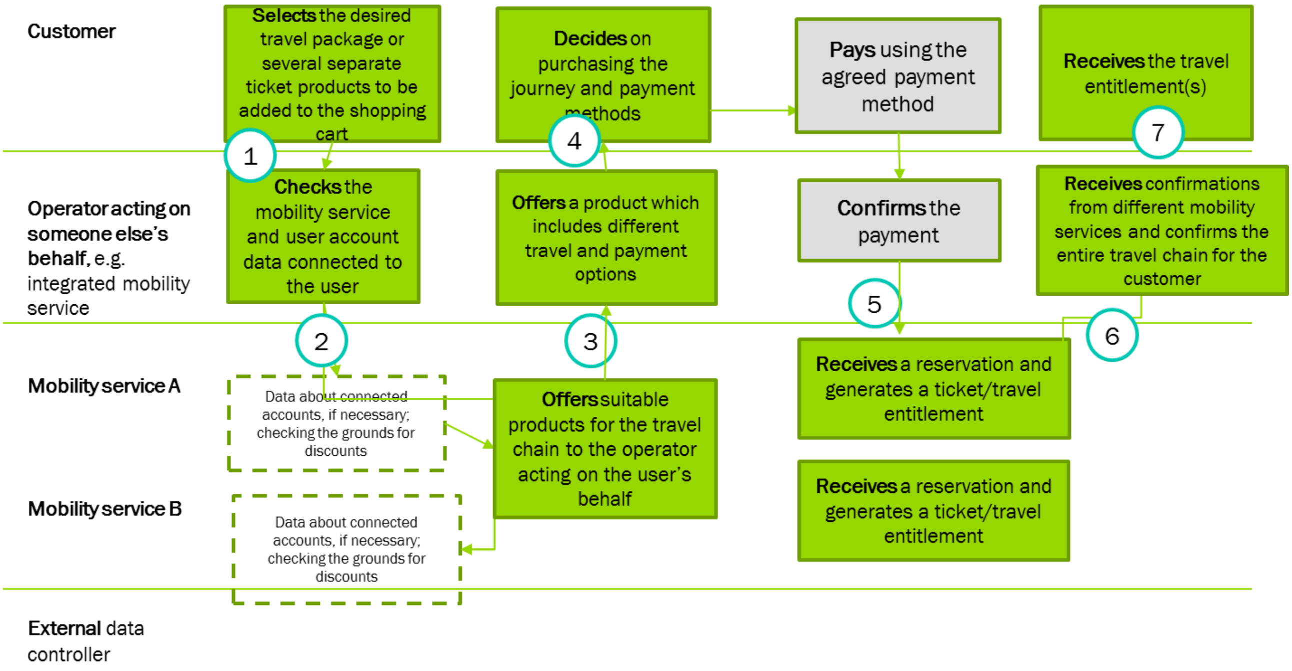
At the first stage

- Search for products (Products)
- Check product availability (Availability)
- Make the reservation (Reservation)
- Process the travel entitlement (Travel entitlement)

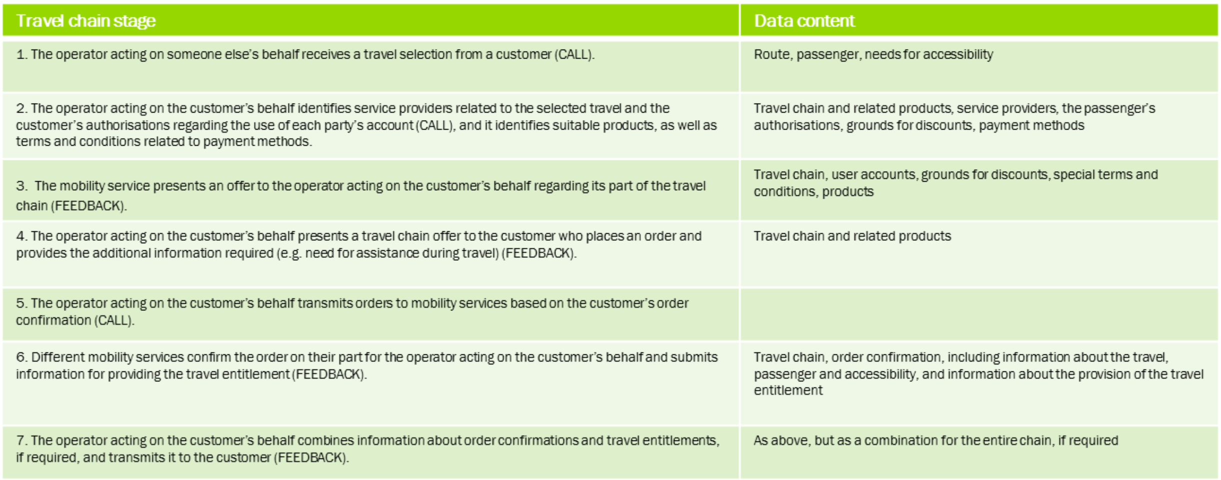
At the second stage

- Authorise the operator acting on someone else's behalf (Consent/Delegation)
- Search for products suitable for the user (Products)
- Check product availability (Availability)
- Make the reservation (Reservation)
- Process the travel entitlement (Travel entitlement)

Purchasing a journey can be described similarly to authorisations. Parties include a customer, an operator acting on someone else's behalf, a mobility service and an external data controller. In the figure, the payment process is highlighted in grey because it is not stipulated in the Act on Transport Services as a function.


Customer
Selects the desired travel package or several separate ticket products to be added to the shopping cart
Decides on purchasing the journey and payment methods
Pays using the agreed payment method
Receives the travel entitlement(s)
7
1
4
Operator acting on someone else's behalf, e.g. integrated mobility service
Checks the mobility service and user account data connected to the user
Offers a product which includes different travel and payment options
Confirms the payment
Receives confirmations from different mobility services and confirms the entire travel chain for the customer
5
6
2
3
Mobility service A
Data about connected accounts, if necessary; checking the grounds for discounts
Offers suitable products for the travel chain to the operator acting on the user's behalf
Receives a reservation and generates a ticket/travel entitlement
Mobility service B
Data about connected accounts, if necessary; checking the grounds for discounts
Receives a reservation and generates a ticket/travel entitlement
External data controller

APIs have been numbered from one to seven. The table below describes APIs and their data content.



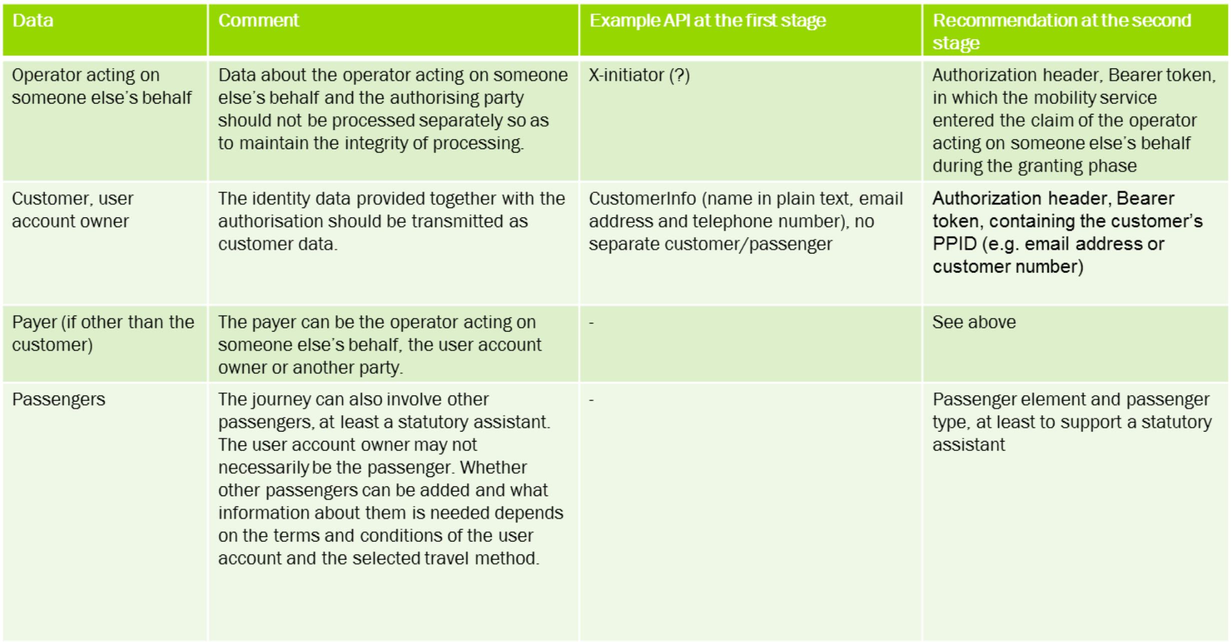
Travel chain stage	Data content
1. The operator acting on someone else's behalf receives a travel selection from a customer (CALL).	Route, passenger, needs for accessibility
2. The operator acting on the customer's behalf identifies service providers related to the selected travel and the customer's authorisations regarding the use of each party's account (CALL), and it identifies suitable products, as well as terms and conditions related to payment methods.	Travel chain and related products, service providers, the passenger's authorisations, grounds for discounts, payment methods
3. The mobility service presents an offer to the operator acting on the customer's behalf regarding its part of the travel chain (FEEDBACK).	Travel chain, user accounts, grounds for discounts, special terms and conditions, products
4. The operator acting on the customer's behalf presents a travel chain offer to the customer who places an order and provides the additional information required (e.g. need for assistance during travel) (FEEDBACK).	Travel chain and related products
5. The operator acting on the customer's behalf transmits orders to mobility services based on the customer's order confirmation (CALL).	
6. Different mobility services confirm the order on their part for the operator acting on the customer's behalf and submits information for providing the travel entitlement (FEEDBACK).	Travel chain, order confirmation, including information about the travel, passenger and accessibility, and information about the provision of the travel entitlement
7. The operator acting on the customer's behalf combines information about order confirmations and travel entitlements, if required, and transmits it to the customer (FEEDBACK).	As above, but as a combination for the entire chain, if required

The data architecture of a sales interface and a comparison with the first stage have been presented in the following table. Parties to the sales interface include an operator acting on someone else's behalf, a customer, a payer and passengers. The columns are as follows:

In columns 1–2, the data architecture has been presented at a logical level.

Column 3 includes a reference to the example solution built at the first stage and describes how the data field in question has been implemented in its API.

Column 4 presents a recommendation how data should be presented so as to meet the requirements set for acting on someone else's behalf and to be as standard as possible.



Data	Comment	Example API at the first stage	Recommendation at the second stage
Operator acting on someone else's behalf	Data about the operator acting on someone else's behalf and the authorising party should not be processed separately so as to maintain the integrity of processing.	X-initiator (?)	Authorization header, Bearer token, in which the mobility service entered the claim of the operator acting on someone else's behalf during the granting phase
Customer, user account owner	The identity data provided together with the authorisation should be transmitted as customer data.	CustomerInfo (name in plain text, email address and telephone number), no separate customer/passenger	Authorization header, Bearer token, containing the customer's PPID (e.g. email address or customer number)
Payer (if other than the customer)	The payer can be the operator acting on someone else's behalf, the user account owner or another party.	-	See above
Passengers	The journey can also involve other passengers, at least a statutory assistant. The user account owner may not necessarily be the passenger. Whether other passengers can be added and what information about them is needed depends on the terms and conditions of the user account and the selected travel method.	-	Passenger element and passenger type, at least to support a statutory assistant

The following table defines the data content of the sales interface regarding travel entitlements.

Data	Comment	Example API at the first stage	Recommendation at the second stage
Start and end dates of a travel entitlement	Date, time and time zone	"validFrom": "2018-10-20T07:15:31.495Z", "validTo": "2018-10-20T07:15:31.495Z"	Identical to the first stage
Points of departure and destination of a travel entitlement	Exact location or region, depending on the travel and product. The exact location may not be known when making the purchase (e.g. zone-specific, serial or monthly tickets)	Coordinates or a location used in the availability call; no location data when purchasing a journey or in the travel authorisation	Distance, FromLocation, ToLocation, Location All locations can be identified on the basis of the location identifier (e.g. a specific station), postal address, coordinates or text.
Physical accessibility and need for assistance	According to the terms and conditions of the transport service, the passenger can book a seat for an assistant (see passengers), or provide information about a wheelchair (this can also be a separate ticket type) or any need for an assistant at stations.	"accessibility": [{ "title": "WHEELCHAIR"...	A sufficient section for indicating any need for assistance, depending on the transport service and its special terms and conditions. Data protection and individual restrictions must be taken into account. Details should be specified verbally.

The following table defines data about products and prices in the sales interface.

Data	Comment	Example API at the first stage	Recommendation at the second stage
Products	Identifiers of mobility service products related to the travel reservation	"productType": "PRODUCT-SINGLE-TICKET" "extraServices" {}	Products are added for discount and compensation groups, and to products that include special terms and conditions
Price	Price per product, total price, inclusion of tax, discounts	"fare": { "currency": "string", "amount": 0, "vatPercent": 0}	Identical to the first stage; schema.org uses "price" and "priceSpecification"
Quantity	Passenger- or travel-specific quantity or other quantity per product row	See price (amount)	Quantity, not mandatory if based on consumption. MinPrice, MaxPrice or, in serial tickets, the price and unit of a single component (unitCode)
Currency		See price (currency, ISO standard)	Identical to the first stage
Payment method	The customer's payment method may have an impact on the price or terms and conditions, such as customer complaints.		PaymentType, if required

The following table describes technical identifiers in the sales interface.

Data	Comment	Example API at the first stage	Recommendation at the second stage
Language		Accept-Language	Identical
Message identifier	A unique ID generated by the caller for each call; used to investigate any errors	X-Message-Id	Identical; however, the message identifier should include a timestamp (e.g. RFC2822) to verify that the identifier is unique and to save the timestamp.
Reservation identifier	A unique ID generated by the party calling the API. The ID remains unchanged throughout all transactions related to the purchase of a single journey.	X-Transaction-Id	Identical; however, the transaction identifier should include a timestamp to verify that the identifier is unique and to save the timestamp. The timestamp indicates when the purchase of the travel chain started.

5. Physical implementation of APIs

The legislation does not directly stipulate how an API should be implemented. The implementation method also affects what standards apply to the identification process and authorisations. However, requirements for having real-time and standard data can usually be fulfilled by using modern solutions, such as APIs based on REST architectural principles. GraphQL or other types of APIs (gRPC, etc.) have not been considered to be sufficiently manageable for open APIs intended for public use or shared use by partners. However, these may also be viable options as technologies and practices develop. Their standard data format is JSON and JSON schema where different sets of schemas can be used (schema.org usually for commercial purposes). In addition, XML and XML schema can be used in REST and SOAP solutions.