Proposal for a New "User Preferences" Feature in OJP 2.0 TripRequest

> This page contains the "building blocks" for our planned proposal for the extension of OJP 2.0 TripRequest with "User Preferences".
>
> Status: Provisional

1. Summary

This is a proposal to add new filter criteria in OJP 2.0 for expressing user preferences in a TripRequest. User preferences are high-level criteria of a passenger's requirements and properties such as price-sensitivity, speediness, convenience, sportiness or environmental friendliness.

Trip planners have been evolving over the years to become more and more comprehensive, detail-rich, fully intermodal, including all public and private operators, and for end-to-end planning. We believe that a good concept of user preferences will become crucial for this evolution. User preferences are easy-to-use and easy-to-understand, yet precise and effective controls to influence trip planning. User preferences will allow the user to actually get those trip planning results which really meet his or her needs.

OJP 1.0 has various low-level, fine-granular controls, but no comprehensive high-level controls for user preferences. This proposal suggests to add these in OJP 2.0.

2. The Problem: The Explosion of Combinations in Intermodal Trip Planning

Mobility has undergone deep, disruptive changes over the last years. Major trends such as urbanization, digitalisation, "mobile first", sharing economy and decarbonisation have triggered hype trends such as MaaS (Mobility as a Service) or ABT (Account Based Ticketing) and new mobility modes such as micro-mobility, ride-hailing, carpooling or on-demand transport.

Trip planners in web apps and smartphone apps have become a crucial element of this transformation. They have evolved from restricted, single-mode route planners or public transport planners into more and more comprehensive, detail-rich, intermodal, end-to-end trip planners.

However, intermodality and comprehensiveness add a new challenge: trips may now be composed of a vast choice of modes and operators, not only on the first and last mile, but also on the main legs of the trip. A truly open, unbiased, intermodal trip planner algorithm could compute thousands or millions of combinations for a trip from A to B, instead of just a few combinations that are typical for a simple, old-fashioned PT planner.

Thus, trip planners are faced with the problem of overwhelming amount of choices:

- Which of the many combinations should be presented to the user?
- Based on what justification should some results be displayed and many other results not?

3. The Solution: User Preferences

3.1.1. Definition

User preferences are a range of criteria or requirements which express what a user prefers when faced with a large choice of options. Each preference gives an indication about how suitable a certain option is for the given user.

Each user preference can be expressed as a criterion (e.g. "sportiness") or as a question (e.g. "how important is environmental-friendly travel for you?"). The actual preference may be expressed on a scale, typically in a few well defined stages (e.g. "very imporant", "average", "not so important", or "yes"/" no").

User preferences may be influenced somewhat by marketing and trends, but at the core, they are very much a personal, individual, private matter. Only the user himself or herself can actually determine what his or her preferences actually are.

3.1.2. Usage

Various preferences may be combined through Boolean AND-operations, e.g. "sportiness is average AND environmental friendly is very important".

A given statement of user preferences allows to narrow down the choice of suitable options. With a good set of user preferences, the choice may be boiled down to just a few options.

Furthermore, the options can be scored and compared based on the preferences, and only the options with the highest scores need to be displayed to the user.

3.1.3. Good User Preferences

Good preferences must be relevant, easy to understand, clear and unambiguous.

The actual preference should be expressed either as a binary yes/no, or quantified on a short scale in 3, 4 or 5 steps with well-defined meanings, so that they can be easily understood and applied by most users. A greater number of steps is both harder to respond to by users and more difficult to implement in an IT system.

4. User Preferences for Trip Options

4.1.1. Mobility Preferences in General

Personal preferences obviously play a crucial role in mobility. In modern, urban areas, people can choose from many mobility options, various modes of transport and many operators. Their personal preferences about convenience, speed, cost and other aspects will usually determine their decision for certain options. Given the same mobility need (moving from A to B), various people will decide for a variety of totally different options, using individual modes, PT, private operators and intermodal combinations.

It thus becomes compelling that trip planners must be controlled first and foremost by user preferences.

4.1.2. User Preferences in Current Trip Planners

A brief survey shows that user preferences are actually a trend in mobility apps. In these apps, users can set their preferences by adjusting some sliders or checkboxes. The trip planning algorithm is then controlled by these settings in order to get well-suited trip-planning suggestions.

4.1.3. How to Find Suitable User Preferences?

Actual, specific criteria for user preferences may be deduced in various ways, from:

- what others do ("best practices"),
- empirical observation ("what is important for myself and for others?")
- practical considerations ("what is easy to understand and unambiguous?").
- what is actually helpful for narrowing down the number of combinations?
- what is actually technically feasible?

5. Proposed User Preferences for Trip Requests

From such observations and considerations, we have deduced the following set user preferences, roughly in order of importance for a majority of users.

Generally, we believe that the given two or three levels or stages are sufficient and appropriate. More levels would be hard to discern and difficult to implement in trip planners.

user preference	**levels / stages (<u>default</u>)**	**description, notes**

low price	very important, important, unimportant	For many people (the "average"), the price for mobliltiy will be a factor of great interest, as with many other things in life. For some people with low budgets, it will be very "important". For others, the price will be "unimportant". Three levels apear adequate, more would be hard to discern/implement.
speed	important, average, unimportant	In most cases, users (travellers) want the trip duration to be short, since the time spent for mobility is seen as unproductive and spoilt time. Again, for some, this may be be more important, for others, less.
comfort	premium, average, basic	Comfort or convenience is another decisive factor, although not so easily measurable. In some mobility domains, this has manifested itself in two or three "classes".
sportiness	high, average, low	Users may be more or less motivated and capable of sportive, self-powered modes of mobility such as walking, bike, scooters, etc., depending on various personal factors such as age, health, exercise or current context (business trip or leasure activity). Sportiness usually correlates with the speed (of walking, biking, etc.).
environmental safety	high, average, low	Environmental concerns are becoming more and more important, however clearly more for some people than for others.
inclusion /exclusion of certain modes or operators	(yes/no)	The user may wish to directly exclude or include certains modes or certain operators. This is an alternative to the other preferences, which will also have an impact on the choice of modes and operators. E.g. "sportiness=low" should have the same effect as "exlude bikesharing".
weather protection	yes/no	Some modes of transport expose the passengers to rain, wind and cold or hot temperatures. On certain trips, passengers will want to avoid this.
extra safe	yes/no	Some modes such as "taxi" are perceived as more secure or safe, less risk-prone, others like "scooter" more. Some users may want to exclude unsafe modes.
extra reliable	yes/no	Some modes may have a record of being more reliable, less prone to delays or cancellations.
extra accessible	yes/no	The requirement of good accessibility imposes various other requirements on trip planning, such as shorter transfers (walks), extended transfer times or adequate infrastructure.
scenic	yes/no	Scenicness of a trip will be a typical requirement for tourists. Trip planners should give preference to routes or trip legs known to be scenic.
TripReason	Enumeration	TRIP REASONs for trip.

6. OJP 1.0 Coverage of User Preferences

Transmodel Part 6 on User Preferences

Transmodel (Part 6) has introduced the concept of user preferences with these words: "...

The methods of selecting a trip proposal in response to a TRIP REQUEST can use a variety of selection and optimization criteria, which can be expressed using a TRIP CONTENT FILTER and a TRIP REQUEST POLICY on the query. For instance:
*— **preference** for a particular OPERATOR;*
*— **preference** for one transport mode (bus, metro…);*
— whether to select a journey by arrival or departure time;
— minimum duration or distance of the whole trip;
— minimum walking distance;
— minimum number of interchanges;
— minimum amount of fare;

*[trip planner]... takes into account the user's constraints or **preferences,** such as minimal trip duration, minimal number of interchanges, cheapest fare, etc., and involves an optimization process using such parameters.*

OJP 1.0 adheres to these statements of the Transmodel Part 6 specification. OJP 1.0 TripRequest has the following parameters that (partly) reflect user preferences (*Statements from the OJP 1.0 specification in italic*)

OJP 1.0 Control	Definition in Specification (original wording from specification in *italic)*	Level	Discussion
Individual Transport Options	*Options how to access/leave the place by individual transport*. Mode must be defined. MaxDistance, MaxDuration, MinDistance, MinDuration or Speed (*Relative speed in percent*) and AdditionalTime may be defined.	low	A control of the first-/last-mile trip leg, restricted to one mode, though. User preferences about the mode and the user's sportiness can be set, although with low-level, fine grained parameters.
TransferLimit	*The maximum number of interchanges the user will accept per trip*.	low	Again, a fine-grained parameter regarding convenience. In practice, this may be difficult to use: If set too high, it has no influence; if too low, no trip options are found.

OptimisationMethod	*The type of the target function that should be applied when searching for optimal trip results. ... fastest* | *minChanges* | *leastWalking* | *leastCost* | *earliestArrival* | *latestDeparture* | *earliestArrivalAndLatestDeparture*.	medium	This parameter comes closest to the idea of superior user preferences. However, it is restricted to a "one of seven" choice, it allows no combinations and no weighing of these aspects.
ItModesToCover	Allows to request extra, monomodal trip options.	none	The parameter has actually no influence on the intermodal planning!
PtModeFilter	*Modes to be considered in trip calculation... List of public transport modes to include or exclude*.	medium	This parameter will sometimes be useful to control PT modes to include or exclude, allowing to narrow-down to PT transport modes that the user finds acceptable.
LineFilter, OperatorFilter	Allows to include/exclude certain PT lines or operators.	low	This might be useful sometimes to exclude unwanted lines or operators.
PrivateModeFilter	*Private mobility options to include/exclude... List of private mobility offers to include or exclude.*	medium	Discussion: A useful parameter to control private modes. No control, however, on the private operators.

In summary, OJP offers some restricted control with IndividualTransportOptions for the first and last mile, provides a general OptimisationMethod selector (1 of 7), and allows to include or exclude certain PT modes, lines, operators and private modes.

Regarding our proposed user preferences above, OJP 1.0

- is comprehensive on include/exclude options,
- but has almost no coverage for the various other preferences about the quality of a trip.

7. User Preferences in EFA JSON API - EFA Trip Planner of Mentz

EFA is a commercial trip planner system by the German company Mentz. (Solutions for public and individual transport - MENTZ). EFA provides a proprietary query API with some user preferences. This includes the following options:

- Connection Options ... makes it possible, for example, to calculate trips for people with reduced mobility, heavy baggage etc.
- Common Options: calcNumberOfTrips Controls the number of responses.
- **ptOptionsActive** (=1): additional options for PT; including: **useProxFootSearch=1** "Taking account of nearby stops", **maxChanges:** Maximum number of changes in one trip. ... default=9; **routeType:** leastinterchange / leasttime / leastwalking; exclMOT_1, ..., exclMOT_19, or **excludedMeans=1,** etc.: exclusion of some modes of transport (0=Train, 2=commuter railway, ...19=citizen bus); inclMOT_1.... or **"includeMeans"** same as with "exclude".
- **itOptionsActive** (=1) Options valid for individual Transport: **trITMOT:** select the "Mode of Transport" (100 footpath, 101 bike & ride, 102 take your bike along, 103 kiss & ride, 104 park & ride), may be set separately for arrival/departure; apparently limited to one. ⚠; **trITMOTvalue<ID>** ... The value of the parameter indicates the maximum time... default 10 minutes.
- Options valid for public transport and individual transport: **changeSpeed:** Sets the speed for interchange paths, when ptOptionsActive=1; has 3 stages: slow (50%), normal (100 %), fast (200 %); influences transfer times and walking speed.

In summary, somewhat comparable to OJP 1.0, EFA has various low-level user preferences and exclusion/inclusion settings, but no high level controls.

8. Proposal for the Addition of User Preferences in OJP 2.0

Given the foregoing analysis, our proposal aims at complementing the OJP standard with the missing feature of high level user preferences about the quality of the trip.

In order to be backwards compatible and nonbreaking, we propose to add an optional new element **`UserPreferences`** in the **`Params`** element of the TripRequest. We prefere an element over an XSD group since the element provides an explicit, visible grouping of the new elements. The **`UserPreferences`** element contains various optional elements, each of which defines one specific user preference.

For each preference, a range of possible values is defined. One of the values, usually an intermediate or most common value, will be the default value. Choosing the default value should have the same effect as leaving the element away.

The general idea is that the trip planning algorithm shows a normal, average behavior if no user preferences or the default values are provided. If a non-default value is provided, however, the algorithm should adapt its search procedures and selection of results to correspond best possible to this preference.

8.1.1. Additions to the XSDs (File OJP_Trips.xsd)

1. (Line 59) Add a new Element **`UserPreferences`** to the **`TripParamStructure`** (thus in the TripRequest **`Params`** Element)

2. (Line 882) Add a new Type **`UserPreferencesStructure`**

File for download: OJP_Trips.xsd

Source Code:

Proposed addtions to OJP_Trips.xsd (Line 59 and 882)

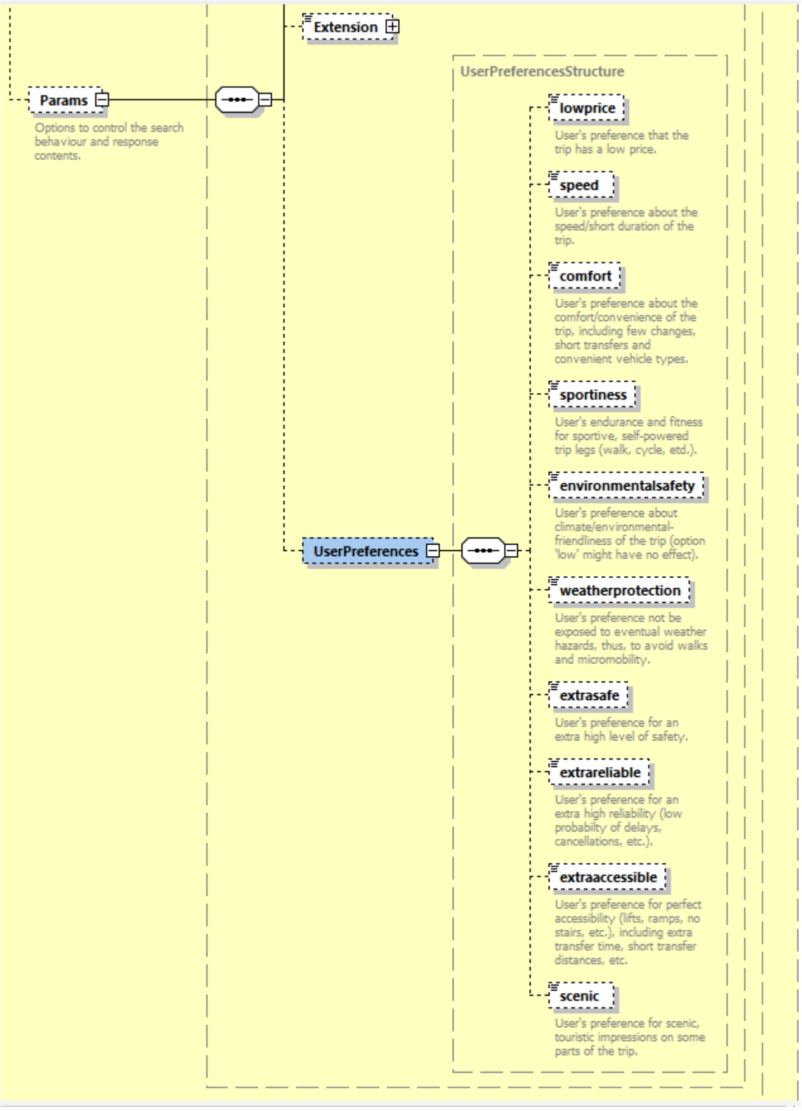
```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema xmlns="http://www.vdv.de/ojp" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:siri="http://www.
siri.org.uk/siri" targetNamespace="http://www.vdv.de/ojp" elementFormDefault="qualified" attributeFormDefault="
unqualified">
        <xs:annotation>
                <xs:documentation>OJP/OJP_Trips.xsd - Request and response definitions for trip requests and
distributed journey planning</xs:documentation>
        </xs:annotation>
        <xs:import namespace="http://www.siri.org.uk/siri" schemaLocation="../siri/siri_model/siri_all-v2.0.xsd"
/>
        <xs:include schemaLocation="OJP_All.xsd"/>
        <xs:annotation>
                <xs:documentation>========================================== Request definitions
==========================================</xs:documentation>
        </xs:annotation>
        <xs:group name="TripRequestGroup">
                <xs:annotation>
                        <xs:documentation>Trip request structure.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="Origin" type="PlaceContextStructure" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Specifies the origin situation from where the user
wants to start.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Destination" type="PlaceContextStructure" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Specifies the destination situation where the user is
heading to.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Via" type="TripViaStructure" minOccurs="0" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Ordered series of points where the journey must pass
through. If more than one via point is given all of them must be obeyed - in the correct order. The server is
allowed to replace a via stop by equivalent stops.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="NotVia" type="NotViaStructure" minOccurs="0" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Not-via restrictions for a TRIP, i.e. SCHEDULED STOP
POINTs or STOP PLACEs that the TRIP is not allowed to pass through. If more than one not via point is given all
of them must be obeyed.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="NoChangeAt" type="NoChangeAtStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>no-change-at restrictions for a TRIP, i.e. SCHEDULED
STOP POINTs or STOP PLACEs at which no TRANSFER is allowed within a TRIP.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Params" type="TripParamStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Options to control the search behaviour and response
contents.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:complexType name="TripParamStructure">
                <xs:annotation>
                        <xs:documentation>Trip request parameter structure.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="TripDataFilterGroup"/>
                        <xs:group ref="TripMobilityFilterGroup"/>
                        <xs:group ref="TripPolicyGroup"/>
                        <xs:group ref="TripContentFilterGroup"/>
                        <xs:element name="FareParam" type="FareParamStructure" minOccurs="0"/>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                        <xs:element name="UserPreferences" type="UserPreferencesStructure" minOccurs="0"
maxOccurs="1" />
                </xs:sequence>
        </xs:complexType>
        <xs:group name="TripDataFilterGroup">
                <xs:annotation>
                        <xs:documentation>Data to be included/excluded from search, f.e. modes, operators.</xs:
documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="PtModeFilter" type="PtModeFilterStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Modes to be considered in trip calculation.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LineFilter" type="LineDirectionFilterStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Lines/Directions to include/exclude.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="OperatorFilter" type="OperatorFilterStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Transport operators to include/exclude.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="PrivateModeFilter" type="PrivateModeFilterStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Private mobility options to include/exclude.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:group name="TripPolicyGroup">
                <xs:annotation>
                        <xs:documentation>Policies that control the trip search behaviour.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="BaseTripPolicyGroup"/>
                        <xs:element name="TransferLimit" type="xs:nonNegativeInteger" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>The maximum number of interchanges the user will
accept per trip.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="OptimisationMethod" type="OptimisationMethodEnumeration" minOccurs="0"
>
                                <xs:annotation>
                                        <xs:documentation>the types of algorithm that can be used for planning
a journey (fastest, least walking, etc)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ItModesToCover" type="IndividualModesEnumeration" minOccurs="0"
maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>For each mode in this list a separate monomodal trip
shall be found - in addition to inter-modal solutions.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:group name="TripMobilityFilterGroup">
                <xs:annotation>
                        <xs:documentation>Parameters the user can set to restrict the mobility options -
particularly for interchanging.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="BaseTripMobilityFilterGroup"/>
                        <xs:element name="LevelEntrance" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>The user needs vehicles with level entrance between
platform and vehicle, f.e. for wheelchair access.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="BikeTransport" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>The user wants to carry a bike on public transport.<
/xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="WalkSpeed" type="OpenPercentType" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Deviation from average walk speed in percent. 100%
percent is average speed. Less than 100 % slower, Greater than 150% faster.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="AdditionalTransferTime" type="xs:duration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Additional time added to all transfers (also to
transfers between individual to public transport).</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:group name="TripContentFilterGroup">
                <xs:annotation>
                        <xs:documentation>Parameters that control the level of detail of the trip results.</xs:
documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="BaseTripContentFilterGroup"/>
                        <xs:element name="IncludeIntermediateStops" type="xs:boolean" default="false"
minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether the result should include intermediate stops
(between the passenger's board and alight stops).</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="IncludeFare" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether the result should include fare information.<
/xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="IncludeOperatingDays" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether the result should include operating day
information - as encoded bit string and in natural language.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="TripSummaryOnly" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>If true, then the response will contain only
summaries of the found trips. Default is false.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:simpleType name="OptimisationMethodEnumeration">
                <xs:annotation>
                        <xs:documentation>the types of algorithm that can be used for planning a journey
(fastest, least walking, etc).</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                        <xs:enumeration value="fastest"/>
                        <xs:enumeration value="minChanges"/>
                        <xs:enumeration value="leastWalking"/>
                        <xs:enumeration value="leastCost"/>
                        <xs:enumeration value="earliestArrival"/>
                        <xs:enumeration value="latestDeparture"/>
                        <xs:enumeration value="earliestArrivalAndLatestDeparture"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="NotViaStructure">
                <xs:annotation>
                        <xs:documentation>NNot-via restrictions for a TRIP, i.e. SCHEDULED STOP POINTs or STOP
PLACEs that the TRIP is not allowed to pass through</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:choice>
                                <xs:element ref="siri:StopPointRef"/>
                                <xs:element ref="StopPlaceRef"/>
                        </xs:choice>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="NoChangeAtStructure">
                <xs:annotation>
                        <xs:documentation>no-change-at restrictions for a TRIP, i.e. SCHEDULED STOP POINTs or
STOP PLACEs at which no TRANSFER is allowed within a TRIP.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:choice>
                                <xs:element ref="siri:StopPointRef"/>
                                <xs:element ref="StopPlaceRef"/>
                        </xs:choice>
                </xs:sequence>
        </xs:complexType>
        <xs:annotation>
                <xs:documentation>========================================== Response definitions
==========================================</xs:documentation>
        </xs:annotation>
        <xs:group name="TripResponseGroup">
                <xs:annotation>
                        <xs:documentation>Trip response structure.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="TripResponseContext" type="ResponseContextStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Context to hold trip response objects that occur
frequently.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="TripResult" type="TripResultStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>The trip results found by the server.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:complexType name="TripResultStructure">
                <xs:annotation>
                        <xs:documentation>Structure for a single trip result and its accompanying error
messages.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="ResultId" type="xs:NMTOKEN">
                                <xs:annotation>
                                        <xs:documentation>Id of this trip result for referencing purposes.
Unique within trip response.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ErrorMessage" type="ErrorMessageStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>Error messages related to this trip result.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:choice>
                                <xs:element name="Trip" type="TripStructure">
                                        <xs:annotation>
                                                <xs:documentation>Detailed information on trip.</xs:
documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="TripSummary" type="TripSummaryStructure">
                                        <xs:annotation>
                                                <xs:documentation>Summary on trip. Only if requestor set
TripSummaryOnly in request.</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                        </xs:choice>
                        <xs:element name="TripFare" type="TripFareResultStructure" minOccurs="0" maxOccurs="
unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="TripSummaryStructure">
                <xs:annotation>
                        <xs:documentation>Structure for trip overview information.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="TripId" type="xs:NMTOKEN">
                                <xs:annotation>
                                        <xs:documentation>Id of this trip for referencing purposes. Unique
within trip response.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Origin" type="PlaceRefStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Describes the origin situation of this trip.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Destination" type="PlaceRefStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Describes the arrival situation of this trip.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Duration" type="xs:duration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Overall duration of the trip.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="StartTime" type="xs:dateTime" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Departure time at origin.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="EndTime" type="xs:dateTime" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Arrival time at destination.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="PTTripLegs" type="xs:nonNegativeInteger" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Number of public transport trip legs.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Distance" type="siri:DistanceType" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Trip distance.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="OperatingDaysGroup" minOccurs="0"/>
                        <xs:element ref="SituationFullRef" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="TripStructure">
                <xs:annotation>
                        <xs:documentation>[an extended form of PT TRIP in TM and NeTEx as it also includes the
initial and final access legs to and from public transport] whole journey from passenger origin to passenger
destination in one or more trip LEGs </xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="TripId" type="xs:NMTOKEN">
                                <xs:annotation>
                                        <xs:documentation>Id of this trip for referencing purposes. Unique
within trip response.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Duration" type="xs:duration">
                                <xs:annotation>
                                        <xs:documentation>Overall duration of the trip.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="StartTime" type="xs:dateTime">
                                <xs:annotation>
                                        <xs:documentation>Departure time at origin.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="EndTime" type="xs:dateTime">
                                <xs:annotation>
                                        <xs:documentation>Arrival time at destination.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Transfers" type="xs:nonNegativeInteger">
                                <xs:annotation>
                                        <xs:documentation>Number of interchanges.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Distance" type="siri:DistanceType" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Trip distance.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="TripLeg" type="TripLegStructure" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Legs of the trip</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="OperatingDaysGroup" minOccurs="0"/>
                        <xs:element ref="SituationFullRef" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:group ref="TripStatusGroup" minOccurs="0"/>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="TripLegStructure">
                <xs:annotation>
                        <xs:documentation>a single stage of a TRIP that is made without change of MODE or
service (ie: between each interchange)</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="LegId" type="xs:NMTOKEN">
                                <xs:annotation>
                                        <xs:documentation>Id of this trip leg. Unique within trip result.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ParticipantRef" type="siri:ParticipantRefStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>[equivalent of PARTICIPANT in SIRI] IT system that is
participating in a communication with other participant(s)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:choice>
                                <xs:element name="TimedLeg" type="TimedLegStructure"/>
                                <xs:element name="TransferLeg" type="TransferLegStructure"/>
                                <xs:element name="ContinuousLeg" type="ContinuousLegStructure"/>
                        </xs:choice>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="TimedLegStructure">
                <xs:annotation>
                        <xs:documentation>passenger TRIP LEG with timetabled schedule.  Corresponds to a RIDE.<
/xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="LegBoard" type="LegBoardStructure">
                                <xs:annotation>
                                        <xs:documentation>Stop/Station where boarding is done</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegIntermediates" type="LegIntermediateStructure" minOccurs="0"
maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>information about the intermediate passed stop points.
</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegAlight" type="LegAlightStructure">
                                <xs:annotation>
                                        <xs:documentation>Stop/Station to alight</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Service" type="DatedJourneyStructure">
                                <xs:annotation>
                                        <xs:documentation>Service that is used for this trip leg.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegAttribute" type="LegAttributeStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>Attributes that are not valid on the whole service,
but only on parts of the journey leg.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="OperatingDaysGroup" minOccurs="0"/>
                        <xs:element name="LegTrack" type="LegTrackStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Geographic embedding of this leg.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="TransferLegStructure">
                <xs:annotation>
                        <xs:documentation>[a specialised type of NAVIGATION PATH in TMv6] description of a LEG
which links other LEGs of a TRIP where a TRANSFER between different LOCATIONs is required</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:choice>
                                <xs:element name="TransferMode" type="TransferModesEnumeration">
                                        <xs:annotation>
                                                <xs:documentation>Mode that is used for this interchange
between public services.</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="ContinuousMode" type="ContinuousModesEnumeration">
                                        <xs:annotation>
                                                <xs:documentation>Mode that is used for this interchange
between public services.</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                        </xs:choice>
                        <xs:element name="LegStart" type="PlaceRefStructure">
                                <xs:annotation>
                                        <xs:documentation>Stop/Station where boarding is done</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegEnd" type="PlaceRefStructure">
                                <xs:annotation>
                                        <xs:documentation>Stop/Station to alight</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="TimeWindowGroup"/>
                        <xs:group ref="TransferDurationGroup"/>
                        <xs:element name="LegDescription" type="InternationalTextStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Text that describes this interchange.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Length" type="LengthType" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Length of this interchange path.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Attribute" type="GeneralAttributeStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>Note or service attribute.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="PathGuidance" type="PathGuidanceStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Structured model further describing this interchange,
its geographic embedding and accessibility.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element ref="SituationFullRef" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="ContinuousLegStructure">
                <xs:annotation>
                        <xs:documentation>[relates to a specific type of RIDE in TM and NeTEx] leg of a journey
that is not bound to a timetable </xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="LegStart" type="PlaceRefStructure">
                                <xs:annotation>
                                        <xs:documentation>Stop/Station where boarding is done</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegEnd" type="PlaceRefStructure">
                                <xs:annotation>
                                        <xs:documentation>Stop/Station to alight</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Service" type="ContinuousServiceStructure">
                                <xs:annotation>
                                        <xs:documentation>Service of this leg. May be "walk" in most cases, but
also cycling or taxi etc.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="TimeWindowGroup"/>
                        <xs:element name="Duration" type="xs:duration">
                                <xs:annotation>
                                        <xs:documentation>Duration of this leg according to user preferences
like walk speed.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegDescription" type="InternationalTextStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Title or summary of this leg for overview.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Length" type="LengthType" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Length of the leg.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="LegTrack" type="LegTrackStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Detailed description of each element of this leg
including geometric projection.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="PathGuidance" type="PathGuidanceStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Structured model further describing this interchange,
its geographic embedding and accessibility.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element ref="SituationFullRef" minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="LegBoardStructure">
                <xs:annotation>
                        <xs:documentation>Describes the the situation at a stop or station at which the
passenger boards a Leg of a trip including time-related information.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="StopPointGroup"/>
                        <xs:element name="ServiceArrival" type="ServiceArrivalStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>describes the arrival situation a this leg board stop
point (empty for first leg) ( group of attributes of TIMETABLED PASSING TIME, ESTIMATED PASSING TIME, OBSERVED
PASSING TIME)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ServiceDeparture" type="ServiceDepartureStructure">
                                <xs:annotation>
                                        <xs:documentation>describes the departure situation at this leg board
stop point ( group of attributes of TIMETABLED PASSING TIME, ESTIMATED PASSING TIME, OBSERVED PASSING TIME)</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="DistributorInterchangeId" type="xs:normalizedString" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Interchange identifier of the distributing line
/service at its boarding. This is not a reference. This identifier is used to recognize in a distributed
environment (e.g. EU-Spirit), that two systems refer to the same line (or service) while using their own
internal references. In EU-Spirit this is used to decide whether an interchange is in fact a stay-seated
scanario (aka "line ID"). See https://eu-spirit.eu/</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="MeetsViaRequest" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>This stop fulfils one of the via requirements stated
in the request data.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="StopCallStatusGroup"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="LegAlightStructure">
                <xs:annotation>
                        <xs:documentation>Describes the situation at a stop or station at which the passenger
alights from a Leg of a trip including time-related information</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="StopPointGroup"/>
                        <xs:element name="ServiceArrival" type="ServiceArrivalStructure">
                                <xs:annotation>
                                        <xs:documentation>describes the arrival situation at the leg alight
```

```
stop point ( group of attributes of TIMETABLED PASSING TIME, ESTIMATED PASSING TIME, OBSERVED PASSING TIME)</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ServiceDeparture" type="ServiceDepartureStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>describes the departure situation at this leg alight
stop point (empty for last leg) ( group of attributes of TIMETABLED PASSING TIME, ESTIMATED PASSING TIME,
OBSERVED PASSING TIME)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="FeederInterchangeId" type="xs:normalizedString" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Interchange identifier of the feeding line/service at
its alighting. This is not a reference. This identifier is used to recognize in a distributed environment (e.g.
EU-Spirit), that two systems refer to the same line (or service) while using their own internal references. In
EU-Spirit this is used to decide whether an interchange is in fact a stay-seated scanario (aka "line ID"). See
https://eu-spirit.eu/</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="MeetsViaRequest" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>This stop fulfils one of the via requirements stated
in the request data.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="StopCallStatusGroup"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="LegIntermediateStructure">
                <xs:annotation>
                        <xs:documentation>Describes the situation at a stop or station that lies between the
LegBoard and LegAlight stop or station including time-related information.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="StopPointGroup"/>
                        <xs:element name="ServiceArrival" type="ServiceArrivalStructure">
                                <xs:annotation>
                                        <xs:documentation>describes the arrival situation a this leg board stop
point (empty for first leg) ( group of attributes of TIMETABLED PASSING TIME, ESTIMATED PASSING TIME, OBSERVED
PASSING TIME)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ServiceDeparture" type="ServiceDepartureStructure">
                                <xs:annotation>
                                        <xs:documentation>describes the departure situation at this leg board
stop point ( group of attributes of TIMETABLED PASSING TIME, ESTIMATED PASSING TIME, OBSERVED PASSING TIME)</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="MeetsViaRequest" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>This stop fulfils one of the via requirements stated
in the request data.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:group ref="StopCallStatusGroup"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="PathGuidanceStructure">
                <xs:annotation>
                        <xs:documentation> description of a piece of a TRIP. May include geographic
information, turn instructions and accessibility information </xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="PathGuidanceSection" type="PathGuidanceSectionStructure" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>one or more path guidance sections that build the
trip Leg</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="PathGuidanceSectionStructure">
                <xs:annotation>
                        <xs:documentation>[an extended definition of a NAVIGATION PATH in TMv6 to include the
textual navigation instructions] description of a piece of a TRIP. May include geographic information, turn
instructions and accessibility information </xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="TrackSection" type="TrackSectionStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>LINK PROJECTION on the infrastructure network of the
TRIP LEG together with time information</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="TurnDescription" type="InternationalTextStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Textual description of a manoeuvre. This should imply
the information from Manoeuvre, TurnAction, and TrackSection.RoadName.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="GuidanceAdvice" type="GuidanceAdviceEnumeration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>various types of guidance advice given to travelle.<
/xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="TurnAction" type="TurnActionEnumeration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>the range of alternative turns that can be described.<
/xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="DirectionHint" type="InternationalTextStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Textual direction hint for better understanding, e.g.
"follow signs to Hamburg".</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Bearing" type="siri:AbsoluteBearingType" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Absolute bearing after the described manoeuvre.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="PathLink" type="PathLinkStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Description of the type of accessibility on this
navigation section.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="SituationFullRef" type="SituationFullRefStructure" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <xs:simpleType name="GuidanceAdviceEnumeration">
                <xs:annotation>
                        <xs:documentation>various types of guidance advice given to travelle</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                        <xs:enumeration value="origin"/>
                        <xs:enumeration value="destination"/>
                        <xs:enumeration value="continue"/>
                        <xs:enumeration value="keep"/>
                        <xs:enumeration value="turn"/>
                        <xs:enumeration value="leave"/>
                        <xs:enumeration value="enter"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="TurnActionEnumeration">
                <xs:annotation>
                        <xs:documentation>the range of alternative turns that can be described</xs:
documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                        <xs:enumeration value="sharp left"/>
                        <xs:enumeration value="left"/>
                        <xs:enumeration value="half left"/>
                        <xs:enumeration value="straight on"/>
                        <xs:enumeration value="half right"/>
                        <xs:enumeration value="right"/>
                        <xs:enumeration value="sharp right"/>
                        <xs:enumeration value="uturn"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:group name="TransferDurationGroup">
                <xs:annotation>
                        <xs:documentation>[an attribute of a CONNECTION (not INTERCHANGE) in TMv6] calculated
duration in a response taking into ccount the request parameters.; TransferDuration plus waiting time is the
minimum interval between arrival and departure time..</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="Duration" type="xs:duration">
                                <xs:annotation>
                                        <xs:documentation>Overall duration of this interchange.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="WalkDuration" type="xs:duration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Walk time as part of the overall interchange duration.
</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="BufferTime" type="xs:duration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Buffer time as part of the overall interchange
duration. Buffer times, f.e. check in/out times, sometimes are mandatory for using certain services as f.e.
airplanes, ferries or highspeed trains.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:simpleType name="AccessFeatureTypeEnumeration">
                <xs:annotation>
                        <xs:documentation>Allowed values for a AccessFeature.</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                        <xs:enumeration value="lift"/>
                        <xs:enumeration value="stairs"/>
                        <xs:enumeration value="seriesOfStairs"/>
                        <xs:enumeration value="escalator"/>
                        <xs:enumeration value="ramp"/>
                        <xs:enumeration value="footpath"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="TransitionEnumeration">
                <xs:annotation>
                        <xs:documentation>Transition types for interchanges.</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                        <xs:enumeration value="up"/>
                        <xs:enumeration value="down"/>
                        <xs:enumeration value="level"/>
                        <xs:enumeration value="upAndDown"/>
                        <xs:enumeration value="downAndUp"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="PathLinkStructure">
                <xs:annotation>
                        <xs:documentation>[TMv6] a link within a PLACE of or between two PLACEs (that is STOP
PLACEs, ACCESS SPACEs or QUAYs,BOARDING POSITIONs,, POINTs OF INTEREST etc or PATH JUNCTIONs) that represents a
step in a possible route for pedestrians, cyclists or other out-of-vehicle passengers within or between a PLACE.
</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="Transition" type="TransitionEnumeration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether path is up down or level .</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="AccessFeatureType" type="AccessFeatureTypeEnumeration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Type of physical feature of PATH LINK.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Count" type="xs:positiveInteger" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Number how often the access feature occurs in this
PathLink</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:annotation>
                <xs:documentation>========================================== Multi-point trip request
==========================================</xs:documentation>
        </xs:annotation>
        <xs:group name="MultiPointTripRequestGroup">
                <xs:annotation>
                        <xs:documentation>Multi-point trip request structure.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="Origin" type="PlaceContextStructure" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Specifies the origin situation from where the user
wants to start.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Destination" type="PlaceContextStructure" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Specifies the destination situation where the user is
heading to.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Via" type="TripViaStructure" minOccurs="0" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Ordered series of points where the journey must pass
through. If more than one via point is given all of them must be obeyed - in the correct order. The server is
allowed to replace a via stop by equivalent stops.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="NotVia" type="NotViaStructure" minOccurs="0" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>Not-via restrictions for a TRIP, i.e. SCHEDULED STOP
POINTs or STOP PLACEs that the TRIP is not allowed to pass through. If more than one not via point is given all
of them must be obeyed.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="NoChangeAt" type="NoChangeAtStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>no-change-at restrictions for a TRIP, i.e. SCHEDULED
STOP POINTs or STOP PLACEs at which no TRANSFER is allowed within a TRIP</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Params" type="MultiPointTripParamStructure" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Options to control the search behaviour and response
contents.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:complexType name="MultiPointTripParamStructure">
                <xs:annotation>
                        <xs:documentation>Multi-point trip request parameter structure.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="TripDataFilterGroup"/>
                        <xs:group ref="TripMobilityFilterGroup"/>
                        <xs:group ref="MultiPointTripPolicyGroup"/>
                        <xs:group ref="TripContentFilterGroup"/>
                        <xs:element name="FareParam" type="FareParamStructure" minOccurs="0"/>
                        <xs:element name="Extension" type="xs:anyType" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:group name="MultiPointTripPolicyGroup">
                <xs:annotation>
                        <xs:documentation>Policies that control the multi-point trip search behaviour.</xs:
documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:group ref="BaseTripPolicyGroup"/>
                        <xs:element name="TransferLimit" type="xs:nonNegativeInteger" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>The maximum number of interchanges the user will
accept per trip.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="OptimisationMethod" type="OptimisationMethodEnumeration" minOccurs="0"
>
                                <xs:annotation>
                                        <xs:documentation>the types of algorithm that can be used for planning
a journey (fastest, least walking, etc)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="MultiPointType" type="MultiPointTypeEnumeration" default="anyPoint"
minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>If a solution for any one of multiple origin
/destination points is sufficient. Or a distinct solution for each of the origin/destination points has to be
found.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:simpleType name="MultiPointTypeEnumeration">
                <xs:annotation>
                        <xs:documentation>How the multiple origin/destination points should be considered</xs:
documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                        <xs:enumeration value="anyPoint"/>
                        <xs:enumeration value="eachOrigin"/>
                        <xs:enumeration value="eachDestination"/>
                </xs:restriction>
        </xs:simpleType>
        <xs:annotation>
                <xs:documentation>========================================== Multi-point trip response
========================================</xs:documentation>
        </xs:annotation>
        <xs:group name="MultiPointTripResponseGroup">
                <xs:annotation>
                        <xs:documentation>Multi-point trip response structure.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="MultiPointTripResponseContext" type="ResponseContextStructure"
minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Context to hold trip response objects that occur
frequently.</xs:documentation>
```

```
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="MultiPointTripResult" type="MultiPointTripResultStructure" minOccurs="
0" maxOccurs="unbounded">
                                <xs:annotation>
                                        <xs:documentation>The trip results found by the server.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:complexType name="MultiPointTripResultStructure">
                <xs:annotation>
                        <xs:documentation>Structure for a single trip result and its accompanying error
messages.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="ResultId" type="xs:NMTOKEN">
                                <xs:annotation>
                                        <xs:documentation>Id of this trip result for referencing purposes.
Unique within multipoint-trip response.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="ErrorMessage" type="ErrorMessageStructure" minOccurs="0" maxOccurs="
unbounded">
                                <xs:annotation>
                                        <xs:documentation>Error messages related to trip result.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:choice>
                                <xs:element name="Trip" type="TripStructure">
                                        <xs:annotation>
                                                <xs:documentation>Information on the trip.</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="TripSummary" type="TripSummaryStructure"/>
                        </xs:choice>
                        <xs:group ref="MultiPointWaitTimeGroup" minOccurs="0"/>
                        <xs:element name="TripFare" type="TripFareResultStructure" minOccurs="0" maxOccurs="
unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="UserPreferencesStructure">
                <xs:annotation>
                        <xs:documentation>A collection of user preferences to allow the algorithm to narrow-
down multimodal trip combinations to those that may be suitable for the user.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="lowprice" minOccurs="0" maxOccurs="1" default="average">
                                <xs:annotation>
                                        <xs:documentation>User's preference that the trip has a low price.</xs:
documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="important"/>
                                                <xs:enumeration value="average"/>
                                                <xs:enumeration value="unimportant"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="speed" minOccurs="0" maxOccurs="1" default="average">
                                <xs:annotation>
                                        <xs:documentation>User's preference about the speed/short duration of
the trip.</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="important"/>
                                                <xs:enumeration value="average"/>
                                                <xs:enumeration value="unimportant"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="comfort" minOccurs="0" maxOccurs="1" default="average">
                                <xs:annotation>
                                        <xs:documentation>User's preference about the comfort/convenience of
the trip, including few changes, short transfers and convenient vehicle types.</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="premium"/>
                                                <xs:enumeration value="average"/>
                                                <xs:enumeration value="basic"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="sportiness" minOccurs="0" maxOccurs="1" default="average">
                                <xs:annotation>
                                        <xs:documentation>User's endurance and fitness for sportive, self-
powered trip legs (walk, cycle, etd.).</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="high"/>
                                                <xs:enumeration value="average"/>
                                                <xs:enumeration value="low"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="environmentalsafety" minOccurs="0" maxOccurs="1" default="average">
                                <xs:annotation>
                                        <xs:documentation>User's preference about climate/environmental-
friendliness of the trip (option 'low' might have no effect).</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="high"/>
                                                <xs:enumeration value="average"/>
                                                <xs:enumeration value="low"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="weatherprotection" minOccurs="0" maxOccurs="1" type="xs:boolean"
default="false">
                                <xs:annotation>
                                        <xs:documentation>User's preference not be exposed to eventual weather
hazards, thus, to avoid walks and micromobility.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="extrasafe" minOccurs="0" maxOccurs="1" type="xs:boolean" default="
false">
                                <xs:annotation>
                                        <xs:documentation>User's preference for an extra high level of safety.<
/xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="extrareliable" minOccurs="0" maxOccurs="1" type="xs:boolean" default="
false">
                                <xs:annotation>
                                        <xs:documentation>User's preference for an extra high reliability (low
probabilty of delays, cancellations, etc.).</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="extraaccessible" minOccurs="0" maxOccurs="1" type="xs:boolean"
default="false">
                                <xs:annotation>
                                        <xs:documentation>User's preference for perfect accessibility (lifts,
ramps, no stairs, etc.), including extra transfer time, short transfer distances, etc.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="scenic" minOccurs="0" maxOccurs="1" type="xs:boolean" default="false">
                                <xs:annotation>
                                        <xs:documentation>User's preference for scenic, touristic impressions
on some parts of the trip.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:group name="MultiPointWaitTimeGroup">
                <xs:annotation>
                        <xs:documentation>Group for wait times at origin/destination.</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="OriginWaitTime" type="xs:duration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Additional wait time at origin of this trip.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="DestinationWaitTime" type="xs:duration" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Additional wait time at destination of this trip.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
        <xs:group name="TripStatusGroup">
                <xs:annotation>
                        <xs:documentation>Parameters which describe the current status of a TRIP</xs:
documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="Unplanned" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether this trip is an additional one that has not
been planned. Default is false.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Cancelled" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether this trip is cancelled and will not be run.
Default is false.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Deviation" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether this trip deviates from the planned service
pattern. Default is false.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Delayed" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether this trip is delayed. Default is false.</xs:
documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Infeasible" type="xs:boolean" default="false" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Whether this trip cannot be used, due to operational
delays and impossible transfers. Default is false.</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:group>
</xs:schema>
```

8.1.2. Visualization from XML Spy


Extension
Params
Options to control the search behaviour and response contents.
UserPreferences
UserPreferencesStructure
lowprice
User's preference that the trip has a low price.
speed
User's preference about the speed/short duration of the trip.
comfort
User's preference about the comfort/convenience of the trip, including few changes, short transfers and convenient vehicle types.
sportiness
User's endurance and fitness for sportive, self-powered trip legs (walk, cycle, etd.).
environmentalsafety
User's preference about climate/environmental-friendliness of the trip (option 'low' might have no effect).
weatherprotection
User's preference not be exposed to eventual weather hazards, thus, to avoid walks and micromobility.
extrasafe
User's preference for an extra high level of safety.
extrareliable
User's preference for an extra high reliability (low probabilty of delays, cancellations, etc.).
extraaccessible
User's preference for perfect accessibility (lifts, ramps, no stairs, etc.), including extra transfer time, short transfer distances, etc.
scenic
User's preference for scenic, touristic impressions on some parts of the trip.

9. Effective Implementation in Trip Planner Algorithms

The exact functionality of a trip planner algorithm is not defined by OJP and is left open to implementers. However, the following general considerations should be observed by any implementation.

9.1. Default Behaviour

The proposed user preferences are an optional addition to the existing OJP 1.0 TripRequest parameters.

If the **`UserPreferences`** element is left away altogether, the user preferences should obviously have no influence, and the algorithm should adopt an average, "default" behaviour.

Equally, if a given user preference is set to the default value, it should have no influence and the default behaviour should result.

If a non-default preference is provided, on the other hand, the algorithm should change its search strategy and the scoring and selection of results in an appropriate way.

9.2. Transparency and Literal Sense

Trip planners should adapt user preferences in a transparent, easy to understand fashion. The impact of every single preference should intuitively be clear from the wording (literal sense) of the preference and the effect should be observable in the results of the trip planner.

Examples:

- *"If I set sportiness to low, the planner provides no results with bicycles and only short walks."*

- *"If I set lowprice to unimportant, I get trip plans including taxis and ridehailing. If I set lowprice to important, I get none of these."*

Rules like this should be made transparent. They are easy to understand and correspond to the literal sense of the words used.

9.3. Basic Influence Patterns

User preferences should influence trip planning in these ways:

- by excluding or including certain modes or operators. *E.g. when low sportiness, exclude bikes.*
- by controlling other medium-level or low-level parameters, such as maximum walk duration, walking speed, etc.. This is legitimate even if those factors could be controlled directly through low-level parameters. *Example: when sportiness is high, use walking speed factor 150 % and allow for walks up to 15 minutes. When sportiness is low, use 50 % / 5 minutes.*
- by scoring of the results.

9.4. Excluding or including certain modes and/or operators

Certain preferences should trigger the exclusion of some modes or some operators, or a low scoring of these.

Examples:

- *when sportiness is low or when wheaterprotection is yes (or other), exclude bikes.*
- *when lowprice is important, exlcude taxis and ride hailing.*

9.5. Controlling Other Medium or Low-level Parameters

Trip planners typically provide many parameter to control the planning.

OJP 1.0 provides a set of medium and low level parameters, as analyzed above, such as maximum walk duration, walking speed, etc..

Actual trip planners may provide many more parameters available through proprietary APIs.

It is reasonalbe and legitimtate to control those factors based on the high-level user preferences, even if those factors could be controlled directly through low-level parameters in OJP or proprietary interfaces.

Examples:

- *when sportiness is high, use walking speed factor 150 % and allow for walks up to 20 minutes.*
- *When sportiness is low, use 50 % / 5 minutes.*

9.6. Scoring

Scoring means that for each search result, a score value (a floating-point number) is calculated which should express the degree of match or fitness of the result to the given query.

The score for a trip should be based on various quality measures (speed, price, comfort, etc.), depending on how important they are to the user.

From the user preferences, weight factors may be derived to influence the scoring formula.

Example: if lowprice is important, the "price" quality may get a weight factor 3; for "average", the factor would be 1; for "unimportant", the factor may be reduced to 0.3.

With a well-crafted scoring algorithm, selecting the right results becomes as simple as sorting for the score and displaying only the best n results.

The actual mathematics of the scoring are out-of-scope for the OJP specification and left open for the implementors. However, the basics should be made transparent.

10. Impact Analysis Based on Examples

The given 10+ new parameters, in combination with the existing OJP 1.0 filters, will allow countless combinations. The study of some "likely" examples for some (stereotypic) example users may help to get an clearer idea of the impact of user preferences.

In all cases, we are referring to a medium- or long-distance trip from and to urban or suburban areas.

User Stereotype	User Preferences	Impact
student	lowprice=important, sportiness=high, environmentalsafety=high	trips with bikes, scooters, PT get high scores; taxi, ridehailing, carpooling get low scores.
business traveller	lowprice=unimportant, speed=important, comfort=premium, weatherprotection, extrareliable	fast trips with PT, taxi, some ride-hailing operators.

foreign tourist	scenic, extrasafe, weatherprotection	trips with PT and Taxi, preferring scenic routes if available.
family trip	lowprice=important, environmentalsafety=high, weatherprotection	trips with PT.
elderly, vulnerable person	weatherprotection, extraaccessible, extrareliable, sportiness=low	trips with PT, taxi, few changes, longer transfer times, short walks only

11. Data Protection

The proposals are, in our estimation, fully compliant with the EU General Data Protection Regulation:

- no personal information such as name, gender, age, home address, citizenship, job, etc. is involved.
- no identifiers, text fields or numbers are involved. Only given values from small "enumerations" are used.
- a given set of user preferences will allow only very loose and uncertain matching with given persons or narrowing down.
- the OJP 1.0 features like IndividualTransportOptions or ModeFilters are comparable regarding data protection: they already do provide some hints about the user. The proposal does not go beyond the OJP 1.0 degree of data exposure.
- as with OJP 1.0, TripRequest remains a purely stateless query.

To fully comply with GDPR, implementations of OJP TripRequest should observe general good practices such as not to log IP addresses and to delete log files after short periods of time.

12. Impact on Booking and Sales of Mobility Services

User preferences will help a user to get trip plans that really match his or her personal needs. Trip planning proposals should become decisively better, more useful and more relevant.

Apart from that, there should be no difference when it comes to booking.