LinkingAlps OJP profile

Design, features, and specification

Version: 02.00

Dissemination level: Restricted (Partners only)

WP: D.T1.5.1 – Specification of API interface (specification)

Author: BLIC (PP7)

Date: 01. April 2021

Document Information

Authors

Name	PartnerNo	Organisation
Christopher König	PP7	BLIC
Jan Grüner	PP7	BLIC
Markus Windegger	PP7	BLIC
Christian Landrock	PP7	BLIC

Peer-Reviewer

Name	PartnerNo	Organisation

Document History

Date	Version	Summary of changes	Dissemination
11.05.2020	00.01	Initializing Document	-
30.10.2020	01.00	First Public Version	Project Partners
11/2020 - 03/2021	01.0X	Changes after Feedback PP/Suppliers New sections; • Gazetteers • Expected Behaviour • OJP Standard Comprehensive List (see Annex)	Project Partners
04.01.2021	02.00	2nd Public Version	Project Partners

Preface

Acronym:	LinkingAlps
Title:	Innovative tools and strategies for linking mobility information services in a decarbonised Alpine Space
Project number:	740
Start Date:	01-10-2019
End Date:	31-06-2022
Call number:	4th call
Priority:	Priority 2 - Low Carbon Alpine Space
Specific objective:	SO2.2 - Increase options for low carbon mobility and transport

Abbreviations

Abbreviation	Definition
ATE	AustriaTech
ARIA	Regional Agency for Innovation and Purchasing Ltd
BLIC	Consulting company for control, information and computer technology GmbH
CMTo	Metropolitan City of Turin
FoT	Federal Office of Transport
LINKS	LINKS Foundation - Leading Innovation & Knowledge for Society
NeTEx	Network Timetable Exchange (CEN/TS 16614 ff)
OJP	Open API for distributed Journey Planning (EN/TS 17118:2017) that allows to engineer one universal interface to link services
RRA-LUR	Regional Development Agency of the Ljubljana Urban Region
SBB	Swiss Federal Railways
SIRI	Service Interface for Real time Information (CEN/TS 15531)
STA	South Tyrolean Transport Structures
VAO	Traffic Information Austria

Interreg
Alpine Space
LinkingAlps
European Regional Development Fund

Table of Contents

Table of schema

Table of Tables

1 Introduction

Transit users travelling across borders often face the problem that travel information for the entire route is not visible immediately. In most cases, travellers must switch between the information systems of different operators, regions or countries in order to plan their entire journey. The LinkingAlps project addresses this problem in the Alpine Space. Its aim is to create a standardised exchange service of travel information between the individual travel information service providers. This way, information can be exchanged between different systems and compiled into a continuous travel chain. Travelers can thus view the entire trip from origin to destination on a single service. This document describes how communication and supported services are handled within LinkingAlps.

This document is based on the OJP (Open Journey Planner) standard version 1.0. It describes in detail the usage of the OJP standard. Among other topics, decisions regarding optional features and differences to the OJP standard are described. Furthermore, special features and important aspects are pointed out. The document does not cover a complete description of the entire OJP Standard. Therefore, the implementation of the profile also requires the use of the OJP standard in the version mentioned above.

The hyperlinks to the official documentation of the OJP standard as well as other related information can be found in Table 1. Generally, the implementation of OJP services should follow best effort principles: if it is possible to support a functionality it should be included.

Table 1 OJP documentation and references.

Name	Link
Standard	http://www.normes-donnees-tc.org/wp-content/uploads/2017/01/TC_278_WI_00278420_E-RS-170118-final3.pdf
xsd-Files	http://www.normes-donnees-tc.org/wp-content/uploads/2017/01/OJP-xsd_CEN-2016.zip
Forum	https://forum.vdv.de/viewforum.php?f=88
GitHub	https://github.com/VDVde/OJP

2 Terms

The terms used terms in this document are described in Table 2.

Table 2 List of used terms.

Term	Explanation
Active system	The active system integrates the routing information from several local journey planners to a combined

Term	Explanation
	seamless route. It is composed of a passive system and a Distributing system. It communicates through an OJP interface. It is a journey planning engine with OJP capabilities. Via the distributing system it is able to detect journeys through adjacent or remote regions and able to create OJP Trip Compositions.
Adjacent region	Region which is adjacent to the local region and has its own "local" journey planning systems.
Adjacent system	Alias for neighbouring system; participating system of an adjacent region.
Distributing system	System that distributes journey planning enquiries to other systems. It sends the request for journey-parts through areas to the corresponding passive servers, receives the responses and is able to create OJP Trip Compositions. It has the knowledge about gazetteers and is able to collect information about exchange points for the whole system.
End user	User of an "end user application". Person asking for journey planning information by using an end user application. Enquirer of a journey plan with a start, an end point and some travel preferences.
End user application	Application used by the end user to have access to JP information generated by the Distributed Journey Planning Service (DRJP). It can be a third-party application connecting by OJP interface to a Participating system or the User Interface Participating system. The providers of the end user applications are named “OJP users” in the LinkingAlps project.
Enquirer	End user asking for information.
Estimated data	Predicted arrival or departure time of a particular means of transport at a particular stop. In the case of real time data, it can change several times during the journey.
Exchange point	Stop points or stop places where the trip leg of one system is connected to the trunk leg of another system. This includes regional stops which match with stops for long distance, or regional stops from adjacent regions. Exchange points are mainly but not exclusively located at borders and in bigger cities.
Exchange point database	Repository/view on a database or a service that is able to list the relevant exchange points of the distributed service. It can be a static system-wide database or be

Term	Explanation
	generated dynamically with requests for exchange points to the responding services.
Gazetteer	Directory of common objects across the local journey planner systems and its system borders. It enables the active system to find the passive system for all geolocations (stops, stations, POIs, address etc.). The gazetteer acts system wide.
Home system	Participating system called by the end user application. It is the system that takes care of the end user travel information request and provides an answer.
Journey	The movement of a traveller from a start point to an end point by using one or more transport modes.
Journey Planner (JP)	System that calculates the journey for a given request. It is able to accept requests directly from end-user services. It is a generalization of OJP Router and OJP responder.
Journey Planning System (JPS)	Alias: Journey Planner.
Local Journey Planner (LJP)	System with a routing engine and access to multimodal data with a particular local, regional or national coverage; “local” underlines its focus on a specific coverage that is limited. LJPs have no transregional (or distributed) OJP routing capabilities.
Location database	Database with all locations relevant for the whole system. The location database is part of the gazetteer.
Long distance schedule data	Schedule data of long-distance traffic.
Long distance transport connection	Trunk legs of the routes that connect at least two OJP systems. They are used to connect two neighbouring or remote systems. Exchange points are defined along the trunk leg which defines all the neighbouring systems.
Neighbouring system	Alias for adjacent system.
OJP Implementer	Travel information service provider that is implementing an OJP service exchange (in most cases on the back-end system of an end user service).
OJP Interface	Application Programming Interface (API) based on CEN/TS 2017: OpenAPI for distributed journey planning and specified in D.T1.5.1 (Specification of the API interface) (including a LinkingAlps OJP Profile).
OJP Trip Composition	Process of combining the different trip legs coming from different OJP Responders. It is transmitted via OJP Interface.

Term	Explanation
OJP User	End-user service provider that uses OJP services from local JPs to provide an end-user service.
Open Journey Planning (OJP)	Standard for communication for distributed journey planning (CEN/TS 17118:2017).
Participating system	Local journey planner, part of the OJP system architecture and the appropriate OJP service.
Passive system	Local journey planner with an OJP interface (API) being able to respond to requests from distributing systems. It is an information source within the system without distributed journey planning capabilities. It communicates through an OJP interface as a responding system. Alias OJP responder, responding system.
Place	Geographic PLACE of any type which may be specified as the origin or destination of a trip. Possible types are: • PointOfInterest • Address • TopographicPlace • StopPlace (stop place): Comprises of one or more locations where vehicles may stop and where passengers may board or leave vehicles or prepare their trip. • StopPoint (stop point): where passengers can board or alight from vehicles
Real time data	The real time of a particular means of transport at a particular stop; only sent after the arrival/departure of the vehicle at a particular stop.
Remote region	Region which is not adjacent to the local region. A remote region is covered by a local LJP.
Remote system	Participating system of a remote region.
Ring Connection Composer (RCC)	EU-Spirit Component: System which is asked by the active servers. In order to fulfil queries of active server, it uses the services of the passive servers, i.e., it distributes the queries and composes the corresponding responses.
Schedule data	Planned data for public transport services.
Service	Technical, self-sufficient unit that bundles related functionalities into a complex of topics and makes them available via a clearly defined interface.
System	Delimitable "structure" consisting of various components which can be regarded as a common

Term	Explanation
	whole due to certain ordered relationships between them.
Trip	Alias: Journey.
Trip leg	Local part of a trip which is calculated by a single Local Journey planning system.

3 LinkingAlps System Architecture

The system used in the LinkingAlps Project is based on a network of existing local, regional or national travel information services (routing platforms) that collaborate on the basis of the CEN OJP exchange interface [1] in order to exchange travel information and routing results.

The LinkingAlps System Architecture consists of the following main components (see D.T1.3/4 (Requirements Document)):

- **A participating system** is part of a decentralised network of journey planners (JPs) established through OJP interfaces. Participating systems can have an active or passive role in the architecture, depending on their tasks. Participating services are distinguished according to the functionality and scope to active and passive systems.
- A **Local Journey Planner** is a system with a routing engine and access to multimodal data with a particular local, regional or national coverage. "Local" underlines its focus on a specific coverage that is limited. A LJP itself has no transnational (or distributed) OJP routing capability.
- **An active system** is a travel information service, in particular a journey planner, to which the end user is connected (that means it is the enquirer's home system). It provides an openAPI service (exchange service), the OJP interface, and actively requests the information from other services through a distributing system. Hence the active system contains a distributing system that has the distribution logic in order to gather the needed information. The active system further integrates the routing information from several local journey planners (active or passive systems) to a combined (seamless) route. Doing so, it has an OJP routing algorithm, that facilitates the trip composition. In order to gather the required trip information, in some cases, the active system also responds to requests from other systems through the OJP interface and consequently takes over the tasks of an OJP responder. In the system architecture description, it is called OJP router, indicating that it comprises an OJP interface, a distributing system and OJP routing. An active system can but must not contain the end user service as well.
- **A passive system** is a travel information service, in particular a local journey planner, that provides an openAPI web service (exchange service and OJP interface) so that other clients can access information from the server. Passive systems are so called **OJP responders** and deliver responses to requests, over OJPinterface, coming from active systems. Passive systems have no distributing system and do not provide OJP routing.

Both, active and passive systems can be in the role of a responding system, as communication in the network is on a peer-to-peer basis. Therefore, the terms "active " and "passive" systems are not used in the system architecture component description (Figure 1). A complete description of the LinkingAlps System Architecture, as well as the request flow diagrams between the active and passive systems (e.g., trip request), can be found in the Deliverable DT1.3.1 of the LinkingAlps project.

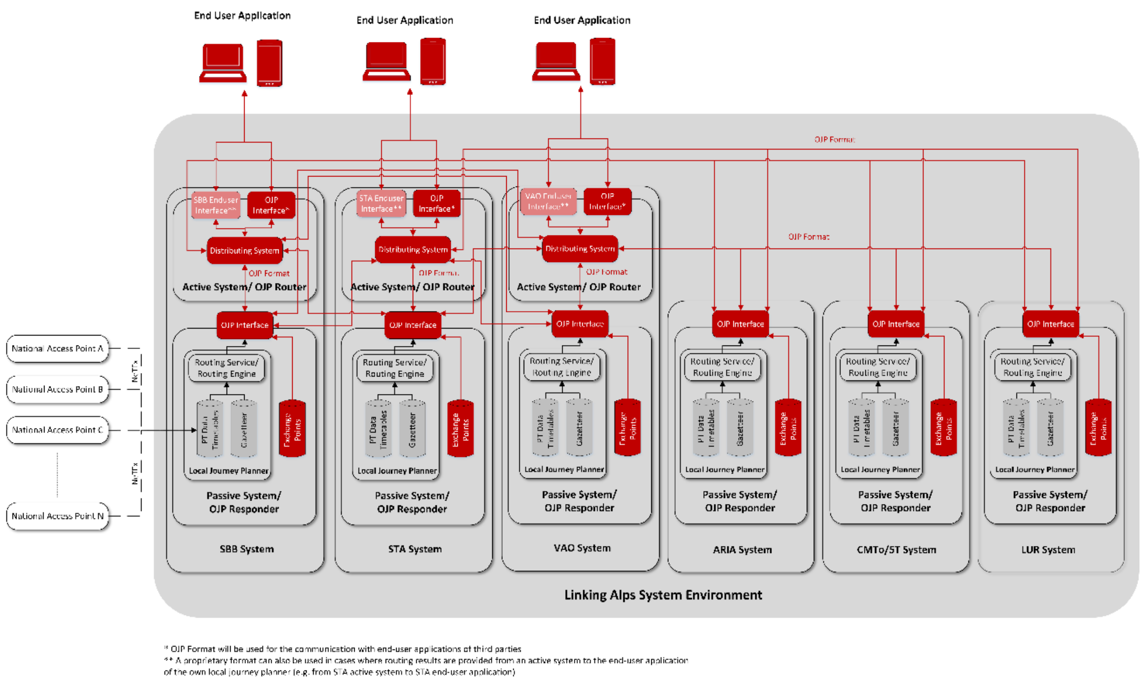

Figure 1 System architecture overview of LinkingAlps System Environment.

An overview of the System-ID (names) used within the LinkingAlps OJP profile can be found in Annex 11.1.

3.1 Communication

3.1.1 Accessing Data (API)

All systems offering OJP services (see chapter 4.1) must provide them via HTTPS using current, state of the art encryption (HTTPS with TLS 1.3). Depending on future developments, an upgrade to different/newer signing methods for the used certificates might become necessary at some point. For the transmission of data (XML) between different systems, via this OJP profile, the implemented API uses a HTTPS REST API. In order to connect to the service, a unique and unguessable API-Key/ID must be used in order to identify and manage access for specific users (groups). For authentication purposes an API-Key shall be used. The Key shall be embedded within the header (bearer token) and body of the request. For more information on the use of the API see Deliverable DT1.4.1 of the LinkingAlps project.

3.1.2 Charset

For the transmission of textual information between the systems via this OJP profile, the UTF-8 charset must be used by all participants. It supports every necessary character and is known to (nearly) all known operation systems. If any Local Journey Planner (LJP) uses a different charset internally, the operator of this LJP needs to implement, validate and maintain a charset conversion functionality between its own charset and UTF-8. The used charset must be specified in the head of the XML file.

3.2 ExchangePoints

Exchange points are stop points or stop places where the trip leg of one system is connected to the trunk leg of another system (Figure 2). This includes regional stops matching stops for long distance or regional stops from adjacent regions. Exchange points are mainly but not exclusively located at borders and in bigger cities.

The identification of exchange points is done by looking for stops and stations used by multiple service providers (a.k.a. areas). From the view of a local journey planner, exchange points may be inside of the covered area, outside of the covered area, or at the border of the covered area. It is worth noting that exchange points do not only exist between services of geographical adjacent areas, but also between services of geographical remote areas. These geographical remote areas are adjacent areas for the LinkingAlps system.

Within the context of LinkingAlps, the detection of exchange points will be initially based on a static exchange, being that the perspective of having an automated system in the future is being considered (dynamic approach). The details for both methods are described in D.T1.3/4 (Requirements Document) of the LinkingAlps project.

For the initial detection of exchange points across multiple systems (journey planers) may use different names and IDs a few base parameters are needed. The following parameters may be useful:

- GPS Coordinates
- IDs
- Modes

However, this initial process is not part of the overall OJP profile and therefore not part of this profile. The process itself is defined within the context of D.T1.3 (ExchangePoints) of the LinkingAlps project.

In some cases, it may be useful for a single ExchangePoint to cover multiple individual stop places (e.g. Bern Central Station consists of four individual StopPlaces: The station itself, “RBS Bahnhof”, “Postauto Bahnhof" and “Bernmobil”). For this reason, it is more useful to give these individual stop places the same ExchangePointID to allow for possible grouping of close and nearby stops. However, this combining of Stops should only be done if the following conditions are fulfilled:

- They must be exchange points
- The system must be able to calculate further trips/legs from these points

Figure 2 describes an exchange point between two different systems. This common station (stop point or stop place) between these systems must have the same ExchangePoint-(Meta)-ID in both systems (see section 4.2.1).

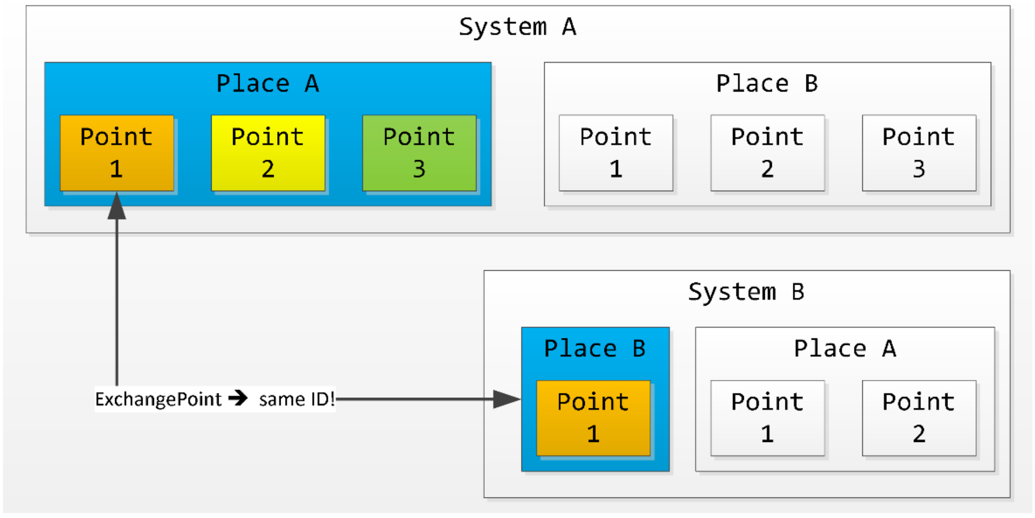

Figure 2 ExchangePoint

4 OJP Services in LinkingAlps

This chapter describes detailed information about the OJP Services used by the LinkingAlps OJP profile. Therefore, it contains descriptions of the supported and unsupported fields and filters as well as example messages for requests and responses of the services. A comprehensive overview of the fields and parameters for the individual services, supported by this profile, can be found in Annex 11.2. The overview is the basis for this profile document[1]. This profile specification is written from the perspective of the passive system. Therefore, requests come

[1] In case of deviations between the documents the comprehensive overview is to be used.

from the active system to the passive system, while responses are sent from the passive system towards the active system. This approach is also true for the comprehensive overview file.

4.1 Supported OJP services

There are currently seven different OJP services described in the OJP Standard [1]. Within the context of the LinkingAlps project, six of them are initially supported, excluding the OJPFare service (see Table 3). It should be noted that this service is likely to be included at some point in the future. For the sake of complete documentation, the names of the related OJP schema files and a short description of each service are given.

Table 3 Supported OJP services.

Service name	Service in OJP CEN/TS 17118:2017	Schema file	Supported
OJPLocationInformation	Location information	OJP_Locations.xsd	Yes
OJPTrip	Trip request	OJP_Trips.xsd	Yes
OJPStopEvent	Departure board	OJP_StopEvents.xsd	Yes
OJPTripInfo	Trip/Vehicle information	OJP_TripInfo.xsd	Yes
OJPExchangePoints	Exchange points	OJP_Locations.xsd	Yes
OJPMultiPointTrip	Distributed journey planning	OJP_Trips.xsd	Yes
OJPFare	Ticket price calculation	OJP_Fare.xsd	No

4.2 Global decisions for all services

This section refers to all the decisions affecting multiple OJP services or the general implementation. The OJP-Profile used in LinkingAlps is V1.0, with the intention to define and provide a migration process to later OJP-Versions, as this is deemed an important factor for the long-term success. However, the specification of a migration path is not part of this profile specification document.

4.2.1 ID handling

Within the context of information processing, unique identifiers provide a simple method to distinguish between (similar) objects. However, these IDs are usually limited to a certain scope, intention or use case and thus many different methods for their generation exist. In many cases these IDs may also encode object features as well, making it easy to ascertain information or grouping information just by looking into the different IDs.

For projects such as LinkingAlps this means that, due to organisational differences, a common ID system for objects, such as stations, locations etc., is not available in the existing systems. However, in most cases, this may not result in any issue, as due to the architecture the individual systems will only provide information for their own region/coverage area. Thus, the individual IDs can remain untouched.

In order to process information in a distributed journey planning context, exchange points from different systems need to be recognised as the same or different stations/places. Therefore, all exchange points need to adhere to the same principles when it comes to generating their IDs. This may not be only limited to exchange points, however, in that case it is of upmost importance. As the OJP standard document does not specify an ID structure for exchange points it was decided that the LinkingAlps project will follow the European **NeTEx** structure format [2]:

```
General Format:
[CC]:[LC]:[OT]:[TI]:[ID]

Example:
SI:SI0:EP:350271b3-c0cd-43e0-a244-940f744b4875:IJPP
```

Please note that all separating characters (“:”) are mandatory, even if a field is empty.

The meaning and length of the different elements is given in Table 4. It should be noted that these IDs should only be used for the identification of exchange points and not for any other characteristic.

Table 4 NeTEx Format elements.

Abbreviation	Name
CC	ISO 3166-1 code of the country (2 characters)
LC	Code uniquely identifying the locality or the provider within the country (region code like the European NUTS code, an authority code, …). The European NUTS code is recommended here, however, this code is not mandatory if the other elements make the code unique (but surrounding ":" must be present).
OT	NeTEx object type (ServiceJourney PassengerStopAssignment Line, etc. using exactly the tag format, UpperCamelCase and no space) corresponding to the XML tag and is provided to avoid any collision of single identifiers being used for different types of objects. A small exception is defined for StopPlace in order to differentiate between monomodal and General STOP PLACEs, the [object will be MonomodalStopPlace or GeneralStopPlace instead of StopPlace (see 6.1 Stop place hierarchy).
TI	Technical identifier for the object, it can be whatever code the system defines (built of upper case or lower case non accented characters, numbers " "--" and "_") but shall be unique for the object and durable (a single object cannot change its identifier).
ID	Code identifying the organisation that defines/manages the IDs (e.g., SBB for Swiss exchange points).

In order to allow compatibility with other systems/projects (e.g., EU-Spirit, see chapter 7 of this document) the LinkingAlps project uses the concept of "meta IDs" where the exchange point ID's are added to the *PrivateCode* element in *StopPointStructure* or *StopPlaceStructure,* keeping the original ID as well. For a better understanding see the following simplified example:

```
<StopPoint>
      <siri:StopPointref>idLocalPoint</siri:StopPointref>
      <StopPointName>
            <Text>the name of the stop</Text>
            </StopPointName>
      <PrivateCode>
            <System>LinkingAlps</System>
            <Value>idExchangePoint</Value>
      </PrivateCode>
      <-- ... -->
</StopPoint>
```

This method will also allow a relatively easy addition of other systems following the same concept, as multiple System/Value instances are possible in the PrivateCode section, which can be easily identified.

The System-ID for LinkingAlps is ***LinkingAlps***.

4.2.2 Language

All language fields and filters in the used OJP services are supported. The behaviour and order of language transmission is described in the following subsections.

4.2.2.1 Names

Names (e.g. cities, station names or general locations) should be sent in the desired language of the requesting user. If a name is not available in the desired language, the original name in the language of the region should be used. This guarantees at least recognition of texts on the passenger information systems. In the unlikely event that the original name is not available in the language of the region, the name should be transmitted in English.

A translation of the names from English to the native language of the requesting end-user or to the original language of the region, by using a translating algorithm, is not recommended. The (historical) name of the object (e.g. city, station, POI, line) in question may be untranslatable, in the sense that it may lose its correct meaning. Also, this can be done by the end user application, if necessary.

4.2.2.2 Texts and descriptions

All other texts and descriptions should be transmitted in the native language of the requesting end user. If the native language of the requesting end user is not available, the texts and descriptions should be transmitted in English, allowing for the use of external translation tools. If English is not available, the original language of the region should be used.

In case the requested language is not available an error message (see 4.2.2.3) will be sent.

4.2.2.3 ErrorMessages regarding the language

Language related error messages can be found in [1] section 8.3.6 and are listed as well in Table 5. It is important to note that general OJP error messages can appear in any message.

Table 5 Language related ErrorMessages.

Code	Description
OJPGENERIC_LANGUAGENOTSUPPORTED	For the display of texts within the result, the server does not support (at least within the context of this request) the language required by the requestor.
OJPGENERIC_EXCEPTIONFROMREQUESTEDLANGUAGE	For the display of texts within the result, the server does not support the language required by the requestor for all of the occurring text elements.

4.2.2.4 Outlook regarding language handling (future OJP Versions)

As of the current version of OJP (V1.0) providing multiple text fields for the use of different languages within the request or response is not supported. However, starting with version 1.1 of the standard sending multiple text elements within the context of the response is allowed, as this is a necessary feature for multilingual regions/countries. An example of such a response is given below.

```
<LocationName>
      <Text xml:lang="de">Bern</ojp:Text>
      <Text xml:lang="fr">Berne</ojp:Text>
      <Text xml:lang="it">Berna</ojp:Text>
      <-- ... -->
</LocationName>
```

For further information regarding the usage of future OJP versions see section 8 of this document.

4.2.3 Error Handling and Messages

The error states when operating OJP services are signalized by error codes, which can be transferred into the ErrorMessage structure. ErrorMessage can occur multiple times in most places and therefore also describe a frequently occurring, multi-layered error situation. In ErrorMessage, error codes can occur that [1]:

- Are inherited from the SIRI services (see [1] section 8.3.5),
- Describe general OJP error situations across services, or
- Indicate service-specific error situations.

The OJP error codes are indicated by a prefix that is specified by the respective service (e.g. “STOPEVENT_”) or shows that there is a general error state (“OJPGENERIC_”) [1].

General OJP Error States are defined in [1] section 8.3.6, service dependent error codes are given in the context of each service. Service specific errors are part of the respective response and can be found in the corresponding sections for each service in this document. The general error structure (ErrorMessageStructure, Table 6) within the OJP Standard is covered in [1] section 8.4.4.2.

Table 6 ErrorMessageStructure.

	+Structure	**Section**
Code	xs:normalizedString	Code for the error situation.
Text	+InternationalText	Description of the error situation.

4.2.4 Warning messages

In some situations, it may not be necessary to abort a request with an error message but rather continue without the specific information. In these cases, the processing system/end-user needs to be informed about the issue. Within the OJP standard warning messages are created by sending Status true while simultaneously returning an error message. A simple representation of the possible situations is given in Table 7 (see [1] section 8.4.1.2).

Table 7 Service delivery status.

Status	**Error**	**Result/Meaning**
TRUE	FALSE	OK
TRUE	TRUE	Warning

FALSE	TRUE	Error

4.2.5 Transport modes

The transport modes are used to distinguish between different ways of transportation. Different transport modes feature different constraints in terms of usage. Within the context of the OJP standard, all transport modes are referred to by their English name and not via any kind of coded identifier.

Table 8 gives an overview of the supported transport modes in the LinkingAlps project. If all modes of a group (e.g. *RailSubmodeEnumeration*) are supported, no individual mode list is given. If only selected modes are supported, a list is given (e.g. *IndividualModesEnumeration*) A complete overview of all transport modes can be found in [1] section 8.4.4.2. The names and definitions of the transport modes follow the Transport Protocol Experts Group (TPEG) standard, which is also used in SIRI.

Table 8 Supported TransportModes.

Group	Supported Modes
IndividualModesEnumeration	walk | cycle | taxi | self-drive-car
PrivateModesEnumeration	carPooling
ContinuousModesEnumeration	walk | demandResponsive | replacementService
TransferModesEnumeration	walk | bikeHire | protectedConnection | guaranteedConnection | remainInVehicle
PtModesEnumeration	All modes are supported.
RailSubmodeEnumeration	All modes are supported.
CoachSubmodeEnumeration	All modes are supported.
MetroSubmodeEnumeration	All modes are supported.
BusSubmodeEnumeration	All modes are supported.
TramSubmodeEnumeration	All modes are supported.
WaterSubmodeEnumeration	All modes are supported.
AirSubmodeEnumeration	This mode is not supported.
TelecabinSubmodeEnumeration	All modes are supported.
FunicularSubmodeEnumeration	All modes are supported.
TaxiSubmodeEnumeration	All modes are supported.

The *individualModes* walk, cycle, taxi and car (self-drive and other-drive) are supported by the implementation of the LinkingAlps OJP profile. Presently, taxi and car are not supported by the LJPs, but their inclusion is being considered as a future development implementation process by some LJPs. The same applies to carpooling. Asking for trips with these modes will result in an empty response. As a general consideration, transport modes not supported by a requesting system may be removed by the receiving system, and a warning (see section 4.2.4) with additional information (e.g. information on unsupported modes) is generated.

When a service offers the option to filter for certain PtModes, the structure of PtModeFilterStructure is used. The information whether a mode is supported is part of the response in the *PlaceResultStructure*[2] (specifically, *ModeStructure*) However, the OJP Standard notes, that this list should only be created upon request[3] (see [1] page 90). In simpler terms:

- Request: Use PtModeFilterStructure to filter/remove for specific PtModes
- Response: Shall only contain PtModes that are requested and whether they are supported or not

4.2.6 Gazetteers

A gazetteer is a directory of common objects across the local journey planner systems and its system borders. It enables the active system to find the passive system for all geolocations (stops, stations, points of interest (POIs), addresses, etc.). The gazetteer acts system-wide. Location identification (unique identifiers, language translations, coordinate systems & geo-locations, modelling of areas in a point representation) shall be harmonised across the systems so that it can be looked up by all distributing services. The gazetteer is contained in the Local Journey Planner (Passive system/OJP Responder) of active and passive systems. The passive system/OJP responder serves as an information source, including information on exchange points, PT Data timetables and the gazetteer. The implementation of the gazetteer repository is responsibility of each active and passive system.

For more information regarding gazetteers within the LinkingAlps context see the Deliverable D.T1.3/4 (Requirements Document).

4.2.7 Expected behaviour regarding optional fields

When describing the LinkingAlps OJP implementation in the following sections the term “optional” is used when the implementation of a parameter or field is not mandatory. This concept has its origin in the OJP standard, where there is a strong separation between fields and parameters that must be supported/implemented in order for the profile to work at all and fields and parameters as well as whole (sub-)structures that may or may not be used in an implementation at all.

It is therefore necessary for the LinkingAlps OJP implementation to make clear which parts of the optional aspects of the standard are required and which are optional. However, as the LinkingAlps system architecture can be viewed from different angles it is important to describe the handling and the implications of the optional LinkingAlps fields and parameters (Table 9). The behaviour described in the table is the agreed compromise for the initial LinkingAlps OJP implementation that may change at later stages.

[2] In case of stop points and stop places.
[3] *PlaceParamStructure.InCludePtModes* (boolean)

Table 9 Expected behaviour for the usage of optional fields.

A to B	Behaviour
AS to PS	The active system does not request optional fields from the passive server. The passive system may ignore unsupported optional fields while completing the request and sends a warning message about the use of unsupported fields.
PS to AS	The passive system will only provide information that was requested. The active system must be able to interpret (process/ignore) the complete response of the passive system, including warning messages. Optional fields are ignored by the active system.
APP	The End-User-Application (App) only requests mandatory (non-optional) fields. The App must be able to interpret warnings and error messages and ignore optional fields.

4.3 OJPLocationInformation

The *OJPLocationInformation* service provides different methods in order to respond with the location to a given (user) request. It uses text matching or GPS coordinates as user input in order to fulfil this task. As an abstraction, the *OJPLocationInformation* service can be used for more complex applications, as “finding the nearest stops/stations for a given coordinate” and “matching text input against the names of locations near a given coordinate” (OJP Description, [1]). The normal use-case of this service is to process a user query into a list of possibly meant locations, which can then be used for feeding other services such as OJPTrip, OJPStopEvents or OJPMultiTrip.

A general description of this service can be found in section 8.5 of the OJP Standard [1]. The XML schema file OJP_Locations.xsd defines data types and structures for use in this service (see Table 1).

A comprehensive overview of the supported fields and parameters for this service can be found in Annex 11.2.

4.3.1 Request

Location information can be gathered by using a ***LocationInformationRequest*** element (type *LocationnformationRequestStructure*). Table 10 gives an overview of the supported request information types for this service. The related OJP Table can be found in [1] (section 8.5.3.1).

The request must contain either:

- *InitialInput*: Name of the location object, which is looked for; GPS Coordinates where to look for locations
- *PlaceRef*: Reference of a Place for which more details are to be retrieved

An empty request for *InitialInput* must be supported. If the request is empty, all available locations of the individual passive system for an additionally specified filter of "Type" must be returned. Only single type returns are allowed. Due to the number of returned locations, pagination may be required and thus the request parameters *NumberOfResults* and *ContinueAt* must be supported.

Table 10 LocationInformationRequestStructure.

		+Structure	Section	Linking Alps (LA)	EU-Spirit (EUS)
a)	InitialInput	+InitialLocationInput	8.5.3.2	Yes	Yes
b)	PlaceRef	+PlaceRef	8.4.5.11	Yes	Yes
Restrictions		+PlaceParam	8.5.3.7	Yes	Yes
Extension				Yes	Yes

The used geo coordinate reference system within the LinkingAlps project is WGS84.

It is possible to set (additional) filter options in order to limit the possible number of results. The following filters (see [1] section 8.5.3.7) are supported within the LinkingAlps context:

- *Type*
- *PtModes (PtMode, Exclude)*

Other filters are optional or not supported (see comprehensive overview of the supported fields and parameters in Annex 11.2). Servers must support all *LocationPolicy* parameters (see [1] section 8.5.3.7).

4.3.2 Response

An element ***PlaceInformationResponse***[4] of the type *PlaceInformationResponseStructure* is used to respond to a location information request. Table 11 gives an overview of the supported response information types for this service. The related OJP Table can be found in [1] (section 8.5.4.1). All location types within the *Place* object (see [1] section 8.4.5) are supported.

Table 11 PlaceInformationResponseStructure.

	+Structure	Section	LA	EUS
ErrorMessage	+ErrorMessage	8.4.4.2	Yes	Yes

[4] Plase note that the response for the LocationInformationRespose is called PlaceInformationResponse in [1]

	+Structure	Section	LA	EUS
ContinueAt	nonNegativeInteger	8.5.3.7	Yes	Yes
Place	+PlaceResult	8.5.4.2	Yes	Yes

Possible error codes that can appear within the context of the response can be found in Table 12 (see [1] section 8.5.4.1).

Table 12 List of possible error codes in LocationInformationResponse.

Code	Description
LOCATION_NORESULTS	No location objects could be found that match the input data.
LOCATION_UNSUPPORTEDTYPE	The requested location types are not supported by the service.
LOCATION_UNSUPPORTEDCOMBINATION	The combination of input data (text string, coordinates, geographical restrictions) cannot be processed by the service.
LOCATION_NOREFINEMENT	The given location object could not be refined.
LOCATION_USAGEIGNORED	The usage type has been ignored.
LOCATION_UNSUPPORTEDPTMODES	The service does not support any restrictions by transport modes.
LOCATION_UNSUPPORTEDLOCALITY	The service does not support any restrictions by localities.

4.4 OJPTrip

The OJPTrip service provides intermodal trip information from an origin location to a destination taking various user preferences into account. In distributed environments, the complete trip is not calculated within one single system, instead the planning task is split and distributed to several planning engines. This service calculates trips within the in LinkingAlps are) between one place of origin and one destination place. Both locations must be a result of an OPJLocationInformation request (see section 4.3).

A comprehensive overview of the supported fields and parameters for this service can be found in Annex 11.2.

4.4.1 Request

Intermodal trip information can be gathered by using a ***TripRequest*** element (type *TripRequestStructure*). Table 13 gives an overview of the supported request information types for this service. The related OJP Table can be found in [1] (section 8.7.3.1).

Table 13 TripRequestStructure.

	+Structure	**Section**	**LA**	**EUS**
Origin	+PlaceContext	8.4.5	Yes	Yes
Destination	+PlaceContext	8.4.5	Yes	Yes
Via	+Via	8.4.6.2	Optional	Partial[5]
NotVia	+NotVia	8.7.3.6	No	Partial
NoChangeAt	+NoChangeAt	8.7.3.7	No	Partial
Params	+tripParam	8.7.3.3	Yes	Yes

The *Via* option is supported by this OJP profile. The support of the parameter *timeAllowance* (part of *Origin* and *Destination*) is optional. The Parameter *dwellTime* in *Via* is not supported.

All filters and policies of the element *Params* are supported unless the related service is not supported (e.g. Fare). For a more detailed list of the supported elements see Annex 11.2. For accessibility reasons all parameters in the Subgroup *BaseTripMobilityFilter* must be supported.

As a filter, the fastest connection and fewest transfers (comfort) are supported (Subgroup *TripPolicy* of the element *Params*). Available options (see [1] section 8.7.3.3) are:

1. fastest
2. minChanges
3. leastWalking
4. earliestArrvial
5. latestDeparture
6. earliestArrivalAndLatestDeparture

In addition, the maxima for walking and cycling distances as well as the maximum number of transfers may be specified by using *IndividualTransportOptions* as part of the *PlaceContextStructure* (see [1] section 8.4.3.2). The need to transport bicycles is supported by the “BikeTransport” Boolean parameter (part of element *Params*).

[5] Via, NotVia and NoChangeAt are not supported by the Ring Connection Composer (RCC). However, it may be supported (optional) on the passive systems.

- PtModeFilter.PtMode and PtModeFilter.Exclude must be supported.
- The policy NumberOfResults and NumberOfResultsBefore/NumberOfResultsAfter must be supported. Other policies are optional to support.
- The content filters IncludeLegs and IncludeIntermediateStops must be supported.
- IndividualTransportOptions must be supported.

4.4.2 Response

An element **TripResponse** of the type *TripResponseStructure* is used to respond to an intermodal trip request. Table 14 gives an overview of the supported response information types for this service. The related OJP Table can be found in [1] (section 8.7.4.1).

A complete TripResponseContext needs to be sent as response. *TripResult* (as Substructure of *TripResponseStructure*) contains no more than the maximum number of results, as defined by the request but all referenced places and situations. A detailed description can be found in [1].

Table 14 TripResponseStructure.

	+Structure	Section	LA	EUS
ErrorMessage	+ErrorMessage	8.4.4.2	Yes	Yes
TripResponseContext	+TripResponseContext	8.7.4.3	Yes	Yes
TripResult	+TripResult	8.7.4.4	Yes	Yes

It is not required to support the policy AcceptDeferredDelivery[6]. Therefore, passive servers will never be asked to return TripSummary instead of Trip within the TripResultStructure.

Possible error codes that can appear within the context of the response can be found in Table 15 (see [1] section 8.7.4.1).

Table 15 List of possible error codes in TripResponse.

Code	Description
TRIP_NOTRIPFOUND	No trip plan could be found that meets all the parameters as they have been set by the user (start and end locations, departure/arrival time and further options possibly set by the user).
TRIP_ORIGINUNKNOWN	The start location (address, stop place, …) for the requested trip is unknown.

[6] Within EU-Spirit this parameter is optional for the passive system and not supported by the active system. The latter will just pass the parameter along to the passive system.

Code	Description
TRIP_DESTINATIONUNKNOWN	The end location (address, stop place, …) for the requested trip is unknown.
TRIP_VIAUNKNOWN	One of the via points is unknown.
TRIP_NOTVIAUNKNOWN	One of the not-via points is unknown.
TRIP_NOORIGIN	No start location has been defined for the trip.
TRIP_NODESTINATION	No end location has been defined for the trip.
TRIP_ORIGINDESTINATIONIDENTICAL	Start and end of the trip are identical.
TRIP_DATETIMEERROR	The requested date and/or time do not make sense.
TRIP_DEPARTUREAFTERARRIVAL	The requested departure time at each origin locations is after the requested arrival time at any destination location.
TRIP_DATEOUTOFRANGE	There is no timetable data available for the requested date.

Additionally, error messages may appear within the context of the TripResultStructure. They can be found in Table 16 (see [1] section 8.7.4.4).

Table 16 List of possible error codes in TripResult.

Code	Description
TRIP_ORIGINEQUIVALENT	The requested origin stop place has been replaced by an equivalent stop place.
TRIP_DESTINATIONEQUIVALENT	The requested destination stop place has been replaced by an equivalent stop place.
TRIP_VIAEQUIVALENT	One of the requested via stop places has been replaced by an equivalent stop place.
TRIP_REALTIMEINCOMPLETE	There is no realtime information available for at least one of the services within this trip result.
TRIP_ITTIMEEXTENDED	The maximum time allowed for using modes of individual transport (mostly walking or cycling) has been extended by the system because otherwise no trip could be found.
TRIP_ITMODECHANGED	The mode of individual transport specified by the user has been replaced by the system because otherwise no trip could be found. Usually this means taking a taxi instead of walking.
TRIP_INCONVENIENTWAITING	The trip plan result contains a long waiting time.

4.5 OJPStopEvent

This service provides information on arrivals and/or departures of public transport services from stops for a requested time or period of time. Restrictions can be set in the request parameters that filter the result contents accordingly. Place needs to be a result of a *LocationInformationRequest.*

A comprehensive overview of the supported fields and parameters for this service can be found in Annex 11.2.

4.5.1 Request

Stop event information can be requested by sending a ***StopEventRequest*** element (of type *StopEventRequestStructure*). Table 17 gives an overview of the supported request information types for this service. The related OJP Table can be found in [1] (section 8.8.2.1).

A request for a timetable (departure or arrival boards) of a station is supported by sending a *StopEvent* request with an empty *StopEventDataFilter*. It is possible to get an operator, public transport mode or line specific timetable at a stop by using the matching filter option of *StopEventDataFilter*.

Table 17 StopEventRequestStructure.

	+Structure	**Section**	**LA**	**EUS**
Place	+PlaceContext	8.4.5	Yes	Yes
Params	+StopEventParam	8.8.2.2	Yes	Yes

All filters and policies are supported. *TimeAllowance* is not needed for this service.

The exchange of real-time data for a specific station is supported by this profile. If the filtering option "includeRealtimeData" (part of the group *StopEventContentFilter*) is set to true, the real-time data will be included in the response. If a local journey planner does not support real-time data, it sends a warning (response).

4.5.2 Response

An element **StopEventResponse** of the type *StopEventResponseStructure* is used to respond to a stop events request. Table 18 gives an overview of the supported response information types for this service. The related OJP Table can be found in [1] (section 8.8.3.1).

A complete StopEventsResponseContext needs to be sent as response. It contains not more than the maximal number of results, defined by the request but all referenced places and situations.

Table 18 StopEventResponseStructure.

	+Structure	Section	LA	EUS
ErrorMessage	+ErrorMessage	8.4.4.2	Yes	Yes
StopEventResponseContext	+StopEventResponseContext	8.8.3.2	Yes	Yes
StopEventResult	+StopEventResult	8.8.3.3	Yes	Yes

TimeAllowance is not needed for this service.

Possible error codes that can appear within the context of the response can be found in Table 19 (see [1] section 8.9.3.1).

Table 19 List of possible error codes in StopEventResponse.

Code	Description
STOPEVENT_DATEOUTOFRANGE	There are no timetables available for the requested date.
STOPEVENT_LOCATIONUNKNOWN	The location (address, stop etc.) for which stop events have been requested is unknown.
STOPEVENT_LOCATIONUNSERVED	There is no public transport service available at all at the locations (address, stop etc.) for which stop events have been requested.
STOPEVENT_NOEVENTFOUND	No departure/arrival could be found within the requested period of time that meets the given restrictions.

4.6 OJPTripInfo

The *OJPTripInfo* service "provides intermodal trip information from an origin location to a destination taking various user preferences into account" (OJP Description, see [1]). As trip information may change over time the request should be done after requests for the services of OJPTrip (section 4.4) and OJPLocationInformation (section 4.3).

A comprehensive overview of the supported fields and parameters for this service can be found in Annex 11.2.

4.6.1 Request

Trip information can be gathered by using a **TripInfoRequest** element (type *TripInfoRequestStructure*). Table 20 gives an overview of the supported request information types for this service. The related OJP Table can be found in [1] (section 8.9.2.1).

Table 20 TripInfoRequestStructure.

		+Structure	Section	LA	EUS
a)	JourneyRef	→ Journey	8.4.4.1	Yes	Yes
a)	OperatingDayRef	→ OperatingDay	8.4.4.1	Yes	Yes
b)	VehicleRef	→ siri:Vehicle	8.4.4.1	Optional	No
b)	TimeOfOperation	dateTime		Optional	No
Params		+TripInfoParams	8.9.2.2	Yes	Yes

JourneyRef and OperatingDayRef are of group DatedJourneyRef, VehicleRef and TimeOfOperation are of group TimedVehicleRef.

All filters and policies for *TripInfoRequest*, as described in the OJP Description (see [1]), must be supported.

4.6.2 Response

An element ***TripInfoResponse*** of the type *TripInfoResponseStructure* is used to respond to a trip information request. Table 21 gives an overview of the supported response information types for this service. The related OJP Table can be found in [1] (section 8.9.3.1).

Table 21 TripInfoResponseStructure.

	+Structure	Section	LA	EUS
ErrorMessage	+ErrorMessage	8.4.4.2	Yes	Yes
TripInfoResponseContext	+TripInfoResponseContext	8.9.3.2	Yes	Yes
TripInfoResult	+TripInfoResult	8.9.3.3	Yes	

TripInfoResponse and TripInfoResult are part of the group TripInfoResponse.

A complete *TripInfoResponseContext* must be returned within the response, containing all referenced places and situations.

Possible error codes that can appear within the context of the response can be found in Table 22 (see [1] section 8.9.3.1).

Table 22 List of possible error codes in TripInfoResponse.

Code	Description
TRIPINFO_JOURNEYREFUNKNOWN	The journey reference used in the request is unknown.

Code	Description
TRIPINFO_VEHICLEUNKNOWN	The vehicle reference used in the request is unknown.
TRIPINFO_NOJOURNEYFOUND	No matching journey could be found for the requested time and journey/vehicle. identifiers.
TRIPINFO_NOGEOINFO	No geographic information available for this vehicle journey.

4.7 OJPExchangePoint

Distributed journey planning requires several journey planning systems planning parts of the whole trip which must be assembled. Each of the planners will therefore get a sub-query to plan: the first planner from the origin of the trip to its system boundaries, the next planner must find trips from these boundaries to its boundaries with the next systems. The service *OJPExchangePoint* provides the exchange points. Further information can be found in section 3.2 of this document.

A comprehensive overview of the supported fields and parameters for this service can be found in Annex 11.2.

4.7.1 Request

Exchange points can be gathered by using an ***ExchangePointsRequest*** element (type *ExchangePointsRequestStructure*). Table 23 gives an overview of the supported request information types for this service. The related OJP Table can be found in [1] (section 8.6.2.1).

An empty request for *PlaceRef* must be supported. If the request is empty, all available exchange points, which match the given data filter options (*ExchangePointsParam*), are returned. With a given PlaceRef, sensible exchange points which match the given data filter options, are returned. Sensible exchange points allow trips from/to the given place.

Table 23 ExchangePointsRequestStructure.

		+Structure	**Section**	**LA**	**EUS**
	PlaceRef	+PlaceRef	8.4.5.11	Yes	Yes
	Params	+ExchangePointsParam	8.6.2.2	Yes	Yes
Extension				Yes	Yes

Table 24 gives an overview of the supported elements in ExchangePointsParamStructure.

Table 24 ExchangePointsParamStructure.

		+Structure	Section	LA	EUS
	Type	stop		Yes	Yes
	Usage	origin | destination | via		Yes	Yes
	PtModes	+PtModeFilter	8.4.3.5	Yes	Yes
	OperatorFilter	+OperatorFilter	8.2.4	Optional	
	TopographicPlaceRef	→ TopographicPlaceCode	8.4.5.1	NO	
	DestinationSystem	siri:ParticipantRef	8.4.4.1	Yes	Yes
	AdjacentSystem	siri:ParticipantRef	8.4.4.1	Yes	Yes
	Language	xs:language		Yes	Yes
	NumberOfResults	xs:positiveInteger		Yes	Yes
	ContinueAt	xs:nonNegativeInteger	8.6.3.1	Yes	yes

Only the type "Stop" is supported for exchange points. The optional parameter "Usage" defines whether a place is an origin, destination or via. Maximal number of results is supported for the request.

It is required to support the following filters/policies for the request and therefore on the response as well:

- PtModes (PtMode, pTModeExclude)
- DestinationSystem
- AdjacentSystem

Due to the number of returned exchange points, pagination may be required and thus the request parameters *NumberOfResults* and *ContinueAt* must be supported. The policy parameter *Language* must be supported as well, resulting in the support of all policy filters for the request.

4.7.2 Response

An element ***ExchangePointsResponse*** of the type *ExchangePointsResponseStructure* is used to respond to a location information request. Table 25 gives an overview of the supported response information types for this service. The related OJP Table can be found in [1] (section 8.6.3.1).

Table 25 ExchangePointsResponseStructure.

	+Structure	Section	LA	EUS
ErrorMessage	+ErrorMessage	8.4.4.2	Yes	Yes
ContinueAt	nonNegativeInterger	8.5.3.7	Yes	Yes
Place	+ExchangePointsResult	8.6.3.2	Yes	Yes

Within the response it is required to fill *TravelDurationEstimate* and *BorderPoint* with appropriate values for all returned exchange points (part of *ExchangePointsResultsStructure*), especially when the filter options for AdjacentSystem are set. *TravelDurationEstimate* shall contain a rough travel time estimation from origin to exchange point (or from exchange point to destination respectively) and is only returned when the PlaceRef element is used in the request. *BorderPoint* tells whether an exchange point is logically (not geographically) located on the border between two regional systems. A list of available transport modes must be created upon request (see [1] section 8.6.3.2).

All returned exchange points must be either of type *StopPlace* or *StopPoint* (part of Place structure in *ExchangePointsResultsStructure*). If a server has exchange points defined on the stop point level it must return those stop points, but also the corresponding stop places.

Possible error codes that can appear within the context of the response can be found in Table 26 (see [1] section 8.6.3.1).

Table 26 List of possible error codes in ExchangePointsResponse.

Code	Description
EXCHANGEPOINTS_NORESULTS	No exchange points could be found that match the query criteria.
EXCHANGEPOINTS _UNKNOWNDESTINATION	The destination system given in the request parameters is unknown.
EXCHANGEPOINTS _UNKNOWNADJACENTSYSTEM	One or more of the adjacent systems given in the request parameters are unknown.

4.8 OJPMultiPointTrip

This service provides intermodal trip information from multiple origin locations to multiple destinations taking various user preferences into account. The structure is similar to the normal trip request/response (see section 4.4). Locations must be a result of an OPJLocationInformation request (see section 4.3).

A comprehensive overview of the supported fields and parameters for this service can be found in Annex 11.2.

4.8.1 Request

Intermodal trip information can be gathered by using a MultiPointTripRequest element (type *MultiPointTripRequestStructure*). Table 27 gives an overview of the supported request information types for this service. The related OJP Table can be found in [1] (section 8.7.3.2).

Table 27 MultiPointTripRequestStructure.

	+Structure	**Section**	**LA**	**EUS**
Origin	+PlaceContext	8.4.5	Yes	Yes
Destination	+PlaceContext	8.4.5	Yes	Yes
Via	+Via	8.4.6.2	Optional	Partial[7]
NotVia	+NotVia	8.7.3.6	No	Partial
NoChangeAt	+NoChangeAt	8.7.3.7	No	Partial
Params	+MultiPointTripParam	8.7.3.3	Yes	Yes

The *Via* option is supported, *NotVia* and *NoChangeAt* are not supported by this service.

There can be either one single *Origin* (or *Destination*), which contains a *PlaceRef* to a regular location (not an exchange point), or a set of origins (or destinations), each containing a *PlaceRef*, where all the referred places are exchange points. This means that at least one location (Origin or Destination) must be an exchange point, unless origin or destination are in the same region of one single sub system. Other combinations are not allowed.

With a set of given exchange points as *Origin* (or *Destination*), either all origins (destinations) contain a *DepArrTime*, or all origins (destinations) contain a *TimeAllowance*, or all contain none of both. Other combinations (in the sense of some containing *DepArrTime* and some *TimeAllowance*) are not allowed.

The same options and features as for “single” trip requests (see section 4.4.1) must be supported, as MultiPoint can be seen as an extension.

4.8.2 Response

An element ***MultiPointTripResponse*** of the type *MultiPointTripResponseStructure* is used to respond to an intermodal multi point trip request. Table 28 gives an overview of the supported

[7] Via, NotVia and NoChangeAt are not supported by the RCC. However, it may be supported (optional) on the passive systems.

response information types for this service. The related OJP Table can be found in [1] (section 8.7.4.2).

As with the rest, the same rules of OJPTrip response (section 4.4.2) apply here as well.

Table 28 MultiPointTripResponseStructure.

	+Structure	Section	LA	EUS
ErrorMessage	+ErrorMessage	8.4.4.2	Yes	Yes
TripResponseContext	+TripResponseContext	8.7.4.3	Yes	Yes
MultiPointTripResult	+MultiPointTripResult	8.7.4.5	Yes	Yes

Possible error codes that can appear within the context of the response can be found in Table 29 (see [1] section 8.7.4.2).

Table 29 List of possible error codes in MultiPointTripResponse.

Code	Description
MULTIPOINTTRIP_NOTALLPOINTSCOVERED	In case a multi-point request with MultiPointType set to eachDestination could not be responded to with a trip solution to each of the destination points. And respectively in case a multi-point request with MultiPointType set to eachOrigin could not be responded to with a trip solution for each of the origin points.
MULTIPOINTTRIP_TOOMANYPOINTS	In case a multi-point request uses too many points as departure or arrival.

5 Extensions of OJP Standard

This chapter describes discussed future OJPExtensions within the LinkingAlps context.

5.1 Environmental Footprint

Due to environment protection and CO2 reduction concerns, it becomes necessary to inform the passengers about the possible environmental impact of their journey. This information is demanded by governmental regulations and ecologically aware passengers.

This requires messages with information about the pollutant produced and the energy supply used. In addition to CO2 information, it should also be possible to transmit information on other environmentally harmful pollutants. Energy supply means all fuels and power connections for the vehicles used. The source of the electricity supply (green electricity, conventional supply) should also be considered. For the *OJPFares* Service (not implemented in LinkingAlps) the possibility of CO2 compensation payments should exist (extension within the related service).

5.1.1 Implementation in the LinkingAlps project

The OJP Standard currently does not support an option to give the end-user any kind of information related to the environmental footprint as part of the journey planning process. As a first step towards supporting ecological aspects the amount of the theoretical CO2 pollution/emission is supported as part of an extension. As the amount of CO2 emitted depends directly on the used mode of transportation and in a larger context the energy source, any kind calculation/algorithm should be based on the aggregation of the individual trip legs respective usage of all used modes of transportation (e.g., train A + train B + taxi). Therefore, the most sensible place for this extension is the OJPTripResponseStructure (see section 4.4.2) in the sub-groups of TimedLeg, TransferLeg and ContinuousLeg. However, it should be noted that a theoretical value, based on long-term planning, might be different from the actual real value/environmental impact (e.g., replacement of trains due to unscheduled maintenance).

Regarding the provided fields/parameters, the information described in Table 30, should initially be provided by the LinkingAlps service. In any case, it should be avoided to provide redundant values (e.g., emitted CO2 per 100 km) which can easily be calculated with the already provided information. The definition of an algorithm or method to provide these values is not part of this profile.

Table 30 Environmental Footprint Extension.

Grouping	Element name	Min: Max	File type	Description
Extension	Emission.CO2Value	0:1	xs:positiveInteger	CO2 value in gram
Extension	Emission.CO2ConfidenceLevel	0:1	xs:decimal	Confidence level

The payment of a compensation fee is not implemented in the LinkingAlps project, because the Service *OJPFare* is not part of this profile and due to the point raised above.

A comprehensive overview of the suggested extension and their placement within the OJP-profile can be found in Annex 11.2.

6 Use-cases

Use cases for the LinkingAlps project are defined in the Deliverable D.T1.2.1.(Use Case Definitions) of the LinkingAlps project.

7 Compatibility with other OJP implementations

In order to exchange data with other OJP projects it is important that the main features and services are supported across all implementations. Therefore, a compatibility marker/notice to EU-Spirit has been given throughout all the services described in this document.

In terms of the previously described (possible) extensions, it should be noted that unless those are adopted or developed across multiple projects the support for these is not existent. For this reason, it seems logical that any extension should remain optional for the sake of compatibility at least until the use of the extension has become common within the OJP community.

7.1 Compatibility with LinkingDanube

The LinkingDanube project (2017-2019)[9] deals with the "linking of services" for transnational, multimodal traveller information and journey planning. The LinkingDanube project can be seen as "proof-of-concept" for the demonstration of linking services via the open journey planning standard, connecting different isolated services without any physical integration into one central database. Operators of traveller information services keep the sovereignty over their data, as only routing results are provided upon request via this interface. One central node, including the necessary logic to manage the requests and assemble the route, enables the communication within this distributed system. As of its proof-of-concept role, the available OJP services are limited, focussing on OJPLocationInformation and OJPTrip.

Providers from the following countries offer the LinkingDanube service:

- Austria
- Czech Republic
- Croatia
- Hungary
- Moldova
- Romania
- Slovakia
- Slovenia

7.2 Compatibility to EU-Spirit

EU-Spirit[10] is a cross-border travel information service for users of public transport systems. It is based on existing local, regional, and national travel information systems which are interlinked via technical interfaces.

[9] LinkingDanube Homepage: https://linkingdanube.eu/
[10] EU-Spirit Homepage: https://eu-spirit.eu/

The EU-Spirit service provides door-to-door travel information for customers who do not only travel within one region. The service provides the calculation of an itinerary between stops, addresses or points of interest in different European regions. The information service includes any carrier of local public transport and long-distance rail and flight services, as well as additional services like map service and fare information. The information about the EU-Spirit service providers is free and provided via the customer local information system in his/hers mother tongue.

Providers from the following countries offer the EU-Spirit service:

- Denmark
- France
- Germany
- Luxembourg
- Poland
- Sweden

EU-Spirit is currently switching to OJP. It needs to be ensured that the passive servers of EU-Spirit are able to communicate with the active systems of LinkingAlps, and that the passive systems of LinkingAlps are able to communicate with the active server of EU-Spirit. Throughout this document, the compatibility between services, fields and parameters has been noted.

7.2.1 Cross communication between LinkingAlps and EU-Spirit

Figure 3 displays a simplified system architecture for both projects, EU-Spirit (left) and LinkingAlps (right). The EU-Spirit components and their basic architecture can be described like the following:

- Active Server (AS): Component which acts as the backend for the UI and uses the services of the RCC (location identification, journey planning, stop events, service information).
- Ring Connection Composer (RCC): Component, which is asked by the active servers, in order to fulfil queries of the active server. It uses the services of the passive servers, i.e., distributing the queries and composing the corresponding responses.
- Passive Server (PS): Component, which is queried by the RCC, in order to fulfil partial tasks (location identification, partial journey planning, stop events, service information).

For a description of the LinkingAlps architecture see chapter 3 of this document.

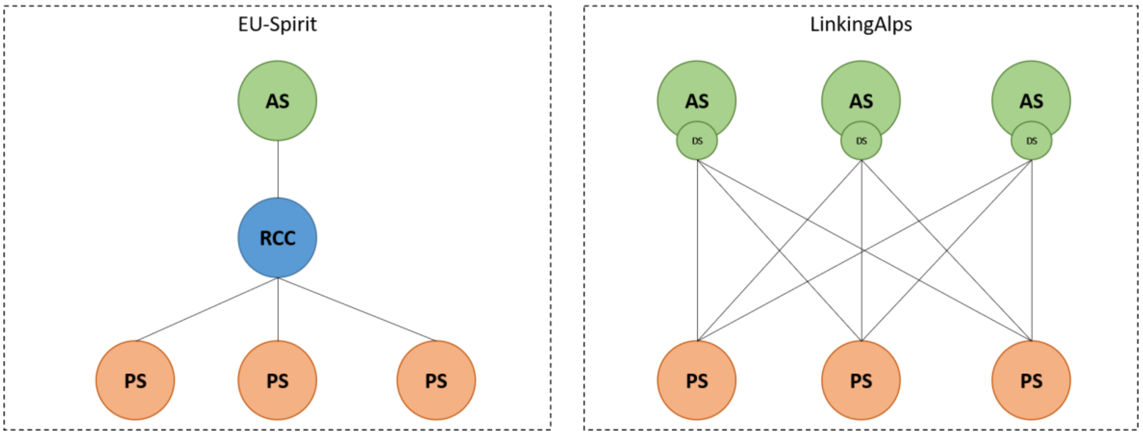

Figure 3 EU-Spirit and LinkingAlps basic system architecture (simplified).

The simplest way of allowing cross communication between the two systems (in terms of technical practicability[11]) is displayed in Figure 4, where the EU-Spirit RCC is connecting to each PS of LinkingAlps individually (request/response) and each LinkingAlps AS is doing the same with the PS of EU-Spirit, bypassing the RCC altogether. With the exchange point IDs both defined in the private section (see section 4.2.1), the identification should, at least in theory, be straightforward. Stations with identifiers for both projects and other projects can therefore be easily identified.

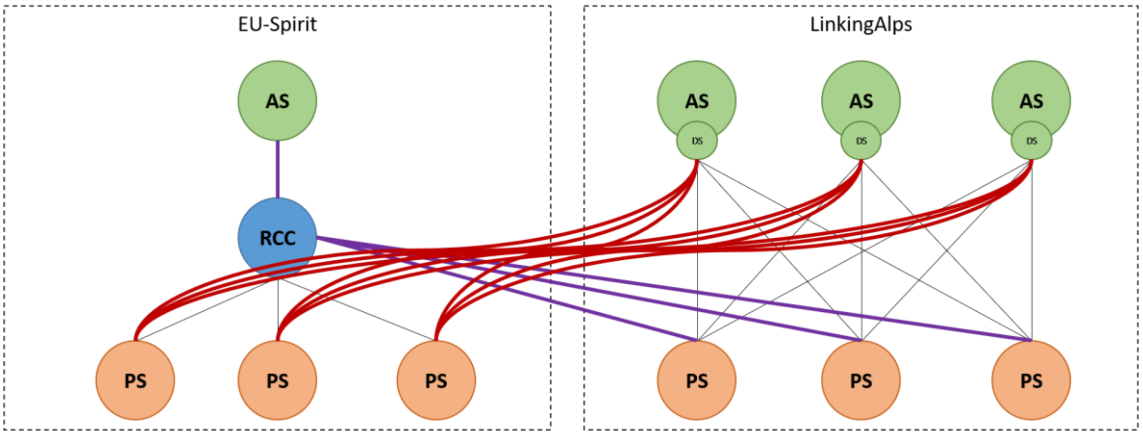

Figure 4 EU-Spirit and LinkingAlps cross communication (simplified).

[11] Political or contractual implications / requirements are not covered in this document and need to be looked at separately.

8 OJP Standard

This section describes the wishes and plans of the LinkingAlps project regarding the overall OJP Standard development as well as its implementation.

8.1 Future OJP Versions developments

- Clarification on Error-Messages: Can the Status be true while setting an ErrorCondition in order to define a warning?[12] Please note: The combination of status and error (see section 4.2.4) is the suggested method and is very likely to be the expected behaviour in the next OJP version.
- Clear upgrade path and outlook regarding compatibility changes between versions as well as intended support timeframes.

8.2 Compatibility with future OJP Versions

The initial OJP version used in LinkingAlps is V1.0. However, it must be assumed that at some point migration to a later version may become necessary. For instance, V1.0 of the standard does not include a sensible method to handle multi-language requirements in the sense of providing the end-user with more than one language to choose from at a time (see section 4.2.2).

It is therefore important to define a clear migration pattern and process to implement and provide new features/information within and outside the LinkingAlps system architecture while at the same time avoiding unnecessary additional implementation costs. The migration process itself is described in WP T3 of the LinkingAlps project.

9 License model

Not yet defined. Will be dealt with in the context of WP T4 of the LinkingAlps project.

10 Bibliography

[1] European Committee for Standardization, „CEN/TC 278 - Intelligent transport systems," 2017. [Online]. Available: http://www.normes-donnees-tc.org/wp-content/uploads/2017/01/TC_278_WI_00278420_E-RS-170118-final3.pdf.

[12] See https://github.com/VDVde/OJP/issues/93 (Last time checked: 2020-08-13),

[2] CEN/TC278/WG3 SG9, „NeTEx - Network Timetable Exchange,“ [Online]. Available: http://netex-cen.eu/. [Zugriff am 10 08 2020].

11 Annex

11.1 System IDs

The System-ID names used within the LinkingAlps OJP profile are described in Table 31. The general idea is to distinguish between different service providers and development phases.

Table 31 System-ID names.

System	Production	Test	Integration
ARIA	RLA-MOV_prod	RLA-MOV_test	RLA-MOV_int (integration system currently is not open)
CMTo/5T	CMTO-5T_prod	CMTO-5T_test	CMTO-5T_int
LUR			
SBB	SBB-SKI_prod	SBB-SKI_test	SBB-SKI_int
STA	STA_prod	STA_test	STA_int
VAO			

11.2 Supported Services, Fields & Parameters (LinkingAlps Profile)

See external document "LinkingAlps_OJP_profile_implementationOverview_v03.00.xlsx" for a comprehensive list of the supported services, fields and parameters.