Hogia
Public Transport Systems

Data Output Interface
Interface Specification

Document identity: IS-PT/I/DOI/4
Revision: M
Date: 2014-04-23

Data Output Interface
Interface Specification

This document is a property of Hogia Public Transport Systems AB. The definition of the interface specification defined in this document is a property of Hogia Public Transport Systems AB.
The design of the Hogia PubTrans system is a property of Hogia Public Transport Systems AB.

Copyright

Revision History

Revision	Date	Update	Updated by
A	2004-12-20		Ulf Bjersing
B	2005-05-10	Changes according to SL IS-PT/I/DOI/4 Rev. A	Ulf Bjersing
C	2005-10-06	Changes according to IS-PUBTUG_I_DOI_CORR_CR_2	Ulf Bjersing
D	2005-11-22	Minor changes in accordance with DII rev D changes. New document identity.	Ulf Bjersing
E	2006-02-06	Service Requirement Pattern pointer moved from Named Journey Pattern to Vehicle Journey Template.	Ulf Bjersing
F	2007-03-23	VehicleJourneyNote added.	Ulf Bjersing
G	2009-11-04	DescriptionNote added to DirectionOfLine. Coordinates added to Zone, enumeration added. Correction of text for view RouteLink.	Ulf Bjersing
H	2010-01-14	Minor editorial correction.	Ulf Bjersing
J	2012-02-01	Change history corrected. No actual interface change.	Ulf Bjersing
K	2012-11-12	A number of changes in accordance with DII rev L.	Ulf Bjersing
L	2013-02-25	Removed chapter *Persons and their roles*	Ulf Bjersing
M	2014-04-23	Restored GID-chapter	Ulf Bjersing

Revision	Date	Update	Updated by

Interface Version History

A leading P in the document revision indicates a version under preparation. A – (minus) indicates that the document was not part of the version of the interface. A / (slash) indicates that the document is no longer used in the version of the interface.

Document/product identity	3.0	4.0
PubTrans - Data Output Interface – 3.0.3	3.0	/
IS-PT/I/DOI/4	-	M

Detailed change history from Revision F to G

Change
Column *DescriptionNote* added to view DirectionOfLine. **Background**: Addition
CentroidCoordinates added to Zone. **Background**: Addition
Enumeration FAREREFP added to *typZoneTypeCode.* **Background**: Addition
New data type *typDescription* nvarchar(255) added. **Background**: Addition
Correction of wording in view RouteLink **Background:** Correction
Minor editorial corrections: Unused data types *typJourneyNumber* and *typText* removed. Comment adjusted for column *TypeCode* in view Stop Point. **Background**: Correction

Detailed change history from Revision G to H

Change
Column *TranslatedValueName* (*typName*) replaced by *TranslatedValueDescription* (*typDecription*) in view Data Type Enumeration. **Background**: Correction

Detailed change history from Revision H to J

Change
Removed incorrect information in change history Revision F to G concerning changed JourneyPatternPoint. **Background**: Correction

Detailed change history from Revision J to K

Change
Key definition views (*Value Type, Variant Type, Key Type, Key Variant Type, Object Type, Publication and Publication For Object Type*) added to *System information* publication. *Key Variant Value* views added to each publication. *Task In Block* added to *Vehicle Schedule* publication. Publication *Vehicle Journey Definition* with views *Vehicle Journey Definition* and *Definition For Vehicle Journey* added. Addition of datatypes: *typVehicleJourneyTypeCode*, *typRegExString*, *typBasedataType*, *typStringKeyValue and typTime*. **Background**: Addition

Detailed change history from Revision K to L

Change
Removed publication *Persons and their roles* including views *Person* and *Employment* **Background**: Correction

Detailed change history from Revision L to M

Change
GID-chapter is reintroduced after being left out from revision K and L. **Background**: Correction

Table of content

1 Introduction

The purpose of this document is to describe the Data Output Interface (DOI) of the PubTrans database.

With DOI it is possible to let other applications have local access to data in PubTrans as if it was a direct access to the main database system.

1.1 Overview

DOI contains a view of the planned public transport operation. DOI does not contain real-time data or short-term changes. That data is available through other PubTrans interfaces.

DOI is based on database replication using a metaphor of *publications* and *subscriptions*. Data is published in a set of publications, each containing a subset of data. The publications are standardised to meet general requirements of data access to PubTrans. Customised publications should generally not be required.

Applications may subscribe to the publications. An application chooses to subscribe to the publications that contain data structures needed for that application. Due to the standardised content of the publications, an application must be able to refine the selection of data in the subscribed data.

Data in publications are modelled using an object-oriented data model based on TRANSMODEL. Each publication contains a subset of related classes. Each class is represented as a relational table, and each row represents an instance of a class, i.e. an object. A column represents either an attribute or a reference to another object or objects.

Date-expanded DOI data is available before 0:00 the day of operation, however, changes are replicated more or less frequently depending of the setup of PubTrans data distribution configuration.

All data in DOI is read-only. No change to data in the subscribed copy in the local database is reflected in PubTrans.

2 Implementation

DOI is based on database replication, which means that a number of tables in a source database are mirrored in one or several target databases.

2.1 Data model

DOI has an object-oriented data model based on PubTrans Model and the European reference data model TRANSMODEL (www.transmodel.org). The design of DOI is intended to be as neutral as possible regarding the client systems' usage of the replicated data. The design has also considered that Vehicle information systems have different capabilities and capacity.

2.1.1 Naming in general

The names of the replicated tables and columns are based on the terminology in TRANSMODEL and PubTrans Model.

2.1.2 Tables and Views

Tables are named *doi4_* followed by the class name, i.e. what a single row represents. Views are named with the class name only. If the class is derived from another class, the name of the super class is written after the name of the view separated by a colon.

Names in [brackets] denote that the super class represents an abstract class and that it is not represented as a table or view in any publication.[1]

2.1.3 Columns

A column value represents either a descriptive value or a foreign key.

Each column is associated with a user defined data type, named on the form typXyz. A user defined data type defines the SQL Server data type, an optional default value and value constraint. They are common for all tables in DOI and are described in chapter *20 Data types*.

2.1.4 Foreign keys

Foreign keys – or object references using an object-oriented terminology – is always a single column value. This makes joining of tables much easier. However, a join is often supposed to also consider the time aspect, i.e. consider the period of existence/validity of joined objects.

The naming of the columns is self-descriptive, with the name of the type of reference and the referred interface view.

[1] Class names made up of several words are described with the words separated by blanks in this document, however when used in actual queries the blanks in the name must be removed.

Foreign keys are of two types:

- A unique identity of the row, not only in the table, but also across all tables[2] in all installations of PubTrans databases. This type is a 16-digit number, and is denoted by typId. A foreign key with a typId-value may be NULL. If not NULL, it is guaranteed to have referential integrity.
- A global identity, which has the same value for the same object across all installations of PubTrans databases. This type is denoted by typGid and the value of this type is normally constructed by concatenating numeric attributes to a 16-digit number. There are also so called "abstract" Gids where the 16-digit number does not contain any special meaning. The typGid-value is guaranteed to be unique at a specific point in time, but may also Point to objects that do not exist at a specific point in time.

Additionally, absolute date and time values are unique per se[3]. There exist two types of values: **typDate** and **typDateTime**.

Please note that no other references than **typId** and **typGid** are guaranteed to be unique. Attributes like Line-numbers, Organisation names and so on may not be used as keys, as they are not guaranteed to be unique and may change over time. For further descriptions of Gid's, see the appendixes.

2.1.5 Column names

Columns have self-descriptive names. The name ends with the name of the data type (without the data type's leading "typ"). The data type is named in a non-implementation specific fashion. Behind this data type name, PubTrans defines how values of the data type should be handled on input. The data types are described in a separate chapter at the end of this document. When column names are referred to in general text in this document they are presented in *italics*.

2.1.6 Column names – foreign keys

Columns that hold foreign keys have names that start with "Is", "Has", "Uses", "StartsAt" or similar to express that they express a relationship to the referred view. The name of the referred view is also part of the column name. It can be found directly in front of the data type "Id" or "Gid". I.e. the column name *IsManagedByOrganisationalUnitId* is built up of three parts; "IsManagedBy", that express the relationship; "OrganisationalUnit", that is the name of the referred view; and "Id" that denotes that it is the column *Id* in the referred view that should be matched.

2.1.7 NULL-values

Columns where NULL-values are permitted will be noted as "May be NULL." in the remark of each view definition.

---

[2] An exception to this is that tables or views that expose a class that is derived from another class will show the same Id for an object that is used in the view or table of the base class.

[3] A date or time is identified by its value, e.g. the object "Day" has no *Id* column, since the date itself is a unique key value.

2.2 Versioning

PubTrans DOI exposes three models for versioning:

- Updateable objects, where attribute changes are updated without creating a new instance and where an object may exist infinite.
- Period state objects, where instances are created successive in time and where only the latest version may exist infinite.
- Dated state objects, where each instance has a list of associated dates where it exists.

Changes in DOI data can be detected differently depending on the versioning model.

2.2.1 Updateable objects

Data in updateable objects is stored in one single row, which is modified during object lifetime. Only columns marked as "May be updated" can be changed during object lifetime, all other columns remain constant.

To detect a change in an updateable object, the column *LastModifiedUtcDateTime* should be used to find what objects that have changed since a specific point in time. Initially this column is set to the creation date/time of the object.

It is not possible to see if and which of the columns that have been changed, unless the user have the previous version to compare with.

Updateable objects usually exist infinite. This is expressed as having a NULL-value in the column *ExistsUptoDate*. If *ExistsUptoDate* is set it means that the object's lifetime ceases at this point in time. Note that the column *ExistsUptoDate* may be updated any number of times during object lifetime, but not after the object has ceased.

Likewise the column *ExistsFromDate* may be updated any number of times, but not after the object has started to exist.

Additional to the built-in versioning of updateable objects, a DOI-client should be aware of that users may do versioning by letting one updateable object cease at a point in time and at the same time create a new updateable object that exists from the same point in time.

A good example is how users do “versioning” of the Line object, which is an updateable object. If there is a significant change in the operation of a Line, the users may close the old Line by setting the *ExistsUptoDate* for that Line, and create a new Line object with the same Line number (and as a consequence the same Gid) that exists from the same point in time as the old Line closes. Note that the rule in PubTrans stating that a Gid must not point to more than one object at a specific point in time is obeyed in this case.

The conclusion is that updateable objects in some cases look like period state objects, because the users may choose to provide data in this way.

2.2.2 Period state objects

Period state objects are stored in one or several rows, one row for each version, with non-overlapping validity periods expressed in the columns *ExistsFromDate* and *ExistsUptoDate*. Only the last version may have infinite validity, which is expressed as having a NULL-value in the column *ExistsUptoDate*.

Changes of a period state object always result in a new row with a new *Id*-value. The only exception is that the column *ExistsUptoDate* may be set for an existing object, meaning that the specific version ceases at that point in time. This column may be set of two reasons; a new version supersedes from that point in time, or the object is ceased by explicitly setting the point in time of the last version of the object (i.e. there is no new version). When a new version is created, the *ExistsUptoDate* field of the previous version is always updated.

To detect these changes a period state object has a column named *LastModifiedUtcDateTime* that makes it possible to detect when the column *ExistsUptoDate* was last changed.

For objects with a Gid-value it is possible to compare versions, because a Gid is always unique at a specific point in time. For period state objects without a Gid it is not possible to compare versions (because it is not relevant).

2.2.3 Dated state objects

For dated state objects there exist tables that contain a cross reference between a date and an object Id. The corresponding views are named DatedXyz.[4] These tables may also contain other columns that are specific for the dated object. Sometimes these Dated-tables also contain redundant data for optimised search and access.

Changes of dated state objects are notified by object version per date as needed. For a valid dated state object, the column *IsReplacedById* is NULL. When an object version is outdated, the column *IsReplacedById* is set to zero or to the Id of the replacing object.

2.2.4 Objects without explicit versioning

There also exist tables with no explicit versioning, which means that there is no versioning information in the table that holds the object's data.

These objects are valid in the context of the validity and version of a parent object. So if a table with non-explicit versioned objects is joined with another table with objects that has versioning, the validity of the non-explicit versioned object is equal to the parent object.

Because an object in a non-versioned table may be referenced from several versioned objects, a client system should never try to deduct validity on these objects in general; just use them in context of their parent object. In addition, the value in the Id column of these objects is not guaranteed to be the same in different instances of DOI databases and should therefore not be used in cross database joins. An object without explicit versioning is an integral part of its parent object.

2.2.5 Objects with special versioning

In some cases there exist objects with special versioning. In such case, this is documented together with the view documentation.

2.2.6 Validity

A period of validity for an object is represented by a pair of columns: *ExistsFromDate* and *ExistsUptoDate*. If *ExistsUpToDate* is NULL the object is considered to have an infinite validity. Also note that NULL is not the default value for *ExistsUpToDate*.

---

[4] Note that DOI contains a view called Dated Vehicle Journey In Dated Block which is not a dated state object, but a view that connects Dated Vehicle Journey and Dated Block.

Validity is also allowed to be negative, i.e. the object turns invalid before it is valid.

2.2.7 Date-expansion process

To optimise read access and simplify for client applications to find valid data for a specific date, PubTrans regularly performs date expansion for some data objects. Date-expansion simply means to create a cross reference between dates and object *Id*'s.

The dated objects in DOI are date-expanded up to an internal release date specified in the table referenced by the Transport Authority view. Please, note that this date may be different for each Transport Authority.

2.3 Backwards compatibility

Earlier versions are supported by a combination of interface views and separate replicated tables. Although it is possible to have several versions of an interface running in the PubTrans central system, this will be used sparingly.

2.3.1 Interface versioning

The replicated tables for DOI version 4 are named **doi4_ObjectName**. These tables may be altered along with the development of DOI version 4. On top of these tables, a set of interface views can be installed in the target database. The users will then have the option to install a set of interface views that corresponds to the version of DOI they want to use.

With this model, upgrades of DOI can be installed together with upgrades of the interface views that maintain the interface version used by an application.

This versioning model of the interface tables should not be regarded as a reason for not upgrading subscribing applications in the future. Nevertheless, it makes it easier to plan and implement upgrades independent of upgrades of DOI.

Interface views will be supported to make it possible to plan and upgrade subscribing applications within a limited time from the latest release of DOI.

The replicated tables for the previous version of DOI, DOI version 3, are named **doiObjectName**. It will thus be possible to combine replicated tables for DOI 3 and DOI 4 without name collisions in the same database.

2.4 Special applications

This section gives some guidelines for using DOI for some special purposes.

2.4.1 Supplying timetable data to vehicle systems

Data supply of vehicle information systems usually means to distribute data to computers in the vehicles. This data distribution is usually faced with two main restrictions: the limited communication bandwidth between the central system and the onboard vehicle system, and the limited capacity of the vehicle system itself. In both cases, the data distribution mechanism must consider the amount of data that actually needs to be transferred and stored.

DOI is defined to allow systems to automatically and continuously supply vehicle systems with data. It is built with the purpose of minimising the amount of data that needs to be transferred to the vehicles while yet enforcing that data is correct even with frequent timetable updates. There should be no need for any

manual steps in processing of the data in an external Vehicle server before transferring the information to the vehicles.

It is also important to note that version management is not as simple as storing two sets of timetables and having a date for switching from one to the other. Instead, version management is made with a much higher resolution and without specific periods. This makes it possible for incremental timetable updates with short notice without flooding the communication system with large data volumes.

To use data according to DOI as the data source, the central server of the vehicle system (referred to as the AVL-Server) is set up as a target database for data replication. It is then the AVL-Server's task to transfer data to the vehicles and assure that vehicles have the correct versions, as described in chapter *13 Dated Timetable.* Due to the design of the versioning in DOI, updating vehicle system data is merely a continuously ongoing process, instead of something that occurs at official timetable changes.

The structure of the replicated data is such that reuse of existing data and incremental updates of the vehicle systems are supported. This is described in more detail in chapter *15 Lines in detail* and *16 Timetable.*

3 Working with replicated data

3.1 How to handle replicated data

The subscribing system will automatically receive new data into the target tables. These data 'arrives' as ordinary INSERT, UPDATE or DELETE operations. It is important to know when and how these operations should be interpreted.

There will be no alterations (ALTER) or deletion (DROP) of tables once the replication is set up and started.

The subscriber is allowed to create own views as long as the WITH SCHEMABINDING option is not used, since it prevents TRUNCATE TABLE statements from being executed (in case of initialisation or re-initialisation).

3.1.1 Request-driven applications

If the presentation system is a request-driven system that refreshes its information at regular intervals or upon user request, no action on data arrival is usually required. Just executing a SELECT statement gets the latest state of information.

3.1.2 Event-driven applications

If the presentation system is event-driven, special actions must be implemented to signal changes. One could set up dedicated triggers to take appropriate actions upon INSERT, UPDATE or DELETE operations in the tables. Any such triggers or other alterations of the concerned tables should be scripted and stored so that they can easily be reapplied if needed.

Note that a replication re-initialisation doesn't fire triggers, since a re-initialisation involves a TRUNCATE TABLE statement being executed on the subscribed table(s) followed by a bulk copy of data to the subscribed table(s). Also be aware of that triggers decrease performance in the replication.

3.2 Local changes

It is not recommended to make local changes of data in replicated tables. It may affect the integrity of the data and the stability of the replication itself. If changes are made, they will not be replicated back to PubTrans.

4 Basic administrative

This publication contains administrative information, such as names of Departments, Organisations, addresses and telephone numbers.

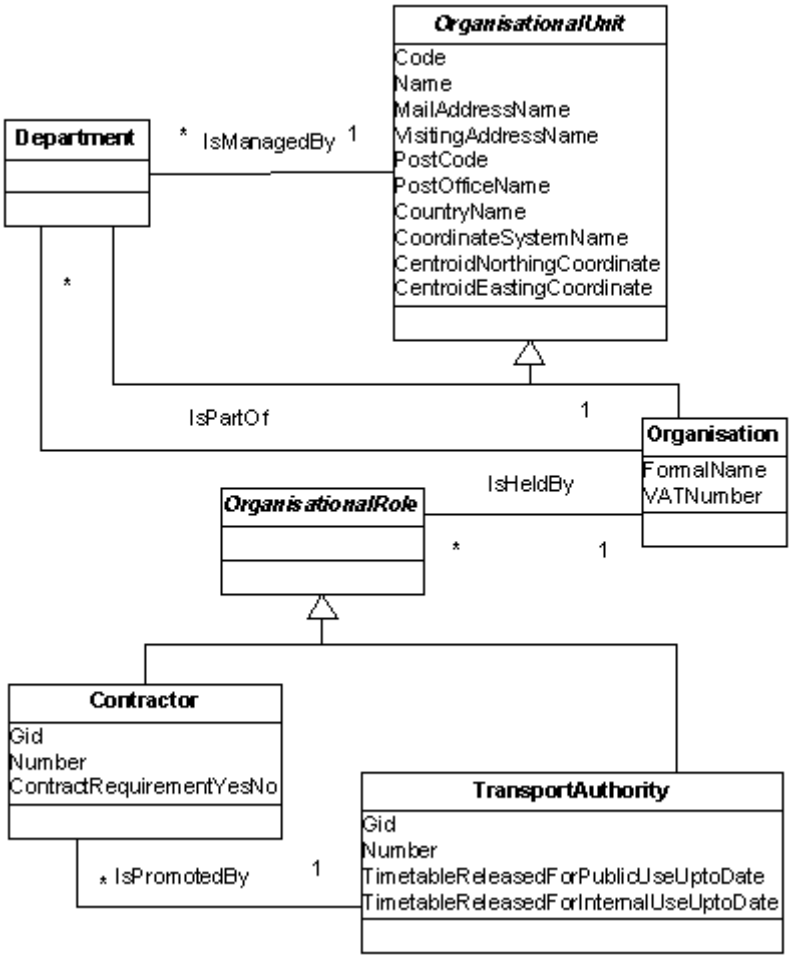

Fig. 1. Basic administrative objects.

4.1 Organisational Unit

An Organisational Unit is a grouping of responsibilities in a public transportation company for planning, passenger information and sales, management and control or economical result. Organisations and Departments are derived from Organisational Units.

Column name	Data type	Remark
Id	typId	Unique. This attribute has the same value as the Id of the associated Organisation or Department.
Code	typCode	

Column name	Data type	Remark
Name	typName	May be updated.
MailAddressName	typName	May be updated. May be NULL.
VisitingAddressName	typName	May be updated. May be NULL.
Postcode	typPostcode	May be updated. May be NULL.
PostOfficeName	typName	May be updated. May be NULL.
CountryName	typName	May be updated May be NULL.
CoordinateSystemName	typName	May be updated. May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL. May be updated.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL. May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.1.1 Versioning

An Organisational Unit is an updateable object.

4.2 Organisation : Organisational Unit

An Organisation is a top-level Organisational Unit. A company is an Organisation.

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	Unique.
VatNumber	typVatNumber	May be updated. May be NULL.
Name	typName	May be updated.
FormalName	typName	May be updated. May be NULL.
MailAddressName	typName	May be updated. May be NULL.
VisitingAddressName	typName	May be updated. May be NULL.
Postcode	typPostcode	May be updated. May be NULL.
PostOfficeName	typName	May be updated. May be NULL.
CountryName	typName	May be updated. May be NULL.
CoordinateSystemName	typName	May be updated. May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL. May be updated.

Column name	Data type	Remark
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL. May be updated.
OrganisationType	typOrganisationType	May be updated. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.2.1 Versioning

An Organisation is an updateable object.

4.3 Department : Organisational Unit

A Department is an Organisational Unit that is part of an Organisation.

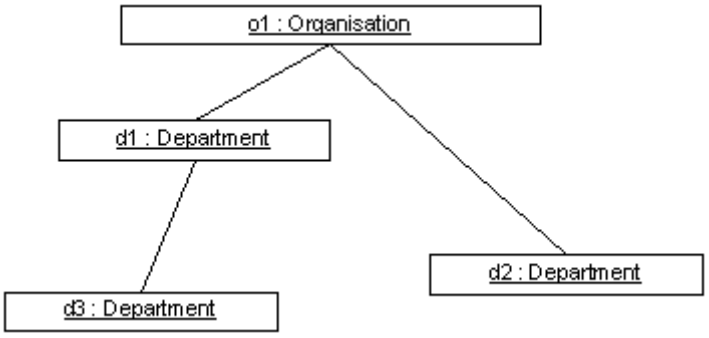

Fig. 2. Example of department hierarchy.

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	Unique in combination with *IsPartOfOrganisationId*.
Name	typName	May be updated.
MailAddressName	typName	May be updated. May be NULL.
VisitingAddressName	typName	May be updated. May be NULL.
Postcode	typPostcode	May be updated. May be NULL.
PostOfficeName	typName	May be updated. May be NULL.
CountryName	typName	May be updated. May be replaced by a reference to a separate Country object in the future. May be NULL.
CoordinateSystemName	typName	May be updated. May be NULL.

Column name	Data type	Remark
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL. May be updated.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL. May be updated.
IsPartOfOrganisationId	typId	The top-level Organisational Unit that this Department is part of.
IsManagedByOrganisationalUnitId	typId	May be updated. The next higher-level Organisational Unit in the hierarchy that manages this Department.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.3.1 Versioning

A Department is an updateable object.

4.4 Transport Authority : [Organisational Role]

A Transport Authority is an Organisation that has the role of organising the Public Transport in a region, while letting Contractors (vehicle operators) carry out the actual vehicle operation.

Column name	Data type	Remark
Id	typId	Unique.
IsOrganisationId	typId	Unique. This is the Id of the Organisation that acts as a Transport Authority.
Gid	typGid	Unique at a specific point in time.
Number	typTransportAuthority-Number	Unique at a specific point in time.
UsesPublicCalendarDefinedBy-TransportAuthorityId	typId	May be updated. May be NULL.
TimetableReleasedForPublicUseUptoDate	typDate	May be updated.
TimetableReleasedForInternalUseUptoDate	typDate	May be updated. Dated data is date-expanded up to this date.
DefaultLanguageCode	typLanguageCode	May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.4.1 Versioning

A Transport Authority is an updateable object.

4.5 Contractor : [Organisational Role]

A Contractor is an Organisation that can operate all or part of a network, on behalf of, or with the approval of a certain Transport Authority. Note that an Organisation can have the role as a Contractor towards different Transport Authorities.

In PubTrans the term Contractor is used to describe the role a vehicle operator holds in relation to a Transport Authority. A Contractor, in PubTrans context, is thus a company that operates vehicles in public transport on behalf of, or with the approval of, a transport authority. In PubTrans we give this term a wider meaning than the strict commercial aspect. TRANSMODEL uses the term OPERATOR to cover this and some related aspects. As the Scandinavian situation is complex with multilevel public authorities, and publicly owned companies more or less involved in the operation of public transport, the use of OPERATOR could cause confusion and is thus avoided in PubTrans 5. The word OPERATOR is replaced by 'Contractor' or 'Transport Authority' accordingly

Column name	Data type	Remark
Id	typId	Unique.
IsOrganisationId	typId	This is the Id of the Organisation that acts as a Contractor. Note that this Id is not unique in this view.
Gid	typGid	Unique at a specific point in time.
Number	typContractorNumber	Unique in combination with *IsPromotedBy-TransportAuthorityId* at a specific point in time.
ContractRequirementYesNo	typYesNo	May be updated.
IsPromotedByTransportAuthorityId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.5.1 Versioning

A Contractor is an updateable object.

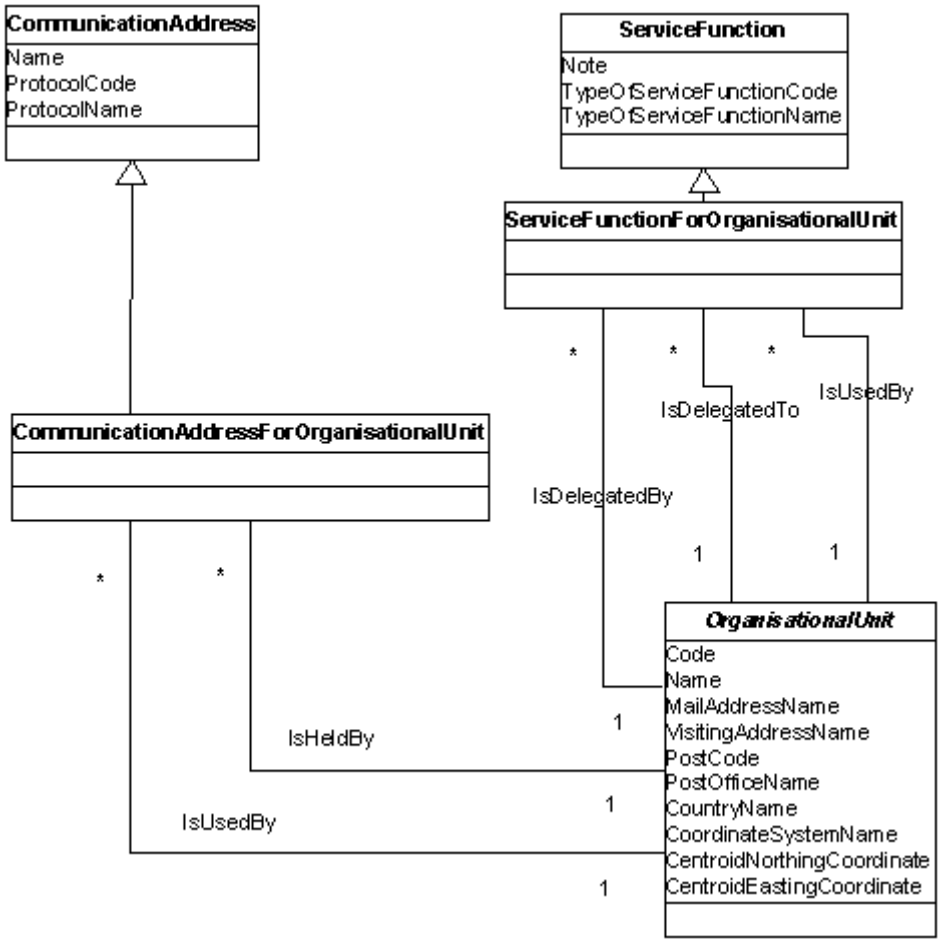

Fig. 3. Communication Address and Service Function.

4.6 Communication Address For Organisational Unit

This view contains information how to communicate with a specific Organisational Unit.

Column name	Data type	Remark
Id	typId	Unique.
Name	typName	+4630366600 for example.
ProtocolCode	typProtocolCode	
ProtocolName	typName	'Telephone' for example.
IsUsedByOrganisationalUnitId	typId	Organisational Unit that can be reached using this Communication Address.

Column name	Data type	Remark
IsHeldByOrganisationalUnitId	typId	Organisational Unit that actually holds the information about the communication address. This information is only provided as information, and should not be used in any presentation system.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.6.1 Examples

If one needs the telephone number of a specific Organisational Unit one can use the following query:

```
SELECT @ItsPhoneNumber = Name
FROM CommunicationAddressForOrganisationalUnit
WHERE IsUsedByOrganisationalUnitId =@OrganisationalUnitId
  AND ProtocolCode = 'PHONE'
  AND (ExistFromDate <= @RequestedDate
       AND (ExistsUptoDate > @RequestedDate
            OR ExistsUptoDate IS NULL))
```

4.6.2 Versioning

A Communication Address For Organisational Unit is a period state object

4.7 Service Function For Organisational Unit

An Organisational Unit on any level within a hierarchy can provide administrative service functions such as *Lost and Found*, *Complaints* et cetera.

Some service functions (such as *Lost and Found*) are managed by a designated Department in an Organisation. Some service functions are even delegated to and handled by external Organisational Units. Typically many Departments within an Organisation use a shared service function for *Lost and Found*. This view represents an expansion of this information, providing information from the perspective of each Department of an Organisation.

Column name	Data type	Remark
Id	typId	Unique.
TypeOfServiceFunctionCode	typServiceFunction-Code	
TypeOfServiceFunctionName	typName	Lost and found, for example.
Note	typNote	May be NULL. Open 9 – 17, for example.
IsUsedByOrganisationalUnitId	typId	Organisational Unit that can use this function.

Column name	Data type	Remark
IsDelegatedByOrganisationalUnitId	typId	Organisational Unit that actually holds the delegation of this service function. This information is only provided as information, and should not be used in any presentation system.
IsDelegatedToOrganisationalUnitId	typId	Organisational Unit that provides the service function.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

4.7.1 Examples

To find which Organisational Unit that provides a specific service function to a named Department, one could use the following query:

```
SELECT @ProviderOfFunctionOrganisationalUnitId =
IsDelegatedToOrganisationalUnitId
FROM ServiceFunctionForOrganisationalUnit
WHERE IsUsedByOrganisationalUnitId = @MyNamedDepartmentId
  AND TypeOfServiceFunctionCode = 'LOSTANDF'
  AND (ExistFromDate <= @RequestedDate
       AND (ExistsUptoDate > @RequestedDate
        OR ExistsUptoDate IS NULL))
```

If one also needs the telephone number to the Lost and Found function one can subsequently use the following query:

```
SELECT @ProviderPhoneNumber = Name
FROM CommunicationAddressForOrganisationalUnit
WHERE IsUsedByOrganisationalUnitId = @ProviderOfFunctionOrganisationalUnitId
  AND ProtocolCode = 'PHONE'
  AND (ExistFromDate <= @RequestedDate
       AND (ExistsUptoDate > @RequestedDate
        OR ExistsUptoDate IS NULL))
```

4.7.2 Versioning

A Service Function For Organisational Unit is a period state object.

5 Tenders and Contracts

This publication contains information about Tenders and Contracts.

5.1 Tender

Column name	Data type	Remark
Id	typId	Unique.
Name	typName	May be updated.
Code	typCode	Unique in combination with *IsPromotedBy-TransportAuthorityId* at a specific point in time.
Note	typNote	May be updated. May be NULL.
IsPromotedByTransportAuthorityId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

5.1.1 Versioning

A Tender is an updateable object.

5.2 Contract

The Contract view corresponds to the existence of a specific Contract between a Transport Authority and a Contractor. The Contract is usually signed as a result of a tendering process. There may be any number of Contracts promoted by a Transport Authority to a Contractor.

Column name	Data type	Remark
Id	typId	Unique.
ReferenceName	typName	Unique in combination with *IsPromotedByTransportAuthorityId* at a specific point in time.
Name	typName	May be updated. May be NULL.
Note	typNote	May be updated. May be NULL.
IsPromotedByTransportAuthorityId	typId	
IsIncludedInTenderId	typId	May be updated. May be NULL.
IsPromotedToContractorOrganisationId	typId	May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.

Column name	Data type	Remark
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

5.2.1 Versioning

A Contract is an updateable object.

5.3 Line In Contract

The Line In Contract view contains information about which Lines that are operated under a certain Contract.

Column name	Data type	Remark
Id	typId	Unique.
IsLineId	typId	
IsPromotedInContractId	typId	
LevelOfOperationCode	typLevelOfOperation-Code	May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

5.3.1 Note

Line In Contract is derived from Contractor Of Line, and thus shares Id with that view.

5.3.2 Versioning

A Line In Contract is an updateable object.

6 Lines

This publication contains information about Lines. A Line is a group of Journey Patterns and Routes generally known to the public by a similar name or number. A Line can also be seen as an administrative group of Vehicle Journeys.

6.1 Line

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Number	typLineNumber	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	typName	May be updated. May be NULL.
Designation	typLineDesignation	May be updated.
DefaultTransportModeCode	typTransportModeCode	
IsDefinedByTransportAuthorityId	typId	
MonitoredYesNo	typYesNo	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

6.1.1 Versioning

A Line is an updateable object. Relations from referring objects (Contracts) will not be broken as long as the same *Id* is retained. When a *Gid* is reused for a new Line object, the old object will be closed, that is, *ExistsUptoDate* be set. The new object will use a new *Id*.

6.2 Direction Of Line

A Line may have one or two Direction Of Lines.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time. May be updated.
DirectionCode	typDirectionCode	
Name	typName	May be updated. The main heading for the

Column name	Data type	Remark
		direction. May be NULL.
IsOnLineId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.
DescriptionNote	typNote	May be updated. May be NULL.

6.2.1 Versioning

A Direction Of Line is an updateable object.

6.3 Group Of Lines

Lines may be grouped into group of Lines for particular purposes, such as quality of service, merged timetables, and Vehicle or driver scheduling or fare collection. A Line may belong to more than one group. It is not mandatory to group Lines.

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	May be updated. May be NULL.
Name	typName	
PurposeOfLineGroupingCode	typLineGroupingCode	
IsManagedByOrganisationId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

6.3.1 Usage note

A Line can at any one specific time only be in one Group Of Lines with *PurposeOfLineGroupingCode* = 'PRODUCT'.

6.3.2 Versioning

A GroupOfLines is an updateable object.

6.4 Line In Group Of Lines

Column name	Data type	Remark
Id	typId	Unique.
IsInGroupOfLinesId	typId	
IsLineId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

6.4.1 Versioning

A Line In Group Of Lines is a period state object.

6.5 Contractor Of Line

This view contains information of which Organisations that are allowed to operate a certain Line. There is also information of which Department within that Organisation that is actually responsible for the control of the day-to-day operation of the Line in question.

Column name	Data type	Remark
Id	typId	Unique.
DefaultYesNo	typYesNo	May be updated.
LevelOfOperationCode	typLevelOfOperation-Code	May be updated.
IsOperatingLineId	typId	
IsContractorOrganisationId	typId	
ControlsLineByOrganisationalUnitId	typId	May be updated. May be NULL.
IsUnderContractId	typId	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

6.5.1 Versioning

A Contractor Of Line is an updateable object.

7 Points, Places and Zones

This publication contains information about Journey Pattern Points, Stop Points, Stop Areas, Zones and what Zones the different Journey Pattern Points are included in. It also contains Route Links and Connection Links.

7.1 Journey Pattern Point

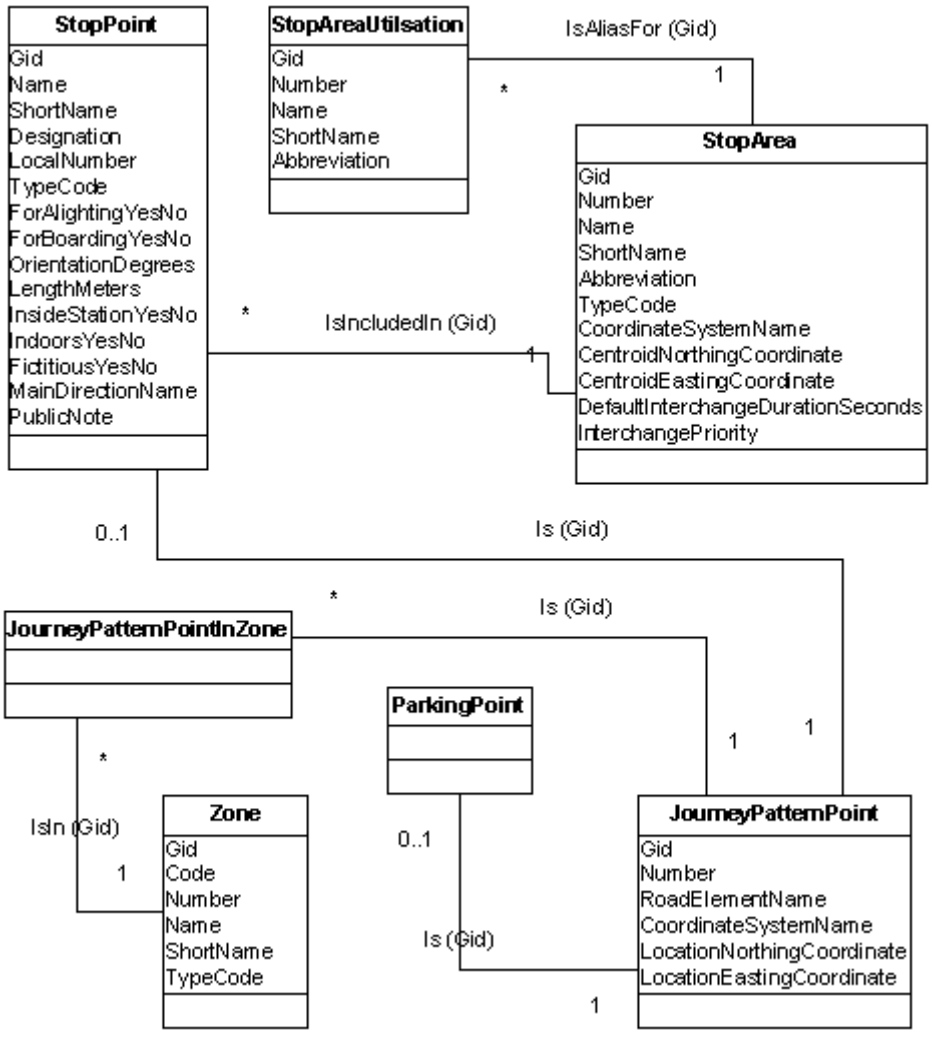

Fig. 4. JourneyPatternPoint

A Journey Pattern Point represents a super class for all types of Points in a Journey Pattern.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.

Column name	Data type	Remark
Number	typJourneyPatternPoint-Number	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
IsDefinedByTransportAuthorityId	typId	
RoadElementName	typName	Name of road/street where the Journey Pattern Point is located. May be NULL.
CoordinateSystemName	typName	May be NULL.
LocationNorthingCoordinate	typCoordinate	Northing. May be NULL.
LocationEastingCoordinate	typCoordinate	Easting. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.1.1 Usage note

The view contains information of the location of Journey Pattern Points:

The *CoordinateSystemName*-field defines which kind of coordinate system that is in use while the *LocationNorthingCoordinate* and *LocationEastingCoordinate* are the coordinates in that coordinate system.

There are also identity fields of the Journey Pattern Points; *Id* and *Gid*, which can be referred to from other views.

ExistsFromDate and *ExistsUptoDate* express the period of validity. The latter field may be set to NULL if there is no known limit for how long the Journey Pattern Point will exist.

To get the name of specific Stop Points, join this view with the Stop Point view using the Journey Pattern Point *Id* as key while also considering the time aspect.

7.1.2 Versioning

A Journey Pattern Point is a period state object.

7.2 Parking Point

Parking Points are Journey Pattern Points where vehicles may stay unattended for a long time. A parking Point can be located at a Garage or elsewhere. It can be used as the end or start of a DeadRun.

Column name	Data type	Remark
Id	typId	Unique.
IsJourneyPatternPointGid	typGid	
LastModifiedUtcDateTime	typDateTime	May be updated.

Column name	Data type	Remark
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.2.1 Usage note

Information about location coordinates et cetera can be found by joining with the view Journey Pattern Point.

7.2.2 Versioning

Parking Point is a period state object.

7.3 Stop Point

Stop Points are Journey Pattern Points where passengers can alight or board.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Name	typName	May be NULL.
ShortName	typShortName	May be NULL.
Designation	typStopPointDesignation	May be NULL.
LocalNumber	typStopPointLocalNumber	Unique in combination with *IsIncludedInStopAreaGid* at a specific point in time.
IsJourneyPatternPointGid	typGid	
IsIncludedInStopAreaGid	typGid	
TypeCode	typStopPointTypeCode	
ForAlightingYesNo	typYesNo	
ForBoardingYesNo	typYesNo	
OrientationDegrees	typDirectionDegrees	Direction of stopping Vehicle. May be NULL.
LengthMeters	typMeters	Available space for vehicles. May be NULL.
InsideStationYesNo	typYesNo	
IndoorsYesNo	typYesNo	
FictitiousYesNo	typYesNo	

Column name	Data type	Remark
MainDirectionName	typName	May be NULL.
PublicNote	typNote	May be NULL.
HasEquipmentProfileId	typId	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.3.1 Usage note

The Stop Point view contains information about the names of the Stop Points; *Name* and *ShortName*.

A Stop Point is always included in a Stop Area. Information about the name and other attributes of the Stop Area can be extracted by joining this view with Stop Area.

A Stop Point is a Journey Pattern Point. Information about location coordinates et cetera can be found by joining with the view Journey Pattern Point.

The joins must take the validity of the involved objects into consideration. A simplified method is to use the Stop Point Validity view, which exposes the valid combinations directly.

7.3.2 Versioning

Stop Point is a period state object.

7.4 Stop Point Validity

This is a redundant view describing which combinations of Stop Point, Stop Area and Journey Pattern Point that are valid in certain periods.

Column name	Data type	Remark
Id	typId	Unique.
IsForStopPointGid	typGid	
IsStopPointId	typId	
IsJourneyPatternPointId	typId	
IsJourneyPatternPointGid	typGid	
IsIncludedInStopAreaId	typId	
IsIncludedInStopAreaGid	typGid	
LastModifiedUtcDateTime	typDateTime	May be NULL. May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.4.1 Usage note

The Stop Point Validity is a simplified way of describing the combined validity of the three object types.

7.4.2 Versioning.

Stop Point Validity is a period state object.

7.5 Stop Point Key

Stop Point Keys are activation keys for default display and announcement related activities at a Stop. There can be multiple keys with the same type code to control different kinds of hardware. Display keys related to a Journey can override these keys. A StopPoint can have one StopPointKey per DeviceName and TypeCode at a specific point in time.

Column name	Data type	Remark
Id	typId	Unique.
IsForStopPointGid	typGid	
TypeCode	typStopPointKeyType-Code	'STOPNAME', 'ADD_INFO', 'STOPMESS', 'CHA_INFO'
DeviceName	typName	I.e. INIT_PA_INTERN, INIT_PA_EXTERN, INIT_HEADSIGN
ParameterData	typParameterData	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.5.1 Versioning

A Stop Point Key is a period state object.

7.6 Station Entrance Point

A Station Entrance Point is a point where a passenger can enter or leave a station. It is used as end point of transfer links to Stop Points inside stations.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
LocalNumber	typStationEntrancePoint-LocalNumber	Unique in combination with *IsIncludedInStopAreaGid* at a specific point in time.
Abbreviation	typAbbreviation	May be NULL.

Column name	Data type	Remark
Name	typName	May be NULL.
IsIncludedInStopAreaGid	typGid	
CoordinateSystemName	typName	May be NULL.
LocationNorthingCoordinate	typCoordinate	Northing. May be NULL.
LocationEastingCoordinate	typCoordinate	Easting. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.6.1 Versioning

A Station Entrance Point is a period state object.

7.7 Station Entrance Point Opening Hours

A Station Entrance Point can be open or closed during different periods. The opening hours might differ depending on Day Type.

Column name	Data type	Remark
Id	typId	Unique.
IsForStationEntrancePointId	typId	
IsInScopeOfServiceCalendarId	typId	
IsOnPublicDayTypeId	typId	
OpeningOffsetDateTime	typOffsetDateTime	
ClosingOffsetDateTime	typOffsetDateTime	

7.7.1 Versioning

A Station Entrance Point Opening Hours is versioned by the owning Station Entrance Point object, thus a Station Entrance Point Opening Hours is an object without explicit versioning.

7.8 Stop Area

A Stop Area is a grouping of Stop Points.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.

Column name	Data type	Remark
Number	typStopAreaNumber	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	typName	
ShortName	typShortName	May be NULL.
Abbreviation	typAbbreviation	May be NULL.
TypeCode	typStopAreaTypeCode	
IsDefinedByTransportAuthorityId	typId	
CoordinateSystemName	typName	May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL.
DefaultInterchangeDurationSeconds	typDurationSeconds	May be NULL.
InterchangePriority	typInterchangePriority	May be NULL
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.8.1 Versioning

A Stop Area is a period state object.

7.9 Stop Area Utilisation

A Stop Area Utilisation is an alias for a Stop Area defined by another Transport Authority.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Number	typStopAreaNumber	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	typName	
ShortName	typShortName	May be NULL.
Abbreviation	typAbbreviation	May be NULL.
IsAliasForStopAreaGid	typGid	
IsDefinedByTransportAuthorityId	typId	

Column name	Data type	Remark
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.9.1 Versioning

A Stop Area Utilisation is a period state object.

7.10 Zone

A Zone represents an area of some kind. Note that there is no spatial data available about this object.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Code	typCode	May be NULL.
Number	typZoneNumber	Unique in combination with *IsDefinedByTransportAuthorityId* and *TypeCode* at a specific point in time.
Name	typName	May be NULL.
ShortName	typShortName	May be NULL.
TypeCode	typZoneTypeCode	
IsDefinedByTransportAuthorityId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.
CoordinateSystemName	typName	May be updated. May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL. May be updated.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL. May be updated.

7.10.1 Versioning

A Zone is a period state object.

7.11 Journey Pattern Point In Zone

This view contains a mapping between Journey Pattern Points and the Zones they are in. A Journey Pattern Point may be member of several Zones at the same time.

Column name	Data type	Remark
Id	typId	Unique.
IsJourneyPatternPointGid	typGid	
IsInZoneGid	typGid	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.11.1 Versioning

A Journey Pattern Point In Zone is a period state object.

7.12 Place

A Place is a generic description of all geographical elements that may be used to **define** the origin and destination of a passenger trip. Note that there is no spatial data available about this object.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Number	typPlaceNumber	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	typName	
ShortName	typShortName	May be NULL.
TypeCode	typPlaceTypeCode	
IsDefinedByTransportAuthorityId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.12.1 Versioning

A Place is a period state object.

7.13 Site : Place

A Site is a well-known PLACE to which a passenger may refer to **indicate** the origin or the destination of a passenger trip.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Number	typPlaceNumber	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	typName	
ShortName	typShortName	May be NULL.
TypeCode	typPlaceTypeCode	
IsDefinedByTransportAuthorityId	typId	
Abbreviation	typAbbreviation	May be NULL.
InternalComment	typComment	May be NULL.
PublicNote	typNote	May be NULL.
CoordinateSystemName	typName	May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

7.13.1 Versioning

A Site is period state object.

7.14 Site Neighbouring Stop Area

This view describes which Stop Areas that are close to a certain Site.

Column name	Data type	Remark
Id	typId	Unique.
IsSiteId	typId	
IsNeighbouringStopAreaGid	typGid	

7.14.1 Versioning

A Site Neighbouring Stop Area is versioned by the owning Site object, thus a Site Neighbouring Stop Area is an object without explicit versioning.

8 Route Links and Points on Route Links

This publication contains information about Route Links and Points on Route Links. Points on a Route Links can for instance be pre-emption points and extra timing points.

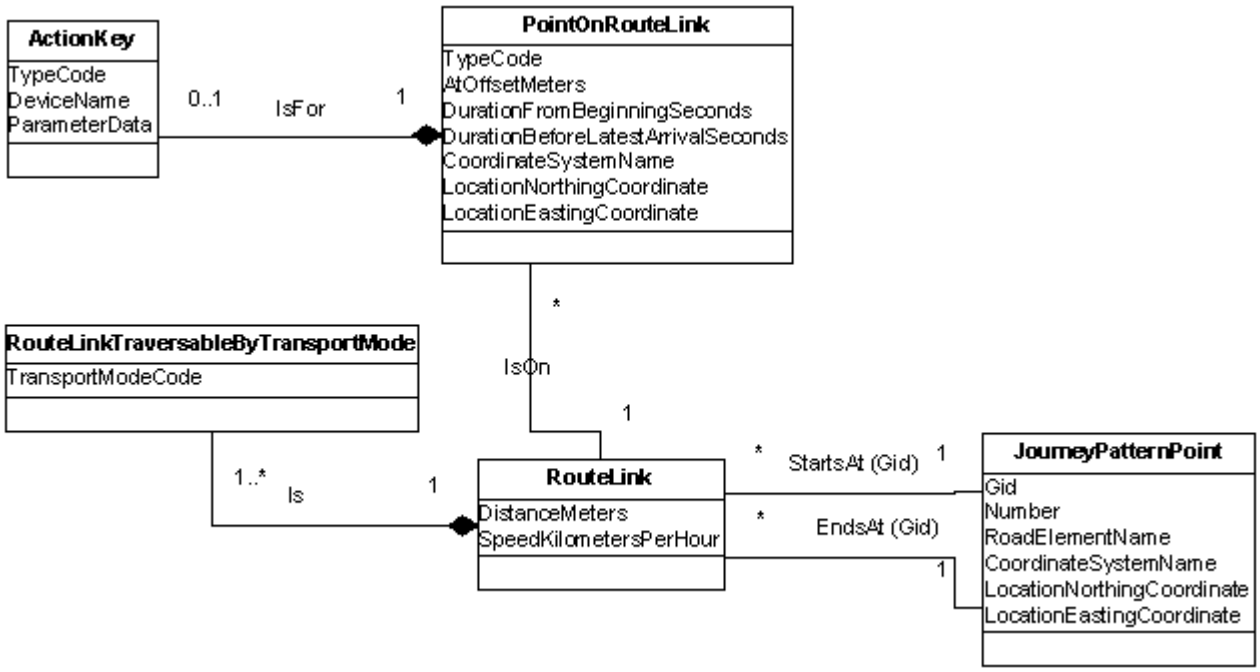

Fig. 5. Route Links and pre-emption Points

8.1 Route

A Route is the physical path of a Vehicle Journey through all the Points along a Journey Pattern.

There is no separate object for Routes in DOI. However, the Route for a specific Journey Pattern may be derived using the view PointInJourneyPattern and using a geographical information system to calculate the shortest path through all Points in the Journey Pattern in the sequence they are passed.

8.2 Route Link

A Route is made up of a continuous sequence of Route Links. A Route Link is a path between two adjacent Journey Pattern Points.

Column name	Data type	Remark
Id	typId	Unique.
StartsAtJourneyPatternPointGid	typGid	Unique in combination with *EndsAtJourneyPatternPointGid* at a specific point in time for a specific *TransportModeCode*.

Column name	Data type	Remark
EndsAtJourneyPatternPointGid	typGid	Unique in combination with *StartsAtJourneyPatternPointGid* at a specific point in time for a specific *TransportModeCode*.
Name	typName	May be NULL.
DistanceMeters	typMeters	
SpeedLimitKmPerHour	typKmPerHour	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

8.2.1 Versioning

A Route Link is a period state object. Observe that a Route Link is unique at a specific point in time for the combination of *StartsAtJourneyPatternPointGid* , *EndsAtJourneyPatternPointGid* and *TransportModeCode*. The relevant *TransportModeCode(s)* are obtained by joining with view Route Link Traversable By Transport Mode.

8.3 Route Link Traversable By Transport Mode

This view defines what transport modes that may traverse a Route Link.

Column name	Data type	Remark
Id	typId	Unique.
IsRouteLinkId	typId	
TransportModeCode	typTransportModeCode	

8.3.1 Versioning

A Route Link Traversable By Transport Mode is versioned by the owning Route Link object, thus a Route Link Traversable By TransportMode is an object without explicit versioning.

There is at least one Route Link Traversable By Transport Mode for each Route Link, but there may actually be more. In rare cases a Route Link will be passable for more than one Transport Mode, e.g. when buses and trams share the same street space.

8.4 Point On Route Link

Additional Points may exist on a Route Link. The location of such Points are defined as an offset from the beginning of the Route Link and optionally also with a coordinate. The type of Point indicates the purpose and usage of the Point.

The view describes Points on links that may be used for different purposes. Typically, it will be used for automatic vehicle location and control.

Column name	Data type	Remark
Id	typId	Unique.
TypeCode	typPointOnLinkType-Code	
AtOffsetMeters	typMeters	May be NULL. The distance offset relative to the beginning of the RouteLink.
DurationFromBeginningSeconds	typDurationSeconds	May be NULL. The time offset from the later of EarliestDeparture and LatestArrival time at the start of the RouteLink.
DurationBeforeLatestArrivalSeconds	typDurationSeconds	May be NULL. The time offset to the LatestArrival time at the end of the RouteLink.
CoordinateSystemName	typName	May be NULL.
LocationNorthingCoordinate	typCoordinate	Northing. May be NULL.
LocationEastingCoordinate	typCoordinate	Easting. May be NULL.
IsOnRouteLinkId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

8.4.1 Using Point On Route Link

There are several different uses for this object. For example:

- Activation Points to request and release traffic light priority.
- Timing Points to force the sending of extra progress reports.
- Mapping Points to enhance the description of the intended path on a Route Link.

8.4.2 Variants of Point On Route Link

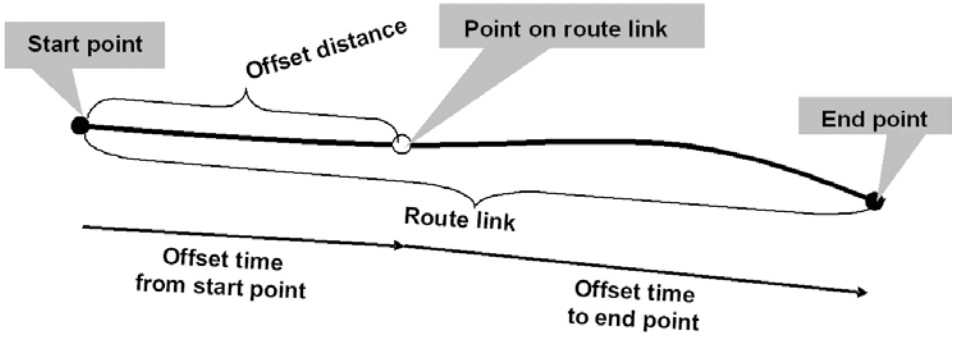

Fig. 6. Point On Route Link.

Table 1 Timing Points

Case no.	Offset dist-ance	Offset time from start Point	Offset time to end Point	Usage	Remark – Timing Points
1	Value	NULL	NULL	The Point is located OffsetMeters from start of Route Link.	Send a VSI Extra Progress Report when this Point is reached. (Use Id of Point On Route Link as PointRef). Do not send DelayReport.
2	Value	Value	NULL	As case 1 plus that the Point has a passage time calculated from a time offset added to the later of latest arrival and earliest departure time at the start point.	Send a VSI Extra Progress Report at the calculated passage time if this Point has not been passed at that time.
3	Value	NULL	Value	As case 1 plus that the Point has a passage time calculated from a time offset subtracted from the latest arrival time at the end point.	Send a VSI Extra Progress Report at the calculated passage time if this Point has not been passed at that time.
4	Value	Value	Value	A combination of case 2 and 3.	See case 2 and 3 respectivly.

Using different combinations of values, it is possible to streamline desired behaviour as described in the table.

Table 2 Mapping Points

Case no.	Offset dist-ance	Offset time from start Point	Offset time to end Point	Usage	Remark – Mapping Points
1	Value	NA	NA	The mapping Point is located OffsetMeters from start of Route Link.	Coordinates for this Point.

Table 3 Activation Points

Case no.	Offset dist-ance	Offset time from start Point	Offset time to end Point	Usage	Remark – Activation Points
1	Value	NULL	NULL	The Point is located AtOffsetMeters from start of Route Link.	Activate when this Point is reached.
2	Value	Value	NULL	As case 1 plus that the Point has a passage time calculated from a time offset relative the later of latest arrival and earliest departure time at the start point.	Activate at the calculated earliest departure time if this Point has not been reached.
3	Value	NULL	Value	As case 1 plus that the Point has a latest arrival time calculated from a time offset relative the latest arrival time at the end point.	Activate when this Point is reached IF the calculated latest arrival time has passed, otherwise do not activate.
4	Value	Value	Value	A combination of case 2 and 3.	Activate if both activation conditions are met as described in case 2 and 3.
5	NULL	Value	NULL	The Point where the Vehicle is according to the specified duration after departing from the start point.	Activate when the calculated earliest departure time has passed.
6	NULL	NULL	Value	The Point where the Vehicle is according to the specified duration before arriving at the end point.	Activate when the calculated latest arrival time has passed.
7	NULL	Value	Value	Combination of case 5 and 6.	Leads to two separate activations.

8.4.3 Versioning

A Point On Route Link is a period state object.

8.5 Action Key

Action Key contains information concerning activation.

Column name	Data type	Remark
Id	typId	Unique.
IsForPointOnRouteLinkId	typId	
TypeCode	typActionTypeCode	E.g. REQ_PRIO (request signal priority).
DeviceName	typName	E.g. INIT_PREEMPTION
ParameterData	typParameterData	May be NULL. E.g.. NotificationNumber;Channel;TelegramType
LastModifiedUtcDateTime	typDateTime	May be NULL. May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

8.5.1 Versioning

An Action Key is versioned by the owning Point On Route Link object, thus an Action Key is actually an object without explicit versioning, and can be joined directly with Point On Route Link without checking the validity of the Action Key itself. However, the redundant data *ExistsFromDate* and *ExistsUptoDate* and *LastModifiedUtcDateTime* are also provided for ease of use.

9 Transfer Links

This publication contains information about different types of transfer links.

9.1 Connection Link

A Connection Link defines a transfer possibility between two Stop Points, usually a walking path.

Column name	Data type	Remark
Id	typId	Unique.
StartsAtJourneyPatternPointGid	typGid	Unique in combination with *EndsAtJourneyPatternPointGid* at a specific point in time.
EndsAtJourneyPatternPointGid	typGid	Unique in combination with *StartsAtJourneyPatternPointGid* at a specific point in time.
TransferModeCode	typTransferModeCode	
DistanceMeters	typMeters	May be NULL.
DefaultDurationSeconds	typDurationSeconds	
Comment	typComment	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

9.1.1 Versioning

A Connection Link is a period state object.

9.2 Station Entry Link

A Station Entry Link defines a transfer possibility from a Station Entrance Point to a StopPoint.

Column name	Data type	Remark
Id	typId	Unique.
StartsAtStationEntrancePointGid	typGid	Unique in combination with *EndsAtJourneyPatternPointGid* at a specific point in time.
EndsAtJourneyPatternPointGid	typGid	Unique in combination with *StartsAtStationEntrancePointGid* at a specific point in time.

Column name	Data type	Remark
TransferModeCode	typTransferModeCode	
DistanceMeters	typMeters	May be NULL.
DefaultDurationSeconds	typDurationSeconds	
Comment	typComment	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

9.2.1 Versioning

A Station Entry Link is a period state object.

9.3 Station Exit Link

A Station Exit Link defines a transfer possibility from a StopPoint to a Station Entrance Point.

Column name	Data type	Remark
Id	typId	Unique.
StartsAtJourneyPatternPointGid	typGid	Unique in combination with *EndsAtStationEntrancePointGid* at a specific point in time.
EndsAtStationEntrancePointGid	typGid	Unique in combination with *StartsAtJourneyPatternPointGid* at a specific point in time.
TransferModeCode	typTransferModeCode	
DistanceMeters	typMeters	May be NULL.
DefaultDurationSeconds	typDurationSeconds	
Comment	typComment	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

9.3.1 Versioning

A Station Exit Link is a period state object.

9.4 Obstacle On Link

An Obstacle On Link defines that there is a hindrance on a transfer link and what the primary mean to overcome this problem is. It could be that two StopPoints are located on different levels, and that a staircase must be used to transfer between them.

Column name	Data type	Remark
Id	typId	Unique.
IsOnTransferLinkId	typId	*Id* of concerned ConnectionLink, StationEntryLink or StationExitLink.
TransferLinkTypeCode	typTransferLinkTypeCode	
PassageDirectionCode	typPassageDirectionCode	
IsHandledByPrimaryBridgingDeviceGid	typGid	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

9.4.1 Versioning

An Obstacle On Link is a period state object.

10 Infrastructure

This publication contains information about infrastructure.

10.1 Bridging Device

A Bridging Device is a device used to pass an obstacle on a TransferLink. A Bridging Device could for example be an elevator, an escalator or a stairway.

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	Unique at a specific point in time.
Number	typBridgingDeviceNumber	Unique in scope of *IsDefinedByTransportAuthorityId* at a specific point in time.
TypeCode	typBridgingDeviceType-Code	
Name	typName	May be NULL.
Abbreviation	typAbbreviation	May be NULL.
IsDefinedByTransportAuthorityId	typId	
PassageDirectionCode	typPassageDirectionCode	
PassageDurationSeconds	typDurationSeconds	
IsLocatedAtStopAreaGid	typGid	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

10.1.1 Versioning

A Bridging Device is a period state object.

10.2 Alternate Bridging Device

This view describes alternate ways to pass the obstacle.

Column name	Data type	Remark
Id	typId	Unique.
IsBridgingDeviceGid	typGid	
IsAlternateToBridgingDeviceId	typId	

Column name	Data type	Remark
AdditionalPassageDurationSeconds	typDurationSeconds	May be NULL.
SequenceNumber	typSequenceNumber	

10.2.1 Usage note

The sequence number denotes in which order the Alternate Bridging Devices should be attempted.

10.2.2 Versioning

An Alternate Bridging Device is versioned by the Bridging Device object referred to in *IsAlternateToBridgingDeviceId*, thus an Alternate Bridging Device is an object without explicit versioning.

10.3 Bridging Device Opening Hours

A Bridging Device can be open or closed (available) during different periods. The opening hours might differ depending on Day Type.

Column name	Data type	Remark
Id	typId	Unique.
IsForBridgingDeviceId	typId	
IsInScopeOfServiceCalendarId	typId	
IsOnPublicDayTypeId	typId	
OpeningOffsetDateTime	typOffsetDateTime	
ClosingOffsetDateTime	typOffsetDateTime	

10.3.1 Versioning

A Bridging Device Opening Hours is versioned by the owning Bridging Device object, thus a Bridging Device Opening Hours is an object without explicit versioning.

10.4 Facility

Column name	Data type	Remark
Id	typId	Unique.
Name	typName	Unique in scope of *IsDefinedByTransportAuthorityId*.
TypeCode	typFacilityTypeCode	
IsDefinedByTransportAuthorityId	typId	
CoordinateSystemName	typName	May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL.

Column name	Data type	Remark
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

10.4.1 Versioning

A Facility is a period state object.

10.5 Car Park : Facility

A Car Park is a type of Facility.

Column name	Data type	Remark
Id	typId	Unique.
Name	typName	Unique in scope of *IsDefinedByTransportAuthorityId*.
TypeCode	typFacilityTypeCode	Always 'CARPARK'
IsDefinedByTransportAuthorityId	typId	
CoordinateSystemName	typName	May be NULL.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL.
IndoorCapacityCount	typCount	
MannedYesNo	typYesNo	May be NULL.
OutdoorCapacityCount	typCount	
SurveillanceYesNo	typYesNo	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

10.5.1 Versioning

A Car Park is a period state object.

10.6 Facility Neighbouring Stop Area

This view describes which Stop Areas that are close to this Facility.

Column name	Data type	Remark
Id	typId	Unique.
IsFacilityId	typId	
IsNeighbouringStopAreaGid	typGid	

10.6.1 Versioning

A Facility Neighbouring Stop Area is versioned by the owning Facility object, thus a Facility Neighbouring Stop Area is an object without explicit versioning.

10.7 Facility Hyper Link

This view describes hyper links that apply to this Facility.

Column name	Data type	Remark
Id	typId	Unique.
IsForFacilityId	typId	
Name	typHyperLinkName	Contains the actual hyper link.
PublicNote	typNote	May be NULL.

10.7.1 Versioning

A Facility Hyper Link is versioned by the owning Facility object, thus a Facility Hyper Link is an object without explicit versioning.

10.8 Facility Opening Hours

A Facility can be open or closed (available) during different periods. The opening hours might differ depending on Day Type.

Column name	Data type	Remark
Id	typId	Unique.
IsForFacilityId	typId	
IsInScopeOfServiceCalendarId	typId	
IsOnPublicDayTypeId	typId	
OpeningOffsetDateTime	typOffsetDateTime	
ClosingOffsetDateTime	typOffsetDateTime	

10.8.1 Versioning

A Facility Opening Hours is versioned by the owning Facility object, thus a Facility Opening Hours is an object without explicit versioning.

10.9 Equipment Profile

Describes combinations of equipment available at Stop Points.

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	Unique in scope of *IsDefinedByTransportAuthorityId*.
Name	typName	May be NULL. May be updated.
IsDefinedByTransportAuthorityId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

10.9.1 Versioning

An Equipment Profile is an updateable object.

10.10 Equipment

This view describes specific details for Equipment included in an Equipment Profile.

Column name	Data type	Remark
Id	typId	Unique.
IsIncludedInEquipmentProfileId	typId	
EquipmentTypeCode	typEquipmentTypeCode	Unique in scope of IsIncludedInEquipmentProfileId at a specific point in time.
DescriptionNote	typNote	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

10.10.1 Versioning

Equipment is a period state object.

11 Public Calendar

This publication covers the published mapping of day types to actual dates.

11.1 Service Calendar

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	Unique in combination with *IsDefinedByTransportAuthorityId* at a specific point in time.
Name	typName	May be updated.
IsDefinedByTransportAuthorityId	typId	
IsManagedByOrganisationalUnitId	typId	May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUpToDate	typDate	May be NULL. May be updated.

11.1.1 Versioning

A Service Calendar is an updateable object.

11.2 Public Day Type

This view contains both the standard day types 1 – 7 corresponding to Monday – Sunday and any service day types defined by the Transport Authority.

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	Unique in combination with *IsDefinedByTransportAuthorityId*.
Name	typName	May be updated.
Abbreviation	typAbbreviation	May be updated. May be NULL.
TypeCode	typPublicDayTypeCode	'STANDARD' or 'SERVICE'.
IsDefinedByTransportAuthorityId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUpToDate	typDate	May be NULL. May be updated.

11.2.1 Versioning

A Public Day Type is an updateable object.

11.3 Alternate Public Day Type Name

Column name	Data type	Remark
Id	typId	Unique.
IsForPublicDayTypeId	typId	
Name	typName	May be updated.
Abbreviation	typAbbreviation	May be updated. May be NULL.
LanguageCode	typLanguageCode	Unique in scope of IsForPublicDayTypeId.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUpToDate	typDate	May be NULL. May be updated.

11.3.1 Versioning

An Alternate Public Day Type Name object is an updateable object.

11.4 Operating Day In Calendar

Column name	Data type	Remark
Id	typId	Unique.
OperatingDayDate	typDate	Unique in combination with *IsPartOfServiceCalendarId*.
IsPartOfServiceCalendarId	typId	
IsPublicDayTypeId	typId	May be updated.
NamedDayName	typName	May be updated. NULL or named days such as 'Christmas day'.
LastModifiedUtcDateTime	typDateTime	May be updated.

11.4.1 Versioning

An Operating Day In Calendar has a special versioning that depends on the calendar the object is part of.

For each Service Calendar object there exists one Operating Day In Calendar for each date in that Calendar. These objects may be modified as updateable objects.

However, they do NOT have columns for *ExistsFromDate* and *ExistsUptoDate*. Instead, they are created and deleted according to changes in the validity of the Calendar. If a Calendar is extended one week, there will be seven new Operating Day In Calendar-objects for that Calendar.

11.4.2 Usage note

The information provided in Service Calendar and its sub objects is not sufficient to resolve which dates a VehicleJourney or Block is valid, since there might be individual exceptions from the Service Calendar. The actual mapping of Vehicle Journey to Operating Day is presented in the view Dated Vehicle Journey.

11.5 Alternate Named Day Name

Column name	Data type	Remark
Id	typId	Unique.
IsForOperatingDayInCalendarId	typId	
Name	typName	May be updated.
Abbreviation	typAbbreviation	May be updated. May be NULL.
LanguageCode	typLanguageCode	Unique in scope of IsForOperatingDayInCalendarId
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUpToDate	typDate	May be NULL. May be updated.

11.5.1 Versioning

An Alternate Named Day Name object is an updateable object.

12 Dated Timetable

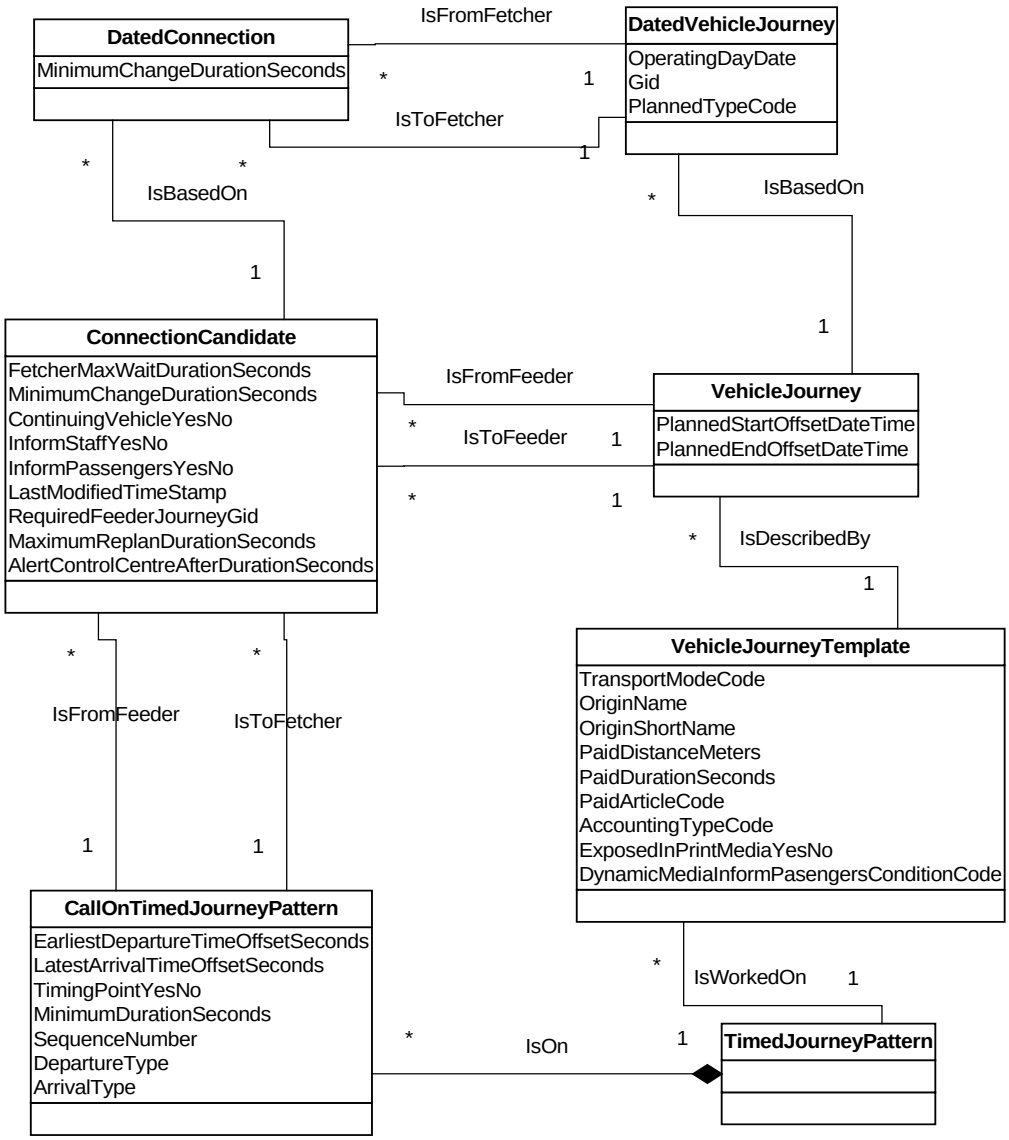

Fig. 7. The central relationships in dated timetable.

Dated timetables are a result of the so-called date-expansion process in PubTrans. This process expands the data according to timetable versions, the calendar, day types and any individual operating day exceptions. There will be one record for each date that a Vehicle Journey is valid in the Dated Vehicle Journey view.

The dated data gives an easy and straight access to which objects are valid for a specific day or period of days. By using the dated data, you do not need to access the intrinsic details of the validity of the timetable.

12.1 Dated Vehicle Journey

This view contains one record per Vehicle Journey and date of operation. Service Journeys and Dead Runs are two different kinds of Vehicle Journeys.

Column name	Data type	Remark
Id	typId	Unique.
OperatingDayDate	typDate	
Gid	typGid	ServiceJourneyGid or DeadRunGid. Unique in combination with *OperatingDayDate* when *IsReplacedById* is NULL.
IsBasedOnVehicleJourneyId	typId	
PlannedTypeCode	typPlannedTypeCode	Always 'NORMAL'.
IsBasedOnVehicleJourneyTemplateId	typId	Redundant data provided to simplify joins.
UsesTimedJourneyPatternId	typId	Redundant data provided to simplify joins.
UsesDestinationPatternId	typId	Redundant data provided to simplify joins. May be updated
UsesServiceRequirementPatternId	typId	Redundant data provided to simplify joins.May be updated.
IsReplacedById	typId	May be NULL. NULL if this object is valid, else zero or replacing object's id. May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.

12.1.1 Recommendations

This view is suitable for several purposes:

- Version management of timetable data. The view describes which Vehicle Journey version (Template) that is valid for each date.
- Data extraction. The view is ideal to find all valid Vehicle Journey versions for an arbitrary period of days.
- Data comparison. With data from this view it is easy to find if an existing version of a Vehicle Journey is valid for a specific date or not.

12.1.2 Example

The date and time of the planned start of the Dated Vehicle Journey is calculated by adding the value of *PlannedStartOffsetDateTime* to the value of *OperatingDayDate*. The same applies for calculation of the date and time of the planned end of the Dated Vehicle Journey. The following query is an example of how to calculate the planned start date/time:

```
SELECT @PlannedStartDateTime =
DVJ.OperatingDayDate + VJ.PlannedStartOffsetDateTime
FROM DatedVehicleJourney as DVJ
INNER JOIN VehicleJourney as VJ
ON DVJ.IsBasedOnVehicleJourneyId = VJ.Id
WHERE DVJ.Id = @RequestedId
```

12.1.3 Versioning

A Dated Vehicle Journey is a dated state object.

Versioning is marked in the column *IsReplacedById* as follows:

- NULL indicates that the object is valid for the specified *OperatingDayDate*.
- NOT NULL indicates that the object is invalid for the specific *OperatingDayDate*.

Please note, that currently NOT NULL is equal to zero, but other values could be used in the future, pointing out the object id of the object that has replaced this object. To be compatible with future changes, the test for valid objects must be WHERE *IsReplacedById* IS NULL and the test for invalid objects must be WHERE *IsReplacedById* IS NOT NULL. In this latter case, do not test for equality with zero.

12.2 Dated Connection

This view shows Dated Connections resulting from pairs of Dated Vehicle Journeys matching a ChangeFrom rule.

Column name	Data type	Remark
Id	typId	Unique.
IsBasedOnConnectionCandidateId	typId	
IsFromFeederDatedVehicleJourneyId	typId	
IsToFetcherDatedVehicleJourneyId	typId	
MinimumChangeDurationSeconds	typDurationSeconds	May be NULL. See usage notes.
IsReplacedById	typId	May be NULL. NULL if this object is valid, else zero or replacing object's *Id*. May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.

12.2.1 Usage notes

There might be a Connection between two Journeys that are dated on different dates. I.e. the fetcher Journey might be defined as starting 5 minutes after midnight on day 2, while the feeder Journey is defined as starting 23:55 on day 1. The two Journeys will then have different values in the *OperatingDayDate* field in the view Dated Vehicle Journey.

MinimumChangeDurationSeconds will be evaluated at date-expansion according to configuration using data from field *DefaultDurationSeconds* in view Connection Link, *DefaultInterchangeDurationSeconds* in view Stop Area and explicit information defined in imported data.

12.2.2 Versioning

A Dated Connection is a dated state object.

Versioning is marked in the column *IsReplacedById* as follows:

- NULL indicates that the object is valid.
- NOT NULL indicates that the object is invalid.

Please note, that currently NOT NULL is equal to zero, but other values could be used in the future, pointing out the object id of the object that have replaced this object. To be compatible with future changes, the test for valid objects must be WHERE *IsReplacedById* IS NULL and the test for invalid objects must be WHERE *IsReplacedById* IS NOT NULL. In this latter case, do not test for equality with zero.

13 Dated Vehicle Schedule

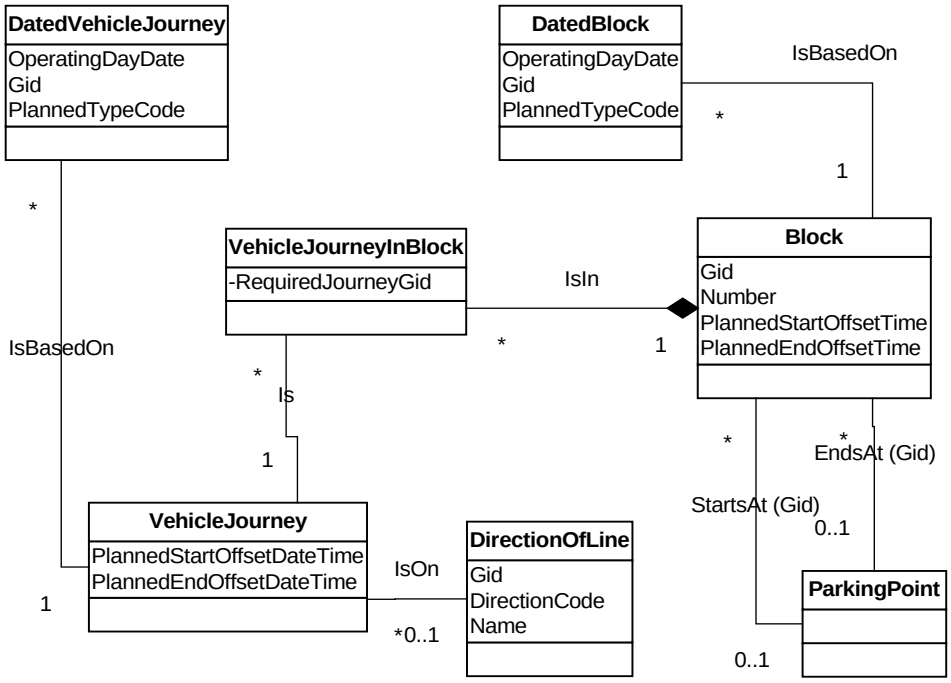

Fig. 8. The central relationships in Dated Vehicle Schedule.

Dated Vehicle Schedules are a result of the so-called date-expansion process in PubTrans. This process expands the data according to schedule versions, the Calendar, Day Types and any individual Operating Day Exceptions. There will be one record for each date that a Block is valid in the Dated Block view.

The dated data gives an easier and straighter access to which objects are valid for a specific day or period of days. By using the dated data, you do not need to access the intrinsic details of the validity of the schedule.

13.1 Dated Block

This view contains one record per Block and date of operation.

Column name	Data type	Remark
Id	typId	Unique.
OperatingDayDate	typDate	
Gid	typGid	BlockGid. Unique in combination with *OperatingDayDate* when *IsReplacedById* is NULL.
IsBasedOnBlockId	typId	

Column name	Data type	Remark
IsReplacedById	typId	May be NULL. NULL if this object is valid, else zero or replacing object's id. May be updated.
PlannedTypeCode	typPlannedTypeCode	Always 'NORMAL'.
LastModifiedUtcDateTime	typDateTime	May be updated.

13.1.1 Usage notes

PubTrans does not require Block data. As a consequence, block data may not be available for all Vehicle Journeys. There are related usage notes in 17.2 Vehicle Journey In Block.

13.1.2 Example

The date and time of the planned start of the Dated Block is calculated by adding the value of *PlannedStartOffsetDateTime* to the value of *OperatingDayDate*. The same applies for calculation of the date and time of the planned end of the DatedBlock. The following query is an example of how to calculate the planned start date/time:

```
SELECT @PlannedStartDateTime =
DatedBlock.OperatingDayDate + Block.PlannedStartOffsetDateTime
FROM DatedBlock
INNER JOIN Block
ON DatedBlock.IsBasedOnBlockId = Block.Id
WHERE DatedBlock.Id = @RequestedId
```

13.1.3 Versioning

A Dated Block is a dated state object.

Versioning is marked in the column *IsReplacedById* as follows:

- NULL indicates that the object is valid for the specified *OperatingDayDate*.
- NOT NULL indicates that the object is invalid for the specific *OperatingDayDate*.

Please note, that currently NOT NULL is equal to zero, but could be other values in the future, pointing out the object *Id* of the object that have replaced this object. To be compatible with future changes, the test for valid objects must be WHERE *IsReplacedById* IS NULL and the test for invalid objects must be WHERE *IsReplacedById* IS NOT NULL. In this latter case, do not test for equality with zero.

14 Lines in detail

This chapter describes the views that contain information about Lines in detail such as the Journey Pattern.

14.1 Journey Pattern

This view contains general information about Journey Patterns without regard to the timing, destination or service requirement aspect. A Journey Pattern is an ordered list of the Journey Pattern Points passed during a Journey. The aspect of alighting and boarding restrictions at the Journey Pattern Points is integral. Thus, two Journeys share the same Journey Pattern only if passengers can alight and board in the same manner in both Journeys.

A Journey Pattern is either a service Journey Pattern or a Dead Run Pattern.

Column name	Data type	Remark
Id	typId	Unique.
PointCount	typCount	Number of Points in the Journey Pattern. Redundant data provided to simplify queries.
IsOnDirectionOfLineId	typId	May be NULL. NULL only if Dead Run Pattern.
StartsAtJourneyPatternPointGid	typGid	Redundant data provided to simplify queries.
EndsAtJourneyPatternPointGid	typGid	Redundant data provided to simplify queries.

14.1.1 Usage note

The order that each Point in the Journey Pattern is passed can be found by joining with the view Point In Journey Pattern.

Details about the points can be found by joining the view Point In Journey Pattern with the view Journey Pattern Point. A record in Point In Journey Pattern has one field called *IsInJourneyPatternId*, which refers to a Journey Pattern, *Id*, in this view, and another field called *IsJourneyPatternPointGid*, that refers to the information in Journey Pattern Point. By joining these views, using the given keys, one can get information on all Stop Points and Via Points in a Journey Pattern.

14.1.2 Example

The following query lists all valid Journey Patterns for a requested date (note that the query is using views described in subsequent chapters):

```
SELECT
  JP.*
FROM
  JourneyPattern AS JP
INNER JOIN
  TimedJourneyPattern AS TJP
ON
  JP.Id = TJP.IsBasedOnJourneyPatternId
INNER JOIN
  VehicleJourneyTemplate AS VJT
ON
  TJP.Id = VJT.IsWorkedOnTimedJourneyPatternId
INNER JOIN
  DatedVehicleJourney AS DVJ
ON
  VJT.Id = DVJ.IsBasedOnVehicleJourneyTemplateId
WHERE
  DVJ.OperatingDayDate = @RequestedDate AND DVJ.IsReplacedById IS NULL
```

14.1.3 Versioning

A Journey Pattern is an object without explicit versioning.

14.2 Point In Journey Pattern

This view is the link between the Journey Pattern and its Journey Pattern Points.

Column name	Data type	Remark
Id	typId	Unique.
IsInJourneyPatternId	typId	
IsJourneyPatternPointGid	typGid	
SequenceNumber	typSequenceNumber	Sequence within JourneyPattern.
DepartureType	typDepartureType	See data types for explanation.
ArrivalType	typArrivalType	See data types for explanation.

14.2.1 Usage note

IsInJourneyPatternId refers to a Journey Pattern *Id*, and *IsJourneyPatternPointGid* refers to a Journey Pattern Point *Gid* (see view Journey Pattern Point for more information). Please note that joining this view with the view Journey Pattern Point requires that the lifetime of the Journey Pattern Point be considered.

SequenceNumber describes the order that Journey Pattern Points should be passed. Sorting the Point In Journey Pattern view by *SequenceNumber* for a selected Journey Pattern, the result will be that all Journey Pattern Points, *IsJourneyPatternPointGid*, in the Journey Pattern are presented in the order they are to be called.

The type of departure and arrival must be examined using the fields *DepartureType* and *ArrivalType*.

14.2.2 Example

The following query is an exemple of how to get information about Journey Pattern Points that are valid for a certain Journey Pattern (@RequestedJourneyPatternId) at a specific point in time (@RequestedDate):

```
SELECT
  JPP.*
FROM
  JourneyPatternPoint AS JPP
INNER JOIN
  PointInJourneyPattern AS PJP
ON
  JPP.Gid = PJP.IsJourneyPatternPointGid
WHERE
  PJP.IsInJourneyPatternId = @RequestedJourneyPatternId
AND
  ( (JPP.ExistsUptoDate IS NULL) OR (JPP.ExistsUptoDate >
  @RequestedDate) )
AND
  JPP.ExistsFromDate <= @RequestedDate
```

14.2.3 Versioning

A Point In Journey Pattern is an object without explicit versioning.

14.3 Named Journey Pattern

This view presents a combination of a Journey Pattern and a Destination Pattern known under a reference name in a period of time.

Column name	Data type	Remark
Id	typId	Unique.
ReferenceName	typName	Unique in combination with *IsOnDirectionOfLineId* and *IsInScopeOfContractorId* at a specific point in time.
Gid	typGid	Unique at a specific point in time. Abstract Gid.
IsOnDirectionOfLineId	typId	Is NULL if DeadRunPattern, otherwise not NULL. Redundant information from referenced JourneyPattern.
IsInScopeOfContractorId	typId	Is NULL if ServiceJourneyPattern, otherwise NOT NULL.
IsJourneyPatternId	typId	
HasDestinationPatternId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	

Column name	Data type	Remark
ExistsUptoDate	typDate	May be NULL. May be updated.

14.3.1 Usage note

As the *Gid* is an abstract Gid, no meaningful information can be extracted by disassembling it. The *ReferenceName* on the other hand could contain some meaningful information for a human reader.

It is guaranteed that the value in attribute *IsOnJourneyPatternId* in the related views Service Requirement and Destination Pattern will match the value in attribute *IsJourneyPatternId* in this view.

14.3.2 Versioning

A Named Journey Pattern is a period state object.

15 Timetable

This publication contains timetable information.

The versioning of the timetable is maintained internally and automatically by PubTrans. It means that, as long as the object's data does not change, there will be only one version of an object in DOI:

- Regardless of how many times the object is imported into PubTrans.
- Regardless of how many source systems that the object is imported from.
- Regardless of how many times the timetable is changed.

To find out which set of objects that are valid for a specific day or period of days, the Vehicle Journey Template view must be joined with the view Dated Vehicle Journey in the publication Dated Timetable.

15.1 Vehicle Journey

Column name	Data type	Remark
Id	typId	Unique.
PlannedStartOffsetDateTime	typOffsetDateTime	
PlannedEndOffsetDateTime	typOffsetDateTime	
IsDescribedByVehicleJourneyTemplateId	typId	

15.1.1 Versioning

A Vehicle Journey is an object without explicit versioning.

15.2 Advance Order Condition

This view contains additional information for Journeys that must be ordered in advance.

Column name	Data type	Remark
Id	typId	Unique.
IsAppliedToVehicleJourneyId	typId	
IsControlledByOrganisationalUnitId	typId	May be NULL.
MinimumDaysInAdvanceCount	typCount	May be NULL.
LatestAbsoluteTime	typTime	May be NULL.
LatestTimeSpanInAdvanceDurationSeconds	typDurationSeconds	May be NULL.
PublicYesNo	typYesNo	May be NULL.
PublicNote	typNote	May be NULL.
InternalNote	typNote	May be NULL.

15.2.1 Usage note

MinimumDaysInAdvanceCount and *LatestAbsoluteTime* should be handled as a pair. If *LatestTimeSpanInAdvanceDurationSeconds* is not NULL the most restrictive criterion applies.

15.2.2 Versioning

An Advance Order Condition object is versioned by the owning Vehicle Journey object, thus an Advance Order Condition is an object without explicit versioning.

15.3 Vehicle Journey Template

Column name	Data type	Remark
Id	typId	Unique.
IsWorkedOnTimedJourneyPatternId	typId	
UsesNamedJourneyPatternGid	typGid	
IsWorkedOnDirectionOfLineGid	typGid	May be NULL. Is NULL for Dead Runs. Redundant data provided to simplify joins.
TransportModeCode	typTransportModeCode	Deducted at time of import from default value for line if not specifically stated for the service journey.
ContractorGid	typGid	May be NULL.
OriginName	typName	May be NULL.
OriginShortName	typShortName	May be NULL.
OriginPlaceGid	typGid	May be NULL.
StartsAtJourneyPatternPointGid	typGid	Redundant data provided to simplify queries.
EndsAtJourneyPatternPointGid	typGid	Redundant data provided to simplify queries.
IsWorkedAccordingToServiceCalendarId	typId	
PaidDistanceMeters	typMeters	May be NULL.
PaidDurationSeconds	typDurationSeconds	May be NULL.
PaidArticleCode	typCode	May be NULL. Code for paid Vehicle type.
IsPaidAccordingToContractId	typId	May be NULL.
AccountingTypeCode	typAccountingType-Code	May be NULL.

Column name	Data type	Remark
ExposedInPrintMediaYesNo	typYesNo	May be NULL.
DynamicMediaInformPassengersCondition-Code	typInformPassengers-ConditionCode	May be NULL.
UsesServiceRequirementPatternId	typId	

15.3.1 Example

To find out the Vehicle Journey Template information for a specific Dated Vehicle Journey, one could use the following query:

```
SELECT
  VJT.*
FROM
  DatedVehicleJourney AS DVJ
INNER JOIN
  VehicleJourneyTemplate AS VJT
ON
  DVJ.IsBasedOnVehicleJourneyTemplateId = VJT.Id
WHERE
  DVJ.Id = @RequestedDatedVehicleJourneyId
```

15.3.2 Notes on references to destination

It is possible to use *UsesNamedJourneyPatternGid* in combination with a date selection criterion and join with view NamedJourneyPattern and then join with the view DestinationPattern using *HasDestinationPatternId* to find the details about destinations.

In some situations it will be easier to bypass this view and instead join view Dated Vehicle Journey with Destinatation Pattern directly using *UsesDestinationPatternId*

15.3.3 Versioning

A Vehicle Journey Template is an object without explicit versioning.

15.4 Vehicle Journey Note

This view contains additional information for Vehicle Journeys that can not be expressed in other data.

Column name	Data type	Remark
Id	typId	Unique.
IsForVehicleJourneyTemplateId	typId	
IsApplicableFromPointInJourneyPatternId	typId	May be NULL. If NULL applicable from first point in journey pattern.
IsApplicableToPointInJourneyPatternId	typId	May be NULL. If NULL applicable to last point in journey pattern.

Column name	Data type	Remark
ConcernsArrivalsYesNo	typYesNo	
ConcernsDeparturesYesNo	typYesNo	
ExposedToStaffYesNo	typYesNo	
PublicYesNo	typYesNo	
ExposedInPrintMediaYesNo	typYesNo	
DynamicMediaInformPassengersCondition-Code	typInformPassengers-ConditionCode	
TextNote	typNote	
ExternalIDName	typName	May be NULL
ExternalSystemIDName	typName	May be NULL.

15.4.1 Versioning

A Vehicle Journey Note is versioned by the owning Vehicle Journey Template object, thus a Vehicle Journey Note is an object without explicit versioning.

15.5 Service Requirement

This view contains a combination of Service Requirements.

Column name	Data type	Remark
Id	typId	Unique.
TransportModeCode	typTransportModeCode	May be NULL.
CapacitySeatingsCount	typCount	May be NULL.
CapacityTotalCount	typCount	May be NULL. Sum of standing and seated.
CapacityPramsCount	typCount	May be NULL.
CapacityWheelchairsCount	typCount	May be NULL.
BusSizeTypeCode	typBusSizeTypeCode	May be NULL.
TrainSizeCarCount	typCount	May be NULL.
TrainSizeShortYesNo	typYesNo	May be NULL.
RequiredEmissionLevelCode	typEmissionLevelCode	May be NULL.

Column name	Data type	Remark
RequiredFuelTypeCode	typFuelTypeCode	May be NULL.
AccessibilityLowEntranceYesNo	typYesNo	May be NULL.
AccessibilityLiftOrRampYesNo	typYesNo	May be NULL.
AccessibilityLowFloorYesNo	typYesNo	May be NULL.

15.5.1 Usage note

Usally the same combination of Service Requirements applies for the whole Journey, however PubTrans supports changing Service Requirements along the Route. This is accomplished by having references to different Service Requirements associated with the different Points in the Journey Pattern. The valid combinations are exposed in the view PointInServiceRequirementPattern.

15.5.2 Versioning

A Service Requirement is an object without explicit versioning.

15.6 Timed Journey Pattern

A Timed Journey Pattern is a Journey Pattern with a unique timing pattern. Timed Journey Patterns tend to be very repetitive on Lines with high frequency and over time. The use of timing patterns offers a high degree of data compression and data transfer optimisation.

Column name	Data type	Remark
Id	typId	Unique.
IsBasedOnJourneyPatternId	typId	

15.6.1 Usage note

This view refers to a Journey Pattern Id by the field IsBasedOnJourneyPatternId. Arrival times and departure times for the Timed Journey Pattern can be retrieved using the Call On Timed Journey Pattern view. Each record in Call On Timed Journey Pattern has a field called IsOnTimedJourneyPatternId, which refers to an Id in this view.

15.6.2 Versioning

A Timed Journey Pattern is an object without explicit versioning.

15.7 Call On Timed Journey Pattern

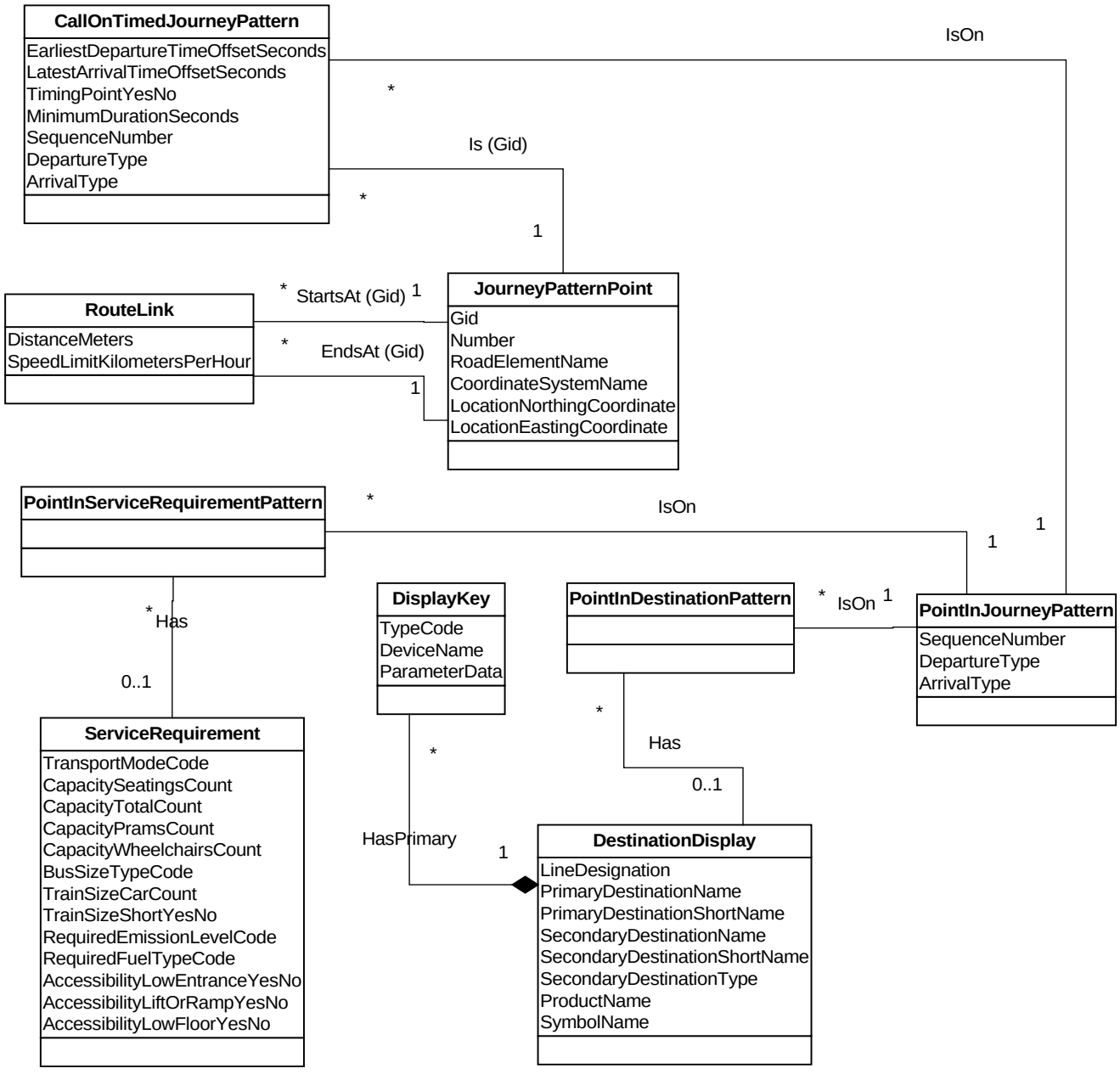

Fig. 9. Call On Timed Journey Pattern

Each object in this view represents a call on a Timed Journey Pattern. Timing data is relative the start of the Journey.

Column name	Data type	Remark
Id	typId	Unique.

Column name	Data type	Remark
IsOnTimedJourneyPatternId	typId	
IsOnPointInJourneyPatternId	typId	
EarliestDepartureTimeOffsetSeconds	typOffsetSeconds	The earliest permitted scheduled time of departure. Used as scheduled departure time if *DepartureType* is a type that should be presented to the public. The time is resolved by adding this offset value to *PlannedStartDateTime* of the Vehicle Journey.
LatestArrivalTimeOffsetSeconds	typOffsetSeconds	The latest arrival time without considering it as late. Used as scheduled arrival time if *ArrivalType* is a type that should be presented to the public. The time is resolved by adding this offset value to *PlannedStartDateTime* of the Vehicle Journey.
TimingPointYesNo	typYesNo	
MinimumDurationSeconds	typDurationSeconds	May be NULL.
IsJourneyPatternPointGid	typGid	Redundant data provided to simplify joins.
SequenceNumber	typSequenceNumber	Redundant data provided to simplify joins.
DepartureType	typDepartureType	Redundant data provided to simplify joins.
ArrivalType	typArrivalType	Redundant data provided to simplify joins.

15.7.1 Usage note

IsOnTimedJourneyPatternId refers to the Timed Journey Pattern it is included in.

The *IsOnPointInJourneyPatternId* field should be matched with *Id* in Point In Journey Pattern to find the related Journey Pattern Point.

The planned departure- and arrival time fields *EarliestDepartureTimeOffsetSeconds* and *Latest-ArrivalTimeOffsetSeconds* are relative the start of the Timed Journey Pattern and thus from the start time of the Vehicle Journey.

The columns *IsJourneyPatternPointGid, SequenceNumber, DepartureType* and *ArrivalType* are redundant and hold data identical to data residing in Point In Journey Pattern. It is thus possible to avoid joining Point In Journey Pattern in some queries.

The *MinimumDurationSeconds* could be used to enhance forecasts.

15.7.2 Versioning

A Call On Timed Journey Pattern is an object without explicit versioning.

15.8 Destination Pattern

Column name	Data type	Remark
Id	typId	Unique.
IsOnJourneyPatternId	typId	

15.8.1 Versioning

A Destination Pattern is an object without explicit versioning.

15.9 Point In Destination Pattern

A Point In Destination Pattern contains a reference to the Destination Display and the Display Keys that apply at a Point in Journey Pattern.

Column name	Data type	Remark
Id	typId	Unique.
IsInDestinationPatternId	typId	Unique in combination with *IsOnPointInJourneyPatternId*.
IsOnPointInJourneyPatternId	typId	Unique in combination with *IsInDestinationPatternId*.
HasDestinationDisplayId	typId	May be NULL.

15.9.1 Usage note

To evaluate if a record from this view is valid on a certain date *IsInDestinationPatternId* should match *UsesDestinationPatternId* in the view Dated Vehicle Journey for the concerned Vehicle Journey.

The *IsOnPointInJourneyPatternId* field should be used to find the related record in the view Call On Timed Journey Pattern.

It is guaranteed that the value of the *IsOnPointInJourneyPatternId* attribute is consistent with the value in the *IsOnJourneyPatternId* attribute of the relevant Destination Pattern.

15.9.2 Versioning

A Point In Destination Pattern is an object without explicit versioning.

15.10 Service Requirement Pattern

Column name	Data type	Remark
Id	typId	Unique.
IsOnJourneyPatternId	typId	

15.10.1 Versioning

A Service Requirement Pattern is an object without explicit versioning.

15.11 Point In Service Requirement Pattern

A Point In Service Requirement Pattern contains a reference to the Service Requirement that apply at a Point In Journey Pattern.

Column name	Data type	Remark
Id	typId	Unique
IsInServiceRequirementPatternId	typId	Unique in combination with *IsOnPointInJourneyPatternId*.
IsOnPointInJourneyPatternId	typId	Unique in combination with *IsInServiceRequirementPatternId*.
HasServiceRequirementId	typId	May be NULL.

15.11.1 Usage note

To evaluate if a record from this view is valid on a certain date *IsInServiceRequirementPatternId* should match *UsesServiceRequirementPatternId* in the view Dated Vehicle Journey for the concerned Service Journey.

The *IsOnPointInJourneyPatternId* field should be used to find the related record in the view Call On Timed Journey Pattern.

It is guaranteed that the value of the *IsOnPointInJourneyPatternId* attribute is consistent with the value in the *IsOnJourneyPatternId* attribute of the relevant Service Requirement Pattern.

15.11.2 Versioning

A Point In Service Requirement Pattern is an object without explicit versioning.

15.12 Destination Display

Column name	Data type	Remark
Id	typId	Unique.
LineDesignation	typLineDesignation	
PrimaryDestinationName	typName	
PrimaryDestinationShortName	typShortName	May be NULL.
HasPrimaryDestinationPlaceGid	typGid	May be NULL.
SecondaryDestinationName	typName	May be NULL.
SecondaryDestinationShortName	typShortName	May be NULL.
HasSecondaryDestinationPlaceGid	typGid	May be NULL.

Column name	Data type	Remark
SecondaryDestinationType	typSecondary-DestinationType	May be NULL.
ProductName	typName	May be NULL.
SymbolName	typName	May be NULL.

15.12.1 Using Destination Display

PubTrans supports alternating destination texts along the Route. The most practical use is to replace or remove so called via-destinations when they become invalid (i.e. when they are passed). If a vehicle system controls the destination signs of the vehicle, it can automatically change destination texts during a Vehicle Journey by using the data provided in the view Point In Destination Pattern and view Destination Display.

15.12.2 Note

The columns LineDesignation, PrimaryDestinationName, PrimaryDestinationShortName, SecondaryDestinationName, SecondaryDestinationShortName and ProductName are deducted at time of timetable import. LineDesignation defaults to Line.Designation if not specifically stated. Primary- and Secondary-DestinationName defaults to relevant Place.Name if not specifically stated. ProductName defaults to GroupOfLines.Name where PurposeOfLineGroupingCode='PRODUCT' if not specifically stated.

15.12.3 Versioning

A Destination Display is an object without explicit versioning.

15.13 Display Key

Display Keys are used as activation keys for (overriding) display and announcement related activities for a Stop on a Journey Pattern. There can be multiple keys with the same type code to control different kinds of hardware. These keys can override related Stop Point keys for the Stop.

Column name	Data type	Remark
Id	typId	Unique.
IsInDestinationDisplayId	typId	
TypeCode	typDisplayKeyType-Code	
DeviceName	typName	I.e. INIT_PA_INTERN, INIT_PA_EXTERN, INIT_HEADSIGN
ParameterData	typParameterData	

15.13.1 Versioning

A Display Key is an object without explicit versioning

15.14 Connection Candidate

Column name	Data type	Remark
Id	typId	Unique.
IsFromFeederVehicleJourneyId	typId	
IsFromFeederCallOnTimedJourneyPatternId	typId	
IsToFetcherVehicleJourneyId	typId	
IsToFetcherCallOnTimedJourneyPatternId	typId	
FetcherMaxWaitDurationSeconds	typDurationSeconds	
ContinuingVehicleYesNo	typYesNo	
InformStaffYesNo	typYesNo	
InformPassengersYesNo	typYesNo	
MaximumReplanDurationSeconds	typDurationSeconds	May be NULL.
MinimumChangeDurationSeconds	typDurationSeconds	May be NULL. Only supplied if an explicit value was provided at import.
AlertControlCentreAfterDurationSeconds	typDurationSeconds	May be NULL.
RequiredFeederJourneyGid	typGid	May be NULL. Only supplied if an explicit value was provided at import.

15.14.1 Usage notes

If *MaximumReplanDurationSeconds* is set to NULL, then the intention is that no automatic replanning of interchange should take place if the original interchange fails.

If the actual wait time during real time operation exceeds the *AlertControlCentreAfterDurationSeconds* value, control centre should act. This is usually only of interest in combination with extensive wait periods expressed in *FetcherMaxWaitDurationSeconds*.

16 Vehicle Schedule

16.1 Block

Column name	Data type	Remark
Id	typId	Unique.
Gid	typGid	
Number	typBlockNumber	
PlannedStartOffsetDateTime	typOffsetDateTime	
PlannedEndOffsetDateTime	typOffsetDateTime	
StartsAtParkingPointGid	typGid	The Parking Point's *JourneyPatternPointGid*. May be NULL.
EndsAtParkingPointGid	typGid	The Parking Point's *JourneyPatternPointGid*. May be NULL.
IsDefinedByContractorId	typId	
TaskCount	typCount	Number of Vehicle Journeys in Block. Redundant data provided to simplify queries.

16.1.1 Versioning

A Block is an object without explicit versioning.

16.1.2 Usage notes

PubTrans does not require block data. As a consequence, block data may not be available for all Vehicle Journeys.

16.2 Vehicle Journey In Block

This view links Vehicle Journeys with Blocks.

Column name	Data type	Remark
Id	typId	Unique.
IsInBlockId	typId	
IsVehicleJourneyId	typId	
RequiredJourneyGid	typGid	May be NULL.
SequenceNumber	typSequenceNumber	Sequence within Block.

16.2.1 Usage notes

This view can be used as a basis to match Dated Blocks with relevant Dated Vehicle Journeys. See the UML-diagram in chapter 14 for illustration of the basic relationships involved.

One consideration is that Dated Vehicle Journeys must be dated on the same date as the Dated Block they are part of; therefore only those records in the view Dated Vehicle Journey that have the same value in the *OperatingDayDate* field as the intended Dated Block are relevant. Additionally, if the *RequiredJourneyGid* field is not NULL, it must be taken into consideration. Normally the *RequiredJourneyGid* field will be set to NULL. If not, then it is an additional requirement that *RequiredJourneyGid* matches the Gid of the Dated Vehicle Journey. If they do not match, then the Dated Vehicle Journey is not part of the Dated Block.

16.2.2 Example

To find out which Dated Vehicle Journeys that actually are in a DatedBlock, one could use the following query:

```
SELECT
  DVJ.*
FROM
  DatedVehicleJourney AS DVJ
INNER JOIN
  VehicleJourneyInBlock AS VJIB
ON
  VJIB.IsVehicleJourneyId = DVJ.IsBasedOnVehicleJourneyId
AND
  (VJIB.RequiredJourneyGid IS NULL OR VJIB.RequiredJourneyGid = DVJ.Gid)
INNER JOIN
  DatedBlock AS DB
ON
  DB.IsBasedOnBlockId = VJIB.IsInBlockId
WHERE
  DB.Id = @RequestedDatedBlockId
AND
  DVJ.DatedOnDate = DB.DatedOnDate
AND
  DVJ.IsReplacedById IS NULL
AND
  DB.IsReplacedById IS NULL
```

16.2.3 Versioning

A Vehicle Journey In Block is an object without explicit versioning.

16.3 Task In Block

Column name	Data type	Remark
Id	typId	Unique.
IsInBlockId	typId	
SequenceNumber	typSequenceNumber	

Column name	Data type	Remark
VehicleJourneyDefinitionId	typId	May be NULL
VehicleJourneyGid	typId	May be NULL, Definition and Gid are never both NULL. I case both are not NULL the Gid is a further restriction on the matching of the Definition. i.e the Task In Block was defined using a VehicleJourneyDefinition with a required vehicle journey number.

16.3.1 Versioning

A Task In Block is an object without explicit versioning.

17 Vehicle Journey Definition

A Vehicle Journey Definition is uniquely defined by the Id column or all the other columns. i.e. there will only be one id for each combination of the other columns. A Task In Block with a Vehicle Journey Definition Id can only match Vehicle Journeys with the same Vehicle Journey Definition Id, there is no need to check all the individual columns.

17.1 Vehicle Journey Definition

Column name	Data type	Remark
Id	typId	Unique.
TypeCode	typVehicleJourneyTypeCode	
FirstCallDepartureTimeOffsetSeconds	typOffsetSeconds	Offset from the OperatingDayDate of the Vehicle Journey
FirstCallJourneyPatternPointGid	typGid	
LastCallArrivalTimeOffsetSeconds	typOffsetSeconds	Offset from the OperatingDayDate of the Vehicle Journey
LastCallJourneyPatternPointGid	typGid	
DirectionOfLineId	typId	*TypeCode*: SERVICE: Never NULL DEADRUN:Always NULL
ContractorId	typId	*TypeCode*: SERVICE: Always NULL DEADRUN: Never NULL

17.1.1 Versioning

A Vehicle Journey Definition is an object without explicit versioning.

17.2 Definition For Vehicle Journey

Extension of the Vehicle Journey view providing the Vehicle Journey Definition Id for each Vehicle Journey. There is a definition for each Vehicle Journey.

Column name	Data type	Remark
VehicleJourneyId	typId	Unique.
VehicleJourneyDefinitionId	typId	

17.2.1 Versioning

A Defintion For Vehicle Journey is an object without explicit versioning.

18 Production resources

This publication contains information about vehicles and garages.

18.1 Vehicle

This view contains general data about the Vehicle.

Column name	Data type	Remark
Id	typId	Unique.
GloballyUniqueName	typName	Unique. Used for identification. Often same as *ChassisName*.
Name	typName	May be updated. May be NULL.
ChassisName	typName	Chassis number. May be NULL.
ChassisMakeName	typName	May be updated. May be NULL.
ModelMakeName	typName	May be updated. May be NULL.
ModelYear	typYear	May be updated. May be NULL.
ModelTypeName	typName	May be updated. May be NULL.
ClassifiedAsTransportModeCode	typTransportModeCode	
CapacityStandingsCount	typCount	May be updated. May be NULL.
CapacitySeatingsCount	typCount	May be updated. May be NULL.
CapacityPramsCount	typCount	May be updated. May be NULL.
CapacityWheelchairsCount	typCount	May be updated. May be NULL.
AccessibilityLowEntranceYesNo	typYesNo	May be updated. May be NULL.
AccessibilityLiftYesNo	typYesNo	May be updated. May be NULL.
AccessibilityRampYesNo	typYesNo	May be updated. May be NULL.
AccessibilityLowFloorYesNo	typYesNo	May be updated. May be NULL.
SizeLengthMeters	typMeters	Size of Vehicle unit as a whole counting both motorcar and trailer(s). May be updated. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.1.1 Versioning

A Vehicle is an updateable object.

18.2 Bus : Vehicle

A Bus is a Vehicle with some additional data specific to Buses.

Column name	Data type	Remark
Id	typId	Unique.
GloballyUniqueName	typName	Unique. Used for identification. Often same as *ChassisName*.
Name	typName	May be updated. May be NULL.
ChassisName	typName	Chassis number. May be NULL.
ChassisMakeName	typName	May be updated. May be NULL.
ModelMakeName	typName	May be updated. May be NULL.
ModelYear	typYear	May be updated. May be NULL.
ModelTypeName	typName	May be updated. May be NULL.
ClassifiedAsTransportModeCode	typTransportModeCode	
CapacityStandingsCount	typCount	May be updated. May be NULL.
CapacitySeatingsCount	typCount	May be updated. May be NULL.
CapacityPramsCount	typCount	May be updated. May be NULL.
CapacityWheelchairsCount	typCount	May be updated. May be NULL.
AccessibilityLowEntranceYesNo	typYesNo	May be updated. May be NULL.
AccessibilityLiftYesNo	typYesNo	May be updated. May be NULL.
AccessibilityRampYesNo	typYesNo	May be updated. May be NULL.
AccessibilityLowFloorYesNo	typYesNo	May be updated. May be NULL.
SizeLengthMeters	typMeters	Size of Vehicle unit as a whole counting both motorcar and trailer(s). May be updated. May be NULL.
BusSizeTypeCode	typBusSizeTypeCode	May be updated.
RegistrationNumber	typRegistrationNumber	May be updated. May be NULL.
RegistrationDate	typDate	May be updated. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.2.1 Versioning

A Bus is an updateable object.

18.3 Fuel Type For Vehicle

This view describes which different types of fuel a Vehicle can use.

Column name	Data type	Remark
Id	typId	Unique.
IsForVehicleId	typId	
FuelTypeCode	typFuelTypeCode	

18.3.1 Versioning

A Fuel Type For Vehicle object is versioned by the owning Vehicle object, thus a Fuel Type For Vehicle is an object without explicit versioning.

Observe that records could be removed from this view when Vehicle is updated.since Vehicle is an updateable object.

18.4 Static Communication Address For Vehicle

This view contains information of a Communication Address used to communicate with a Vehicle. It will only expose those types of Communication Addresses that does not change often.

Column name	Data type	Remark
Id	typId	Unique.
Name	typName	+4630366600 for example.
ProtocolCode	typProtocolCode	
ProtocolName	typName	'Telephone' for example.
IsInVehicleId	typId	Vehicle that can be reached using this communication address.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.4.1 Examples

If one needs the telephone number of a specific Vehicle one can use the following query:

```
SELECT @ItsPhoneNumber = Name
FROM StaticCommunicationAddressForVehicle
WHERE IsInVehicleId =@VehicleId
AND ProtocolCode = 'PHONE'
```

18.4.2 Versioning

A Static Communication Address For Vehicle is a period state object.

18.5 Vehicle Utilisation

This view contains data related to the ownership of a Vehicle.

Column name	Data type	Remark
Id	typId	Unique.
IsUtilisationOfVehicleId	typId	
IsOwnedByOrganisationId	typId	A Vehicle can only have one owner at a specific point in time.
ContractedInContractReferenceName	typName	May be NULL.
IsContractedByTransportAuthorityId	typId	May be NULL.
IsManagedByOrganisationalUnitId	typId	May be NULL.
HasDefaultParkingPlaceId	typId	Refers to Garage. May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.5.1 Versioning

A Vehicle Utilisation is a period state object.

18.6 Vehicle Reference

This view describes which *Gid*s that can be used to reference a certain Vehicle.

Column name	Data type	Remark
Id	typId	Unique.
IsForVehicleId	typId	
Number	typVehicleNumber	The Vehicle number used by the Contractor. Unique in scope of *IsDefinedForContractorId* at a specific point in time.
Gid	typGid	Vehicle-Gid. Unique at a specific point in time.
IsDefinedForContractorId	typId	
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.6.1 Versioning

A Vehicle Reference is a period state object.

18.7 Garage : [Parking Place]

Note: Garage has a centroid location. A number of Parking Points of type 'GARAGE' can be associated with a Garage.

Column name	Data type	Remark
Id	typId	Unique.
IsOwnedByOrganisationId	typId	
CoordinateSystemName	typName	May be NULL. May be updated.
CentroidNorthingCoordinate	typCoordinate	Northing. May be NULL. May be updated.
CentroidEastingCoordinate	typCoordinate	Easting. May be NULL. May be updated.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.7.1 Versioning

A Garage is an updateable object.

18.8 Parking Point At Garage

This view describes which Parking Points that are located at a Garage.

Column name	Data type	Remark
Id	typId	Unique.
IsAtGarageId	typId	
HasParkingPointGid	typGid	The Parking Point's *JourneyPatternPointGid*. Only Parking Points of type 'GARAGE' are valid.

18.8.1 Versioning

A Parking Point At Garage object is versioned by the owning Garage object, thus a Parking Point At Garage is an object without explicit versioning.

Observe that records could be removed from this view when Garage is updated.since Garage is an updateable object.

18.9 Garage Utilisation

This view contains data about the utilisation of a Garage by a Contractor.

Column name	Data type	Remark
Id	typId	Unique.
Number	typGarageNumber	The internal garage number within the utilising Contractors Organisation. Unique in scope of *IsByContractorId* at a specific point in time.
Name	typName	
IsUtilisationOfGarageId	typId	
IsByContractorId	typId	
IsManagedByOrganisationalUnitId	typId	May be NULL.
LastModifiedUtcDateTime	typDateTime	May be updated.
ExistsFromDate	typDate	
ExistsUptoDate	typDate	May be NULL. May be updated.

18.9.1 Versioning

A Garage Utilisation is a period state object.

19 System Information

This publication contains information and meta-data.

19.1 System Status

This view can be used to monitor the status of the replication. It contains one row after the replication is set up. The row is updated at a frequency that should be agreed on as sufficient to monitor the replication of data.

Column name	Data type	Remark
Id	typId	Unique.
IsDateExpandedUptoDate	typDate	May be NULL. Date expansion process is not completed from this date and onwards.
UpdateIntervalDurationSeconds	typDurationSeconds	Intended timespan between heartbeats. How often this record is supposed to be updated
LastHeartbeatUtcDateTime	typDateTime	May be updated. Time of last heartbeat.

19.1.1 Usage notes

The column *IsDateExpandedUptoDate* will be NULL when the date-expansion is completed. When date-expansion is complete data is available upto the value stated in column *TimetableReleasedForInternalUseUptoDate* in view Transport Authority.

If the *LastHeartbeatDateTime* is significantly earlier than *UpdateIntervalDurationSeconds* before current time then the replication is suspicious.

19.1.2 Versioning

System Status is an object that is not versioned.

19.2 Data Type Enumeration

This view exposes a snapshot of currently approved translations for different enumeration values of system datatypes.

Column name	Data type	Remark
Id	typId	Unique.
DataTypeName	typName	Name of data type according to column "Data type" in chapter 21 Data types.
ValueCode	typCode	Unique in combination with *DataTypeName* and *LanguageCode.* Enumeration value according to value in column "Values" in chapter 21 Data types.

Column name	Data type	Remark
LanguageCode	typLanguageCode	
TranslatedValueDescription	typDescription	May be updated. Exposes currently approved translation of the enumeration value in indicated language.

19.2.1 Usage notes

It is not guaranteed that translations will be available for all enumeration values. Each system using this view must have a fallback strategy how to handle missing enumeration values.

19.2.2 Versioning

Data Type Enumeration is an object that is not versioned. Rows could be updated, added or removed at any time.

19.3 Publication

This view exposes the *Code* and *Number* used to identify a Publication .

Column name	Data type	Remark
Number	typPublicationNumber	Unique.
Code	typCode	Unique. E.g. PPZ
Name	typName	E.g. Points Places and Zones

19.3.1 Versioning

Publication is not versioned. See versioning note on Key Variant Type for further details.

19.4 Object Type

This view exposes the *Code* and *Number* used to identify an Object Type (a view) .

Column name	Data type	Remark
Number	typObjectTypeNumber	Unique.
Code	typCode	Unique. E.g. SA
Name	typName	E.g. StopArea

19.4.1 Versioning

Object Type is not versioned. See versioning note on Key Variant Type for further details.

19.5 Value Type

This view exposes the *Id* used to identify a Value Type and also how it restricts possible values for an extra attribute of this type.

Column name	Data type	Remark
Id	typId	Unique.
Name	typName	Unique. E.g. ContractorCode.
BaseDataType	typBaseDataType	Name of the base type restricting the value type, e.g. varchar(8). May be updated.
RegularExpression	typRegExString	A regular expression restricting permitted values of this value type. May be updated. May be NULL.
Description	typDescription	May be updated.

19.5.1 Versioning

Value Type is not versioned. See versioning note on Key Variant Type for further details.

19.6 Variant Type

This view exposes the *Id* and *Code* used to identify a Variant Type.

Column name	Data type	Remark
Id	typId	Unique.
Code	typCode	Unique. A typical usage is a language code according to ISO 639-1. E.g. sv.
Name	typName	E.g. Svenska

19.6.1 Versioning

Variant Type is not versioned. See versioning note on Key Variant Type for further details.

19.7 Key Type

This view exposes which Value Types an object of a certain Object Type can be extended with.

Column name	Data type	Remark
Id	typId	Unique.
ExtendsObjectTypeNumber	typObjectTypeNumber	Which type of objects that uses this key, e.g.StopArea.
IsOfValueTypeId	typId	The type of value. E.g. ContractorCode.
Name	typName	Unique in combination with *ExtendsObjectTypeNumber.* E.g. MaintenanceContractorCode
Description	typDescription	May be updated. E.g. Company responsible for cleaning StopArea shelter..

19.7.1 Versioning

Key Type is not versioned. See versioning note on Key Variant Type for further details.

19.8 Key Variant Type

This view exposes which Variant Types that are currently allowed or required for a certain Key Type.

Column name	Data type	Remark
Id	typId	Unique.
IsForKeyTypeId	typId	
IsVariantTypeId	typId	May be NULL. When NULL no variants exists for this key type.
IsMandatory	typYesNo	May be updated.
InValidFromDateTime	typDateTime	May be updated, but only from NULL. May be NULL. No new key variant values might be imported for this variant from the datetime set.

19.8.1 Versioning

Views in this publication are not versioned and will with the exception of the System Status view rarely be changed. Objects that are referred to by other objects will not be removed.

Key Variant Values are validated according to *Key Variant Type*s valid at import time in PubTrans. There is no guarantee that a *Key Variant Type* will continue to be available for import at a later time.

19.9 Publication For Object Type

This view exposes which Object Types (views) that are included in which Publications.

Column name	Data type	Remark
ObjectTypeNumber	typObjectTypeNumber	Unique in combination with PublicationNumber
PublicationNumber	typPublicationNumber	

19.9.1 Versioning

Publication For Object Type is not versioned.

20 Key Variant Value

The Key Variant Value is a view that is included in all publications, except in the *System information* publication.

The same schema definition is used in all publications, and to avoid repeating it in each publication it is instead described separately in this chapter.

The view contains extra attributes that extends various objects. Allowed attributes are defined in the views of the *System information* publication. The Key Variant Value views are located in a separate schema per publication, the schema name for the publication can be found as the *Code* column of the Publication view in the *System Information* publication.

20.1 Key Variant Value

Column name	Data type	Remark
Id	typId	Unique.
IsOfKeyVariantTypeId	typId	
IsForObjectId	typId	
StringValue	typStringKeyValue	

20.1.1 Examples

If one needs the value for a certain key type variant on a specific StopArea, for exemple the Name of a StopArea in a certain language one can use a query like this:

```
SELECT
 V.StringValue
FROM StopArea SA
  INNER JOIN PPZ.KeyVariantValue V ON V.IsForObjectId = SA.Id
  INNER JOIN KeyVariantType KVT ON KVT.Id = V.IsOfKeyVariantTypeId
  INNER JOIN VariantType VT ON VT.Id = KVT.IsVariantTypeId
  INNER JOIN KeyType KT ON KVT.IsForKeyTypeId = KT.Id
WHERE
     SA.Gid  = @RequestedStopAreaGid
 AND VT.Code = @RequestedLanguageVariantTypeCode
 AND KT.Name = @RequestedKeyTypeName
 AND SA.ExistsFromDate <=@RequestedDate
 AND (SA.ExistsUptoDate IS NULL OR SA.ExistsUptoDate >@RequestedDate)
```

20.1.2 Versioning

The Key Variant Value is versioned by the owning object; observe that records could be removed from this view when an updateable object is updated.

21 Data types

Below is a list of the data types used in the views that can be published through the Data Output Interface.

Data type	Values	SQL-type	Remark
typAbbreviation	1-8 characters.	nvarchar(8)	User defined.
typAccountingTypeCode	PAYBYSCH PAYIF_SO DED_IF_N	varchar(8)	Pay by schedule regardless if signed on or not. Pay only if signed on. Deduct if not signed on.
typActionTypeCode	REQ_PRIO REL_PRIO SEND_DEP SEND_ARR SEND_DUE SEND_EXT OTHER	varchar(8)	Request traffic light priority Release traffic light priority Report departure Report arrival Report due Send extra progress report Other action request.
typArrivalType	0 = No Stop. 1 = Stop, but no alighting. 2 = Stop for alighting if necessary. 3 = Always Stop for alighting. 5 = Flexible Stop - end of alighting stretch.	smallint	Only types >= 2 should be presented as arrivals to the public.
typBaseDataType	1-50 characters	varchar(50)	User defined, for example nvarchar(16), INT32 etc
typBridgingDeviceNumber	A 1-9 digit number.	numeric(9, 0)	
typBridgingDeviceTypeCode	DOOR ELEVATOR STAIRWAY ESCALAT CONVEYOR	varchar(8)	Doors, elevators, stairways, escalators and conveyorbelts are different types of devices that can be used to bridge parts of a walking link.
typBlockNumber	1-5 digit number.	numeric(5,0)	

Data type	Values	SQL-type	Remark
typBusSizeTypeCode	 ARTICUL BOGGIE DDECKER MINI SMALL STANDARD	varchar(8)	Code for generic bus sizes. Articulated bus (Swedish - ledbuss) Boggie bus Double decker bus Mini bus Small bus Standard or normal bus
typCode	1-8 uppercase alphanumeric characters.	varchar(8)	User defined alphanumeric short identity of an object.
typComment	1-255 alphanumeric characters.	nvarchar(255)	
typContractorNumber	A 1-4 digit number.	numeric(4, 0)	
typCoordinate	Depends on the location system.	varchar(30)	
typCount	0-999 999 999 999.	numeric(12,0)	
typDate	Any valid date.	datetime	typDate means to ignore the time part of the date time value.
typDateTime	Any valid absolute time, with both date and time component. A left out time component means 00:00:00 that date.	datetime	typDateTime means to use both the date and time part as an absolute Point in time.

Data type	Values	SQL-type	Remark
typDepartureType	0 = No Stop. 1 = Stop, but no boarding. 2 = Stop for boarding if necessary. 3 = Always Stop for boarding. 5 = Flexible Stop – beginning of boarding stretch	smallint	Only types >= 2 should be presented as departures to the public.
typDescription	1-255 characters.	nvarchar(255)	
typDirectionDegrees	Between 0-360 degrees.	numeric(5, 2)	
typDirectionCode	'1' or '2'.	char(1)	A number describing a direction of a JourneyPattern on a Line. A direction can be only one of two: 1 should be interpreted as odd or anti clockwise direction. 2 should be interpreted as even or clockwise directions.
typDisplayKeyTypeCode	O_DESTIN O_S_NAME O_S_MESS ADD_INFO O_C_INFO	varchar(8)	Overriding destination Overriding Stop name Overriding Stop message Additional Stop information Overriding change information
typDurationSeconds	0-2 000 000 000.	int	
typEmployeeNumber	A 1-5 digit number.	numeric(5, 0)	
typEmissionLevelCode	EURO_I EURO_II EURO_III EURO_IV EURO_V EURO_VI	varchar(8)	Emission standard according to European norms.
typEquipmentTypeCode	BENCH RAINSHEL CURB SIGNBOAR	varchar(8)	Bench Rainshelter Curb Signboard

Data type	Values	SQL-type	Remark
typFacilityTypeCode	CARPARK OTHER	varchar(8)	Car park Other
typFuelTypeCode	BATTERY BIOGAS CNG DIESEL DIESELN DIESELL E85 ELECTRIC HYBRIDDB HYDROGEN LPG PETROLL PETROLUL UNKNOWN	varchar(8)	Electric (battery) Biogas Natural gas Diesel Diesel, standard Diesel, light Blend of 85% ethanol and 15% petrol Electric (overhead Line) Hybrid: diesel/battery Hydrogen Light petroleum gas Petrol, leaded Petrol, unleaded
typGarageNumber	0-99999.	numeric(5,0)	
typGender	'M' or 'F'.	varchar(1)	M = male, F = female.
typGid	16-digit number.	numeric(16,0)	Global id for an object across all PubTrans instances. See separate GID-documentation of how this value is to be created and interpreted.
typHyperLinkName	1-1024 characters		
typId	16-digit number.	numeric(16,0)	Id for an object is unique among all Pubtrans instances.
typInformPassengers-ConditionCode	ALWAYS ONLYIF_O ONLYIF_S NEVER	varchar(8)	Describes under which conditions passengers should be informed. Always. Only if the Journey is ordered. Only if the Journey is signed on. Never.
typInitials	1-8 alpha-numeric characters.	nvarchar(8)	

Data type	Values	SQL-type	Remark
typInterchangePriority	1-16	smallint	InterchangePriority is a value in the range 1 - 16 used to influence Journey Planners of which stop that should be chosen for changes between two lines that run in parallel. The value 1 represents the highest priority.
typJourneyPattern-PointNumber	1-99 999 999.	numeric(8, 0)	
typKmPerHour	0-200.	smallint	
typLanguageCode	According to ISO 639.1	varchar(8)	ISO 639.1 Language Codes. For example sv - swedish, da - danish, en - english, nn - nynorsk norwegian and nb - bokmål Norwegian.
typLevelOfOperationCode	ALLOF MOSTOF MANYOF SOMEOF OCCAS	varchar(8)	Defines the level of operation for a Contractor on a Line. ALLOF means the Line is exclusively operated in one Contract. The other values indicate different levels of a non-exclusive operation.
typLineDesignation	A 0-8 character string with printable characters.	nvarchar(8)	The shortest form by which a Line is presented to the public, including suffix.

Data type	Values	SQL-type	Remark
typLineGroupingCode	DISTRICT MARKET PRODUCT TRAFFIC PRESENTA OTHER	varchar(8)	DISTRICT Grouping of lines in geographic districts. Examples: Linjer på/till Värmdö, Linjer i City. MARKET Grouping of lines into administrative marketing groups/responsibilities. Examples: Marknadsområde Norr, City, Söder. PRODUCT Grouping of lines into product identities: Examples: S-BUS, A-BUS, Expressbuss, Flextrafiken, Stombuss. TRAFFIC Grouping of lines into traffic types: Examples: Pendeltåg, Regionaltåg, Tunnelbana, Stadsbuss, Förortsbuss PRESENTA Grouping of lines that should be presented using the name of the Line Group thus overriding Line.Name etcetera. OTHER Grouping of lines for other purposes.
typLineNumber	A 1-4 digit number.	numeric(4, 0)	1-9999. 0 has a special meaning.
typMeters		decimal(12, 2)	The length in meters.
typName	1-50 characters.	nvarchar(50)	Full-length name.
typNote	1-255 characters.	nvarchar(255)	
typObjectTypeNumber		int	
typOccupationCode	DRIVER SERVICE TRA-CONT MANAGER OTHER	varchar(8)	Driver Service staff Traffic controller Manager Other
typOffsetDateTime	A time span relative a predefined “day zero” value.	datetime	This type is used to define relative times, when the time component can be more than 24 hours.

Data type	Values	SQL-type	Remark
typOffsetSeconds	A time offset in seconds.	int	
typOrganisationType	GOVERNME COMPANY COMMUNIT	varchar(8)	Government Company Community
typParameterData	Any text.	varchar(50)	Device specific parameters used to control a certain device
typPassageDirectionCode	UP, DOWN, ANY	varchar(8)	The direction in which a Bridging Device can be passed/must be passed
typPercent	-9999.99 - +9999.99 %.	numeric(8, 4)	
typPlaceNumber	A 1-8 digit number.	numeric(8,0)	0-99999999. 0 has a special meaning.
typPlaceTypeCode	SITE STOPAREA NEIGHBOU ADDRESS STREET OTHER	varchar(8)	Public site Stop area Neighbourhood Address Street Other
typPlannedTypeCode	NORMAL EXTRA	varchar(8)	
typPointOnLinkTypeCode	ACTION MAPPING TIMING	varchar(8)	
typPostcode		nvarchar(10)	
typProtocolCode	PHONE FAX SMS EMAIL PAGER TETRA MOBITEX MRS5000 UNKNOWN	varchar(8)	
typPublicationNumber		int	
typPublicDayTypeCode	STANDARD SERVICE	varchar(8)	
typRegExString	1-255 characters.	varchar(255)	A regular expression string

Data type	Values	SQL-type	Remark
typRegistrationNumber	National variants (use RegExp).	nvarchar(20)	Vehicle's registration number.
typSecondaryDestinationType	E M V T C U	char(1)	The usage of a destination text: E=End station/terminus of Route M=Message V=Via T=Transfer at this station/stop and continue to Primary Destination C=Continue to this end station after transferring at Primary Destination U=Unknown
typSequenceNumber		numeric(12, 0)	
typServiceFunctionCode	BACKUPCO COMPLAIN CONTROLC LOSTANDF SALES_AG	varchar(8)	Backup control centre Complaints office Control centre Lost and found office Sales agent
typShortName	1-16 characters.	nvarchar(16)	Short name.
typStationEntrance-PointLocalNumber	1–99.	numeric(2, 0)	The local station entrance number within the Stop Area.
typStopAreaNumber	1-999999.	numeric(6, 0)	
typStopAreaTypeCode	AIRPORT BUSTERM FERRYBER METROSTN RAILWSTN TRAMSTN SHIPBER TAXITERM UNKNOWN	varchar(8)	Airport, bus terminal, ferry berth, metro station, railway station, tram station, ship berth, taxi terminal or unknown. The primary function of the Stop Area in the public transportation system.
typStopPointKeyTypeCode	STOPNAME STOPMESS ADD_INFO CHA_INFO	varchar(8)	Default Stop name Default Stop message Additional Stop information Default change information
typStopPointDesignation	A 1-4 character string with printable characters.	nvarchar(4)	This is a gate, track or bus Stop identification to the public.
typStopPointLocalNumber	1-999.	numeric(3, 0)	The local number within a StopArea.

Data type	Values	SQL-type	Remark
typStopPointTypeCode	BUSSTOP REFUGE PLATFORM TRACK GATE PIER ENTRANCE EXIT UNKNOWN	varchar(8)	To be used in a multi-modal system to inform passengers about type of Stop Point.
typStringKeyValue		nvarchar(max)	May be NULL
typTime		datetime	May be NULL
typTransferLinkTypeCode	CONNECTI ST_ENTRY ST_EXIT	varchar(8)	Connection Link Station Entry Link Station Exit Link
typTransferModeCode	WALK, TAXI	varchar(8)	
typTransportAuthorityNumber	A 1-3 digit number.	numeric(3, 0)	This number should usually be officially agreed between all Transport Authorities.
typTransportMode-Code	NULL = unspecified BUS TRAM METRO TRAIN FERRY SHIP TAXI	varchar(8)	
typVatNumber	VAT-number according to EU-specifications.	nvarchar(20)	
typVehicleJourneyTypeCode	SERVICE, DEADRUN	varchar(8)	Service Journey Dead Run
typVehicleNumber	0-99999.	numeric(5,0)	
typYear	A valid year.	numeric(4,0)	
typYesNo	0 or 1, interpreted as No or Yes.	bit	
typZoneNumber	A 1-7 digit number.	numeric(7,0)	

Data type	Values	SQL-type	Remark
typZoneTypeCode	PARISHZ MUNICIPZ COUNTYZ TRAFFICZ TARIFFZO TALK_GRP INFO_REG PARKINGA FAREREFP	varchar(8)	Parish Zone Municipality Zone County Zone Traffic analysis Zone Tariff Zone Radio talk group coverage area Information region Parking Area –grouping Parking Points Fare Reference Point Zone

Appendices

22 How to set up the replication

To be able to utilise DOI data, the following things must be made.

22.1 Target database

To be able to receive replicated data, the user must define a target database. This database must be a Microsoft SQL Server database. However, the version and service pack requirement must be checked with the PubTrans system administrator before a new target database is to be attached to the replication system.

22.2 Initialisation of the target database

Initiation of tables is optional which means that the subscriber will not have to design the tables on their own. If the tables do not exist in the subscriber's database by the time of initiation they will be added to the database, the same applies to user defined data types.

When the subscriber is initialised or re-initialised with data, SQL Server first performs a 'truncate table' statement.

At initialisation, the primary key will be automatically set to the column *Id* in the tables. This must not be altered by the client application.

The subscriber may apply other indexes if needed. It is recommended that all additions to the target database be saved as SQL-scripts so they easily can be applied again if necessary. These scripts could also be of use when analysing the performance of the replication.

22.3 Interface views and backward compatibility

DOI uses interface views on top of the replicated tables. The interface views must be installed in the target database.

The user must select an interface view version. New subscribers should always use the latest version of the interface views, while existing applications can install a version that corresponds to an earlier version of DOI.

Scripts for interface views will be available for download from the PubTrans User Group web site (http://www.usergroup.pubtrans.com/).

Each of the underlying tables are guaranteed to contain all of the fields represented in the corresponding interface view, although a publication may contain more columns and tables than what is represented by the corresponding interface views.

At some stage the solution with interface views becomes inefficient. Older versions of DOI will then be supported from a separate replicated database with its own set of interface views.

22.4 Network

The publisher and subscriber must be located on the same network. The network must have the necessary bandwidth for fast and reliable data replication.

Within a Windows domain, you may use any network protocol that MS SQL Server replication supports. However, if you are crossing domain boundaries, you must use TCP/IP.

The user should also be aware of that the source database must be able to do so-called push replications to the target database. This may require special considerations if the replication is through a firewall.

22.5 User account and permissions

It is required that the publisher's assigned SQL-login to the subscriber have dbo permissions in the subscription database to make sure the proper permissions are granted.

If the protocol named pipes is used in communication between the servers the publisher's SQL Agent account needs Windows permissions on the subscribing system in order to access the machine. This can be established in one of two ways, either by using a domain account or two identical local accounts on both the local and the remote machine.

22.6 Configuration Data

The publisher needs the following information from the subscriber:

- Subscriber physical location
- IP address
- Server name
- Database name
- Database login name
- Database password

23 Replication background information

23.1 Why use database replication?

The subscription mechanism is preferred to direct access to the PubTrans database for several reasons:

- The subscribed data becomes a part of the application's local database. Dependency of direct access to and response time from PubTrans is limited. This means robust and self-contained applications and it puts the main responsibility of application performance on the supplier of the local application.
- Content and structure of publications is easier to maintain over time. A transparent data structure, like the publications, hides the internal structure of the PubTrans database and limits the side effects of most changes and optimizations that are made to PubTrans over time.
- The viewed data in the publication could be designed to present a logical view of the database and hide the way this structure actually is implemented. This reduces complexity in using the data in the publications.
- The software needed for the mechanism to work is a standard part of the database management system. It is easy to maintain and a proven concept.

23.1.1 Advantages with SQL Server replication

The publication and subscription mechanism is based on Microsoft SQL Server Replication.

Compared to direct database access or file transfer, the main advantages by using the built- in replication mechanism of Microsoft SQL Server are:

- It is a well-known and tested mechanism.
- It is flexible in the way of how and when data is distributed.
- It works over various network connections.
- It distributes the workload on the central system to each Destination application.
- It provides some degree of data redundancy, which improves fault tolerance.
- It is not a homemade solution depending on a critical resource or competence.

23.1.2 Limitations

Although replication is possible to database engines other than MS SQL Server, this will not be provided direct from PubTrans kernel databases. Moving data from PubTrans to a database engine other than MS SQL Server will always involve a primary step where data first is replicated to a target MS SQL Server database outside of PubTrans kernel databases, and a secondary step where data is moved from that database to the database managed by another database engine.

The supplier of PubTrans data, normally only supports the primary form of replication. The user is responsible for implementation of the second step.

23.2 Replication model

The replication model is composed of the following functional roles: publisher, distributor, subscribers, publications, articles, and subscriptions. They are explained below.

23.2.1 Publisher

The Publisher is a server that makes data available for replication to other servers. The Publisher can have one or more publications, each representing a logically related set of data. In addition to being the server where you specify which data is to be replicated, the Publisher also detects which data has changed during transactional replication and maintains information about all publications at that site.

23.2.2 Distributor

The Distributor is a server that hosts the distribution database and stores history data, and/or transactions and metadata. The role of the Distributor varies depending on which type of replication you implement. For more information, see Types of Replication in SQL Server Books Online. A remote Distributor is a server that is separate from the Publisher and is configured as a Distributor of replication. A local Distributor is a server that is configured to be both a Publisher and a Distributor of replication.

23.2.3 Subscribers

Subscribers are servers that receive replicated data. Subscribers subscribe to publications, not to individual articles within a publication, and they subscribe only to the publications that they need, not all of the publications available on a Publisher. Depending on the type of replication and replication options you choose, the Subscriber could also propagate data changes back to the Publisher or republish the data to other Subscribers.

23.2.4 Publication

A publication is a collection of one or more articles from one database. This grouping of multiple articles makes it easier to specify a logically related set of data and database objects that you want to replicate together.

23.2.5 Article

An article is a table of data, a partition of data, or a database object that is specified for replication. An article can be an entire table, certain columns (using a vertical filter), certain rows (using a horizontal filter), a stored procedure or view definition, the execution of a stored procedure, a view, an indexed view, or a user-defined function.

23.2.6 Subscription

A subscription is a request for a copy of data or database objects to be replicated. A subscription defines which publication will be received, where, and when. Synchronisation or data distribution can be requested either by the Publisher (a push subscription) or by the Subscriber (a pull subscription). A publication can support a mixture of push and pull subscriptions.

23.2.7 MS SQL Server replication models

Microsoft SQL Server supports the following replication topologies:

- Central publisher
- Central subscriber

- Central publisher with remote distributor
- Publishing subscriber

Several processes are responsible for copying and moving data between the Publisher and Subscriber. These are the Snapshot Agent, Distribution Agent, Log Reader Agent, Queue Reader Agent, and Merge Agent. For more information about the agent processes, see the chapter Agents and Monitors in SQL Server Books Online.

Global Identifiers

PubTrans uses a numbering convention called *Global Identifiers* or *GID* for short. A GID is a numeric identifier that has the same value for the same object across all installations of PubTrans databases. GID is used extensively in the interfaces as keys for referring to PubTrans objects. It is assumed that a client application can map its internal data model to the proper GID when exchanging data with PubTrans.

GID Construction

A GID is a 16-digit number. Currently, there are two main types of GID:

Key-based: A 16-digit number constructed by concatenating numeric attributes associated with the object. Key based GIDs starts with the number '9'.

Abstract: A 16-digit number that does not contain any special meaning. An abstract GID does not start with the number '9'.

Key based GID

Key based GIDs consist of a four-digit class identifier, a three-digit Transport Authority number and a 9-digit value divided into one or several fields.

Each field contains a numeric attribute data from the object. The fields have different representation for each class; either constructed by concatenating numeric attributes of the object in question or as an abstract number that does not contain any special meaning. This is explained for each type of GID below.

A key based GID is defined in the context of a Transport Authority, i.e. it is unique within a Transport Authority. If an object is referenced by more that one Transport Authority, it will have a GID for each Transport Authority. Thus, it is possible that several different GIDs refer the same object in PubTrans.

A GID is uniquely referring zero or one object version at a specific point in time, but the same GID may refer different object versions at different times.

Each field has two reserved values; zero that means 'unknown' and an all-9-value which is reserved for testing purposes. A GID with a zero-field does not refer any object. The zero-value field is used in some cases to let PubTrans assign the field automatically. A GID with an all-9-value shall only be used for test purposes.

Abstract GID

An abstract GID is a non-descriptive 16-digit number. An abstract GID is always less than 9000000000000000. Abstract GIDs are assigned automatically by PubTrans. Because abstract GIDs are not based on key data provided from external systems, there is no way to see if two abstract GIDs from different PubTrans database instances represents the same object or not. In fact, abstract GIDs in different PubTrans databases never overlap, because an abstract GID is globally unique within a PubTrans group.

Object Types with GID

The following object types (classes) have global identifiers in PubTrans.

Class Name	Class Id
Block	9041
Bridging Device	9095
Contractor	9013
Dead Run	9016
Deviation Case	9076
Deviation Message (obsolete)	*9071*
Direction (Direction of Line)	9014
Direction of Line (obsolete)	*9012*
Duty	9061
Employee	9051
Journey Pattern Point	9025

Class Name	Class Id
Line	9011
Place	9091
Service Journey	9015
Station Entrance Point	9023
Stop Area	9021
Stop Point	9022
Transport Authority	9010
Vehicle	9031
Virtual Vehicle	9038
Zone	9081

GID Format

Bold numbers indicates fixed values, and *italic* numbers indicate variable fields.

Transport Authority

9	**0**	**1**	**0**	*1*	*2*	*3*	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Not Used								

A *transport authority numb*er must be between 1 and 998.

Transport authority numbers should be coordinated between co-operating transport authorities, preferably on a national level.

Line

9	**0**	**1**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	**0**	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Line Number				Not used				

A *line number* must be between 1 and 9998. The line number must be unique within a transport authority.

A line number is not necessarily the number presented to the public. In PubTrans, the public identification of the line is called *line designation*, which is a separate attribute from the line number.

If alphanumeric characters are used in the public identification of the line, it is necessary to define a numeric counterpart for use in GID, which can be considered as an internal alternative identification of the line.

If a line is operated by more that one transport authority (split responsibility), the line will have one GID per transport authority, with different transport authority number for, but preferably with same line number.

Direction of Line (old format)

This GID is obsolete in PubTrans 5. It will be supported in DOI 3 for backwards compatibility. It is replaced by the GID described in section 0 nedan.

9	**0**	**1**	**2**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Line Number				Direction	Not used			

A *direction-of-line number* must be 0 or 1.

Direction of line is used to separate journey patterns and routes into two main groups running in opposite directions.

Contractor

9	**0**	**1**	**3**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	**0**	**0**	**0**	**0**	**0**
Class Id				Transport Authority Number			Contractor Number				Not Used				

A *contractor number* must be between 1 and 9998. A contractor number must be unique within a transport authority.

The contractor number denotes a vehicle operator organisation that is engaged to operate public transportation within a region governed by a transport authority. If the same vehicle operators are engaged by several transport authorities, they will have a GID within each transport authority.

It is assumed that it is the transport authority that coordinates and assigns the contractor number for each engaged vehicle operator.

Direction (Direction of Line)

This GID replaces the direction of line GID in section 0 ovan and will be used in PubTrans 5 and in DOI 4. The terms *direction* and *direction of line* are used interchangeably.

9	0	1	4	1	2	3	1	2	3	4	1	0	0	0	0
Class Id				Transport Authority Number			Line Number				Direction	Not used			

A *direction number* must be 1 (corresponding to 1 in the obsolete direction of line GID format) or 2 (corresponding to 0 in the obsolete direction of line GID format).

Direction (of line) is used to separate journey patterns and routes into two main groups running in opposite directions.

Service Journey

9	0	1	5	1	2	3	1	2	3	4	1	2	3	4	5
Class Id				Transport Authority Number			Line Number				Journey Number				

A *service journey number* must be between 1 and 99998. A service journey number must be unique within a line and for a specific operating day.

The service journey GID uniquely defines a service journey within PubTrans. If it is important to retain the service journey GID over time, it is strongly recommended to avoid renumbering of the service journeys in the source system each time data is provided to PubTrans.

Dead Run

9	0	1	6	1	2	3	1	2	3	4	1	2	3	4	5
Class Id				Transport Authority Number			Contractor Number				Dead Run Number				

A *dead run number* must be between 1 and 99998. In the context of one transport authority, a dead run number must be unique within the contractor for a specific operating day.

Dead run numbers can be assigned sequentially over time to avoid duplicate numbers.

Stop Area

9	0	2	1	1	2	3	1	2	3	4	5	6	0	0	0
Class Id				Transport Authority Number			Stop Area Number						Not used		

A *stop area number* must be between 1 and 999998. A stop area number must be unique within a transport authority.

In PubTrans it is possible to define a stop area that can be utilised by several transport authorities. If several transport authorities share the operation at a stop area, they can assign a GID each.

Stop Point

9	**0**	**2**	**2**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*1*	*2*	*3*
Class Id				Transport Authority Number			Stop Area Number						Local Stop Point Number		

A *local stop point number* must be between 1 and 998. A local stop point number must be unique within a stop area.

This is the local number within the stop area assigned to each stop point, e.g. track or gate number. There is also a GID for globally numbering of stop points within a transport authority, see Journey Pattern Point in section 0 nedan.

Station Entrance Point

9	**0**	**2**	**3**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*1*	*2*	*3*
Class Id				Transport Authority Number			Stop Area Number						Local Station Entrance Number		

A *local station entrance point number* must be between 1 and 998. A local station entrance point number must be unique within a stop area.

Journey Pattern Point

9	**0**	**2**	**5**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Class Id				Transport Authority Number			Journey Pattern Point Number								

A *journey pattern point number* must be between 1 and 999 999 998. A journey pattern point number must be unique within a transport authority.

A journey pattern point is any type of point that can occur in a journey pattern, i.e. a point that can be used as a reference point in a scheduling system. Journey pattern points are: stop points, via-points and parking points. The journey pattern point number usually corresponds to a stop number.

Vehicle

9	**0**	**3**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport			Contractor Number				Vehicle Number				

	Authority Number		

A *vehicle number* must be between 1 and 99 998. In the context of one transport authority, a vehicle number must be unique within a contractor.

If a vehicle is identified in context of more than one transport authority, a vehicle will have a GID for each transport authority.

If a vehicle becomes operated by another vehicle operator (usually as a change of ownership) and still is in the context of a transport authority using PubTrans, the vehicle will get a new GID, because the contractor number will change. Probably the vehicle number as well, because each vehicle operator assigns internal vehicle numbers. Thus, a vehicle GID is not an identifier for the physical vehicle. However, PubTrans can handle several vehicle-GIDs referencing to one physical vehicle if some other identifier for the physical vehicle is provided at import of vehicle data.

Virtual Vehicle

9	**0**	**3**	**8**	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
Class Id				Virtual Vehicle Number											

A *virtual vehicle number* must be between 1 and 999 999 999 998. A virtual vehicle number corresponds to one vehicle within one PubTrans system at a certain point in time.

Virtual vehicle numbers are used as temporary vehicle-identifier tags when information about the actual vehicle numbers is not available. Virtual vehicle numbers should only be used when it is not possible to use actual vehicle numbers. If virtual vehicle numbers are used, then it is preferable that a virtual vehicle number is attached for as long as possible to a certain vehicle.

Block

9	**0**	**4**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Block Number				

A *block number* must be between 1 and 99998. In the context of one transport authority, a block number must be unique within a contractor.

Employee

9	**0**	**5**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Employee Number				

An *employee number* must be between 1 and 99998. In the context of one transport authority, an employee number must be unique within a contractor.

The employee GID is mostly used to identify drivers within different vehicle operators.

Duty

9	**0**	**6**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*5*
Class Id				Transport Authority Number			Contractor Number				Duty Number				

A *duty number* must be between 1 and 99998. In the context of one transport authority, a duty number must be unique within a contractor.

Deviation Message

This GID is obsolete in PubTrans 5.

9	**0**	**7**	**1**	*1*	*2*	*3*	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*
Class Id				Transport Authority Number			Message System Number			Message Number					

A deviation message GID consists of two parts: a *message system number* that must be between 1 and 998, and a *deviation message number* that must be between 1 and 999998.

The message system number identifies the source system of the message and the message number is a unique identifier of the deviation message within that system. Both numbers are assigned sequentially by PubTrans

NOTE: In PubTrans 4 a valid message system number can be zero.

Deviation Case

9	**0**	**7**	**6**	*1*	*2*	*3*	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Class Id				Transport Authority Number			Not used	Deviation Case Number							

A deviation case number must be between 1 and 99999998.

The deviation case number is assigned sequentially by PubTrans

Zone

9	0	8	1	1	2	3	1	2	1	2	3	4	5	6	7
Class Id				Transport Authority Number			Type of Zone Number		Zone Number						

A zone GID consists of two parts: a *type-of zone-number* that must be between 1 and 98, and a *zone number* that must be between 1 and 9 999 998.

The following zone type numbers is reserved:

Zone type	Usage
10	Administrative zone of transport authority.
11	Local administrative zone (corresponding to Swedish 'kommun').
12	Regional administrative zone (corresponding to Swedish 'län').
13-19	Custom defined administrative zone types.
20	Tariff zones.
21-29	Custom defined additional tariff zone types. Can be used if different types of tariff zones are used in parallel.
30-39	Custom analysis zone types.
40-49	Custom technical systems zone type, e.g. radio coverage zones.
50	Parking area
51-89	Reserved for future use
90-98	Reserved for system supplier specific use.

Place

9	0	9	1	*1*	*2*	*3*	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Class Id				Transport Authority Number			Not used	Place Number							

A *place number* must be between 1 and 99 999 998 and must be unique within a transport authority.

Bridging Device

9	**0**	**9**	**5**	*1*	*2*	*3*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Class Id				Transport Authority Number			Bridging Device Number								

A *bridging device number* must be between 1 and 999 999 998, both values inclusive. The values 0 and 999 999 999 are reserved for special purposes. A bridging device number must be unique within a transport authority.