Real-time Output Interface 3.0
Interface Specification

Document identity:	IS-PT/I/ROI/3
Revision:	G
Date:	2018-03-26

Document identity
IS-PT/I/ROI/3
Date
2018-03-26
Revision
G

Real-time Output Interface 3.0
Interface Specification

Copyright

Revision History

Revision	Date	Update	Updated by
A	2005-05-12	First release	Ulf Bjersing
B	2005-10-20	Second release	Ulf Bjersing
C	2006-01-11	Corrections and minor changes	Ulf Bjersing
D	2006-04-12	Minor changes in XSD files.	Ulf Bjersing
E	2006-06-20	SubscriptionUpdateRequest added in ROI-ToPubTrans.xsd Minor corrections in state diagrams and associated texts.	Ulf Bjersing
F	2011-04-27	Clarifications concerning Assignments. Added exception to the sequence of the data distribution. Additional deviation reasons added.	Ulf Bjersing
G	2018-03-26	Added information describing Vehicle Journey Deviation Event. Clarification about new/modified subscription start. Added attributes in DatedVehicleJourney according to ROI-FromPubTrans.xsd version 3.0.9.	Linnéa Raatikainen

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**
IS-PT/I/ROI	G
ROI-SharedTypes.xsd	1.0.7
ROI-ToPubTrans.xsd	3.0.7
ROI-FromPubTrans.xsd	3.0.9
PT-SharedTypes.xsd	1.0.6
PT-XMLStream.xsd	1.0.4
XMLStream.xsd	1.0.5

Table of content

1 Introduction

The purpose of this document is to describe NOPTIS Real-time Output Interface (ROI) and how a client system shall utilise the content and functionality of ROI.

1.1 What is ROI?

ROI is one of the NOPTIS public interfaces. With ROI it is possible for client applications to receive real time data and deviation information about the public transport operation. The data is provided in a uniform format, regardless of which source system it came from.

ROI can be used for many types of purposes where there is use for real-time data. Before any communication take place, it is therefore necessary to define the business goals for the client application, take the decisions of how ROI shall be utilised and data interpreted to fulfil these requirements, and finally design and implement the interface processing in the client application.

Through ROI, An integrator system distributes the production plan for the near future, the short term changes made to the production plan, the updated predictions and recorded events based on vehicle progress reporting in real-time from source systems. ROI does not contain all data about the long term planned operation. That data is available through other NOPTIS interfaces, primarily DOI.

At initialisation of a session, the client applications subscribe to what lines or stops the application wants information about. As long as the session is alive, the client application will receive data in real time about the subscribed objects.

A client system receives the real time data as XML-structured messages. The integrator system pushes these messages on to the client system as they happen. From the client system point of view, ROI is event driven.

The payload data flow is one way, i.e. from The integrator system to the client system. However, the client system must be able to provide initial data and feedback during the session, so in that sense it is a two-way communication.

ROI provides data as messages.

1.2 Related Standards

ROI is modelled based on TRANSMODEL, the European reference data model for public transportation. The design of ROI has also been inspired by the CEN standardisation in progress, mainly the work done in Trident and SIRI.

2 Process Overview

In an ROI perspective, the client system can be viewed as two parts:

- The interface processing that manages the communication with the integrator system and adapts data for the application.
- The application itself that consumes the data for some purpose.

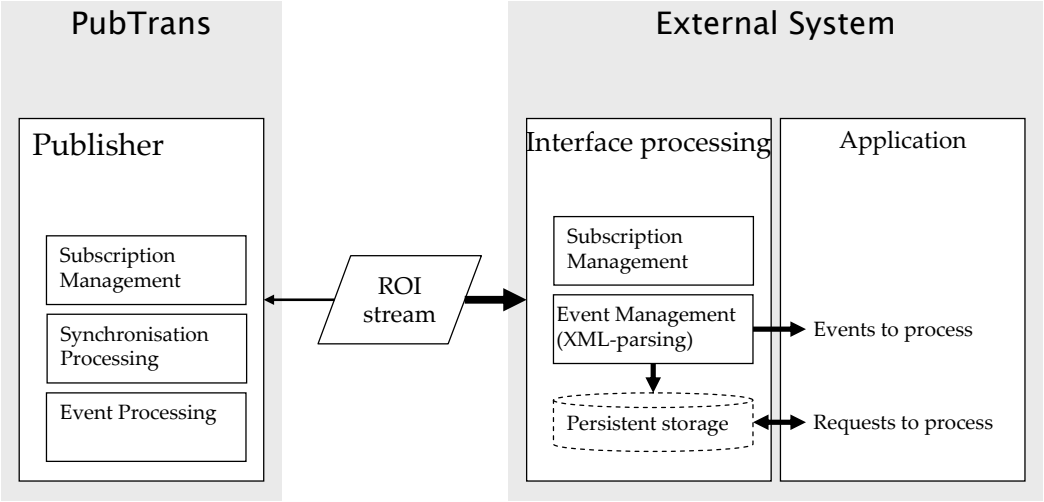

Fig. 1. ROI process overview

This document focuses on the interface processing. The document describes the content of the ROI stream and gives explanations and recommendations how this data shall be processed.

2.1 Process Phases

In short, to establish, maintain and terminate a ROI-session using the XML stream mechanism, the following steps are required:

1. The external system establishes an XML-stream over a TCP-connection to the integrator system. Now there is a two way communication established. See the appendixes for details of initialising an XML-stream.
2. Client sends a subscription request, describing which lines and stops to subscribe on and a time window that says how far ahead in time the integrator system shall provide the production plan.
3. The integrator system starts the synchronisation processing, i.e. the process to update the client with relevant parts of the production plan. The client is required to acknowledge the reception at certain intervals.
4. As soon as the client acknowledges reception of a chunk of the production plan, the integrator system starts to push events concerning objects for that part of the production plan.

5. When the integrator system has provided the production plan for the requested time window, processing enters the normal state, where events are pushed as they happen, and the production plan is rolled out to keep up with the time window.
6. Either the integrator system or the client terminates the communication.

Some interesting thing to notice:

- The way synchronisation works means that a client gets the data about the current public transport operation first. As soon as this data is confirmed, the integrator system pushes events about this operation. This means that clients only interested in real-time information will have a short start-up time.
- Once the current operation has been synchronised, event data has higher priority than further synchronisation of the production plan.
- Normally, the process is endless in the sense that the communication only has to be terminated because the integrator system or the client system is taken down for maintenance or failure, or that the communication link is broken.
- For each subscriber, the integrator system keeps track of how far the distribution of data has reached. The integrator system also recognizes if the client need recovery of data after a restart or communication problem. The clients do not have to store any state data and needs only a few configuration parameters.

2.2 Process in Client

2.2.1 Client Configuration

The version of the XSD shall be agreed upon for each implementation. If XML-streams are used, configuration requirements for XML-streams are found in the appendixes.

2.2.2 Receiving Data

Typically the client receives XML encoded messages concerning different types of objects; most common are messages concerning arrivals and departures.

The client shall validate that the XML-syntax in the message is consistent with the agreed schema definition. After successful validation, the client shall extract the data of interest, perform any type conversion and pass the data on to the main application logic, for instance to update a database or fire an event.

2.2.3 Caching Data

ROI supports applications that do not cache objects; this is an option that can be selected in the subscription. See section 2.3.4 for details.

A client system can choose to use an event model or a request driven approach based on database lookup.

If the client application is request driven the client needs to cache data in one way or another. This could be made in a volatile storage, which means that a full synchronisation must be made at each start-up, or as a

persistent storage, which means that synchronisation can start from the position where it was stopped last session.

2.2.4 Sending Data

The client sends data by writing XML elements to the integrator system. The client must ensure that the XML complies with the agreed XSD, otherwise an error will occur. The communication process when using XML-streams is described in the appendixes.

2.2.5 Client Performance

The client system must be able to receive and process messages from the integrator system without significant delay.

Data volumes and number of messages depends on the following factors:

- The scope of the client's subscription, i.e. the number of arrivals and departures during rush hour that the client wishes real-time data about.
- The size of the subscribed time window. During synchronisation up to the point where the production plan has been distributed for the whole time window, the number of messages will be significantly higher.
- The traffic situation. A situation that changes generates more messages than a situation that is stable.

The integrator system pushes events with higher priority than synchronisation of production plan. However, the client will receive everything in sequence, so a slow processing of synchronisation data can punish the overall performance.

There is a mechanism to avoid overloading the client with to much synchronisation data in start up situations. The integrator system will periodically request information from the client concerning which the last successfully received message was. No further synchronisation data will be sent before there is an acceptable response time from the client. Events related to already synchronised data will still be sent as the integrator system pushes events for objects in the production plan that the client application has confirmed it has received.

2.2.6 Using ROI only?

ROI is designed so that systems can choose to use a combination of DOI and ROI or use ROI only.

- Using DOI and ROI means that data describing stop points, stop areas, destinations, vehicles and service requirements in detail shall be retrieved from DOI. ROI refers to those objects in DOI instead of expanding the data. This option is recommended for best performance, since the payload in ROI can be limited. Client systems with a large subscription scope shall use this method.
- Using ROI only means that a client system does not need DOI. On the other hand this option increases the depth of ROI-data because some redundant data need to be provided about each object. This option is only recommended for client system with a small subscription scope.

2.3 Subscription

The purpose of the subscription is to limit the processing load on the integrator system, the client system and the communication system in between them. By letting the client systems define their scope of interest, only real-time data concerning objects within this scope need to be transferred and processed in the client.

The subscription can be either journey or stop centric. For instance, a stop display control system will probably subscribe on the stops where displays are located.

The client can modify the scope of the subscription if necessary.

A new or extended subscription will subscribe for data in upcoming production plans.

The integrator system retains subscription data for client systems until the subscription is actively terminated.

2.3.1 Subscription Configuration

The client system must have functions for configuring the subscription scope. The vehicle journey and deviation case subscription scope is defined using so called Global Identifiers (GID), a public key syntax that the integrator system uses to define numeric keys for different types of business objects. GID is explained in the appendixes.

The recommended way of implementing a configuration function is to look up the possible stops in DOI, let the user select them by name and/or location and store the corresponding GID for the selected stops in the system configuration.

2.3.2 The Look-ahead Window

Each subscription defines a look-ahead window, which defines how long time in advance the production plan shall be sent to the client. Different type of clients will require different look-ahead windows. The look-ahead window cannot be longer than the existing production plan. ROI typically holds a production plan covering the rest of the operating day and maybe also the following operating day.

Data about vehicle journeys will not be exposed to the client until a vehicle journey's timetabled start time falls within the look-ahead window.

2.3.3 The Start Time of Subscription

Each subscription can define a start time. If omitted, the subscription starts from the current date/time.

Note also that a subscription only covers items added to the production plan after the subscription was added or extended.

See also section 2.5 Distributing Data

2.3.4 Resumed Subscriptions

The publisher retains subscription configuration and state for client systems until a subscription is actively terminated. There are two options available for the client after re-establishing a broken connection to the same publisher peer:

- The client can request that the subscription is resumed from the point of the last known successfully transferred message, and thus keep an unbroken history of data.
- The client can request that the subscription is resumed from a later point in time, and thus get a faster recovery.

2.4 Client Options for Subscription

2.4.1 Type of Objects to Subscribe On

The client can choose which main areas to include in a subscription:

- Vehicle Journeys, which is the real-time data about vehicle journeys; progress, predictions, arrivals, departures and connections etc.
- Deviation Cases, which is additional information to passengers, staff etc provided in different formats for different media.
- Assignments, which are the resource allocations of vehicles, drivers etc.
- Network Deviations, which is additional information, not related to any specific journey, concerning stop points, elevators and station entrances.

2.4.2 Depth of Subscription

The client can choose the depth of the subscription. Clients that rely on ROI-data only, have a number of options to expand the content of the messages. The following options are available for Vehicle Journey subscriptions:

- Expand Line Data, which means that additional data from DOI concerning the line is appended to each dated vehicle journey.
- Expand Vehicle Operator Data, which means that additional data from DOI concerning the vehicle operator is appended to each dated vehicle journey.
- Expand Stop Data, which means that additional data from DOI about the stop is appended to each arrival and departure.
- Expand Destination Data, which means that additional data from DOI concerning destination display information is appended to each departure.
- Expand Vehicle Data, which means that additional data from DOI concerning the assigned vehicle is appended to each monitored vehicle journey.
- Expand Service Requirement Data, which means that additional data from DOI concerning service requirements information is appended to each departure.
- Expand Deviation Message Data, which means that the public note of the referred deviation message is expanded and appended to each arrival, departure and network deviation. This option should only be used in narrow subscriptions where the Deviation Case area is excluded. It is usually more efficient to extract the information from the Deviation Case area.

The idea with the deeper forms of subscription is to support client systems that do not work with persistent data. See section 2.2.6 for details regarding using this option.

2.5 Distributing Data

Data is distributed by sending events in form of XML messages according ROI. Distribution of data is of three types:

- Initial distribution of all objects within the subscription scope. This establishes the initial data with the last known state of operation.
- Subsequent additional distribution of objects. This includes objects that are added during the subscription and objects that successively falls within the subscription time window.
- Subsequent updates of data in real-time. These events are always referring to an object provided in the initial distribution.

2.5.1 Sequence of Events

It is not uncommon that different events contain references to data that is assumed to have been delivered to the subscriber already. Therefore, the sequence of how messages are delivered is important.

As a general rule, the events will be sent in an order that ensures that a referenced object is distributed before a referencing object. There are exceptions to this:

- When a chunk of vehicle journeys are initially distributed, an update event for a vehicle journey can sometimes be sent before the vehicle journey create event. This update can be ignored, because the last known state will be included when the vehicle journey is distributed, so no loss of data will occur.
- Connections will be distributed with a reference to the arrival or departure in the other end of the connection that may refer an object that is not distributed.
- A vehicle journey assignment can under certain circumstances be distributed before the referenced dated-vehicle journey has been distributed.

2.5.1.1 Initial Distribution

The initial distribution contains objects within the subscription scope with their last state in the following sequence.

1. Deviation Case Create Event, and for each deviation case its Deviation Message Version Events and Publication Decision Events. Only deviation cases within the subscription scope are distributed.
2. All current and future valid Assignment Events that the subscriber is authorised to see are distributed if requested by the subscriber.
3. All current and future valid Network Deviation Events are distributed if requested by the subscriber.
4. Vehicle Journey Create Event, and for each vehicle journey all its Arrival Create Events, Departure Create Events and Connection Create Events. Only vehicle journeys within the subscription scope are distributed. This is what is called the *production plan*.

2.5.1.2 Additional Distribution

Objects that have been added after the initial distribution are distributed as they become visible within the subscription time window.

2.5.1.3 Real-time Update Events

Real-time update events normally refer to an object already distributed. They are sent in the order they are entered into the integrator system. For a subscriber, it means that update events of different types will not arrive in any specific order.

2.6 Recovery

When a client re-establishes a connection, the integrator system publisher can perform recovery for that client. The purpose of the recovery is to bring the client up to date with its look-ahead window as defined in the subscription.

2.6.1 Using Recovery

A client system can choose if it wants to recover from the point where communication was terminated last time, have data repeated from a specified time, or to drop recovery and start from the current situation. A client system that uses recovery might have a longer start-up time because synchronisation is always performed in sequence based on the timetabled start time of the dated vehicle journeys. To request recovery, the client should not provide a start date/time in the resume subscription request. In case the client does, historical data will be lost.

If the client has been disconnected for a long period of time, or if the client connects for the first time, the client will experience a continuous stream of messages. This could prove a high load of work for the client. However; the integrator system will limit the flow of data by periodically checking how much data the client has been able to process.

2.6.2 Skipping Recovery

Client systems that do not want to recover historical data about finished vehicle journeys shall define a start/date time in the resume subscription request that corresponds to the time the client system expects to be in operation. This will speed start-up after system failure.

2.6.3 Limitations of Recovery

Recovery is limited back in time to the start of the current production plan, which usually means today and yesterday.

Recovery only includes the final states of objects, i.e. it is not possible to recover intermediate state changes.

3 Concepts

This chapter presents a number of concepts that ROI is based upon.

3.1 Production Plan

A *production plan* is a specific version of the intended operation for a specific operation day. It represents a conventional image of one version of the latest valid plan, frozen at a point in time defined by the user, according to local requirements.

The initial production plan forms the basis for the daily operation. Control actions are applied during the operation day on the production plan according to certain rules and principles. The current production plan mirrors these changes and is the basis for applying real time reports from vehicles.

The production plan exposed in ROI contains the integrated view of all input to the integrator system; the long term planned operation as provided through DII, the short term changes and passenger deviation information as provided through RII, and finally real-time progress reporting for vehicles as provided through VSI.

3.1.1 Content

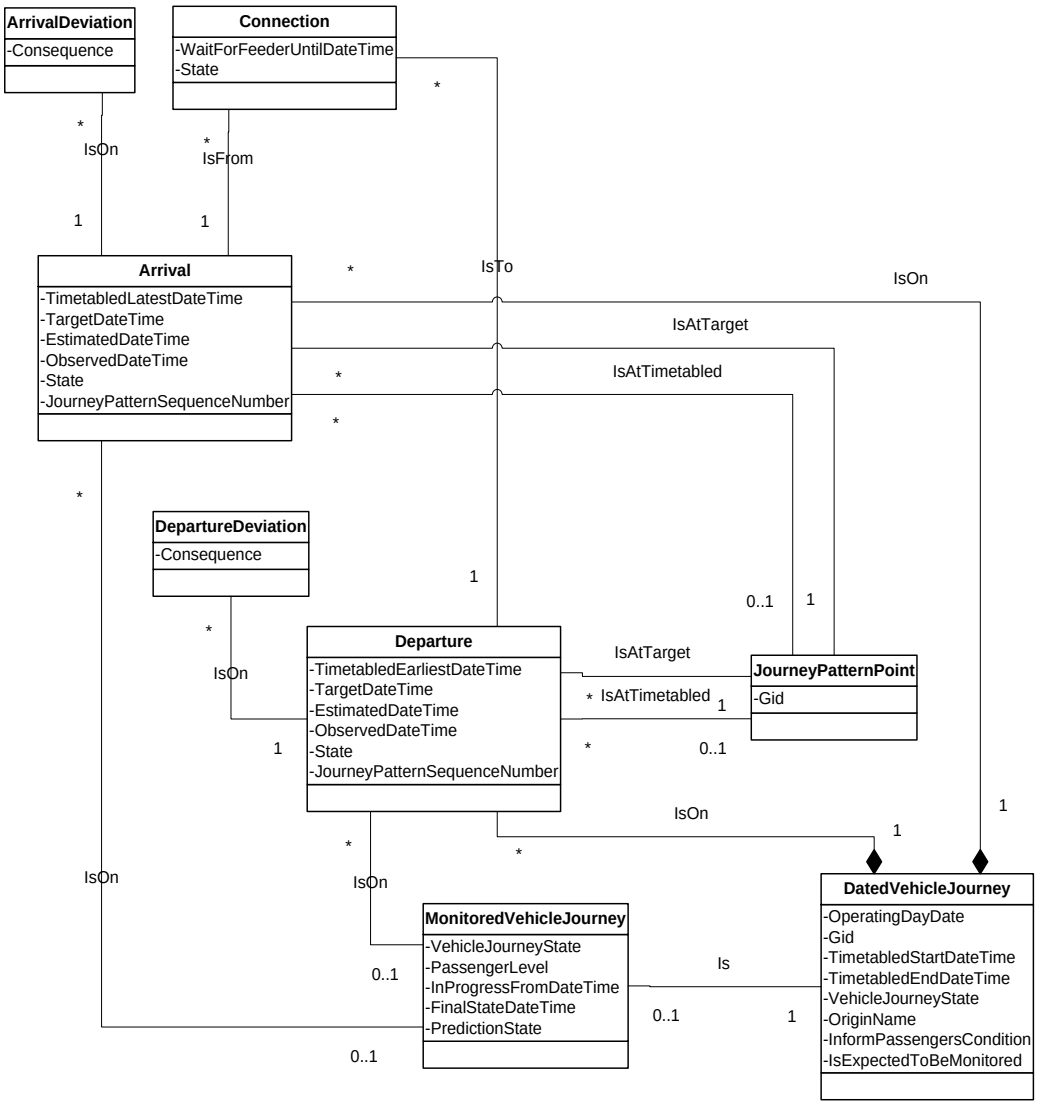

Fig. 2. Production plan

The most important content in the production plan are:

- Dated Vehicle Journeys.
- Dated Arrivals and Dated Departures, the arrival/departure of Dated Vehicle Journey's at individual stops.
- Dated Connections, the matching pairs of a Dated Arrival on one Dated Vehicle Journey with a Dated Departure of another Dated Vehicle Journey where interchange shall be monitored.
- Deviations applied to arrivals, departures or Vehicle Journeys as data changes and/or state changes.

These objects are described in detail in the following sections.

3.2 Vehicle Journeys

3.2.1 Dated Vehicle Journey

A *dated vehicle journey* is a vehicle journey that shall be operated on a specific operating day. The dated vehicle journey can be either long term planned, i.e. a *normal dated vehicle journey*, or short term planned, i.e. an *extra dated vehicle journey*. In the integrator system, a normal vehicle journey is an object that is provided through DII, while an extra dated vehicle journey is created as a control action submitted through RII.

3.2.2 Monitored Vehicle Journey

A *monitored vehicle journey* describes a journey recognised as being in process, at the time of the vehicle monitoring. In the integrator system, a monitored vehicle journey will be related to a *dated vehicle journey*. This means that monitoring and states will be based on the dated vehicle journey present in the latest valid plan.

A monitored vehicle journey is always associated with a physical vehicle that can be monitored through a vehicle detection system of any kind.

The integrator system creates a monitored vehicle journey when the first progress report referencing a particular dated vehicle journey is received through VSI from an external vehicle detection system. From that point, the state of the monitored vehicle journey is continuously updated according to subsequent progress reports submitted via VSI (see state diagram below). A dated vehicle journey can only have one monitored instance.

As long as the monitored vehicle journey is not completed, there is always some uncertainty regarding the quality of detection of vehicle movement. In order for a consumer system to detect this, state machines include states for un-normal operation.

It is strongly advisably that the vehicle monitoring system sends its first progress report when the vehicle is ready to board at the initial terminus, which means before the actual departure. This is information that can be used by display systems at terminals and by the operations control to get information that the vehicle is at the first stop ready to board and depart.

If the integrator system receives a progress report regarding a dated vehicle journey that is not yet assigned, it should make an implicit assignment and start to monitor the vehicle journey.

3.2.3 States

The figure below shows the combined states of a dated vehicle journey (white) and a monitored vehicle journey (grey).

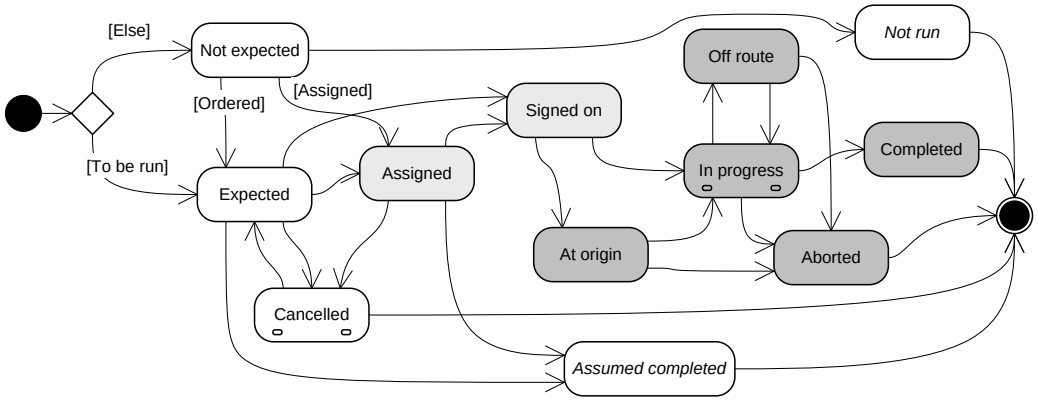

Fig. 3. Vehicle journey states

Initially, a dated vehicle journey can be either **not expected** or **expected**. This has to do with if the vehicle journey originally was scheduled to run or to be ordered. A vehicle journey that was scheduled to be ordered is not expected to run until explicitly ordered. Usually, a dated vehicle journey that is not expected shall not be presented or be presented with the conditions to order it.

The **assigned** state tells the client system that now there is a vehicle assigned to run the dated vehicle journey. Assignments can be made in advance by the operations control to allocate a specific vehicle to work certain dated vehicle journeys.

A dated vehicle can also be **cancelled**, and the cancellation can optionally be revoked.

The integrator system sets states based on progress reporting through VSI and RII. Initially, the assignment of a vehicle journey can be confirmed by the driver, which is indicated by the state **signed on**. As soon as the first progress report is received in VSI, the integrator system starts vehicle journey monitoring by creating a monitored vehicle journey. If the report indicated an arrival at the first stop the initial state will be **at origin**. If the report indicated a departure from the first or later stop or if it indicated an arrival at a later stop the state is set to **in progress**.

If the vehicle system detects that a vehicle is not following the expected route, it can change the state to **off route**.

If the vehicle finally reaches its destination, the vehicle journey receives the state **completed**, else it is **aborted**. Once in progress, a cancellation or sign off will be regarded as the monitored vehicle journey has been aborted. If an aborted dated vehicle journey is resumed again, the integrator system will create a new instance of a monitored vehicle journey.

3.2.3.1 Implicit states

If an expected vehicle journey is not cancelled and never becomes in progress, it should at some point in time be considered as **assumed completed**. If a not expected dated vehicle journey is never run, it should at some point in time be considered as **not run**. This could for instance be the time when the deadline for ordering expires.

3.2.3.2 In progress sub states

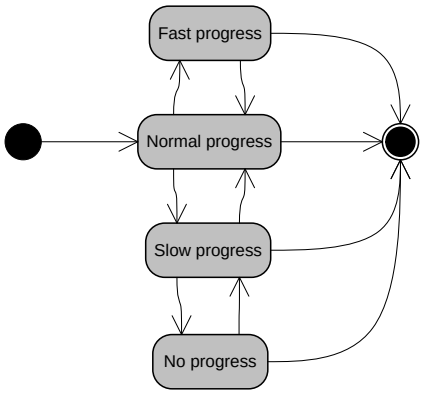

*Fig. 4. Sub states of the **in progress** state for a monitored vehicle journey.*

The **in progress** state has several sub states, telling the current state of progress. If, for instance the vehicle has been stuck in congestion, this could be indicated by **slow progress** or **no progress**. This state can be used to update passenger expectations further along the route.

3.3 Arrivals and Departures

An arrival and departure at a point in journey pattern on a dated vehicle journey has four different types of data associated with them:

- The timetabled data, representing the initial planned schedule.
- The target data, representing the most recent plan available for passenger information purposes. This is initially equal to the timetable data, but can be updated in real-time as a consequence of re-planning, optimisation or forecasts.
- The estimated data, representing the most recent best guess, based on forecasts made by the integrator system or an external system.
- The observed data, representing the actual performance as recorded by a vehicle monitoring system.

Dated arrivals and departures have a state model consisting of super states for the dated vehicle journey and different sub states for the call. A client application must inspect these states and update passenger information accordingly.

3.3.1 Arrival

An arrival describes the part of a call up to that a vehicle have arrived to a stop point (or passed it). A *dated arrival* contains the original and last valid plan for an arrival. In addition, a *monitored arrival* contains supplementary data based on vehicle monitoring such as observed arrival.

3.3.1.1 Timetabled Arrival

The timetabled latest arrival time and the timetabled point in a journey pattern describe when and where the arrival was initially scheduled.

Sometimes the initial point in journey pattern cannot be decided in the planning phase. In this case, the stop point could be an imaginary stop point, and the physical stop point is assigned in real time.

3.3.1.2 Target Arrival

The targeted latest arrival time and the journey pattern point intended for the arrival are initially equal to the timetable data, but can be updated in real-time as a consequence of short term re-planning, optimisation or forecasts.

The targeted journey pattern point is the most recent plan to what stop point a vehicle will arrive at.

3.3.1.3 Estimated Arrival

The estimated arrival time is the mean time as forecasted by the integrator system or an external system. This time is the most recent best guess.

3.3.1.4 Observed Arrival

The observed arrival time is the actual arrival time as recorded by a vehicle monitoring system. The integrator system should not set this time unless a progress report with an explicit time is received.

3.3.1.5 States

The figure below shows the combined states of a dated arrival (white) and a monitored arrival (grey).

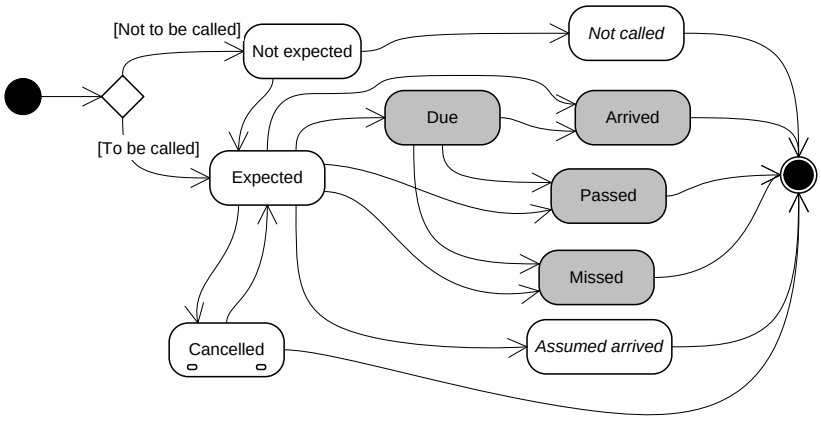

Fig. 5. Arrival states

Initially, an arrival can be either **not expected** or **expected**. This has to do with if the specific arrival originally was scheduled to be called at or to be ordered. If a dated vehicle journey is not expected, then all arrivals are also not expected. On the other hand, if a vehicle journey is scheduled to be expected, there may still be individual arrivals that shall be ordered and therefore have the state not expected. An arrival that was scheduled to be ordered is not expected to be called until explicitly ordered. Usually, an arrival that is not expected shall be presented with the conditions to order it or not be presented at all.

Just before arrival the arrival may optionally get the state **due**, which means that the vehicle is just about to arrive or pass. The timing of when a vehicle is due is up to the vehicle monitoring system to decide. This is

an optional state that requires that the automatic vehicle location system is capable of provide such information.

An arrival can finally be **not called**, **arrived**, **passed** or **missed**. The not called-state and the missed-state will be set for all arrivals where an arrived- or passed-state is missing when the integrator system receives a progress report further down the route. A missed-state means that a call was expected and that a vehicle implicitly is arrived or passed, but there was no observation recorded. A not-called-state is set when the no call was recorded and the stop was not expected to be called.

3.3.1.6 Implicit states

If an expected dated arrival is not cancelled and not is monitored, it will at some point in time be considered as **assumed arrived**. If a not expected dated arrival is never called, it will at some point in time be considered **not called**. This could for instance be the time when the deadline for ordering expires.

3.3.1.7 Cancelled sub states

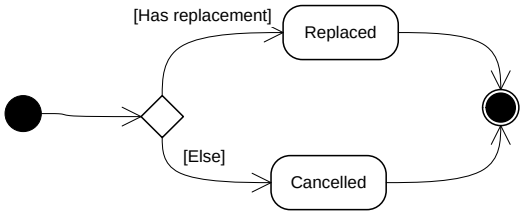

Fig. 6. Cancelled sub states for arrival

A cancelled arrival can **replaced**, which means that alighting is cancelled and replaced by an arrival of another vehicle journey, or **cancelled**, which means that the possibility to alight at the stop is not possible or not permitted.

3.3.2 Departure

A *departure* contains the last valid plan for a departure. In addition, a departure contains supplementary data based on vehicle monitoring.

3.3.2.1 Timetabled Departure

The timetabled earliest departure time and the timetabled point in a journey pattern describe when and where the departure was initially scheduled.

Through the departure object it is also possible to find the relevant destination texts that shall be presented.

3.3.2.2 Target Departure

The targeted earliest departure time and the journey pattern point intended for the departure are initially equal to the timetable data, but can be updated in real-time as a consequence of short term re-planning, optimisation or forecasts.

The targeted journey pattern point is the most recent plan to what stop point a vehicle will depart from.

3.3.2.3 Estimated Departure

The estimated departure time is the mean time as forecasted by the integrator system or an external system. This time is the most recent best guess.

3.3.2.4 Observed Departure

The observed departure time is the actual departure time as recorded by a vehicle monitoring system. The integrator system should not set this time unless there is a progress report received with an explicit time.

3.3.2.5 States

The figure below shows the combined states of a dated departure (white) and a monitored departure (grey).

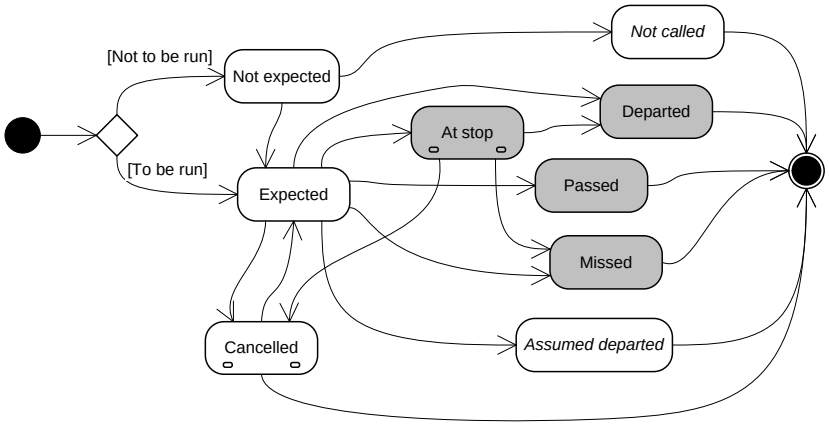

Fig. 7. Departure states

Initially, a dated departure can be either **not expected** or **expected**. This has to do with if the specific departure originally was scheduled to be called at or to be ordered. If a dated vehicle journey is not expected, then all departures are also not expected. On the other hand, if a vehicle journey is scheduled to be expected, there may still be individual departures that shall be ordered and therefore have the state not expected.

When a vehicle has arrived the dated departure receives the state **at stop**. When the vehicle leaves the stop it receives the state **departed**. If the vehicle passes without stopping, the departure will be flagged as **passed**.

The **not called** state and **missed**-state will be set for all departures where a departed or passed-state is missing if the integrator system receives a progress report further down the route, A missed-state means that a call was expected but there was no observation recorded. All the final states can be used for clear-down of a departure board.

3.3.2.6 Implicit states

If an expected dated arrival is not cancelled and not is monitored, it will at some point in time be considered as **assumed departed**. A departure that was scheduled to be ordered is not expected to be called until explicitly ordered. Usually, a dated departure that is not expected shall be presented with the conditions to order it or not be presented at all. If a not expected departure is never called, it will at some point in time be considered **not called**.

3.3.2.7 Cancelled sub-state

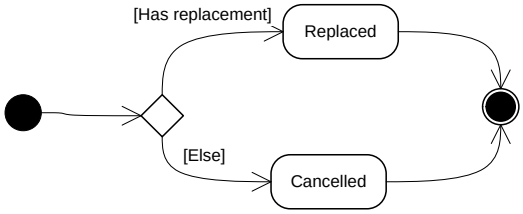

Fig. 8. Cancelled sub states for departure

A cancelled departure can be **replaced**, which means that boarding is cancelled and replaced by a departure of another vehicle journey, or **cancelled**, which means that the possibility to board at the stop is not possible or not permitted.

3.4 Connections

3.4.1 Connection

A *connection* is the intended meeting of two vehicles, usually for the purpose of exchanging passengers. The arriving vehicle is called a *feeder*, and the departing vehicle is called a *fetcher*. A connection is always one-way. If the vehicles shall exchange passengers in both directions, this is expressed as two connections.

A *connection* in NOPTIS means a dependency between an arrival on a dated vehicle journey and a departure on another dated vehicle journey. A connection specifies the maximum time the fetcher should wait. This value can be updated in real time through control actions.

3.4.2 States

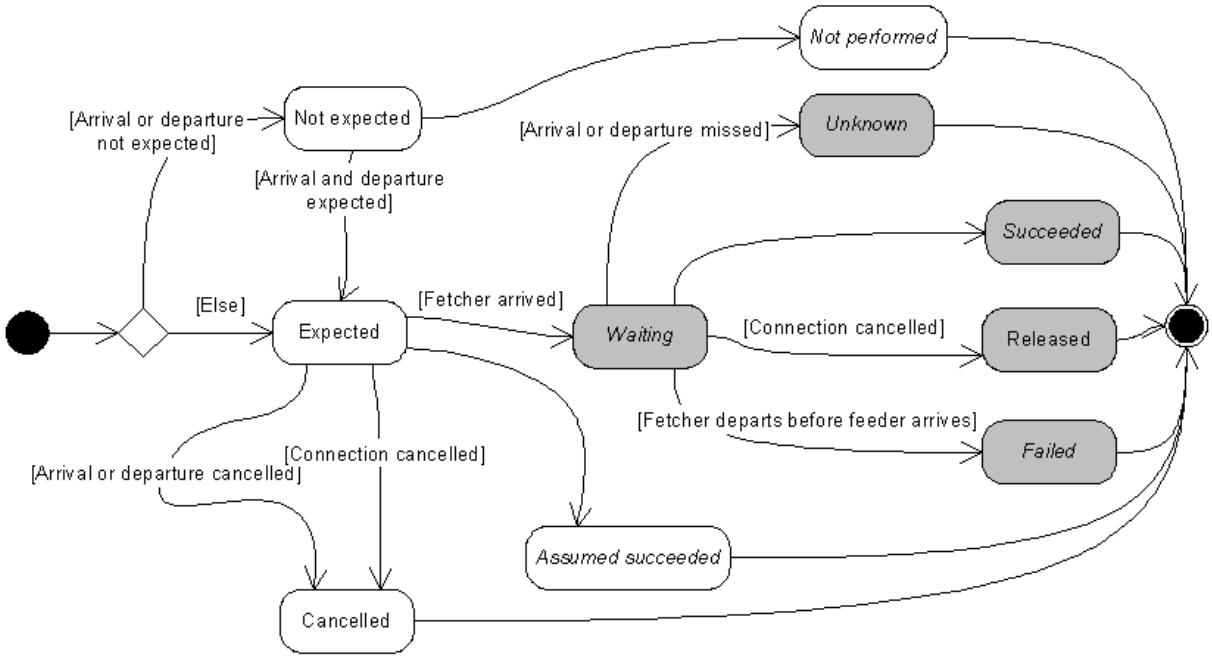

Fig. 9. Connection states

Connection states are affected by short term operation control, i.e. control actions..

If the feeder arrival or fetcher departure is not expected, then the connection itself has the initial state **not expected**, otherwise it is initially **expected**.

Not expected connections may become **expected** as a result of a Journey Ordering control action.

A connection can be **cancelled** for several reasons; as a consequenceof a control action where the feeder arrival or the fetcher departure is cancelled, or the connection itself is cancelled.

An **expected** connection has the implicit state **waiting** as soon as the fetcher has arrived at the stop and its planned earliest departure time has passed.

A monitored connection can be **released** if the fetcher no longer needs to wait for the feeder. This is accomplished with a Connection Cancellation control action.

3.4.2.1 Implicit Final States

Final states for connections are not set explicitly. This has to do with that it is difficult to know when a final state can be set during an operation day. Instead these can be deducted when analysing connections afterward:

If one or both of the involved dated vehicle journeys was never monitored and the connection state was **not expected**, its final state shall be interpreted as **not performed**; otherwise, if the connection state was **expected**, it is shall be interpreted as **assumed succeeded**.

The **failed** state can be deducted as when the fetcher has departed before the feeder's arrival (possibly with consideration to a minimal interchange time).

The **unknown** state can be deducted as when either the arrival or the departure has the state **missed**. In this case, it is not possible to conclude whether the connection succeeded or failed.

f none of the other final states apply then the final state shall be interpreted as **succeeded**.

3.5 Deviation Case

A *deviation case* is the container for all information that concerns a specific deviation. The first part of this chapter describes the different aspects of a deviation case.

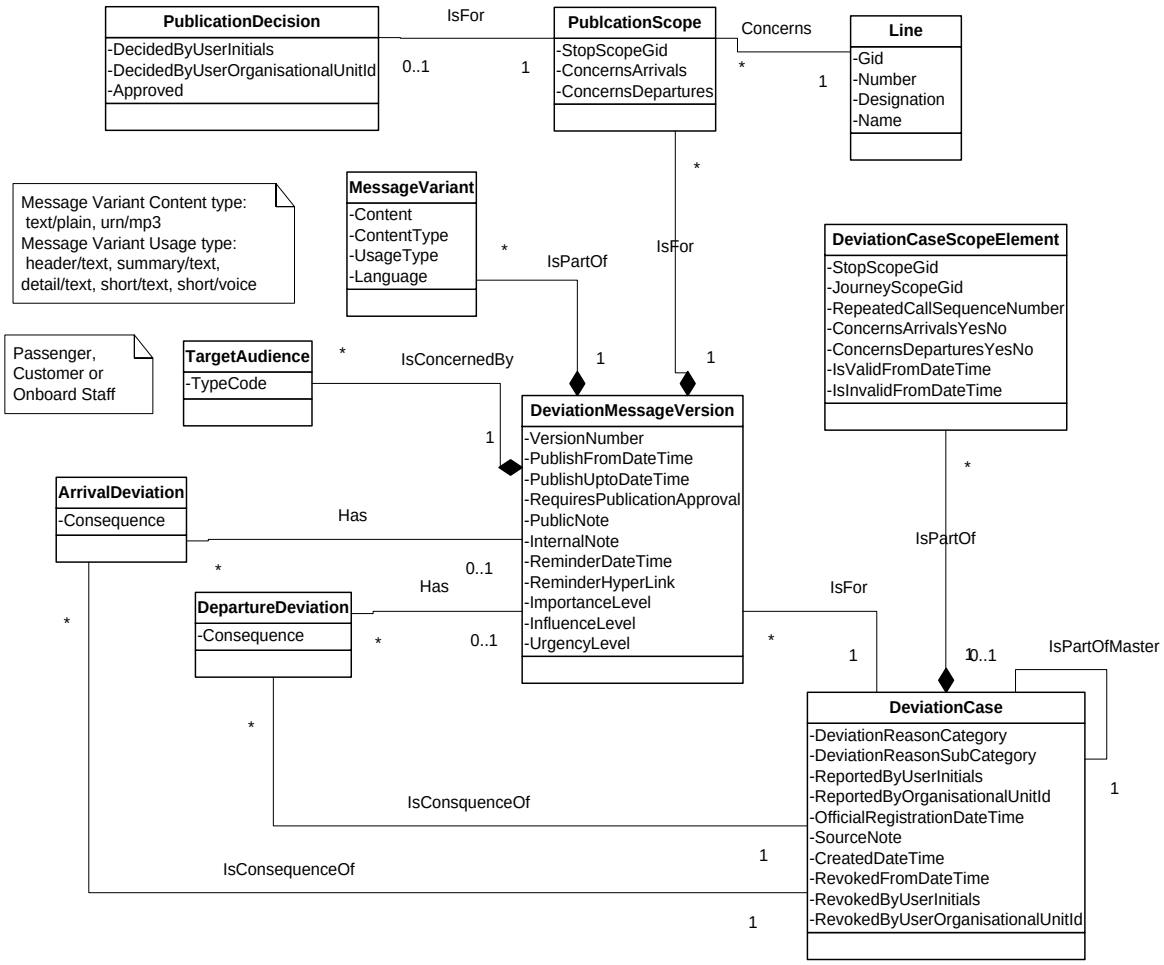

Fig. 10. Logical structure for a deviation case

The deviation case includes:

- The *identity* of the case, which also is exposed to both submitting systems and consuming systems.
- Administrative data about who has created the case and when the case has been created.
- The *reason* behind the deviation case.
- An optional *deviation message* with variants to suit different media.
- The *target audience* that shall have the message, e.g. passengers, drivers etc.
- The *priority* of the deviation message
- The *scope* to publish the deviation message in.
- An *operation action*, which can be either a *control action* that describe a short term temporary change in the operation, or an *information action* that is information only.
- The *scope* for the control action, i.e. what part of the operation it applies to.

These items are described in detail in the following sections.

3.5.1 Master- and Sub Cases

Deviation cases can be grouped, which can help to manage cases that relates to each other. The purpose of the grouping is entirely up to the user.

In ROI the grouping is exposed to other applications. This means that all applications using ROI also will see what deviation cases that are grouped together.

3.5.2 Reason for Deviation

Behind a deviation there is a reason, and it is generally a good idea to provide this reason, because it sometimes helps both staff and passengers to conclude about the consequences, and act accordingly. Examples of reasons for a deviation are vehicle breakdown, shortage of vehicles, no driver, accident, driving the wrong way, assault, strike, traffic congestion, weather conditions.-

3.5.3 Scope

A *scope* defines a part of the operation, i.e. a set of arrivals and departures. The set can be expressed in many ways, which not is presented here. Details are found in the RII XML-schemas.

3.5.3.1 Types of Scope

A deviation case has two different types of scope:

- The *operation actions scope*, which is the scope that applies for control actions and information actions. This scope defines what part of the operation that is directly affected. Each control action has its special type of operation action scope.
- The *publication scope*, which is the scope for deviation messages. This scope defines what passengers that are affected by the deviation, i.e. to where the deviation message shall be distributed or announced. This is a general expression containing one or several lines. For each line, it is possible to limit the scope to certain stop points.

3.5.3.2 Scope Categories

In many cases, information about a deviation is intended for persons needing information about departures only or arrivals only. A scope is therefore divided into two main categories:

- Departures.
- Arrivals.

By pointing out the arrivals and departures as separate entities, information can be addressed specifically to boarding or alighting passengers.

3.5.4 Closing a Deviation Case

To close a deviation case means to shorten its validity to 'now'. This has the effect that the end validity of the deviation messages in the case is passed, and that the deviation messages should not be presented any longer. Some types of control actions are also affected by closing a deviation case.

When a deviation has expired, it is sometimes not sufficient to just remove the deviation message, but the passengers must be informed that the operation is normal again. This can be made in two different ways:

- By updating the existing deviation case with a new deviation message, and a new validity. This method takes effect immediately.
- By ending the original case and creating a new deviation case with the new information. In this method, the user selects the time to finish the existing case and when the new case is valid from (usually the same time).

In both cases, the effect is that the original information is replaced from the start time of the new validity.

3.6 Deviation Message

3.6.1 Target Audience

A target audience classifies a deviation message in terms of what roles the receivers have in the public transport system.

- **Customer**: a person who is interested in information about public transportation but is not a passenger, i.e. not interested in information about deviations.
- **Passenger**: a person that is using the public transport system in any way, e.g. intending to travel in a near future, walking to the stop, waiting at a station, riding with a train, etc., i.e. a person that is interested of deviation information.
- **Onboard staff**: persons working onboard a public transport vehicle involved in passenger information, e.g. a bus driver, a ticket collector, a deck officer, etc.

3.6.2 Priority

The priority of a deviation message is made up of three components:

- Importance
- Influence
- Urgency

The purpose of this three dimensional classification, is to make it possible to let different client information systems take decisions about if, how, when and through what channels and media a deviation message shall be distributed.

In order to have a conformant interpretation of the three dimensions, it is recommended that the definition and use of the dimensions are consistent across all users of an integrator system.

3.6.2.1 Importance

This is how important a deviation message is in relation to other deviation messages. Importance is a relative entity that can support each presentation system to do the optimal within its limits. The importance gives a presentation system a hint of what message to present first or repeated more often or highlight or whatever method it takes to draw attention to the information.

3.6.2.2 Influence

This is a hint about how many persons that is influenced by a deviation. In a deviation message, the user can define a publication scope that tells what lines etc. a message shall be published on, but this does not tell how many people (customers, passengers) that are influenced. A deviation on a local bus line does not influence many people, while a deviation on a central metro line does.

With the level of influence it is possible to classify a deviation message in a number of generic levels of how many people that is influenced by a deviation. The level of influence can be used by presentation systems to select if a message shall be presented or not on specific media. For instance, messages that influence many people usually are presented in media for mass communication.

3.6.2.3 Urgency

This is a hint of how fast information about a deviation shall reach the receivers. The level of urgency defines the priority for distribution of information. When the level of urgency is high, this may imply that the information shall be distributed using fast ways to reach the receivers, e.g. SMS to mobile phones, speaker calls at stations etc.

3.6.3 Message Variants

Despite the general principle of not providing device specific formatting of information, it is not possible for a one-size-fits-all format either. An example is when information shall be provided as both text and audio. Although there exists ways to synthesise speech from text, this is not always available, so instead text and audio have to be provided in parallel. Another example is that some media is forgivable for verbose information, while other requires that the information is very compact. To deal with this, it is possible to provide optional message variants of the content that shall be presented.

To each deviation message, any number of message variants can be appended. The message variants to provide in each case are up to the providing system to decide. However, in order to provide presentation systems in a consistent way, it is recommended that the user defines a standard set of message variants that all providers of deviation reports shall support.

The message variants have three main properties:

- Some content.
- A content type that tells the consumer how to interpret the content. Example: 'text/plain' or 'urn/mp3', the last one describing that the content is a hyperlink to an mp3 audio file.
- A usage type that tells the consumer of recommended usage of the content, i.e. if it is adapted to media with some limitation, for instance short texts adapted for displays with limited number of characters. Example: 'audio/short', 'text/long'.

RII defines a number of predefined message variants, and it is possible to add customised message variants if the predefined variants are not sufficient.

3.6.4 Message Approval

There is a distinction between submitting a deviation report and the approval of publication of the deviation message in the report. The idea is that it shall be easy for many involved persons to provide information that something has happened but only possible for authorised persons to approve publication of information about it.

When a user submits a deviation message where the user have publication rights for the lines as defined in the deviation case publication scope, the publication can be approved in the same report.

A user can also submit a deviation message with a scope that includes lines for which the user does not have publication rights. In this case, the deviation message will only be flagged as approved for the lines the user has publication rights for. For the other lines, the deviation message will be flagged as not approved for publishing. Such messages shall not be presented to the selected target audience.

4 Message Reference

This chapter describes the payload messages in ROI. In addition there might be message types that are specific for the transport mechanism.

4.1 Root Elements

The root elements are the wrappers defining the message sequence to and from the integrator system.

4.1.1 Message Types

4.1.1.1 To Integrator Messages[1]

The message types that an external system can send to the integrator system.

Category	Description
Subscription Messages	Messages used to administrate the client's subscriptions.

In addition to the payload content, some of the attributes and elements of the root elements might be inherited from the transport mechanism used. The current implementation inherits from the general XML-stream transport mechanism.

[1] Named ToPubTransMessages in XSD for backward compatibility reasons.

4.1.1.2 From Integrator Messages[2]

The message types that the integrator system sends to an external system. These message types can be divided into a number of categories.

Category	Description
Vehicle Journey Create Events	Messages that inform the client to create objects in the client's production plan cache. These messages are sent during recovery and synchronisation. They contain the object's last known state in the production plan.
Vehicle Journey Delete Events	Messages that inform the client to remove objects in the client's production plan cache
Vehicle Journey Update Events	Messages that inform the client to update objects in the client's production plan cache.
Deviation Case Create Events	Messages that inform the client to create objects in the client's deviation case cache. These messages are sent during recovery and synchronisation. They contain the object's last known state.
Deviation Case Update Events	Messages that inform the client to update objects in the client's deviation case cache.
Other Events	Messages that inform the client to create or update other objects in the client's cache.
Subscription Messages	Messages used to administrate the client's subscriptions.

In addition to the payload content, some of the attributes and elements of the root elements might be inherited from the transport mechanism used. The current implementation inherits from the general XML-stream transport mechanism.

4.2 Vehicle Journey Create Events

Messages from the integrator system that inform the client to create objects in the client's production plan cache

4.2.1 Message Types

4.2.1.1 Vehicle Journey Create Event

This is a message that informs the client to create a dated vehicle journey object in the client's production plan cache. If the journey is already in progress when this element is transferred, a monitored vehicle journey object will also be provided.

[2] Named FromPubTransMessages in XSD for backward compatibility reasons.

4.2.1.2 Arrival Create Event

This is a message that informs the client to create an arrival object in the client's production plan cache.

4.2.1.3 Departure Create Event

This is a message that informs the client to create a departure object in the client's production plan cache.

4.2.1.4 Connection Create Event

This is a message that informs the client to create a connection object in the client's production plan cache.

4.3 Vehicle Journey Delete Events

Messages from the integrator system that inform the client to remove objects in the client's production plan cache because the production plan has changed. This is only applicable to objects valid tomorrow or later. Objects valid today can not be removed with this mechanism. They can instead be cancelled, which is accomplished by updating their state, see Vehicle Journey Update Events.

4.3.1 Message Types

4.3.1.1 Dated Vehicle Journey Delete Event

This is a message that informs the client to remove a dated vehicle journey object and the related arrival and departure objects in the client's production plan cache.

4.3.1.2 Connection Delete Event

This is a message that informs the client to remove a connection object in the client's production plan cache.

4.4 Vehicle Journey Update Events

Messages from the integrator system that inform the client to update objects in the client's production plan cache

4.4.1 Message Types

4.4.1.1 Vehicle Journey Update Event

This is a message that informs the client to update a dated or monitored vehicle journey object in the client's production plan cache. The only possible update action for a dated vehicle journey object is to change the dated vehicle journey state or the origin name. When a dated vehicle journey is first monitored a message containing a monitored vehicle journey object with a reference to the dated vehicle journey is sent to the client. Possible update actions for a monitored vehicle journey are vehicle journey state changes, prediction state changes as well as information about which physical vehicle that work a vehicle journey. During periods when prediction is unreliable messages will contain information about the last recorded arrival or departure.

4.4.1.2 Arrival Update Event

This is a message that informs the client to update an arrival object in the client's production plan cache. The update could consist of changing the arrival state, changing the target arrival time or changing the target journey point. It could be an update of the estimated arrival time or a report of the actual observed arrival time.

4.4.1.3 Arrival Deviation Event

This is a message that informs the client to add, update or remove an arrival deviation object in the client's production plan cache. An arrival deviation clarifies the consequence of a certain control action for an arrival. There can be multiple arrival deviations that are valid at the same time for an arrival. In some situations arrival deviations for stops that are not subscribed will also be sent. This is the case when these deviations affect departures at subscribed stops.

4.4.1.4 Departure Update Event

This is a message that informs the client to update a departure object in the client's production plan cache. The update could consist of changing the departure state, changing the destination information, target departure time or changing the target stop point. It could be an update of the estimated departure time or a report of the actual observed departure time.

4.4.1.5 Departure Deviation Event

This is a message that informs the client to add, update or remove a departure deviation object in the client's production plan cache. A departure deviation clarifies the consequence of a certain control action for a departure. There can be multiple departure deviations that are valid at the same time for a departure. In some situations departure deviations for stops that are not subscribed will also be sent. This is the case when these deviations affect arrivals at subscribed stops.

4.4.1.6 Vehicle Journey Deviation Event

This is a message that informs the client to add, update or remove a vehicle journey deviation object in the client's production plan cache. A vehicle journey deviation clarifies the consequence of a certain control action for a vehicle journey. There can be multiple vehicle journey deviations that are valid at the same time for a vehicle journey.

Vehicle Journey Deviations hold information that apply to all arrivals and departures and should therefore be considered in addition to the Arrival and Departure Deviations that apply for a specific stop.

4.4.1.7 Connection Update Event

This is a message that informs the client to update a connection object in the client's production plan cache. The update could concern a prolonged wait time or a change of status.

4.5 Deviation Case Create Events

Messages from the integrator system that inform the client to create objects in the client's deviation case cache

4.5.1 Message Types

4.5.1.1 Deviation Case Create Event

This is a message that informs the client to create a deviation case object in the client's deviation case cache.

4.6 Deviation Case Update Events

Messages from the integrator system that inform the client to update objects in the client's deviation case cache

4.6.1 Message Types

4.6.1.1 Deviation Case Update Event

This is a message that informs the client to update a deviation case object in the client's deviation case cache. Updates are restricted to revoking the deviation case and to group it (or ungroup it) under a master deviation case.

4.6.1.2 Deviation Message Version Event

This is a message that informs the client to store a new version of a deviation message object in the client's deviation case cache.

4.6.1.3 Publication Decision Event

This is a message that informs the client to update a publication decision object in the client's deviation case cache to the supplied publication decision or to revoke a previous publication decision.

There can only be one approved publication decision for the same publication scope within a deviation message version at a certain time.. When a new version is approved then the previous approved version, if any, will be revoked.

4.7 Other Events

Messages from the integrator system that inform the client to update objects in the client's cache

4.7.1 Message Types

4.7.1.1 Assignment Event

This is a message that informs the client to create, update the state of, or revoke a block assignment, a vehicle journey assignment, a driver assignment or a duty assignment object in the client's cache.

4.7.1.2 Network Deviation Event

This is a message that informs the client to create or revoke a network deviation object in the client's cache. A network deviation concerns a network object such as a stop point, a station entrance or a bridging device. It is used to express that the network object is closed, that there is limited access to it or that there is some

special information such as a warning for passing trains that applies to the network object during a certain period of time. There can be multiple network deviations for the same network object.

4.8 Subscription Messages

These are messages sent to and from the integrator system that are used to administrate the clients' subscription and the synchronisation of data.

The integrator system retains subscription configuration and state for the client system until the subscription is actively terminated. The client is permitted to modify the scope of the subscription at any time by first terminating the current subscription and then establishing a new one. It is also possible to modify the scope of the subscription using the subscription update request.

4.8.1 Message Types

4.8.1.1 Subscription Request

This is a message from the client telling the integrator system that it wants to create a subscription.

The purpose of the subscription is to limit the processing load on the integrator system, the client system and the communication system in between them. By letting the client systems define their scope of interest, only real-time data concerning objects within this scope need to be transferred and processed in the client.

The subscription can be either journey or stop centric. For instance, a stop display control system will probably subscribe on the stops where displays are located

4.8.1.2 Subscription Response

This is a message from the integrator system telling the client that the subscription has been accepted.

4.8.1.3 Subscription Update Request

This is a message from the client telling the integrator system that it wants to update the scope of a subscription. The request must contain the complete scope of interest including those parts of the previous scope of interest that should be retained.

4.8.1.4 Subscription Update Response

This is a message from the integrator system telling the client that the subscription update has been accepted.

4.8.1.5 Subscription Termination Request

This is a message from the client telling the integrator system that it wants to terminate a subscription.

4.8.1.6 Subscription Termination Response

This is a message from the integrator system telling the client that the termination of subscription has been accepted.

4.8.1.7 Subscription Resume Request

This is a message from the client telling the integrator system that it wants to resume subscription after the connection has been broken and re-established.

4.8.1.8 Subscription Resume Response

This is a message from the integrator system telling the client that the subscription has resumed.

4.8.1.9 Subscription Error Response

This is a message from the integrator system telling the client that the subscription has not resumed. The subscription could not resume as requested from the requested start date/time, since data of interest has been purged from the production plan before it could be transferred. This could be the case if the subscriber has requested a start time so far back in time that data is already purged. It could also be the case if the subscriber requests a start time earlier than the start time of a previous request for the same subscription.

4.8.1.10 Subscription Error Report

This is a message from the integrator system telling the client that data of interest for a subscription has been lost since it was purged from the production plan before it could be transferred. This could be the case if the connection to a subscriber has been broken for several days, or if the client is not capable of receiving the flow of data fast enough. The subscription will continue with the available data, but the client's historical data will not be continuous.

4.8.1.11 Synchronisation Report

This is a message from the integrator system telling the client that the synchronisation of data for a subscription has reached a certain point in time. Additionally it will indicate if the recovery phase is complete or not.

5 Embedded Elements Reference

This chapter provides additional description of elements embedded in the requests.

5.1 Vehicle Journeys

5.1.1 Dated Vehicle Journey

5.1.1.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Operating Day Date	Attribute	The date of the day of public transport that this dated vehicle journey is associated with. This is not necessarily the same as the actual date. It is not uncommon for dated vehicle journeys starting after midnight, to be associated with the previous date.
GID	Attribute	The identifier of the dated vehicle journey. Unique among all dated vehicle journeys for the operating day.
Timetabled Start DateTime	Attribute	The long term planned start time.
Timetabled End DateTime	Attribute	The long term planned end time.
State	Attribute	A dated vehicle journey can be expected, not expected or cancelled.
Origin Name	Attribute	The full length form of the place where the vehicle journey originates.
Origin Short Name	Attribute	The short length form of the place where the vehicle journey originates.
Inform Passengers Condition	Attribute	Defines under which conditions passengers should be informed, See XSD for details.
Is Expected To Be Monitored	Attribute	Defines if real time reports should be expected or not for this dated vehicle journey.
Planned Type Code	Attribute	Provided for extra dated vehicle journeys as EXTRA, default is NORMAL.
Uses Named Journey Pattern Id	Attribute	Provided for extra dated vehicle journeys. Which named journey pattern the vehicle journey uses.
Reinforced Vehicle Journey Gid	Attribute	Provided for extra dated vehicle journeys that reinforce another vehicle journey. The GID value of the vehicle journey being reinforced.
Reinforced Dated Vehicle Journey Id	Attribute	Provided for extra dated vehicle journeys that reinforce another vehicle journey. The Id value for the dated vehicle journey being reinforced.

Name	Type	Description
Line	Element	Provided for service journeys. Details about the number and designation of the line this vehicle journey is on.
Direction of Line Ref	Element	Provided for service journeys. Contains a GID reference to the direction this vehicle journey is on.
Vehicle Operator	Element	Details about the vehicle operator that works this vehicle journey.

This element is part of the Vehicle Journey Create Event.

There is also a reduced version of this element only containing the instance id, timestamp, origin name, origin short name and vehicle journey state which is used in the Vehicle Journey Update Event.

5.1.2 Monitored Vehicle Journey

5.1.2.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
State	Attribute	After the vehicle journey becomes monitored (in progress) the monitored vehicle journey state overrides the dated vehicle journey state. See the Concept chapter for more details..
Passenger Level	Attribute	Describes if the vehicle is empty, seats are available, some passengers needs to stand or passengers are left behind at stops.
In Progress From DateTime	Attribute	When the vehicle journey was first monitored.
Final State DateTime	Attribute	When the vehicle journey reached its final state.
Prediction State	Attribute	NORMAL, LOSTCONTACT or UNRELIABLE. This gives a hint of the quality of the prediction. Observe that there is also a case where the technical system in the vehicle is out of order. This means that there will be no monitored vehicle journey instance associated with the affected dated vehicle journey. The monitored vehicle journey instance is created only after the vehicle is monitored as being in progress.
Dated Vehicle Journey Ref	Element	Contains an Id-reference to the concerned Dated Vehicle Journey instance, as well as a GID and date reference to the Dated Vehicle Journey.
Assigned Vehicle Ref	Element	A GID reference to the assigned physical vehicle. May be provided depending on configuration.
Assigned Vehicle	Element	Expanded data provided for subscriptions with option ExpandVehicleData=TRUE. Alternatively available by joining with view Vehicle in DOI.
Last Observed Call	Group	This is a reference to the last observed arrival or departure. Only provided when prediction is unreliable and information is available.

This element is part of the Vehicle Journey Create Event and the Vehicle Journey Update Event.

5.1.3 Arrival

5.1.3.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Timetabled Latest DateTime	Attribute	The long term planned latest arrival time.
Target DateTime	Attribute	The arrival time according to the latest valid plan. This is the time that should be exposed to the passengers. It is based on a combination of the estimated arrival time, control actions, connection protection and business rules according to the current configuration.
Estimated DateTime	Attribute	The arrival time according to the latest prediction. This is the estimated arrival time.
Observed DateTime	Attribute	The time when the vehicle actually arrived at the stop.
State	Attribute	See the Concept chapter for more details about arrival state.
Type	Attribute	Denotes conditions for alighting at a stop.
Journey Pattern Sequence Number	Attribute	The sequence number of the concerned point in journey pattern.
Visit Count Number	Attribute	Normally each stop is called only once on a Journey. However, some Journeys have such Journey Patterns that the same stop is called again. In such instances more information must be provided so that the central system can distinguish between the first call and consecutive calls. A call referring to the first time a stop is called occurs in the Journey Pattern has Visit Count = 1. The next time the same stop is called the value 2 should be used.
Dated Vehicle Journey Ref	Element	Contains an Id-reference to the concerned Dated Vehicle Journey instance, as well as a GID and date reference to the Dated Vehicle Journey.
Direction Of Line Ref	Element	A GID reference to the concerned direction of line.
Monitored Vehicle Journey Ref	Element	Contains an Id-reference to the concerned Monitored Vehicle Journey instance. Only provided for vehicle journeys in progress.
Target Journey Pattern Point Ref	Element	A GID reference to the platform where the arrival should take place according to the latest valid plan.

Name	Type	Description
Target Stop Point	Element	Expanded data provided for subscriptions with option Expand-StopData=TRUE if the journey pattern point is a stop point. Alternatively, stop point information can be obtained by joining view StopPoint and JourneyPatternPoint in DOI.
Timetabled Journey Pattern Point Ref	Element	A GID reference to the platform where the arrival should take place according to the long term plan. If this element is omitted then the timetabled journey pattern point is the same as the target journey pattern point.
Timetabled Stop Point	Element	Expanded data provided for subscriptions with option Expand-StopData=TRUE if the timetabled journey pattern point is different from the target journey pattern point and it is a stop point. Alternatively, stop point information can be obtained by joining view StopPoint and JourneyPatternPoint in DOI.
Observed Journey Pattern Point Ref	Element	A GID reference to the platform where the arrival did take place. If this element is omitted and the arrival state is ARRIVED, then the observed journey pattern point is the same as the target journey pattern point.
Observed Stop Point	Element	Expanded data provided for subscriptions with option Expand-StopData=TRUE if the observed journey pattern point is different from the target journey pattern point and it is a stop point. Alternatively, stop point information can be obtained by joining view StopPoint and JourneyPatternPoint in DOI.

This element is part of the Arrival Create Event.

There is also a reduced version of this element containing the instance id, timestamp, target date time, state and the attributes and elements that have changed since last time the event was sent. The reduced version is used in the Arrival Update Event.

5.1.4 Departure

5.1.4.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Timetabled Earliest DateTime	Attribute	The long term planned earliest departure time.

Name	Type	Description
Target DateTime	Attribute	The departure time according to the latest valid plan. This is the time that should be exposed to the passengers. It is based on a combination of the estimated departure time, control actions, connection protection and business rules according to the current configuration.
Estimated DateTime	Attribute	The departure time according to the latest prediction. This is the estimated departure time.
Observed DateTime	Attribute	The time when the vehicle actually arrived at the stop.
State	Attribute	See the Concept chapter for more details about arrival state.
Type	Attribute	Denotes conditions for boarding at a stop.
Journey Pattern Sequence Number	Attribute	The sequence number of the concerned point in journey pattern.
Visit Count Number	Attribute	Normally each stop is called only once on a Journey. However, some Journeys have such Journey Patterns that the same stop is called again. In such instances more information must be provided so that the central system can distinguish between the first call and consecutive calls. A call referring to the first time a stop is called occurs in the Journey Pattern has Visit Count = 1. The next time the same stop is called the value 2 should be used.
Dated Vehicle Journey Ref	Element	Contains an Id-reference to the concerned Dated Vehicle Journey instance, as well as a GID and date reference to the Dated Vehicle Journey.
Direction Of Line Ref	Element	A GID reference to the concerned direction of line.
Monitored Vehicle Journey Ref	Element	Contains an Id-reference to the concerned Monitored Vehicle Journey instance. Only provided for vehicle journeys in progress.
Target Journey Pattern Point Ref	Element	A GID reference to the platform where the arrival should take place according to the latest valid plan.
Target Stop Point	Element	Expanded data provided for subscriptions with option ExpandStopData=TRUE if the journey pattern point is a stop point. Alternatively, stop point information can be obtained by joining view StopPoint and JourneyPatternPoint in DOI.
Timetabled Journey Pattern Point Ref	Element	A GID reference to the platform where the arrival should take place according to the long term plan. If this element is omitted then the timetabled journey pattern point is the same as the target journey pattern point

Name	Type	Description
Timetabled Stop Point	Element	Expanded data provided for subscriptions with option ExpandStopData=TRUE if the timetabled journey pattern point is different from the target journey pattern point and it is a stop point. Alternatively, stop point information can be obtained by joining view StopPoint and JourneyPatternPoint in DOI
Destination Display Ref/Stop Area Ref	Element	Reference to row in view DestinationDisplay of DOI or Reference to the end stop area of the vehicle journey.
Destination Display/Stop Area	Element	DestinationDisplay: Expanded data provided for subscriptions with option ExpandDestinationData=TRUE. Alternatively available by joining with view DestinationDisplay in DOI. StopArea: Expanded data provided for subscriptions with option ExpandStopData=TRUE Alternatively, stop area information can be obtained from DOI.
Service Requirement Ref	Element	Omitted only for extra dated vehicle journeys where the service requirements are unknown.
Service Requirement	Element	Expanded data provided for subscriptions with option ExpandServiceRequirementData=TRUE. Alternatively available by joining with view ServiceRequirement in DOI.
Observed Journey Pattern Point Ref	Element	A GID reference to the platform where the arrival did take place. If this element is omitted and the departure state is DEPARTED, then the observed journey pattern point is the same as the target journey pattern point.
Observed Stop Point	Element	Expanded data provided for subscriptions with option ExpandStopData=TRUE if the observed journey pattern point is different from the target journey pattern point and it is a stop point. Alternatively, stop point information can be obtained by joining view StopPoint and JourneyPatternPoint in DOI.

This element is part of the Departure Create Event.

There is also a reduced version of this element containing the instance id, timestamp, target date time, state and the attributes and elements that have changed since last time the event was sent. The reduced version is used in the Arrival Update Event.

5.1.5 Connection

5.1.5.1 Content

Name	Type	Description

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Wait For Feeder Until DateTime	Attribute	This is how long the fetcher must wait if the feeder is delayed.
State	Attribute	Expected waiting, released, succeeded, failed, cancelled
Is Continuing Vehicle	Attribute	If the connection concerns two vehicle journeys worked by the same vehicle then the passenger can stay seated.
Min Change Duration	Attribute	The shortest time to transfer from the feeder to the fetcher.
Is Exposed To Staff	Attribute	Shows if information concerning the connection is exposed to the onboard staff.
Is Exposed to Passengers	Attribute	Shows if information concerning the connection is exposed to the public.
Arrival Ref	Element	Contains a reference to the concerned feeder arrival instance.
Departure Ref	Element	Contains a reference to the concerned fetcher departure instance.

5.1.6 Vehicle Journey Deviation

A deviation that concern all arrivals and departures on a vehicle journey is exposed as a vehicle journey deviation instead of individual arrival and departure deviations.

5.1.6.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Is Valid	Attribute	This attribute is used to show if a deviation is valid or not. If not valid, it should be removed from the client's cache.
Consequence	Attribute	A categorisation of the deviation.
Dated Vehicle Journey Ref	Element	Contains a reference to the concerned dated vehicle journey instance.
Deviation Case Ref	Element	Contains a reference to the underlying deviation case.
Deviation Message Version Ref	Element	Contains a reference to the underlying deviation message version.
Deviation Message	Element	Contains the text of the public note. Redundant information provided depending on subscription options.

5.1.7 Arrival Deviation

5.1.7.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Is Valid	Attribute	This attribute is used to show if a deviation is valid or not. If not valid, it should be removed from the client's cache.
Consequence	Attribute	A categorisation of the deviation, i.e. change of platform, , broken connection[3],, change of vehicle[4]
Arrival Ref	Element	Contains a reference to the concerned arrival instance.
Alternate Arrival Ref	Group	Contains a reference to the vehicle journey and stop of the concerned arrival instance. Only provided in the case when this deviation affects a previous departure from another stop, and the client subscribes to that departure. In those cases this extra data might be needed to present meaningful information since that subscriber might not have access to data describing this arrival in detail
Deviation Case Ref	Element	Contains a reference to the underlying deviation case.
Deviation Message Version Ref	Element	Contains a reference to the underlying deviation message version.
Deviation Message	Element	Contains the text of the public note. Redundant information provided depending on subscription options.

5.1.8 Departure Deviation

5.1.8.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Is Valid	Attribute	This attribute is used to show if a deviation is valid or not. If not valid, it should be removed from the client's cache.

[3] If a published connection fails

[4] Passengers must leave the current vehicle and board a replacing vehicle if there is a change of vehicle on a vehicle journey in progress.

Name	Type	Description
Consequence	Attribute	A categorisation of the deviation, i. e. change of platform, change of service level[5].
Departure Ref	Element	Contains a reference to the concerned departure instance.
Alternate Departure Ref	Group	Contains a reference to the vehicle journey and stop of the concerned departure instance. Only provided in the case when this deviation affects a later arrival at another stop, and the client subscribes to that arrival. In those cases this extra data might be needed to present meaningful information since that subscriber might not have access to data describing this departure in detail.
Deviation Case Ref	Element	Contains a reference to the underlying deviation case.
Deviation Message Version Ref	Element	Contains a reference to the underlying deviation message version.
Deviation Message	Element	Contains the text of the public note. Redundant information provided depending on subscription options.

[5] If the service level differs from what is expected. It could concern wheelchair access, low floor access, restrictions in capacity for instance.

5.2 Network Deviations

5.2.1 Network Deviation

This concerns stop points, station point entrances and bridging devices such as elevators etcetera.

5.2.1.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
Status	Attribute	Status could be closed, limited access or a warning.
Valid From DateTime	Attribute	Start of period of time that the status applies.
Invalid From DateTime	Attribute	End of period of time that the status applies.
Network Object Ref	Group	Choice of Stop Point Ref, Bridging Device Ref or Station Entrance Point Ref.
Deviation Case Ref	Element	Contains a reference to the underlying deviation case.
Deviation Message Version Ref	Element	Contains a reference to the underlying deviation message version.
Deviation Message	Element	Contains the text of the public note Redundant information provided depending on subscription options.

5.3 Assignments

The states **assigned** and **signed on** are recorded in separate Assignment instances. The state applies from the time indicated by attribute Valid From Date Time. The attribute Invalid From Date Time marks the time from when the opposite state (***unassigned/signed off***) starts to apply or when the assignment instance was overridden by another assignment instance.

Observe that the attribute Invalid From Date Time is not provided in advance. Instead it is set when this Assignment instance is overridden by another Assignment instance, or the opposite state starts to apply.

A Vehicle can only be associated with one assignment instance having state **signed on** and no invalid from date time.

A Vehicle can be associated with several assignment instances having state **assigned** and no invalid from date time.

A Block/Vehicle Journey/Duty/Driver can only be associated with one assignment instance with no invalid from date time.

5.3.1 Block Assignment

5.3.1.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
State	Attribute	State could be assigned or signed on.
Operating Day Date	Attribute	The date of the day of public transport that the referred block is associated with.
Valid From DateTime	Attribute	Start of period of time when this assignment instance is valid. Do not confuse this with the start and end time of the block itself.
Invalid From DateTime	Attribute	End of period of time when this assignment instance is valid. Do not confuse this with the start and end time of the block itself.
Block Ref	Element	Contains a reference to the underlying block.
Assigned Vehicle Ref	Element	Contains a reference to the assigned physical vehicle.

5.3.2 Vehicle Journey Assignment

5.3.2.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
State	Attribute	State could be assigned or signed on.
Operating Day Date	Attribute	The date of the day of public transport that the vehicle journey is associated with.
Train Size Car Count	Attribute	Number of cars in train.
Valid From DateTime	Attribute	Start of period of time when this assignment instance is valid. Do not confuse this with the start and end time of the vehicle journey itself.
Invalid From DateTime	Attribute	End of period of time when this assignment instance is valid. Do not confuse this with the start and end time of the vehicle journey itself.
Dated Vehicle Journey Ref	Element	Contains a reference to the underlying vehicle journey.
Assigned Vehicle Ref	Element	Contains a reference to the assigned physical vehicle.

5.3.3 Duty Assignment

5.3.3.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
State	Attribute	State could be assigned or signed on.
Operating Day Date	Attribute	The date of the day of public transport that the referred duty is associated with.
Valid From DateTime	Attribute	Start of period of time when this assignment instance is valid. Do not confuse this with the start and end time of the duty itself.
Invalid From DateTime	Attribute	End of period of time when this assignment instance is valid. Do not confuse this with the start and end time of the duty itself.
Duty Ref	Element	Contains a reference to the underlying duty.
Assigned Employee Ref	Element	Contains a reference to the assigned employee.

5.3.4 Driver Assignment

5.3.4.1 Content

Name	Type	Description
Id	Attribute	A unique identity for this instance of the object.
Timestamp	Attribute	The time when the instance was created or last modified in the integrator system.
State	Attribute	State could be assigned or signed on.
Operating Day Date	Attribute	The date of the day of public transport that this assignment is associated with.
Valid From DateTime	Attribute	Start of period of time when this assignment instance is valid
Invalid From DateTime	Attribute	End of period of time when this assignment instance is valid.
Employee Ref	Element	Contains a reference to the employee.
Assigned Vehicle Ref	Element	Contains a reference to the assigned physical vehicle.

5.4 Subscriptions

5.4.1 Subscription Request

5.4.1.1 Content

Name	Type	Description
Message Id	Attribute	A unique identity for request.
Start Utc Date Time	Attribute	Data that is no longer valid after this point in time should be excluded. Thus, vehicle journeys with an end date/time before this point in time should be excluded. If this attribute is omitted then the current time will be used as the start date/time.
Vehicle Journey Event Selection	Element	Defines the scope of and the look-ahead window for vehicle journey events. If omitted, then no Vehicle Journey Create Events, Vehicle Journey Delete Events or Vehicle Journey Update Events will be sent from the integrator system in this subscription.
Deviation Case Event Selection	Element	Defines the scope of deviation case events. If omitted no Deviation Case Create Events or Deviation Case Update Events will be sent from the integrator system in this subscription.
Assignment Event Selection	Element	Defines that assignment events should be sent. If omitted no Assignment Events will be sent from the integrator system in this subscription.
Network Deviation Event Selection	Element	Defines that network deviation events should be sent. If omitted no Network Deviation Events will be sent from the integrator system in this subscription

5.4.2 Subscription Update Request

5.4.2.1 Content

Name	Type	Description
Message Id	Attribute	A unique identity for request.
Subscription Id	Attribute	The subscription to be resumed..
Vehicle Journey Event Selection	Element	Defines the scope of and the look-ahead window for vehicle journey events. If omitted, then no Vehicle Journey Create Events, Vehicle Journey Delete Events or Vehicle Journey Update Events will be sent from the integrator system in this subscription.
Deviation Case Event Selection	Element	Defines the scope of deviation case events. If omitted no Deviation Case Create Events or Deviation Case Update Events will be sent from the integrator system in this subscription.

Name	Type	Description
Assignment Event Selection	Element	Defines that assignment events should be sent. If omitted no Assignment Events will be sent from the integrator system in this subscription.
Network Deviation Event Selection	Element	Defines that network deviation events should be sent. If omitted no Network Deviation Events will be sent from the integrator system in this subscription

5.4.3 Subscription Resume Request

5.4.3.1 Content

Name	Type	Description
Message Id	Attribute	A unique identity for request.
Subscription Id	Attribute	The subscription to be resumed.
Start Utc Date Time	Attribute	Data that is no longer valid after this point in time should be excluded from the subscription. Thus, vehicle journeys with an end date/time before this point in time should be excluded. If omitted then the subscription is resumed from the last confirmed time, which means that no data will be lost.
Synchronised Up to Utc DateTime	Attribute	This attribute should contain the value from the last received Synchronisation Report. It could also be used to force the synchronisation to start from an earlier time, if previously received data has been lost. This attribute is only optional if the attribute Start Utc Date Time is provided. Observe that if this attribute is left out and the StartUtcDateTime is back in time, then previously sent data might be retransmitted.

5.4.4 Subscription Termination Request

5.4.4.1 Content

Name	Type	Description
Message Id	Attribute	A unique identity for request.
Subscription Id	Attribute	The subscription to be terminated. If this attribute is omitted then all current subscriptions held by the requesting Peer should be terminated.

Appendixes

Using XML-streams

XML-streams are a simple and efficient way to exchange XML-data over a TCP-connection. All messages are sent as XML elements inside an indefinitely long XML document. The end tag of the XML document is transmitted just prior to the shutdown of the TCP/IP connection.

Terminology

The following terminology is used in this appendix.

Term	Meaning
Client	The peer that is configured to initiate (and at failure re-initiate) the communication. This could be either the integrator system or the External system.
External system	The external application that is communicating with the integrator system over the XML-stream.
Message batch	The sequence of messages sent from one peer to another.
Peer	The application at each end of the communication link, i.e. the integrator system and the External system.
Integrator System	The server application that an External system is communicating with over the XML-stream.
Request	A message sent by one peer to the other as a request to process data provided in the message.
Response	A message from the peer that processed a request sent to the requesting peer with information about the result of the request. If an error occurred, it will be an Error Response.
Server	The peer that responds to incoming requests to initiate the communication.

Concepts

XML-stream concept uses a generic concept of request and response messages sent as XML-formatted messages over a TCP-connection. The definition of the generic mechanism for message exchange is separated from the definition of actual payload.

The payload is hooked into the XML-stream message model by deriving the payload messages from the Message element type.

Requirements

In order to use XML-streaming, a peer must fulfil the requirements described in this section.

Communication

The peer must be able to communicate with the other peer over a TCP-connection. The TCP-connection must have sufficient bandwidth.

Security

For increased security, the XML-streams TCP-connection can be established using Secure Sockets Layer (SSL) or the newer Transport Layer Security (TLS). It is the system administrator that defines the requirements for using SSL/TLS.

If it is the connection itself that shall be protected, it is sufficient that only the server shall have a certificate. If it is required to authenticate the clients, each client must also have a certificate. If it is not required to identify them uniquely, only that a client is authorised, then the clients can all share the same certificate.

XML Parser Type

A peer application must use an XML-parser that can deliver each element for further processing as they arrive. This means that it should be a SAX-parser or similar. A DOM-parser will not work, because it requires that a complete document is loaded before further processing can occur.

Configuration

Before configuring the communication, it has to be decided which peer that shall act as a client.

Server Configuration

The system administrator decides the *IP address* and *port number* where the server will be available. When running, the server process shall listen on this port for incoming TCP/IP-connection requests. The server should also be configured with a unique *peer id*.

Client Configuration

The client needs to be configured with the servers *IP address* and *port number*. The client should also be configured with a unique *peer id*.

Data encoding

Definition and detection of the text encoding of the XML-stream should follow relevant standards for XML-documents. The used encoding must be agreed for each implementation. Missing definition of encoding will be interpreted as UTF-8.

Initialisation

Example of Initialisation

This example shows how a session is initialised for the public interfaces, RII.

Client Request

The client always initialises the communication by opening a TCP/IP socket to the server on the port defined for the specific installation.

When the TCP/IP connection is established, the client starts the streaming by sending the initial part of the XML-stream.

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

This is then followed by the initial root element:

```
<RII:ToPubTransMessages
  xmlns:RII="http://www.usergroup.pubtrans.com/RII/2.0"
  xmlns:PT="http://www.usergroup.pubtrans.com/PT/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.usergroup.pubtrans.com/RII/2.0 RII-
ToPubTrans.xsd"
  PeerId="OCS_H1"
  LastProcessedMessageId="3976"
  DocumentLayoutVersion="2.0.0"
  MaxMessageInterval="PT60S"
>
```

Server Response

If the client is authorised, the server accepts the TCP-connection and replies in a similar way:

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

This then is followed by:

```
<RII:FromPubTransMessages
  xmlns:RII="http://www.usergroup.pubtrans.com/RII/2.0"
  xmlns:PT="http://www.usergroup.pubtrans.com/PT/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.usergroup.pubtrans.com/RII/2.0 RII-
FromPubTrans.xsd"
  PeerId="PubTrans01"
  LastProcessedMessageId="6612518"
  DocumentLayoutVersion="2.0.0"
  MaxMessageInterval="PT60S"
>
```

Attributes

Some of the attributes in the samples above are specific for the XML-stream transport mechanism, and other are specific to the NOPTIS interface. Declaration of namespaces and document layout version is specific for

each interface; peer id, last processed message id and max message interval is specific for the XML-stream. The XML-stream attributes are explained below.

Peer Id

The *peer id* is the session identity of a peer. The peer id must be a unique identifier in the context of the other peer, and must be agreed upon for each implementation.

Last Processed Message Id

The *last processed message id* contains a reference to the last successfully received message from the opposite peer, i.e. on the parallel stream working in the opposite direction. This attribute is optional and can be omitted if no messages have been processed yet or if the peer does not support this function.

If a *last processed message id* is provided, the application at the other peer shall re-transmit only the messages that are newer than the message with the specified id. If the *last processed message id* is not provided it is up to the implementation to decide what messages that should be transmitted. Specifically, a decision must be taken for how long time back in time the applications shall recover the communication.

Max Message Interval

If no message has been sent after half this time interval, an Idle message must be sent. If at the receiving end, no message has been received within the time interval, the receiver shall try to send an error message followed by the closing root element and then terminate the connection. It is the client that is responsible for re-establishing the connection regardless which peer that detects a time-out.

Termination

Normal

The message stream can be terminated by either peer. A peer request to terminate the connection by sending the closing element that corresponds to the opening element:

```
</ToPubTransMessages>
```

Then the peer waits for the other peer to respond with its closing element:

```
</FromPubTransMessages>
```

After this confirmation, the client can close the socket.

Abnormal

If an error occurs, the TCP/IP-connection should be closed and the client should try to re-establish it. Possible errors are defined in section 0.

It is recommended that all causes for abnormal termination are logged and that it is possible for a system administrator to be alerted in real time.

Message Exchange

Timestamp Sequence Order

Time stamped messages can be sent in any order. However, if two messages with timestamps concerning the same object arrive in the wrong time sequence, The integrator system may process the 'late' message different, because of the wrong time sequence.

Id Sequence Order

The Id of a message shall at least be unique per session. The Id can have any value. However, it is important that the External system internally keep track of its sequence of messages so it can understand how long processing has been acknowledged when the integrator system provides the *last processed message id* at reconnection.

Message Response

Some types of messages require responses. To optimise performance, a client can send messages without waiting for response on each message. Instead responses are processes asynchronously. Using asynchronous message responses put some requirements on the client application:

- The client application must keep a state for all sent but not yet acknowledged messages.
- If a response is not received within a configurable *response time out* time, the message shall be regarded as not received. Client behaviour in this case is implementation specific.

Keep-alive

If no other messages are sent, each peer must ensure that a connection is alive and send an idle message at the interval specified by the other peer in the attribute *max message interval*. This is explained in a section above.

Last processed message

It is possible for a peer to ask the other peer what message the other peer has last received and processed.

The other peer answers this message with a corresponding LastProcessedMessageResponse message containing the *last processed message id*.

Error Handling

Generally, both peers should be able to handle all error situations mentioned in this section.

TCP-connection Cannot Be Established

When a client is not connected to the server (regardless of the cause: closed, crashed, not connected yet), the client shall continuously try to connect to the server until the connection is established.

However, in order to reduce the load in the reconnection process the client should implement a retry-scheme for re-establishing the TCP-connection. Such a scheme can include retries for a specified number of times with a short delay between each attempt.

Error Writing On the TCP–Socket

In this case the peer closes the socket without sending any information. The other peer will detect the situation due to the timeout mechanism (if not reported by the socket API). The client then tries to re-establish the TCP-connection again.

SSL or TLS Protocol Error

This error indicates that the client cannot connect using SSL or TLS. In this case the client closes the socket without sending any information.

The client should log this error, and then try to re-establish the connection again.

Time Out

A peer shall monitor the time out of a connection. The connection should be considered broken if no message is received after a configurable *maximum idle- time* value.

When a peer detects a time out, it shall try to send an error message followed by the closing root element and then terminate the connection.

It is the client that is responsible for re-establishing the connection regardless which peer that detects a time-out.

Schema Version Not Supported

If a peer cannot support a schema version, the peer shall send a sequence consisting of the opening root element, an error report, the closing root element, and then close the TCP-connection. It is also recommended that the system administrator is alerted. This can be a configuration error or require a software update.

Error in Parsing and Validating XML Data

If a peer detects errors in XML parsing or XSD validation, the peer shall send an error message followed by the closing root element and then close the TCP-connection. If the schema definition supports it, an error response can be sent just before the closing root element.

Error in Data Format

Even if data is valid from a schema definition point of view, a peer can detect that the data provided for some other reason is not consistent with data format required. This error can for instance occur if reference data in a peer is erroneous or that data conflicts with integrity rules. In this case the peer shall send an error message.

This type of error can be an indication of that reference data is out of synch between the two peers. It is therefore recommended that these types of errors are logged and optionally alerted to the system administrator.

Error Types

Each interface has its own set of error codes. General types of errors using an XML-stream are listed below.

Error type	Error codes	Meaning
INTERNALERROR	100	An internal error has occurred.
TIMEOUT	101	
SERVICECLOSED	102	The service is closed, or will be closed.
NOTUNDERSTOOD	110	The XML is not understood because t is not well formed.
NOTUNDERSTOOD	111	The XML is not understood because it is not valid according to the schema definition.
NOTUNDERSTOOD	112	Requested schema version is not supported.
NOTSUPPORTED	Custom	An element or attribute is not supported.
NOTSUCCEDED	Custom	The operation could not be carried out for some reason.
NOTPERMITTED	Custom	The operation is not permitted.

Each interface defines a common set of custom error codes. In addition, each implementation may require additional error codes. The principle is that an error code should represent one type of error specific enough to locate the error.

Logging

In order to monitor the communication between the peers, it is recommended that both peers as minimum implement the following logging:

- Application software version, build version.
- Version of the schema definition file used to parse the XML-stream.

Errors that occur with time and reason. Faults which do not cause errors should be logged with a less severity.

Global Identifiers

NOPTIS 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 versions of the object. GID is used extensively in the interfaces as keys for referring to objects. It is assumed that a client application can map its internal data model to the proper GID when exchanging data.

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.

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.

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. 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 database instances represents the same object or not.

Object Types with GID

The following object types (classes) have global identifiers:

Class Name	Class Id
Block	9041

Class Name	Class Id
Line	9011

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

Class Name	Class Id
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 numbe*r 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. 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.

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)

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 (odd) or 2 (even).

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. 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.

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.

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 below.

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 Authority Number			Contractor Number				Vehicle 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), 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.

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 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 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.

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.