SIRI - Realisation guide for public transport in Switzerland

Based on the official CEN SIRI schema 2.0q

Authors	SKI
Status	In progress
Version	0.9
Last change	05.04.2023
Copyright	The document is free. Proliferation in unchanged form is explicitly supported.

DISCLAIMER

This document is only for reference purposes to the SIRI request/response services on Openmobilitydata.swiss. For all other purposes please contact qs.ski@sbb.ch for further information

Table of Contents

1 Preliminary remarks 14
1.1 Supported versions 14
1.2 Document structure and scope 15
1.2.1 Scope 15
1.2.2 Description of usage of the document 15
1.2.3 Description of the Structure sections 15
1.3 Binding nature 16
1.4 Referenced documents 16

2 Introduction 18
2.1 General task definition 18

3 Basic Interface description 19
3.1 Overview 19
3.2 Authentication 19
3.3 Compression 21
3.4 Messaging pattern 21
3.5 Participant identifiers, message & subscription management 22
3.6 Request / Response mechanism 23
3.7 Publish / Subscribe mechanism 23
3.8 Recovery & Restart 24
3.9 Service detection / Query URL 25
3.10 Recognition of service / URL for request 25
3.11 Optional fields and default values 27

4 Basic concepts of the data model 28
4.1 SIRI & Transmodel 28
4.2 SIRI requirements 28
4.3 Yearly timetable basis 29
4.4 Operator 29

4.5 Line 29
4.6 Direction 30
4.6.1 30
4.6.2 30
4.7 Stop points and stop assignments 30
4.8 Vehicle journey 31
4.9 Vehicle modes and product categories 31
4.10 Vehicle features 32
4.11 Topographic format 34
4.12 Time format and trains after midnight 35
4.13 Operating day and data frame 36
4.14 Language of text elements 36
5 Common message types and structures 37
5.1 SubscriptionRequest 37
5.1.1 Business 37
5.1.2 Structure 37
5.1.3 38
5.2 SubscriptionResponse 38
5.2.1 Business 38
5.2.2 Structure 38
5.2.3 39
5.3 ResponseStatus 39
5.3.1 Business 39
5.3.2 Structure 39
5.3.3 40
5.4 DataSupplyRequest 40
5.4.1 Business 40
5.4.2 41
5.4.3 41
5.5 ServiceDelivery 41
5.5.1 Business 41

5.5.2 Structure 42
5.5.3 42
5.6 DataReceivedAcknowledgement 42
5.6.1 Business 43
5.6.2 Structure 43
5.6.3 Example 44
5.7 CheckStatusRequest 44
5.7.1 Business 44
5.7.2 Structure 44
5.7.3 Example 45
5.8 CheckStatusResponse 45
5.8.1 Business 45
5.8.2 Structure 46
5.8.3 Example 46
5.8.4 47
5.9 TerminateSubscriptionRequest 47
5.9.1 Business 47
5.9.2 Structure 47
5.9.3 Example 48
5.10 TerminateSubscriptionResponse 49
5.10.1 Business 49
5.10.2 Structure 49
5.10.3 Example 49
5.10.4 51
5.11 ErrorConditions for Requests and Deliveries 51
5.11.1 Business 53
5.11.2 54
5.11.3 54
5.12.1 54
5.12.1 54
5.12.2 54

6 ProductionTimetable 55
6.1 Business 55
6.4 ProductionTimetableSubscriptionRequest 60
6.4.1 Business 60
6.4.2 Structure 60
6.4.3 Example 61
6.4.4 62
6.5 ProductionTimetableRequest 62
6.5.1 Business 62
6.5.2 Structure 62
6.5.3 64
6.6 ProductionTimetableSubscriptionResponse 64
6.6.1 Business 64
6.6.2 Structure 64
6.6.3 Example 64
6.6.4 65
6.7 ProductionTimetableDelivery 65
6.7.1 Business 65
6.7.2 Structure 65
6.7.3 66
6.8 DatedTimetableVersionFrame 66
6.8.1 Business 66
6.8.2 Structure 66
6.8.3 68
6.9 DatedVehicleJourney 68
6.9.1 Business 68
6.9.2 Structure 68
6.9.3 70
6.10 DatedCall 70
6.10.1 Business 71
6.10.2 Structure 71
6.10.3 74

7 EstimatedTimetable 75
7.1 Business 75
7.2.1 75
7.3.1 75
7.4 EstimatedTimetableSubscriptionRequest 75
7.4.1 Business 75
7.4.2 Structure 76
7.4.3 Example 77
7.4.4 78
7.5 EstimatedTimetableRequest 78
7.5.1 Business 78
7.5.2 Structure 78
7.5.3 80
7.6 EstimatedTimetableSubscriptionResponse 80
7.6.1 Business 80
7.6.2 Structure 80
7.6.3 Example 80
7.6.4 81
7.7 EstimatedTimetableDelivery 81
7.7.1 Business 81
7.7.2 Structure 81
7.7.3 82
7.8 EstimatedJourneyVersionFrame 82
7.8.1 Business 82
7.8.2 Structure 82
7.8.3 83
7.9 EstimatedVehicleJourney 83
7.9.1 Business 83
7.9.2 Structure 83
7.9.3 88
7.9.4 88
7.10 RecordedCall 88

7.10.1 Business ... 88
7.10.2 Structure ... 88
7.10.3 ... 93
7.10.4 ... 93
7.11 EstimatedCall ... 93
7.11.1 Business ... 93
7.11.2 Structure ... 93
7.11.3 ... 99
7.11.4 ... 99
7.11.5 ... 99
7.13 Data freshness ... 99
7.14 Disregarding the PreviewInterval ... 100
7.15 Completeness of incremental updates ... 100
7.16 Incremental updates and IsCompleteStopSequence ... 101
7.16.1 IsCompleteStopSequence and active state of a journey ... 102
7.16.2 IsCompleteStopSequence and full history of a journey ... 103
7.17 Handling of Predictions ... 104
7.17.1 Quality of Prognoses and Prediction Windows ... 105
7.17.2 Reporting Rules and Monitored Journeys ... 105
7.17.3 Temporal Reporting Behaviour - Sensitivity Threshold ... 107
7.17.4 PredictionInaccurate - InCongestion ... 108
7.17.5 Unexpected Termination of Monitoring ... 109
7.18 Prediction Quality ... 110
7.18.1 ... 110
7.18.2 ... 110
7.18.3 ... 110
7.18.4 ... 110
7.19 Handling of recorded / actual data ... 110
7.19.1 Functional definition of arrival and departure ... 111
7.19.2 Triggering events ... 111
7.19.3 RecordedCall with missing actual times ... 113
7.19.4 Immutability of actual times ... 113

7.19.5 RecordedCalls in initial loads 114
7.19.6 Expected times in RecordedCall 114
7.19.7 Special case recording of cancelled calls 114
7.19.8 Special case exceptional passthrough 114
7.19.9 Special case optional stop 115
7.19.10 Special case vehicle at stop 115
7.19.11 Illustration of the default case 119
7.19.12 Event not triggered 119
7.19.13 Loss of connection 121
7.19.14 Events not triggered and prediction unknown 122
7.19.15 Events not triggered after a certain point 123
7.19.16 Extra journey without real-time feed 124
8.1.1 126
8.1.2 126
8.1.3 126
8.1.4 126
8.2.1 126
8.2.2 126
8.2.3 126
8.2.4 126
8.2.5 126
8.3.1 126
8.3.2 126
8.3.3 126
8.4.1 126
8.4.2 126
8.4.3 126
8.5.1 126
8.5.2 126
8.5.3 126
8.6.1 126
8.6.2 126

8.6.3 ... 126
8.7.1 ... 126
8.7.2 ... 126
8.7.3 ... 126
8.8.1 ... 126
8.8.2 ... 126
8.8.3 ... 126
8.8.4 ... 126

10 Case Collection ... 128
10.1 Use of Order and VisitNumber in this chapter ... 128
10.2 Establishing the baseline with an initial load ... 129
10.2.1 Business ... 129
10.2.2 Example ... 129
10.3 Delay ... 134
10.3.1 Business ... 134
10.3.2 Example ... 135
10.4 Platform update ... 141
10.4.1 Business ... 141
10.4.2 Example ... 142
10.5 Extra journey ... 144
10.5.1 Business ... 144
10.5.2 Example ... 145
10.6 Journey cancellation ... 146
10.6.1 Business ... 146
10.6.2 Example ... 147
10.7 Partial cancellation ... 149
10.7.1 Business ... 149
10.7.2 Example ... 150
10.8 Extra stop ... 151
10.8.1 Business ... 151
10.8.2 Example ... 152

10.9 Exceptional passthru 154
10.9.1 Business 154
10.9.2 Example 154
10.10 Rerouting 159
10.10.1 Business 159
10.10.2 Example 160
10.11 Recording of calls 167
10.11.1 Business 167
10.11.2 Example 167
10.12 Vehicle waiting at stop 167
10.12.1 Business 167
10.12.2 Example 167
10.13 Boarding activity update 168
10.13.1 Business 168
10.13.2 Example 168
10.14 No alighting update 170
10.14.1 Business 170
10.14.2 Example 170
10.15 No boarding update 174
10.15.1 Business 174
10.15.2 Example 174
10.16 Replacement transport and journey relations 177
10.16.1 Business 177
10.16.2 Example 178
10.17 Train formation update 180
10.17.1 Business 181
10.17.2 Example 181
10.18 Occupancy update 193
10.18.1 Business 193
10.18.2 Example 194
10.19 Line timetable over production timetable 198
10.19.1 Business 198

10.19.2 Example 198
10.20 Situation affecting line (TODO) 199
10.20.1 Business 199
10.20.2 Example 199
10.21 Situation affecting stop place (TODO) 199
10.21.1 Business 199
10.21.2 Example 199
10.22 Situation affecting elevator and impacting accessibility (TODO) 199
10.22.1 Business 199
10.22.2 Example 199
10.23.1 199

Table of Figures

Figure 2: Journey ordering in a daily timetable. 56
Figure 3: Preview Time and Active First Message. 107
Figure 4: Reporting Behaviour in the case of PredictionInaccurate. 109
Figure 5: Loss of Contact. 110
Figure 6 - Illustration of the baseline example with the most important information from the perspective of an arbitrary passenger information system, e.g., stop sequence, timings, platforms and whether a stop must be requested explicitly (depicted by the red stop button). 130
Figure 7 - Illustration of the delay example with timings representing the predictions 135
Figure 8 - Illustration of the delay example with an update rule applied to the expected times 138
Figure 9 - Illustration of the example: platform change at the fourth stop 142
Figure 10 - Illustration of the partial cancellation example: different interpretations are possible depending on whether the vehicle is railbound (exceptional passthrough) or unbound (bypassing / rerouting) 149
Figure 11 - Illustration of the extra call example: different interpretations are possible depending on whether the vehicle is railbound (exceptional stop) or unbound (bypassing / rerouting) 152
Figure 12 - Illustration of the rerouting example: original route is partially cancelled and replaced 160
Figure 13 - Illustration of the journey relation example: due to a cancellation, passengers must transfer from the train to different replacement buses at call “30” and call “B”. 178
Figure 14 - Illustration of the formation example: train travels in shortened formation from call 30 onwards 182
Figure 15 - Illustration of the train formation at the first call («..StopPoint:998375543» / Order «10») with the vehicle features and the coaches mapped to sections on platform 2 («..:Quay:76007252:11») 183
Figure 16 - Illustration of the occupancy examples: coach 2 cannot accept wheelchair passengers anymore, the 2nd class sections of coach 3 are packed and coach 4 has limited capability for bicycles 194

Table of Tables

Table 1: Referenced documents 17
Table 2: Server recognition parts. 26
Table 3: ServiceFeature and VehicleFeature (SIRI-1, 5.14). 34
Table 4: Delivery Content Elements: 41
Table 5: System and Application Error Conditions 51
Table 6: Application Error Conditions Related to Request Parameters 53
Table 9: ArrivalBoardingActivityEnumeration. 73
Table 10: DepartureBoardingActivityEnumeration. 74
Table 12: PredictionInaccurateReasonEnumeration 94
Table 13: Occupancy mapping proposal VDV - SIRI 95
Table 14: CallStatus - allowed values and usage 97
Table 15: ArrivalBoardingActivityEnumeration 98
Table 16: DepartureBoardingActivityEnum 99
Table 19: Overview of the use cases relevant in public transport CH 128

1 Preliminary remarks

This document describes the realisation specifications for public transport in Switzerland, based on the official SIRI 2.0 standard by usage of XSD 2.0o (German abbreviation: SIRI-RV = SIRI-RG SIRI realisation guidelines).

It provides detailed clarifications and describes deviations from the official SIRI standard specification documents with the aim of achieving consistent use throughout public transport in Switzerland. These guidelines should be followed by all producers of ITCS and they should be demanded as qualification criteria within call for tenders. It will be treated as SIRI profile but does not yet represent the official Swiss national profile (as coordination with other public transport operators is pending).

The realisation specifications in this document will be agreed for all multi-modal customer information in public transport in Switzerland. and they are the result of the agreement process regarding consistent handling of CEN standards NeTEx, SIRI and Transmodel within public transport in Switzerland.

The realisation specifications mainly concern:

- Detailed clarifications about items which have intentionally abstract and open definitions in the standard.
- Detailed clarifications about items which have been handled inconsistently within public transport in Switzerland so far.
- Abandonment of elements of the standard.
- Intentional deviations from the official standard within public transport in Switzerland.

The guidelines should be able to be used as unique profile in order to allow simple usage for the implementation of consumer systems of the client relevant information.

Only data which follow the specifications will be accepted by the system leader. Each deviation must be corrected by the data provider or - if accepted by the system leader - they will be developed and operated for money.

1.1 Supported versions

This realisation guide is based on **SIRI XSD 2.0o**. Whenever version differences or clarifications are present in the Structure sections, it will be emphazised in the respective “business requirement” / “remark” rows below the elements.
The corresponding XSD schemas can be found at: https://github.com/SIRI-CEN/SIRI/releases/tag/v2.0o

1.2 Document structure and scope

1.2.1 Scope

The present SIRI realization guidelines for public transport Switzerland are derived from the official SIRI CEN standards. This document replaces the official SIRI documentation within public transport Switzerland and therefore includes the complete information which is necessary for the implementation or the comprehension of the SIRI interface. Furthermore, it includes references and hints to VDV453 and VDV454 documents which are widely used in Switzerland and by the national data hub SKI CUS.

1.2.2 Description of usage of the document

This document shows all elements of the norm that are deemed necessary for the data exchange. We use four sections:

- Business: Telling the business story of the element and how it is used in public transportation in Switzerland
- Structure: Contains the detailed physical structure of the element with examples and more information about the business level
- Example: A detailed example
- Hints: Hints for the implementation.

The tables in the “structure” section have the same form as the original description in the standard (or the XSD for that matter), but are also adapted to the needs of the public transport in Switzerland. In some cases, the cardinality may change and fields may become mandatory or optional.

Additionally, we use the following information:

- LATER: This element or field is ignored during import, but it may be included in a future version.

Value transformations and value lists are to be used. Additional values are not allowed.

In some cases, there are references to the HRDF format currently used in the data exchange of time tables in Switzerland. This is also to help implementers to understand how to work with it or if a HRDF timetable is used as the baseline timetable.

Within the tables which describe the XML structure of a data element it is mentioned whether the given element is a mandatory or an optional one. If the usage of the element differs from the original XSD the usage value is represented in red characters within this document.

1.2.3 Description of the Structure sections

- For the purposes of this document, the terms and definitions given in EN 15531-1:2015 apply.

- For the purposes of this document, the symbols and abbreviations given in EN 15531-1:2015 apply.

1.3 Binding nature

This document describes the way in which the SIRI standard is specifically applied and interpreted in Switzerland. It forms the basis for agreements concerning the connection between the individual public transport partners for exchanging real time data.

The precise volume of data is recorded in a contract.

1.4 Referenced documents

Reference	Referenced document full name and version
SIRI-1	Public transport - Service interface for real-time information relating to public transport operations - Part 1: Context and framework; English version DIN EN 15531-1:2015
SIRI-2	Public transport - Service interface for real-time information relating to public transport operations – Part 2: Communications; English version DIN EN 15531-2:2015
SIRI-3	Public transport - Service interface for real-time information relating to public transport operations – Part 3: Functional service interfaces; English version DIN EN 15531-3:2015
SIRI-Conversion	Mapping file for values between SIRI and VDV and NeTEx (Excel)
SIRI-HB	SIRI handbook v1.4
NeTEx-1	DIN CEN/TS 16614-1 (2014). Public transport - Network and Timetable Exchange (NeTEx) – Part 1: Public transport network topology exange format; English version EN 16614-1:2014
NeTEx-2	DIN CEN TS16614-2 (2014). Public transport - Network and Timetable Exchange (NeTEx) - Part 2: Public transport scheduled timetables exchange format; English version CEN/TS 16614-2:2014
NeTEx-3	DIN CEN TS16614-3 (2016). Public transport - Network and Timetable Exchange (NeTEx) –

	Part 3: Public transport fares exchange format; English version CEN/TS 16614-3:2016,
NeTEx-RG	NeTEx_realisation_guide_tp_suisse
NeTEx-Id	NeTEx id reference (Excel)
NeTEx-Mapping	Mapping document for values between NeTEx and HRDF (Excel)
VDV-453	Verband Deutscher Verkehrsunternehmen VDV VDV-Schrift 453 - Ist-Daten-Schnittstelle Fahrplanauskunft Version 2.6, Köln (D), 2017
VDV-454	Verband Deutscher Verkehrsunternehmen VDV VDV-Schrift 454 - Ist-Daten-Schnittstelle Fahrplanauskunft Version 2.2, Köln (D), 2017
TM-SNCF	Modèle IV (SNCF, PPT)
VDV-462	VDV 462 NeTEx Schrift (1/2017): Standardisierter Austausch von Liniennetz und Fahrplandaten mit der europäischen Norm CEN-TS 16441 'NeTEx'
VDV-462_M	VDV 462 NeTEx-Mappingtabelle (Excel)
HRDF	HAFAS-Rohdatenformat-5.20.39_d
HRDF-CH	INFO+ HRDF-Exportschnittstelle 5.20.39. Spezifikation der nach HRDF 5.20.39 exportierten Datei- und Zeilentypen in INFO+ (Version 1.2, 29.9.2017)
CERTU	Application de la norme SIRI (v2.0) en Île-de-France « Local Agreement SIRI » Profil SIRI pour l'Île-de-France Version 2.4
TfNSW	SIRI 2.0-TfNSW Implementation Specification Using SIRI at TfNSW Version 1.0.1

Table 1: Referenced documents

2 Introduction

2.1 General task definition

The task performed by the SIRI interface is the transmission of client relevant information to one or more partners. The data transmitted via this interface is also required for the provision of timetable data in information systems.

This document sets out the Swiss-wide standard for the implementation of the SIRI interface and of individual data structures with regard to the mutual exchange of client relevant information for different modes of transport between public transport companies by use of ITCS (Intermodal Transport Control System) and so-called data hubs (German abbreviation: DDS - Datendrehscheiben).

The document specifically describes:

- Which data may be exchanged between public transport partners
- Which SIRI elements are supported within public transport in Switzerland
- Explicit deviations from the corresponding SIRI standard and EU profiles
- The format of individual data elements
- The data flows in terms of content and time
- Which agreements are necessary with respect to metadata
- Which needs have to be taken into account when operating the interface
- How data is to be interpreted
- Which business use cases have to be supported and how

3 Basic Interface description

3.1 Overview

Messaging Pattern:

- HTTPS with TLS 1.2 or 1.3
- POST messages with Content-Type XML (SIRI payload)
- Publish/Subscribe interaction
- A subscription is typically established early in the morning and is valid one operating day
- The only supported subscription filter is OperatorRef
- Direct Delivery (without SOAP Headers or wsdl schema respectively)
- The publisher pushes the corresponding SIRI service data as soon as updates are available (without delays or DataReady/DataSupply overhead)
- Simple (Multipart) Despatch, i.e., sending a DataReceivedAcknowledgement after the payload (or last data package) was received is optional
- The consumer sends CheckStatusRequests typically every minute (and Hearbeat is not supported)
- In case of an error or for recovery purposes (new baseline with initial load), a subscription is terminated and re-established (consumer responsibility)

Authentification: OAuth 2.0, Client Credentials Grant Type

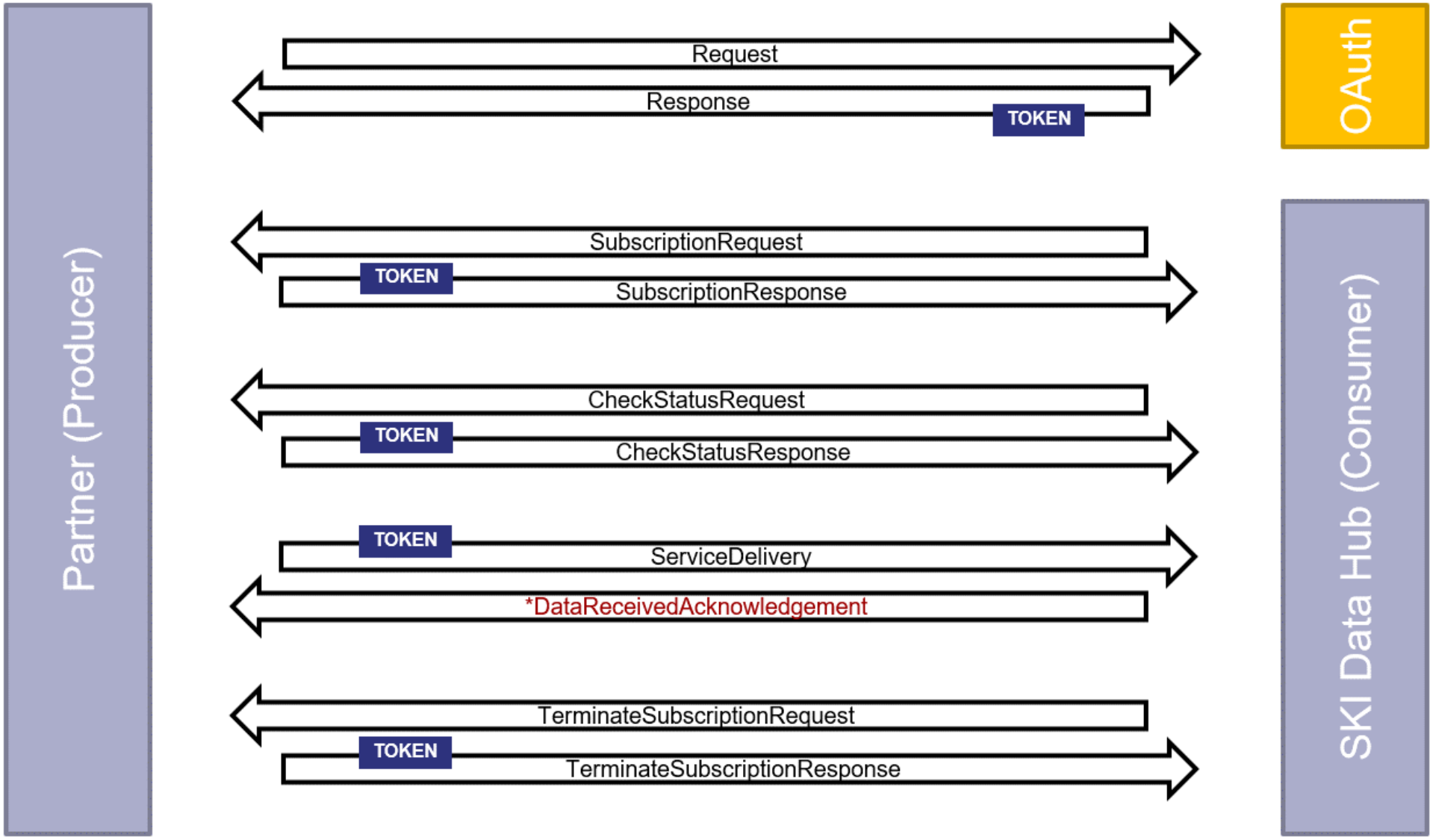
XSD: siri.xsd, Version 2.0

Encoding: UTF-8

Date/Time Format: ZULU Time

3.2 Authentication

A partner does not necessarily have to build an OAuth infrastructure. Since HTTPS is enforced, a partner may for example simply use an IP filter or assign SKI (in the role of the client) a stable bearer token instead. In other words, a partner system must always authentificate itself via OAuth when communicating with the SKI data hub, but in the opposite direction the partner is free to propose an appropriate mechanism. The following figure shows roughly illustrates the message flow.



* Optional for Partner, but always sent by SKI data hub (see SIRI-2, Kapitel 8.1.2 ff)

Three environments are available with the following URLs:

Environment	Service URL *	Token URL **
TEST	https://api.test.ddip.odpch.ch:443	https://token.test.ddip.odpch.ch:443/oauth2/token?grant_type=client_credentials
INT	https://api.int.ddip.odpch.ch:443	http://token.int.ddip.odpch.ch:443/oauth2/token?grant_type=client_credentials
PROD	https://api.prod.ddip.odpch.ch:443	https://token.prod.ddip.odpch.ch:443/oauth2/token?grant_type=client_credentials

* The full service or request URL will depend on the SIRI version, service type and the participant code (also called sender ID) as well as some other parameters. Example: https://api.test.ddip.odpch.ch/siri20/ps-xml/et/tac-out-et_test/ski-ddip-in-et_test

** Validity of the JWT (JSON Web Tokens) is typically 60 minutes.

3.3 Compression

To limit the size of the messages and therefore safe bandwith, gzip HTTP compression is recommended.

3.4 Messaging pattern

(SIRI-1, 1.1.2; SIRI-1, 3.2.15; SIRI-2, 5.1.3; TfNSW, p. 41)
The “Data Exchange Patterns of Interaction” define the interaction process that must be undertaken by the Consumer and the Provider to specify the desired payload and to deliver the requested payload. SIRI offers two main patterns of interaction for data exchange: Request/Response and Publish/Subscribe. The patterns are complementary.

According to the SIRI specification, implementations may support both or only one pattern. Using the Request/Response pattern, the Consumer is able to request a single data delivery which contains all the currently available data.

Using the Publish/Subscribe pattern, the Consumer is able to tell the Provider that they are not only interested in a single data delivery but also wish to be notified whenever the data changes in the future. The main interaction steps are:

- Subscription setup: The Consumer sends a message to the Provider to create a subscription. The message may include parameters that indicate his specific interests and a predefined expiry time

- Notification/Delivery: The Provider notifies the Consumer that new data is available (“Fetched Delivery”) or he sends the data directly (“Direct Delivery”).
- Subscription termination: The Consumer sends a message to the Provider toterminate the subscription before their predefined expiry time

The SIRI publish/subscribe pattern of interaction follows the paradigm described in the W3C candidate standard ‘Publish-Subscribe Notification for Web Services (WS-PubSub)’. SIRI uses the same separation of concerns, and a similar terminology for Publish/Subscribe concepts as is used in WS-PubSub.

Available Options:

a. Direct Delivery pattern: The payload data is sent directly to the Consumer as soon as the Provider has it available.
b. Fetched Delivery pattern: The fetched delivery is a two-step process: The Provider does not send the payload data directly to the Consumer but sends a notification message instead. The Consumer fetches the payload later using a dedicated DataSupply request.

In Switzerland we only use the Direct Delivery (SIRI-2, 5.2.2) pattern without SOAP headers.

In Direct Delivery, the burden of holding and queuing messages is distributed to the client, with some advantages for scaling, as the central server needs neither retain data nor allocate computation resource to service the additional data supply steps. The interaction is simpler, with fewer messages being exchanged, and a simpler mediation. Direct Delivery is appropriate for deployment with fast, reliable communications and with adequate processing capability on the Consumer. It is especially efficient when most of the updates are relevant to the client, and are used immediately.

For ensuring that no data is lost, the server can check the receipt of every delivery. This procedure is called Acknowledged Despatch. In reliable communication with low expectation of data loss or where data loss can be tolerated, the server can alternatelively despatch data and ignore whether the client sends an acknowledgment proving data receipt. In this case, communication relies on Simple Despatch.

For maximum efficiency Simple (Multipart) Despatch (SIRI-2, 5.5) will be used.

3.5 Participant identifiers, message & subscription management

The requests and responses of SIRI functional services use a common set of header elements to track messages and subscriptions:

- Each request message can be given an identifier by the requestor (MessageIdentifier) that is echoed back in the response (RequestMessageRef). This allows a requestor to match request and responses.
- All messages are timestamped with an identifier in UTC (see also section 4.12 Time format).**Error! Reference source not found.** Time format).

3.6 Request / Response mechanism

The common elements involved in SIRI request/response interactions are as follows:

- SIRI functional service requests are sent from a ConsumerService to a ProducerService, which returns a delivery satisfying the request Topics and Policies.
- A ServiceRequest contains one or more SIRI functional requests for specific functional services. One service (e.g. ET) is included per service request. Each SIRI functional service request inherits common service request properties.
- A ServiceDelivery contains one or more delivery elements for a specific SIRI functional service - or ErrorConditions if responses cannot be returned. Each SIRI functional service delivery inherits common service delivery properties (Note: Subscription attributes are not populated in deliveries for request/response).
- The RequestMessageRef in the ServiceDelivery messages references the MessageIdentifier of a ServiceRequest to which it responds. The RequestMessageRef on each individual functional service delivery, within the ServiceDelivery, references a MessageIdentifier on the corresponding functional service request.

3.7 Publish / Subscribe mechanism

(SIRI-HB, p.26)

The common elements involved in SIRI publish/subscribe interactions are as follows:

- The SubscribingService requests subscriptions from a SubscriptionManager associated with the ProducerService which creates and manages subscriptions.
- Each subscription is given a unique identifier (SubscriptionCode) by the subscriber, which is specified on the subscription request for a specific SIRI functional service (SIRIXxxFs-SubscriptionRequest) as the SubscriptionIdentifier, and then included in all deliveries sent back in response to that subscription. The identifier must be unique within the ParticipantCode of the subscriber (SubscriberRef).
- One or more functional service subscription requests may be submitted at a time as part of a single SubscriptionRequest. All must be for the same type of service. In response to the subscription request, a SubscriptionResponse message is returned with a ResponseStatus containing the identifier of each functional service subscription (SubscriptionRef) and whether it has been successfully created.
- A subscription request includes an InitialTerminationTime which specifies a desired lease period. This will be echoed back as the ValidUntil time on responses.
- The ProducerService subsequently monitors the real-time feed and if the request criteria of the subscription are matched, creates deliveries that are sent by the ProducerService to the ConsumerService nominated on the subscription request. Each delivery message includes the subscription identifier (SubscriptionRef) and the ParticipantCode of the subscriber, termed the SubscriberRef.
- The subscriber, consumer and producer service keep mirrored representations of the active subscriptions. Recovery from system failure or loss of connection is discussed separately below.

- The SubscribingService requests needs to be configured to know the address i.e. URL of the system to which new subscription requests are to be sent. The request can contain a ConsumerEndPointAddress to which delivery responses are to be sent and a distinct SubscriberEndPointAddress to which an acknowledgement for the subscription request is to be sent.
- An optional capability of SIRI is to have a SubscriptionFilter that will group subscriptions from the same participant for the same type of service so that notifications and delivery data can be sent as a single message. Otherwise each subscription will give rise to separate notification and delivery messages.
- All messages are timestamped with an identifier in UTC (see also section 4.12).

In the traditional SIRI publish/subscribe setting (see section 5.1) each subscription is given a ***SubscriptionIdentifier*** by the subscribing system. This identifier is used later in the communication in order to refer to the subscription, e.g. when the source delivers data or when the subscriber terminates a subscription.
According to CEN/TS 15531-2, chapter 7.1.1 the subscription identifier needs to be unique across all current subscriptions of a consumer within the same SIRI service.
However, a ***TerminateSubscriptionRequest*** cannot distinguish between SIRI service type and on top of that allows for the "All" option. Now imagine a setting where a source delivers data for both SIRI-ET and SIRI-PT to the same consumer. If the subscription endpoint at the source is the same for all services (e.g. http://SIRI.data.source:1234/endpoint) and the consumer uses the same ***RequestorRef*** for both services (e.g. “Consumer1”), then the source will not know which subscriptions to terminate if a Terminate-All-Subscriptions request is received from "Consumer1" - only SIRI-ET or SIRI-PT subscriptions or all subscriptions across both service types? A similar problem arises if “Consumer1” wants to terminate an individual subscription but the ***SubscriptionIdentifier*** is only unique within SIRI-ET or SIRI-PT (not across the services) – does the consumer want to terminate the SIRI-ET or SIRI-PT subscription?
Hence in in Switzerland, if a source delivers data through different SIRI services to the same consumer, it is required that either:

a) the data producer provides different endpoints for different SIRI services (e.g. http://SIRI.data.source:1234/ET_endpoint and http://SIRI.data.source:1234/PT_endpoint) or
b) the data consumer uses different RequestorRefs for different SIRI services (e.g. “Consumer1_ET” and “Consumer1_PT”)

in order to avoid ambiguities in the subscription handling.

3.8 Recovery & Restart

(SIRI-HB, p.28)
For Simple Despatch, data is sent without an acknowledgement of receipt. For a Request/Response data supply interaction, the Requestor knows that it is waiting for a response and can determine whether it has been satisfied. However for asynchronous, i.e. Publish/Subscribe interaction, the Notification

Producer can send a data message to the Consumer at any arbitrary point in time; so, if for some reason the message fails to arrive, the transmission failure will not be detected by either party, and no recovery action will be attempted. In many cases this is sufficient, since another update shortly after will update the messages as effectively as a recovery action.

In other cases, mechanisms are needed to allow for recovery from system failure of either producer, consumer or communications link. These are described in Part 2 of the SIRI specification and partly in chapters 5.7/5.8. In brief: a CheckStatus request can be sent between any two participants to see if the other is still working. The response (typically from producer to consumer) includes a ***ServiceStartedTime*** which can be used to detect if the producer has failed and was restarted. It is up to the subscribing service (consumer) to detect interruptions of service and hence terminate and re-establish subscriptions if necessary.

i. To reduce traffic/load and to clarify roles in CheckStatusRequest/Response, **only the data consumer sends CheckStatusRequests.**

ii. **A direct heartbeat mechanism is not implemented.**

iii. The consumer is responsible to signal to the producer in case a message needs to be delivered again (i.e. data loss occurred). **In case of a problem** (e.g. signaled by the producer with a new ServiceStartedTime), **the consumer must initiate the subscription termination and resend a SubscriptionRequest**, i.e., trigger a reset of the subscription which in turn results in the delivery of a new initial load or fresh baseline by the producer.

3.9 Service detection / Query URL

(VDV-454, 4.4)
Since changes within the system environment of a partner, who is acting as a server, may also have an impact on the addressing of the application, it is recommended that the addressing of queries should be designed to be configurable on the client side.

Changes to the URL of a service on the server side **must** always be agreed with the clients.

3.10 Recognition of service / URL for request

Changes in the system environment of the partner, who acts as a server to the data exchange, may have implications for the URL. Therefore, it is recommended that the addressing for requests is always configurable on the client side.

Service-URL changes must be coordinated with all subscribers beforehand.

The information about the server must be part of the URL and also set in the XML as an element. The following attributes can be used to identify a system. Both the sender of the message (**system**) and also the **environment** from which the message is sent are required in the server recognition. Both parts are concatenated by a "_". Further optional parts can be used as shown in Table 2:Table 2:

- <**partner system name**>[-<**application**>-<**direction**>-<**service**><**numbering**>]_<**environment**>
- It is recommended that all parts are lower case
- It corresponds to the XML elements ***RequestorRef*** and ***ProducerRef*** in the SIRI payload.

The server recognition consists of the following parts in Switzerland:

Identifier Part	**Description**	**Examples**
Partner	The partner system name may be chosen freely. The separator "_" is not an allowed part within the system name. We suggest that the partner name consists of the abbreviation of the partner information and if necessary additions to identify the system. E.g.: „sbb", „sbbfpl", „aags", „riv", „zvv", „zvb", „svb-lio", „svb-dss"	tpf
Application	Name of partner system	-trip
Direction	Delivery direction from the point of view of sending system can be sent (out = sending delivery, in = receiving delivery)	-in -out
Service	SIRI service	-et -pt -sx
Numbering	If multiple connections are required with the same system	1
Platform	Platform identifier. Other platform identifiers can be used only after mutual agreement. It is not necessary for a partner to have all these environments. However, a mapping between the two involved environment sets is necessary.	test, int, prod

Table 2: Server recognition parts.

Examples.: zvv_test, zvv_prod, riv_prod, sbb_int, sbb_prod, svb-dds_test, svb-dds_prod, tpf-trip-in-et1_test

3.11 Optional fields and default values

SRIIn the context of SIRI-ET, a data producer must establish a proper initial state or so called “baseline” for each and every journey after a new subscription is established with the consumer regularly or after an error (for example connection loss) has occurred. A baseline of a journey is synonymous to an ***EstimatedVehicleJourney*** message with ***IsCompleteStopSequence*** equal to ‘true’. A complete stop sequence, in turn, means that not only each and every called stop of the journey pattern (valid at time of processing) must be included but also each and every mandatory element / structure as well as all optional elements / structures where the value / content is different from the default value as specified in the XSD schema.

In subsequent incremental updates optional values can be omitted if not specified otherwise in the following sections. **Careful:** multiplicity 1:1 elements must always be delivered, even in incremental updates.

4 Basic concepts of the data model

4.1 SIRI & Transmodel

(SIRI-HB, p.20)
The payload content of SIRI messages is intended to be compatible with Transmodel, the CEN reference data model for public transport, which sets out common terminology and data abstractions for representing public transport information systems. Thus, it should be possible to map data readily from a Transmodel compliant database or application into SIRI messages for data exchange and vice versa, and someone familiar with Transmodel should be able to recognize the application payload elements of the SIRI content model:

- SIRI uses Transmodel terms and data elements wherever they exist, and assumes the same cardinalities as Transmodel for any associations between elements.
- In some cases, SIRI introduces additional elements or attributes for additional concepts not yet in Transmodel. Mostly this consists of adding additional attributes to existing Transmodel concepts.
- In certain other cases, additional elements are introduced in SIRI for convenience to group existing elements, for example, a SIRI call element groups the Transmodel entities STOP IN SEQUENCE, various PASSING TIMEs, and other additional attributes. These additional elements do not break conformity with Transmodel but can be regarded as “implementation views."

4.2 SIRI requirements

(SIRI-HB, p.28)
The deployment of a concrete SIRI implementation entails the following:

- The choice of one or more appropriate SIRI Functional Services to meet the application needs.
- A country and/or local profile that will set out **delivery method,** the **capabilities** to be supported, the **data reference system** for stop identifiers etc.
- Allocation of unique **Participant** Identifiers to identify the different systems.
- A **Producer application** for the functional service deployed to a server at a known endpoint URL.
- A **Consumer application** for the functional service deployed to a client.
- A **Status service** to check that the service is available and working correctly and that there has not been an interruption.

4.3 Yearly timetable basis

In public transport Switzerland the yearly (or also called periodic) timetable is published in HRDF (see .../timetables-hrdf), GTFS (.../timetables-gtfs) as well as NeTEx format (see .../timetable-{yyyy}-netex). SIRI data therefore references entities or data elements used in HRDF and NeTEx. This chapter describes the most important entities as introduced in the timetable and referenced in the real-time udpates (by ID).

4.4 Operator

(NeTEx-1, 7.4.5 / NeTEx-RG, 5.5 Organisation)
In Switzerland organisations are identified by their so called “GO” number which is assigned and managed by the SKI system DiDok (see .../goch).

- ⇨ The SIRI XML element ***OperatorRef*** must reference the GO number corresponding to the concessionaire of a given line or journey.
- ⇨ The referenced GO number must match the value of the corresponding line or journey in the yearly timetable (SKI INFO+).
- ⇨ ***OperatorRef*** is a mandatory field in public transport Switzerland and must be provided in the following format:
  <*UIC country code, e.g. “ch”*>:1:Organisation:<*GO number of* concessionaire *according to the master data in DiDok*>
  e.g. “ch:1:Organisation:11”
- ⇨ Any line timetable (PT) or journey update (ET) without ***OperatorRef*** is discarded.

4.5 Line

(NeTEx-RG, 8.9 Line; VDV-453 SBB Spez., 6.1.6; SIRI-1, 5.9)
Transmodel defines a LINE as a grouping of ROUTEs that are known to the public by a similar name or number. These ROUTEs are usually very similar to each other from the topological point of view, being variants of a core route with some deviations on certain parts only. Often the vehicle journeys on these ROUTEs are scheduled jointly with tight synchronisation, in order to provide a regular service on this specific LINE. They are often grouped together for presentation of the timetable to the public. Two ROUTEs using the same infrastructure path (or parallel tracks), but with opposite DIRECTIONs, will generally belong to the same LINE. A LINE can also be defined as a group of one or more JOURNEY PATTERNs.

- ⇨ Every line as referenced in SIRI data must have a globally unique ID (within public transport CH) mapped to ***LineRef***, at most two directions as defined in section 4.6, exactly one ***VehicleMode*** and a designation (mapped to ***PublishedLineName***) with which passengers can uniquely (at least given the operator) identify the line in timetables, on stationary or vehicle displays etc.
- ⇨ The allowed format of ***LineRef*** is:

<*UIC country code, e.g. "ch"*>:1:Line:<*GO number of* concessionaire *according to the master data in DiDok*>:<*ID unique within the declared organization, e.g. a combination of the product category / submode according to section 4.9 and the official line number according to the yearly timetable*>
e.g. "ch:1:Line:11:IC1"

4.6 Direction

(SIRI-1, 5.9; NeTEx-1, 8.4.5.5.1 Direction, NeTEx-RG, 8.6 Direction)
Direction in NeTEx is defined as a classification for the general orientation of ROUTEs. In order to distinguish references to LINEs and the particular DIRECTIONs in which a LINE runs, separate unambiguous LINE reference, and DIRECTION reference elements are used in SIRI interfaces. These constitute a small model of related elements whose values are agreed bilaterally. In the normalised Transmodel schema, LINE and ROUTE are distinct entities. LINE and DIRECTION are properties of a ROUTE whereas a ROUTE is associated with a JOURNEY PATTERN (and each VEHICLE JOURNEY follows a JOURNEY PATTERN). A DIRECTION is a relative sense of traversal of a ROUTE (e.g., inbound/outbound) and in SIRI represented by a string ***DirectionRef*** which may have a (standardized) direction identifier associated with it.

Only values conforming with the yearly timetable or SKI INFO+ master data are allowed:

a. "ch:1:Direction:H" which corresponds to direction outbound
b. "ch:1:Direction:R" which corresponds to direction inbound

Careful: direction names/texts from the perspective of passengers, e.g., "Zürich HB", "Lausanne" etc., are transmitted in ***Origin***-/***DestinationDisplay***.

4.7 Stop points and stop assignments

(NeTEx-RG, 8.23)
The allocation of a SCHEDULED STOP POINT (i.e., a STOP POINT of a SERVICE PATTERN or JOURNEY PATTERN) to a specific STOP PLACE for a PASSENGER SERVICE and, also possibly, a QUAY and BOARDING POSITION.

In public transport Switzerland the stop place model is specified by SKI DiDok. Four levels are possible:

a. Stop place.
b. Stop area within a stop place, e.g. a grouping of multiple quays/platforms.
c. Quay or platform, i.e., a location within a stop place or possibly stop area where passengers can actually board or alight from a vehicle.

d. Sectors as a finer subdivision of quays/platforms to provide customers with more precise information about where (on a possibly 100m long platform) a certain part of the train or specific wagon (e.g. with a restaurant or wheelchair access) is stopping.

⇨ Any stop place or quay/platform referenced in the yearly timetable as well as real-time data exchanges must exist in the DiDok master data.

⇨ In SIRI data the XML element ***StopPointRef*** must either correspond to a stop place or quay/platform that can be uniquely identified in the DiDok master data.

⇨ The SIRI XML elements ***Arrival***-/***DeparturePlatformName*** must contain the publicly known designation of a quay/platform corresponding to BEZEICHNUNG in the DiDok master data, e.g. “5”, “A” etc. If quay/platform name is combined with one or multiple sectors, the allowed format is:

<quay/platform name><sector name at most 3 digits of which the second one may be a hyphen in case of 4 sectors>,

e.g. “13AB” or “2B-F”.

⇨ In the SIRI structure ***Arrival***-/***DepartureStopAssignment*** the quay references (****QuayRef***) must correspond to DiDok quays/platforms, i.e., match with the BEZEICHNUNG_BETRIEBLICH or the Swiss Location ID (SLOID) of the quay/platform.

4.8 Vehicle journey

(NeTEx-RG, 10.4; SIRI-3, 5.5.4.3; 6.5.5)
Each ***DatedVehicleJourney*** or ***EstimatedVehicleJourney*** must have a ***FramedVehicleJourneyRef*** containing a journey ID that is unique within any given operating day represented by ***DataFrameRef***. The latter ensures that a journey ID can be reused for the “same” journey (from the point of view of a passenger due to same line and timings) at another operating day. ***DataFrameRef*** is always transmitted in the date format yyyy-mm-dd.

4.9 Vehicle modes and product categories

(NeTEx-RG, 9.9.2; SIRI-Conversion)
In Switzerland mode information is subject to the national standard V580 part 06 “Harmonization of vehicle modes”. Two levels of mode information are relevant in the context of SIRI:

1. “Verkehrsmittelkategorie” which corresponds to VehicleMode in SIRI.

   The NeTEx/SIRI and VDV454 modes correspond as follows:

VehicleMode (NeTEx/SIRI)	**ProduktID** (VDV453/454)
rail	Zug

bus	Bus
tram	Tram
metro	Metro
water	Schiff
funicular	Standseilbahn
cableway	Kabinenbahn
ferry	Schiff
coach	Bus
trolleyBus	Bus
air	-
lift	Aufzug
snowAndIce	-
other	Unbekannt
rackRailService (SIRI 2.1)	Zahnradbahn

2. “Angebotskategorie” which corresponds to the sub modes in SIRI. However, since the standardized sub modes in CH are not an exact match with the modes provided by NeTEx/SIRI, ProductCategoryRef must be used instead.

   ProductCategoryRef must contain a reference corresponding to a sub mode from V580 part 06 “Harmonization of vehicle modes” otherwise the journey is discarded. The reference must be transmitted in the following format:

   <*UIC country code, e.g. “ch”*>:1:TypeOfProductCategory:”<*sub mode or officially called “Angebotskategorie” conforming with V580 part 06*>

   Example: “ch:1:TypeOfProductCategory:B” which corresponds to the default sub mode of vehicle mode ‘bus’ (see chapter 4.1 “Vehicle mode bus” of V580-06).

4.10 Vehicle features

Y: Feature is available
---: Feature is not available

Feature	Vehicle	Service
Unknown	Y	Y
firstClass	Y	Y
secondClass	Y	Y
thirdClass	Y	Y
economyClass	Y	Y
businessClass	Y	Y
sleeper	Y	Y
couchette	Y	Y
specialSeating	Y	Y
freeSeating	Y	Y
recliningSeats	Y	Y
babyCompartment	Y	Y
familyCarriage	Y	Y
suitableForWheelChairs	Y	Y
lowFloor	Y	Y
stepFreeAccess	Y	Y
boardingAssistance	Y	Y
onboardAssistance	Y	Y
unaccompaniedMinorAssistance	Y	Y
audioInformation	Y	Y
visualInformation	Y	Y
displaysForVisuallyImpaired	Y	Y
audioForHearingImpaired	Y	Y
bikeCarriage	Y	Y
baggageStorage	Y	Y
nextStopIndicator	Y	Y
stopAnnouncements	Y	Y
passengerInformationFacility	Y	Y
restaurantService	Y	Y
snacksService	Y	Y
trolley	Y	Y
bar	Y	Y

foodNotAvailable	Y	Y
beveragesNotAvailable	Y	Y
bistro	Y	Y
smoking	Y	Y
noSmoking	Y	Y
mobileUseZone	Y	Y
mobilePhoneFreeZone	Y	Y
toilet	Y	Y
noToilet	Y	Y
shower	Y	Y
wheelchairAcccessToilet	Y	Y
babyChange	Y	Y
undefinedServiceFacility	Y	Y
Telephone	Y	Y
passengerWifi	Y	Y
audioServices	Y	Y
videoServices	Y	Y
businessServices	Y	Y
ticketMachines	--	Y
ticketSales	--	Y
mobileTicketing	--	Y

Table 3: ServiceFeature and VehicleFeature (SIRI-1, 5.14).

4.11 Topographic format

(SIRI-1, 5.3, NeTEx-RG, 5.13, NeTEx-1, 7.6.3.2)

LATER

A number of the SIRI functional services include geospatial point coordinates among their response data, for example for the positions of moving public transport vehicles. The actual coordinate system to be used is parameterized: SIRI supports the Geographic Mark-up Language (GML) coordinate formats and data reference systems.

Locations should be defined according to WGS84/GML (normally EPSG:4326). It is recommended to always use the appropriate number of decimal places that correspond to the measurement accuracy. If the data will be used to localize stationary objects like stops or facilities on a map, we recommend a precision of 5 decimal places.

Currently SKI CUS does not use topographic information. In case that the topographic information will be added in the future, CUS will use the same data as the upstream systems.

4.12 Time format and trains after midnight

(NeTEx-RG, 2.4 ; SIRI-1, 5.2; VDV-454, 3.6., p. 31)
Time format consists only of the hour, minutes (and seconds) of a 24 hour clock. E.g. 23:55:00. Times that pass midnight of the current OperatingDay are marked with the “DayOffset” Element.

All timestamps are stated in UTC (Coordinated Universal Time). The use of UTC avoids problems with changeover between summer and winter time zones. Differences from the UTC time zone are coded in accordance with ISO 8601 (e.g.: 2021-04-07T18:39:00Z). In accordance with ISO 8601, if no time difference is given, the time is in UTC; this may be further indicated by the presence of a Z suffix (2021-04-30T12:00:00 corresponds to 2002-04-30T12:00:00Z). In other words, the first 19 characters are obligatory and correspond to local time or UTC.

- ⇨ All deliveries are expected in Zulu-time.
- ⇨ Deviations from this time zone are coded in accordance with ISO 8601 (e.g.: 2021-04-07T18:39:00+01:00).
- ⇨ Time units smaller than one second are ignored (except for timestamp metadata). In case of railway journeys, the seconds of call timings are often ignored (accuracy is generally a minute) and therefore delivered as “00”. However, for regional and local public transport (bus, tram etc.) call timings are typically delivered with non-zero seconds.
- ⇨ If a ServiceJourney runs over midnight, then DayOffset is to be set to “1”. 2 would mean two days and so on. DayOffset can be found in Arrival and Departure.
- ⇨ Times such as 25:30h as a synonym for 1:30h (as is possible in some planning systems and also present in the yearly timetable) is **not** permitted.
- ⇨ Overlaps from one day to the next must be identified by means of a date change in UTC format, e.g.:
  - Before midnight: “2014-07-09T23:55:00”
  - After midnight: “2014-07-10T01:30:00”

⇨ We have negative DayOffsets for international night trains that start abroad. It is based on the information in NeTS and therefore the arrival in Switzerland is the relevant date for the DayOffset.

4.13 Operating day and data frame

(VDV-454, 7., p. 153)
While in the yearly timetable and also VDV454 the operating day is essential, the concept itself does not exist within SIRI. It can, however, be emulated. By the operating day we mean the time period for the validity of timetables within an ITCS (Intermodal Transport Control System). Different ITCS might use different operating days.

- ⇨ A production timetable must always be valid for one operating day
- ⇨ The operating day is mapped to the XML element ***DataFrameRef*** as part of the ***FramedVehicleJourneyRef*** structure.
- ⇨ ***DataFrameRef*** has the date format <yyyy>-<mm>-<dd>, e.g. “2023-02-02”

4.14 Language of text elements

(SIRI-1, 5.4; VDV-454, 10.3, p. 165)
In SIRI a natural language data type, which is a string incorporating the XML lang attribute, is used for text elements that might be provided in different languages. If, for example, a producer wants to provide a ***DestinationName*** in French (“Genève Aéroport”) and German (“Genf Flughafen”), the element will be present twice, each with the repsevtive language attribute. However, in Switzerland the following rules apply:

⇨ Natural language strings must be transmitted in one language - the official one.
In the example above, this would result in a single ***DestinationName*** “Genève Aéroport”.

⇨ In other words, “One per language” of the CEN SIRI specification must be IGNORED and the multiplicity of all natural language string elements is assumed to be at most 1. Consuming system usually display strings as delivered, independently of the transmitted language.

5 Common message types and structures

5.1 SubscriptionRequest

(SIRI-2, 7.1.2)
The Subscription request is sent to the [Subscriber] endpoint of a SIRI Functional Service.

The specific SIRI Functional Service Subscription Requests are wrapped within a general SubscriptionRequest element and any corresponding delivery for the Functional Service is similarly wrapped within a ServiceDelivery element. There is a different SIRI Subscription Request message type for each different SIRI Functional Service type and also a distinct SIRI Subscription Delivery message type for each response type.

When requesting each new Functional Service Subscription, a subscriber can specify both a [Subscriber] endpoint address and a [Notify] endpoint that identifies a separate Consumer address. Confimation that the subscription has been created goes to the [Subscriber] endpoint. The [Notify] endpoint determines the internet address where data ready notifications should be sent.

5.1.1 Business

A fundamental question is whether the consumer must be able to signal to the producer that a message needs to be delivered again. For the time being, **the consumer must resend a SubscriptionRequest in case of a problem**, i.e. renew the subscription! This is the same behavior as specified in VDV (a SubscriptionRequest is equivalent to a VDV453 AboAnfrage).

5.1.2 Structure

Request from Subscriber to Producer for a subscription. Answered with a SubscriptionResponse.

Element	Usage	Structure	Description	Example
RequestTimestamp	1:1	xsd:dateTime	Time of subscription request issued by Requestor.	---
RequestorRef	1:1	→ParticipantCode	Identifier of requestor -- Identifier of Participant.	
See section 3.10 for the allowed format.				
MessageIdentifier	0:1	MessageQualifier	Optional Arbitrary unique reference to the Subscription Request message.	

Used for logging, i.e., is required to be unique per subscription. UUID is recommended.				
		Choice	SIRI Functional Service Subscriptions. For a given SubscriptionRequest, must all be of the same type.	
ProductionTimetableSubscriptionRequest	-1:*	Substructure	See section 6.4 ProductionTimetableSubscriptionRequest	
EstimatedTimetableSubscriptionRequest	-1:*	Substructure	See section 7 EstimatedTimetable.	
SituationExchangeSubscriptionRequest	-1:*	Substructure	See Swiss SX profile based on VDV736.	

5.2 SubscriptionResponse

(SIRI-2, 7.1.3)
After the Notification Producer has received the subscription request, it acknowledges with a single SubscriptionResponse message: this contains a separate ResponseStatus instance for each individual subscription processed. The response is sent to the [Subscriber] endpoint of the request. The response may include an Addresselement which indicates the actual subscription manager endpoint.

5.2.1 Business

-

5.2.2 Structure

Response from Producer to Consumer to inform whether subscriptions have been created. Answers a previous SubscriptionRequest.

Element	Usage	Structure	Description
ResponseTimestamp	1:1	xsd:dateTime	Time individual response element was created.
ResponderRef	1:1	→ParticipantCode	Participant reference that identifies responder.
RequestMessageRef	0:1	→MessageQualifier	Reference to an arbitrary unique reference associated with the request which gave rise to this response.
ResponseStatus	1:1	Substructure	Contains information about the processing of an individual service subscription – either success info or an error condition.
See section 5.3.			

ServiceStartedTime	0:1 (1:1)	xsd:dateTime	Time at which service providing the subscription was last started. Can be used to detect restarts. If absent, unknown.
Mandatory in case of re-subscription (e.g., after a problem occurred). See also chapter 3.8.			

5.3 ResponseStatus

(SIRI-2, 7.1.3)

The ResponseStatus element supplies information on whether an individual SIRI Functional Service subscription request could be processed. If a subscription is granted, the response can include information on the maximum possible update rate of the data producing system, as well as the available data horizon. If the subscription could not be created, it should contain an appropriate error condition. The Subscriber may then resubmit a corrected request to create just these subscriptions.

5.3.1 Business

-

5.3.2 Structure

Contains information about the processing of an individual service subscription - either success info or an error condition.

Element	Usage	Structure	Description	Example
ResponseTimestamp	0:1	xsd:dateTime	Time individual response element was created.	
SubscriptionRef	1:1	→SubscriptionQualifier	Unique identifier of subscription within Service and Subscriber.	
SKI DDIP always generates simple integer numbers 1, 2, 3 etc. For PT, ET and SX data streams typically only a single subscription is needed. Since the service type and participant code are already encoded in the request URL (which is configured before subscriptions are even accepted), an ID of “1” is suffices to uniquely identify the subscription.				
Status	1:1	xsd:boolean	Whether the request could be processed successfully or not. Default is true.	
ErrorCondition	0:1	Substructure	Error Condition that applies to the Termi-nateSubscriptionResponse.	

Mandatory in case of an error or if Status is set to 'false'. See chapter 5.11.				
		Choice	One of the following error codes:	
CapabilityNotSupportedError (a)	–1:1	ErrorStructure	Capability not supported.	
AccessNotAllowedError (b)	–1:1	ErrorStructure	Requestor is not authorised to the service or data requested.	
NoInfoForTopicError (c)	–1:1	ErrorStructure	Valid request was made but service does not hold any data for the requested topic expression.	
AllowedResourceUsageExceededError (d)	–1:1	ErrorStructure	Valid request was made but request would exceed the permitted resource usage of the client.	
OtherError (e)	–1:1	ErrorStructure	Error other than a well-defined category.	
Description	0:1	ErrorDescription	Description of the error in plain text.	
ValidUntil	0:1	xsd:dateTime	End of the data horizon of the data producer; omitted if the request lies completely within the data horizon.	
Confirmation of the InitialTerminationTime if different from what was requested by the subscriber. Time at which the producer expects the termination and renewal of the subscription (since it will no longer be valid).				

5.4 DataSupplyRequest

(SIRI-2, 8.2.3.2)
NOT TO BE USED

5.4.1 Business

DataSupplyRequest is equivalent to VDV453 DatenAbrufenAnfrage, and is only used in implementations using the fetched delivery mechanism. Since only the direct delivery mechanism is supported in Switzerland, DataSupplyRequest is not to be used.

5.5 ServiceDelivery

(SIRI-2, 6.2.2.1)
Delivery responses are sent to [Consumer] endpoint for the request. The delivery may be received as a single step Direct Delivery or as the last step of a Fetched Delivery. A ServiceDelivery message functions as a response to a DataSupplyRequest.

Each Delivery comprises a general ServiceDelivery message, containing one or more SIRI Functional Service delivery responses, for example ConnectionMonitoringFeederDelivery, StopMonitoringDelivery etc.

Most Deliveries contain one or more instances of different types of 'Item' specific to the service, for example MonitoredStopVisit, GeneralMessage etc. All items have a time of recording and an optional identifier; they may variously contain other model elements associated with the item type.

ServiceDelivery contains any general parameters that are common to all delivery types.

5.5.1 Business

Response from Producer to Consumer to deliver payload data. Either answers a direct ServiceRequest or satisfies a subscription asynchronously. May be sent directly in one step or be fetched in response to a Data Supply Request.

A ServiceDelivery is equivalent to a VDV454 “DatenAbrufenAntwort”.

Table 4: Delivery Content Elements:

Delivery	Item (Recorded At)	Primary Association	Children
ProductionTimetableDelivery	DatedTimetable	DatedVehicleJourney	DatedCall
EstimatedTimetableDelivery	EstimatedJourneyVersionFrame	EstimatedVehicleJourney EstimatedServiceJourneyInterchange	EstimatedCall & RecordedCall
VehicleMonitoringDelivery	VehicleActivity	MonitoredVehicleJourney	MonitoredCall
FacilityMonitoringDelivery	FacilityMonitoringDelivery	FacilityCondition	-
SituationExchangeDelivery	SituationExchangeDelivery	PtSituationElement	Affects Consequence PublishingAction

5.5.2 Structure

Element	Usage	Structure	Description	Example
ResponseTimestamp	1:1	xsd:dateTime	Time individual response element was created	2017-05-24T12:01:05Z
ProducerRef	0:1	→PartipicantCode	Participant reference that identifiers producer of data. May be available from context.	
See section 3.10 for the allowed format.				
MoreData	0:1 (1:1)	xsd:Boolean	For multipart despatch, the MoreData element in the ServiceDelivery message indicates whether content of a xxxDelivery contains all the updated data or, whether for implementation reasons, the transmission has been split into several sub-messages. Each MoreData flag in a ServiceDelivery chain indicates that there is further data; in the last ServiceDelivery, MoreData must be set to 'false'. Default is false.	false
MoreData is mandatory in case a ServiceDelivery was split in multiple parts.				
		Choice	One of the following Deliveries:	
ProductionTimetableDelivery	–1:1	Substructure	See section 6 ProductionTimetable.	
EstimatedTimetableDelivery	–1:1	Substructure	See section 7 EstimatedTimetable.	

5.6 DataReceivedAcknowledgement

(SIRI-2, 8.1.2)

If the system supports the SIRI optional ConfirmDelivery capability, and is configured for confirmed delivery, the consumer shall acknowledge that data has arrived. After receiving all the data, the Consumer will return a DataReceivedAcknowledgement message to the [GetData] endpoint of the Producer.

5.6.1 Business

This structure does not exist in VDV. The following behavior is implemented in the dynamic data integration platform (DDIP), depending on whether it acts as the producer or consumer system:

(1) DDIP is the consumer system and receives a DirectDelivery:
DDIP synchronously returns a DataReceivedAcknowledgement. You can set whether this Ack is sent back immediately and the data is stored asynchronously or whether the data is first stored and then the Ack is sent. If the saving goes wrong DDIP responds with a 500 Internal Server Error.

(2) DDIP is the producer system and has sent a DirectDelivery:
DDIP will check whether the HTTP status code is 200/OK (or else the response will be logged in the failed directory) and whether the SIRI payload is valid according to the schema.

In short, as a consumer SKI DDIP will always send a ***DataReceivedAcknowledgement***, but as a producer/supplier we do not necessarily require it (and won't react to failed delivery attempts either).

5.6.2 Structure

Response from Consumer to Producer to acknowledge that data has been received. Used as optional extra step if reliable delivery is needed. Answers a ServiceDelivery.

Element	Usage	Structure	Description	Example
ResponseTimestamp	1:1	xsd:dateTime	Time individual response element was created.	2017-12-17T09:30:47Z
ConsumerRef	0:1	→ParticipantCode	Consumer who is acknowledging the data delivery.	---
See section 3.10 for the allowed format.				
RequestMessageRef	0:1	→MessageQualifier	Reference to a unique identifier associated with the request which gave rise to this response.	---

Status	0:1 (1:1)	xsd:Boolean	Whether data could be processed or not. Default is true. False if error.	false
Mandatory in case of a problem.				
ErrorCondition	0:1 (1:1)	Substructure	Error Condition that applies to a Data-ReceivedAcknowledgement.	
Mandatory in case of an error or if Status is set to 'false'. See also chapter 5.11.				

5.6.3 Example

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0"
xmlns="http://www.siri.org.uk/siri">
  <DataReceivedAcknowledgement>
    <ResponseTimestamp>2019-11-01T15:00:05.556793Z</ResponseTimestamp>
    <ConsumerRef>ddip_int_et</ConsumerRef>
    <Status>true</Status>
  </DataReceivedAcknowledgement>
</Siri>
<!-- CREATOR: DDIP (SI-DDIPAPP01-I)//-->
<!-- TIMESTAMP: 01.11.2019 16:00:12//-->
<!-- REQUEST-URL: http://siri20.ps-xml.et.ddip_test_et.ddipm.int.opentransportdata.swiss:8882/ddip_test_et//-->
<!-- REMOTE_END_POINT-URL: 192.168.46.71:52804//-->
```

5.7 CheckStatusRequest

(SIRI-2, 9.5.2)
Status polling is used to monitor the availability of the SIRI functional services. This can either be on demand Status Check, or an automatic Heartbeat.

Request from Consumer to Producer to check whether services are working. The Endpoint address to which request is sent determines which service is checked. If the client wishes to establish whether the service is still “alive”, it sends a CheckStatusRequest to the server and waits for the reply (CheckStatusResponse). The message is sent to the [CheckStatus] endpoint for the SIRI functional service.

The CheckStatusRequest also enables the client to detect whether a service has been started again, and that the subscriptions have been lost. Within CheckStatusRequest the server specifies the last start time of the service (ServiceStarted timestamp). A start time after the set-up of a subscription indicates that it has been restarted at some point in between.

5.7.1 Business

There is exactly one service polling CheckStatusRequests.

5.7.2 Structure

Element	Usage	Structure	Description	Example
RequestTimestamp	1:1	xsd:dateTime	Time of request.	2017-12-17T09:30:47Z
RequestorRef	1:1	→ParticipantCode	Identifier of requestor - Participant Code.	cus_test
See section 3.10 for the allowed format.				
MessageIdentifier	1:1	MessageQualifier	Arbitrary unique identifier for this message. Can be used to reference it subsequently.	34ffc199-c6be-4b7a-8698-446336b1488f

5.7.3 Example

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.3"
xmlns="http://www.siri.org.uk/siri">
  <CheckStatusRequest>
    <RequestTimestamp>2018-09-26T08:57:19.2854765Z</RequestTimestamp>
    <RequestorRef>cus_test</RequestorRef>
    <MessageIdentifier>e8a619b1-b579-4c02-a045-602c6635a342</MessageIdentifier>
  </CheckStatusRequest>
</Siri>
```

5.8 CheckStatusResponse

(SIRI-2, 9.5.3)

The CheckStatusResponse indicates the availability of the SIRI Functional Service. If the System is completely unavailable there will be no reply. The CheckStatusResponse also provides status information that can be used to establish if there has been an outage. If the Service started time is later than the creation time for the subscription, then it is likely the subscriptions are not current and that the data set of the Consumer may be incomplete.

The CheckStatusResponse is sent to the [ReportStatus] endpoint indicated by the request. The CheckStatusResponse describes the total availability of all information channels of a service, i.e. messages sent to any endpoint, and shall return false if any endpoint is not working. If a single channel is unavailable, the entire service is no longer considered available, i.e. both data supply, subscription management and shall be unavailable.

5.8.1 Business

-

5.8.2 Structure

Response from Producer to Consumer to inform whether services are working. Answers a previous *CheckStatusRequest*.

Element	Usage	Structure	Description	Example
ResponseTimestamp	1:1	xsd:dateTime	Time of Response.	2017-03-24T13:01:05Z
ProducerRef	0:1	→ParticipantCode	Identifies the Producer or Service whose status is being checked.	sncf_prod
See section 3.10 for the allowed format.				
ResponseMessageIdentifier	0:1	MessageQualifier	An arbitrary unique reference associated with the response which may be used to reference it.	
RequestMessageRef	0:1	→MessageQualifier	Reference to identifier of check status message that this response acknowledges.	a4cb68be-3d77-4b07-8210-0d85b258826b
Status	0:1 (1:1)	xsd boolean	Whether the service is available. False if not available. Default is true.	true
Mandatory in case of a problem.				
ErrorCondition	0:1 (1:1)	Substructure	Error Condition that applies to a Check-StatusResponse.	
Mandatory in case of an error or if Status is set to 'false' respectively. See also chapter 5.11.				
ServiceStartedTime	0:1 (1:1)	xsd:dateTime:	Specifies the time of the start of the service. If the service is not available to deliver data, No value should be given here.	2017-03-24T06:01:05Z
Mandatory in case of re-subscription after a problem occurred. See also chapter 3.8.				

5.8.3 Example

CH example with Response Status = true:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0"
xmlns="http://www.siri.org.uk/siri">
  <CheckStatusResponse>
    <ResponseTimestamp>2018-09-26T08:56:47.359328Z</ResponseTimestamp>
    <ProducerRef>cus_test</ProducerRef>
    <RequestMessageRef>530e506b-a9f9-49ff-8691-bb8afe3d6061</RequestMessageRef>
```

```
    <Status>true</Status>
    <ServiceStartedTime>2018-09-13T11:37:29</ServiceStartedTime>
  </CheckStatusResponse>
</Siri>
```

With Response Status = false:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0"
xmlns="http://www.siri.org.uk/siri">
  <CheckStatusResponse>
    <ResponseTimestamp>2018-09-26T08:56:47.359328Z</ResponseTimestamp>
    <ProducerRef>cus_test</ProducerRef>
    <RequestMessageRef>530e506b-a9f9-49ff-8691-bb8afe3d6061</RequestMessageRef>
    <Status>false</Status>
     <ErrorCondition>
         <OtherError>
           <ErrorText>ROOT SERVICE NOT AVAILABLE</ErrorText>
         </OtherError>
     </ErrorCondition>
    <ServiceStartedTime>2018-09-13T11:37:29Z</ServiceStartedTime>
  </CheckStatusResponse>
</Siri>
```

5.9 TerminateSubscriptionRequest

(SIRI-2, 7.3; 7.3.2)
A Subscriber terminates its subscriptions to a service by sending a TerminateSubscriptionRequest to the Subscription Manager, i.e. the [Subscriber] endpoint of the SIRI functional service. A TerminateSubscription-Request may contain either a specific subscription identifier, or a special value of All, indicating that all subscriptions for the subscriber should be terminated.

5.9.1 Business

-

5.9.2 Structure

Request from Subscriber to Subscription Manager to terminate a subscription. Answered with a TerminateSubscriptionResponse.

Element	Usage	Structure	Description	Example
RequestTimestamp	1:1	xsd:dateTime	Creation time of notice of change message.	
RequestorRef	1:1	ParticipantCode	Identifies the Requestor.	
See section 3.10 for the allowed format.				
MessageIdentifier	0:1	MessageQualifier	Arbitrary unique identifier for this message. Can be used to reference it subsequently.	
		Choice	Either All or a named Subscription. Participant identifier of Subscriber. Subscription ref will be unique within this. SIRI v1.3.	
All (a)	–1:1	EmptyType	Terminate all subscriptions for the Subscriber	
SubscriptionRef (b)	–1:1	SubscriptionQualifier	Identifies the specific subscription to be terminated.	

5.9.3 Example

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.3"
xmlns="http://www.siri.org.uk/siri">
  <TerminateSubscriptionRequest>
    <RequestTimestamp>2018-09-26T08:56:54.1051097Z</RequestTimestamp>
    <RequestorRef>ddip_prod</RequestorRef>
    <MessageIdentifier>a179ad7c-be01-4f18-b333-c6334e9c211d</MessageIdentifier>
    <All />
    <!-- <SubscriptionRef>012311</SubscriptionRef> -->
  </TerminateSubscriptionRequest>
</Siri>
```

5.10 TerminateSubscriptionResponse

(SIRI-2, 7.3.3)
The TerminateSubscriptionResponse is sent back to the [Subscriber] endpoint indicated by the request and will contain an acknowledgment or error code for each subscription terminated.

5.10.1 Business

-

5.10.2 Structure

Response from Subscription Manager to Consumer to inform whether subscriptions have been removed. Answers a TerminateSubscriptionRequest.

Element	Usage	Structure	Description	Example
ResponseTimestamp	1:1	xsd:dateTime	Creation time of response.	
ResponderRef	0:1	ParticipantCode	Identifies the Producer	
See section 3.10 for the allowed format.				
RequestMessageRef	0:1	MessageQualifier	Reference to a message for which this is the response	
TerminationResponseStatus	**0:***	Substructure	Status of each response to each subscription termination.	
SubscriptionRef (P) TerminationResponseStatus	1:1	→SubscriptionQualifier	Unique Identifier of Subscription.	
Status (P) TerminationResponseStatus	0:1 (1:1)	xsd:boolean	Whether the subscription could be cancelled. Default is true.	
Mandatory in case of a problem.				
ErrorCondition (P) TerminationResponseStatus	0:1 (1:1)	Substructure	Error Condition that applies to the TerminateSubscriptionResponse.	
Mandatory in case of an error or if Status is set to 'false' respectively. See also chapter 5.11.				

5.10.3 Example

CH example with Response Status = true:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0"
xmlns="http://www.siri.org.uk/siri">
  <TerminateSubscriptionResponse>
    <ResponseTimestamp>2018-09-26T08:56:54.026993Z</ResponseTimestamp>
    <ResponderRef>cus_test</ResponderRef>
    <RequestMessageRef xsi:type="MessageRefStructure">a179ad7c-be01-4f18-b333-c6334e9c211d</RequestMessageRef>
    <TerminationResponseStatus>
      <ResponseTimestamp>2018-09-26T08:56:54.026993Z</ResponseTimestamp>
      <SubscriberRef>ddip_test</SubscriberRef>
      <SubscriptionRef>0001231</SubscriptionRef>
      <Status>true</Status>
    </TerminationResponseStatus>
  </TerminateSubscriptionResponse>
</Siri>
```

With Response Status = false:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0"
xmlns="http://www.siri.org.uk/siri">
  <TerminateSubscriptionResponse>
    <ResponseTimestamp>2018-09-26T08:56:54.026993Z</ResponseTimestamp>
    <ResponderRef>cus_test</ResponderRef>
    <RequestMessageRef xsi:type="MessageRefStructure">a179ad7c-be01-4f18-b333-c6334e9c211d</RequestMessageRef>
    <TerminationResponseStatus>
      <ResponseTimestamp></ResponseTimestamp>
      <SubscriberRef>ddip_test</SubscriberRef>
      <SubscriptionRef>0001231</SubscriptionRef>
      <Status>false</Status>
      <ErrorCondition>
        <UnknownSubscriptionError/>
      </ErrorCondition>
    </TerminationResponseStatus>
  </TerminateSubscriptionResponse>
</Siri>
```

5.11 ErrorConditions for Requests and Deliveries

(SIRI-2, 5.7)

Requests or deliveries may fail for a variety of reasons, in practice the reasons fall into two groups: Systemic and Application:

1. *Systemic error conditions prevent the request from being interpreted further: for example the request times out, or the request itself cannot be validated against the prescribed XML schema version, or the system does not support the requested version level; in which case the request will be rejected outright. The request string is echoed back to assist diagnosis. Typically, the errors occur in the communications or transport layer. They are generic, i.e. not specific to the SIRI application.*

2. *Application level error conditions involve the interpretation of the request parameters and the detection of an error condition according to the semantics of the application. For example, a Stop Monitoring request might ask for information about a stop that is not covered by the system. Most terms of a request have a specific application error condition associated with them.*

Table 5: System and Application Error Conditions

Group	***Condition***	***Description***	***Code***
Success	*OK (true)*	*Request successful*	*200*
Systemic Error	*RequestTimeout*	*Server not responding*	*408*
	InvalidRequest	*The server does not "understand" the request. The client should not repeat the request.*	*400*
	Unauthorized	*User name and password are required for the request, or credentials not satisfied*	*401*
	Forbidden	*The server "understands" the request but cannot carry it out.*	*403*
	NotFound	*The requested URL was not found.*	*404*
Distribution	*UnknownParticipant*	*Recipient for a message to be distributed is unknown.*	*601*

	UnknownEndpoint	*Endpoint to which a message is to be distributed is unknown.*	*602*
	EndpointDeniedAccess	*Distribution message could not be delivered because not authorised.*	*603*
	EndpointNotAvailable	*Recipient of a message to be distributed is not available.*	*604*
Access	*UnapprovedKey*	*User authentication key is not approved.*	*610*
Application Error	*VersionNotSupported*	*Version of SIRI interface is not supported.*	*701*
	CapabilityNotSupported	*Service does not support the requested capability.*	*704*
	ServiceNotAvailable	*Functional service is not available to use (but it is still capable of giving this response).*	*710*
	AccessNotAllowed	*Requestor is not authorised to the service or data requested because a capability is not enabled.*	*720*
	InvalidDataReferences	*Request contains references to identifiers that are not known.*	*730*
	BeyondDataHorizon	*Request is for data outside of real-time data horizon.*	*732*
	NoInfoForTopic	*Valid request was made but service does not hold any data for the requested topic expression.*	*740*

Each Application Error Condition arising from a failed request comprises an error code and a textual description. Each term of a SIRI Functional Service request typically has a different error condition associated with it. The SIRI schema defines explicit error code for application errors wherever possible: these are reified as concrete tags, for example, NoInfoForTopicError, UnknownSubscriber, etc. The most specific error condition should always be returned - the catchall OtherError should only be used in exceptional circumstances.

The following table relates the request terms to their possible error conditions.

Table 6: Application Error Conditions Related to Request Parameters

Group	***Subgroup***	***Request Term***	***Possible Error Condition***	***Notes***
System	*Version*	*SIRIVersion*	*VersionNotSupported*	*Shall respond with same level.*
Endpoint References	*Endpoint*	*ParticipantRef*	*UnknownSubscriber*	*I.e. Control Centre not known.*
	Identity	*SubscriptionIdentifier*	*UnknownSubscription*	---
	Credentials	*Credentials*	*AccessNotAllowed*	*Not known, or not authorised.*
Common Content	*Version*	*SIRIServiceVersion*	*VersionNotSupported*	---
	Topic Filters	*Topic*	*NoInfoForTopic. Capability-NotSupported.*	*Depends on message type.*
		TemporalWindow	*BeyondDataHorizon*	*Depends on message type.*
		DetailLevelToReturn	*CapabilityNotSupported*	*Depends on message type.*
		Language	*CapabilityNotSupported*	---
	Policy	*VolumeFilter*	*AllowedResourceUsage Exceeded*	---
	Delivery	*DeliveryMethod*	*CapabilityNotSupported*	---
Subscription	*Lease*	*InitialTerminationTime*	*BeyondDataHorizon*	*Depends on message type.*
	Notification Filters	*SensitivityFilter*	*AllowedResourceUsage Exceeded*	*Depends on message type.*
	Heartbeat Policy	*HeartbeatPolicy*	*CapabilityNotSupported*	---

5.11.1 Business

The transformation between the SIRI and VDV error protocol **must be handled by DDIP** (Dynamic Data Integration Platform).

- xxxError is equivalent to the VDV453 Fehlernummer in Bestaetigung of a DatenAbrufenAntwort.
- **Careful**: VDV Fehlernummer has mandatory usage (1:1), whereas SIRI ErrorCondition/xxxError is optional (only mandatory in case of an error).
- The child element xxxError/ErrorText is equivalent to VDV453 Fehlertext.

The ErrorText and Fehlertext values can most likely be copied in an import. For xxxError and Fehlernummer, an import requires a value transformation with a lookup-table (ErrorCondition mapping) as well as a custom function. The ErrorCondition structure (child elements “xxxError”, “ErrorText” and possibly “Description”) must be transformed to a simple attribute Fehlernummer and vice versa. Some mapping examples:

- Fehlernummer = “0” in VDV is equivalent to the absence of an ErrorCondition structure in SIRI (since no error is present)
- Fehlernummer between “200” and “299” in VDV is equivalent to an ErrorCondition/InvalidDataReferenceError structure in SIRI (with the respective ErrorText etc.)
- Fehlernummer between “300” and “399” in VDV is equivalent to an ErrorCondition/OtherError structure in SIRI (with the respective ErrorText etc.)
- …mapping TO BE DONE

6 ProductionTimetable

(SIRI-1, B.3.3.4; SIRI-HB, 7)
As each Operational Day approaches, detailed planning will take place, detailing the operational activities for the day and vehicles and drivers that will be assigned to support these activities. The SIRI Production Timetable can provide information about the intended operation for a specified operational day and is published one or more days in advance of the Operational Day in question.

The SIRI Production Timetable Service transmits daily timetables that include any planned updates that are known about at the time of transmission. The service is used typically to communicate between Scheduling systems and AVMS systems, and also between AVMS systems and intelligent clients of the AVMS system to distribute the latest timetables. The timetables exchanged should cover all lines covered by the AVMS system. SIRI-PT may be used in conjunction with the SIRI EstimatedTimetable service to provide a base timetable.

The SIRI Production Timetable Service is also used to transmit the planned interchanges between journeys, including information about the linking of vehicle parts through the interchange, such as whether passengers are able to remain seated in the vehicle:

- *The Request Topics allow a Consumer system to specify that only timetables for a specific timetable version. Operator, Line, Direction or with a specific stop (Haltfilter) are to be returned.*
- *The Delivery can include the times at stops, block and journey pattern information, and information about available Connections.*

An AVMS system may be aware of additional operational journeys and CALLs such as dead runs, and layovers. As the schedule information system usually only knows about VEHICLE JOURNEYs that are relevant to the passenger, the AVMS should only transmit passenger carrying VEHICLE JOURNEYs to the schedule information system.

6.1 Business

SIRI PT is analogous to VDV454 REF-AUS, i.e., provides a consumer with updated timetables at the start of an operating day (i.e., daily) even before real-time data is transmitted over ET (or VDV454 AUS). However, PT data is not necessary for the operation of an ET service (see 10.2).

The maximum number of available validity periods and the time when a production timetable of a given data producer is transmitted will be agreed for the whole supply chain (producer, data hub, consumer).

A daily timetable consists of a number of ***DatedTimetableVersionFrame***s or line (and direction) timetables, and a line timetable consists of a number of ***DatedVehicleJourney***s as illustrated in the following figure:

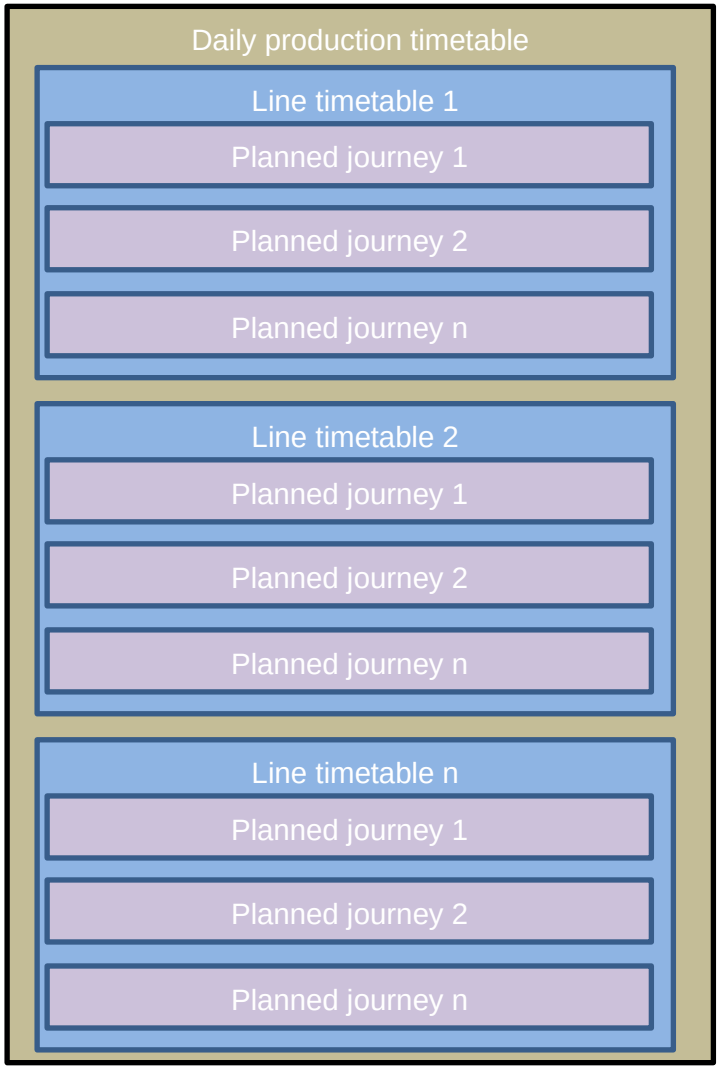


Figure 1: Journey ordering in a daily timetable.

⇨ A “version of the timetable for a LINE and DIRECTION” is always a daily line timetable, i.e., validity period is a full operating day.

⇨ Each ***DatedTimetableVersionFrame*** contains all the journeys (within the validity period) of a single ***LineRef*** and ***DirectionRef***.

A daily production timetable is either delivered as (crucial differences are marked yellow):

a) One single ProductionTimetable delivery containing all line timetables aka ***DatedTimetableVersionFrame*** (for the operating day).
Roughly illustrated as follows (all in one single ServiceDelivery):

ServiceDelivery
ProductionTimetableDelivery
DatedTimetableVersionFrame of Line=”ch:1:Line:151:31” and DirectionRef=”ch:1:Direction:H”
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
… every single journey of this line and direction within the validity period (operating day)
DatedTimetableVersionFrame of Line=”ch:1:Line:151:31” and DirectionRef=”ch:1:Direction:R”
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
… every single journey of this line and direction within the validity period (operating day)
DatedTimetableVersionFrame of Line=”ch:1:Line:151:25” and DirectionRef=”ch:1:Direction:H”
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney

DatedVehicleJourney
… every single journey of this line and direction within the validity period (operating day)
DatedTimetableVersionFrame
for all other line timetables (line and direction combinations) of operator 151 that are relevant for the operating day
…

Typical delivery size is in the 20MB.

However, when transforming such PT deliveries to VDV454 (REF-AUS), DDIP will split single-form-deliveries into individual line timetables of form b) below as most consumer VDV (e.g. SKI CUS) cannot process them in one single enormous delivery.

b) A ProductionTimetable delivery per line timetable aka ***DatedTimetableVersionFrame*** (for the operating day).
Roughly illustrated as follows (separate deliveries for each line timetable):

ServiceDelivery
ProductionTimetableDelivery
DatedTimetableVersionFrame of Line="ch:1:Line:151:31" and DirectionRef="ch:1:Direction:H"
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
… every single journey of this line and direction within the validity period (operating day)
ServiceDelivery
ProductionTimetableDelivery
DatedTimetableVersionFrame of Line="ch:1:Line:151:31" and DirectionRef="ch:1:Direction:R"
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
… every single journey of this line and direction within the validity period (operating day)
ServiceDelivery

ProductionTimetableDelivery
DatedTimetableVersionFrame of Line=”ch:1:Line:151:25” and DirectionRef=”ch:1:Direction:H”
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
DatedVehicleJourney
… every single journey of this line and direction within the validity period (operating day)
ServiceDelivery (for all other line timetables of operator 151 that are relevant for the operating day)
…

In public transport CH we only support “complete line timetables”, i.e., we expect one ***DatedTimetableVersionFrame*** or line timetable per line (***LineRef***) and direction (***DirectionRef***). Moreover, each and every ***DatedTimetableVersionFrame*** must be complete in the sense that it must contain all the journeys of this line and direction that are relevant for a given operating day (from the perspective of passengers).

- ⇨ A line timetable always contains all journeys within the validity period (one operating day).
- ⇨ A production timetable is generally delivered with the transition to the next operating day and therefore precedes any exchange of estimated journey updates. However, should a consumer receive PT data after ET updates for a corresponding journey, the former will be ignored. In other words, PT data must never override ET data for the same data frame or operating day.
- ⇨ A production timetable must also include all journeys that are already active at the time of the subscription start, i.e., where the arrival- or departure time of any stop falls within the validity period.
- ⇨ Journeys that are cancelled in the producing system before the production timetable is delivered must still be delivered in their respective line timetable and flagged with ***Cancellation*** ‘true’. On the other hand, journeys, that are missing in a line timetable compared to the last delivery or the yearly timetable, are assumed (by the consumer) to be cancelled.
- ⇨ An empty line timetable can be a complete line timetable too. An empty line timetable results in the cancellation of all journeys of the line within the provided validity period.
- ⇨ An extra journey (compared to the last PT delivery or the yearly timetable) must always be flagged appropriately with ***ExtraJourney*** set to ‘true’. On the other hand, a journey, that cannot be matched in the yearly timetable or is not present in a previous PT delivery, is assumed (by the consumer) to be an ***ExtraJourney***.

⇨ If the data consumer cannot interpret a line timetable delivery, e.g., because multiple ***DatedVehicleJourney***s are present with the same ID, the complete line timetable should be discarded.

6.4 ProductionTimetableSubscriptionRequest

(SIRI-3, 5.4)
The ProductionTimetableSubscriptionRequest requests the asynchronous delivery of the information described by a ProductionTimetableRequest. The ProductionTimetableSubscriptionRequest Policy parameters control the processing of the subscription.

6.4.1 Business

- A subscription is valid for at most an operating day and must be renewed every morning, typically between 4:00 and 6:00 (at the start of the new operating day).
- Regular subscription changes during the day are not recommended and are only allowed under special circumstances.
- Changes impacting a subscription must always be discussed with the operations team beforehand.

6.4.2 Structure

Element	Usage	Structure	Description	Example
SubscriptionIdentifier	1:1	SubscriptionQualifier	See SIRI Part 2 for common SubscriptionRequest parameters. 7.1.1, SIRI-2 The SubscriptionIdentifier is an Endpoint Property that uniquely identifies each individual functional service subscription. The subscription identifier is made up of two parts: the Participant Reference, and a SubscriptionIdentifier which will be unique within the Subscriber's Participant Reference and SIRI Functional Service type. It will be included in subsequent deliveries to the Consumer, and is also used to	1

			manage the Subscription at a later date. 7.1.2.3, SIRI-2 Identifier to be given to Subscription. Unique within SubscriberRef and service type.	
See section 3.7.				
InitialTerminationTime	1:1	xsd:dateTime	See SIRI Part 2 for common SubscriptionRequest parameters. 7.1.2.3, SIRI-2 Requested end time for subscription. See SIRI subscription procedures.	2017-12-17T09:30:47Z
Typically in the morning when the operating day is chaning. Generally somewhere between the arrival oft he last journey of the old operating day and the departure of the first journey oft he new operating day. Must be stable within a subscription, i.e., must not change in case of an unscheduled re-subscription (e.g. system malfunction).				
ProductionTimetableRequest	1:*	Substructure	ProductionTimetableRequest	See 6.5.3

6.4.3 Example

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0o"
xmlns="http://www.siri.org.uk/siri">
    <SubscriptionRequest>
        <RequestTimestamp>2018-09-26T08:56:54.1362933Z</RequestTimestamp>
        <RequestorRef>ddip_prod</RequestorRef>
        <MessageIdentifier>73ce520d-4919-491a-a327-6113d714bb51</MessageIdentifier>
        <ProductionTimetableSubscriptionRequest>
            <SubscriptionIdentifier>1</SubscriptionIdentifier>
            <InitialTerminationTime>2018-09-27T23:56:54.1362933Z</InitialTerminationTime>
            <ProductionTimetableRequest>
                <RequestTimestamp>2018-09-26T08:56:54.1362933Z</RequestTimestamp>
                <ValidityPeriod>
                    <StartTime>2022-02-20T14:20:00</StartTime>
                    <EndTime>2022-02-20T14:20:00</EndTime>
                </ValidityPeriod>
```

```
                <OperatorRef>ch:1:Organisation:11</OperatorRef>
                <!-- filter by line is only supported in special cases, e.g. exclusion for more efficient testing
                <Lines>
                    <LineDirection>
                        <LineRef>ch:1:Line:11:S1</LineRef>
                        <DirectionRef>ch:1:Direction:H</DirectionRef>
                    </LineDirection>
                </Lines>
                -->
            </ProductionTimetableRequest>
        </ProductionTimetableSubscriptionRequest>
    </SubscriptionRequest>
</Siri>
```

6.5 ProductionTimetableRequest

(SIRI-3, 5.3)
The ProductionTimetableRequest states which timetables should be returned. The ProductionTimetableRequest can be used in both a direct request, and for a subscription. If used for a subscription, additional ProductionTimetableSubscriptionPolicy parameters can be specified. The primary Topic term on the request is the time window for which timetables are to be returned.

- *The Request Topics allow a Consumer system to specify that only timetables for a specific timetable Version. Operator, Line, Direction or with a specific stop (Haltfilter) are to be returned.*

6.5.1 Business

-

6.5.2 Structure

Element	Usage	Structure	Description	Example
RequestTimestamp	1:1	xsd:dateTime	See SIRI Part 2 for common properties of SIRI Functional Service Requests. Time of Request 6.1.4, SIRI-2	2017-12-17T09:30:46Z
ValidityPeriod	1:1	ClosedTimestampRangeStruc-ture	Start and end of timetable validity (time window) of journeys for which schedules	---

			are to be returned. Refers to the departure time at the first stop of each VEHICLE JOURNEY. If blank the configured data horizon will be used.	
StartTime (P) ValidityPeriod	1:1	xsd:dateTime	Start of timetable validity (time window) of journeys for which the timetable must be provided. Refers to the departure time at a stop of each VEHICLE JOURNEY. If blank the configured data horizon will be used.	2017-12-17T09:30:47Z
EndTime (P) ValidityPeriod	1:1	xsd:dateTime	End of timetable validity (time window) of journeys for which the timetable must be provided. Refers to the departure time at the a stop of each VEHICLE JOURNEY. If blank the configured data horizon will be used.	2017-12-17T09:30:47Z
OperatorRef	0:*	→OperatorCode	Filter the results to include only results for the specified operator or operators. SwissBusinessID (SBOID)	ch:1:Organisation:11
Lines	0:1	Substructure	---	---
Filtering by line is only used in special cases, e.g., exclusion of lines for quality reasons.				
LineDirection (P) Lines	0:*		---	---
LineRef (P) LineDirection	1:1	→LineCode	Filter the results to include only results for the given LINE or LINEs.	ch:1:Line:11:S2
DirectionRef (P) LineDirection	0:1	→DirectionCode	Filter the results to include only journeys for VEHICLEs running in a specific relative DIRECTION.	ch:1:Direction:H
See section 4.6 for the allowed values.				

6.6 ProductionTimetableSubscriptionResponse

(SIRI-2, 7.1.3)

-

6.6.1 Business

-

6.6.2 Structure

The structure of a SubscriptionResponse is the same for all Functional Services. See section 5.2.

6.6.3 Example

Example with response Status=true:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0o"
xmlns="http://www.siri.org.uk/siri">
    <SubscriptionResponse>
        <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
        <ResponderRef>ddip_int</ResponderRef>
        <RequestMessageRef>73ce520d-4919-491a-a327-6113d714bb51</RequestMessageRef>
        <ResponseStatus>
            <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
            <SubscriptionRef>1</SubscriptionRef>
            <Status>true</Status>
        </ResponseStatus>
        <ServiceStartedTime>2018-08-26T08:56:54.072608Z</ServiceStartedTime>
    </SubscriptionResponse>
</Siri>
```

With response Status=false:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0o"
xmlns="http://www.siri.org.uk/siri">
    <SubscriptionResponse>
```

```
        <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
        <ResponderRef>ddip_int</ResponderRef>
        <RequestMessageRef>73ce520d-4919-491a-a327-6113d714bb51</RequestMessageRef>
        <ResponseStatus>
            <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
            <SubscriptionRef>1</SubscriptionRef>
            <Status>false</Status>
            <ErrorCondition>
                <OtherError />
                <Description>Unknown error</Description>
            </ErrorCondition>
        </ResponseStatus>
        <ServiceStartedTime>2018-08-26T08:56:54.072608Z</ServiceStartedTime>
    </SubscriptionResponse>
</Siri>
```

6.7 ProductionTimetableDelivery

(SIRI-3, 5.5.1)
In essence the ProductionTimetableDelivery returns a timetable as a versioned set of DatedVehicleJourney instances, each having two or more Dated-Calls, for example that a DatedVehicleJourney may have JourneyPatternInfo and ServiceInfo. DatedCalls may also have TargetedInterchange elements giving information about timetabled connections.

6.7.1 Business

-

6.7.2 Structure

Element	Usage	Structure	Description	Example
ResponseTimestamp	1:1	xsd:dateTime	Time individual response element was created	2017-05-24T12:01:05Z
SubscriptionRef	0:1 (1:1)	→SubscriptionQualifier	Unique identifier of subscription within Service and Subscriber.	
Mandatory within the context of a subscription.				

Status	0:1 (1:1)	xsd:Boolean	Whether the complete request could be processed successfully or not. Default is true. If any of the individual requests within the delivery failed, should be set to false.	false
Mandatory in case of a problem.				
ErrorCondition	0:1 (1:1)		Description of any error or warning conditions that applies to the overall request. More Specific error conditions should be included in the error conditions attached to each functional service response that fails.	
Mandatory in case of an error or if Status is set to 'false' respectively. See also chapter 5.11.				
DatedTimetableVersionFrame	1:1	Structure	A version of the timetable to run on a specified date. See DatedTimetableVersionFrame element.	---
Corresponds to a line-direction timetable as described in section 6.1.				

6.8 DatedTimetableVersionFrame

(SIRI-3, 5.5.4.1)
Each production timetable is returned as a DatedTimetableVersionFrame element. Each DatedTimetableVersionFrame contains a version of the timetable for a LINE and DIRECTION and comprises one or more DatedVehicleJourney elements.

Provides a schedule of DATED VEHICLE JOURNEY for a LINE and DIRECTION.

6.8.1 Business

Equivalent to VDV454 REF-AUS “Linienfahrplan”.

6.8.2 Structure

Element	Usage	Structure	Description	Example

RecordedAtTime	1:1	xsd:dateTime	Time at which data was recorded	2014-12-17T09:30:48Z
LineRef	1:1	→LineCode	Identifier for the LINE.	ch:1:Line:106:23
DirectionRef	1:1	→DirectionCode	Reference to the DIRECTION the VEHICLE is running along the LINE, for example, "in" or "out", “clockwise”. Distinct from a destination.	ch:1:Direction:H
See section 4.6 for the allowed values.				
VehicleMode	1:1	VehicleModesEnum	A means of transportation such as bus, rail, etc.	rail
PublishedLineName	1:1	NLString	Name or Number by which the LINE is known to the public.	
OperatorRef	1:1	→OperatorCode	Filter the results to include results of the specified operator.	ch:1:Organisation:11
ProductCategoryRef	1:1	→TypeOfProductCategory	PRODUCT CATEGORY of journey - classifies, for example; express, local.	ch:1:TypeOfProductCategory:S
See chapter 4.9 for the allowed values.				
ServiceFeatureRef	0:*	→ServiceFeatureCode	Classification of service into arbitrary Service categories, e.g. school bus. SIRI provides a recommended set of values covering most usages. See for example the SIRI facilities enumerations.	
OriginDisplay	0:1	NLString	Name of ORIGIN of journey.	Genf
In the official language only.				
DestinationDisplay	0:1	NLString	Name of DESTINATION of journey.	Zürich HB
In the official language only.				
LineNote	0:*	NLString	Additional text associated with LINE.	Redirected for Carneval
DatedVehicleJourney	0:*	Structure	Provides schedule information about the VEHICLE JOURNEY along which a VEHICLE is running.	

6.9 DatedVehicleJourney

(SIRI-3, 5.5.4.2)
Each DatedVehicleJourney contains an ordered list of DatedCall elements representing the sequence of stop CALLs, as well as other properties that apply to the VEHICLE JOURNEY as a whole. Provides schedule information about the VEHICLE JOURNEY along which a VEHICLE is running.

6.9.1 Business

Calls must always be ordered by the ***Aimed*Time***s in ascending order (origin first/top, destination last/bottom in the stop sequence).
Equivalent to VDV454 REF-AUS “SollFahrt”.

6.9.2 Structure

Element	Usage	Structure	Description	Example
DatedVehicleJourneyCode	0:1 (1:1)	VehicleJourneyCode	Identifies the DATED VEHICLE JOURNEY. The preferred construction is using FramedVehicleJourneyRef from SIRI 2.0. For backward compatibility it is still possible to use the deprecated DatedVehicleJourneyCode	---
Mandatory in case of an extra journey compared to the yearly timetable, i.e., in case ExtraJourney=true.				
FramedVehicleJourneyRef	1:1 (0:1)	Substructure	Identifies the DATED VEHICLE JOURNEY. The preferred construction is using FramedVehicleJourneyRef from SIRI 2.0.	---
May only be omitted in case of an extra journey compared to the yearly timetable, i.e., in case ExtraJourney=true where DatedVehicleJourneyCode is used instead.				
DataFrameRef (P) FramedVehicleJourneyRef	1:1	→DataFrameQualifier	Unique identifier of data frame within participant service. Used to ensure that the DatedVehicleJourneyRef is unique with the data horizon of the producer.	2022-06-15
Operating day of journey.				

DatedVehicleJourneyRef (P) FramedVehicleJourneyRef	1:1	→DatedVehicleJour-neyCode	A reference to the DATED VEHICLE JOURNEY that the VEHICLE is making.	ch:1:ServiceJour-ney:11:99712-0001-1-1
Must be unique within the operating day or DataFrameRef respectively.				
		Choice		
ExtraJourney (a)	–0:1 (–1:1)	xsd:boolean	Whether this journey is an addition to the plan Default is false.	
Mandatory in case of an extra journey compared to the yearly timetable (omitted if false).				
Cancellation (b)	–0:1 (–1:1)	xsd:Boolean	Whether this journey is a Cancellation of a journey in the plan. Default is false.	
Mandatory in case of a cancelled journey compared to the yearly timetable (omitted if false).				
VehicleMode	1:1	VehicleModesEnum	A means of transportation such as bus, rail, etc.	rail
PublishedLineName	1:1	NLString	Name or Number by which the LINE is known to the public.	
OperatorRef	1:1	→OperatorCode	Filter the results to include results of the specified operator.	ch:1:Organisation:11
ProductCategoryRef	1:1	→TypeOfProductCategory	PRODUCT CATEGORY of journey - clas-sifies, for example; express, local.	ch:1:TypeOfProductCate-gory:S
See chapter 4.9 for the allowed values.				
VehicleFeatureRef	0:*	→VehicleFeatureCode	Features of VEHICLE providing journey. Recommended SIRI values based on TPEG are given in SIRI documentation and enumerated in the siri_facilities pack-age.	suitableForWheelChairs
JourneyNote	0:*	NLString	Additional descriptive text associated with journey. One per language.	
OriginDisplay	0:1	NLString	The appropriate text to be used as Origin for this VEHICLE JOURNEY. Can be overridden at individual calling points.	Bern
Name of the origin stop as displayed on monitors in the vehicle.				

DestinationDisplay	0:1	NLString	The appropriate text to be used as Destination for this VEHICLE JOURNEY. Can be overridden at individual calling points.	Zürich HB
Name of the origin stop as displayed on monitors in the vehicle.				
VehicleRef	0:1	→VehicleCode	A reference to the specific VEHICLE assigned to a journey.	
For busses, trams and vehicle types of similar complexity, VehicleRef is generally used to communicate its public identifier that is constant throughout its service life. It is typically an integer number (unique per transport organisation) written on the back, sides and front of the vehicle, i.e., visible to the passengers. This number is stable compared to the TrainNumberRef that typically changes each operating day. Since a train (formation), possibly composed of multiple waggons, doesn't have a single stable identifier but one for each waggon (european vehicle number or also called UIC waggon number), VehicleRef is not used for trains (use TrainNumberRef instead). A use case of VehicleRef is that of a consuming system mapping a given Bluetooth low energy Beacon identifier to a journey (or timetable entry).				
TrainNumbers	1:1	Substructure	TRAIN NUMBERs for journey.	
TrainNumberRef (P) TrainNumbers	1:*	→TrainNumber	TRAIN NUMBER assigned to vehicle journey.	
Must reference the INFO+ journey number (Fahrt-/Zugnummer) in Switzerland.				
JourneyParts	0:1	Substructure	Journey parts making up journey.	
JourneyPartInfo (P) JourneyParts	0:*	Substructure	Information about parts of journey.	
LATER Relevant in splitting or joining scenarios where different parts of a train might have different destinations and also typically formation changes occur.				
DatedCalls	1:1	Substructure	Complete stop sequence along the route pattern, in calling order. CALLs are in order within a JOURNEY PATTERN.	
DatedCall (P) DatedCalls	2:*	Substructure	Individual DatedCall. See below.	
The complete stop sequence matching with the yearly timetable, i.e., including cancelled calls (which are flagged accordingly with Cancellation=true).				

6.10 DatedCall

(SIRI-3, 5.5.4.3)

Each DatedCall describes the PASSING TIMEs of a VEHICLE JOURNEY at a stop, together with other data elements relating to the CALL. A journey must contain at least two CALLs. A DatedCall provides information about a CALL in a DATED VEHICLE JOURNEY.

6.10.1 Business

Equivalent to VDV454 REF-AUS “SollHalt”.

6.10.2 Structure

Element	Usage	Structure	Description	Example
StopPointRef	1:1	→StopPointCode	Reference to a SCHEDULED STOP POINT. Defaults to that of context i.e. that specified on StopPoint.	ch:1:ScheduledStop-Point:8501008
See section 4.7.				
VisitNumber	0:1	VisitNumberType	For JOURNEY PATTERNs that involve repeated visits by a VEHICLE to a stop, the VisitNumber count is used to distinguish each separate visit.	10
StopPointName	0:1	NLString	Name of Stop. Once in a specific language.	Genève
In the official language only.				
		Choice		
ExtraCall (a)	–0:1 (–1:1)	xsd:boolean	Whether this call is an addition to the planned timetable: Default is false.	
Mandatory in case of an extra call compared to the yearly (planned) timetable (omitted if false).				
Cancellation (b)	–0:1 (–1:1)	xsd:Boolean	Whether this call is a cancellation of a previously announced call (or planned according to the long-term timetable). If omitted, defaults to 'false': the call is not cancelled.	
Mandatory in case of a cancelled call compared to the yearly timetable (omitted if false).				
RequestStop	0:1	xsd:boolean	Whether VEHICLE stops only if requested explicitly by passenger. Stop request of the	---

			passanger is necessary for the vehicle to stop,	
OriginDisplay	0:1	NLString	The published ORIGIN at this point in the journey; used to help identify the VEHICLE to the passengers. Since VEHICLEs can change their published ORIGIN during a journey, the Origin included here should be what the VEHICLE will display when it reaches this stop. The ORIGIN DISPLAY is specified by exception: if none is specified, the value from any previous DatedCall instances for the same journey will be used. If there are none then any values from the underlying DatedVehicleJourney will be used, or failing that, from the DatedTimetableVersionFrame.	Zürich HB
In the official language only.				
DestinationDisplay	0:1	NLString	The (destination) text displayed on the vehicle when arriving at a stopThe published DESTINATION at this point in the journey; used to help identify the VEHICLE to the passengers. Since VEHICLEs can change their published DESTINATION during a journey, the destination included here should be what the VEHICLE will display when it reaches this stop. The DESTINATION DISPLAY is specified by exception: if none is specified, the value from any previous DatedCall instances for the same journey will be used. If there are none then any values from the underlying	Zürich HB

			DatedVehicleJourney will be used, or failing that, from the DatedTimetableVersion-Frame.	
In the official language only.				
CallNote	0:1	NLString	Text annotation that applies to this CALL. Once in a specific language.	---
AimedArrivalTime	0:1	xsd:dateTime	Aimed Arrival Time in either the original or Production Timetable. Can be omitted at the end stop.	2017-12-17T09:30:00Z
For trains seconds should be set to 00.				
ArrivalPlatformName	0:1	NLString	Bay or platform name. Inherited property. Can be omitted if the same as the Depar-turePlatformName.	9
See section 4.7 and also use case 10.4.				
ArrivalBoardingActivity	0:1	ArrivalBoardingActivi-tyEnum	Type of alighting allowed at stop. alighting | noAlighting | passthru	Alighting
We do not allow default values here!				
ArrivalStopAssignment	0:1	Substructure	Assignment of arrival of Scheduled STOP POINT to a physical QUAY (platform). If not given, assume same as for departure	---
AimedQuayRef (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY (Platform) to use according to the planned timetable.	
See section 4.7.				

Value	Description
alighting	Passengers may alight.
noAlighting	Passengers may not alight.
passthru	Vehicle does not stop.

Table 9: ArrivalBoardingActivityEnumeration.

AimedQuayName (P) ArrivalStopAssignment	0:*	NLString	Scheduled QUAY (Platform) name. Can be used to indicate a platform change. One per language	
AimedDepartureTime	0:1	xsd:dateTime	Aimed Departure Time in either the original or Production Timetable.	2017-12-17T09:35:00Z
For trains seconds must be set to 00.				
DeparturePlatformName	0:1	NLString	Bay or platform (QUAY) name from which VEHICLE will depart.	8
See section 4.7 and also use case 10.4.				
DepartureBoardingActivity	0:1	DepartureBoardingActivi-tyEnum	Type of boarding allowed at stop. boarding | noBoarding| passthru	boarding

Value	Description
alighting	*Passengers may alight.*
noAlighting	*Passengers may not alight.*
passthru	*Vehicle does not stop.*

Table 10: DepartureBoardingActivityEnumeration.

DepartureStopAssignment	0:1	Substructure	Assignments of departure platform for SCHEDULED STOP POINT to a physical QUAY.	---
AimedQuayRef (P) DepartureStopAssignment	0:1	→QuayCodeType	Physical QUAY (Platform) to use according to the planned timetable.	---
See section 4.7.				
AimedQuayName (P) DepartureStopAssignment	0:*	NLString	Scheduled QUAY (Platform) name. Can be used to indicate a platform change. One per language	

7 EstimatedTimetable

(SIRI-1, B.3.3.5, SIRI-3, 6.1)
The SIRI EstimatedTimetable service is used by an AVMS or real-time hub to inform interested systems of the current status of all known vehicle journeys. This enables schedule information systems to provide up-to-the-minute information for short-term journey planning. It can also be used to support intelligent displays that calculate the deviations from the timetables themselves using a timetable and a real time difference delay by the SIRI Stop Monitoring Service. A further use is the historic logging of changes to journey times. It can be used to exchange changes to a timetable established by the SIRI ProductionTimetable service.

- *The Request Topics allow a Consumer system to specify that only timetables for a specific timetable Version. Operator , Line , Direction or with a specific stop (Haltfilter) are to be returned.*
- *The Subscription Policies allow a subscriber to specify the amount of change to allow before sending an update.*
- *The Delivery returns predicted real-time changes to the timetable.*

7.1 Business

Formation information is only available from SIRI 2.1 onwards.

VDV454 introduces a flag “MitBereitsAktivenFahrten” with which a consumer requests to a producer to also deliver updates for already active journeys at the time when the subscription is established. In SIRI this is the default behavior since no such flag exists. See footnote 1 in section 7.19.5 for the definition of “active”.

7.4 EstimatedTimetableSubscriptionRequest

(SIRI-3, 6.4.1)
The request is used for the asynchronous delivery of the information described by an EstimatedTimetableRequest. The EstimatedTimetableSubscriptionRequest Policy parameters control the processing of the subscription.

7.4.1 Business

See section 7.13 and onwards.

7.4.2 Structure

Element	Usage	Structure	Description	Example
SubscriberRef	0:1	→ParticipantCode	SIRI-2, 7.1.2.1: Identifier of requestor - Identifier of Participant.	sbb_prod
Must contain also the environment as defined in section 3.5.				
SubscriptionIdentifier	1:1	SubscriptionQualifier	Part 2 7.1.1, SIRI-2 The SubscriptionIdentifier is an Endpoint Property that uniquely identifies each individual functional service subscription. The subscription identifier is made up of two parts: the Participant Reference, and a SubscriptionIdentifier which will be unique within the Subscriber's Participant Reference and SIRI Functional Service type. It will be included in subsequent deliveries to the Consumer and is also used to manage the Subscription at a later date. See section 7.1.2.3, SIRI-2. Identifier to be given to Subscription. Unique within SubscriberRef and service type.	sncf_prod
See section 3.7.				
InitialTerminationTime	1:1	xsd:dateTime	Part 2 SIRI-2, 7.1.2.3 Requested end time for subscription. See SIRI subscription procedures.	2018-07-05T01:00:00Z
Is usually set to 24h. The normal subscription time is between 02:00-03:15 in the morning.				
EstimatedTimetableRequest	1:1	Substructure	See EstimatedTimetableRequest.	---

IncrementalUpdates	0:1	Boolean	Whether the producer should return the complete set of data, or only provide updates to the previously returned data i.e. changes to the expected deviation (delay or early time). Default is 'true'. If false each subscription response will contain the full information as specified in this request. (SIRI 2.0)	true
IGNORED AT IMPORT Tag not supported since it is always assumed to be 'true' in Switzerland.				
ChangeBeforeUpdate	0:1	PositiveDurationType	If incremental update, Threshold value in seconds after which the deviations from planned schedule or the last message are to be transmitted. (Optional from SIRI 2.0)	PT30S
IGNORED AT IMPORT A hysteresis value of **30 seconds** is used throughout Switzerland. SBB uses fixed 30 seconds. There may be exceptions (e.g. data from DB).				

7.4.3 Example

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0o"
xmlns="http://www.siri.org.uk/siri">
    <SubscriptionRequest>
        <RequestTimestamp>2018-09-26T08:56:54.1362933Z</RequestTimestamp>
        <RequestorRef>ddip_prod</RequestorRef>
        <MessageIdentifier>73ce520d-4919-491a-a327-6113d714bb51</MessageIdentifier>
        <EstimatedTimetableSubscriptionRequest>
            <SubscriptionIdentifier>123111</SubscriptionIdentifier>
            <InitialTerminationTime>2018-09-27T08:56:54.1362933Z</InitialTerminationTime>
            <EstimatedTimetableRequest version="1.3">
                <RequestTimestamp>2018-09-26T08:56:54.1362933Z</RequestTimestamp>
                <PreviewInterval>PT90M0S</PreviewInterval>
                <OperatorRef>ch:1:Organisation:11</OperatorRef>
                <OperatorRef>ch:1:Organisation:849</OperatorRef>
```

```
                <!-- filter by line is only supported in special cases, e.g. exclusion for more efficient testing
                <Lines>
                    <LineDirection>
                        <LineRef>ch:1:Line:11:S1</LineRef>
                        <DirectionRef>ch:1:Direction:H</DirectionRef>
                    </LineDirection>
                </Lines>
                -->
            </EstimatedTimetableRequest>
        </EstimatedTimetableSubscriptionRequest>
    </SubscriptionRequest>
</Siri>
```

7.5 EstimatedTimetableRequest

(SIRI-3, 6.3)

The request states which real-time timetables to return. The EstimatedTimetableRequest can be used in both, a direct request and for a subscription. If used for a subscription, additional EstimatedTimetableSubscriptionPolicy parameters can be specified.

- *The Request Topics allow a Consumer system to specify that only timetables for a specific timetable Version. Operator , Line , Direction or with a specific stop (Haltfilter) are to be returned.*
- *The Subscription Policies allow a subscriber to specify the amount of change to allow before sending an update.*

7.5.1 Business

-

7.5.2 Structure

Element	Usage	Structure	Description	Example
RequestTimestamp	1:1	xsd:dateTime	See SIRI Part 2 Common properties of SIRI Functional Service Requests. Time of Request 6.1.4, SIRI-2	<RequestTimestamp>2017-06-24T12:01:05Z</RequestTimestamp>
MessageIdentifier	0:1	MessageQualifier	See SIRI Part 2 Common properties of SIRI Functional Service Requests.	<MessageIdentifier>a4cb68be-3d77-4b07-

			Arbitrary unique reference to this message. 6.1.4, SIRI-2	8210-0d85b258826b</MessageIdentifier>
PreviewInterval	0:1	PositiveDurationType	Maximum Preview time in minutes requested by the schedule information system. Preview horizon only limits real time prediction data, information on additional journeys, cancellations, etc., that are beyond the interval may still be included.	30
Preview Intervals can be set between 30 and 180 minutes. The value is in minutes. When starting a new subscription the system should populate the preview with journeys that have already started. The way it does this is currently not fully deterministic: • all still active journeys in the preview interval, • journeys that started in the last 30 minutes, • messages from the last n minutes could be possible mechanisms.				
OperatorRef	1:1	→OperatorCode	Filter the results to include results of the specified operator.	ch:1:Organisation:11
Mandatory in Switzerland. It is needed for filtering (FilterByOperatorRef is always true).				
Lines	0:*	Substructure	Filter the results to include only VEHICLEs along the given LINEs	
Filtering by line is only used in special cases, e.g., exclusion of lines for quality reasons.				
LineDirection (P) Lines	0:*	Substructure	---	---
LineRef (P) LineDirection	0:1	→LineCode	Filter the results to include only results for the given LINE. If no *LineRef* is specified as a subscription filter, this implicitly implies transmission of data for all LINEs in the AVMS.	ch:1:Line:11:IC1

DirectionRef (P) LineDirection		→DirectionCode	Filter the results to include only journeys for VEHICLEs running in a specific relative DIRECTION.	ch:1:Direction:H
See section 4.6 for the allowed values.				

7.6 EstimatedTimetableSubscriptionResponse

(SIRI-2, 7.1.3)

-

7.6.1 Business

-

7.6.2 Structure

The structure of a SubscriptionResponse is the same for all Functional Services. See section 5.2.

7.6.3 Example

CH example with Response Status = true:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0o"
xmlns="http://www.siri.org.uk/siri">
    <SubscriptionResponse>
        <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
        <ResponderRef>ddip_int</ResponderRef>
        <RequestMessageRef xsi:type="MessageRefStructure">73ce520d-4919-491a-a327-6113d714bb51</RequestMessageRef>
        <ResponseStatus>
            <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
            <SubscriptionRef>000023</SubscriptionRef>
            <Status>true></Status>
        </ResponseStatus>
        <ServiceStartedTime>2018-08-26T08:56:54.072608Z</ServiceStartedTime>
    </SubscriptionResponse>
</Siri>
```

With Response Status = false:

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0o"
xmlns="http://www.siri.org.uk/siri">
    <SubscriptionResponse>
        <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
        <ResponderRef>ddip_int</ResponderRef>
        <RequestMessageRef xsi:type="MessageRefStructure">73ce520d-4919-491a-a327-6113d714bb51</RequestMessageRef>
        <ResponseStatus>
            <ResponseTimestamp>2018-09-26T08:56:54.072608Z</ResponseTimestamp>
            <SubscriptionRef>000023</SubscriptionRef>
            <Status>false></Status>
            <ErrorCondition>
                <OtherError />
                <Description>Unknown error</Description>
            </ErrorCondition>
        </ResponseStatus>
        <ServiceStartedTime>2018-08-26T08:56:54.072608Z</ServiceStartedTime>
    </SubscriptionResponse>
</Siri>
```

7.7 EstimatedTimetableDelivery

(SIRI-3, 6.5)
The EstimatedTimetableDelivery returns the predicted arrival times of a VEHICLE or group of VEHICLEs.

7.7.1 Business

-

7.7.2 Structure

Element	Usage	Structure	Description	Example
ResponseTimestamp	1:1	xsd:dateTime	Time individual response element was created	2017-05-24T12:01:05Z

SubscriptionRef	0:1 (1:1)	→SubscriptionQualifier	Unique identifier of subscription within Service and Subscriber.	
Mandatory within the context of a subscription.				
Status	0:1 (1:1)	xsd:Boolean	Whether the complete request could be processed successfully or not. Default is true. If any of the individual requests within the delivery failed, should be set to false.	false
Mandatory in case of a problem.				
ErrorCondition	0:1 (1:1)		Description of any error or warning conditions that applies to the overall request. More Specific error conditions should be included in the error conditions attached to each functional service response that fails.	
Mandatory in case of an error or if Status is set to 'false' respectively. See also chapter 5.11.				
EstimatedJourneyVersionFrame	1:1	Structure	See EstimatedJourneyVersionFrame element.	---

7.8 EstimatedJourneyVersionFrame

(SIRI-3, 6.5.4)
Each production timetable is returned as an EstimatedTimetableVersionFrame element. Each EstimatedTimetableVersionFrame comprises one or more EstimatedVehicleJourney elements grouped for a version of the timetable.

7.8.1 Business

-

7.8.2 Structure

Element	Usage	Structure	Description	Example
RecordedAtTime	**1:1**	xsd:dateTime	Time at which data was recorded.	2017-03-24T12:01:05Z

EstimatedVehicleJourney	**1:***	Substructure	Provides real-time information about a VEHICLE JOURNEY.	
See section 7.9.				
EstimatedServiceJourneyInterchange	0:*	Substructure	Connection parameters for a monitored SERVICE JOURNEY INTERCHANGE between a feeder and distributor journey. SIRI 2.0	---
LATER				

7.9 EstimatedVehicleJourney

(SIRI-3, 6.5.5)
Each EstimatedVehicleJourney contains an ordered list of EstimatedCall elements, as well as other properties.

There are three different ways to match the journeys described in the real-time data to the planned schedule:

- *A full DatedVehicleJourneyRef, if known.*
- *A signature of the journey comprising its origin and destination points and arrival and departure times at those points.*
- *A real-time VEHICLE JOURNEY code (EstimatedVehicleJourneyCode).*

Method two allows installations that do not have a Production Timetable service still to make use of data.

7.9.1 Business

See chapter 7.13 and onwards.
Calls must always be ordered by the ***Aimed*Time***s in ascending order (origin first/top, destination last/bottom in the stop sequence).
Equivalent to VDV454 AUS “IstFahrt”.

Structure

Element	Usage	Structure	Description	Example
RecordedAtTime	0:1	xsd:dateTime	Time at which data for individual journey was recorded if different from that on frame. SIRI 2.0	2018-03-24T12:01:05Z

LineRef	1:1	→LineCode	Reference to the LINE of the VEHICLE JOURNEY.	ch:1:Line:11:IC1
DirectionRef	1:1	→DirectionCode	Reference to the relative DIRECTION the VEHICLE is running along the LINE	ch:1:Direction:H
See section.4.6 for the allowed values. **Careful:** direction names/texts from the perspective of passengers, e.g., "Zürich HB", "Lausanne" etc., are transmitted in ***Origin***-/***DestinationDisplay***.				
FramedVehicleJourneyRef	1:1	Substructure	Reference to the VEHICLE JOURNEY. The Frame is provided by the EstimatedJourneyVersionFrame that contains the journey.	ch:1:ServiceJourney:2018:723:0000:101-1
DataFrameRef (P) FramedVehicleJourneyRef	1:1	→DataFrameQualifier	Unique identifier of data frame within participant service. Used to ensure that the DatedVehicleJourneyRef is unique with the data horizon of the producer.	2022-06-15
Operating day of journey.				
DatedVehicleJourneyRef (P) FramedVehicleJourneyRef	1:1	→DatedVehicleJourneyCode	A reference to the DATED VEHICLE JOURNEY that the VEHICLE is making.	ch:1:ServiceJourney:11:99712-0001-1-1
Must be unique within the operating day or DataFrameRef respectively.				
DatedVehicleJourneyIndirectRef	1:1	Substructure	If no VEHICLE JOURNEY reference is available, identify it by origin and destination and the scheduled times at these stops.	
Journey start- and end timestamp should be filled as well. These start- and stop times must not be altered in an ET update (even in case of partial cancellation) and correspond to the journey as planned in the yearly or production timetable. Mostly redundant information, but it helps some systems in identifying/matching the journeys.				
OriginRef (P) DatedVehicleJourneyIndirectRef	1:1	→ScheduledStopPoint	Origin stop of journey as planned in the yearly or production timetable.	
AimedDepartureTime (P) DatedVehicleJourneyIndirectRef	1:1	xsd:dateTime	Aimed departure time at origin.	
DestinationRef (P) DatedVehicleJourneyIndirectRef	1:1	→ScheduledStopPoint	Destination stop of journey as planned in the yearly or production timetable.	
AimedArrivalTime (P) DatedVehicleJourneyIndirectRef	1:1	xsd:dateTime	Aimed arrival time at destination.	
		Choice		

ExtraJourney (a)	–0:1 (–1:1)	xsd:boolean	Whether this journey is an addition to the plan. Can only be used when both participants recognise the same schedule version. Default is 'false'.	true
Mandatory in case of an extra journey compared to the production or yearly (if PT service is not used) timetable (omitted if false).				
Cancellation (b)	–0:1 (–1:1)	xsd:boolean	Whether this journey is a cancellation of a journey in the plan. Can only be used when both participants recognise the same schedule version. Default is 'false'.	true
Mandatory in case of a cancelled journey compared to the production or yearly (if PT service is not used) timetable (omitted if false).				
VehicleMode	1:1	VehicleModesEnum	A method of transportation such as bus, rail, etc.	
Mandatory in Swiss public transport. See chapter 4.9 for the allowed values.				
PublishedLineName	1:1	NLString	Name or Number by which the LINE is known to the public. One per language	
DestinationDisplayAtOrigin	0:* (0:1)	NLString	DESTINATION DISPLAY name shown for journey at the origin. One per language.	
Via	0:*	Substructure	Description of a VIA point on a journey.	
PlaceRef (P) Via	0:1	→JourneyPlaceCode	Identifier of a VIA point of the journey.	
PlaceName (P) Via	0:* (0:1)	NLString	The Name of a VIA point of the journey, used to help identify the LINE. One per language (Unbounded 0:* since +SIRI 2.0).	
PlaceShortName (P) Via	0:* (0:1)	NLString	Short name of a VIA point of the journey, used to help identify the LINE. One per language (Unbounded 0:* since +SIRI 2.0).	
ViaPriority (P) Via	0:1	xsd:integer	Relative priority to give to VIA name in displays. 1=high. Default is 2.	
OperatorRef	1:1	→OperatorCode	OPERATOR of journey.	ch:1:Organisation:11

Mandatory in Swiss public transport. See section 4.4 for the allowed values				
ProductCategoryRef	1:1	→ProductCategoryCode	PRODUCT CATEGORY of journey - classifies, for example; express, local.	ch:1:TypeOfProductCategory:TGV
Mandatory in Swiss public transport. See section 4.9 for the allowed values.				
ServiceFeatureRef	0:*	→ServiceFeatureCode	Classification of service into arbitrary Service Features, e.g. school bus.	wheelchairAcccessToilet
See section 4.10 for details.				
VehicleFeatureRef	0:*	→VehicleFeatureCode	Feature of VEHICLE. E.g. 'suitableForWheelChairs'.	suitableForWheelChairs
See section 4.10 for details.				
VehicleJourneyName	0:* (0:1)	NLString	Name of VEHICLE JOURNEY. (One per language (Unbound 0:* since +SIRI 2.0).	---
See section 6.9 for details.				
JourneyNote	0:*	NLString	Additional descriptive text associated with journey. One per language (Unbounded 0:* since +SIRI 2.0).	---
See chapter 6.9 for details.				
OriginAimedDepartureTime	0:1	xsd:dateTime	Theoretical departure time of the journey at its origin stop point. Element is from the JourneyEndTimesGroup in VehicleJourneyInfoGroup.	
DestinationAimedArrivalTime	0:1	xsd:dateTime	Theoretical arrival time of the journey at its destination stop point. Element is from the JourneyEndTimesGroup of the VehicleJourneyInfoGroup.	
Monitored	0:1	xsd:boolean	Whether there is real-time information available for journey, if not present, not known. Inheritable.	true
MonitoringError	0:1 (1:1)	xsd:NMTOKEN	If Monitored is 'false', a (list of) reason(s) for non-availability of real-time data. Examples are "GPS", "GPRS",	NoError

			"Radio" etc. MonitoringError must not be used if Monitored is omitted or set to 'true'.	
Mandatory in case Monitored is ‘true’.				
PredictionInaccurate	0:1	xsd:boolean	Whether the prediction was flagged as being inaccurate because of congestion.	false
Occupancy	0:1	OccupancyEnum	An approximate figure of how occupied the journey is after departing from a given stop, e.g. 'manySeatsAvailable' or 'standingRoomOnly'. If omitted: Passenger load is unknown. Occupancies and capacities for individual VEHICLEs, e.g. parts of a COMPOUND TRAIN, can also be specified in more detail for the departure on CALL level.	
VehicleRef	0:1	→VehicleNumber	Designation of vehicle.	
For busses, trams and vehicle types of similar complexity, VehicleRef is generally used to communicate its public identifier that is constant throughout its service life. It is typically an integer number (unique per transport organisation) written on the back, sides and front of the vehicle, i.e., visible to the passengers. This number is stable compared to the TrainNumberRef that typically changes each operating day. Since a train (formation), possibly composed of multiple waggons, doesn't have a single stable identifier but one for each waggon (european vehicle number or also called UIC waggon number), VehicleRef is not used for trains (use TrainNumberRef instead). A use case of VehicleRef is that of a consuming system mapping a given Bluetooth low energy Beacon identifier to a journey (or timetable entry).				
TrainNumbers	1:1	Substructure	One or more TRAIN NUMBER reference.	
TrainNumberRef (P) TrainNumbers	1:*	→TrainNumber	UE regulation 454/2011 primary code or UIC TrainNumber.	
Must reference the INFO+ journey number (Fahrt-/Zugnummer) in Switzerland.				
JourneyParts	0:1	Substructure	One or more JOURNEY PART INFO.	
JourneyPartInfo (P) JourneyParts	1:*	Substructure	Information about Parts of JOURNEY.	
LATER Relevant in splitting or joining scenarios where different parts of a train might have different destinations and also typically formation changes occur.				

RecordedCalls	0:1	Substructure	Complete sequence of stops already visited along the route path, in calling order. Only used if observed stop data is being recorded.	
RecordedCall (P) RecordedCalls	0:*	Substructure	Individual Recorded Call. See below. Only used if observed stop data is being recorded.	
EstimatedCalls	0:1	Substructure	Complete sequence of stops along the route path, in calling order. Normally this is only the onwards stops from the vehicle's current position.	
EstimatedCall (P) EstimatedCalls	2:*	Substructure	Individual EstimatedCall. See below.	
IsCompleteStopSequence	1:1	xsd:boolean	Whether the above CALL sequence is complete, i.e. represents every CALL of the route and so can be used to replace a previous CALL sequence.	true
See section 7.15 and 7.16.				

7.10 RecordedCall

(SIRI-3, 6.5.5.2)
Each RecordedCall describes the observed times at a stop that has already been visited.

7.10.1 Business

See chapter 7.19.

Equivalent to VDV454 AUS "IstHat" with "PrognoseStatus" equal to "Real".

7.10.2 Structure

Element	Usage	Structure	Description	Example

StopPointRef	1:1	→StopPointCode	Reference to a SCHEDULED STOP POINT.	ch:1:ScheduledStop-Point:8507000
See section 4.7.				
VisitNumber	0:1	VisitNumberType	For JOURNEY PATTERNs that involve repeated visits by a VEHICLE to a stop, the VisitNumber count is used to distinguish each separate visit.	20
StopPointName	0:* (0:1)	NLString	Name of stop. One per language.	Bern
		Choice		
ExtraCall (a)	–0:1 (–1:1)	xsd:boolean	Whether this CALL was an addition to the plan.	true
Mandatory in case of an extra stop compared to the production or yearly (if PT service is not used) timetable (omitted if false).				
Cancellation (b)	–0:1 (–1:1)	xsd:boolean	Whether this CALL was a cancellation of a CALL in the plan	false
Mandatory in case of a cancelled stop compared to the production or yearly (if PT service is not used) timetable (omitted if false).				
PredictionInaccurate	0:1	xsd:boolean	Whether the prediction for a specific stop or the whole journey is considered to be of a useful accuracy or not. Default is 'false', i.e. prediction is considered to be accurate. If prediction is degraded, e.g. because of a situation, PredictionInaccurate is used to indicate a lowered quality of data. Inherited property. PredictionInaccurate can be used in combination with InCongestion, but is more general.	false
PredictionInaccurateReason	0:1 (1:1)	PredictionInaccurateReasonEnumeration	Can be used to inform the passenger about the reason for a change of the prediction (in)accuracy in case PredictionInaccurate is set to 'true'. +SIRI v2.1	
Mandatory if PredictionInaccurate is set to ‘true’ from SIRI version 2.1 and onwards				

Occupancy	0:1	OccupancyEnumeration	An approximate figure of how occupied the journey is after departing from a given stop, e.g. 'manySeatsAvailable' or 'standingRoomOnly'. If omitted: Passenger load is unknown. Occupancies and capacities for individual VEHICLEs, e.g. parts of a COMPOUND TRAIN, can also be specified in more detail for the departure on CALL level.	
See Table 13.				
AimedArrivalTime	0:1 (1:1)	xsd:dateTime	Planned arrival time in either the original or Production Timetable. Can be omitted if the time is for the VEHICLE at the origin stop.	2017-03-24T13:01:05+01:00
Must be transmitted for every stop with the exception of the origin (where it is explicitly not allowed).				
ExpectedArrivalTime	0:1	xsd:dateTime	Predicted time of arrival of VEHICLE at time of record. If different from aimed arrival time	2017-03-24T13:02:05+01:00
ActualArrivalTime	0:1	xsd:dateTime	Recorded time of arrival of VEHICLE.	2017-03-24T13:01:35+01:00
If a vehicle does not stop at an expected StopPoint or the actual time cannot be ascertained, then ActualArrivalTime is omitted. The consumer will process the ExpectedArrivalTime (or AimedArrivalTime as the final fallback) instead.				
ArrivalStatus	0:1	CallStatusEnum	Classification of the timeliness of the arrival part of the CALL according to a fixed list of values. This may reflect a presentation policy, for example CALLs less than one minute behind target time are still classified as on-time. Applications may use this to guide their own presentation of times. If not specified, same as DepartureStatus. See Part2 common definitions.	arrived
See Table 14.				

ArrivalPlatformName	0:1	NLString	Bay or platform name. Inherited property. Can be omitted if the same as the DeparturePlatformName. If there not an arrival platform name separate from the departure platform name, the precedence is (i) any arrival platform on any related dated timetable CALL element, (ii) any departure platform name on this CALL element; (iii) any departure platform name on any related dated timetable CALL.	`2
See section 4.7.				
ArrivalBoardingActivity	0:1	ArrivalBoardingActivityEnumeration	Type of alighting activity allowed at stop. alighting | noAlighting | passthru.	noAlighting
See Table 15.				
ArrivalStopAssignment	0:1	Substructure	Assignment of arrival of Scheduled STOP POINT to a physical QUAY (platform). If not given, assume same as for departure.	
AimedQuayRef (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY to use according to the planned timetable.	
See section 4.7.				
AimedQuayName (P) ArrivalStopAssignment	0:1	NLString	Scheduled Platform name. Can be used to indicate a platform change.	2
ExpectedQuayRef (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY to use according to the real-time prediction.	
See section 4.7.				
ExpectedQuayName (P) ArrivalStopAssignment	0:1			
ActualQuayRef (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY actually used.	
See section 4.7.				
ActualQuayName	0:1			

(P) ArrivalStopAssignment				
AimedDepartureTime	0:1 (1:1)	xsd:dateTime	Planned departure time in either the original or Production Timetable. Can be omitted if the time is for the VEHICLE at the origin stop.	2017-03-24T13:06:05+01:00
Must be transmitted for every stop with the exception of the destination (where it is explicitly not allowed).				
ExpectedDepartureTime	0:1	xsd:dateTime	Recorded time of departure of VEHICLE.	2017-03-24T13:01:35+01:00
DeparturePlatformName	0:1	NLString	Bay or platform (QUAY) name from which VEHICLE actually departef.	2
See section 4.7.				
ActualDepartureTime	0:1	xsd:dateTime		
If a vehicle does not actually stop at an expected stop or the actual time cannot be ascertained due to error, then ActualDepartureTime is omitted. The consumer will process the ExpectedDepartureTime (or AimedDepartureTime) instead. See chapter 7.19 for details on how to handle such cases.				
DepartureStatus	0:1	CallStatusEnum	Classification of the timeliness of the departure part of the CALL, according to a fixed list of values. This may reflect a presentation policy, for example CALLs less than one minute behind target time are still classified as on-time. Applications may use this to guide their own presentation of times. See Part2 common definitions.	departed
See ArrivalStatus for the allowed values.				
DepartureBoardingActivity	0:1	DepartureBoardingActivityEnum	Type of boarding activity allowed at stop. boarding | noBoarding | passthru.	noBoarding
See Table 16.				
DepartureStopAssignment	0:1	Substructure	Assignment of departure of Scheduled STOP POINT to a physical QUAY (platform).	
See ArrivalStopAssignment				

7.11 EstimatedCall

(SIRI-3, 6.5.5.3)
Each EstimatedCall describes the estimated future times at a stop. A journey shall contain at least two CALLs of any type.

NOTE
If the Occupancy field in the EstimatedCall structure is filled, this represents a pdicted passenger load. If the corresponding field is filled in a MonitoredVehicleJourney, this overwrites the last current passenger occupancy value.

7.11.1 Business

In case of a cancellation (or extra stop) the reason should always be transmitted as a ***CallNote***, (or CancellationReason from SIRI 2.1 onwards). Be aware that a ***CallNote*** will complement the ***JourneyNote***.

Equivalent to VDV454 AUS “IstHat” with “PrognoseStatus” equal to “Prognose”.

7.11.2 Structure

Element	Usage	Structure	Description	Example
StopPointRef	1:1	→StopPointCode	Reference to a SCHEDULED STOP POINT.	ch:1:ScheduledStop-Point:8501008
See section 4.7.				
VisitNumber	0:1	VisitNumberType	For JOURNEY PATTERNs that involve repeated visits by a VEHICLE to a stop, the VisitNumber count is used to distinguish each separate visit.	30
StopPointName	0:* (0:1)	NLString	Name of stop. One per language.	Bern
		Choice		

ExtraCall (a)	–0:1 (–1:1)	xsd:boolean	Whether this CALL is an addition to the plan. Can only be used when both participants recognise the same schedule version.	true
Mandatory in case of an extra stop compared to the production or yearly (if PT service is not used) timetable (omitted if false).				
Cancellation (b)	–0:1 (–1:1)	xsd:boolean	Whether this CALL is a cancellation of a CALL in the plan. Can only be used when both participants recognise the same schedule version.	true
Mandatory in case of a cancelled stop compared to the production or yearly (if PT service is not used) timetable (omitted if false).				
PredictionInaccurate	0:1	xsd:boolean	Whether the prediction was flagged as being inaccurate because of congestion.	true
Mandatory in case PredictionInaccurate is 'true' from SIRI version 2.1 and onwards. **Table 12: PredictionInaccurateReasonEnumeration** **Value** / **Description** *vehicleInTrafficJam* – Prediction is inaccurate because of a traffic jam *technicalProblem* – Prediction is inaccurate because of technical problems. *dispatchAction* – Prediction is inaccurate because of a despatching alteration. *missingUpdate* – Prediction is inaccurate because communication errors have prevented any updates. *unknown* – Prediction is inaccurate but the reason for an inaccurate prediction is unknown. See also section 7.17 for details.				
Occupancy	0:1	OccupancyEnum	How full the VEHICLE is at the stop. Enumeration. If omitted: Occupancy is as for journey. Enumeration. See Part2 Common definitions. full | seatsAvailable | standingAvailable	

Table 13: Occupancy mapping proposal VDV - SIRI

SIRI	VDV	Remarks
Full	überfüllt	
standing available	stark besetzt	In our opinion the meaning deviates slightly depending on its interpretation. It is useful to know what the producing system means with “highly crowded” and whether it is equivalent to the expectations of the consumer system with “standing available”.
seats available	schwach besetzt	
n.a.	unbekannt	In case that some occupancy information was available and it changes to “not known = DE: unbekannt” status because it was not transmitted then the previous known value for occupancy must be removed.

BoardingStretch	0:1	xsd:boolean	Whether this is a Hail and Ride stop.	true
RequestStop	0:1	xsd:boolean	Whether VEHICLE stops only if requested explicitly by passenger.	true
OriginDisplay	0:* (0:1)	NLString	The name of the origin of the VEHICLE JOURNEY; used to help identify the VEHICLE to the public: Since VEHICLEs can change their displayed origin during a journey, the origin included here should be what the VEHICLE will display as it reaches this stop. (SIRI v2.0). One per language	St. Gallen
DestinationDisplay	0:* (0:1)	NLString	The name of the destination of the VEHICLE JOURNEY; used to help identify the VEHICLE to the public: Since VEHICLEs can change their destination during a journey, the destination included here should be what the VEHICLE will display when it reaches this stop. One per language.	Luzern
Should only be transmitted if different from DestinationDisplay on journey level, i.e., if it depends on the position of the vehicle within the stop sequence.				

CallNote	0:* (0:1)	NLString	Text annotation that applies to this CALL.	Waiting for another train
Currently we cannot provide CallNotes in different languages, only one CallNote per notice in all available languages. With the next VDV version it will be possible to have notice text in different languages.				
AimedArrivalTime	0:1 (1:1)	xsd:dateTime	Planned arrival time in either the original or Production Timetable. if the time is for the VEHICLE at the origin stop.	2018-07-04T06:29:00Z
Must be transmitted for every stop with the exception of the origin (where it is explicitly not allowed)..				
ExpectedArrivalTime	0:1	xsd:dateTime	Estimated time of arrival of VEHICLE.	2018-07-04T06:31:48Z
See also chapter 7.17. Whenever ExpectedArrivalTime is not available/set, AimedArrivalTime (target time) is to be used as the expected time.				
ArrivalStatus	0:1	CallStatusEnum	Classification of the timeliness of the arrival part of the CALL according to a fixed list of values. This may reflect a presentation policy, for example CALLs less than one minute behind target time are still classified as on-time. Applications may use this to guide their own presentation of times. If not specified, same as DepartureStatus. See Part2 common definitions.	arrived

Table 14: CallStatus - allowed values and usage

CallStatus	Description
onTime	can be used to indicate that omitted expected times are to be interpreted as equal to the aimed times or in other words on time.
early	available, but currently not actively communicated
delayed	a state which always applies when not "onTime" and not "early"
cancelled	can be used in special cases (partial cancellations) where only the departure is cancelled (see example in chapter 7.16.2). If both the arrival and departure are cancelled, always use the separate element ***Cancellation*** instead.
arrived	can be used in case a vehicle has actually arrived at a stop but not yet departed
departed	not used since an EstimatedCall must be transformed into a RecordedCall as soon as the vehicle has actually departed.
missed	available ("indefinite delay")

The same values and information are used in DepartureStatus.

ArrivalPlatformName	0:1	NLString	Bay or platform name. Inherited property. Can be omitted if the same as the DeparturePlatformName. If there not an arrival platform name separate from the departure platform name, the precedence is (i) any arrival platform on any related dated timetable CALL element, (ii) any departure platform name on this CALL element; (iii) any departure platform name on any related dated timetable CALL.	`2
See section 4.7.				
ArrivalBoardingActivity	0:1	ArrivalBoardingActivityEnumeration	Type of alighting activity allowed at stop. alighting | noAlighting | passthru.	alighting

Table 15: ArrivalBoardingActivityEnumeration

Value	Description
alighting	Passengers may alight.
noAlighting	Passengers may not alight.
passthru	Vehicle does not stop.

ArrivalStopAssignment	0:1	Substructure	Assignment of arrival of Scheduled STOP POINT to a physical QUAY (platform). If not given, assume same as for departure.	
AimedQuayRef (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY to use according to the planned timetable.	
See section 4.7.				
AimedQuayName (P) ArrivalStopAssignment	0:1	NLString	Scheduled Platform name. Can be used to indicate a platform change.	2
ExpectedQuayRef (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY to use according to the real-time prediction.	
See section 4.7.				
ExpectedQuayName (P) ArrivalStopAssignment	0:1	→QuayCodeType	Physical QUAY actually used.	
AimedDepartureTime	0:1	xsd:dateTime	Timetabled departure time of the VEHICLE in either the original or Production Timetable.	2017-12-17T09:30:00Z
Must be transmitted for every stop with the exception of the destination (where it is explicitly not allowed).				
ExpectedDepartureTime	0:1	xsd:dateTime	Estimated time of departure of the VEHICLE from the onwards timing point..	2017-12-17T09:31:48Z
See also chapter 7.17. Whenever ExpectedArrivalTime is not available/set, AimedArrivalTime (target time) is to be used as the expected time.				
DepartureStatus	0:1	CallStatusEnum	Classification of the timeliness of the departure part of the CALL, according to a fixed list of values. This may reflect a presentation policy, for example CALLs less than one minute behind target time are still classified as on-time. Applications	arrived

			may use this to guide their own presentation of times. See Part2 common definitions.	
See ArrivalStatus for the allowed values.				
DepartureBoardingActivity	0:1	DepartureBoardingActivityEnum	Type of boarding activity allowed at stop. boarding | noBoarding | passthru.	boarding

Table 16: DepartureBoardingActivityEnum

Value	Description
boarding	Passengers may board.
noBoarding	Passengers may not board.
passthru	Vehicle does not stop.

DepartureStopAssignment	0:1	Substructure	Assignment of departure of Scheduled STOP POINT to a physical QUAY (platform).	
See ArrivalStopAssignment				

7.13 Data freshness

It is expected that new messages are published as soon as feasible after the source data has been changed. For example, the following events trigger an ***EstimatedJourney*** update:

- **A.** Hysteresis occurs
- **B.** Preview interval is entered (not always necessary)
- **C.** Change of journey or call attributes
  - **C.1** Update of the expected arrival or departure times - 10.3
  - **C.2** Update of the platform - 10.4
  - **C.3** Update of the train formation - 10.17
  - **C.4** Update of the journey or call notes
  - **C.5** Update of the origin or destination display texts
  - **C.6** Update of occupancy (for example after vehicle has departed at a stop) - 10.18
  - **C.7** Other attribute changes
- **D.** Change of the route
  - **D.1** Cancellation of stops - 10.7 and 10.9

D.2 Addition of extra stops - 10.8
D.3 Other route changes - 10.10

E. Complete cancellation of a journey (only possible prior to the start of the journey) - 10.6
F. Delivery of an extra journey (with respect to the planned/periodic timetable) - 10.5
G. Data transmission is lost (Monitored set to 'false')
H. Data transmission is re-established (Monitored set to 'true')
I. Change of the feeder or distributor journey and the related interchanges
J. Actual data is recorded when arriving or departing at a stop (***EstimatedCall*** is replaced by a ***RecordedCall***) - 7.19.2 to 7.19.16
K. And other events

7.14 Disregarding the PreviewInterval

It is important to note that the ***PreviewInterval*** can or even must be disregarded in some instances, e.g. in case of significant despatching alterations (see a. in 7.16 for a list) like cancellations, exceptional stops, journey replacement, changes to the route or train formation etc. The sooner such updates are transmitted, the sooner customer information systems like journey planner can inform the passengers.

7.15 Completeness of incremental updates

To reduce the amount of data transferred from producer to consumer the SIRI concept of incremental updates can be applied in this profile.

Using this mechanism, the same information is not repeated over and over again, instead only updated attributes for an element are transferred, and only updates for those elements that have any updates are transferred.

There are some exceptions to this reduction mechanism:

a. The delivered updates must be valid according to the applicable SIRI XML Schema definitions. This means that mandatory elements will always be included when their parent element is included in the update, whether their content is changed or not.
b. The rules introduced in the following chapter 7.16 must be observed.

Note that this reduction mechanism does not rule out that incremental updates occasionally include previously transferred values for technical reasons or as a feature for consumers.

7.16 Incremental updates and IsCompleteStopSequence

CEN/TS 15531-3, chapter 6.8.4 “Changes to Journeys & Routing” provides the option to either communicate journey updates as incremental updates or complete stop sequences (with or without cancelled stops) in case of major despatching alterations (e.g. partial cancellations and extra calls).
In the context of the EPIP-RT incremental updates are generally supported and recommended but with some restrictions (some were already introduced in the preceding chapter 7.15). The following rules must be observed when using incremental updates:

1. After a significant despatching alteration the very first ***EstimatedVehicleJourney*** update must be a complete stop sequence, i.e.,
   a. This rule applies in the case of additional journeys compared to the production or yearly timetable as well as for platform changes, partial or complete cancellations, general route changes, formation changes and most of the use cases described in chapter 10.
   b. This rule does not apply in the case of delays (10.3), change of boarding activity (10.13 to 10.15), change of occupancy (10.18) and generally in case of a ***JourneyRelation*** being added (10.16) without changes to the stop sequence. See the subsequent rule 2. in this case.
   c. ***IsCompleteStopSequence*** = ‘true’.
   d. The complete and currently valid state of the journey (with all the ***EstimatedCalls*** as well as ***RecordedCalls***) must be transmitted.
   e. Whether cancelled calls must be transmitted and marked explicitly (with ***Cancellation*** = ‘true’) or omitted entirely depends on the consumer, i.e., type of customer information system, its capabilities, how it matches the real-time data with the planned timetable and what exactly it wants to display to the passengers. We differentiate between the two cases described in the subsequent sections 7.16.1 and 7.16.2. However, the latter (full history of journey) is always the preferred variant .
   f. Whether additional calls (with respect to the previously transmitted state of the journey) must be marked explicitly (with ***ExtraCall*** = ‘true’) or not depends on the consumer, i.e., type of customer information system, its capabilities, how it matches the real-time data with the planned timetable and what exactly it wants to display to the passengers. We differentiate between the two cases described in the subsequent sections 7.16.1 and 7.16.2. However, the latter (full history of journey) is always the preferred variant.
   g. The calls are sorted in descending order according to their aimed arrival and departure time (origin stop = top list item; destination stop bottom list item).
   h. Optional elements with default value must be transmitted if their current value differs from the default value (regardless of whether the current value has changed in relation to the last update for this journey).
   i. Generally, the ***EstimatedVehicleJourney*** must contain all attributes for which no reference or master data exists.
   j. ***DatedVehicleJourneyIndirectRef***, if previously transmitted, must remain constant analogous to ***FramedVehicleJourneyRef***, even if the origin and destination have changed in an update, as both are unique identifiers used in the matching process with the planned timetable.

k. As a result of ***IsCompleteStopSequence*** = ‘true’, the first ***EstimatedVehicleJourney*** update after a significant despatching alteration sets a fresh initial state and therefore replaces all the previously received information for this journey.

2. In the case of delays (10.3), change of boarding activity (10.13 to 10.15), change of occupancy (10.18) and generally in case of a ***JourneyRelation*** being added (10.16) without changes to the stop sequence, incremental updates are highly recommended. This also applies for updates after significant despatching alterations, given that at least one ***EstimatedVehicleJourney*** with ***IsCompleteStopSequence*** = ‘true’ was previously transmitted according to rule 1.

In essence, these two rules imply that incremental updates are only supported in specific use cases and not allowed otherwise. Rule 2 will already reduce the data volume significantly.

7.16.1 IsCompleteStopSequence and active state of a journey

A customer information system, e.g. journey planner, only wants to show the currently active or valid state of the journey and is not interested in the full history or it wants to calculate the delta itself. From a journey planner perspective, consequences of despatching alterations (or deltas) are generally calculated by comparing the original route according to the timetable or the route according to a previous real-time update with the incoming journey. For such a topological comparison of the nodes and edges of a routing graph that incorporates real-time updates, too much information in the form of explicitly transmitted but cancelled stops as well as flagged extra calls might increase the complexity significantly. The room for potential errors on data exchange and producer side of things is high and a likely result might be that a journey planner and its delta calculation arrive at a different conclusion than what the SIRI information or the data producer implies. A complete stop sequence must therefore, at any given point in time of the journey, only contain the recorded stops where the vehicle actually departed and the still to be called stops where the vehicle will arrive according to the newest estimates at that point in time.
Within the context of this use case, the following rules replace the placeholders e. and f. in addition to all other rules of 1. in the preceding section 7.16:

e. Any cancelled call must be omitted, i.e., is neither transmitted as ***Recorded***- nor ***EstimatedCall***.
f. Any additional call (with respect to the previously transmitted state of the journey) is explicitly transmitted as ***Recorded***- or ***EstimatedCall*** analogous to regular calls without the ***ExtraCall*** flag.

Consider a journey with stop sequence … **A => B => C** … according to the planned timetable. This journey is affected by a rerouting (see 10.10) where an extra call at stop **D** replaces the cancelled stops **B** and **C**:

- 1st journey update, after the despatching alteration is triggered in the SIRI source system, must be the complete stop sequence … **A => D** … according to the rules above and in chapter 7.16.
- 2nd journey update for a refinement of the expected times at the extra stop could be an incremental update with call **D** only.

7.16.2 IsCompleteStopSequence and full history of a journey

A customer information system, e.g. journey planner, wants to show why the original journey (as per previously transmitted state) can't be made. The full history is also necessary in cases where a consuming system doesn't match (or compare) the real-time update with the planned journey from the timetable. One of the benefits of explicitly transmitting cancelled calls is that the producer can provide additional information with them, for example as a ***CancellationReason*** or ***CallNotes***.

Within the context of this use case, the following rules replace the placeholders e. and f. in addition to all other rules of 1. in the preceding section 7.16:

e.

- All cancelled calls, for which the ***AimedArrivalTime*** is not yet reached at the time of the update, are explicitly transmitted as ***EstimatedCall*** and marked with ***Cancellation*** = 'true'. It is highly recommended to always send additional information as ***CancellationReason*** or ***CallNotes*** that might be relevant for the passengers.
- All cancelled calls, for which the time of the update is already past the ***AimedArrivalTime***, are explicitly transmitted as ***RecordedCall*** and marked with ***Cancellation***=true.

f.

- All additional calls (with respect to the previously transmitted state of the journey) are explicitly transmitted as ***EstimatedCalls*** and marked with ***ExtraCall*** = 'true' (unless it is an additional journey, i.e., has ***ExtraJourney*** = 'true', then ***ExtraCall*** is implicitly true for all calls).
- Whenever an ***EstimatedCall*** with ***ExtraCall*** = 'true' is recorded, the ***ExtraCall*** flag is retained in addition to the application of all the rules described in chapter 7.19.2 to 7.19.16 (as in the standard case).

Consider a journey with stop sequence … **A => B => C** … according to the planned timetable. This journey is affected by a rerouting (see 10.10) where extra calls at stop **D** and **E** replace the cancelled stops **B** and **C**:

- 1st journey update, after the despatching alteration is triggered in the SIRI source system, must be the complete stop sequence:
  … **A => B** (***Cancellation*** = 'true') **=> D** (***ExtraCall***= 'true') **=> C** (***Cancellation*** = 'true') **=> E** (***ExtraCall***= 'true') … sorted by the aimed times and according to the rules above as well as in chapter 7.16.
- 2nd journey update for example after a refinement of the expected times at the extra stop could be an incremental update with call **D** only.
- …
- Nth journey update after the vehicle actually arrived at stop **A** would be an incremental update with the newly recorded call **A** only.
- In the (N+1)th journey update (incremental or not) after the vehicle would have passed the actually cancelled stop **B** (i.e. its ***AimedArrivalTime***) the producer will also transmit a ***RecordedCall*** for said stop **B**, since the full history is preserved. In this case of a "virtual" stop where the vehicle neither actually arrives nor departs, the call recording can be triggered by any event described in chapter 7.13 as long as the ***AimedArrivalTime*** of the cancelled stop is reached. It is <u>not</u> recommended to deliver a ***RecordedCall*** update of a cancelled stop separately / on its own, compared

to regular ***RecordedCall*** updates that must trigger a delivery on their own. Note that the ***Cancellation*** = 'true' flag must be retained in the ***RecordedCall***.

- Suppose the worst case happens and extra stop **E** (the new destination) is also cancelled in the "last second" while the passengers are sitting in the vehicle and waiting for the departure at stop **D**. Now, according to the above rules, the last complete stop sequence for this journey will be:
  … **A** (recorded) **=> B** (recorded with ***Cancellation*** = 'true') **=> D** (arrival recorded with an ***ActualArrivalTime***, ***DepartureStatus*** = cancelled as well as ***Cancellation*** = 'true') **=> C** (***Cancellation*** = 'true') **=> E** (***Cancellation*** = 'true') …
  By combining the information that
  - **E** is the new destination stop since the 1st journey update,
  - the ***RecordedCall*** for stop **D** is now marked with ***Cancellation***=true and that
  - no ***ActualDepartureTime*** was recorded

  consuming systems like displays or journey planner can determine the appropriate actions.

7.17 Handling of Predictions

(SIRI-3, 6.6)

i. If a VEHICLE logs off from its block or becomes temporarily unattainable ('loss of contact'), the Producer must be able to revoke a previously reported prediction. In this case, for every VEHICLE JOURNEY already reported, the Producer can send an ***EstimatedVehicleJourney*** message with ***Monitored*** set to 'false' on journey level.

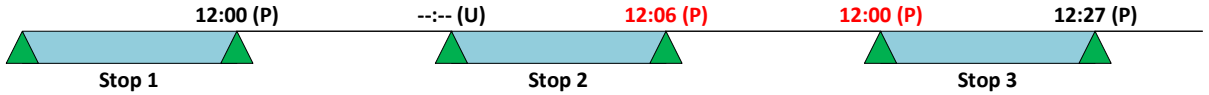
ii. If the VEHICLE is experiencing unusual conditions, such as road blockages, making it hard to make reliable prognosis the ***PredictionInaccurate*** element can be activated. It is up to the producer to continue providing prognosis updates or not.

iii. If a prediction is omitted in an update message, the data producer usually wants to indicate that no prediction is available, e.g., if ***Expected*Time*** is unknown. Conversely, if no prediction is available for the arrival or departure at a stop, i.e., ***Expected*Time*** is unknown or the inaccuracy exceeds a threshold, then the respective ***EstimatedCall*** must not have an ***Expected*Time*** (XML tags are omitted). In such a case, it is up to the consuming system how to proceed and whether it should fall back to the scheduled/aimed times. However, from SIRI 2.1 onwards data producer must indicate unknown predictions by using ***Arrival***- and ***DeparturePredictionUnknown***.

iv. According to SIRI part 3, ***EstimatedCall*** objects (as elements of the ordered list ***EstimatedCalls***) are sorted "in calling order", i.e., by the arrival and departure time in ascending order, with the origin stop being the first (after XML "<" or starting tags of the call sequence) and the destination stop being the last entry in the list (before XML "</" or ending tags of the call sequence). The same applies to ***RecordedCalls***. The following cases of inconsistent predictions are forbidden and must be smoothed out by the data producer (e.g., by applying an update rule to the ***Expected*Time***s of the subsequent calls):

   a. A vehicle must not depart at stop [n] earlier than it arrives at said stop [n].

b. A vehicle must not arrive earlier at stop [n+1] than it departs at the preceding stop [n] (see the times marked red in the following picture).



P = predicted/expected as per ***EstimatedCall***
U = unknown/missing prediction

A stop sequence with arrival or departure times not in ascending order potentially leads to problems with routing or information display of interchanges. See also the examples in use case chapter 10.3.

7.17.1 Quality of Prognoses and Prediction Windows

Customers expect VEHICLE JOURNEY information from real-time enabled systems to be more accurate, informative and timely than the planned schedules. For example, with the help of the SIRI Estimated Timetable service the schedule information system can mark VEHICLE JOURNEYs as "delayed x min", "punctual" or "cancelled".

Systems should therefore be able to distinguish between there being no messages from the Producer because there is no significant delay, from the case when the Producer is unable to generate any messages for the VEHICLE at all, for example, because the VEHICLE has no radio equipment. In the first case the VEHICLE JOURNEY should be explicitly marked as punctual in the schedule information system; and in the latter case, as having an unknown accuracy. SIRI includes a number of elements (Monitored, PredictionInaccurate, InCongestion, InPanic) that may be used to represent this.

In order to be able to make reliable predictions, the schedule information system should base its VEHICLE JOURNEY information on information from explicit messages, and not on conclusions drawn from the assumed successful passing of events.

The further ahead in time a prediction is being attempted, the less certain factors will influence it, and so the greater likelihood of inaccuracy. This increasing indeterminism also applies to the prediction of traffic flows. In practice, predictions are only reliable for VEHICLE JOURNEYs in the near future. To avoid misrepresenting the apparent accuracy of prognoses, every producer system has a restricted PreviewInterval, which defines a rolling forward window within which predictions will be attempted.

In Switzerland for interconntected VEHICLE JOURNEY (e.g. with stay-seated). The prognoses for the second VEHICLE JOURNEY should also be provided, when it is outside the preview interval, when the first VEHICLE JOURNEY starts.

7.17.2 Reporting Rules and Monitored Journeys

The Producer shall report every VEHICLE JOURNEY for which there are predictions at least once.

The Producer only transmits the predictions for the VEHICLE JOURNEYs that start within the Preview Interval. A future VEHICLE JOURNEY lies within the Preview Interval if the departure time of the VEHICLE JOURNEY at the origin SCHEDULED STOP POINT lies in the time window from the current time until the end time given by the Preview Interval.

The prediction message shall be generated as early as possible within the Preview Interval, that is, as soon as the Producer can give a prediction for the start of the VEHICLE JOURNEY. If this is not possible, a message shall be generated for the VEHICLE JOURNEY after the comparison between the planned and real-time data (after logon, after first radio contact, before/at/after the start stop).

As soon as a follow-on VEHICLE JOURNEY falls into the Preview time, it too shall be reported with its entire delay profile as soon as possible (i.e. with the prediction for the start stop and all subsequent stops).

For every transmitted VEHICLE JOURNEY, the predicted delay profile is reported up to the end of the VEHICLE JOURNEY, even if the VEHICLE JOURNEY finishes after the prediction horizon.

A VEHICLE JOURNEY that has been reported once shall continue to be reported until it is completed, or until predictions become impossible. The schedule information system can therefore assume a VEHICLE JOURNEY to be punctual if it has received an initial message for the journey from the Producer, and if the Monitored property has not been set to false. Without the active transmission of this information, the journey is regarded as unmonitored, and the schedule information system switches back to using only planning data for the journey.

This procedure ensures that the schedule information system only marks VEHICLE JOURNEYs as punctual when they are actually monitored by the Producer, and are being transmitted without error. It is possible to further improve the reliability if, in the case of punctual vehicles too, the Producer continues to send messages from time to time after the first real-time message.

The disadvantage is the slightly greater volume of data that needs to be transmitted. The re-initialisation of a real-time data subscription (e.g. after a restart) in particular would generate a new message to the schedule information system from every VEHICLE affected by the subscription currently in operation.

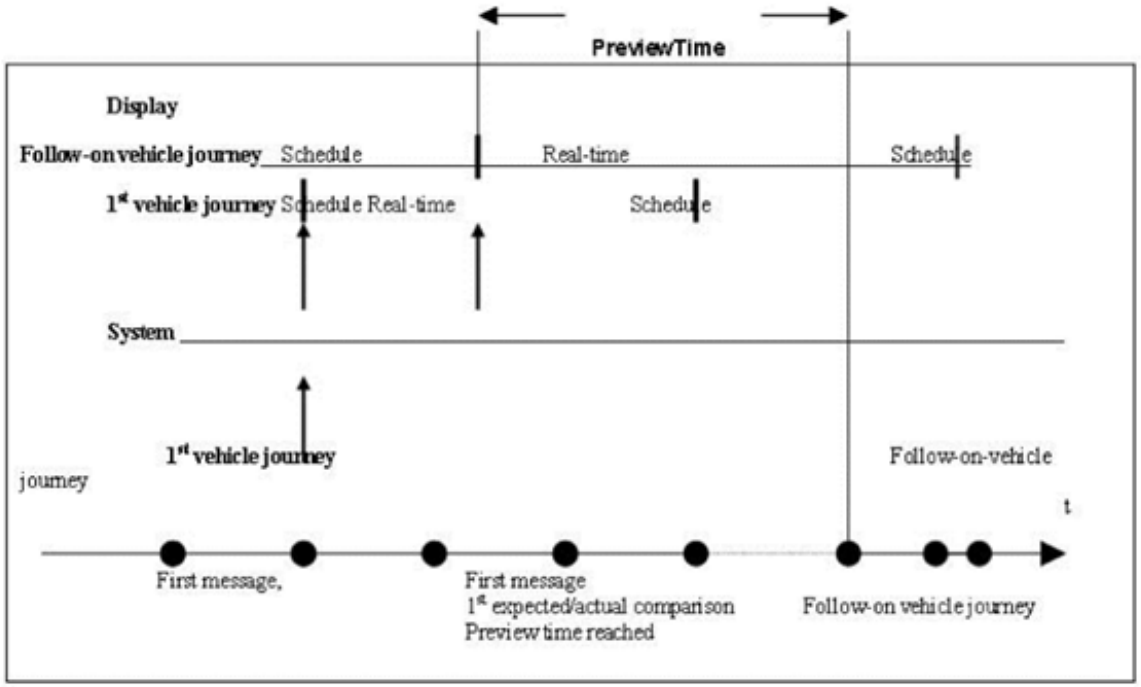


Figure 2: Preview Time and Active First Message.

7.17.3 Temporal Reporting Behaviour - Sensitivity Threshold

The SIRI Estimated Timetable service reports the current absolute arrival and departure times for a VEHICLE JOURNEY. The content of a delay message therefore does not represent information as a difference to a value in the preceding message, e.g. "+5 min", or to the static planned schedule. Every individual SIRI Estimated Timetable message can be interpreted on its own. It can be made available in the schedule information system if there is a corresponding reference to the planned data.

Stop Visits and monitored journeys may also be assigned a descriptive categorization of their relative progress; for example 'onTime', 'late', 'early': this progress status is a presentation hint to display systems that may reflect an informational policy of the operator, for example that trains less than two minutes behind the planned times should be regarded as on-time regardless.

The triggering of the communication of VEHICLE JOURNEY delays however is relative to differences in the data since the last message, and takes the form of a hysteresis function: As soon as a delay prediction for a stop exceeds the last transmitted value by a specific threshold (either negatively or positively), the Producer triggers a real-time message to the schedule information system which overwrites the old value.

It is recommended that a single sensitivity threshold be globally defined for the entire real-time data subscription, with a value in the range of one to two minutes. The alternative approach of defining the threshold values for each LINE, or even each VEHICLE JOURNEY or interval is not recommended because of the resulting complexity. Instead, the presence of a global sensitivity value should be interpreted as signifying that the schedule information system does not wish to receive any messages whose difference in content lies below the threshold value. For transmissions in the opposite direction however, the Producer may suppress messages that lie above the threshold as it is in a better position to determine whether it would be useful to transmit the data or not. For example, the messages could be suppressed when the headways are short, i.e. the VEHICLEs run in rapid succession. This allows the Producer to dynamically vary threshold values as appropriate.

A preview interval can be used to limit the data horizon for predictions. Other changes, for example additional journeys or journey cancellations, will still be included even if beyond the prediction horizon.

7.17.4 PredictionInaccurate - InCongestion

If the AVMS establishes that a VEHICLE is travelling extremely slowly or is at a complete standstill, it can activate the InCongestion and PredictionInaccurate elements in the JourneyProgressGroup structure for the relevant VEHICLE JOURNEY. At the same time, the operation of hysteresis mentioned above is suspended, i.e. the Producer suppresses further messages as long as PredictionInaccurate is active. This avoids the continuing exchange of redundant messages that would arise as the delay increased. The system avoids sending delay messages when it shall be assumed they will only increase further after another 2 min. Additional note for VDV behavior: Messages will only be sent again after PredictionInacccurate was revoked.

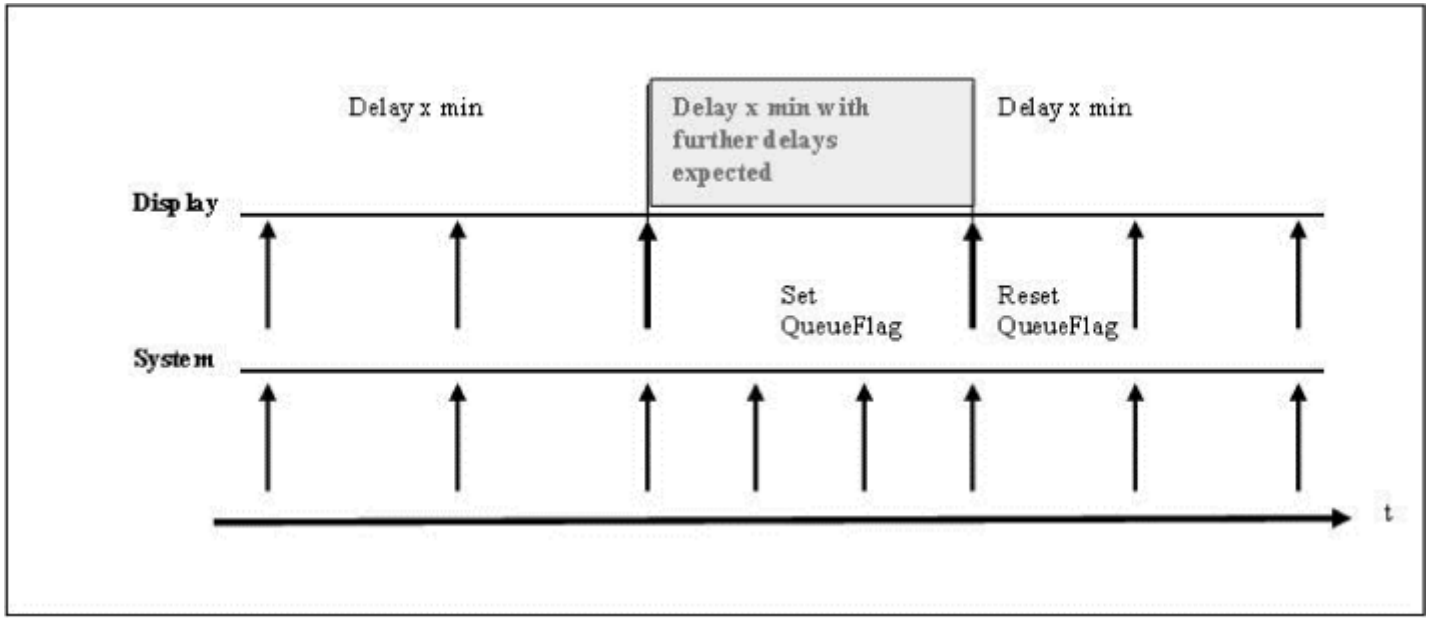


Figure 3: Reporting Behaviour in the case of PredictionInaccurate.

With the transmission of a new current prediction, the Producer deactivates the PredictionInaccurate status and signals that the change sensitivity function has been resumed.

The schedule information system can provide special behaviour to process PredictionInaccurate content. In contrast to how it handles a breakdown in communication, the schedule information system does not revert to the backup level of planned data but is able to trigger messages such as "20 min delay, more delays expected".

The PredictionInaccurate is an overall status: it may be further explained by additional elements such as InCongestion and InPanic.

7.17.5 Unexpected Termination of Monitoring

If a VEHICLE logs off from its block, or becomes unattainable via radio ('loss of contact'), the Producer shall be able to revoke a previously reported prediction. In this case, for every VEHICLE JOURNEY already reported, the Producer sends the Consumer system an EstimatedVehicleJourney message with the Monitored attribute set to false (i.e. unmonitored). This action makes the schedule information system aware of the inaccuracy pertaining

to predictions for these VEHICLE JOURNEYs, so that it can inform passengers. After a message reporting a VEHICLE JOURNEY as unmonitored, the journey has the same status as if it had not been reported at all. A more specific diagnosis of the error condition (e.g. GPRS, radio, etc.) can be included in the form of a MonitoringError code, allowing Producer systems to handle different failure modes differently.

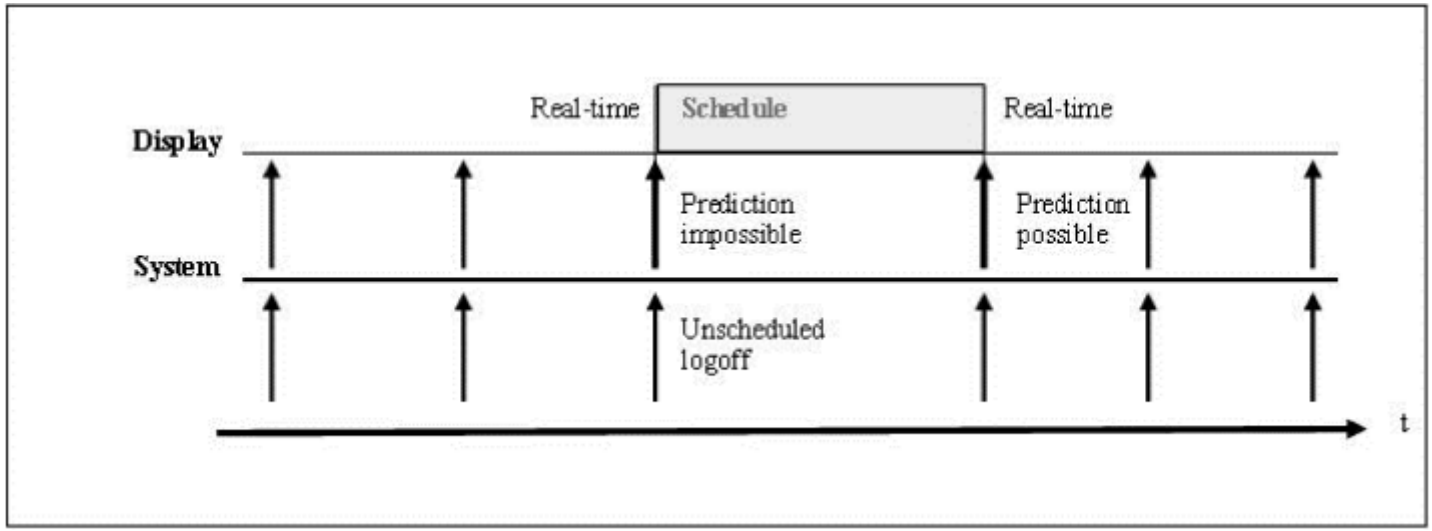


Figure 4: Loss of Contact.

7.18 Prediction Quality

(SIRI-3, 6.7)
LATER
Only ***PredicationInaccurate*** is supported for the time being.

7.19 Handling of recorded / actual data

Recording the actually observed/measured times (and possibly other information) at the arrival and departure of a vehicle is the only use case that requires the triggering of arrival and departure events as defined in this chapter. Such events in turn enable the consuming (customer information) system to extrapolate where the vehicle is located within the route of the line or journey and thus update, for example, a display with all the relevant

customer information. ***RecordedCalls*** also enable the consuming system to calculate real-time interchanges and to convert between the StopMonitoring (SM) (e.g., to trigger updates of display content) and ConnectionMonitoring (CM) services (e.g., to inform passengers in the feeder vehicle thus for connection protection).

7.19.1 Functional definition of arrival and departure

From a functional point of view, arrival and departure events as introduced above are defined as follows:

7.19.1.1 Arrival

When the first passenger is able to alight at a particular location (in relation to the given journey and stop). As a general rule, this is the point in time when the vehicle doors are (or could be) opened for the first time after the door-lock is released. It is irrelevant whether passengers actually board or alight at the stop or whether the doors were opened in the first place.

7.19.1.2 Departure

When the last passenger is able to board at a particular location (in relation to the given journey and stop). As a general rule, this is the point in time when the vehicle doors are (or could be) closed for the last time before the door-lock is engaged. It is irrelevant whether passengers actually board or alight at the stop or whether the doors were opened in the first place.

In some cases, the condition of an opening or closing door is not satisfied and thus an event as defined above cannot be recorded accurately:

- A vehicle passes through a stop without actually stopping or is only stopping briefly without opening its doors, e.g., in case the stop was optional and no one requested it. In such a case the arrival and departure event are both recorded at the same time when the vehicle passes the stopping position.
- An arrival event at a stop where alighting is prohibited is either recorded at the time when the door-lock is released for the first time, when the vehicle comes to a halt (if door-lock was never released) or when the vehicle passes the stopping position (if vehicle doesn’t actually stop).
- A departure event at a stop where boarding is prohibited is either recorded at the time when the door-lock is engaged for the last time, when the vehicle starts moving (if door-lock was never released) or when the vehicle passes the stopping position (if vehicle doesn’t actually stop).

7.19.2 Triggering events

Many transport organisations are not technically equipped to determine the arrival and departure exactly as defined above. Common implementations record the following events with varying accuracy:

- Vehicle doors are unlocked after stopping (hence aforementioned special cases not supported)
- Vehicle enters the catchment circle of a stop (with minimum inaccuracy)
- Train triggers entry signal, rail side (with minimum inaccuracy)

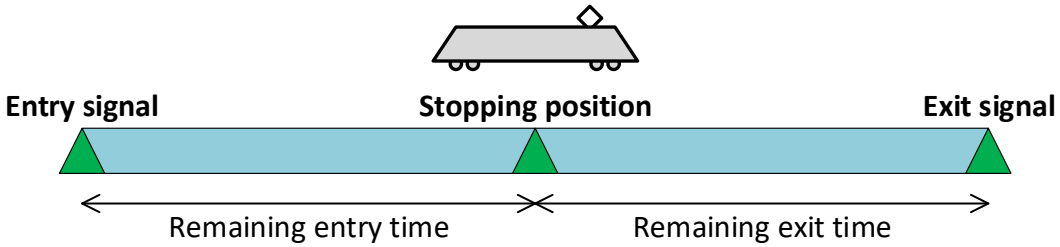
Commonly recorded departure events include:

- Vehicle doors are locked when stationary (hence aforementioned special cases not supported)
- Vehicle exits the catchment circle of a stop (with minimum inaccuracy)
- Train triggers exit signal, rail side (with minimum inaccuracy).

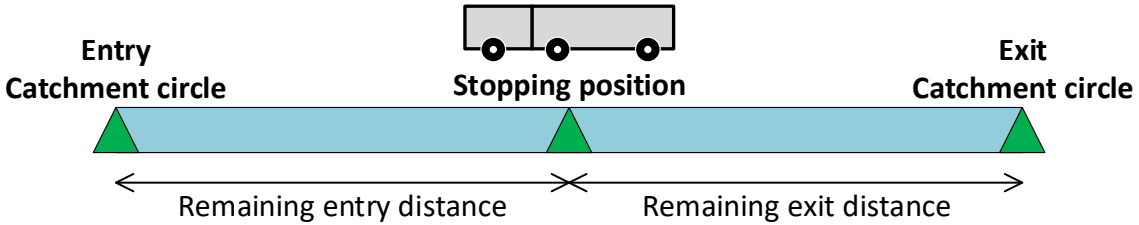
Operational situations, such as the following, complicate matters (although some can be partly alleviated by better data provision):

- Actuation of the light signalling system by means of the door locking button, i.e., if the “green” phase is missed, the door locking button must be pressed once again (rapid open/close)
- Departure of multiple vehicles at a stop-and-go platform is not in order of arrival
- Vehicle leaving the catchment circle of a stop to facilitate overtaking (and stopping outside of the catchment circle)
- Stopping before entering the catchment circle due to an occupied platform or bus bay
- Vehicle doors must be locked so that the doors actually close (e.g., when waiting at a stop in winter)

Example with rail entry and exit signal at a stop:



Example with entry and exit of a catchment circle of a stop:



The actual times (to be transmitted in ***RecordedCalls***) must be calculated as follows:

- Actual arrival time at the stopping position recorded for an arrival event =

  <Time at the entry signal> + <Remaining entry time>

- Actual departure time at the stopping position recorded for a departure event =

  <Time at the exit signal> - <Remaining exit time>

7.19.3 RecordedCall with missing actual times

If an actual time or status is omitted in a ***RecordedCall*** update, then the data producer usually wants to indicate that no such data was recorded, e.g., that ***ActualArrivalTime*** is unknown. In such a case, it is up to the consuming system how to proceed and whether it should fall back to the predicted/expected times.

7.19.4 Immutability of actual times

Actual times will, by their very nature, not be updated. A recorded actual time per stop and arrival or departure event will generally only be transmitted once (and possibly once more as part of a complete stop sequence).

7.19.5 RecordedCalls in initial loads

If, in the initial load of a new subscription, all currently active[1] journeys are transmitted, then the ITCS must also include the already transmitted ***RecordedCalls*** (for each and every ***EstimatedJourney***) of those stops which are already in the past or were recorded respectively.

7.19.6 Expected times in RecordedCall

In case of a problem (technical error, signal/connection loss etc.), where no actual time can be recorded, the data producer must nevertheless generate a ***RecordedCall***, but without the respective ***Actual*Time***, and transmit an ***EstimatedJourney*** at the latest with the next message triggering event (see chapter 7.13), including the next arrival or departure event at a subsequent stop.

The consumer, on the other hand, must be able to fall back to the last known (or in principle the most accurate) expected time. Consequently, when generating a ***RecordedCall*** out of an ***EstimatedCall***, the data producer must retain the last known prediction or ***Expected*Time*** corresponding to the recorded ***Actual*Time***.

7.19.7 Special case recording of cancelled calls

See the example message flow in a rerouting case in chapter 7.16.2.

If participants did not explicitly agree on a different mechanism, “full history” specified in chapter 7.16.2 is assumed. Thus cancelled calls must always be recorded similar to regular calls with the exception that they are flagged with ***Cancellation*** = ‘true’ and no expected or actual times are transmitted. ***RecordedCall*** updates of cancelled stops are always delivered with the next regular update and not on their own. Prerequisite for the recording of a cancelled call is that the ***AimedArrivalTime*** of the cancelled stop is exceeded.

7.19.8 Special case exceptional passthrough

If a vehicle passes through a stop without actually stopping or is only stopping briefly without opening its doors (releasing its door-locks), then the data producer will transmit an ***EstimatedCall*** with the properties ***Arrival***- and/or ***DepartureBoardingActivity*** set to ‘passThru’ independently of whether the

[1] “Active” in this context is synonymous with the journey falling inside the ***ValidityPeriod*** that was requested by the subscriber. A journey, on the other hand, falls inside the ***ValidityPeriod*** if the ***AimedDepartureTime*** or ***AimedArrivalTime*** at any stop lies between the ***Start***- and ***EndTime*** of the subscribed ***ValidityPeriod*** (regardless of whether the aimed times at a previous or subsequent stop fall outside of the ***ValidityPeriod***).
Careful: Journeys which themselves fall outside the ***ValidityPeriod*** but are linked to a journey within the ***ValidityPeriod*** by a ***JourneyRelation*** (SIRI 2.1), are also regarded as falling inside the ***ValidityPeriod***. Examples are journeys where the vehicle is split or joins with or is replaced by another vehicle

passthrough was planned, expected or exceptional (not known prior to the occurrence of the event) immediately after the event was registered or not later than with the next message triggering event.

From SIRI 2.1 onwards, ***BoardingActivity*** can also be specified in a ***RecordedCall***. It is therefore possible to explicitly "record" an exceptional passthrough with respect to the SIRI-ET interface. Thus, instead of a ***BoardingActivity*** update wrapped in an ***EstimatedCall*** as an effect to the passthrough event, only the following ***RecordedCall*** update is triggered but in this case with the required ***BoardingActivity*** information.

If arrival and departure events are triggered by entry and exit signals or catchment circles, then passthrough events are generally also supported. However, if the arrival and departure events are actually triggered by door opening/closing signals (as described in chapter 7.19.1), then a passthrough won't trigger anything, not a recording of the ***Actual*Time*** nor a ***RecordedCall*** update. The latter will only be generated with the next message triggering event (see chapter 7.13), including the next arrival or departure event at a subsequent stop. Without additional mechanisms (e.g., manually triggered by the driver), this potentially delays or even impedes the deletion of messages from display boards (which must happen immediately after departure).

7.19.9 Special case optional stop

If a vehicle does not actually stop in case ***RequestStop*** is 'true' for the corresponding ***EstimatedCall***, then the ITCS must behave as though the vehicle had stopped. As a result, not only a ***RecordedCall*** must be generated with records of the actual times (at the very least), but also the journey immediately deleted from any display boards (as usual). Furthermore, the behaviour will be in accordance with chapter 7.19.8 as well as chapter 7.19.1.

7.19.10 Special case vehicle at stop

Consider a vehicle that is calling a stop sequence … A => B => C …:

1. On the road or rails between stop A and B (vehicle has already departed at stop A but not yet arrived at stop B), ***EstimatedCalls*** are transmitted for all the upcoming stops. In particular, SIRI-ET updates are transmitted for adjustments to, say, the ***Expected*Time*** at stop B.

2. As soon as the vehicle actually arrives at stop B (trigger is, for example, a release of door-lock event), a ***RecordedCall*** update is transmitted with the corresponding ***ActualArrivalTime*** (see example message i. below).

3. Now consider that (still at stop B) the vehicle needs to wait a bit longer than scheduled or expected (common for connection protection) and that a corresponding SIRI-ET update is triggered (by the control centre) with an adjustment to the ***ExpectedDepartureTime*** for the delayed departure at stop B.

4. The SIRI-ET update corresponding to the aforementioned event in step 3. must at least consist of the ***RecordedCall*** referencing stop B (per ***StopPointRef***) as well as the adjusted ***ExpectedDepartureTime***.

5. After the vehicle actually departs at stop B (trigger is, for example, an engaging door-lock event), the aforementioned ***RecordedCall*** for stop B is updated with the ***ActualDepartureTime***.

As general rules in this context:

a. Only a single type of call (either recorded or estimated) is associated with a stop at any given time in an ***EstimatedVehicleJourney*** lifecycle or sequence of update message.

b. After the actual arrival at a stop is recorded, all potential updates referring to this stop (in addition to ***Actual*Time*** adjustments) must be transmitted in the respective ***RecordedCall***, even updates of a predictive or "estimated" nature.

The above scenario would unfold like:

i) Incremental update of RecordedCall for the actual arrival at stop B:

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1"
xsi:schemaLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <RecordedCalls>
      <RecordedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <ActualArrivalTime>2022-07-04T06:31:48Z</ActualArrivalTime>
      </RecordedCall>
```

```
      </RecordedCalls>
      <IsCompleteStopSequence>false</IsCompleteStopSequence>
     </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
   </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

ii) Incremental update of RecordedCall for the delayed departure at stop B (if applicable):

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1"
xsi:schemaLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <RecordedCalls>
      <RecordedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <ExpectedDepartureTime>2022-07-04T06:34:48Z</ExpectedDepartureTime>
      </RecordedCall>
     </RecordedCalls>
     <IsCompleteStopSequence>false</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
```

```
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

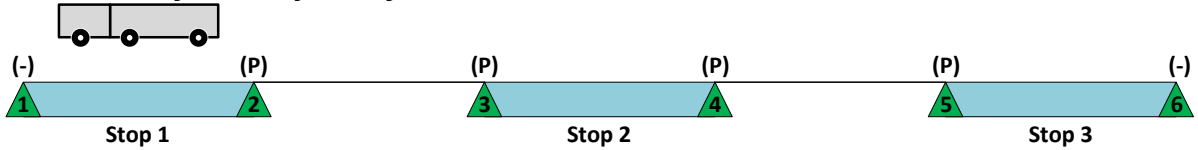
iii) Incremental update of RecordedCall for the actual departure at stop B:

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1"
xsi:schemaLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <RecordedCalls>
      <RecordedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <ActualDepartureTime>2022-07-04T06:35:00Z</ActualDepartureTime>
      </RecordedCall>
     </RecordedCalls>
     <IsCompleteStopSequence>false</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

7.19.11 Illustration of the default case

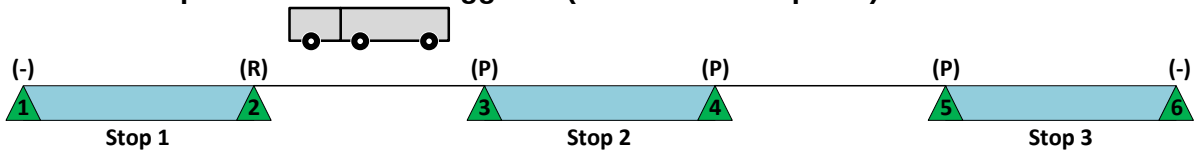
All events are triggered properly and result in an immediate ***EstimatedVehicleJourney*** update with the respective actual data wrapped in a ***RecordedCall***.

Initial delivery before journey is active:



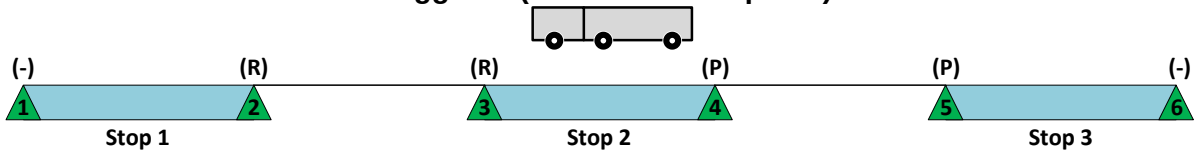
P = predicted/expected as per ***EstimatedCall***

State after departure event 2 is triggered (*RecordedCall* update):



P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

State after arrival event 3 is triggered (*RecordedCall* update):



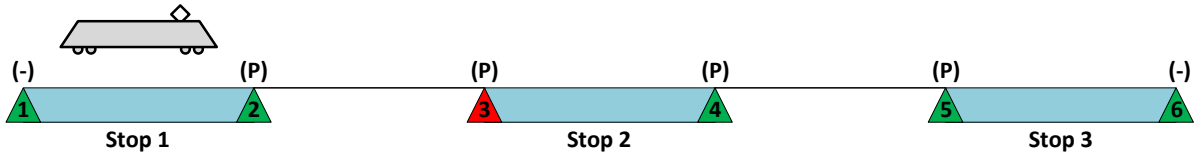
P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

7.19.12 Event not triggered

This example illustrates the behaviour specified in chapter 7.19.6.

Arrival event 3 cannot be triggered. The ***RecordedCall*** update corresponding to said event is transmitted only after the next message triggering event, which in this example is departure event 4. However, since event 3 was not triggered properly, i.e., no actual time was recorded, the respective ***RecordedCall*** holds an ***ExpectedArrivalTime*** (the latest prediction) instead of an ***ActualArrivalTime***.

Initial delivery before journey is active:

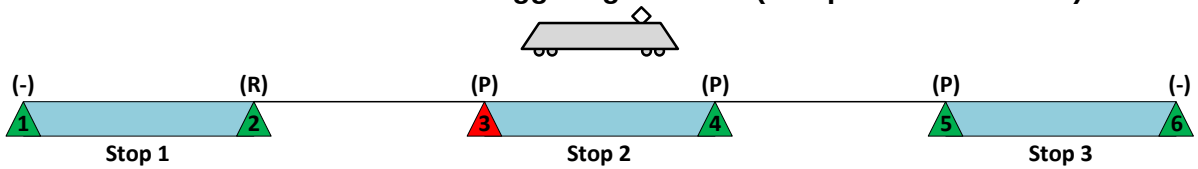


P = predicted/expected as per ***EstimatedCall***

State after departure event 2 is triggered (*RecordedCall* update):

P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

State after arrival 3 occurs without triggering an event (no update transmitted):



P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

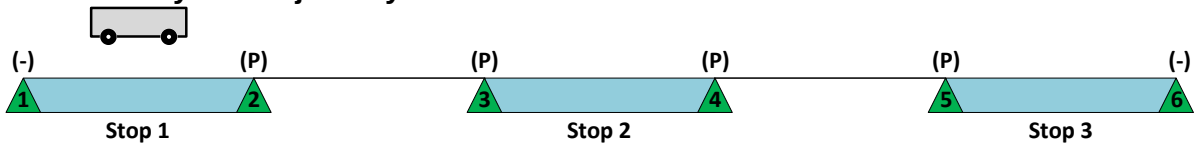
State after departure event 4 is triggered (*RecordedCall* update for events 3 and 4):

P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***
R* = predicted/expected as per ***RecordedCall***

7.19.13 Loss of connection

Arrival event 3 is properly triggered but the corresponding ***RecordedCall*** update cannot be transmitted immediately due to a communication dead spot. Only after the next message triggering event (in this example departure event 4) is the ***RecordedCall*** update transmitted with a delay (merged with the update corresponding to event 4). Both ***RecordedCalls*** hold a proper ***Actual*Time*** (regardless of the delay in case of event 3).

Initial delivery before journey is active:

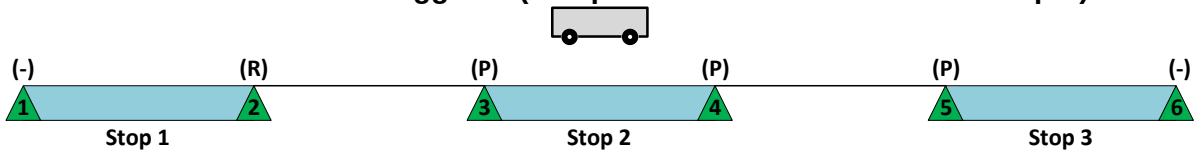


P = predicted/expected as per ***EstimatedCall***

State after departure event 2 is triggered (*RecordedCall* update):

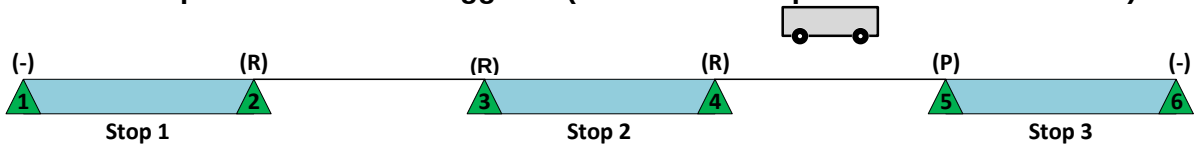
P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

State after arrival event 3 is triggered (no update transmitted due to dead spot):



P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

State after departure event 4 is triggered (*RecordedCall* update for events 3 and 4):

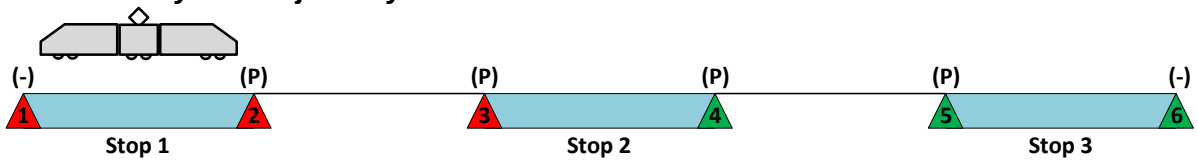


P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

7.19.14 Events not triggered and prediction unknown

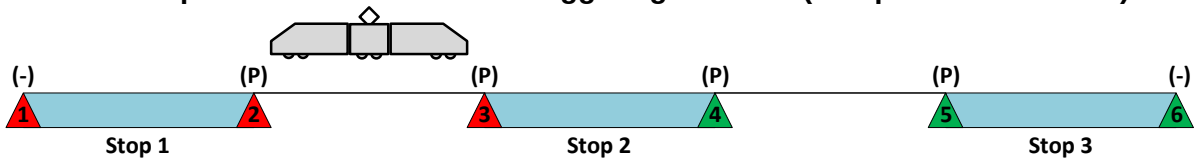
Events 1 to 3 are not triggered. All other events, signals and the connection to the ITCS are working. Predictions can be transmitted if hysteresis occurs or if attributes change. From event 4 onwards, all events are triggered. However, the inaccuracy for the predictions corresponding to events 1 to 3 exceeds a threshold and thus equate “unknown”.

Initial delivery before journey is active:



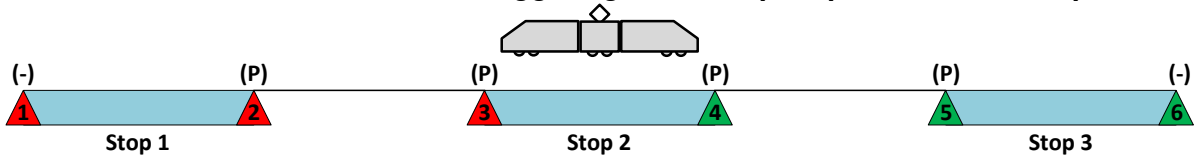
P = predicted/expected as per ***EstimatedCall***

State after departure 2 occurs without triggering an event (no update transmitted):



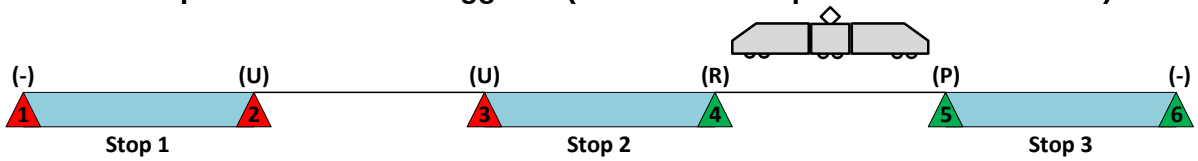
P = predicted/expected as per ***EstimatedCall***

State after arrival 3 occurs without triggering an event (no update transmitted):



P = predicted/expected as per ***EstimatedCall***

State after departure event 4 is triggered (*RecordedCall* update for events 2 to 4):



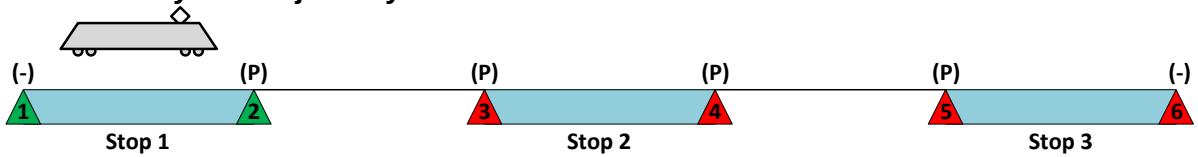
P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***
U = ***RecordedCall*** with unknown/missing expected as well as actual time

If the predictions are reliable after all (inaccuracy within threshold), the ***RecordedCalls*** corresponding to events 2 and 3 will hold the respective ***Expected*Times*** (i.e., latest predictions) instead (analogous to example 7.19.12) .

7.19.15 Events not triggered after a certain point

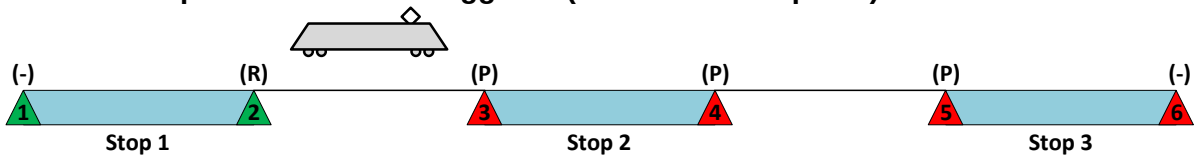
Events from 3 onwards are not triggered. As a result, only predictions within ***ExpectedCalls*** are transmitted (no ***RecordedCall*** updates, except for departure 2).

Initial delivery before journey is active:



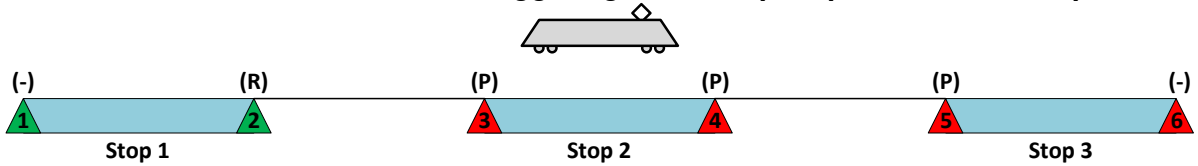
P = predicted/expected as per ***EstimatedCall***

State after departure event 2 is triggered (*RecordedCall* update):



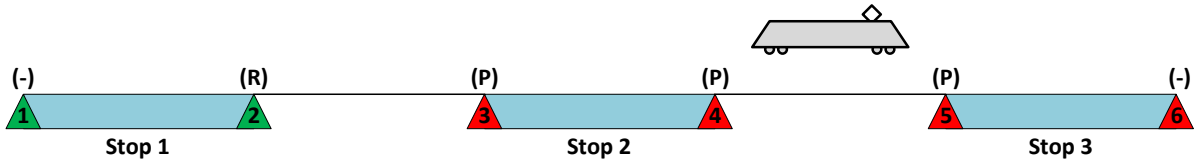
P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

State after arrival 3 occurs without triggering an event (no update transmitted):



P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

State after departure 4 occurs without triggering an event (no update transmitted):

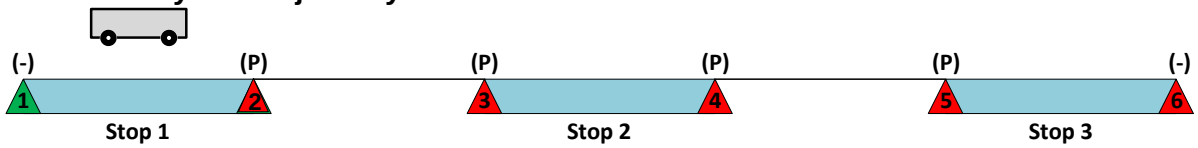


P = predicted/expected as per ***EstimatedCall***
R = recorded/actual as per ***RecordedCall***

7.19.16 Extra journey without real-time feed

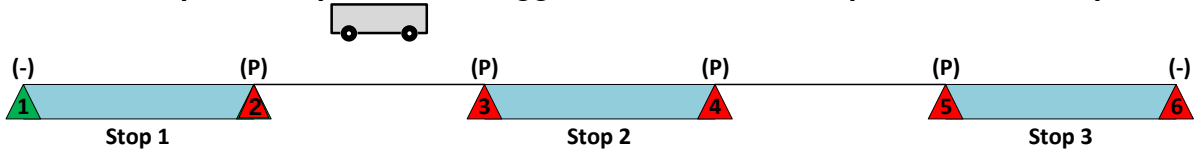
A journey is transmitted with both ***ExtraJourney*** as well as ***Monitored*** set to 'false'. No real-time or planned/periodic timetable data is available. The initial delivery of the journey (which is the only one in its lifetime) might include predictions. However, no ***RecordedCall*** updates are transmitted (or other messages for that matter).

Initial delivery before journey is active:



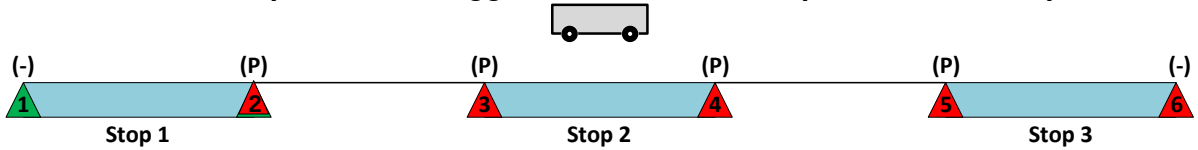
P = predicted/expected as per ***EstimatedCall***

State after departure 2 (no event is triggered nor is there an update transmitted):



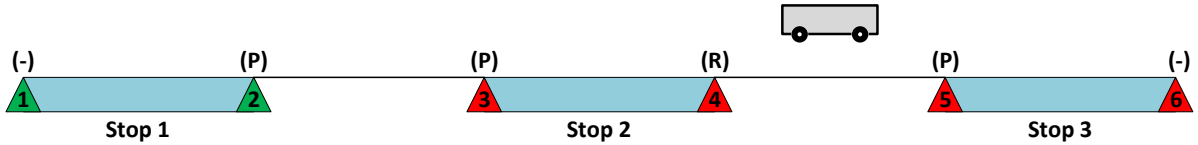
P = predicted/expected as per ***EstimatedCall***

State after arrival 3 (no event is triggered nor is there an update transmitted):



P = predicted/expected as per ***EstimatedCall***

State after departure 4 (no event is triggered nor is there an update transmitted):



P = predicted/expected as per ***EstimatedCall***

10

10 Case Collection

Table 19: Overview of the use cases relevant in public transport CH

Use Case	Mandatory/Optional	Service
10.2 **Establishing the baseline with an initial load**	Mandatory	ET
10.3 **Delay**		
10.4 **Platform update**		
10.5 **Extra journey**		
10.6 **Journey cancellation**		
10.7 **Partial cancellation**		
10.8 **Extra stop**		
10.9 **Exceptional passthru**		
10.10 **Rerouting**		
10.11 **Recording of calls**		
10.12 **Vehicle waiting at stop**		
10.13 **Boarding activity update** 10.14 **No alighting update** 10.15 **No boarding update**		
10.16 **Replacement transport and journey relations**	Optional	
10.17 **Train formation update**		
10.18 **Occupancy update**		
10.19 **Line timetable over production timetable**		PT
10.20 **Situation affecting line (TODO)**	Mandatory	SX
10.21 **Situation affecting stop place (TODO)**	Optional	
10.22 **Situation affecting elevator and impacting accessibility (TODO)**		

10.1 Use of Order and VisitNumber in this chapter

Order as the optional element of a call is only used in the following use case examples to help the reader identify the stops in the illustrations (instead of having to check the ***StopPointRef*** again and again in the baseline example). ***Order*** or ***VisitNumber*** are generally not useful from a programmatic perspective, but only from a human readers perspective. In fact, after significant despatching alterations like extra calls or changes of the route, ***Order*** might introduce inconsistencies if it is used to identify a stop or to determine the actual ordering of the call sequence.

It is recommended to always use ***StopPointRef*** in combination with ***Aimed*Time*** for the matching process and ***Aimed*Time*** for the ordering process.

10.2 Establishing the baseline with an initial load

10.2.1 Business

A requirement of the incremental update function in the context of the EstimatedTimetable is that the data producer must establish a proper initial state or so called “baseline” for each and every journey after a new subscription is established with the client / consumer or an error (for example connection loss) has occurred. A baseline is all the more imperative if no ProductionTimetable service is implemented.

A baseline of a journey is synonymous to an ***EstimatedVehicleJourney*** with ***IsCompleteStopSequence*** equal to ‘true’ delivered in a so called «initial load». A complete stop sequence, in turn, means that not only each and every called stop of the journey pattern (valid at time of processing) must be included but also each and every mandatory element/structure as well as all optional elements/structures where the value/content is different from the default value as specified in the XSD schema.

10.2.2 Example

The XML example describes a complete stop sequence to establish a proper baseline as part of the “initial load”. Such an initial load will be delivered by the producer for each and every relevant journey after the consumer communicates success within the SubscriptionResponse. Most of the ET examples in the subsequent chapters will be based on this journey:

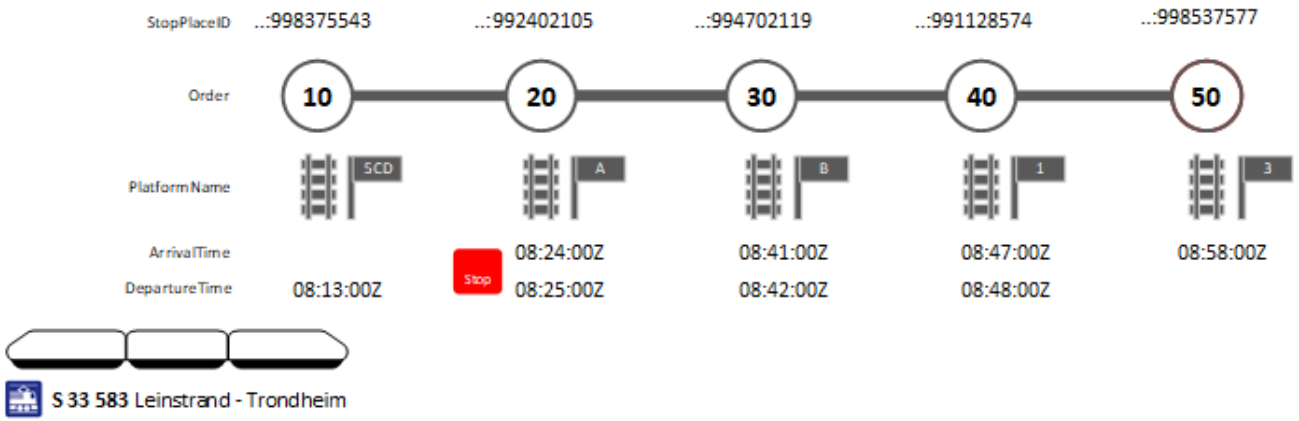


Figure 5 - Illustration of the baseline example with the most important information from the perspective of an arbitrary passenger information system, e.g., stop sequence, timings, platforms and whether a stop must be requested explicitly (depicted by the red stop button).

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>ski-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>ch:1:Direction:H</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
```

```
      </FramedVehicleJourneyRef>
      <VehicleMode>rail</VehicleMode>
      <PublishedLineName>S33</PublishedLineName>
      <OriginName>Leinstrand</OriginName>
      <DestinationName>Trondheim</DestinationName>
      <OperatorRef>ch:1:Organisation:231</OperatorRef>
      <ProductCategoryRef>ch:1:TypeOfProductCategory:S</ProductCategoryRef>
      <VehicleFeatureRef>highFloor</VehicleFeatureRef>
      <VehicleFeatureRef>foldableRampAvailable</VehicleFeatureRef>
      <VehicleFeatureRef>wheelchairAssistanceAvailableIfBooked</VehicleFeatureRef>
      <VehicleFeatureRef>noWheelchairAcccessToilet</VehicleFeatureRef>
      <Monitored>true</Monitored>
      <!-- VehicleRef is only included for the sake of comparison with TrainNumberRef
           since a train (formation), possibly composed of multiple waggons, doesn't
           have a single stable identifier but one for each waggon (european vehicle
           number or also called UIC waggon number). For busses, trams and vehicle
           types of similar complexity, VehicleRef is generally used to communicate its
           identifier that is constant throughout its service life.
           A use case of VehicleRef is that of a consuming system mapping a given
           Bluetooth low energy Beacon identifier to a journey (or timetable entry). -->
      <VehicleRef>ch:1:Vehicle:231:1029</VehicleRef>
      <TrainNumber>
       <TrainNumberRef>ch:1:TrainNumber:231:583</TrainNumberRef>
      </TrainNumber>
      <EstimatedCalls>
       <EstimatedCall>
        <StopPointRef>ch:1:StopPlace:998375543</StopPointRef>
        <Order>10</Order>
        <!-- The origin of a journey, i.e., first call in the stop sequence, is
             identified by the absence of an arrival or AimedArrivalTime to be precise. -->
        <Occupancy>manySeatsAvailable</Occupancy>
        <AimedDepartureTime>2022-01-11T08:13:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:13:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DeparturePlatformName>5CD</DeparturePlatformName>
        <!-- Optional elements like *BoardingActivity can technically be omitted if the
             value is equal to the default according to the schema.
             Conversely, optional elements must always be included if different from
```

```
            the default values. -->
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:71</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76007252:71</ExpectedQuayRef>
        </DepartureStopAssignment>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:992402105</StopPointRef>
        <!-- The use of Order is not recommended and Order/VisitNumber will be ignored at import.
             See chapter 10.1. Order is only used in the examples for the sake of readability (since the common
             Request policy capability VisitNumberisOrder has default value 'false', Order is favored here). -->
        <Order>20</Order>
        <RequestStop>true</RequestStop>
        <AimedArrivalTime>2022-01-11T08:24:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2022-01-11T08:24:00Z</ExpectedArrivalTime>
        <ArrivalStatus>onTime</ArrivalStatus>
        <ArrivalPlatformName>A</ArrivalPlatformName>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <ArrivalStopAssignment>
         <AimedQuayRef>ch:1:Quay:76011270:622</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76011270:622</ExpectedQuayRef>
        </ArrivalStopAssignment>
        <AimedDepartureTime>2022-01-11T08:25:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:25:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DeparturePlatformName>A</DeparturePlatformName>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76011270:622</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76011270:622</ExpectedQuayRef>
        </DepartureStopAssignment>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:StopPlace:994702119</StopPointRef>
        <Order>30</Order>
        <CallNote>Side of alighting: to the right</CallNote>
        <AimedArrivalTime>2022-01-11T08:41:00Z</AimedArrivalTime>
```

```
        <ExpectedArrivalTime>2022-01-11T08:41:00Z</ExpectedArrivalTime>
        <ArrivalStatus>onTime</ArrivalStatus>
        <ArrivalPlatformName>B</ArrivalPlatformName>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <ArrivalStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007112:31</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76007112:31</ExpectedQuayRef>
        </ArrivalStopAssignment>
        <AimedDepartureTime>2022-01-11T08:42:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:42:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DeparturePlatformName>B</DeparturePlatformName>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007112:31</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76007112:31</ExpectedQuayRef>
        </DepartureStopAssignment>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:StopPlace:991128574</StopPointRef>
        <Order>40</Order>
        <AimedArrivalTime>2022-01-11T08:47:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2022-01-11T08:47:00Z</ExpectedArrivalTime>
        <ArrivalStatus>onTime</ArrivalStatus>
        <ArrivalPlatformName>1</ArrivalPlatformName>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <ArrivalStopAssignment>
         <AimedQuayRef>ch:1:Quay:76011221:2939</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76011221:2939</ExpectedQuayRef>
        </ArrivalStopAssignment>
        <AimedDepartureTime>2022-01-11T08:48:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:48:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DeparturePlatformName>1</DeparturePlatformName>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76011221:2939</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:76011221:2939</ExpectedQuayRef>
```

```
       </DepartureStopAssignment>
      </EstimatedCall>
      <EstimatedCall>
       <StopPointRef>ch:1:StopPlace:998537577</StopPointRef>
       <Order>50</Order>
       <AimedArrivalTime>2022-01-11T08:58:00Z</AimedArrivalTime>
       <ExpectedArrivalTime>2022-01-11T08:58:00Z</ExpectedArrivalTime>
       <ArrivalStatus>onTime</ArrivalStatus>
       <ArrivalPlatformName>3</ArrivalPlatformName>
       <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
       <ArrivalStopAssignment>
        <AimedQuayRef>ch:1:Quay:76011262:8</AimedQuayRef>
        <ExpectedQuayRef>ch:1:Quay:76011262:8</ExpectedQuayRef>
       </ArrivalStopAssignment>
       <!-- The destination of a journey, i.e., last call in the stop sequence, is
            identified by the absence of a departure or AimedDepartureTime to be
            precise. -->
      </EstimatedCall>
     </EstimatedCalls>
     <!-- A baseline message is a complete stop sequence by nature. -->
     <IsCompleteStopSequence>true</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.3 Delay

10.3.1 Business

Incremental updates support: ***IsCompleteStopSequence*** = ‘false’ **allowed**

Updates of the expected / estimated arrival and departure times with respect to the scheduled / aimed times communicated in the yearly or production timetable. By far the most common use case, e.g., bus is delayed due to heavy traffic.

See also chapter 7.17. Smaller delays might affect only individual calls but larger delays usually require the application of an update rule as described in the last section of the referenced chapter above. A stop sequence with arrival or departure times not in ascending order potentially leads to problems with routing or information display of interchanges.

10.3.2 Example

10.3.2.1 Smaller delays

ET example of a delay at two subsequent stops, delivered as an incremental update:

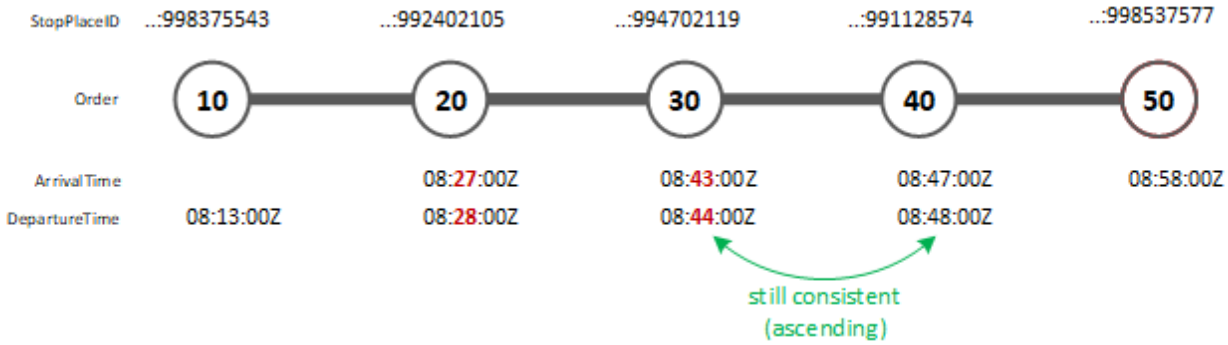


Figure 6 - Illustration of the delay example with timings representing the predictions

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>ski-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
```

```
      <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
      <LineRef>ch:1:Line:231:S23</LineRef>
      <DirectionRef>2</DirectionRef>
      <FramedVehicleJourneyRef>
       <DataFrameRef>2022-01-11</DataFrameRef>
       <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
      </FramedVehicleJourneyRef>
      <EstimatedCalls>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:992402105</StopPointRef>
        <Order>20</Order>
        <!-- Delay of 3 minutes is expected, which triggers an update of the
             Expected*Time and optionally of the *Status.
             Note that AimedArrivalTime must always be transmitted, even in an incremental update (as seen here). -->
        <AimedArrivalTime>2022-01-11T08:24:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2022-01-11T08:27:00Z</ExpectedArrivalTime>
        <ArrivalStatus>delayed</ArrivalStatus>
        <!-- Delay of 3 minutes is expected -->
        <ExpectedDepartureTime>2022-01-11T08:28:00Z</ExpectedDepartureTime>
        <DepartureStatus>delayed</DepartureStatus>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:994702119</StopPointRef>
        <Order>30</Order>
        <!-- Delay of 2 minutes is expected -->
        <ExpectedArrivalTime>2022-01-11T08:43:00Z</ExpectedArrivalTime>
        <ArrivalStatus>delayed</ArrivalStatus>
        <!-- Delay of 2 minutes is expected -->
        <ExpectedDepartureTime>2022-01-11T08:44:00Z</ExpectedDepartureTime>
        <DepartureStatus>delayed</DepartureStatus>
       </EstimatedCall>
       <!-- Being an incremental update, other calls of the remaining (not yet
            recorded) part of the journey can be omitted if, and only if, the
            Expected*Times at subsequent stops are strictly monotonically increasing
            and without inconsistencies. -->
      </EstimatedCalls>
      <!-- An incremental update is not a complete stop sequence by design.
           A complete stop sequence would require all remaining (future) EstimatedCalls and
```

```
          all already observed (past) RecordedCalls. -->
     <IsCompleteStopSequence>false</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.3.2.2 Large delay and update rule

ET example of a larger delay for the same stops as in the example above. In this case the delay is large enough so that chronological inconsistencies would emerge without the application of an update or interpolation rule to all the currently remaining ***EstimatedCalls*** of the journey:

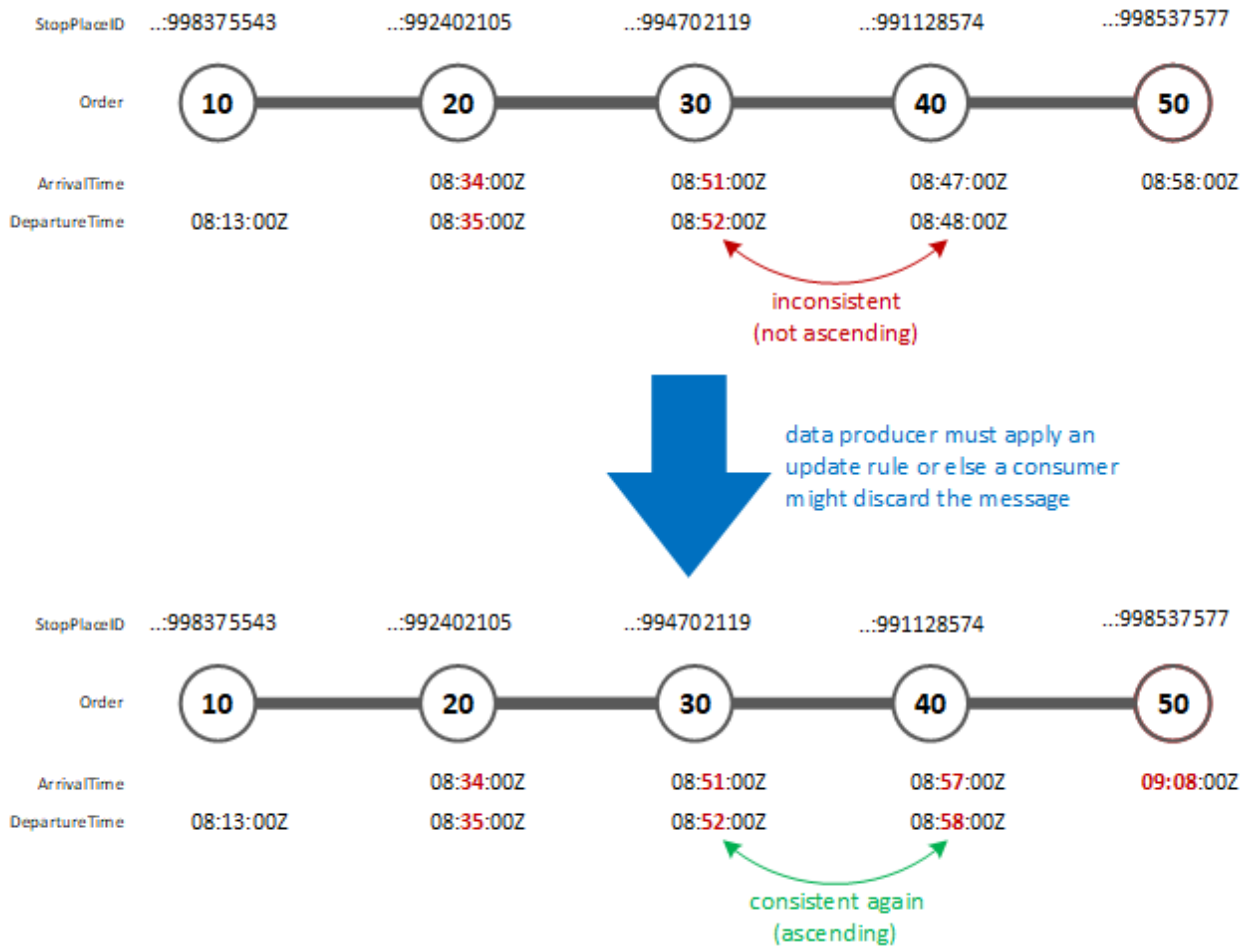


Figure 7 - Illustration of the delay example with an update rule applied to the expected times

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
```

```
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <EstimatedCalls>
      <EstimatedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:992402105</StopPointRef>
       <Order>20</Order>
       <!-- Larger delay of 10 minutes is expected, which triggers an update of the
            Expected*Time and optionally of the *Status.
            Note that AimedArrivalTime must always be transmitted, even in an incremental update (as seen here). -->
       <AimedArrivalTime>2022-01-11T08:24:00Z</AimedArrivalTime>
       <ExpectedArrivalTime>2022-01-11T08:34:00Z</ExpectedArrivalTime>
       <ArrivalStatus>delayed</ArrivalStatus>
       <!-- Delay of 10 minutes is expected. -->
       <ExpectedDepartureTime>2022-01-11T08:35:00Z</ExpectedDepartureTime>
       <DepartureStatus>delayed</DepartureStatus>
      </EstimatedCall>
      <EstimatedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:994702119</StopPointRef>
       <Order>30</Order>
       <!-- Delay of 10 minutes is expected. -->
       <ExpectedArrivalTime>2022-01-11T08:51:00Z</ExpectedArrivalTime>
       <ArrivalStatus>delayed</ArrivalStatus>
       <!-- Delay of 10 minutes is expected. -->
```

```
        <ExpectedDepartureTime>2022-01-11T08:52:00Z</ExpectedDepartureTime>
        <DepartureStatus>delayed</DepartureStatus>
      </EstimatedCall>
      <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
        <Order>40</Order>
        <!-- Delay of 10 minutes is expected.
             Note that AimedArrivalTime(Order=40) - AimedDepartureTime(Order=30) = 5
             minutes, which is smaller than the expected delay of 10 minutes. The delay
             is therefore large enough so that inconsistencies would emerge without the
             application of an update or interpolation rule to all remaining (future)
             EstimatedCalls of the journey (as per currently observed state of the
             vehicle. -->
        <ExpectedArrivalTime>2022-01-11T08:57:00Z</ExpectedArrivalTime>
        <ArrivalStatus>delayed</ArrivalStatus>
        <!-- Delay of 10 minutes is expected. -->
        <ExpectedDepartureTime>2022-01-11T08:58:00Z</ExpectedDepartureTime>
        <DepartureStatus>delayed</DepartureStatus>
      </EstimatedCall>
      <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:998537577</StopPointRef>
        <Order>50</Order>
        <!-- Delay of 10 minutes is expected (in this case for the destination, i.e.,
             last call in stop sequence).
             Note that the same logic applies as for the preceding call, i.e., update
             or interpolation rule is applied.
             It is recommended that, in case of delays, the data producer always
             includes all remaining (future) EstimatedCalls with updated
             Expected*Times. -->
        <ExpectedArrivalTime>2022-01-11T09:08:00Z</ExpectedArrivalTime>
        <ArrivalStatus>delayed</ArrivalStatus>
      </EstimatedCall>
    </EstimatedCalls>
    <!-- An incremental update is not a complete stop sequence by design.
         Although all the remaining (future) EstimatedCalls are included in this
         example, omission of the already observed (past) RecordedCalls results in
         IsCompleteStopSequence = 'false'. -->
    <IsCompleteStopSequence>false</IsCompleteStopSequence>
```

```
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.4 Platform update

10.4.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = ‘true’ required

Either simple textual updates of the ****PlatformName*** (publicly known designation of estimated or actual quay depending on the typ of call) or more detailed information with aimed and expected quay references in ****StopAssignment*** which point to the entities in the timetable or stop place master data (where the official designations, coordinates etc. can be found). A typical example: the originally planned platform (as per yearly timetable) is platform «3», however, due to dispatching alterations two hours in advance the train will arrive/depart on platform «4». An event in the rail control system triggers the delivery of an ET journey update with ****PlatformName*** set to the new value in the respective call.

If arrival and departure are sharing the same quays in ****StopAssignment***, the assignment for the arrival may be omitted. The ****StopAssignment*** structure contains quay references in the format of technical identifier.

Depending on the stop place model, this ID does not have to coincide with the platform numbers which are known to the public. Requirement for using ****StopAssignment*** is that producer and consumer have agreed on a common stop place (and therefore quay) model. In Switzerland the technical identifier of such quays must always enable a consuming system to uniquely match the corresponding entities in the DiDok master data.

10.4.2 Example

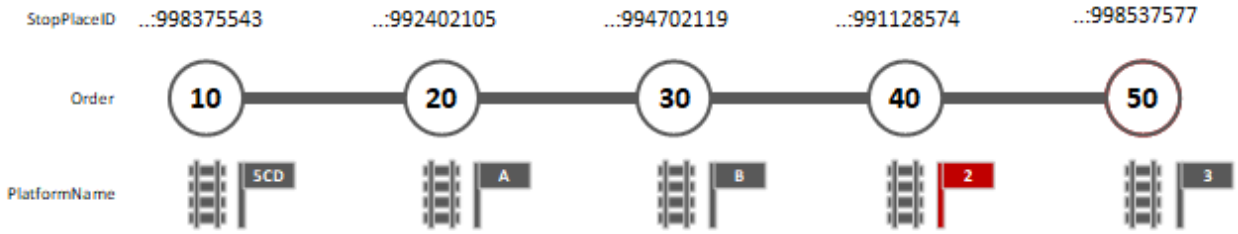


Figure 8 - Illustration of the example: platform change at the fourth stop

Careful: even though an update message as a result of a platform change must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <EstimatedCalls>
```

```
      <EstimatedCall>
       <!-- A StopPointRef (acting as primary key of the call) can point to various
            types of objects or rather levels within a "stop" data model.
            Commonly implemented levels of abstraction are either one of the
            following:
             (1) StopPointRef points to a ScheduledStopPoint according to Transmodel
                 definition (which implicitly points to a StopPlace and/or Quay
                 per NeTEx StopAssignment).
             (2) StopPointRef points to a StopPlace according to Transmodel
                 definition.
             (3) StopPointRef points to a Quay according to Transmodel definition
                 (be it an actual platform edge or some other type of quay, e.g. stop
                  area).
            Option (2) is often implemented because it offers the advantage that
            short-term platform changes are simple updates of *PlatformName and/or
            ExpectedQuayRef, whereas option (1) and (3) result in an update of the
            whole stop sequence ("Rerouting" use case) with the cancellation and
            addition of calls. -->
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <Order>40</Order>
       <!-- *PlatformName carries the descriptive code (visible on signs or displays
            on the physical object) known to passengers, whereas the *StopAssignment,
            or *QuayRefs respectively, are the technical identifier that a consuming
            system uses as query parameter when requesting information (e.g. name,
            coordinates etc.) about the StopPlace or Quay from the master data
            system. -->
       <ArrivalPlatformName>2</ArrivalPlatformName>
       <ArrivalStopAssignment>
         <ExpectedQuayRef>ch:1:Quay:76011221:519</ExpectedQuayRef>
       </ArrivalStopAssignment>
       <DeparturePlatformName>2</DeparturePlatformName>
       <DepartureStopAssignment>
        <ExpectedQuayRef>ch:1:Quay:76011221:519</ExpectedQuayRef>
       </DepartureStopAssignment>
      </EstimatedCall>
      <!-- Rest of the calls are omitted for the sake of simplicity.
           Keep in mind that an update message after a significant dispatching
           alteration must always be a complete stop sequence. -->
```

```
      </EstimatedCalls>
      <!-- A platform update must always be a complete stop sequence. -->
      <IsCompleteStopSequence>true</IsCompleteStopSequence>
     </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
   </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.5 Extra journey

SIRI part 3 (CEN/TS 15531-3), chapter 6.8.3 states:
There are two situations in which additional VEHICLE JOURNEYs shall be communicated by the data sender:

- ***Production Stage Addition***
  The data sender and the schedule information system both know the same version of the timetable; however, in the control centre, an additional VEHICLE JOURNEY is added before transmission of the planned schedules in the SIRI Production Timetable service. In this case, the additional VEHICLE JOURNEY is communicated in a DatedTimetableVersionFrame as a DatedVehicle-Journey marked with the ExtraJourney property.
- ***Real-time Addition***
  After completion of the SIRI Production Timetable transmission, an additional VEHICLE JOURNEY is added by the dispatcher. In this case, the additional VEHICLE JOURNEY is communicated as an EstimatedVehicleJourney marked with the ExtraJourney property.

10.5.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = ‘true’ required

A typical use case is a replacement journey in case of a situation (accident) or, a bit more specific, trains between Geneva and Geneva International Motor Show for reinforcement. An extra journey is either running on a known route - e.g. running in parallel to a scheduled journey because of high passenger demand - or it is running on a different/new route.

From the perspective of a consuming system, all journeys

a. received over PT which cannot be matched with the yearly timetable or
b. received over ET which cannot be matched with the (daily) production timetable (or yearly timetable if PT is not implemented)

must be regarded as additional journeys independently of whether the flag ***ExtraJourney*** is set to ‘true’. However, data producers are obligated to flag extra journeys and assign a unique journey ID if necessary (unique within the operating day and organisation). If ‘false’, i.e., equal to the default value according to the schema, the flag is omitted

A producer might still want a journey to be displayed as an extra journey even though a consumer is able to match it in the production or yearly timetable since it was planned days or months in advance but is unexpected from the point of view oft he passengers. Conversely, if a journey is processed with ***ExtraJourney*** ‘true’ and a matching journey is found in the production or yearly timetable, the update (given that ***IsCompleteStopSequence*** is ‘true’) will override any existing information and no error must be thrown.

10.5.2 Example

Careful: even though an update message as a result of an extra journey must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="UTF-8"
<!-- (C) Copyright 2005-2018 CEN SIRI SBB -->
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
  <ServiceDelivery>
    <ResponseTimestamp>2018-04-11T04:24:56Z</ResponseTimestamp>
    <ProducerRef>ski-out-et_prod</ProducerRef>
    <RequestMessageRef>a08d40cb-d438-47c8-af73-22e9c8080a38</RequestMessageRef>
    <EstimatedTimetableDelivery version="2.0">
      <ResponseTimestamp>2018-04-11T04:34:56Z</ResponseTimestamp>
      <SubscriptionRef>2</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2018-04-11T04:34:56Z</RecordedAtTime>
        <EstimatedVehicleJourney>
          <LineRef>ch:1:Line:11:71410</LineRef>
          <DirectionRef>ch:1:Direction:H</DirectionRef>
          <EstimatedVehicleJourneyCode>ch:1:ServiceJourney:11:71410-001</EstimatedVehicleJourneyCode>
          <!-- ExtraJourney set to ‘true’ to indicate an additional journey with respect to the production
               timetable (or yearly timetable if PT is not implemented). The producer signals to the consumer
               that this journey must be displayed as an extra journey (in any case). -->
          <ExtraJourney>true</ExtraJourney>
          <VehicleMode>rail</VehicleMode>
```

```
          <PublishedLineName>IC1</PublishedLineName>
          <ProductCategoryRef>ch:1:TypeOfProductCategory:IC</ProductCategoryRef>
          <VehicleFeatureRef>lowFloor</VehicleFeatureRef>
           <VehicleFeatureRef>foldableRampAvailable</VehicleFeatureRef>
           <VehicleFeatureRef>wheelchairAssistanceByDriverAvailable</VehicleFeatureRef>
          <Monitored>true</Monitored>
          <EstimatedCalls>
            <!-- EstimatedCalls are omitted for the sake of simplicity. -->
          </EstimatedCalls>
          <!-- The initial message of an extra journey must always be a complete stop sequence. -->
          <IsCompleteStopSequence>true</IsCompleteStopSequence>
        </EstimatedVehicleJourney>
      </EstimatedJourneyVersionFrame>
    </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.6 Journey cancellation

SIRI part 3 (CEN/TS 15531-3), chapter 6.8.2 states:
There are two situations in which cancelled VEHICLE JOURNEYs need to be communicated by the data producer:

- ***Production Stage Cancellation***
  The data producer and the consumer system both know the same version of the timetable; however, in the control centre a VEHICLE JOURNEY is cancelled before transmission of the planned schedules in the SIRI Production Timetable service. In this case, the VEHICLE JOURNEY to be cancelled in the planned schedule is communicated in a DatedTimetableVersionFrame, as a Dated-VehicleJourney marked as a Cancellation.
- ***Real-time Cancellation***
  After completion of the SIRI Production Timetable transmission, a VEHICLE JOURNEY is cancelled in the control centre. In this case, the VEHICLE JOURNEY to be cancelled is communicated as an EstimatedVehicleJourney marked as a Cancellation.

10.6.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = ‘true’ required

From the perspective of a consuming system, any journey

a. of the yearly timetable for which no matching ***DatedVehicleJourney*** is received (in any line timetable) over PT or

b. of the (daily) production (or yearly timetable if PT is not implemented) for which no matching ***EstimatedVehicleJourney*** is received (in the initial load after the subscription is established) over ET

must be regarded as a cancelled journey. However, data producers are obligated to explicitly deliver cancelled journeys and flag them accordingly. If 'false', i.e., equal to the default value according to the schema, the flag is omitted.

A ***Dated***- or ***EstimatedVehicleJourney*** with ***Cancellation*** = 'true' (on journey-level) must always be transmitted with the complete history of its stop sequence, i.e., with all the recorded and estimated stops (including cancelled and extra stops) up until the cancellation of the journey (as indicated by ***IsCompleteStopSequence***).

A producer and consumer (e.g. journey planner) might not agree on the timetable version or a consumer might not implement a PT service although the producer is providing one. As a result the producer and consumer will likely disagree on whether some journeys are cancelled or not. The producer might communicate a journey as cancelled over PT and therefore not deliver it over ET at all. That is why a consumer only listening to the ET data will not be informed explicitly that said journey is actually cancelled compared to the yearly timetable. The consumer is only informed implicitly by assuming cancellation based on the fact that said journey is not delivered in the initial load after any ET subscription. Setting proper baselines as described in chapter 10.2 is absolutely crucial.

10.6.2 Example

Careful: even though an update message as a result of a journey cancellation must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
```

```
      <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
      <LineRef>ch:1:Line:231:S23</LineRef>
      <DirectionRef>2</DirectionRef>
      <FramedVehicleJourneyRef>
       <DataFrameRef>2022-01-11</DataFrameRef>
       <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
      </FramedVehicleJourneyRef>
      <DatedVehicleJourneyIndirectRef>
       <OriginRef>ch:1:ScheduledStopPoint:998375543</OriginRef>
       <AimedDepartureTime>2022-01-11T08:13:00Z</AimedDepartureTime>
       <DestinationRef>ch:1:ScheduledStopPoint:998537577</DestinationRef>
       <AimedArrivalTime>2022-01-11T08:58:00Z</AimedArrivalTime>
      </DatedVehicleJourneyIndirectRef>
      <Cancellation>true</Cancellation>
      <VehicleMode>rail</VehicleMode>
      <PublishedLineName>S33</PublishedLineName>
      <OriginName>Leinstrand</OriginName>
      <DestinationName>Trondheim</DestinationName>
      <OperatorRef>ch:1:Operator:231</OperatorRef>
      <ProductCategoryRef>ch:1:TypeOfProductCategory:S</ProductCategoryRef>
      <VehicleRef>1029</VehicleRef>
      <TrainNumber>
       <TrainNumberRef>ch:1:TrainNumber:231:583</TrainNumberRef>
      </TrainNumber>
      <EstimatedCalls>
       <!-- EstimatedCalls are omitted for the sake of simplicity. -->
      </EstimatedCalls>
      <!-- A journey cancellation must always be a complete stop sequence. -->
      <IsCompleteStopSequence>true</IsCompleteStopSequence>
     </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
   </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.7 Partial cancellation

10.7.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = ‘true’ required

For example a technical issue with the locomotive that requires the train to end the journey prematurely at a stop not corresponding to the original destination. Passengers are asked to change trains (or even transfer to a replacement bus etc.) at the unexpected terminus.

A cancelled stop is always flagged explicitly with ***Cancellation*** = ’true’. See also chapter 7.16. What exactly a cancellation of a call implies or how a passenger information system should display it depends on the VehicleMode:

- For rail traffic a cancellation as specified in the XML example below will generally imply an exceptional passthrough as illustrated in the first part of Figure 9. A partial cancellation or multiple cancelled calls will either correspond to multiple such passthroughs or be accompanied by extra calls and in fact correspond to a change of the route or use case 10.10 (e.g., in case of damaged rail tracks).
- For busses or other unbound traffic, the XML example below is illustrated in the second part of Figure 9 which generally corresponds to a change of the route but without exceptional stops.

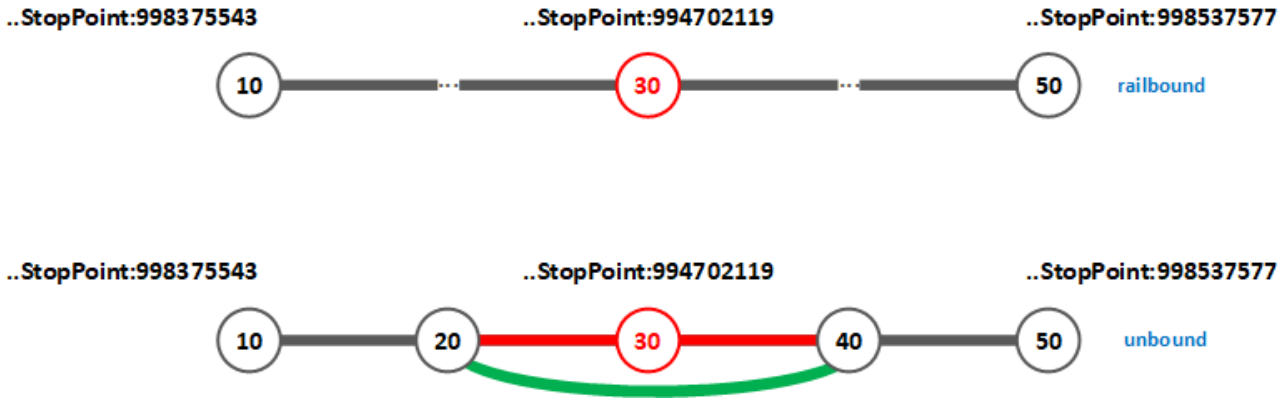


Figure 9 - Illustration of the partial cancellation example: different interpretations are possible depending on whether the vehicle is railbound (exceptional passthrough) or unbound (bypassing / rerouting)

10.7.2 Example

Careful: even though an update message as a result of partial cancellation / cancelled stops must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <SituationRef>ch:1:SituationNumber:oind23go198a56789</SituationRef>
     <EstimatedCalls>
      <EstimatedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:994702119</StopPointRef>
       <Order>30</Order>
       <Cancellation>true</Cancellation>
       <!-- It is highly recommended to always specify a reason in case of a
            Cancellation. For example:
       <CallNote>Stop is cancelled for security reasons.</CallNote> -->
      </EstimatedCall>
      <!-- Rest of the calls are omitted for the sake of simplicity.
           Keep in mind that an update message after a significant dispatching
           alteration must always be a complete stop sequence. -->
```

```
      </EstimatedCalls>
      <IsCompleteStopSequence>true</IsCompleteStopSequence>
     </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
   </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.8 Extra stop

10.8.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = ‘true’ required

An unplanned stop (with respect to the production or yearly timetable) is required for example due to an accident on the route (artificially inserted waiting time and option for the passengers so that they - at least partially - remain in control to fall back to another connection).

An extra stop is always flagged explicitly with ***ExtraStop*** = ’true’. See also chapter 7.16.

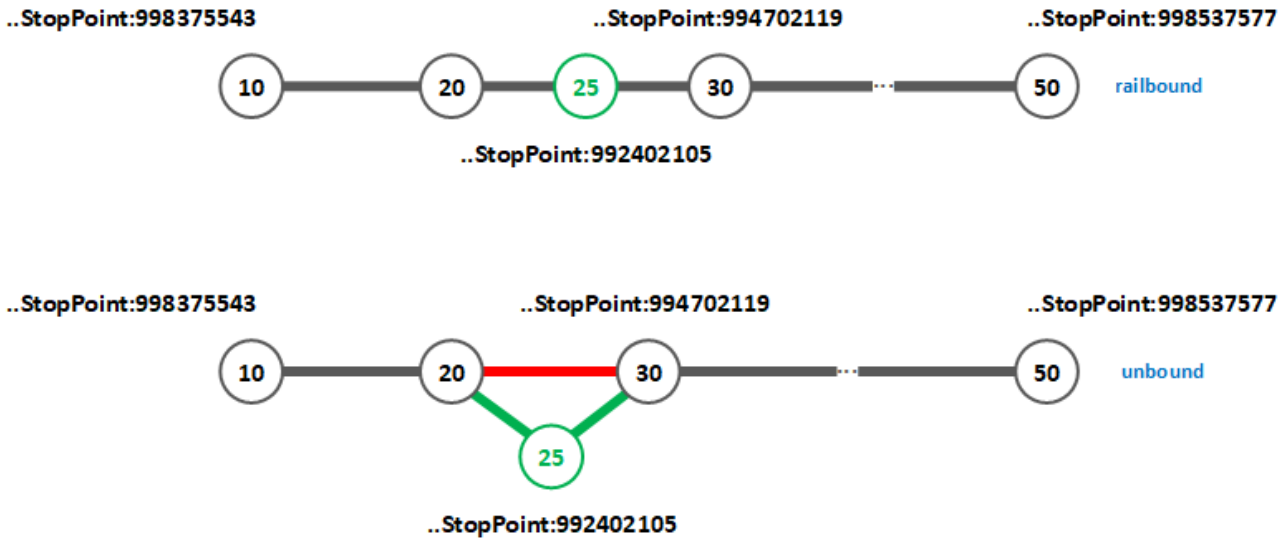


Figure 10 - Illustration of the extra call example: different interpretations are possible depending on whether the vehicle is railbound (exceptional stop) or unbound (bypassing / rerouting)

10.8.2 Example

Careful: even though an update message as a result of extra stops must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
```

```
    <EstimatedJourneyVersionFrame>
     <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
     <EstimatedVehicleJourney>
      <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
      <LineRef>ch:1:Line:231:S23</LineRef>
      <DirectionRef>2</DirectionRef>
      <FramedVehicleJourneyRef>
       <DataFrameRef>2022-01-11</DataFrameRef>
       <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
      </FramedVehicleJourneyRef>
      <EstimatedCalls>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:992402105</StopPointRef>
        <Order>25</Order>
        <ExtraCall>true</ExtraCall>
        <RequestStop>false</RequestStop>
        <AimedArrivalTime>2022-01-11T08:31:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2022-01-11T08:31:00Z</ExpectedArrivalTime>
        <ArrivalStatus>onTime</ArrivalStatus>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <ArrivalStopAssignment>
         <AimedQuayRef>ch:1:Quay:55864460:9</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:55864460:9</ExpectedQuayRef>
        </ArrivalStopAssignment>
        <AimedDepartureTime>2022-01-11T08:32:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:32:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:55864460:9</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:55864460:9</ExpectedQuayRef>
        </DepartureStopAssignment>
       </EstimatedCall>
       <!-- Rest of the calls are omitted for the sake of simplicity.
            Keep in mind that an update message after a significant dispatching
            alteration must always be a complete stop sequence. -->
      </EstimatedCalls>
      <IsCompleteStopSequence>true</IsCompleteStopSequence>
```

```
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.9 Exceptional passthru

10.9.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = 'true' required

A planned stop of the journey cannot be called for example due to an accident or unplanned roadworks (unplanned in the sense of not taken into account in the periodic timetable). Such a case of an exceptional passthru is specified within the respective EstimatedCall by setting Arrival- and DepartureBoardingActivity to 'passThru'. In special caces where, for example, a vehicle is scheduled to arrive at platform '9AB' and depart from '9CD' but for whatever reason the arrival is cancelled, it is also possible that only the ArrivalBoardingActivity is set to 'passThru'.

In addition to BoardingActivity, Arrival- and/or DepartureStatus can be set to 'cancelled'. From SIRI 2.1 onwards the status 'cancelled' must always be used in combination with Arrival-/DepartureCancellationReason (e.g. "notRequested" or "roadworks").

10.9.2 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (C) Copyright 2005-2018 CEN SIRI SBB EMX -->
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
  <ServiceDelivery>
    <ResponseTimestamp>2017-09-25T09:18:59Z</ResponseTimestamp>
    <ProducerRef>cus_prod</ProducerRef>
    <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
    <EstimatedTimetableDelivery version="2.0" xmlns="http://www.siri.org.uk/siri">
      <ResponseTimestamp>2018-04-11T13:50:56Z</ResponseTimestamp>
      <SubscriptionRef>25150636626</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2018-04-11T13:50:56.123456Z</RecordedAtTime>
        <EstimatedVehicleJourney>
          <LineRef>ch:1:Line:21557:S5</LineRef>
```

```
        <DirectionRef>ch:1:Direction:H</DirectionRef>
        <DatedVehicleJourneyRef>85:86:21557:001</DatedVehicleJourneyRef>
        <PublishedLineName>S 5</PublishedLineName>
        <DirectionName>Luzern</DirectionName>
        <ProductCategoryRef>ch:1:TypeOfProductCategory:S</ProductCategoryRef>
        <Monitored>true</Monitored>
        <EstimatedCalls>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508321</StopPointRef>
            <StopPointName>Giswil</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedDepartureTime>2018-04-11T13:05:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:05:18Z</ExpectedDepartureTime>
            <DeparturePlatformName>3</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508327</StopPointRef>
            <StopPointName>Ewil Maxon</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:09:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:09:42Z</ExpectedArrivalTime>
            <ArrivalPlatformName>1</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:09:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:10:24Z</ExpectedDepartureTime>
            <DeparturePlatformName>1</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <!-- An exceptional transit or passthru indicated by the respective BoardingActivity and Status -->
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508313</StopPointRef>
            <StopPointName>Sachseln</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <ArrivalStatus>cancelled</ArrivalStatus>
            <ArrivalPlatformName>1</ArrivalPlatformName>
```

```
            <ArrivalBoardingActivity>passThru</ArrivalBoardingActivity>
            <DepartureStatus>cancelled</DepartureStatus>
            <DeparturePlatformName>1</DeparturePlatformName>
            <DepartureBoardingActivity>passThru</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508314</StopPointRef>
            <StopPointName>Sarnen</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:17:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:16:30Z</ExpectedArrivalTime>
            <ArrivalPlatformName>1</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:19:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:19:06Z</ExpectedDepartureTime>
            <DeparturePlatformName>1</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508323</StopPointRef>
            <StopPointName>Sarnen Nord</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:20:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:21:24Z</ExpectedArrivalTime>
            <ArrivalPlatformName>1</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:20:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:22:00Z</ExpectedDepartureTime>
            <DeparturePlatformName>1</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508316</StopPointRef>
            <StopPointName>Alpnach Dorf</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:24:00Z</AimedArrivalTime>
```

```
            <ExpectedArrivalTime>2018-04-11T13:25:12Z</ExpectedArrivalTime>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:24:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:25:54Z</ExpectedDepartureTime>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508317</StopPointRef>
            <StopPointName>Alpnachstad</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:28:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:28:42Z</ExpectedArrivalTime>
            <ArrivalPlatformName>1</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:29:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:29:12Z</ExpectedDepartureTime>
            <DeparturePlatformName>1</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508318</StopPointRef>
            <StopPointName>Hergiswil</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:33:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:33:24Z</ExpectedArrivalTime>
            <ArrivalPlatformName>3</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:34:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:34:00Z</ExpectedDepartureTime>
            <DeparturePlatformName>3</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508319</StopPointRef>
            <StopPointName>Horw</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
```

```
            <AimedArrivalTime>2018-04-11T13:38:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:38:30Z</ExpectedArrivalTime>
            <ArrivalPlatformName>3</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:38:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:39:06Z</ExpectedDepartureTime>
            <DeparturePlatformName>3</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8516351</StopPointRef>
            <StopPointName>Kriens Mattenhof</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:40:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:41:18Z</ExpectedArrivalTime>
            <ArrivalPlatformName>2</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:41:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:41:54Z</ExpectedDepartureTime>
            <DeparturePlatformName>2</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8508321</StopPointRef>
            <StopPointName>Luzern Allmend/Messe</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:42:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:42:54Z</ExpectedArrivalTime>
            <ArrivalPlatformName>2</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-04-11T13:42:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-04-11T13:43:30Z</ExpectedDepartureTime>
            <DeparturePlatformName>2</DeparturePlatformName>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <StopPointRef>ch:1:ScheduledStopPoint:8505000</StopPointRef>
```

```
            <StopPointName>Luzern</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Luzern</DestinationDisplay>
            <AimedArrivalTime>2018-04-11T13:47:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-04-11T13:48:48Z</ExpectedArrivalTime>
            <ArrivalPlatformName>14</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
          </EstimatedCall>
        </EstimatedCalls>
        <IsCompleteStopSequence>true</IsCompleteStopSequence>
      </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.10 Rerouting

SIRI part 3 (CEN/TS 15531-3), chapter 6.8.4 states:
In the case of major despatching alterations, it may be necessary to retransmit information for the entire route, including new planning and prognosis times. This is the case for any additional journeys not included in the planning data (see chapter 10.5), as well as for any path changes to an existing route or stop sequence of a journey. IsCompleteStopSequence of an EstimatedVehicleJourney shall be set to "true" to inform the schedule information system that an entire route is being exchanged: if the path change involves the omission of stops when compared with the original plan, they are omitted altogether.

10.10.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = 'true' required

Rerouting of a journey is for example required in case of a broken train or road accident effectively blocking a section of the route.

In case of path changes due to the cancellation of stops, the corresponding calls must always be transmitted explicitly (instead of being omitted) and also marked with ***Cancellation*** = 'true' at least in the first update after the information is made available to the producing system (considering incremental udpates) or in any delivery with ***CompleteStopSequence*** = 'true' to properly establish a new baseline. Rerouting is a combination of use cases 10.7 and **Error! Reference source not found.**. See also chapter 7.16.2 for the specifics in case of historization of rerouted calls.

Depending on the ***StopPlace*** model and what type of object is referenced in ***StopPointRef*** of a call, rerouting is necessary more frequently:

a. In rail traffic ***StopPointRef*** of a call generally corresponds to a stop place which implies that platform/quay changes are mapped to simple ****PlatformName*** or ****StopAssignment*** updates without changes to the overall stop sequence, i.e., use case 10.4.

b. However, in local public transport (bus, tram etc.) ***StopPointRef*** of a call typically corresponds to a platform/quay directly which implies that changes to a platform/quay will always result in a rerouting update according to use case 10.10. This complicates matters for consuming systems like journey planner because of routing graph calculations.

10.10.2 Example

10.10.2.1 Combination of cancelled and extra calls

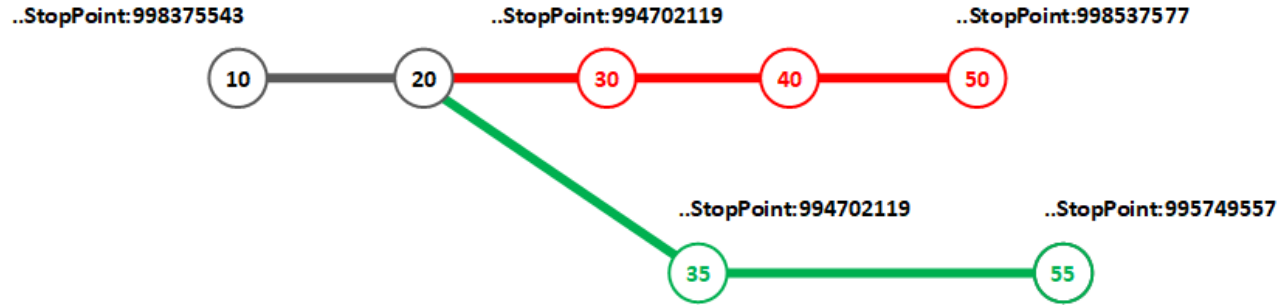


Figure 11 - Illustration of the rerouting example: original route is partially cancelled and replaced

Careful: even though an update message as a result of a rerouting must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="utf-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
```

```
    <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
    <SubscriptionRef>1506</SubscriptionRef>
    <EstimatedJourneyVersionFrame>
     <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
     <EstimatedVehicleJourney>
      <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
      <LineRef>ch:1:Line:231:S23</LineRef>
      <DirectionRef>2</DirectionRef>
      <FramedVehicleJourneyRef>
       <DataFrameRef>2022-01-11</DataFrameRef>
       <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
      </FramedVehicleJourneyRef>
      <EstimatedCalls>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:994702119</StopPointRef>
        <Order>30</Order>
        <Cancellation>true</Cancellation>
        <CallNote>Stop is cancelled and the journey rerouted due to a derailment.</CallNote>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:994702119</StopPointRef>
        <Order>35</Order>
        <ExtraCall>true</ExtraCall>
        <CallNote>Side of alighting: to the right</CallNote>
        <AimedArrivalTime>2022-01-11T08:46:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2022-01-11T08:46:00Z</ExpectedArrivalTime>
        <ArrivalStatus>onTime</ArrivalStatus>
        <ArrivalPlatformName>1</ArrivalPlatformName>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <ArrivalStopAssignment>
         <AimedQuayRef>ch:1:Quay:99057203:22</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:99057203:22</ExpectedQuayRef>
        </ArrivalStopAssignment>
        <AimedDepartureTime>2022-01-11T08:47:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:47:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DeparturePlatformName>1</DeparturePlatformName>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
```

```
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:99057203:22</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:99057203:22</ExpectedQuayRef>
        </DepartureStopAssignment>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
        <Order>40</Order>
        <Cancellation>true</Cancellation>
        <CallNote>Stop is cancelled and the journey rerouted due to a derailment.</CallNote>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:998537577</StopPointRef>
        <Order>50</Order>
        <Cancellation>true</Cancellation>
        <CallNote>Stop is cancelled and the journey rerouted due to a derailment.</CallNote>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:995749557</StopPointRef>
        <Order>55</Order>
        <ExtraCall>true</ExtraCall>
        <AimedArrivalTime>2022-01-11T08:58:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2022-01-11T08:58:00Z</ExpectedArrivalTime>
        <ArrivalStatus>onTime</ArrivalStatus>
        <ArrivalPlatformName>2</ArrivalPlatformName>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <ArrivalStopAssignment>
         <AimedQuayRef>ch:1:Quay:99053042:61</AimedQuayRef>
         <ExpectedQuayRef>ch:1:Quay:99053042:61</ExpectedQuayRef>
        </ArrivalStopAssignment>
       </EstimatedCall>
       <!-- Rest of the calls are omitted for the sake of simplicity.
            Keep in mind that an update message after a significant dispatching
            alteration must always be a complete stop sequence. -->
      </EstimatedCalls>
      <IsCompleteStopSequence>true</IsCompleteStopSequence>
     </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
```

```
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.10.2.2 Example of a journey from Zürich to Bern rerouted through Brugg AG

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- A redirection can't be distinguished from extra call use case. -->
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0"
xsi:schemaLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
  <ServiceDelivery>
    <ResponseTimestamp>2018-04-11T05:30:00Z</ResponseTimestamp>
    <ProducerRef>cus_prod</ProducerRef>
    <RequestMessageRef>e2fca467-0403-4216-8ce0-c2d3324aa52c</RequestMessageRef>
    <EstimatedTimetableDelivery version="2.0" xmlns="http://www.siri.org.uk/siri">
      <ResponseTimestamp>2018-04-11T05:30:00Z</ResponseTimestamp>
      <SubscriptionRef>25150636626</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2018-04-11T05:30:00Z</RecordedAtTime>
        <EstimatedVehicleJourney>
          <LineRef>ch:1:Line:001408:IC1</LineRef>
          <DirectionRef>ch:1:Direction:H</DirectionRef>
          <FramedVehicleJourneyRef>
            <DataFrameRef>2018-04-11</DataFrameRef>
            <DatedVehicleJourneyRef>85:11:1408:001</DatedVehicleJourneyRef>
          </FramedVehicleJourneyRef>
          <PublishedLineName>IC 1</PublishedLineName>
          <DirectionName>Bern</DirectionName>
          <ProductCategoryRef>ch:1:TypeOfProductCategory:ICE</ProductCategoryRef>
          <Monitored>true</Monitored>
          <EstimatedCalls>
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8503000</StopPointRef>
              <StopPointName>Zürich HB</StopPointName>
              <AimedDepartureTime>2018-04-11T05:50:00Z</AimedDepartureTime>
              <ExpectedDepartureTime>2018-04-11T05:50:24Z</ExpectedDepartureTime>
              <DeparturePlatformName>14</DeparturePlatformName>
              <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
            </EstimatedCall>
```

```
            <!-- Redirection and extra stop, route is different -->
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8500307</StopPointRef>
              <StopPointName>Brugg AG</StopPointName>
              <ExtraCall>true</ExtraCall>
              <AimedArrivalTime>2018-04-11T06:18:00Z</AimedArrivalTime>
              <ExpectedArrivalTime>2018-04-11T06:18:12Z</ExpectedArrivalTime>
              <ArrivalPlatformName>4</ArrivalPlatformName>
              <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
              <AimedDepartureTime>2018-04-11T06:19:00Z</AimedDepartureTime>
              <ExpectedDepartureTime>2018-04-11T06:19:24Z</ExpectedDepartureTime>
              <DeparturePlatformName>4</DeparturePlatformName>
              <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
            </EstimatedCall>
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8507000</StopPointRef>
              <StopPointName>Bern</StopPointName>
              <AimedArrivalTime>2018-04-11T06:50:00Z</AimedArrivalTime>
              <ExpectedArrivalTime>2018-04-11T06:50:12Z</ExpectedArrivalTime>
              <ArrivalPlatformName>8</ArrivalPlatformName>
              <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            </EstimatedCall>
          </EstimatedCalls>
          <IsCompleteStopSequence>true</IsCompleteStopSequence>
        </EstimatedVehicleJourney>
      </EstimatedJourneyVersionFrame>
    </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.10.2.3 Example of a journey from Basel SBB to Bern rerouted through Sissach with Liestal being cancelled

```
<?xml version="1.0" encoding="utf-8"?>
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:schemaLocation="http://www.siri.org.uk/siri D:\Arbeit\SIRI\siri-2.0o-
xsd\siri.xsd" version="2.0o">
  <ServiceDelivery>
    <ResponseTimestamp>2018-11-06T08:57:52Z</ResponseTimestamp>
    <ProducerRef>ETReroutingExample</ProducerRef>
```

```
    <RequestMessageRef>e2fca467-0403-4216-8ce0-c2d3324aa52c</RequestMessageRef>
    <EstimatedTimetableDelivery>
      <ResponseTimestamp>2018-11-06T08:57:32Z</ResponseTimestamp>
      <SubscriptionRef>1</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2018-11-06T08:57:32Z</RecordedAtTime>
        <EstimatedVehicleJourney>
          <LineRef>ICE373</LineRef>
          <DirectionRef>ch:1:Direction:H</DirectionRef>
          <FramedVehicleJourneyRef>
            <DataFrameRef>2018-11-06</DataFrameRef>
            <DatedVehicleJourneyRef>85:11:373:001</DatedVehicleJourneyRef>
          </FramedVehicleJourneyRef>
          <DirectionName>Bern</DirectionName>
          <ProductCategoryRef>ch:1:TypeOfProductCategory:ICE</ProductCategoryRef>
          <Monitored>true</Monitored>
          <EstimatedCalls>
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8500010</StopPointRef>
              <StopPointName>Basel SBB</StopPointName>
              <DestinationDisplay>Bern</DestinationDisplay>
              <AimedDepartureTime>2018-10-06T19:59:00Z</AimedDepartureTime>
              <ExpectedDepartureTime>2018-10-06T20:04:00Z</ExpectedDepartureTime>
              <DepartureStatus>delayed</DepartureStatus>
              <DeparturePlatformName>5</DeparturePlatformName>
              <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
            </EstimatedCall>
            <!--Cancelled Call because of rerouting-->
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8500023</StopPointRef>
              <StopPointName>Liestal</StopPointName>
              <Cancellation>true</Cancellation>
              <OriginDisplay>Basel SBB</OriginDisplay>
              <DestinationDisplay>Bern</DestinationDisplay>
              <AimedArrivalTime>2018-10-06T20:08:00Z</AimedArrivalTime>
              <ArrivalStatus>cancelled</ArrivalStatus>
              <ArrivalPlatformName>1</ArrivalPlatformName>
              <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
              <AimedDepartureTime>2018-10-06T20:09:00Z</AimedDepartureTime>
```

```
      <DepartureStatus>cancelled</DepartureStatus>
      <DeparturePlatformName>1</DeparturePlatformName>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <!--ExtraCall because of rerouting-->
    <EstimatedCall>
      <StopPointRef>ch:1:ScheduledStopPoint:8500026</StopPointRef>
      <StopPointName>Sissach</StopPointName>
      <ExtraCall>true</ExtraCall>
      <OriginDisplay>Basel SBB</OriginDisplay>
      <DestinationDisplay>Bern</DestinationDisplay>
      <AimedArrivalTime>2018-10-06T20:17:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-10-06T20:17:00Z</ExpectedArrivalTime>
      <ArrivalPlatformName>1</ArrivalPlatformName>
      <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
      <AimedDepartureTime>2018-10-06T20:18:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-10-06T20:18:00Z</ExpectedDepartureTime>
      <DeparturePlatformName>1</DeparturePlatformName>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <StopPointRef>ch:1:ScheduledStopPoint:8500218</StopPointRef>
      <StopPointName>Olten</StopPointName>
      <OriginDisplay>Basel SBB</OriginDisplay>
      <DestinationDisplay>Bern</DestinationDisplay>
      <AimedArrivalTime>2018-10-06T20:25:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-10-06T20:31:00Z</ExpectedArrivalTime>
      <ArrivalPlatformName>11</ArrivalPlatformName>
      <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
      <AimedDepartureTime>2018-10-06T20:28:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-10-06T20:34:00Z</ExpectedDepartureTime>
      <DeparturePlatformName>11</DeparturePlatformName>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <StopPointRef>ch:1:ScheduledStopPoint:8507000</StopPointRef>
      <StopPointName>Bern</StopPointName>
      <AimedArrivalTime>2018-10-06T20:56:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-10-06T21:03:00Z</ExpectedArrivalTime>
```

```
              <ArrivalPlatformName>4</ArrivalPlatformName>
              <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            </EstimatedCall>
          </EstimatedCalls>
          <IsCompleteStopSequence>true</IsCompleteStopSequence>
        </EstimatedVehicleJourney>
      </EstimatedJourneyVersionFrame>
    </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.11 Recording of calls

10.11.1 Business

Recording of the actual arrival and departure time is mandatory in public transport CH.

10.11.2 Example

See chapter 7.19 for all the cases that must be considered when generating a ***RecordedCall***.

10.12 Vehicle waiting at stop

10.12.1 Business

Incremental updates support: ***IsCompleteStopSequence*** = ‘false’ **allowed**

10.12.2 Example

See chapter 7.19.10 for payload examples of this use case.

10.13 Boarding activity update

10.13.1 Business

Incremental updates support: ***IsCompleteStopSequence*** = ‘false’ **allowed**

We differentiate between:

a. Change of ***BoardingActivity** to either ‘noAlighting’ or ‘noBoarding’ over ET. Those events are rarely encountered. For example, if applied in case of a school bus journey, restrictions of the boarding activity are usually already known and thus communicated via periodic timetable. Only in exceptional situations like XX is a short-term change of the activity necessary.

b. Change of ***BoardingActivity** to ‘passThru’ over ET. A scheduled stop cannot be “called”, for example, due to an accident or unplanned road-works (might be planned but too short-term for it to be already updated in the periodic timetable). In exceptional cases where, for example, a vehicle is scheduled to arrive at platform ‘9AB’ and depart from ‘9CD’, but for whatever reason the arrival is cancelled, it is also possible that only ***ArrivalBoardingActivity*** is set to ‘passThru’ (or vice versa).

10.13.2 Example

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
```

```
      <!-- Reference to SIRI-SX for more (general) information about the situation
           (in this case: no alighting/boarding at some stops). -->
      <SituationRef>ch:1:SituationNumber:ghijkl0123456789</SituationRef>
      <EstimatedCalls>
       <!-- The Vehicle is en route to the third stop, hence there are no
            EstimatedCalls for the first two stops anymore. Any RecordedCalls for
            these stops are not shown in this example. -->
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:994702119</StopPointRef>
        <Order>30</Order>
        <!-- Because the town is overcrowded, it is no longer allowed to get off
             the vehicle at this stop. -->
        <ArrivalBoardingActivity>noAlighting</ArrivalBoardingActivity>
        <!-- No change in DepartureBoardingActivity, but included just to avoid
             ambiguity. -->
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
       </EstimatedCall>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
        <Order>40</Order>
        <!-- Because the town is overcrowded, the stop is (temporarily) not in use
             And the vehicle will not stop there. -->
        <ArrivalBoardingActivity>passThru</ArrivalBoardingActivity>
        <DepartureBoardingActivity>passThru</DepartureBoardingActivity>
       </EstimatedCall>
       <!-- No changes w.r.t. the last stop, i.e. the destination of a journey,
            hence no EstimatedCall for that stop. -->
      </EstimatedCalls>
      <!-- The stop sequence is not complete because only the calls with changes
           have been included in the message. -->
      <IsCompleteStopSequence>false</IsCompleteStopSequence>
     </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
   </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.14 No alighting update

10.14.1 Business

For some stops of a journey passengers are only allowed to board the vehicle whereas alighting is prohibited.

10.14.2 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
  <ServiceDelivery>
    <ResponseTimestamp>2018-05-09T14:39:05Z</ResponseTimestamp>
    <ProducerRef>cus_prod</ProducerRef>
    <RequestMessageRef>a08d40cb-d438-47c8-af73-22e9c8080a38</RequestMessageRef>
    <EstimatedTimetableDelivery version="2.0" xmlns="http://www.siri.org.uk/siri">
      <ResponseTimestamp>2018-05-09T14:39:56Z</ResponseTimestamp>
      <SubscriptionRef>25150636626</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2018-05-09T14:39:56Z</RecordedAtTime>
        <EstimatedVehicleJourney>
          <!-- Bus from Lugano, Centro to Lamone-Cadempino with partial prohibition of alighting = Boarding only -->
          <LineRef>ch:1:Line:11:441334</LineRef>
          <DirectionRef>ch:1:Direction:H</DirectionRef>
          <EstimatedVehicleJourneyCode>85:11:441334:001</EstimatedVehicleJourneyCode>
          <ExtraJourney>true</ExtraJourney>
          <PublishedLineName>441</PublishedLineName>
          <DirectionName>Lamone-Cadempino, Stazione</DirectionName>
          <ProductCategoryRef>ch:1:TypeOfProductCategory:B</ProductCategoryRef>
          <Monitored>true</Monitored>
          <EstimatedCalls>
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8591603</StopPointRef>
              <StopPointName>Lugano, Centro</StopPointName>
              <OriginDisplay>Lugano, Centro</OriginDisplay>
              <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
              <AimedDepartureTime>2018-05-09T14:58:00Z</AimedDepartureTime>
              <ExpectedDepartureTime>2018-05-09T14:58:00Z</ExpectedDepartureTime>
              <DeparturePlatformName>3</DeparturePlatformName>
```

```
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <!-- Halt mit Aussteigeverbot = noAlighting; nur zum Einsteigen = boarding -->
      <StopPointRef>ch:1:ScheduledStopPoint:8588973</StopPointRef>
      <StopPointName>Lugano, Autosilo</StopPointName>
      <OriginDisplay>Lugano, Centro</OriginDisplay>
      <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
      <AimedArrivalTime>2018-05-09T14:59:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-05-09T14:59:00Z</ExpectedArrivalTime>
      <ArrivalPlatformName>A</ArrivalPlatformName>
      <ArrivalBoardingActivity>noAlighting</ArrivalBoardingActivity>
      <AimedDepartureTime>2018-05-09T14:59:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-05-09T14:59:00Z</ExpectedDepartureTime>
      <DeparturePlatformName>A</DeparturePlatformName>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <!-- Halt mit Aussteigeverbot = noAlighting; nur zum Einsteigen = boarding -->
      <StopPointRef>ch:1:ScheduledStopPoint:8579001</StopPointRef>
      <StopPointName>Lugano, Piazza Molino Nuovo</StopPointName>
      <OriginDisplay>Lugano, Centro</OriginDisplay>
      <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
      <AimedArrivalTime>2018-05-09T15:01:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-05-09T15:01:00Z</ExpectedArrivalTime>
      <ArrivalBoardingActivity>noAlighting</ArrivalBoardingActivity>
      <AimedDepartureTime>2018-05-09T15:01:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-05-09T15:01:00Z</ExpectedDepartureTime>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <!-- Halt mit Aussteigeverbot = noAlighting; nur zum Einsteigen = boarding -->
      <StopPointRef>ch:1:ScheduledStopPoint:8579006</StopPointRef>
      <StopPointName>Lugano, Vignola</StopPointName>
      <OriginDisplay>Lugano, Centro</OriginDisplay>
      <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
      <AimedArrivalTime>2018-05-09T15:03:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-05-09T15:03:00Z</ExpectedArrivalTime>
      <ArrivalBoardingActivity>noAlighting</ArrivalBoardingActivity>
```

```
      <AimedDepartureTime>2018-05-09T15:03:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-05-09T15:03:00Z</ExpectedDepartureTime>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <!-- Halt mit Aussteigeverbot = noAlighting; nur zum Einsteigen = boarding -->
      <StopPointRef>ch:1:ScheduledStopPoint:8578997</StopPointRef>
      <StopPointName>Lugano, Gerra</StopPointName>
      <OriginDisplay>Lugano, Centro</OriginDisplay>
      <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
      <AimedArrivalTime>2018-05-09T15:06:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-05-09T15:06:00Z</ExpectedArrivalTime>
      <ArrivalBoardingActivity>noAlighting</ArrivalBoardingActivity>
      <AimedDepartureTime>2018-05-09T15:06:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-05-09T15:06:00Z</ExpectedDepartureTime>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <!-- Halt mit Ein- und Aussteigen = Stop with boarding and alighting -->
      <StopPointRef>ch:1:ScheduledStopPoint:8578996</StopPointRef>
      <StopPointName>Trevano, Centro Studi</StopPointName>
      <OriginDisplay>Lugano, Centro</OriginDisplay>
      <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
      <AimedArrivalTime>2018-05-09T15:10:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-05-09T15:10:00Z</ExpectedArrivalTime>
      <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
      <AimedDepartureTime>2018-05-09T15:11:00Z</AimedDepartureTime>
      <ExpectedDepartureTime>2018-05-09T15:11:00Z</ExpectedDepartureTime>
      <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
    </EstimatedCall>
    <EstimatedCall>
      <!-- Halt mit Ein- und Aussteigen = Stop with boarding and alighting -->
      <StopPointRef>ch:1:ScheduledStopPoint:8575343</StopPointRef>
      <StopPointName>Comano, Studio TV</StopPointName>
      <OriginDisplay>Lugano, Centro</OriginDisplay>
      <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
      <AimedArrivalTime>2018-05-09T15:16:00Z</AimedArrivalTime>
      <ExpectedArrivalTime>2018-05-09T15:16:00Z</ExpectedArrivalTime>
      <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
```

```
            <AimedDepartureTime>2018-05-09T15:16:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-05-09T15:16:00Z</ExpectedDepartureTime>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <!-- Halt mit Ein- und Aussteigen = Stop with boarding and alighting -->
            <StopPointRef>ch:1:ScheduledStopPoint:8575317</StopPointRef>
            <StopPointName>Cadempino, Municipio</StopPointName>
            <OriginDisplay>Lugano, Centro</OriginDisplay>
            <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
            <AimedArrivalTime>2018-05-09T15:21:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-05-09T15:21:00Z</ExpectedArrivalTime>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-05-09T15:21:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-05-09T15:21:00Z</ExpectedDepartureTime>
            <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <!-- Endhalt = Terminal stop -->
            <StopPointRef>ch:1:ScheduledStopPoint:8575319</StopPointRef>
            <StopPointName>Lamone-Cadempino, Stazione</StopPointName>
            <OriginDisplay>Lugano, Centro</OriginDisplay>
            <DestinationDisplay>Lamone-Cadempino, Stazione</DestinationDisplay>
            <AimedArrivalTime>2018-05-09T15:24:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-05-09T15:24:00Z</ExpectedArrivalTime>
            <ArrivalPlatformName>A</ArrivalPlatformName>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <DepartureBoardingActivity>noBoarding</DepartureBoardingActivity>
          </EstimatedCall>
        </EstimatedCalls>
        <IsCompleteStopSequence>true</IsCompleteStopSequence>
      </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.15 No boarding update

10.15.1 Business

For example a school bus that only allows passenger to alight from the vehicle but not to board it.

10.15.2 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0" xsi:sche-
maLocation="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
  <ServiceDelivery>
    <ResponseTimestamp>2018-05-09T17:19:05Z</ResponseTimestamp>
    <ProducerRef>cus_prod</ProducerRef>
    <RequestMessageRef>a08d40cb-d438-47c8-af73-22e9c8080a38</RequestMessageRef>
    <EstimatedTimetableDelivery version="2.0" xmlns="http://www.siri.org.uk/siri">
      <ResponseTimestamp>2018-05-09T16:34:56Z</ResponseTimestamp>
      <SubscriptionRef>25150636626</SubscriptionRef>
      <EstimatedJourneyVersionFrame>
        <RecordedAtTime>2018-05-09T16:34:56Z</RecordedAtTime>
        <EstimatedVehicleJourney>
          <!-- Bus from Giswil to Kleinteil, Post with partial prohibition of boarding = Alighting only -->
          <LineRef>ch:1:Line:11:36122</LineRef>
          <DirectionRef>ch:1:Direction:H</DirectionRef>
          <EstimatedVehicleJourneyCode>85:11:36122:001</EstimatedVehicleJourneyCode>
          <ExtraJourney>true</ExtraJourney>
          <PublishedLineName>361</PublishedLineName>
          <DirectionName>Kleinteil, Post</DirectionName>
          <ProductCategoryRef>ch:1:TypeOfProductCategory:B</ProductCategoryRef>
          <Monitored>true</Monitored>
          <EstimatedCalls>
            <EstimatedCall>
              <StopPointRef>ch:1:ScheduledStopPoint:8572893</StopPointRef>
              <StopPointName>Giswil, Bahnhof</StopPointName>
              <OriginDisplay>Giswil</OriginDisplay>
              <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
              <AimedDepartureTime>2018-05-09T17:39:00Z</AimedDepartureTime>
              <ExpectedDepartureTime>2018-05-09T17:39:00Z</ExpectedDepartureTime>
              <DeparturePlatformName>1</DeparturePlatformName>
```

```
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
      </EstimatedCall>
      <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:8572892</StopPointRef>
        <StopPointName>Giswil, Kirche</StopPointName>
        <OriginDisplay>Giswil</OriginDisplay>
        <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
        <AimedArrivalTime>2018-05-09T17:40:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2018-05-09T17:39:45Z</ExpectedArrivalTime>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <AimedDepartureTime>2018-05-09T17:40:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2018-05-09T17:40:00Z</ExpectedDepartureTime>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
      </EstimatedCall>
      <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:8572917</StopPointRef>
        <StopPointName>Grossteil, Schulhaus</StopPointName>
        <OriginDisplay>Giswil</OriginDisplay>
        <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
        <AimedArrivalTime>2018-05-09T17:42:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2018-05-09T17:42:00Z</ExpectedArrivalTime>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <AimedDepartureTime>2018-05-09T17:42:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2018-05-09T17:42:00Z</ExpectedDepartureTime>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
      </EstimatedCall>
      <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:8582768</StopPointRef>
        <StopPointName>Grossteil, Zopf</StopPointName>
        <OriginDisplay>Giswil</OriginDisplay>
        <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
        <AimedArrivalTime>2018-05-09T17:44:00Z</AimedArrivalTime>
        <ExpectedArrivalTime>2018-05-09T17:44:00Z</ExpectedArrivalTime>
        <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
        <AimedDepartureTime>2018-05-09T17:44:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2018-05-09T17:44:00Z</ExpectedDepartureTime>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
      </EstimatedCall>
      <EstimatedCall>
```

```
            <!-- Halt mit Einsteigeverbot -->
            <StopPointRef>ch:1:ScheduledStopPoint:8580014</StopPointRef>
            <StopPointName>Grossteil, Rüti</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
            <AimedArrivalTime>2018-05-09T17:46:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-05-09T17:46:00Z</ExpectedArrivalTime>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-05-09T17:46:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-05-09T17:46:00Z</ExpectedDepartureTime>
            <DepartureBoardingActivity>noBoarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <!-- Halt mit Einsteigeverbot -->
            <StopPointRef>ch:1:ScheduledStopPoint:8582883</StopPointRef>
            <StopPointName>Grossteil, Haltenrain</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
            <AimedArrivalTime>2018-05-09T17:47:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-05-09T17:47:00Z</ExpectedArrivalTime>
            <ArrivalBoardingActivity>alighting</ArrivalBoardingActivity>
            <AimedDepartureTime>2018-05-09T17:47:00Z</AimedDepartureTime>
            <ExpectedDepartureTime>2018-05-09T17:47:00Z</ExpectedDepartureTime>
            <DepartureBoardingActivity>noBoarding</DepartureBoardingActivity>
          </EstimatedCall>
          <EstimatedCall>
            <!-- Endhalt mit Einsteigeverbot -->
            <StopPointRef>ch:1:ScheduledStopPoint:8572919</StopPointRef>
            <StopPointName>Kleinteil, Post</StopPointName>
            <OriginDisplay>Giswil</OriginDisplay>
            <DestinationDisplay>Kleinteil, Post</DestinationDisplay>
            <AimedArrivalTime>2018-05-09T17:55:00Z</AimedArrivalTime>
            <ExpectedArrivalTime>2018-05-09T17:55:00Z</ExpectedArrivalTime>
            <ArrivalBoardingActivity>noAlighting</ArrivalBoardingActivity>
            <DepartureBoardingActivity>noBoarding</DepartureBoardingActivity>
          </EstimatedCall>
        </EstimatedCalls>
      </EstimatedVehicleJourney>
    </EstimatedJourneyVersionFrame>
```

```
    </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.16 Replacement transport and journey relations

LATER (SIRI 2.1)

10.16.1 Business

a. Incremental updates support if only a ***JourneyRelation*** is added or changed within the ***EstimatedVehicleJourney*** (as described in the XML example below): ***IsCompleteStopSequence*** = 'false' **allowed**

b. Incremental updates support if the ***EstimatedVehicleJourney*** is otherwise changed, e.g. ***Cancellation*** = 'true' is set or the route has changed: **not allowed**, i.e., always ***IsCompleteStopSequence*** = 'true' required.

JourneyRelations are used in case a journey is affected by a situation and subject to replacement services or in some instances of border crossing where the same journey is operated by different organisations. ***JourneyRelations*** are generally transmitted in pairs, i.e., on journey A that is related to by journey B and on journey B that relates to journey A. Therefore the ***JourneyRelationTypes*** come in pairs, e.g., 'ReplacedByJourney' and 'ReplacementOfJourney'.

In case of replacement or support (see CEN/TS 15531-2, chapter "JourneyRelationType"), one must always specify the respective ***JourneyPartInfo*** (or call otherwise) to specify exactly which part of the route is being replaced or supported as illustrated in Figure 12.

10.16.2 Example

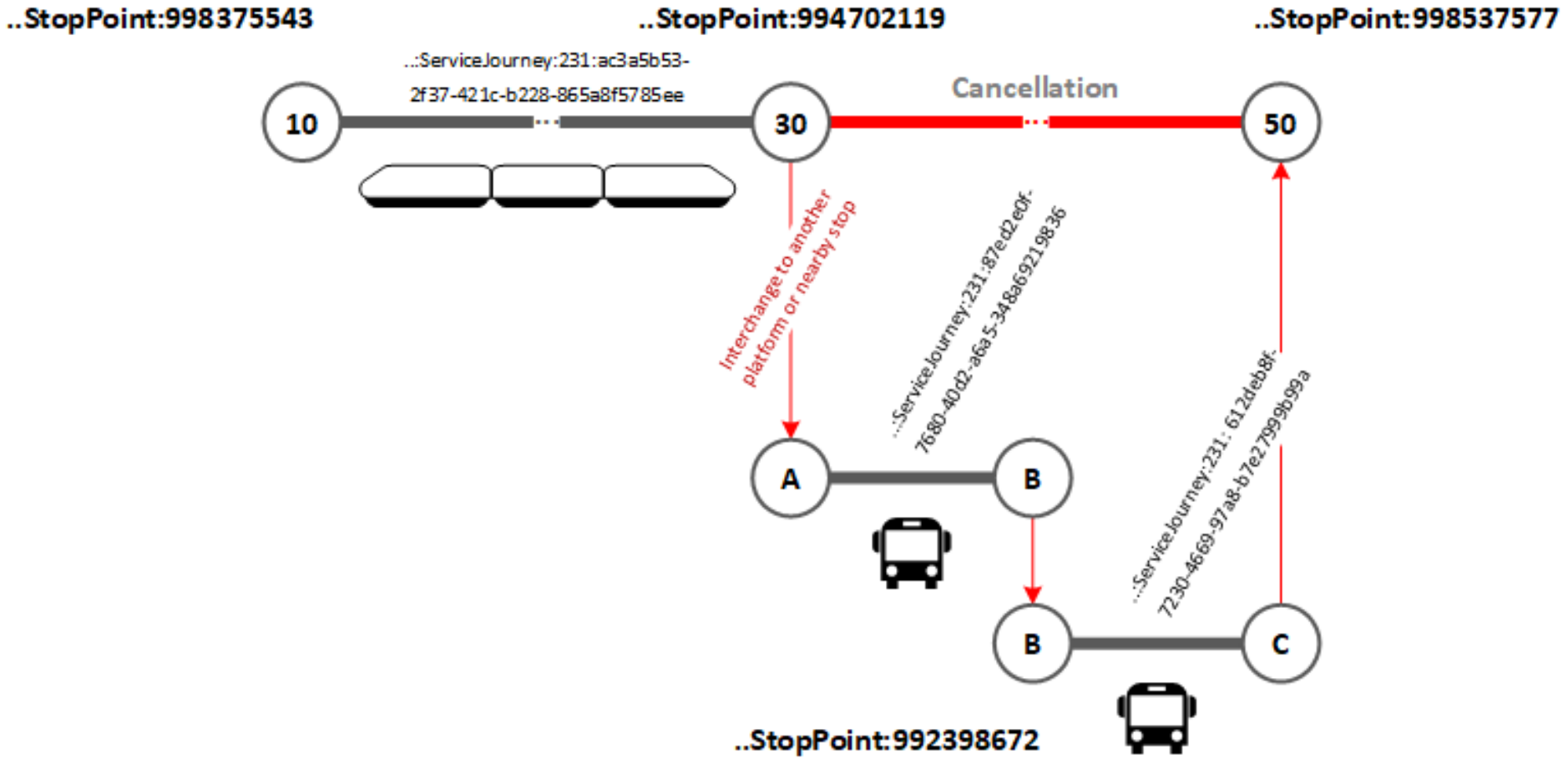


Figure 12 - Illustration of the journey relation example: due to a cancellation, passengers must transfer from the train to different replacement buses at call “30” and call “B”.

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

```
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <JourneyRelations>
      <JourneyRelation>
       <JourneyRelationType>ReplacedByJourney</JourneyRelationType>
       <JourneyParts>
        <JourneyPartInfo>
         <FromStopPointRef>ch:1:ScheduledStopPoint:994702119</FromStopPointRef>
         <ToStopPointRef>ch:1:ScheduledStopPoint:998537577</ToStopPointRef>
         <StartTime>2022-01-11T08:42:00Z</StartTime>
         <EndTime>2022-01-11T08:58:00Z</EndTime>
        </JourneyPartInfo>
       </JourneyParts>
       <RelatedJourney>
        <FramedVehicleJourneyRef>
         <DataFrameRef>2022-01-11</DataFrameRef>
         <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:87ed2e0f-7680-40d2-a6a5-348a69219836</DatedVehicleJourneyRef>
        </FramedVehicleJourneyRef>
        <JourneyParts>
         <JourneyPartInfo>
          <FromStopPointRef>ch:1:ScheduledStopPoint:994702119</FromStopPointRef>
```

```
          <ToStopPointRef>ch:1:ScheduledStopPoint:992398672</ToStopPointRef>
          <StartTime>2022-01-11T08:42:00Z</StartTime>
          <EndTime>2022-01-11T08:53:00Z</EndTime>
         </JourneyPartInfo>
        </JourneyParts>
        <!-- If the journeys are related as JourneyParts, the latter will generally be
             different from the perspective of the journey that relates to or is
             related by the other journey. In case journeys overlap or “meet” at a
             call and journey parts are irrelevant (or not used at all), the CallInfo
             will be used instead and generally be identical with the one from the
             JourneyRelation on level higher. -->
       <RelatedJourney/>
       <RelatedJourney>
        <FramedVehicleJourneyRef>
         <DataFrameRef>2022-01-11</DataFrameRef>
         <DatedVehicleJourneyRef>ch:1:ServiceJourney:231: 612deb8f-7230-4669-97a8-b7e27999b99a</DatedVehicleJourneyRef>
        </FramedVehicleJourneyRef>
        <JourneyParts>
         <JourneyPartInfo>
          <FromStopPointRef>ch:1:ScheduledStopPoint:992398672</FromStopPointRef>
          <ToStopPointRef>ch:1:ScheduledStopPoint:998537577</ToStopPointRef>
          <StartTime>2022-01-11T08:53:00Z</StartTime>
          <EndTime>2022-01-11T08:58:00Z</EndTime>
         </JourneyPartInfo>
        </JourneyParts>
       <RelatedJourney/>
      <JourneyRelation/>
     </JourneyRelations>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.17 Train formation update

LATER (SIRI 2.1)

10.17.1 Business

Incremental updates support: **not allowed**, i.e., always ***IsCompleteStopSequence*** = ‘true’ required

A data producer may indicate changes to the formation either on journey or call level:

a. If a ***FormationCondition*** is specified on journey level, the producer wants to indicate that changes to ***CompoundTrain***s are specified within ***JourneyPartInfo*** and thus apply to multiple calls. The producer might also want to indicate that the resulting customer information is shown to the end users at all stages of the journey.
b. A ***FormationCondition*** on journey level is useful in decoupling/splitting scenarios as depicted in the introductory example of this chapter where different parts of the journey have different destinations.

A ***FormationCondition*** on call level allows the producer to indicate exactly at which stops the formation changes happen. The producer might also want to indicate that the resulting customer information is shown to the end users at these specific stops of the journey.

Formation updates are usually not communicated for an already active journey but rather in the baseline message at the start of the operating day.

10.17.2 Example

Figure 14 illustrates the train formation for the first ***JourneyPart*** mapped to the platform of the first call at stop place «ch:1:ScheduledStop-Point:998375543» (see the ***DepartureFormationAssignments*** for the mapping). At call «ch:1:ScheduledStopPoint:994702119» the train with order 2 (marked blue in Figure 13 below) is decoupled and continues independently to Destination «B».

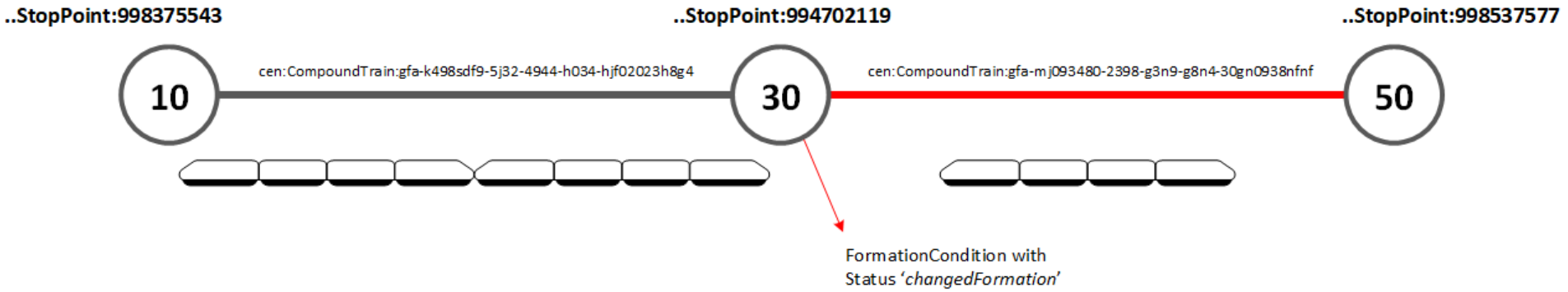


Figure 13 - Illustration of the formation example: train travels in shortened formation from call 30 onwards

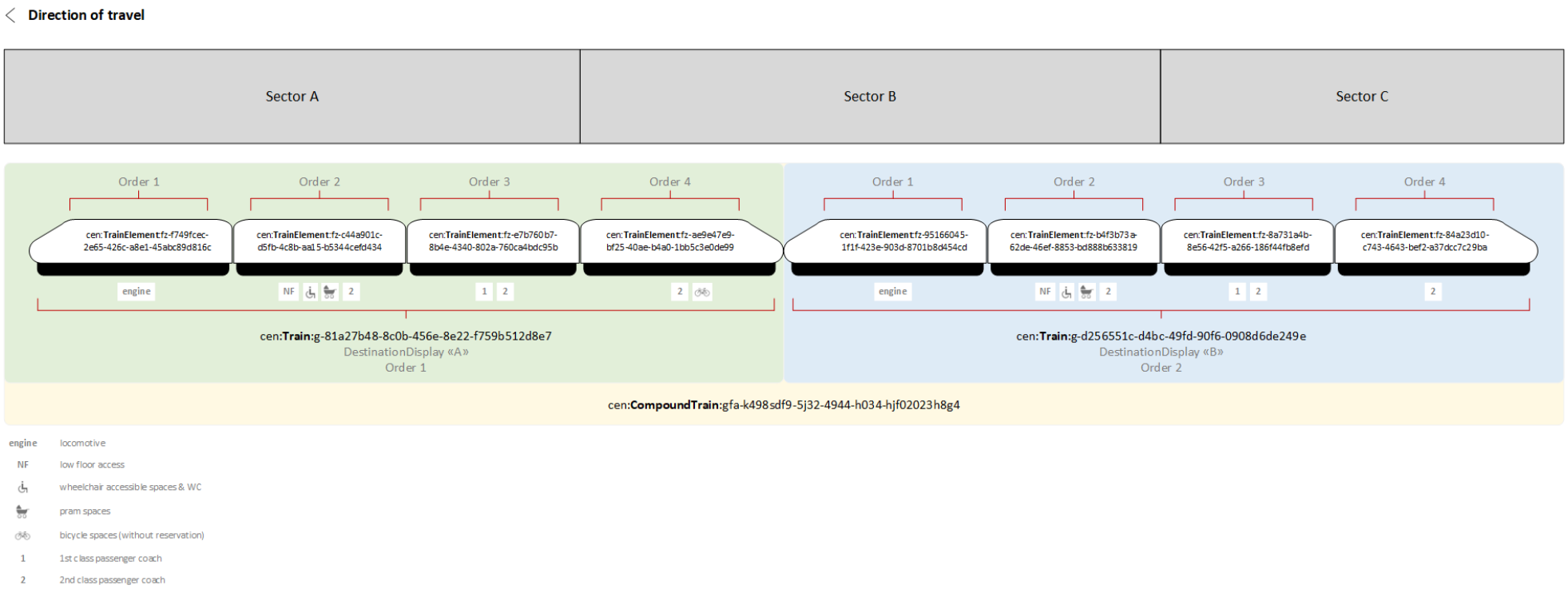


Figure 14 - Illustration of the train formation at the first call («..StopPoint:998375543» / Order «10») with the vehicle features and the coaches mapped to sections on platform 2 («..:Quay:76007252:11»)

Careful: even though an update message as a result of a formation change must always be a complete stop sequence, some information is omitted in the following example for the sake of simplicity.

```
<?xml version="1.0" encoding="utf-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
 <ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
```

```
<EstimatedTimetableDelivery>
 <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
 <SubscriptionRef>1506</SubscriptionRef>
 <EstimatedJourneyVersionFrame>
  <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
  <EstimatedVehicleJourney>
   <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
   <LineRef>ch:1:Line:231:S23</LineRef>
   <DirectionRef>2</DirectionRef>
   <FramedVehicleJourneyRef>
    <DataFrameRef>2022-01-11</DataFrameRef>
    <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
   </FramedVehicleJourneyRef>
   <FormationCondition>
    <FormationStatus>
     <Status>changedFormation</Status>
    </FormationStatus>
   </FormationCondition>
   <FormationCondition>
    <FormationStatus>
     <Status>missingVehicles</Status>
    </FormationStatus>
   </FormationCondition>
   <JourneyParts>
    <JourneyPartInfo>
     <CompoundTrainRef>ch:1:CompoundTrain:gfa-k498sdf9-5j32-4944-h034-hjf02023h8g4</CompoundTrainRef>
     <FromStopPointRef>ch:1:ScheduledStopPoint:998375543</FromStopPointRef>
     <ToStopPointRef>ch:1:ScheduledStopPoint:994702119</ToStopPointRef>
     <StartTime>2022-01-11T08:13:00Z</StartTime>
     <EndTime>2022-01-11T08:41:00Z</EndTime>
    </JourneyPartInfo>
    <JourneyPartInfo>
     <CompoundTrainRef>ch:1:CompoundTrain:gfa-mj093480-2398-g3n9-g8n4-30gn0938nfnf</CompoundTrainRef>
     <FromStopPointRef>ch:1:ScheduledStopPoint:994702119</FromStopPointRef>
     <ToStopPointRef>ch:1:ScheduledStopPoint:998537577</ToStopPointRef>
     <StartTime>2022-01-11T08:42:00Z</StartTime>
     <EndTime>2022-01-11T08:58:00Z</EndTime>
    </JourneyPartInfo>
```

```
      </JourneyParts>
      <TrainElements>
       <TrainElement>
        <TrainElementCode>ch:1:TrainElement:fz-f749fcec-2e65-426c-a8e1-45abc89d816c</TrainElementCode>
        <TrainElementType>engine</TrainElementType>
       </TrainElement>
       <TrainElement>
        <TrainElementCode>ch:1:TrainElement:fz-c44a901c-d5fb-4c8b-aa15-b5344cefd434</TrainElementCode>
        <TrainElementType>carriage</TrainElementType>
        <FareClasses>secondClass</FareClasses>
        <LowFloor>true</LowFloor>
        <Facilities>
         <VehicleFeature>
          <VehicleFeatureCode>ch:1:VehicleFeature:a-66f73888-0357-4bd0-bcdc-b3c28d4cc4ca</VehicleFeatureCode>
          <Name>strollerArea</Name>
          <Icon/>
         </VehicleFeature>
         <VehicleFeature>
          <VehicleFeatureCode>ch:1:VehicleFeature:a-1829903d-5610-4c6f-b772-745299a487f3</VehicleFeatureCode>
          <Name>wheelchairArea</Name>
          <Icon/>
         </VehicleFeature>
        </Facilities>
       </TrainElement>
       <TrainElement>
        <TrainElementCode>ch:1:TrainElement:fz-e7b760b7-8b4e-4340-802a-760ca4bdc95b</TrainElementCode>
        <TrainElementType>carriage</TrainElementType>
        <FareClasses>secondClass firstClass</FareClasses>
       </TrainElement>
       <TrainElement>
        <TrainElementCode>ch:1:TrainElement:fz-ae9e47e9-bf25-40ae-b4a0-1bb5c3e0de99</TrainElementCode>
        <TrainElementType>carriage</TrainElementType>
        <FareClasses>secondClass</FareClasses>
       </TrainElement>
       <TrainElement>
        <TrainElementCode>ch:1:TrainElement:fz-95166045-1f1f-423e-903d-8701b8d454cd</TrainElementCode>
        <TrainElementType>engine</TrainElementType>
       </TrainElement>
```

```
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-b4f3b73a-62de-46ef-8853-bd888b633819</TrainElementCode>
       <TrainElementType>carriage</TrainElementType>
       <FareClasses>secondClass</FareClasses>
       <LowFloor>true</LowFloor>
       <Facilities>
        <VehicleFeature>
         <VehicleFeatureCode>ch:1:VehicleFeature:a-80223747-bc4b-49c5-808f-8ca3fdbdaabf</VehicleFeatureCode>
         <Name>strollerArea</Name>
         <Icon/>
        </VehicleFeature>
        <VehicleFeature>
         <VehicleFeatureCode>ch:1:VehicleFeature:a-9c69d40e-16af-41eb-8ff0-dc41d0d82d2f</VehicleFeatureCode>
         <Name>wheelchairArea</Name>
         <Icon/>
        </VehicleFeature>
       </Facilities>
      </TrainElement>
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-8a731a4b-8e56-42f5-a266-186f44fb8efd</TrainElementCode>
       <TrainElementType>carriage</TrainElementType>
       <FareClasses>secondClass firstClass</FareClasses>
      </TrainElement>
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-84a23d10-c743-4643-bef2-a37dcc7c29ba</TrainElementCode>
       <TrainElementType>carriage</TrainElementType>
       <FareClasses>secondClass</FareClasses>
      </TrainElement>
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-b2915517-fc3c-4b0a-9975-14fecfaf41f6</TrainElementCode>
       <TrainElementType>engine</TrainElementType>
      </TrainElement>
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-e28932a2-6a37-4d8b-b0fe-f43a84d825f2</TrainElementCode>
       <TrainElementType>carriage</TrainElementType>
       <FareClasses>secondClass</FareClasses>
       <LowFloor>true</LowFloor>
       <Facilities>
```

```
        <VehicleFeature>
         <VehicleFeatureCode>ch:1:VehicleFeature:a-a8bb7529-0708-411c-bd5f-dcb3655c6eb2</VehicleFeatureCode>
         <Name>strollerArea</Name>
         <Icon/>
        </VehicleFeature>
        <VehicleFeature>
         <VehicleFeatureCode>ch:1:VehicleFeature:a-d150f1dd-8db3-476d-ae3a-8bce8fac51f1</VehicleFeatureCode>
         <Name>wheelchairArea</Name>
         <Icon/>
        </VehicleFeature>
       </Facilities>
      </TrainElement>
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-8dfc603f-20aa-4470-97fd-833451b0dfb7</TrainElementCode>
       <TrainElementType>carriage</TrainElementType>
       <FareClasses>secondClass firstClass</FareClasses>
      </TrainElement>
      <TrainElement>
       <TrainElementCode>ch:1:TrainElement:fz-ef2ee6e8-2154-49ac-bed0-c660c38c37ce</TrainElementCode>
       <TrainElementType>carriage</TrainElementType>
       <FareClasses>secondClass</FareClasses>
      </TrainElement>
     </TrainElements>
     <Trains>
      <Train>
       <TrainCode>ch:1:Train:g-81a27b48-8c0b-456e-8e22-f759b512d8e7</TrainCode>
       <TrainComponents>
        <TrainComponent>
         <Order>1</Order>
         <TrainElementRef>ch:1:TrainElement:fz-f749fcec-2e65-426c-a8e1-45abc89d816c</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>2</Order>
         <TrainElementRef>ch:1:TrainElement:fz-c44a901c-d5fb-4c8b-aa15-b5344cefd434</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>3</Order>
         <TrainElementRef>ch:1:TrainElement:fz-e7b760b7-8b4e-4340-802a-760ca4bdc95b</TrainElementRef>
```

```
        </TrainComponent>
        <TrainComponent>
         <Order>4</Order>
         <TrainElementRef>ch:1:TrainElement:fz-ae9e47e9-bf25-40ae-b4a0-1bb5c3e0de99</TrainElementRef>
        </TrainComponent>
       </TrainComponents>
      </Train>
      <Train>
       <TrainCode>ch:1:Train:g-d256551c-d4bc-49fd-90f6-0908d6de249e</TrainCode>
       <Components>
        <TrainComponent>
         <Order>1</Order>
         <TrainElementRef>ch:1:TrainElement:fz-95166045-1f1f-423e-903d-8701b8d454cd</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>2</Order>
         <TrainElementRef>ch:1:TrainElement:fz-b4f3b73a-62de-46ef-8853-bd888b633819</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>3</Order>
         <TrainElementRef>ch:1:TrainElement:fz-8a731a4b-8e56-42f5-a266-186f44fb8efd</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>4</Order>
         <TrainElementRef>ch:1:TrainElement:fz-84a23d10-c743-4643-bef2-a37dcc7c29ba</TrainElementRef>
        </TrainComponent>
       </Components>
      </Train>
      <Train>
       <TrainCode>ch:1:Train:g-a219f577-1263-48eb-8e66-d6632e32f7f7</TrainCode>
       <TrainComponents>
        <TrainComponent>
         <Order>1</Order>
         <TrainElementRef>ch:1:TrainElement:fz-b2915517-fc3c-4b0a-9975-14fecfaf41f6</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>2</Order>
         <TrainElementRef>ch:1:TrainElement:fz-e28932a2-6a37-4d8b-b0fe-f43a84d825f2</TrainElementRef>
```

```
        </TrainComponent>
        <TrainComponent>
         <Order>3</Order>
         <TrainElementRef>ch:1:TrainElement:fz-8dfc603f-20aa-4470-97fd-833451b0dfb7</TrainElementRef>
        </TrainComponent>
        <TrainComponent>
         <Order>4</Order>
         <TrainElementRef>ch:1:TrainElement:fz-ef2ee6e8-2154-49ac-bed0-c660c38c37ce</TrainElementRef>
        </TrainComponent>
       </TrainComponents>
      </Train>
     </Trains>
     <CompoundTrains>
      <CompoundTrain>
       <CompoundTrainCode>ch:1:CompoundTrain:gfa-k498sdf9-5j32-4944-h034-hjf02023h8g4</CompoundTrainCode>
       <Components>
        <TrainInCompoundTrain>
         <Order>1</Order>
         <TrainRef>ch:1:Train:g-81a27b48-8c0b-456e-8e22-f759b512d8e7</TrainRef>
         <DestinationDisplayAtOrigin>A</DestinationDisplayAtOrigin>
        </TrainInCompoundTrain>
        <TrainInCompoundTrain>
         <Order>2</Order>
         <TrainRef>ch:1:Train:g-d256551c-d4bc-49fd-90f6-0908d6de249e</TrainRef>
         <DestinationDisplayAtOrigin>B</DestinationDisplayAtOrigin>
        </TrainInCompoundTrain>
       </Components>
      </CompoundTrain>
      <CompoundTrain>
       <CompoundTrainCode>ch:1:CompoundTrain:gfa-mj093480-2398-g3n9-g8n4-30gn0938nfnf</CompoundTrainCode>
       <Components>
        <TrainInCompoundTrain>
         <Order>2</Order>
         <TrainRef>ch:1:Train:g-a219f577-1263-48eb-8e66-d6632e32f7f7</TrainRef>
         <DestinationDisplayAtOrigin>A</DestinationDisplayAtOrigin>
        </TrainInCompoundTrain>
       </Components>
      </CompoundTrain>
```

```
      </CompoundTrains>
      <EstimatedCalls>
       <EstimatedCall>
        <StopPointRef>ch:1:ScheduledStopPoint:998375543</StopPointRef>
        <Order>10</Order>
        <Occupancy>manySeatsAvailable</Occupancy>
        <AimedDepartureTime>2022-01-11T08:13:00Z</AimedDepartureTime>
        <ExpectedDepartureTime>2022-01-11T08:13:00Z</ExpectedDepartureTime>
        <DepartureStatus>onTime</DepartureStatus>
        <DeparturePlatformName>2</DeparturePlatformName>
        <DepartureBoardingActivity>boarding</DepartureBoardingActivity>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11</AimedQuayRef>
         <AimedQuayName>2</AimedQuayName>
         <QuayType>platform</QuayType>
        </DepartureStopAssignment>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:A</AimedQuayRef>
         <AimedQuayName>A</AimedQuayName>
         <QuayType>platformSector</QuayType>
        </DepartureStopAssignment>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:B</AimedQuayRef>
         <AimedQuayName>B</AimedQuayName>
         <QuayType>platformSector</QuayType>
        </DepartureStopAssignment>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:C</AimedQuayRef>
         <AimedQuayName>C</AimedQuayName>
         <QuayType>platformSector</QuayType>
        </DepartureStopAssignment>
        <DepartureStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:D</AimedQuayRef>
         <AimedQuayName>D</AimedQuayName>
         <QuayType>platformSector</QuayType>
        </DepartureStopAssignment>
        <DepartureFormationAssignment>
         <TrainComponentRef>ch:1:TrainElement:fz-f749fcec-2e65-426c-a8e1-45abc89d816c</TrainComponentRef>
```

```
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:A</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-c44a901c-d5fb-4c8b-aa15-b5344cefd434</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:A</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-e7b760b7-8b4e-4340-802a-760ca4bdc95b</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:A</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-ae9e47e9-bf25-40ae-b4a0-1bb5c3e0de99</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:B</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-95166045-1f1f-423e-903d-8701b8d454cd</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
```

```
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:B</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-b4f3b73a-62de-46ef-8853-bd888b633819</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:B</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-8a731a4b-8e56-42f5-a266-186f44fb8efd</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:C</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureFormationAssignment>
        <TrainComponentRef>ch:1:TrainElement:fz-84a23d10-c743-4643-bef2-a37dcc7c29ba</TrainComponentRef>
        <VehicleStatus>
         <Status>open</Status>
        </VehicleStatus>
        <TrainStopAssignment>
         <AimedQuayRef>ch:1:Quay:76007252:11:C</AimedQuayRef>
        </TrainStopAssignment>
       </DepartureFormationAssignment>
       <DepartureOrientationRelativeToQuay>&lt;= A-B-C-D</DepartureOrientationRelativeToQuay>
      </EstimatedCall>
      <!-- Subsequent calls and formation data is omitted for the sake of simplicity. -->
     </EstimatedCalls>
     <IsCompleteStopSequence>true</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
```

```
    </EstimatedJourneyVersionFrame>
   </EstimatedTimetableDelivery>
  </ServiceDelivery>
</Siri>
```

10.18 Occupancy update

LATER (SIRI 2.1)

10.18.1 Business

Incremental updates support: ***IsCompleteStopSequence*** = ‘false’ **allowed**

10.18.2 Example

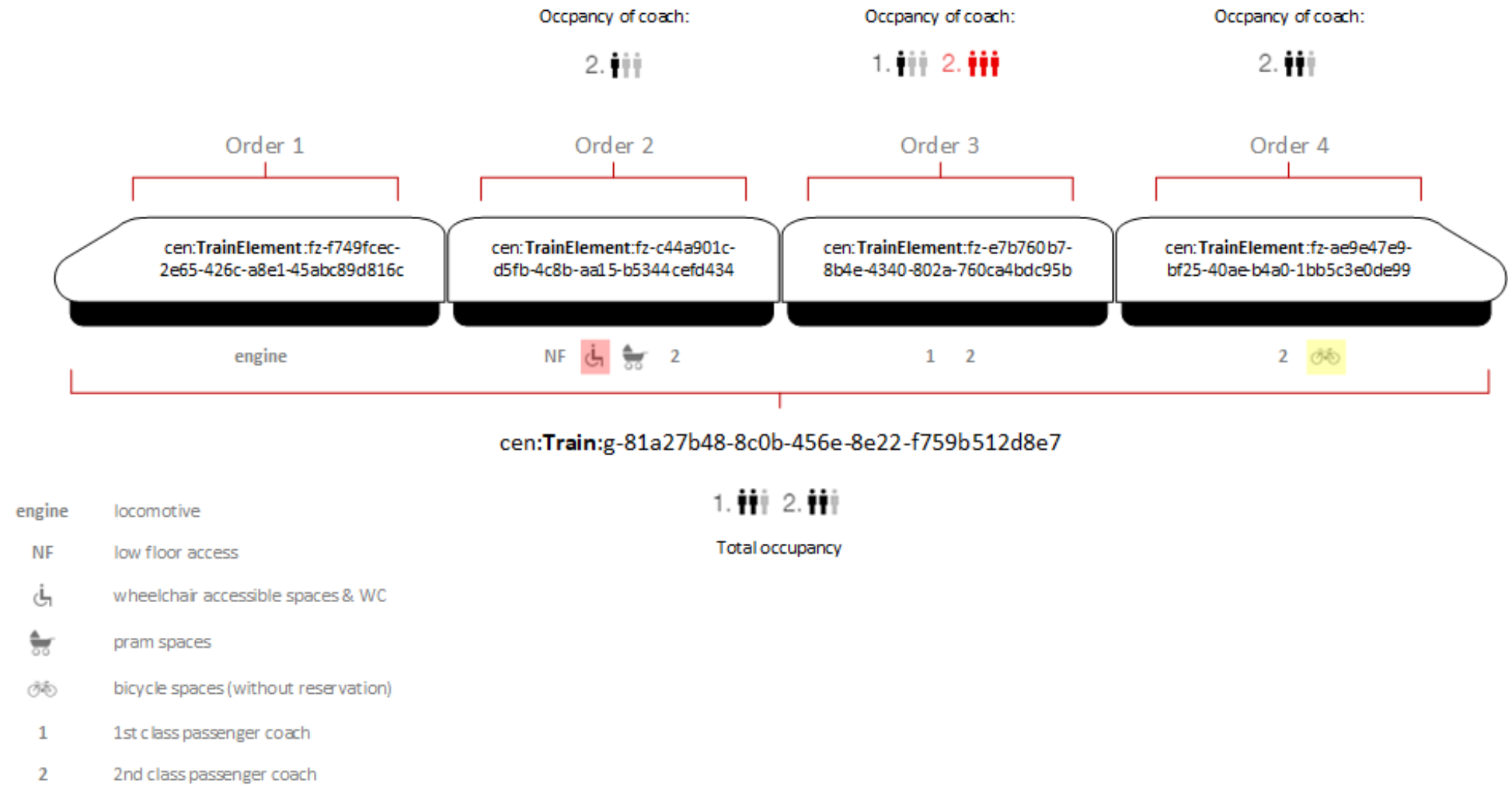


Figure 15 - Illustration of the occupancy examples: coach 2 cannot accept wheelchair passengers anymore, the 2nd class sections of coach 3 are packed and coach 4 has limited capability for bicycles.

10.18.2.1 Global occupancy measurement

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <RecordedCalls>
      <RecordedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <Order>40</Order>
       <RecordedDepartureOccupancy>
        <OccupancyLevel>fewSeatsAvailable</OccupancyLevel>
        <OccupancyPercentage>73</OccupancyPercentage>
        <WheelchairOnboardCount>1</WheelchairOnboardCount>
        <PramsOnboardCount>1</PramsOnboardCount>
        <BicycleOnboardCount>2</BicycleOnboardCount>
       </RecordedDepartureOccupancy>
      </RecordedCall>
     </RecordedCalls>
     <IsCompleteStopSequence>false</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
```

```
 </ServiceDelivery>
</Siri>
```

10.18.2.2 Occupancy measurement per coach

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <RecordedCalls>
      <RecordedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <Order>40</Order>
       <RecordedDepartureOccupancy>
        <TrainRef>ch:1:Train:g-81a27b48-8c0b-456e-8e22-f759b512d8e7</TrainRef>
        <TrainElementRef>ch:1:TrainElement:fz-c44a901c-d5fb-4c8b-aa15-b5344cefd434</TrainElementRef>
        <OccupancyLevel>manySeatsAvailable</OccupancyLevel>
        <OccupancyPercentage>11</OccupancyPercentage>
        <WheelchairOnboardCount>2</WheelchairOnboardCount>
        <PramsOnboardCount>1</PramsOnboardCount>
        <BicycleOnboardCount>0</BicycleOnboardCount>
       </RecordedDepartureOccupancy>
```

```
       <RecordedDepartureOccupancy>
        <TrainRef>ch:1:Train:g-81a27b48-8c0b-456e-8e22-f759b512d8e7</TrainRef>
        <TrainElementRef>ch:1:TrainElement:fz-ae9e47e9-bf25-40ae-b4a0-1bb5c3e0de99</TrainElementRef>
        <OccupancyLevel>fewSeatsAvailable</OccupancyLevel>
        <OccupancyPercentage>56</OccupancyPercentage>
        <WheelchairOnboardCount>0</WheelchairOnboardCount>
        <PramsOnboardCount>0</PramsOnboardCount>
        <BicycleOnboardCount>3</BicycleOnboardCount>
       </RecordedDepartureOccupancy>
      </RecordedCall>
     </RecordedCalls>
     <IsCompleteStopSequence>false</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.18.2.3 Group reservation

```
<?xml version="1.0" encoding="UTF-8"
<Siri xmlns="http://www.siri.org.uk/siri" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.1" xsi:schemaLoca-
tion="http://www.siri.org.uk/siri path/to/schema/siri.xsd">
<ServiceDelivery>
  <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
  <ProducerRef>cen-out-et_prod</ProducerRef>
  <RequestMessageRef>0fdf2bbc-d18a-40ba-a0cb-b2e6d0ff6931</RequestMessageRef>
  <EstimatedTimetableDelivery>
   <ResponseTimestamp>2022-01-11T08:11:46Z</ResponseTimestamp>
   <SubscriptionRef>1506</SubscriptionRef>
   <EstimatedJourneyVersionFrame>
    <RecordedAtTime>2022-01-11T08:11:44Z</RecordedAtTime>
    <EstimatedVehicleJourney>
     <RecordedAtTime>2022-01-11T08:11:43Z</RecordedAtTime>
     <LineRef>ch:1:Line:231:S23</LineRef>
     <DirectionRef>2</DirectionRef>
     <FramedVehicleJourneyRef>
      <DataFrameRef>2022-01-11</DataFrameRef>
```

```
      <DatedVehicleJourneyRef>ch:1:ServiceJourney:231:ac3a5b53-2f37-421c-b228-865a8f5785ee</DatedVehicleJourneyRef>
     </FramedVehicleJourneyRef>
     <EstimatedCalls>
      <EstimatedCall>
       <StopPointRef>ch:1:ScheduledStopPoint:991128574</StopPointRef>
       <Order>40</Order>
       <ExpectedDepartureOccupancy>
        <TrainRef>ch:1:Train:g-81a27b48-8c0b-456e-8e22-f759b512d8e7</TrainRef>
        <TrainElementRef>ch:1:TrainElement:fz-e7b760b7-8b4e-4340-802a-760ca4bdc95b</TrainElementRef>
        <FareClass>secondClass</FareClass>
        <TotalNumberOfReservedSeats>21</TotalNumberOfReservedSeats>
        <GroupReservation>
         <NameOfGroup>Hjorten omsorgssenter</NameOfGroup>
         <NumberOfReservedSeats>17</NumberOfReservedSeats>
        </GroupReservation>
       </ExpectedDepartureOccupancy>
      </EstimatedCall>
     </EstimatedCalls>
     <IsCompleteStopSequence>false</IsCompleteStopSequence>
    </EstimatedVehicleJourney>
   </EstimatedJourneyVersionFrame>
  </EstimatedTimetableDelivery>
 </ServiceDelivery>
</Siri>
```

10.19 Line timetable over production timetable

10.19.1 Business

See the two options introduced in chapter 6.1.

10.19.2 Example

The ***DatedVehicleJourney*** data itself is almost identical to the ***EstimatedVehicleJourney*** one, hence no explicit example other than what is provided in the referenced chapter above.

10.20 Situation affecting line (TODO)

10.20.1 Business

10.20.2 Example

10.21 Situation affecting stop place (TODO)

10.21.1 Business

10.21.2 Example

10.22 Situation affecting elevator and impacting accessibility (TODO)

LATER

10.22.1 Business

10.22.2 Example