

SUTI Use Cases



TABLE of CONTENT

1.	<i>Use Cases general Description and Use</i>	8
1.1	How to use Use Cases in SUTI	8
1.2	An introduction to SUTI terms and relations between terms	8
1.2.1	Introduction	8
1.2.2	Message, Provider, Client, Request, Response and Flow	8
1.2.3	Who can be Client and who can be Provider? What is a Self Declaration in SUTI?	8
1.2.4	SUTI BLOCKS (Sections)	8
1.2.5	Route, Node, Content, Passenger, Parcel, Pickup, Destination and Trip	9
1.2.6	Order, SubOrder, Repetitive Order, OrderTemplate, DriverSession, Tour, Agreement	9
1.2.7	Events	9
1.2.8	Time	10
1.2.9	Address, Location and Geographic Location	10
1.2.10	Content (Passenger, Parcel) demands, Attributes, Connection	10
1.2.11	Resource, Driver, Vehicle, Attributee	10
1.2.12	Economy, Price, Taximeter, FormOfPayment, Cash, Account, SocialServiceFee	10
1.3	Terms outside the use in SUTI	11
1.3.1	Booking, Segment, Reservation	11
1.3.2	Fleet, Settlement (Dispatch)	11
1.4	Techniques used for describing SUTI (XML and JSON)	11
1.4.1	Introduction	11
1.4.2	Differences and similarities between XML and JSON	11
1.4.2.1	Similarities between XML and JSON	12
1.4.3	Differences	12
1.5	Communication alternatives for SUTI	12
1.5.1	SUTI Communication using HTTP Post on each side.	13
1.6	SUTI Relation to GDPR	14
2.	<i>Use cases</i>	15
2.1	Order	15
2.1.1	Order flow	15
2.1.1.1	Basic flow	15
2.1.1.2	Typical flow	17
2.1.1.3	Extensive flow (Node by Node)	19
2.1.1.4	Extensive flow with traffic control (Node by Node)	21
2.1.1.5	Multiple client order combined into a shared provider order	23
2.1.1.6	Order by Order	25
2.1.1.6.1	General	25
2.1.1.7	driverSession	27
2.1.1.7.1	General	27
2.1.1.7.2	Time lines	29
2.1.1.8	Order Trading	31
2.1.1.9	PreeOrder with preeOrder flag "PROVIDER"	33
2.1.1.10	PreeOrder with preeOrder flag "CLIENTCONFIRM"	34
2.1.1.11	Order Forwarded	35
2.1.2	Additional processes	37
2.1.2.1	Order status	37
2.1.2.2	Alteration in dispatch with dispatch reject	38
2.1.2.3	Alteration in dispatch with new dispatch proposal	39
2.1.2.4	Resource allocation	40
2.1.2.4.1	The requested resource is not available	40
2.1.2.4.2	The requested resource is available	40
2.1.2.5	Use Case request a fixed price and how to place an order	41
2.1.2.6	General	41

2.1.3	Order events Client	44
2.1.3.1	Order/node cancellation	44
2.1.3.1.1	Order/node cancelation accept	44
2.1.3.1.2	Order/node cancellation accepted with consequence	44
2.1.3.1.3	Order/node cancellation reject	44
2.1.3.2	Order alteration	44
2.1.3.3	Late order	46
2.1.3.3.1	The hard way orderReject (Client is responsible for the order)	46
2.1.3.3.2	The softer way pickup confirmation (Provider is responsible for the order)	46
2.1.4	Order events Provider	47
2.1.4.1	Order reject	47
2.1.4.2	Order reject request	47
2.1.4.3	Node events	48
2.1.4.3.1	Message flow for node events	48
2.1.4.4	Request for additional or corrected node information	48
2.1.5	Order by Order and driverSession special processes	50
2.1.5.1	Resource description	50
2.1.5.2	Adding new order	50
2.1.5.3	Change of orders	50
2.2	Repetitive Orders	51
2.2.1	SUTI message OrderTemplate 2800	51
2.2.2	SUTI message OrderTemplateConfirmation 2801	54
2.2.3	SUTI message ScheduleElementConfirmation 2810	56
2.2.4	SUTI message ConfirmationOfRecievedMessage 7099	57
2.2.5	SUTI message ScheduleElementConfirmation 2810	58
2.2.6	SUTI message ReStart 7021	59
2.2.7	SUTI message RequestForTrafficInformation 4000	60
2.2.8	SUTI message RequestedTrafficInformation 4001	61
2.2.9	SUTI message Pickup Confirmation 4010	64
2.2.10	SUTI message Order reject request 2005	65
2.2.11	SUTI message Orderstatus 2531	66
2.2.12	SUTI message RequestForOrderInformation 6810	68
2.2.13	SUTI message RequestedOrderInformation 6800	69
2.2.14	SUTI message RequestGeograhpyinformation 1050	73
2.2.15	SUTI message TransferGeograhpyinformation 1051	73
2.2.16	SUTI message Keepalive 7000	73
2.2.17	SUTI message Keepalive 7001	74
2.2.18	SUTI message RequestForOrderInfo 2540	75
2.2.19	SUTI message OrderInfo 2541	78
2.2.20	Flowcharts	80
2.2.21	Ej implementerade delar	83
2.3	SUTI flagstops	84
2.3.1	Flagstop forløb	84
2.3.2	Flagstops i SUTI	84
2.3.2.1	Authorization	84
2.3.2.2	Bookingen	84
2.4	Delivery Note and associated messages	86
2.4.1	How to request a Msg 6500 Delivery Note	86
2.4.1.1	Request for Msg 6500 by using process element in a Msg 2000	86
2.4.1.2	Request for Msg 6500 by using process element in a Msg 2901	86
2.4.1.3	Request for Msg 6500 by sending a Msg 6510	86
2.4.1.4	Responding to a Msg 6510 with a Msg 6511	87
2.4.2	Sending a Msg 6500 Delivery Note	87
2.4.3	Answering a Msg 6500	88
2.4.3.1	Msg 6501 deliveryNoteAccept	88
2.4.3.2	Msg 6502 deliveryNoteReject	89
2.4.3.3	Msg 6503 deliveryNoteWait	89
2.5	Accounting	94
2.5.1	Basic Accounting from Provider to Client	94
2.5.1.1	General	94

2.5.1.2	Flow chart	95
2.5.1.3	Information content	95
2.5.2	Direct accounting from client to provider	99
2.5.2.1	General	99
2.5.2.2	Flow chart	99
2.6	Exchanging resource information	100
2.6.1	General	100
2.6.2	Resource request and response	100
2.6.3	bulkLocation request and response	101
3.	Use of information elements	102
3.1	General descriptions	102
3.1.1	General use of information elements	102
3.1.2	ID (Identification elements) Structures and Usage in SUTI	102
3.1.2.1	Background	102
3.1.2.3	Rules for usage of attribute source	103
3.1.2.5	SUTI Id Description	104
3.1.3	Resource description	105
3.1.3.1	GENERAL	105
3.1.3.1.1	VEHICLE GENERAL	105
3.1.3.1.2	DRIVER GENERAL	105
3.1.3.2	SEATING PROPERTIES in VEHICLE	106
3.1.3.2.1	DIRECTION	106
3.1.3.2.2	ACCESS	106
3.1.3.2.3	LEGSPACE	106
3.1.3.3	SEATING POSITION in VEHICLE	107
3.1.3.3.1	ROW	107
3.1.3.3.2	PLACE IN A ROW	107
3.1.3.4	ATTRIBUTES on VEHICLE	108
3.1.3.5	ATTRIBUTES on DRIVER	108
3.1.3.6	EXAMPLE of CONFIGURATION	109
3.1.3.6.1	NORMAL CONFIGURATION	109
3.1.3.6.2	ALTERNATIVE CONFIGURATION	110
3.1.3.6.3	SPECIAL CONFIGURATION STRETCHER	111
3.1.3.6.4	SPECIAL CONFIGURATION WHEELCHAIR	112
3.1.4	Content	113
3.1.5	Addresses (addressNode)	113
3.1.5.1	Minimum requirements	113
3.1.5.2	Attributes in addressType	113
3.1.5.3	Elements in addressType	114
3.1.6	Geographical location (geographicLocation)	114
3.1.6.1	Resolution WGS-84 decimal degrees	114
3.1.6.2	Attributes and elements in geographicLocation	114
3.1.6.3	Minimum requirements geographicLocation	115
3.1.7	Economy	115
3.2	List of Information elements	116
3.2.1	Agreement	116
3.2.1.1	General	116
3.2.1.2	Minimum requirements	116
3.2.2	idMsg	116
3.2.2.1	General	116
3.2.2.2	Minimum requirements	116
3.2.3	idOrder	116
3.2.3.1	General	116
3.2.3.2	Minimum requirements	116
3.2.4	infoTimeStamp	116
3.2.4.1	General	116
3.2.4.2	Minimum requirements	117
3.2.5	msgTimeStamp	117
3.2.5.1	General	117

3.2.5.2	Minimum requirements	117
3.2.6	orgReceiver	117
3.2.6.1	General	117
3.2.6.2	Minimum requirements	117
3.2.7	orgSender	117
3.2.7.1	General	117
3.2.7.2	Minimum requirements	117
3.2.8	referencesTo	117
3.2.8.1	General	117
3.2.8.2	Minimum requirements	117
3.2.9	Event types	118
3.2.9.1	General	118
3.2.9.2	Minimum requirements	118
3.2.9.2.1	Event Type: 1701 invehicle (alt passengerinvehicle)	118
3.2.9.2.2	Event Type: 1702 exitvehicle (alt passengerdropped)	118
3.2.9.2.3	Event Type: 1703 noshow	119
3.2.9.2.4	Event Type: 1704 parcelinvehicle	119
3.2.9.2.5	Event Type: 1705 parceldropped	119
3.2.9.2.6	Event Type: 1706 actiondone	119
3.2.9.2.7	Event Type: 1707 navigationdone	119
3.2.9.2.8	Event Type: 1708 cancelatdoor	119
3.2.9.2.9	Event Type: 1709 vehicleatnode	119
3.2.9.2.10	Event Type: 1710 infotocontent	120
3.2.9.2.11	Event Type: 1711 dispatchconfirmationsent	120
3.2.9.2.12	Event Type: 1712 delayconfirmationsent	120
3.2.9.2.13	Event Type: 1713 arrivalconfirmationsent	120
3.2.9.2.14	Event Type: 1714 start	121
3.2.9.2.15	Event Type: 1715 stop	121
3.2.9.2.16	Event Type: 1716 acceptOrder	121
3.2.9.2.17	Event Type: 1717 vehicleExitNode	121
3.2.9.2.18	Event Type: 1718 contentNoshow	121
3.2.9.2.19	Event Type: 1719 contentAtNode	122
3.2.10	idPayment/idEkInfo	123
3.2.11	locationList	123
3.2.12	Ratings	123
3.2.13	Order Status	124
3.2.14	idAction	125

4. Examples of Self Declaration Flows **126**

4.1 Objectives of this chapter **126**

4.2 Order Flows **127**

4.2.1	Trip Order Flow	127
4.2.1.1	Supported Messages	127
4.2.1.2	Flow order	128
4.2.1.2.1	Dispatch	128
4.2.1.2.2	Dispatch using "anbudsområde"	129
4.2.1.3	Messages	129
4.2.1.3.1	OrgSender and orgReceiver nodes	129
4.2.1.3.2	MSG 2000 Order	129
4.2.1.3.3	MSG 2001 Order confirmation	131
4.2.1.3.4	MSG 2005 Order Reject Request	132
4.2.1.3.5	MSG 2006 Order Reject Request Accepted	132
4.2.1.3.6	MSG 2007 Order Reject Request Reject	132
4.2.1.3.7	MSG 2010 Order Cancellation Request	132
4.2.1.3.8	MSG 2011 Order cancellation accepted	133
4.2.2	Node by Node Flow	134
4.2.2.1	Flow of start of new circulation:	134
4.2.2.2	Flow provider wants next node:	134
4.2.2.3	Messages	135

4.2.2.3.1	orgSender and orgReceiver nodes	135
4.2.2.3.2	MSG 2000 Order	135
4.2.2.3.3	MSG 2001 Order confirmation	137
4.2.2.3.4	MSG 2010 Order Cancellation Request	137
4.2.2.3.5	MSG 2011 Order cancellation accepted	137
4.2.2.3.6	MSG 3003 Dispatch confirmation	137
4.2.2.3.7	MSG 4000 Request for traffic information	138
4.2.2.3.8	MSG 4001 Requested traffic information	139
4.2.3	Order by Order Flow	141
4.2.3.1	Flow order	141
4.2.3.2	Dispatch	141
4.2.3.2.1	Provider cancelation	141
4.2.3.2.2	Order alteration	141
4.2.3.3	Dispatch using "anbudsområde"	141
4.2.3.3.1	Provider cancelation	142
4.2.3.3.2	Order alteration	142
4.2.3.4	Messages	142
4.2.3.4.1	OrgSender and orgReceiver nodes	142
4.2.3.4.2	MSG 2000 Order	143
4.2.3.4.3	MSG 2000 Order alteration	146
4.2.3.4.4	MSG 2001 Order confirmation	148
4.2.3.4.5	MSG 2005 Order Reject Request	148
4.2.3.4.6	MSG 2006 Order Reject Request Accepted	148
4.2.3.4.7	MSG 2007 Order Reject Request Reject	148
4.2.3.4.8	MSG 2010 Order Cancellation Request	149
4.2.3.4.9	MSG 2011 Order cancellation accepted	149
4.2.4	Session Order Flow	149
4.3	Assignment of vehicle flows (Dispatch)	150
4.3.1	Normal Flow	150
4.3.2	Irregularity Flow 1: The suggested driver may not accept the order.	150
4.3.3	Irregularity Flow 2: Late Dispatching from Provider.	151
4.4	Monitoring Flows (Traffic Control)	152
4.4.1	Prerequisites general	152
4.4.2	Monitor Flow Pick up at a simple address for a preebooked passenger (Standard)	153
4.4.2.1	Estimation of time for vehicle at node (Stage 1)	153
4.4.2.1.1	Normal Flow	153
4.4.2.1.2	Irregularity Flow 1: The Client has due to vehicle position estimated a late Passenger pickup	153
4.4.2.2	Vehicle at node (Stage 2)	154
4.4.2.2.1	Normal Flow	154
4.4.2.2.2	Irregularity Flow 1: Client decides not to accept vehicle at node	154
4.4.2.2.3	Irregularity Flow 2: The passenger has notified the Client that he/she is at node	155
4.4.2.2.4	Irregularity Flow 3: The driver notifies that vehicle is not at the correct pick-up address.	155
4.4.2.3	Passenger in vehicle (Stage 3)	156
4.4.2.3.1	Normal Flow	156
4.4.2.3.2	Irregularity Flow 1: Client does not accept the identification of the passenger	156
4.4.2.3.3	Irregularity Flow 2: Client has not got a Vehicle at node prior to Passenger in vehicle	156
4.4.2.3.4	Irregularity Flow 3: The driver signals that no content is at node	156
4.4.2.3.5	Irregularity 4: The passenger signals that no vehicle is at address	159
4.4.2.3.6	Irregularity 5: The passenger or client decides to interrupt node	159
4.5	Technical Flows	160
4.5.1	MSG 7000 Keep alive	160
4.5.2	MSG 7001 Keep alive confirmation	160
4.5.3	MSG 7030 Syntax Error	160
4.5.4	MSG 7031 Not operation	160
5.	Alterations	161
5.1	Rev 1 2011-06-23	161
5.2	Rev 2 2012-03-02	161

5.3	Rev 3 2012-03-16	161
5.4	Rev 4 Version 2013 2012-10-30	161
5.5	Rev 5 Version 2013 2013-03-04	161
5.6	Rev 6 Version 2013.1 2013-05-29	161
5.7	Rev 7 Version 2014.1 2015-01-09	161
5.8	Rev 8 Version 2015-2015-04-13	161
5.9	Rev 9 Version 2015. 2015-09-08	161
5.10	Rev 10 Version 2016. 2016-02-26	162
5.11	Rev 11 Version 2017. 2017-05-15	162
5.12	Rev 12 Version 2018. 2017-10-02	162
5.13	Rev 13 Version 2018 2018-01-16	162
5.14	Rev 14 Version 2019	162
5.15	Rev 15 Version 2021	162

1. Use Cases general Description and Use

1.1 How to use Use Cases in SUTI

1.2 An introduction to SUTI terms and relations between terms

1.2.1 Introduction

To fully understand the SUTI communication it helps to understand the terms we use in the standard. This chapter gives description of the most fundamental terms and some fundamental relations between these terms.

However this is just an introduction. For all terms there are deeper descriptions in chapter 3 of Use Cases.

SUTI in itself is an abbreviation in swedish for "Standardiserat Utbyte av Trafik Information". In English it would mean Standardized Exchange (Use) of Traffic Information. It thus reduces the scope to Traffic, Traveling or Transportation.

Furthermore we have restricted the scope to Demand-Responsive Transportation. i.e. not covering situations run by timetables.

The terms can either be found as elements in our XML representation or as attributes or as descriptions of elements and attributes.

1.2.2 Message, Provider, Client, Request, Response and Flow

Message is obviously the most fundamental of all terms. With the SUTI protocol two parts named Client and Provider exchange Messages in order to fulfill some demands on each part.

The demand is expressed as a Request in a Message. The other part will act on the Request with a Response in one (or sometimes several consecutive Messages).

For each type of demand there will often be a chain of Messages. Such a chain of Messages is described in Use Cases. We use the word Flow to describe such sequences of Messages.

1.2.3 Who can be Client and who can be Provider? What is a Self Declaration in SUTI?

In SUTI we normally use Client for the part Requesting to solve a demand and Provider as the part that Responds. Thus a part may be Client in one Use Case and provider in another.

Inside complete Flows the Client always stays Client. A dominating Flow in SUTI is the Order Flow. In such cases the Client is always the part acting as a client for a Passenger (or more generally Content) while the Provider act as a part for the Vehicle fulfilling the traveling demand.

The Client should always express its demands in a complete list of Flows. Such a list of Flows we call Self Declaration.

1.2.4 SUTI BLOCKS (Sections)

Each SUTI Message has a number and a Short Name. All Messages are also grouped into a Block. At the moment we have 8 different Blocks. The first character (at the moment digit) for the message number is the same for all Messages in the same Block.

The Blocks 2-6 (Order, Dispatch, Traffic Control, Communication and Report) are all related to handling of an Order (see below).

Block 1 consists of Messages for General exchanges of Information normally connected to Resources.

Block 8 consists of Messages for Accounting between Client and Provider.

Block 7 consists of Messages for Technical control of the protocol

1.2.5 Route, Node, Content, Passenger, Parcel, Pickup, Destination and Trip

Since SUTI is working in a transportation environment we need some fundamental words for what we transport and how we transport.

Each transport has at least one Content. The dominating Content types are Passenger or Parcel, although there are a few additional very specific types in our enumeration list.

They will be a part of a Route in SUTI. A Route is a sequence of Nodes. A Node is not just a geographical place, it also contains the Content and descriptions of all other requirements to solve the situation of a Node. Thus the Route is a sequence of activities.

The Node describes activities in SUTI. Therefore the Node has a Node type. Most common is Pickup or Destination. A single Content usually has one Node for Pickup and one Node for Destination.

SUTI does not have a word to describe the Route for a single Content. We recommend to use the word Trip to describe that.

However to be noted a Route can be very complex having several different Content and several Nodes.

1.2.6 Order, SubOrder, Repetitive Order, OrderTemplate, DriverSession, Tour, Agreement

These words all describe containers of Nodes.

Order consists of a Route. The Route can either be dynamically built up by additions of Nodes or being static. Since some Orders were built up complexly from more simple orders the word SubOrder was introduced as a reference to the more simple Orders.

A Repetitive Order is a plan for Orders over a period of time e.g. School transports. The Repetitive Order is represented as an OrderTemplate having Calendar and scheduled Routes.

DriverSession consists of Orders where Orders can be dynamically added to a DriverSession. In this case it is assumed that the Orders are simple Orders (Base Order). Ideally these Orders should be Trips, however not specifically stated so. No references to SubOrders are allowed. We only have two levels inside SUTI.

Tour is an accounting representation of several Nodes and its Events (see below) used for accounting (settlement) purposes between parts in an Agreement. A Tour may be the effect of several Orders performed in a sequence. If DriverSessions is used the Tour would most likely represent one DriverSession.

All these containers refers to an Agreement between parties giving the legacy for exchanging and acting on both parts.

1.2.7 Events

Events are the registrations and communications of specifically activities for each Node, like "Vehicle at Node", "Passenger in Vehicle" etc. The events is normally the effect of the Driver making registrations using the equipment in the Vehicle. Not all equipment in the Vehicles may support all different Events specified in SUTI.

Since the first described Event was a PickupConfirmation all Events are unfortunately gathered under this tag. Had we not promised to be backwards compatible it should have had the name Event. It still has an attribute EventType, that clarifies that there are many different Events under PickupConfirmation.

1.2.8 Time

We have a lot of situations using Time. Obviously timing is crucial in most situations. We have a Times Type that we most often use. In SUTI we can then work with four different kind of TimeTypes. We use Scheduled, Estimated, Actual and ASAP (As Soon As Possible).

Our guidance for use is as follows

- Scheduled means the time that was planned and agreed with the Traveler (mostly)
- Estimated means the time that successively arises as a process evolves
- Actual signals that something has actually happened
- ASAP is used when there is a request that something shall happen as son as possible after the Time that is presented.

To observe is that this allows for sending Events in advance with an Estimated Time

A Time can also be accompanied by a DwellTime. This means that a Node (Activity) require additional Activity Time, e.g. some form of assistance.

1.2.9 Address, Location and Geographic Location

To describe the geography, e.g. in a Node we use the word Address. There are many ways to express an Address. Geographic Location is one of them. With Geographic Location we always mean the coordinates.

Inside an Address you can also find attrbutes like street, streetno etc. Amongst them there is also the word Location. In this sentence it means an area e.g. a postal area or a taxiarea. This is an exception in SUTI. When we talk about Location it is almost always the geographical coordinates that is ment.

1.2.10 Content (Passenger, Parcel) demands, Attributes, Connection

Besides the Route of the Content there a few other aspects of a Traveler oc Parcel to be mentioned. There are restrictions for the Transport depending on characteristics of the Content. The main type of such restrictions are gathered into Attributes.

For a Content there may be many such Attributes. Each type of Attribute is described in the SUTI Attribute List. For a Self Declaration it is essential to describe what types of Attribute that may occur.

A Connection in SUTI mean that a Node is dependant on a Connection with another Transport. Most often this is eihter a flight or train train transport.

1.2.11 Resource, Driver, Vehicle, Attributee

The resources Vehicle and driver are carefully described in chapter 3 of Use cases. The Attributes of a Content are condenced into demands on Vehicle or Driver. Thus both of them have a long list of Attributes also described in the SUTI Attribute List.

1.2.12 Economy, Price, Taximeter, FormOfPayment, Cash, Account, SocialServiceFee

The Economy part is large but consists mainly of two parts Price and formOfPayment.

Within Price it is possible to have several types of price calculations. One of them is a description of a Taximeter that the Transport is supposed to use, not the brand of a Taximeter, but how the Taximeter should be set in order to measure the Price of the Transport.

Inside FormOfPayment it is described how the Transport should be paid. There are several such options possible. To be noted here are three common ones. Cash being used if the Passenger pays in car (either by cash or card). Account if payment takes place outside car in an invoicing transaction. SocialService Fee in case the Passenger should pay a part of the Price.

1.3 Terms outside the use in SUTI

1.3.1 Booking, Segment, Reservation

Booking is a term used in the travel industry for the purpose where Reservations take place for different types of activities and resources. The Reservations are normally gathered into a PNR having one common description of payer and user (Passenger, Parcels).

A PNR can hold many Segments. Segments can be of different types like a flight or a taxi trip.

Normally these processes are a part inside a Client system and are not a part of a SUTI process. SUTI can be used to ask a provider to perform a taxi Segment. In such a case the Trip is transformed to an Order.

1.3.2 Fleet, Settlement (Dispatch)

A Provider for Order normally dispatches the Order to a Vehicle of a Fleet of Vehicles. This process is normally a process inside the Provider system and is normally not a part of a SUTI process.

However there is an alternative in SUTI for the Client to reject a Dispatching proposal. Block 3 of SUTI handles the results and proposals for the Dispatching process.

As a Response for an Order Status an Order may be in Dispatching Mode, meaning that the Provider are looking for a Vehicle to solve the Order.

The Provider normally has a process named Settlement to settle the economy between the Vehicle and the Provider Company. This process is also an internal process inside the Provider.

For the settlement between Client and Provider the term Accounting is used.

1.4 Techniques used for describing SUTI (XML and JSON)

1.4.1 Introduction

SUTI established its first implementation based on description in XML. From 2002 until 2020 XML has been the only "language" for SUTI. From 2021 JSON will be introduced as a second "language". In Use Cases you will therefore almost exclusively find examples in XML. As times go by we will introduce JSON examples as well.

In fact we did not just introduce XML by chance. It gave us the opportunity to define a schema that all messages inside the SUTI communication shall relay to. The XML Schema is a very powerful tool to secure compliance.

Does the introduction of JSON mean that Use Cases is not a guidance for JSON implementation?

On the contrary we mean. The way we introduce JSON is that to its major parts works the same way as an XML implementation. It would therefore be a rather simple activity to convert our examples to JSON. However this document is extensive and the introduction of JSON examples will be a successive work starting with examples that are found to be necessary with a JSON example.

From start our position is that the readers will have no problems of understanding how corresponding examples look in JSON. The Technical Committee is always at service if you will need such examples.

1.4.2 Differences and similarities between XML and JSON

From the introduction above you will find that we have really tried to build such a big similarity between the two "languages" that any example in XML is easy to understand how it should look like in JSON, once you are familiar with how the construction in JSON has been made. There are differences.

Our first ambition was to create JSON as a pure conversion. We left that because we found many disadvantages. Instead we have tried to make it a little simpler without cutting out any functionality. The reason is of course that JSON is a simpler "language" only having simple properties with text or digits and arrays of such properties.

1.4.2.1 Similarities between XML and JSON

All messages are the same having the same type of functionality. Only one message have got a completely new name, it is the 4010-message that when constructed was given the name pickupConfirmation. Over the years it developed as eventVehicle but our decision to be backward compatible gave us no chance to change. In JSON it is called eventVehicle.

Our promise with JSON is that we will be backward compatible from the version we decide to be SUTI 2021

All our properties that have enumeration is kept and the enumerations are the same. As with XML we introduce a schema definition possibility and all messages shall comply with these definitions.

We have kept the meaning and structure of all major objects like order, resource, node, content, economy etc.

We have also kept the names of the properties with a few changes, e.g. long have been lon when describing coordinate.

1.4.3 Differences

The biggest difference is the structure of messages. We have no root-tag. All messages have a structure composed of two objects.

One of the objects is a msg-object with the same basic structure as in XML describing sender, receiver, msgType etc. The difference being that referencesTo becomes an array of id-objects.

The other object in the message is the functional object like order, bulkLocation etc. Every such type of message is described in the schema as a msgObject, e.g. msgOrder. Several different msgTypes may share the same msgObject-description.

Unlike XML you will find all messages defined in the schema, not just as a comment but with its definitions because every message is an object named SUTI_XXXX_messageName, e.g. SUTI_2000_order.

In the schema you will find that we have avoided using Type when not necessary. The object types are instead just named with their name like node, content, id etc. This also gives us the opportunity to make arrays of such objects. In the XML-version building arrays did most often not require the definition of array. When we did we often used List to express that we have a List of objects, which per definition is an array. In JSON we have decided to use List for all arrays e.g. attributeList is an array of attributes.

When we are using arrays, they will be just arrays of the same type and does not repeat the object name for each individual array-element. This means that all array-elements are of the same object without the plural s, e.g. nodes is an array of node-objects. There may objects like referencesTo, that is an array of different idObjects having the same basic structure. In this case it is of course necessary to state which idType that each object represent.

1.5 Communication alternatives for SUTI

The most efficient way to communicate between Client and Provider is to use a two way communications where the Provider will be server for communication Client – Provider and the Client will be server for communication Provider – Client.

Words that we do not use in SUTI are Service and API.

Use of Service as a Technical Service has, as the word API, a relation to SUTI messages. These two words usually describes a communication containing Request and Response often in combination with Methods and Results. You can fulfill a communication demand by setting up a Technical Service or expose Methods in an API.

The reason not to use the terms Service and API is that they normally do not express Message Flows the same way as SUTI. Setting up a Services or API does not exclude the possibility to use the SUTI structures. However SUTI normally set up Flows in both directions. To fully implement the same demand as is solved in SUTI would normally therefore require Services or API both for Client and Provider.

It is important that the communication between Client and Provider can handle messages that is a part of the SUTI

standard but not handled by this particular Client or Provider without error or stop. Instead MSG 7030 or MSG 7031 shall be used to inform the other part that a message is not a part of the set of SUTI messages that the system can handle.

Today three different methods of SUTI communication exists

- Two way IP-protocol socket communication using different ports for Client and Provider. To secure such links they are often embedded into VPN-communication. There are some obvious advantages with this method. But there are often more disadvantages due to overhead in VPN-protocol and difficulties with firewalls on both sides rendering a change or setting up a new communication to a project with many involved parties.
- Client holding an URL where Provider polls for messages to Provider and uses the URL to deliver messages. The method have the advantages that the Client only set up one URL for all involved Providers. It however has the negative effect of polling generating a huge overhead.
- A third relatively new method has evolved. It is the use of Webb-sockets having the advantages of the first method and relieving of overhead. Today this method is strongly recommended as an alternative for new or changed SUTI-communication.

From 2020 we introduce the recommendation for an asynchronous communication using HTTP Post messages on both sides. It is thus an extension of the method using one URL on the Client side. We have also decided to set a standard for how to use such a solution. See below.

The use of HTTP Post can be introduced as a common service for all SUTI-links on Provider side and can significantly reduce the cost of establishing new links.

1.5.1 SUTI Communication using HTTP Post on each side.

Main objective is that both Client and Provider each establish one HTTPS Webservice reacting on POST following these rules:

1. The URL is constructed with three parameters senderSutid, receiverSutid and msgType. Reason is to identify exactly which SUTI-link that the message is a part of and what type of message that should be expected in the Body.
2. Authorization shall be Basic
3. Content-Type shall contain application/json; charset=UTF-8 if it is SUTI JSON that is sent in the Body, application/xml; charset=UTF-8 if it is SUTI xml.
4. To furthermore strengthen the security it is advised that the server uses white-listing of calling part securing that only authorized partners can use the service.

cURL example for a "Keep Alive" JSON format

```
curl --location --request POST 'https://holmedal.se/v1?senderSutid=systemprov_provsite_0001&receiverSutid=systemclient_clientsite_0009&msgType=7000' \
--header 'Content-Type: application/json; charset=UTF-8' \
--header 'Authorization: Basic aG9sbWVhYmV6QTFCMmMj' \
--data-raw '{"SUTI_7000_keepAlive": {"msg": {
    "msgType": "7000",
    "msgTimeStamp": "2019-12-18T10:39:00.000",
    "orgSender": {
        "name": "ThisProvider",
        "idOrg": {
            "src": "SUTI:idLink",
            "id": "systemprov_provsite_0001",
            "unique": true}
        },
    "orgReceiver": {
        "name": "ThisClient",
        "idOrg": {
            "src": "SUTI:idLink",
            "id": "systemclient_clientsite_0009",
            "unique": true}
        }
    }
}}
```

```

    },
    "idMsg": {
      "src": "systemprov_provsite_0001:MSGID",
      "id": "2019121812345678",
      "unique": true
    },
    "referencesTo": [{
      "idMsg": {
        "src": "systemclient_clientsite_0009:MSGID",
        "id": "2019121887654321",
        "unique": true}
    }
  ]
}

```

cURL example "Keep Alive" XML format

```
curl --location --request POST 'https://holmedal.se/v1?senderSutiId=systemclient_clientsite_0009&receiverSutiId=systemprov_provsite_0001&msgType=7000' \
--header 'Content-Type: application/xml; charset=UTF-8' \
--header 'Authorization: Basic aG9sbWVhYm91Q000MmMj' \
--data-raw '<SUTI>
  <orgSender name="ThisClient">
    <idOrg src="SUTI:idLink" id="systemclient_clientsite_0009" unique="true"/>
  </orgSender>
  <orgReceiver name="ThisProvider">
    <idOrg src="SUTI:idLink" id="systemprov_provsite_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="7000" msgName="Keep alive">
    <idMsg src=" systemclient_clientsite_0009:idMsg" id="2015121414795234" unique="true"/>
  </msg>
</SUTI>'
```

1.6 SUTI Relation to GDPR

In the SUTI communication there may be elements that falls into the GDPR like passenger name, id, contact information, attributes etc and driver informations also falling into the GDPR legislation. It is the parties responsibility to secure that they have the necessary acceptance from all included individuals that the communication is accepted.

It is also the responsibility to each party of the communication to ensure it acts in accordance with GDPR and other possible legislation regarding use of personell information.

It is strongly recommended that all communication using SUTI elements are encrypted.

It is also stongly recommended that personal information are restricted only to what is absololutely essential to fulfill the missions it is supposed to fulfill.

As an example SUTI has made adoptions to the protocol to secure the possibility to minimize of personell information like using calls between passenger and driver without revealing telephonenumber to each other.

2. Use cases

2.1 Order

2.1.1 Order flow

2.1.1.1 Basic flow

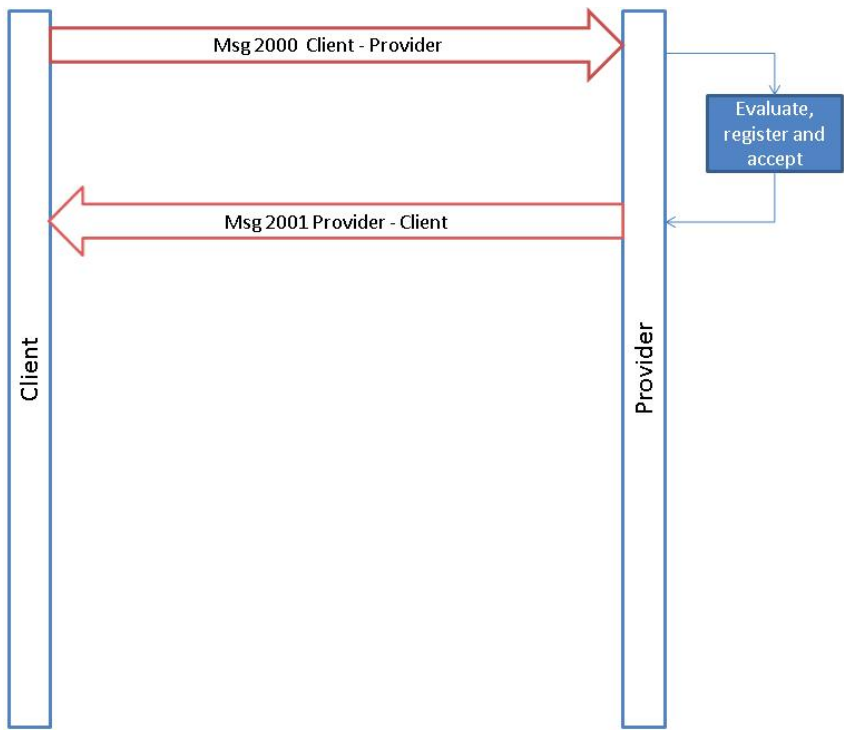
General

This flow can be used when the client only has the need to send orders to the provider and has no need for knowledge about how the order proceeds by the provider. With the addition of other Order Events (see below) like Order/node cancellation a fairly general simple order process can be maintained.

Recommended Solution

	Client	Provider	Comment
1	Sends MSG2000		Sent with a unique order id.
2		Receives MSG2000	Stores the order with an unique order id.
3		Immediately sends a MSG2001 as acceptance	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001		Stores the order id from the provider including the source used.

Flow chart



2.1.1.2 Typical flow

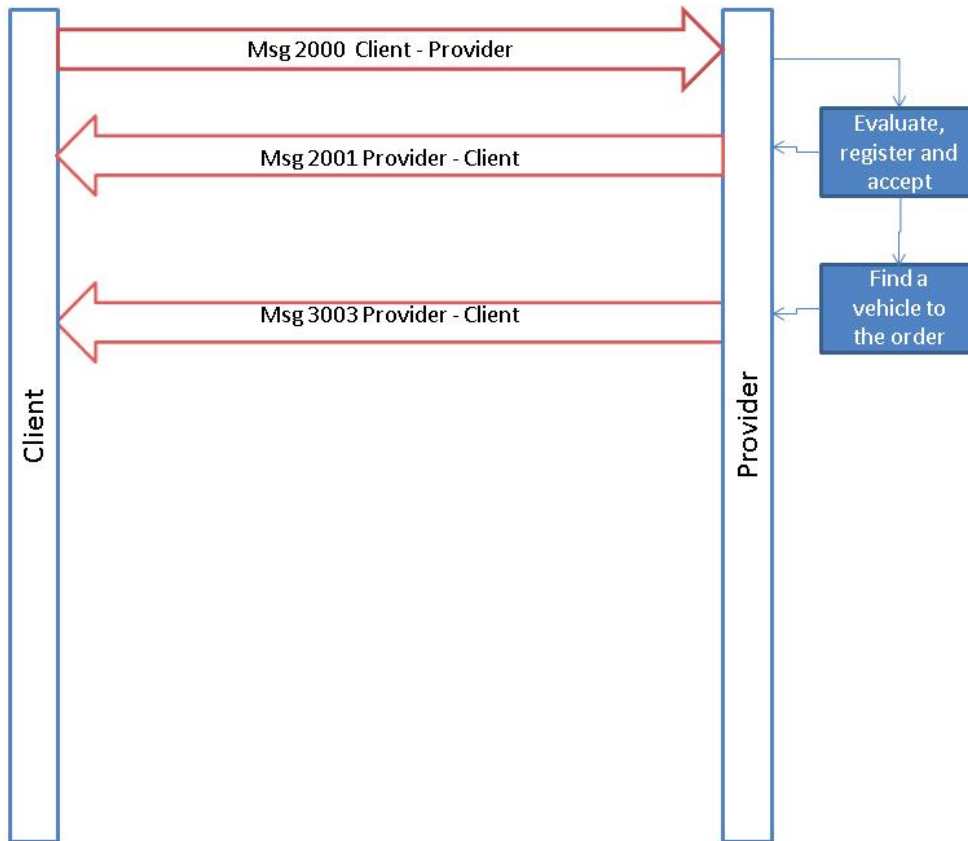
General

This flow can be used when the client has the need to send orders to the provider and need information about how the process continues at each node and has to know how the order was fulfilled. As for Basic Flow additions of Order Events forms the process to a general order process.

Recommended Solution

	Client	Provider	Comment
1	Sends MSG2000		Sent with a unique order id.
2		Receives MSG2000	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001		Stores the order id from the provider including the source used.
5		Sends MSG3003	Shall be sent after a vehicle has accepted the order. Shall include information about the identity of the vehicle
6		Sends MSG4010 for node number 1	At least one message shall be sent for each node
8		Sends MSG4010 for node number 2	
9		Sends MSG 6001 Order report	Shall be sent as soon as the receipt is printed out in the vehicle (order completed)

Flow chart



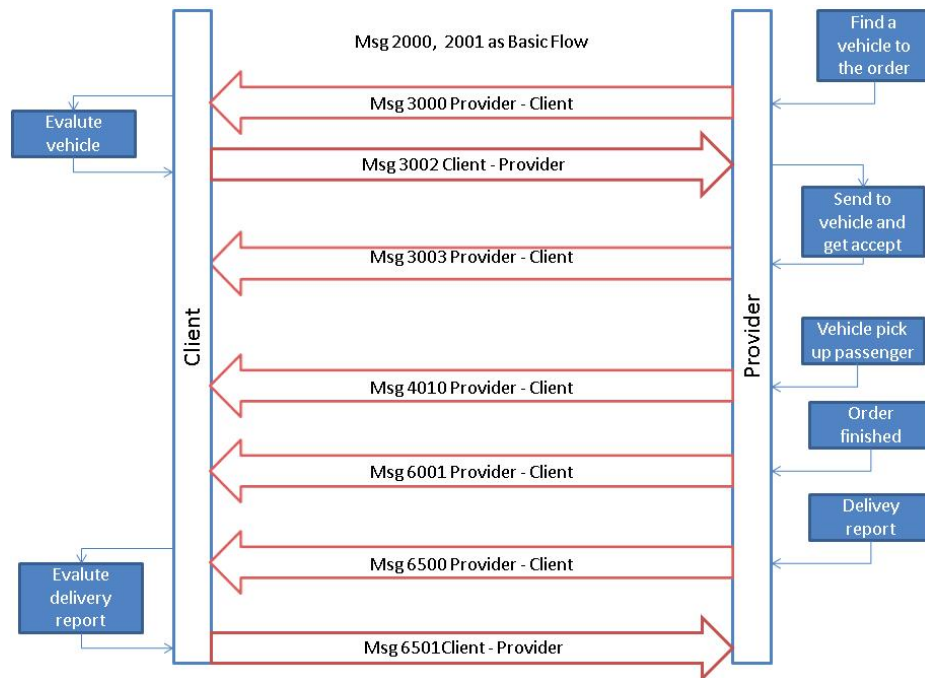
2.1.1.3 Extensive flow (Node by Node)**General**

This flow can be used when the client has the need to control and accept the resources that the provider suggests for each order. In addition the use of Delivery Note includes the possibility to exchange information necessary to handle the fulfilment of the agreement and its economical consequences. As for Basic Flow additions of Order Events forms the process to a general order process.

Recommended Solution

	Client	Provider	Comment
1	Sends MSG2000		Sent with a unique order id.
2		Receives MSG2000	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001		Stores the order id from the provider including the source used.
5		Sends MSG3000	Shall include the suggested vehicle
6	Sends MSG3002 as approval of the suggested vehicle		
7		Sends MSG3003	Shall be sent after a vehicle has accepted the order. Shall include information about the identity of the vehicle
8		Sends MSG4010 for node number 1	1709 vehicleatnode
9		Sends MSG4010 for node number 1	1701 passengerinvehicle
10		Sends MSG4010 for node number 2	1702 passengerdropof 8 to 10 shall be repeated for each set of nodes.
11		Sends MSG 6001 Order report	Shall be sent as soon as the receipt is printed out in the vehicle (order completed)
12		Sends 6500 delivery note	
13	Sends MSG6501 deliverynoteaccept		

Flow chart



2.1.1.4 Extensive flow with traffic control (Node by Node)**General**

This flow can be used when the client has an agreement that the provider shall put resources available to the client. When the resource is available the client will add work to each vehicle by providing more nodes. The client will end the process by a separate message (4020). The method for starting the process can differ but generally the first node is always present in the initial order as is represented in this flow.

The client must always accept the suggested resources before starting the process to pick-up first node.

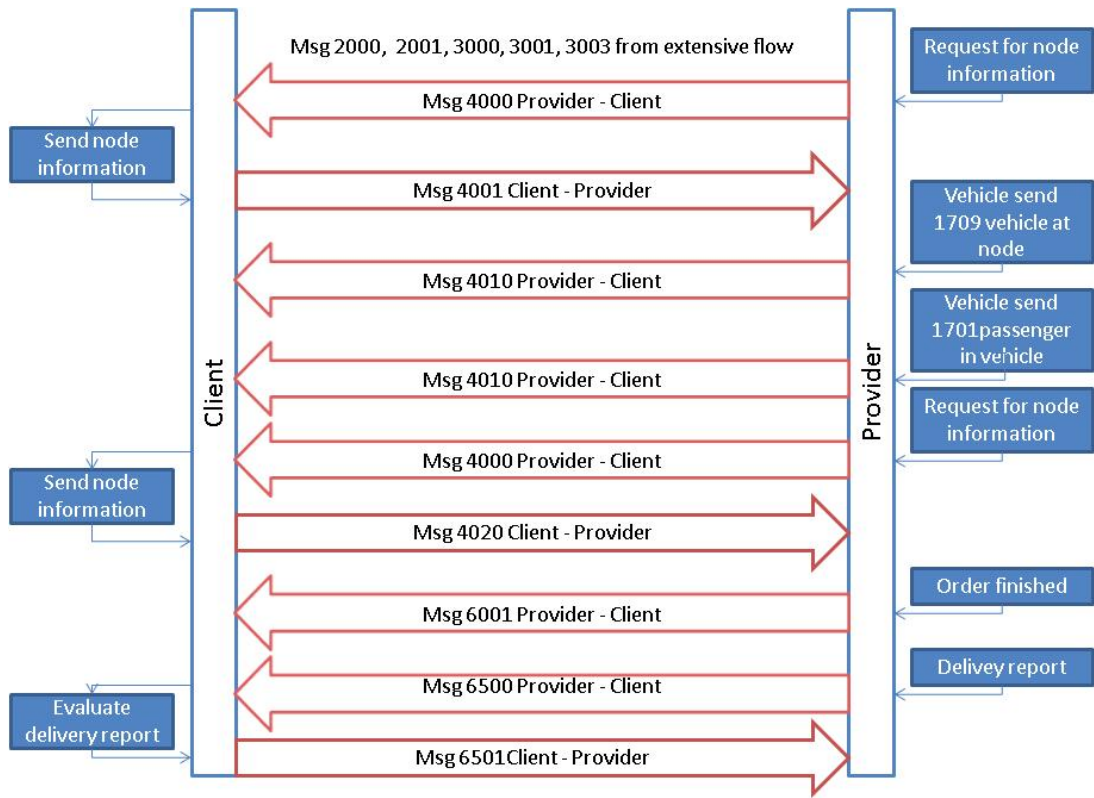
As for other flows above additions of Order Events forms the process to a general order process. Normally this is restricted to 2010 (Order cancellation request) prior to accept of resources (3002).

Recommended Solution

	Client	Provider	Comment
1	Sends MSG2000		Sent with a unique order id.
2		Receives MSG2000	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001		Stores the order id from the provider including the source used.
5		Sends MSG3000	Shall include the suggested vehicle
6	Sends MSG3002 as approval of the suggested vehicle		
7		Sends MSG3003	Shall be sent after a vehicle has accepted the order. Shall include information about the identity of the vehicle
8		Sends MSG4000	
9	Sends MSG4001		8 to 9 will be repeated for all nodes until the last node has been sent.
8		Sends MSG4010 for node number 1	1709 vehicleatnode
9		Sends MSG4010 for node number 1	1701 passengerinvehicle
8		Sends MSG4000	
9	Sends MSG4001		
10		Sends MSG4010 for node number 2	1702 passengerdropof 8 to 10 shall be repeated for each set of nodes.
		Sends MSG4000	
	Sends MSG 4020		Provider receives MSG 4020 and end the tour.
11		Sends MSG 6001 Order report	Shall be sent as soon as the receipt is printed out in the vehicle (order completed)
12		Sends 6500 delivery note	

13	Sends MSG6501 deliverynoteaccept		
----	-------------------------------------	--	--

Flow chart



2.1.1.5 Multiple client order combined into a shared provider order**General**

This describes a typical flow where the provider combines two orders from the client into a new combined order to be performed in a single resource. This flow can only be used if the client has set the allowRouting-flag in process to true and the orders are restricted to one combination of pick-up/drop-off for each content.

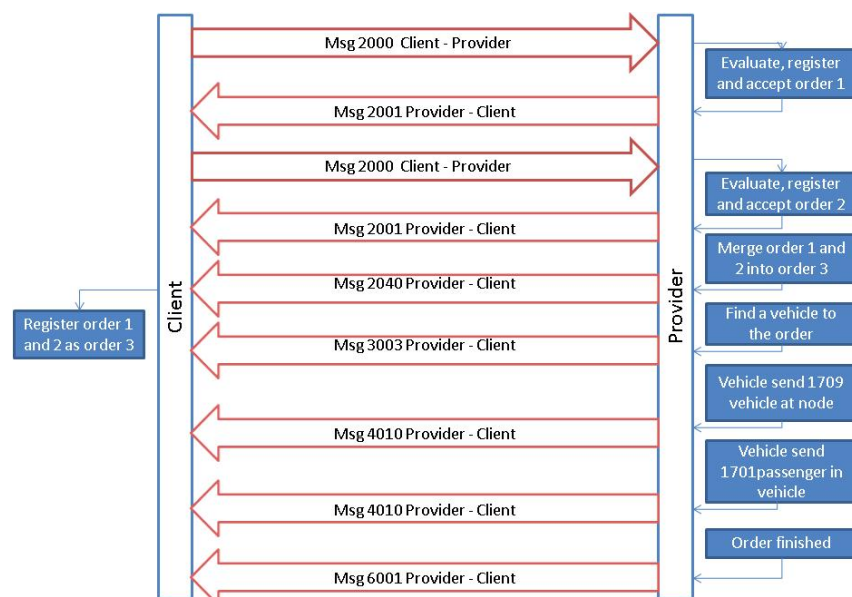
Recommended Solution

	Client	Provider	Comment
1	Sends MSG2000 for first order		Sent with a unique order id.
2		Receives MSG2000 for first order	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance for first order	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001 for first order		Stores the order id from the provider including the source used.
5	Sends MSG2000 for second order		Sent with a unique order id.
6		Receives MSG2000 for second order	Stores the order with a unique order id.
7		Immediately sends a MSG2001 as acceptance for second order	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo.
8	Receives MSG2001 for second order		Stores the order id from the provider including the source used.
9		Sends MSG2040	Shall include id of the linked orders together and the id of the combined order
10	Receives MSG2040		Stores the order information and after that be prepared to handle the new order information
11		Sends MSG3003	Shall be sent after a vehicle has accepted the order. Shall include information about the identity of the vehicle Shall refer to the combined order id and all of the suborder ids.
12		Sends MSG4010 for node number 1 suborder 1.	At least one message shall be sent for each node. Shall refer to the order id and the id of the

			suborder that includes the original node (suborder 1). This is normally node number 1 in the combined order.
13		Sends MSG4010 for node number 1 suborder 2.	At least one message shall be sent for each node. Shall refer to the order id and the id of the suborder that includes the original node (suborder 2). This is normally node number 2 in the combined order.
14		Sends MSG4010 for node number 2 suborder 1	At least one message shall be sent for each node. Shall refer to the order id and the id of the suborder that includes the original node (suborder 1). This is normally node number 3 in the combined order.
15		Sends MSG4010 for node number 2 suborder 2	At least one message shall be sent for each node. Shall refer to the order id and the id of the suborder that includes the original node (suborder 2). This is normally node number 4 in the combined order.
16		Sends MSG 6001 Order report	Shall be sent as soon as the receipt is printed out in the vehicle (order completed). Shall refer to the combined order id and all of the suborder ids

This way to handle multiple orders is restricted to orders with only 1 pickup and 1 dropoff node.

Flow chart



2.1.1.6 Order by Order

2.1.1.6.1 General

The order-by-order has developed as a method to send simple orders (base orders) and get them combined into a driverSession. From SUTI 2017 there is a new method to do this by introduction of a whole new set of messages. See chapter driverSession for more details.

However the method is already implemented for several links and is supported by all major provider systems in the Nordic countries. Therefore we have introduced this description to the standard.

The method is a way to impement traffic control for client and use simple orders instead of the node-by-node method. The provider systems will always sort the nodes by estimated time. The way the orders are gathered into a driverSession is by use of an idVehicle, which is the same for all orders in a driverSession.

In this chapter you will find an example where three orders are combined to one common routing. Recommended solution. This example contains 3 travellers (Ada, Beda and Ceasar).

Recommended Solution

	Client	Provider 1	Comment
1	Sends MSG2000 for the first part (Ada)		Sent with a unique order id for Ada.
2		Receives MSG2000 for the first part (Ada)	Stores the order with a unique order id for Ada.
3		Immediately sends a MSG2001 as acceptance (Ada)	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001 for the first part (Ada)		Stores the order id from the provider including the source used.
5		Sends MSG3000 for the first part (Ada)	Shall include the suggested vehicle
6	Sends MSG3002 as approval of the suggested vehicle (Ada)		
7		Sends MSG3003 (Ada)	Shall be sent after a vehicle has accepted the order. Shall include information about the identity of the vehicle
8	Sends MSG2000 for the second part (Beda)		Sent with a unique order id for Beda.
10		Receives MSG2000 for the second part (Beda)	Stores the order with a unique order id for Beda.
11		Immediately sends a MSG2001 as acceptance (Beda)	Shall be sent without any delay.
12		Sends MSG4010	1701 passengerinvehicle

		for Ada	
12		Sends MSG4010 for Beda	1701 passengerinvehicle
13	Sends MSG2000 for the third part (Ceasar)		Sent with a unique order id for Ceasar.
14		Receives MSG2000 for the second part (Ceasar)	Stores the order with a unique order id for Ceasar.
15		Immediately sends a MSG2001 as acceptance (Ceasar)	Shall be sent without any delay.
16		Sends MSG4010 for Ceasar	1701 passengerinvehicle
17		Sends MSG4010 for Ceasar	1702 passengerdropped
18		Sends MSG4010 for Beda	1702 passengerdropped
19		Sends MSG4010 for Ada	1702 passengerdropped
20	Sends 4020 End of Order.		This indicates that this mission is finished. It shall contain idVehicle and prices. Note that the text field shall not be used to tend this information.

To be observed:

If there is a change in times for Ada or Beda when entering new orders msg 2000 must be sent with odrerAlteration=true and holding new estimated times for these orders.

2.1.1.7 driverSession

2.1.1.7.1 General

The driverSession is a development of order-by-order where simple orders (base orders with no suborder) can be added to a driverSession and when modified several orders can be modified simultaneously.

The driverSession is a type of ordermsg (2100) containing description of all resources for entire order. The order is named driverSession to indicate that it may consist of several base orders that can be added from start or may be added in a successively mode. They will all be performed in a continuous session using the same vehicle and driver from first pickup of first order to last dropoff of last order, all in accordance with node-by-node and order-by-order.

However with driverSession the provider may implement communication to vehicle in which way the provider choose to do. To differentiate from node-by-node and order-by-order not all info must be communicated immediately. The provider system constructs a nodelist by sorting alla ordernodes by estimated time including possible consideration of dwell time. The nodelist is sent to client in all accept and reject messages as a response to all updates of the session.

None of the major provider systems have from 2017 the support to “keep” the base order information and successively release to vehicle, although this may be a preferred situation. Theefore it I wise if the client choose to implement a solution close to order-by-order that has a support from the major providersystems.

In order to make it possible for the providersystem to develop a support for successive communication of nodes to vehicle the client system must provide the provider not only with expected times.but also with information regarding expected times at node. These times shall be presented in dwell time for the node (dwell time means expected time from arrival at node to pickup or dropoff).

When such an implementation is done by a providersystem it may accommodate different types of equipment in a vehicle fleet. Until that happens the providersystem may have problems communicating certain changes to driverSessions.

For driverSession the dispatch will take place for the entire session. Thus msg 3xxx will only be used for first msg 2100. Once the vehicle is accepted this will be the vehicle for all the orders inside the driverSession. With this also follows the right for the provider to reject updates of the driverSession wich is not inside the resources of the chosen vehicle (and driver).

Observe that for the dispatch the provider do not need to investigate the resurces of each order. Only the resources of the session is vital for this purpose.

The Traffic control will be related to the nodes of each order and the sessions will be ended by the client sending a 4020 msg.

The development of the driverSession uses the same msg 2100 as the first initiation of the session. However the dispatch will only take place once. Once the vehicle is accepted after msg 3003 no more dispatch msg shall be sent. In all 2100 msg there are a changelog holding the instructions from client for orders in the session. By using the change-values added, updated, cancelled or moved the client gives the instruction to add, update, cancel or remove order from session. In case the order is moved it may (or may not) appear in other sessions. Such an order cannot be performed without being placed into another driverSession. If it is cancelled it cannot be reffered to again with its original orderid.

The reporting may take place for either each order or for the entire driverSession.

Below you will find a recommended solution for three orders (Ada, Beda and Caesar) in correspondence with order-by-order above.

Recommended Solution

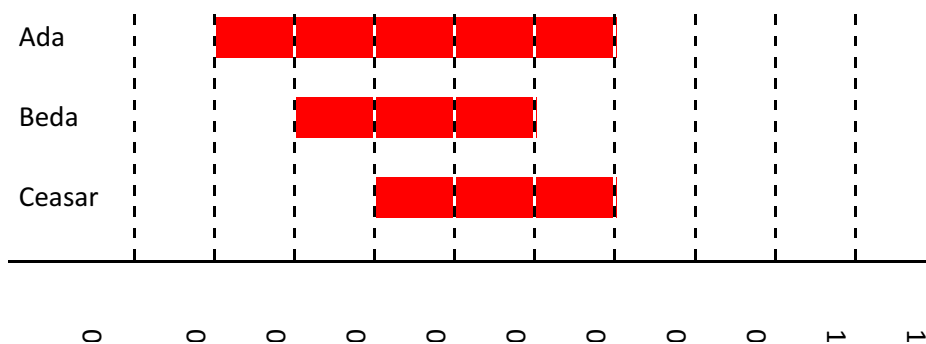
	Client	Provider 1	Comment
1	Sends MSG2100 including just the		Sent with a idSession and a unique order id for Ada.

	Ada order. The changelog of the session will only contain the Ada order stating Ada is added.		
2		Receives MSG2100 for the first part (Ada)	Stores the session with a unique id by the provider and theAda order with a unique order id for Ada.
3		Immediately sends a MSG2101 as acceptance. The msg 2101 will contain the nodelist (sessionNodes) only containing the nodes of Ada	Shall be sent without any delay. In ReferecesTo the correspondent sessionid by the provider will be sent. After this point the Provider has taken the ownership of the order and can't use msg 2102 if he for some reason cant finished the order. Instead the Provider has to use msg 2105 driverSession reject request.
4	Receives MSG2101 for the first part (Ada)		Stores the order id from the provider including the source used.
5		Chooses a vehicle based on the demands of the resources in the driverSession and sends MSG3000	Shall include the suggested vehicle
6	Sends MSG3002 as approval of the suggested vehicle		
7		Sends MSG3003	Shall be sent after a vehicle has accepted the session. Shall include information about the identity of the vehicle
8	Sends MSG2100 including both Ada and Beda. In the changelog Beda will be added and if the times is changed for Ada it will be marked as updated		Sent with a unique order id for Beda.
10		Receives MSG2100 for the second part (Beda)	Stores the new order with a unique order id for Beda. Calculates a new nodelist for the session
11		Immediately sends a MSG2001 as acceptance (Beda) and a nodelist containing both the nodes of Ada and Beda	Shall be sent without any delay.
12		Sends MSG4010 for Ada	1701 passengerinvehicle
12		Sends MSG4010 for Beda	1701 passengerinvehicle
13	Sends MSG2100 containing the		Sent with a unique order id for Ceasar.

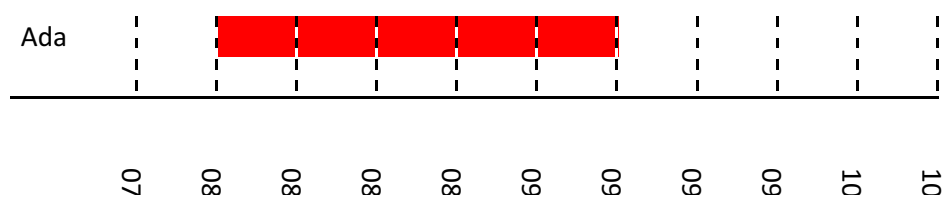
	three nodes Ada, Beda and Caesar. In the changelog the order Caesar will be marked added and if there is a change of times in the other orders they will be marked as updated		
14		Receives MSG2100	Stores the new order with a unique order id for Ceasar. Calculates a new nodelist for the session.
15		Immediately sends a MSG2001 as acceptance (Ceasar)	Shall be sent without any delay.
16		Sends MSG4010 for Ceasar	1701 passengerinvehicle
17		Sends MSG4010 for Ceasar	1702 passengerdropped
18		Sends MSG4010 for Beda	1702 passengerdropped
19		Sends MSG4010 for Ada	1702 passengerdropped
20	Sends 4020 End of Order.		This indicates that this session is finished. It shall contain idVehicle and prices. Note that the text field shall not be used to tend this information.

2.1.1.7.2 Time lines

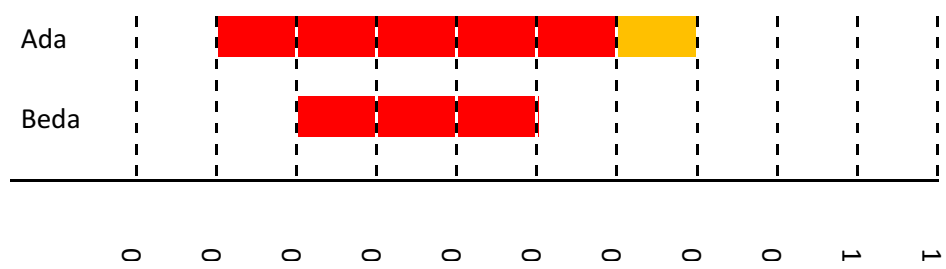
Original time lines for the 3 travellers (Requested pickup and dropoff):



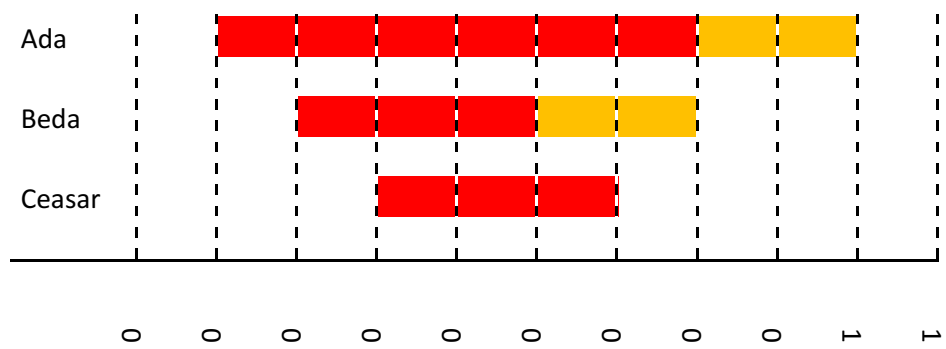
Time line 1 First order accepted (Ada)



Time line 2 Second order accepted (Ada and Beda)



Time line 3 Third order accepted (Ada, Beda and Ceasar)



2.1.1.8 Order Trading

General

Order trading is in itself a fairly simple flow as described below. The dynamics of the flow is that it contains sending of order to multiple Providers that may offer vehicles according to specifications given in the order. Ultimately one of the orders will be accepted and all others cancelled.

Recommended solution

	Client	Provider 1	Provider 2	Comment
1	Sends MSG2000			Sent with a unique order id.
2		Receives MSG2000	Receives MSG2000	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance	Immediately sends a MSG2001 as acceptance	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001 Receives MSG2001			Stores the order id from the provider including the source used.
5		Sends MSG3000		Shall include the suggested vehicle
6	Sends MSG3002 as approval of the suggested vehicle			
7		Sends MSG3003		Shall be sent after a vehicle has accepted the order. Shall include information about the identity of the vehicle
8	Sends MSG 2010 Order Cancellation			
9			Receives MSG 2010	
10			Sends MSG 2010 Order Cancellation Accepted	
11				
12				
13				

When using the order trading, the start location tag can be used to give the Client a sophisticated opportunity to choose “the closest” vehicle. The vehicle start position can be setup as following example.

```
<vehiclestartLocation typeOfCoordinate="WGS-84" lat="58.39193" long="13.42448" precision="10">
  <vehicleDistance range="1500">
    <startTime>
      <time timeZone="1" time="2010-08-18T07:00:00"/>
    </startTime>
    <stopTime>
      <time timeZone="1" time="2010-08-18T07:00:30"/>
    </stopTime>
  </vehicleDistance>
  <vehicleDistance range="5000">
    <startTime>
      <time timeZone="1" time="2010-08-18T07:00:31"/>
    </startTime>
    <stopTime>
      <time timeZone="1" time="2010-08-18T07:01:00"/>
    </stopTime>
  </vehicleDistance>
</vehiclestartLocation>
```

This allows the Provider that has a vehicle in a distance closer than 1500 meter radius to send a Msg 3000 between 07:00:00 and 07:00:30. If no vehicle is available no Msg 3000 can be sent. However between 07:00:31 and 07:01:00 Providers having a vehicle between 1500 and 5000 meter radius can send a Msg 3000.

2.1.1.9 PreeOrder with preeOrder flag “PROVIDER”

General

PreeOrder is a preebooked order submitted to the Provider and cannot have the timetype as soon as possible. The preebooked order can either be sent as a part of Msg 2800 repetitive order or as a single order Msg 2000. In both cases the process flag preeorder must be used. In the flow below Msg 2000 will be used. For Msg 2800 see chapter 2.1.11.

In this case the provider has the control and is the part that signals that no more alterations will be accepted by sending Msg 2050 Order freeze to the Client.

Recommended solution

	Client	Provider	Comment
1	Sends MSG2000 with process preeOrder flag “PROVIDER” used		Sent with a unique order id.
2		Receives Msg 2000	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance for first order	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo. After this point the Provider has taken the ownership of the order and can't use msg 2002 if he for some reason cant finished the order. Instead the Provider has to use msg 2005 Order reject request.
4	Receives MSG2001		Stores the order id from the provider including the source used.
5	The Client make an alteration by sending Msg 2070 with process flag preeOrder “PROVIDER” used.		Sent with the same orderId as the original order.
6		Receives Msg 2070	Replace old order with the new order.
7		Immediately sends a MSG2071 as acceptance for first order	Shall be sent without any delay.
8	Receives MSG2071 for altered order		Stores the order id from the provider including the source used.
9		Sends Msg 2050	At this point no more alterations are allowed. If need to alter an order arise after this point, the Client has to cancel the order and send a new order.
	Receives MSG 2050		

Flow chart

2.1.1.10 PreeOrder with preeOrder flag “CLIENTCONFIRM”**General**

PreeOrder is a preebooked order submitted to the Provider and cannot have the timetype as soon as possible. The preebooked order can either be sent as a part of Msdg 2800 repetitive order or as a single order Msg 2000. In both cases the process flag preeorder must be used. In the flow bellow Msg 2000 will be used. For Msg 2800 see chapter 2.1.11.

In this case the Client has the control and shall either confirm the pree booking with a Msg 2000 or cancel the pree booking with Msg 2010. If the Client doesn't do this the order will be converted to an order by the Provider and sent on dispatch.

Recommended solution

	Client	Provider	Comment
1	Sends MSG2000 with process preeOrder flag “CLIENTCONFIRM” used		Sent with a unique order id.
2		Receives Msg 2000	Stores the order with a unique order id.
3		Immediately sends a MSG2001 as acceptance for first order	Shall be sent without any delay. Both order ids shall be sent in ReferecesTo.
4	Receives MSG2001		Stores the order id from the provider including the source used.
5	The Client make an alteration by sending Msg 2070.		Sent with the same orderId as the original order.
6		Receives Msg 2070	Replace old order with the new order.
7		Immediately sends a MSG2071 as acceptance for first order	Shall be sent without any delay.
8	Receives MSG2071 for altered order		Stores the order id from the provider including the source used.
	Sends Msg 2000 to confirm the pree booking		The Client send msg 2000 to confirm that the preebooking shall be converted to order. After this point the Client cant alter the order. . If need to alter an order arise after this point, the Client has to cancel the order and send a new order.
3		Immediately sends a MSG2001 as acceptance of the conversion of the pree booking into an order.	Shall be sent without any delay. Both order ids shall be sent in ReferencesTo.
9		Sends Msg 2050	At this point no more alterations are allowed. If need to alter an order arise after this point, the Client has to cancel the order and send a new order.
	Receives MSG 2050		

Flow chart**2.1.1.11 Order Forwarded****General**

Order Forward is used to forward an order to a third party (Provider B) and to keep track of this order at all involved parties.

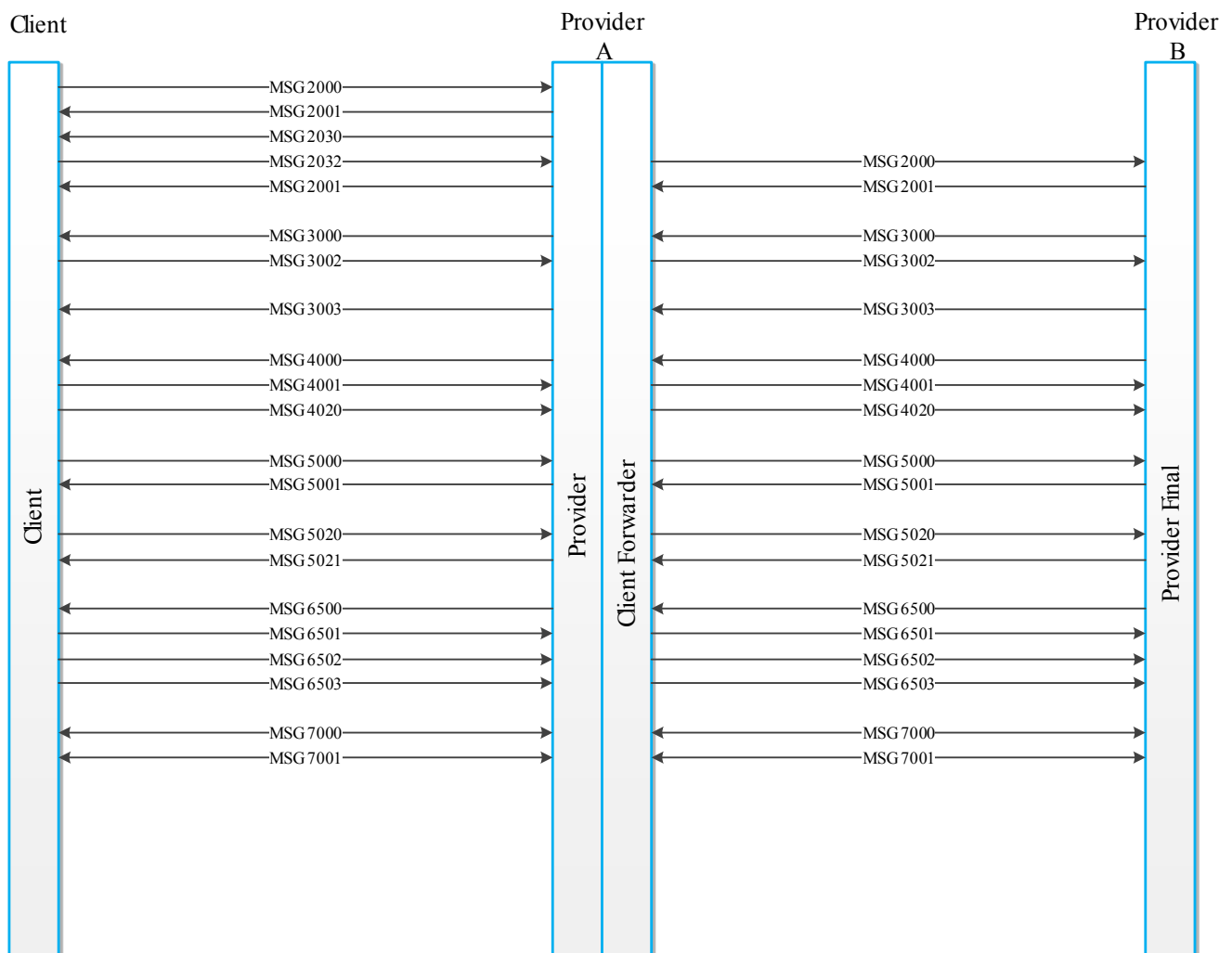
Recommended solution

	Client	ProviderA / Client forwarder	Provider B	Comment
1	Sends MSG2000 to Provider A with process allowForward set to True			Sent with a unique order id.
2		Receives Msg 2000 from Client		Stores the order with a unique order id.
3		Immediately sends a MSG2001 to Client as acceptance for first order		Shall be sent without any delay. Both order ids shall be sent in ReferencesTo.
4	Receives MSG2001 from Provider A			Stores the order id from the provider including the source used.
5		Sends MSG 2030 to Client with full information about Provider B in referencesTo.		Sent with full information about the intended provider (Provider B)
	Receives MSG 2030 from Provider A			Stores information.
	Sends MSG 2032 to Provider A to accept			Sent immediately if Client accepts the proposed Provider (Provider B)
		Receives MSG 2032 from Client		Receives and stores information
		Sends MSG 2000 to Provider B with full information about Client and Provider A in referencesTo..		Sent with a unique order id including idOrder from Client and information about Client..
			Receives Msg 2000 from Client forwarder	Stores the order with a unique order id including information from Client.
			Immediately sends a MSG2001 to Client forwarder as acceptance for first order with full information about idOrder and organisation in referencesTo	Shall be sent without any delay. All order ids shall be sent in ReferencesTo

		Receives MSG2001 from Provider B		Stores the order with a unique order id including information from boothClient and Provider B.
		Immediately sends a MSG2001 to Client as acceptance for first order. Shall include all information about order and organisation from both Provider A and Provider B in referencesTo.		
	Receives MSG2001 from Provider B and stores all information about idOrder and organisations from referencesTo			Stores the order with a unique order id including information from booth Provider A and Provider B.

After this is completed every message from Client shall be sent though Provider A via Client forwarder to Provider B and every message from Provider B shall be sent though Client forwarder via Provider A to Client. Every message shall have full information about involved order id's and organisations in referencesTo.

Flow Chart



2.1.2 Additional processes

2.1.2.1 Order status

General

This process is simple. The client ask for the information the provider has of a specified order and the provider sends this information to the client. This process can be used for several reasons:

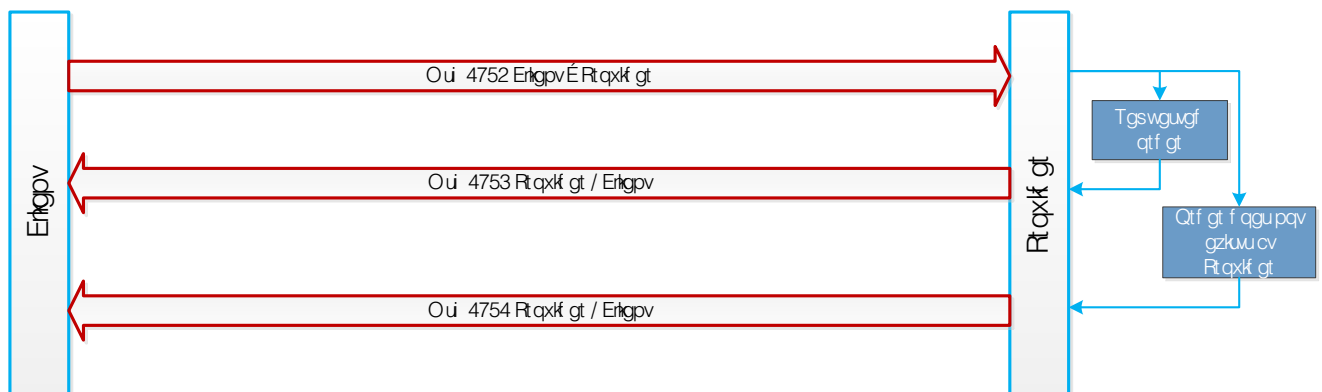
- the clients has not received appropriate messages in due time and need to secure its information status on an order
- the client has received a request from a traveller on a node and wants to secure its information status on an order
- the client has had a failure and wants to secure its information status on an order

Observe in process below the provider sends a 2531 with the requested order information. There is an alternative that the provider sends 2532 in case the order is not registered by the provider.

Recommended Solution

	Client	Provider	Comment
n	Sends MSG2530		Request for order status.
n+1		Receives MSG2530	Receive and generate a response to the request
n+2		Sends MSG2531	The message shall be sent without delay.
n+3	Receives MSG2531		

Flow chart



2.1.2.2 Alteration in dispatch with dispatch reject

General

These described process alternatives comes into action when the client must accept a suggested resource and the first suggestion is not accepted by client.

The client must set the process flag dispatch="true" and dispatchResponsible="client" in its initial order to make this process possible.

As seen below the provider initiate by sending its proposed resources. The client must answer immediately with either msg 3002, as described in the flow-chapter, or with 3001 in case the suggested resource is not accepted.

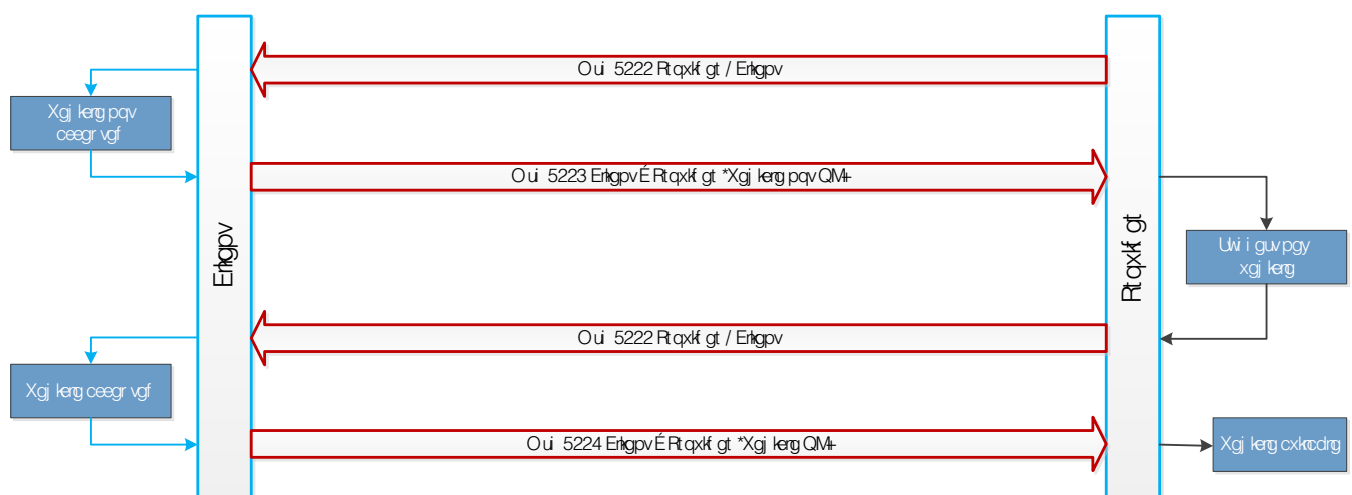
If the answer is 3001, the provider should send a new alternative suggestion with a new 3000.

This continues until the client answers with a 3002 or the provider has no more suggestions. In this case the provider should send 2005 Order reject request.

Recommended Solution

	Client	Provider	Comment
n		Sends MSG3000	Shall include id of the suggested vehicle.
n+1	Receives MSG3000		Receive and evaluate if the suggested vehicle is ok.
n+2	Sends MSG3001		The suggested vehicle is not ok
n+3		Receive MSG3001	Receive and start the procedure of finding a new vehicle
n+4		Sends MSG3000	Shall include id of the new suggested vehicle.
n+5	Receives MSG3000		Receive and evaluate if the suggested vehicle is ok.
n+6	Sends MSG3002		The suggested vehicle is ok
n+7		Receive MSG3002	Receive and send the order out to the suggested vehicle

Flow chart



2.1.2.3 Alteration in dispatch with new dispatch proposal

General

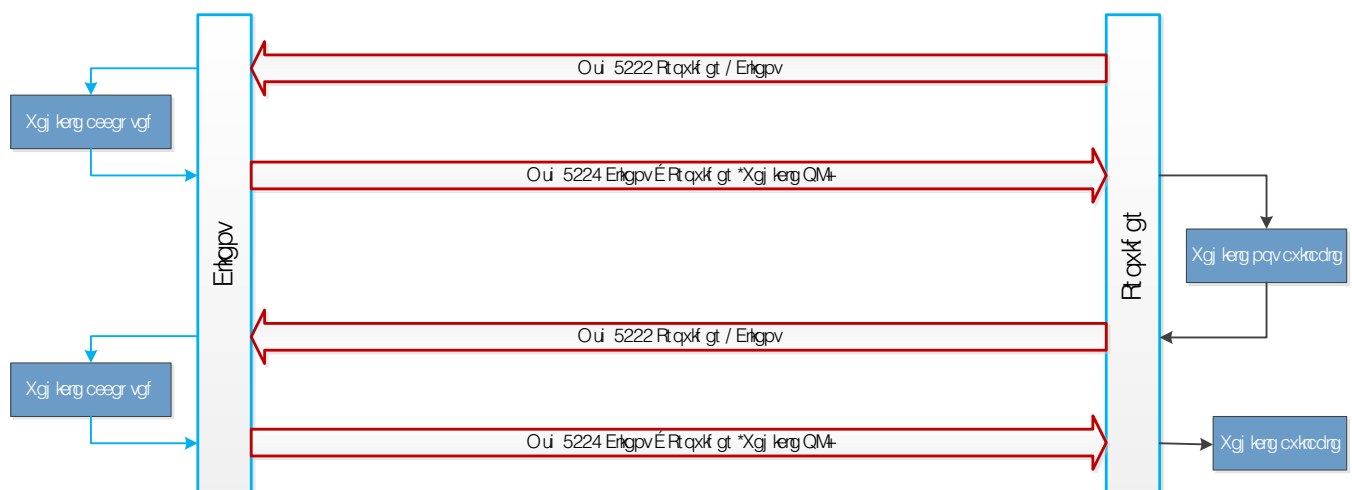
This process comes into action when the provider wants to change the resources of an order. It requires that there already exist a dispatch to a vehicle. This can either be accepted by client as in the preceding paragraph or just released to client in case the client does not have the option to accept dispatch (dispatchResponsible="provider"). For this last case the process is simple. The provider sends a new 3000 and the client stores the new alternative. Last 3000 will be the final result.

In case the client must accept the dispatch proposal a new process of the same kind as in paragraph above will take place.

Recommended Solution

	Client	Provider	Comment
n		Sends MSG3000	Shall include id of the suggested vehicle.
n+1	Receives MSG3000		Receive and evaluate if the suggested vehicle is ok.
n+2	Sends MSG3002		The suggested vehicle is ok
n+3		Receive MSG3002	Receive and send the order out to the suggested vehicle
n+4			<i>Suggested vehicle not available.</i>
N+5		Sends MSG3000	Shall include id of the suggested vehicle.
n+6	Receives MSG3000		Receive and evaluate if the suggested vehicle is ok.
n+7	Sends MSG3002		The suggested vehicle is ok
n+8		Receive MSG3002	Receive and send the order out to the suggested vehicle
			<i>Suggested vehicle available.</i>

Flow chart



2.1.2.4 Resource allocation

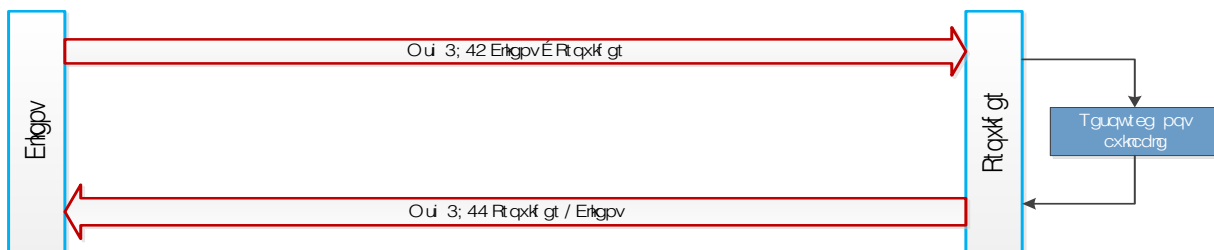
General

Resource allocation is used when the client wishes to either inquire if there is a resource available at a certain time in a certain place. Message 1920, 1921 and 1922 shall be used in this case.

2.1.2.4.1 The requested resource is not available

	Client	Provider	Comment
n	Sends MSG 1920		Shall contain the necessary data to the provider should be able to answer the question..
n+1		Receives MSG 1920	Receive and evaluate if the requested resource is available.
n+2		Sends MSG 1922	The requested resource is not available.
n+3	Receives MSG 1922		Receive the information that requested resource is not available.

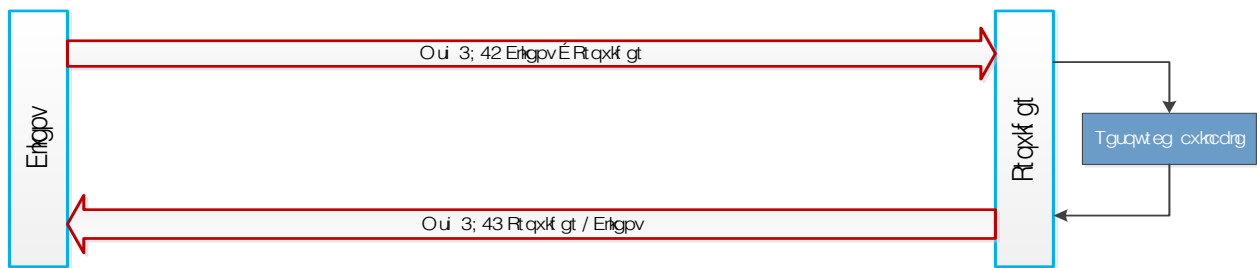
Flow chart



2.1.2.4.2 The requested resource is available

	Client	Provider	Comment
n	Sends MSG 1920		Shall contain the necessary data to the provider should be able to answer the question..
n+1		Receives MSG 1920	Receive and evaluate if the requested resource is available.
n+2		Sends MSG 1921	The requested resource is available.
n+3	Receives MSG 1921		Receive the information that requested resource is available.

Flow chart



2.1.2.5 Use Case request a fixed price and how to place an order

2.1.2.6 General

This flow can be used for a client to ask a provider for a fixed price.

Client	Provider	Comment
Sends Msg 1501	Answer with 1601	Provider creates an Authorization key to client as a security measure
Sends Msg 2000		If the client accepted the price send an order with the Authorization key received in msg 1601 in referencesTo
	Answer with msg 2001 or msg 2002	If everything is ok then accept the order, or if the Authorization key is invalid then reject the order.

*Example
Message 1501
"Request for*

Price"

```

- <SUTI>
- <orgSender name="Frogne App Exchange">
  <idOrg src="SUTI" id="frogne_app_002" unique="true" />
</orgSender>
- <orgReceiver name="HOLMEDAL SYSTEMS">
  <idOrg src="SUTI" id="Megtax_frogne_app_002" unique="true" />
</orgReceiver>
- <msg msgType="1501" msgName="Request for Price">
  <idMsg src="FROGNE:MSGID" id="2015061500000925" unique="true" />
- <infoRequest>
  - <requestItem>
    - <requestPrice>
      <idOrder src="FROGNE:PRICEID" id="xmlmobilos-30-148" unique="false" />
    - <agreement name="Frogne App Avtal">
      <idAgreement src="FROGNE:AGREEMENTID" id="Avtal001" unique="true" />
    - <product>
      <idProduct src="FROGNE:PRODUCTID" id="Produkt001" unique="true" />
    </product>
    </agreement>
    <process dispatch="true" dispatchResponsible="provider" trafficControl="false"
      report="true" preorderedVehicle="false" allowRouting="false"
      manualDispatch="false" orderAlteration="false" />
    - <route>
      - <node nodeSeqno="1" nodeType="pickup">
        - <addressNode street="Mossvågen" streetNo="11" location="Torup"
          postalNo="31441">
          <geographicLocation typeOfCoordinate="wgs-84" lat="56.9581756"
            long="13.0846214" precision="20" />
        </addressNode>
        - <timesNode>
          <time timeType="scheduledtime" timeZone="1" time="2015-06-
            15T10:45:00" />
        </timesNode>
        - <contents>
          - <content contentType="traveller" name="Benny jönsson">
            - <resourceContent>
              - <vehicle>
                - <capacity>
                  <seats noOfSeats="1" />
                </capacity>
              </vehicle>
            </resourceContent>
          </content>
        </contents>
      </node>
      - <node nodeSeqno="2" nodeType="destination">
        - <addressNode street="Mossvågen" streetNo="12" location="Torup"
          postalNo="31441">
          <geographicLocation typeOfCoordinate="wgs-84" lat="56.9559745"
            long="13.0823860" precision="20" />
        </addressNode>
        - <contents>
          - <content contentType="traveller" name="Benny jönsson" />
        </contents>
      </node>
    </route>
  </requestPrice>
</requestItem>
</infoRequest>
</msg>
</SUTI>

```

Example of the answer from provider message 1601 "Price response". And the security key that the client should have in the message 2000 "Order". This key is something the provider calculates, for example with the address locations, timesNode in the request message 1501.

ie" />

.55">

Example of the order message 2000, if the client want to book the transport.

```
- <SUTI>
- <orgSender name="Frogne App Exchange">
  <idOrg src="SUTI" id="frogne_app_002" unique="true" />
</orgSender>
- <orgReceiver name="HOLMEDAL SYSTEMS">
  <idOrg src="SUTI" id="Megtax_frogne_app_002" unique="true" />
</orgReceiver>
- <msg msgType="2000" msgName="Order">
  <idMsg src="FROGNE:MSGID" id="2015061500000926" unique="true" />
- <referencesTo>
  <idAuthorization src="Megtax_frogne_app_002:Authorization" id="0x0002421BE086465DD8"
    unique="true" />
</referencesTo>
- <order>
  <idOrder src="FROGNE:BOOKID" id="254" unique="true" />
- <agreement name="Frogne App Avtal">
  <idAgreement src="FROGNE:AGREEMENTID" id="Avtal001" unique="true" />
- <product>
  <idProduct src="FROGNE:PRODUCTID" id="Produkt001" unique="true" />
</product>
</agreement>
<process dispatch="true" dispatchResponsible="provider" trafficControl="false" report="true"
  preorderedVehicle="false" allowRouting="false" manualDispatch="false" orderAlteration="false" />
- <economyOrder>
  <price priceResponsible="client" fixedPrice="true" vatIncluded="true" />
- <formOfPayment>
  <payment amount="66" paymentType="account" />
</formOfPayment>
</economyOrder>
- <route>
- <node nodeSeqno="1" nodeType="pickup">
  - <addressNode street="Mossvågen" streetNo="11" location="Torup" postalNo="31441">
    <geographicLocation typeOfCoordinate="wgs-84" lat="56.95817560"
      long="13.08462140" precision="0" />
  </addressNode>
</node>
</route>
```

2.1.3 Order events Client

2.1.3.1 Order/node cancellation

Order/node cancellation can only be sent by the Client.

In the case of node cancellation it's necessary to provide the node in referencesTo.

The different conditions that regulate the different answers shall be agreed upon in the agreement.

2.1.3.1.1 Order/node cancellation accept

In the simplest case a message 2010/2020 is sent from the Client to the Provider and the Provider accepts the cancellation with a message 2011/2021 without any delay. The message need to have the attribute cancellationAcceptance set to TRUE and cancelationConsequence set to FALSE.

In such a case the Client cannot send the same order with the same idOrder. Should the Client do so the Provider should without any delay answer with 2003 orderreject and the rejectreason shall be 2513 ordercancelled.

2.1.3.1.2 Order/node cancellation accepted with consequence

This alternative shall be used if the Provider has started using resources that can be base for a claim on the Client. The message need to have the attribute cancellationAcceptance set to TRUE and cancelationConsequence set to TRUE. The accepted reasons for an acceptance with consequences shall be agreed upon in the agreement.

The economical consequence shall be stated in economycancellation

2.1.3.1.3 Order/node cancellation reject

This alternative shall be used if the Provider can't cancel the order. Reason for this can be that the order has already begun or the order is already finished. The accepted reasons for a rejection shall be agreed upon in the agreement.

2.1.3.2 Order alteration

An accepted way to make an alteration of an already accepted order is as follows.

The Client sends a new updated order with the same idOrder as the old one. This order shall have a referencesTo that includes the Providers idOrder. If the execution of the order hasn't started the Provider just replaces the old order with the new one. If the execution has started and a msg 4010 pickup has been sent for a node it's impossible to alter this node. In this case only future alterations will be accepted. This way of doing an alteration give the possibility to do alterations after the execution of the order has started. If this way is used to handle an alteration need to be agreed upon by both Client and Provider in the agreement. If the new altered order should exceed some limit agreed upon in the agreement it's impossible to use msg 2002 order reject as the Provider has accepted the ownership of the order. Msg 2002 can only be used to reject the original order. After the acceptance of the order the Provider has the responsibility for the order and must use msg 2005 order reject request after the acceptance of the order. If this method is used the process flagg orderAlteration should be set to TRUE.

Another accepted way of doing an alteration is to send a msg 2010 order cancelation followed by a new order. This order must not include any reference to the Providers old idOrder. In this case it's not possible to make alterations after the execution of the order has started.

To be discussed: The technical committee has a suggestion to handle the last case. If the Client want to alter an order after the execution has started it could be possible to use a new set of messages, node reject request that is built in the same way as order reject request. If the Client wants' to alter a node in a order that has been started the Client send a msg 2020 node cancelation and after the acceptance from the provider the Client sends a msg 4001 that includes the new node information.

The technical committee also suggests that addition of nodes to an existing order by sending a msg 4001 with new node information without the Providers system sending a msg 4000 to request such information.

2.1.3.3 Late order

A late order is an order that is sent from Client to provider after it actually should have been performed or the scheduled time is “impossible” to reach. It’s necessary that this is regulated in the agreement.

In the SUTI standard the times on a node has the following timetypes. In the text below we use pickup but it’s appropriate for other eventtypes.

2105	scheduled	This is the scheduled time for the vehicle to be at the pickup address. This normally the same time as the time that the Traveler and Client have agreed upon to be the pickup time. Thus this is the latest time that the traveler expects the vehicle to be on the pickup address.
2106	promised	In case where is a difference between the scheduled time and the time given to the Traveler the Client has the option to submit this agreed time by using 2106 promised. Both 2106 promised and 2105 scheduled must be included in the message in this case.
2108	asap	This is used for an immediate order where the Client and the Traveler has agreed that the vehicle shall be at the pickup address as soon as possible after the set time. Thus the set time could either be the booking time or a “best guess” from the Client and it is normally the earliest time that the vehicle shall be on the pickup address.
2107	estimated	This is a value that can be used from either client or Provider depending of the situation. The value means that has been an estimation of a new time for pickup. This is a “best guess” and not related to promise to the Traveler and/or agreement..
2104	actual	It’s the actual time of the event. This must be submitted by the Provider.
	Timetype not specified	In this case the default value is the same as 2104 actual e.g. in the message block.

There are two ways to handle a late order. The way to handle these two ways shall be specified and agreed upon in the agreement.

2.1.3.3.1 The hard way orderReject (Client is responsible for the order)

The Provider shall use IdRejectreason 2505 to late. The Client has the option to send the order back with the same idOrder with appropriate alterations. An appropriate alteration should be to change the value of timetype from scheduledtime to asapttime or send a new scheduledtime within an acceptable timeframe.

2.1.3.3.2 The softer way pickup confirmation (Provider is responsible for the order)

The provider sends a message 4010 pickup confirmation. The message shall contain at least a asap timetype with the value of time set to the earlier given scheduled time. And in the best of worlds the message should also contain an estimated time.

2.1.4 Order events Provider

2.1.4.1 Order reject

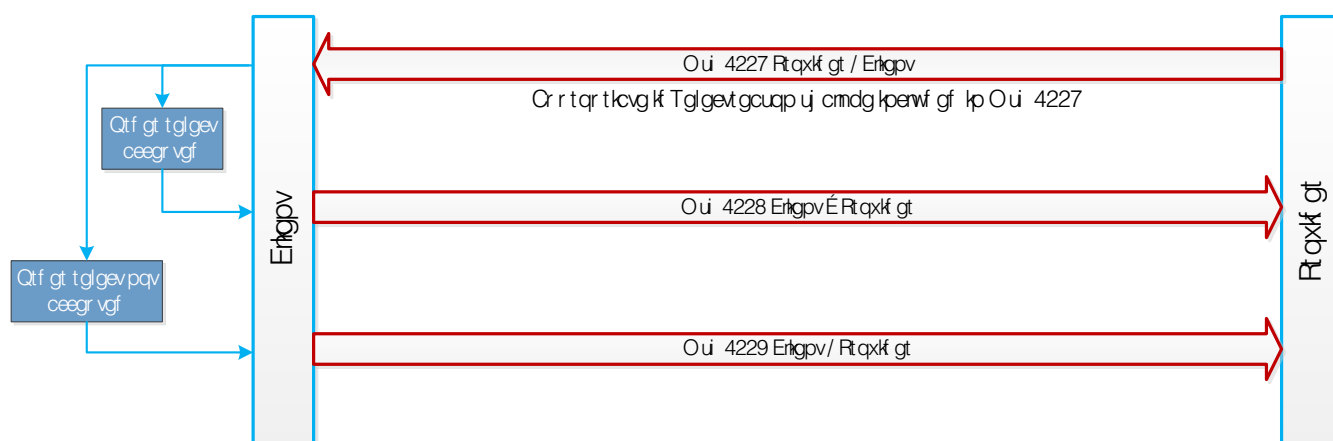
An order reject shall be sent without any delay and the reasons for an order reject shall be agreed upon in the agreement and no order accept shall be sent. If the reasons for reject aren't specified in the agreement every order reject shall include a reject reason.

2.1.4.2 Order reject request

Order reject request can only be sent after that the Provider has accepted the order is within the agreement.

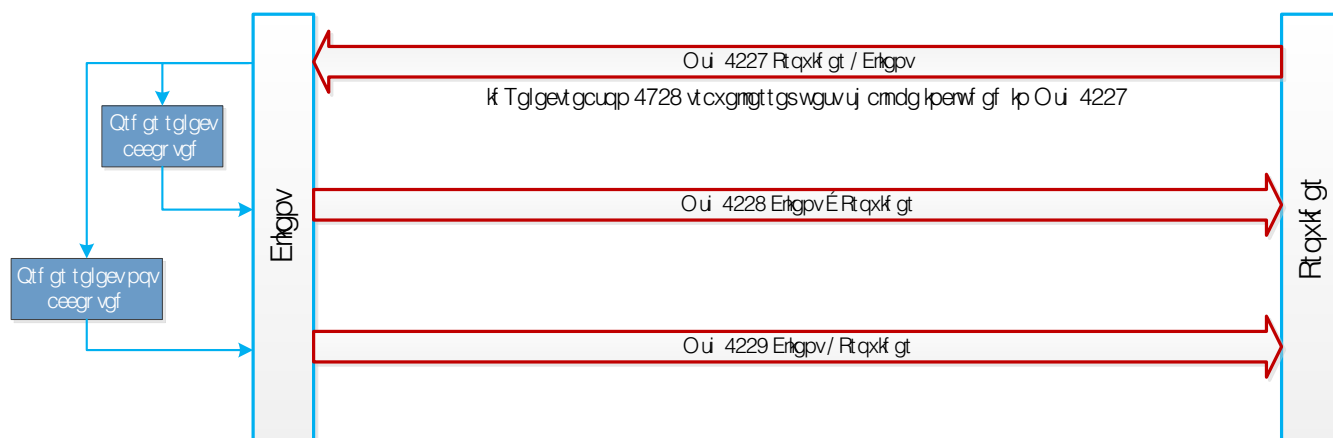
Sending the order reject request is done then the provider wishes that should not to any part be handled by the provider. The reason for this can be lack of capacity. If this is the case the message should contain idrejectreason 2504 nocapacity. The Client should without no delay answer with any of the messages 2006 order reject request accepted or 2007 order reject request rejected depending on the agreement situation.

Flow chart



There is a special case of order reject that may occur depending on agreement situation. This is handling of a request from the Traveler to cancel the order made directly to provider. This situation can occur when the provider has the authority to handle such requests. The procedure is the same as above but the idrejectreason should be 2506 travellerrequest. The provider must in both cases wait for the answer from the Client before setting the order status to cancel.

Flow chart



2.1.4.3 Node events

All node events are normally initiated by driver. Therefore this type of information is always true in the sense that the driver actually did the action. Errors depending on troubles with equipment is unlikely to occur.

Since long it has always been the question of quality of the messages. Is the vehicle really at the node? Is the passenger actually dropped off?

From 2017 SUTI has introduced two type of responses to the 4010 that is normally sent and is showed below.

The 4011 will be the normal response saying from client that the event has been received without complaints. But the client may respond with 4012 saying that the event has been received but with complaints. The complaint will be placed in a reject form that the provider shall send to driver.

The last action of a node event is however the driver. If the passenger has been pickedup so is the truth, therefore the driver may either change his action or repeat the action.

The driver may however repeat any 4010 message and last 4010 for an event type is the truth wether the client system has responded 4011 or 4012 (the event types are described in 3.2). If the driver has repeated the 4010 for the eventtype he/she has all right to believe that the client system shall register the event as true even if the response is 4012.

All node events except the 1710 vehicle at node should always only be sent using actual time types that has values very close to the actual time of day. The information type is of type real time and the value of the information is depending on the quality and and accuracy of the information.

With the 1710 an estimation of vehicle at node may be sent. Then it is obvious that it should contain estimated times.

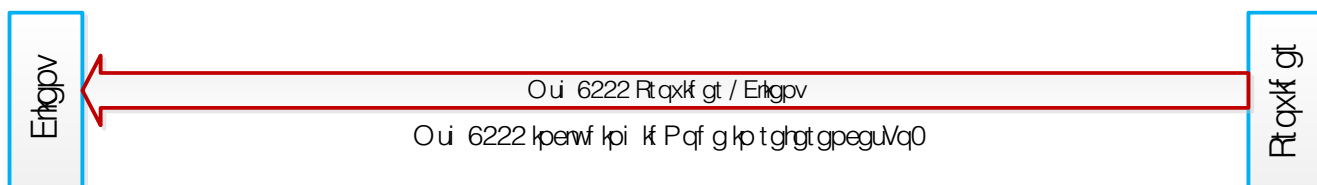
2.1.4.3.1 Message flow for node events

	Client	Provider	Comment
1		Sends MSG4010	Shall include the node number and a timetype containing attribute 2107 estimatedtime.
2	Sends either msg 4011 or msg 4012 after receipt of msg 4010		In case 4012 ith should include reasons possiible to send to vehicle and driver
3		Sends msg 4010 again	The information is either changed due to errors in last msg 4010 or repeated
4	Sends either msg 4011 or msg 4012 after receipt of msg 4010		If the msg 4010 was repeated with the same information the client should send 4011

2.1.4.4 Request for additional or corrected node information

If the information in a node in an order or a node in requested traffic information is incomplete it is possible to send a msg 4000 to request for additional or corrected information for instance the need of information about accurate positions or correct addresses. To request more information send a 4000 with idNode in referencesTo set to the node in question.

Flow chart



2.1.5 Order by Order and driverSession special processes

2.1.5.1 Resource description

Although orders are sent as basic orders the resource description at order level should contain the resource description for the entire session and should be the same for all orders. And especially important in the first order since this order guides the choice of vehicle for dispatch.

For driverSession this is much more defined since the driverSession holds the resource description necessary for dispatch and the providersystem can ignore the resources at each order.

2.1.5.2 Adding new order

For order by order the entire nodelist is always presented to driver in the vehicle equipment. It is likely that this will also be true for the first implementations of driverSession. A new order that requires action from the driver should be placed not later than as an immediate reaction (answer) to the message 4010 for previous node.

A certain preplacing of orders may be wise in order to secure that the driver always knows next step. Sending uncertain preplanning long in advance has on the other hand been tested with poor result resulting in problems communicating with vehicle and driver.

With driverSession when the provider has developed the full responsibility for the communication with vehicle and driver it may however be wise to send the whole planning of a session all the time.

In case the driver has started with intent to pickup or dropoff of a passenger, and an addition takes place of a new order inside the session, the driver has to make a change of action. It is possible and may be a very good planning. It is however wise to attend the driver by sending an extra message to vehicle to assure that the message gets through. The action may result in a longer time and distance to next node than else would be expected.

2.1.5.3 Change of orders

In the order by order changes is always sent either with order cancellation or resending the order with orderAlteration=True or a combination of order cancellation and resending the order depending on the situation and provider system.

In driverSession this is much clearer defined. The following holds true for driverSession and is normally also true for order by order.

1 No cancellation of order or move of order from session may take place after a 4010 has been sent for first node of an order. The provider system can reject this.

In case the vehicle has started its action to pickup a passenger and this is broken by a cancellation or move there has obviously been a cost for this by the provider. The solution for this will be dependant on the agreement. In the reporting of a driverSession however the time and distance will be included. It is recommended to minimize this type of action and reduce to very late cancellations by passengers.

If the 4010 for the first pickup is a true noshow the client may cancel nodes in sequence after the first node inside an order. It is done by sending an update of the order and only include one node. The entire order cannot be cancelled or moved.

In case that the client has contact with the passenger and "knows" that it should not be a true noshow, the client may send 4012 requesting the driver to search further. In case the driver then sends a pickup confirmation after a successful pickup this will replace the noshow.

2 Update may only occur for nodes where successful 4010 has not been received.

3 Update of other nodes than first node of an order in a session may not concern content but is restricted to times and geography. Problems with content shall be exchanged on first node in every order.

4 Update of an order in a session after 4010 dropoff of last node in the order is not acceptable and should result in msg 2102 for such an update.

2.2 Repetitive Orders

2.2.1 SUTI message OrderTemplate 2800

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2800" msgName="orderTemplate">
    <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true" orderTemplateName="A1"/>
      <orderTemplateCalendar>
        <validPeriod validFrom="2012-01-01T00:00:00" validTo="2015-12-31T23:59:59"/>
      </orderTemplateCalendar>
      <orderTemplateAgreement>
        <idAgreement src="HDSKOL_0001:FGrupp" id="empty" unique="true" />
        <product>
          <idProduct src="HDSKOL_0001:GruppKundId" id="HD-SKOL" unique="true" />
        </product>
      </orderTemplateAgreement>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1" scheduleElementFunction="Update"
scheduleElementResponse="J">
          <scheduleElementDate date="2013-12-09"/>
          <scheduleElementReferencesTo>
            <idOrder src="PLANET_0001:BestNr" id="486" unique="false"/>
            <idOrder src="HDSKOL_0001:HD_OrderNr" id="0" unique="true"/>
          </scheduleElementReferencesTo>
          <scheduleElementRoute>
            <!-- nodeType="1803" ==> "pickup" -->
            <node nodeSeqno="1" nodeType="1803">
              <addressNode addressName="Gruvgatan 1">
                <idZone src="HDSKOL_0001:Omr" id="11601" unique="true"/>
              </addressNode>
              <timesNode>
                <!-- timeType="2101" ==> "scheduledtime" -->
                <!-- timeType="2102" ==> "estimatedtime" -->
                <!-- timeType="2103" ==> "promisedtime" -->
                <time timeZone="1" time="2013-12-09T08:00:00" timeType="2101"/>
              </timesNode>
              <contents>
                <!-- contentType="1001" ==> traveller -->
                <content contentType="1001" name="empty">
                  <resourceContent>
                    <vehicle>
                      <idVehicle src="HDSKOL_0001:TrTill" id="0000_00" unique="true"/>
                      <capacity>
                        <seats noOfSeats="3"/>
                      </capacity>
                    </vehicle>
                  </resourceContent>
                </content>
              </contents>
            </node>
            <!-- nodeType="1804" ==> "destination" -->

```

```

<node nodeSeqno="2" nodeType="1804">
  <addressNode addressName="*VÄSTRA HAGASKOLAN ">
    <idZone src="HDSKOL_0001:Omr" id="21501" unique="true" />
  </addressNode>
  <timesNode>
    <time timeZone="1" time="2013-12-09T08:30:00" timeType="2101"/>
  </timesNode>
  <contents>
    <!-- contentType="1001" => traveller -->
    <content contentType="1001" name="empty">
      <resourceContent>
        <vehicle>
          <idVehicle src="HDSKOL_0001:TrTill" id="0000_00" unique="true"/>
          <capacity>
            <seats noOfSeats="3"/>
          </capacity>
        </vehicle>
      </resourceContent>
    </content>
  </contents>
</node>
</scheduleElementRoute>
</scheduleElement>
</scheduleElements>
</orderTemplate>
</msg>
</SUTI>

```

SUTI Message 2800	Kommentar
<pre> <orgSender name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgSender> </pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:FGrupp"
<pre> <orgReceiver name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgReceiver> </pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr" id="486"
msgType="2800"	SUTI nod <msg/>
msgName="orderTemplate"	SUTI nod <msg/>
<pre> <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNN" unique="true"/> </pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
orderTemplateName="A1"	SUTI nod <idOrderTemplate/> Namn på turen
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/> Unikt id för turen
src="HDSKOL_0001:GruppKundId" id="HD-SKOL"	SUTI nod <idProduct/> GruppKundId = Grupptillstånd, fakturamottagare
scheduleElementResponse="J"	SUTI nod <scheduleElement/>
validFrom="2012-01-01T00:00:00"	SUTI nod <validPeriod/> , anges med tid
validTo="2015-12-31T23:59:59"	SUTI nod <validPeriod/> , anges med tid
src="HDSKOL_0001:FGrupp" id=""	SUTI nod <idAgreement/>
src="HDSKOL_0001:TrTill" id="0000_00"	SUTI nod <idVehicle/>
scheduleElementSequenceNbr="1"	SUTI nod <scheduleElement/>

SUTI Message 2800	Kommentar
<code>scheduleElementFunction="Update"</code>	SUTI nod <code><scheduleElement/></code> , U anges som <i>"Update"</i> , B anges som <i>"Delete"</i> och I anges som <i>"Insert"</i>
<code>date="2013-12-09"</code>	SUTI nod <code><scheduleElementDate/></code> För fasta dagar anges inte datum utan noden <code><scheduleElementWeekday/></code> används där den angivna dagen har värdet = <i>"true"</i> För icke fasta dagar används noden <code><scheduleElementDate/></code> och datumet anges som <code>date="2013-12-09"</code>
<code>src="PLANET_0001:BestNr" id="486"</code>	SUTI nod <code><idOrder/></code>
<code>src="HDSKOL_0001:HD_OrderNr" id="0"</code>	SUTI nod <code><idOrder/></code>
<code>noOfSeats="3"</code>	SUTI nod <code><seats/></code>
<code>addressName="Gruvgatan 1"</code>	SUTI nod <code><addressNode/></code>
<code>src="HDSKOL_0001:Omr" id="11601"</code>	SUTI nod <code><idZone/></code>
<code>time="2013-12-09T08:00:00"</code>	SUTI nod <code><time/></code> det kan finnas flera <code><time/></code> noder under noden <code><timesNode/></code> med olika <i>timeType</i> attribut.
<code>addressName="*VÄSTRA HAGASKOLAN "</code>	SUTI nod <code><addressNode/></code>
<code>src="HDSKOL_0001:Omr" id="21501"</code>	SUTI nod <code><idZone/></code>
<code>time="2013-12-09T08:30:00"</code>	SUTI nod <code><time/></code> det kan finnas flera <code><time/></code> noder under noden <code><timesNode/></code> med olika <i>timeType</i> attribut.

2.2.2 SUTI message OrderTemplateConfirmation 2801

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2801" msgName="orderTemplateconfirmation">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurIdNr" id="238" unique="true" orderTemplateName="A1"/>
      <orderTemplateCalendar/>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1" scheduleElementFunction="Insert">
          <scheduleElementDate date="2013-04-30"/>
        </scheduleElement>
        <scheduleElementReferencesTo>
          <idOrder src="PLANET_0001:BestNr" id="0" unique="false"/>
          <idRejectReason src="PLANET_0001:Txt" id="Kan ej inplaneras" unique="false"/>
        </scheduleElementReferencesTo>
        </scheduleElement>
        <scheduleElement scheduleElementSequenceNbr="2" scheduleElementFunction="Insert">
          <scheduleElementDate date="2013-05-02"/>
        </scheduleElement>
        <scheduleElementReferencesTo>
          <idOrder src="PLANET_0001:BestNr" id="12189" unique="false"/>
        </scheduleElementReferencesTo>
        </scheduleElement>
        <scheduleElement scheduleElementSequenceNbr="3" scheduleElementFunction="Insert">
          <scheduleElementDate date="2013-05-03"/>
        </scheduleElement>
        <scheduleElementReferencesTo>
          <idOrder src="PLANET_0001:BestNr" id="25105" unique="false"/>
        </scheduleElementReferencesTo>
        </scheduleElement>
      </scheduleElements>
    </orderTemplate>
  </msg>
</SUTI>
```

SUTI Message 2801	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurIdNr"
msgType="2801"	SUTI nod <msg/>
msgName="orderTemplateconfirmation"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
orderTemplateName="A1"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:TurIdNr" id="238"	SUTI nod <idOrderTemplate/>

<i>scheduleElementSequenceNbr="1"</i>	SUTI nod <i><scheduleElement/></i>
<i>scheduleElementFunction="Insert"</i>	SUTI nod <i><scheduleElement/></i> , U anges som <i>"Update"</i> , B anges som <i>"Delete"</i> och I anges som <i>"Insert"</i>
<i>Date="2013-04-30"</i>	<p>SUTI nod <i><scheduleElementDate/></i></p> <p>För fasta dagar anges inte datum utan noden <i><scheduleElementWeekday/></i> används där den angivna dagen har värdet = <i>"true"</i></p> <p>För icke fasta dagar används noden <i><scheduleElementDate/></i> och datumet anges som <i>date="2013-12-09"</i></p>
<i>src="PLANET_0001:Txt" id="Kan ej inplaneras"</i>	SUTI nod <i><idRejectReason/></i>
<i>src="PLANET_0001:BestNr" id="0"</i>	SUTI nod <i><idOrder/></i>

2.2.3 SUTI message ScheduleElementConfirmation 2810

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2810" msgName="scheduleElementConfirmation">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <scheduleElementOrderList>
      <scheduleElementOrder>
        <scheduleElementOrderReferencesTo>
          <idOrder src="PLANET_0001:BestNr" id="486" unique="false"/>
          <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true"/>
        </scheduleElementOrderReferencesTo>
        <scheduledElementOrderDate date="2013-12-09"/>
      </scheduleElementOrder>
      <scheduleElementOrder>
        <scheduleElementOrderReferencesTo>
          <idOrder src="PLANET_0001:BestNr" id="487" unique="false"/>
          <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true"/>
        </scheduleElementOrderReferencesTo>
        <scheduledElementOrderDate date="2013-12-10"/>
      </scheduleElementOrder>
    </scheduleElementOrderList>
  </msg>
</SUTI>
```

SUTI Message 2810	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurlIdNr"
msgType="2810"	SUTI nod <msg/>
msgName="scheduleElementConfirmation"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
src="PLANET_0001:BestNr" id="486"	SUTI nod <idOrder/>
date="2013-12-09"	SUTI nod <scheduledElementOrderDate/>

2.2.4 SUTI message ConfirmationOfReceivedMessage 7099

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="7099" msgName="ConfirmationOfReceivedMessage">
    <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
    <referencesTo>
      <idOrder src="PLANET_0001:BestNr" id="486" unique="false"/>
      <!-- idRejectReason skickas vid eventuella fel -->
      <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/>
    </referencesTo>
  </msg>
</SUTI>
```

SUTI Message 7099	Kommentar
<pre><orgSender name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:MsgId"
<pre><orgReceiver name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr"
msgType="7099"	SUTI nod <msg/>
msgName="ConfirmationOfReceivedMessage"	SUTI nod <msg/>
<pre><idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="PLANET_0001:BestNr" id="486"	SUTI nod <idOrder/> skickar tillbaka det Bestnr som vi fick i msg 2810.
	Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/> Om svaret är OK skickas inte noden <idRejectReason/>
	Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/> Om svaret är OK skickas inte noden <idRejectReason/>

2.2.5 SUTI message ScheduleElementConfirmation 2810

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2810" msgName="scheduleElementConfirmation">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <scheduleElementOrderList>
      <scheduleElementOrder orderListClosed="true">
        <scheduledElementOrderDate date="2013-12-09"/>
      </scheduleElementOrder>
    </scheduleElementOrderList>
  </msg>
</SUTI>
```

SUTI Message 2810	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges.
msgType="2810"	SUTI nod <msg/>
msgName="scheduleElementConfirmation"	SUTI nod <msg/>
<pre><idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
orderListClosed="true"	SUTI nod <scheduleElementOrder/>
date="2013-12-09"	SUTI nod <scheduledElementOrderDate/>

2.2.6 SUTI message ReStart 7021

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="7021" msgName="ReStart">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
  </msg>
</SUTI>
```

SUTI Message 7021	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges.
msgType="7021"	SUTI nod <msg/>
msgName="ReStart"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken

2.2.7 SUTI message RequestForTrafficInformation 4000

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="4000" msgName="RequestForTrafficInformation">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <referencesTo>
      <idOrder src="HDSKOL_0001:HD_OrderNr" id="1254505" unique="true"/>
      <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true"/>
    </referencesTo>
    <manualDescriptionMsg sendtoInvoice="false" sendtoVehicle="false" sendtoOperator="false"
manualText="AltLayout=N" vehicleConfirmation="false"/>
  </msg>
</SUTI>
```

SUTI Message 4000	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurlIdNr"
msgType="4000"	SUTI nod <msg/>
msgName="RequestForTrafficInformation"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="HDSKOL_0001:HD_OrderNr" id="1254505"	SUTI nod <idOrder/>
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
sendtoVehicle="true" sendtoOperator="false"	SUTI nod <manualDescriptionMsg/>
sendtoVehicle="false" sendtoOperator="true"	SUTI nod <manualDescriptionMsg/>
manualText="AltLayout=N"	SUTI nod <manualDescriptionMsg/>
Saknas	Behövs inte då vi alltid skall svara

2.2.8 SUTI message RequestedTrafficInformation 4001

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="4001" msgName="RequestedTrafficInformation">
    <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
    <referencesTo>
      <!-- Reference to the message that this message answers. -->
      <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
    </referencesTo>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurlNr" id="238" unique="true" orderTemplateName="A1"/>
      <orderTemplateCalendar/>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1">
          <scheduleElementReferencesTo>
            <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
          </scheduleElementReferencesTo>
          <scheduleElementRoute>
            <!-- nodeType="1803" => "pickup" -->
            <node nodeSeqno="1" nodeType="1803">
              <addressNode addressName="Frågatan 1" postalNo="12345">
                <manualDescriptionAddress sendtoInvoice="false" sendtoVehicle="true" sendtoOperator="true"
manualText="#H:08:00#Frågatan 1#PN: 12345#Adr.Anm" vehicleConfirmation="true"/>
                <attributeAddress>
                  <attribute>
                    <idAttribute src="HDSKOL_0001:AdrAnm" id="Adr.Anm" unique="false"/>
                  </attribute>
                </attributeAddress>
              </addressNode>
              <timesNode>
                <!-- timeType="2101" => scheduledtime -->
                <time timeZone="1" time="2013-12-09T08:00:00" timeType="2101"/>
              </timesNode>
              <contents>
                <!-- contentType="1001" => traveller -->
                <content contentType="1001" name="Elev 1">
                  <attributeContent>
                    <attribute>
                      <idAttribute src="HDSKOL_0001:ElevAnm" id="Elev Anm." unique="true"/>
                    </attribute>
                    <attribute>
                      <idAttribute src="HDSKOL_0001:FAnm" id="Elev Adr. Anm." unique="false"/>
                    </attribute>
                  </attributeContent>
                  <!-- sendtoVehicle and sendtooperator values according to the 4000 -->
                  <manualDescriptionContent sendtoVehicle="true" sendtoOperator="false" sendtoInvoice="false"
vehicleConfirmation="true" manualText="#ELEV 1#Elev Adr. Anm.#Elev Anm."/>
                </content>
              </contents>
            </node>
            <!-- nodeType="1803" => "pickup" -->
            <node nodeSeqno="2" nodeType="1803">
              <addressNode addressName="ABC gatan 1" postalNo="41476">
                <manualDescriptionAddress sendtoInvoice="false" sendtoVehicle="true" sendtoOperator="true"
manualText="#H:08:15#ABC gatan 1#PN: 41476#Adr.Anm" vehicleConfirmation="true"/>
                <attributeAddress>

```

```

    <attribute>
    <idAttribute src="HDSKOL_0001:AdrAnm" id="Adr.Anm" unique="false"/>
    </attribute>
  </attributeAddress>
</addressNode>
<timesNode>
  <!-- timeType="2101" => scheduledtime -->
  <time timeZone="1" time="2013-12-09T08:15:00" timeType="2101"/>
</timesNode>
<contents>
  <!-- contentType="1001" => traveller -->
  <content contentType="1001" name="Elev 2">
    <attributeContent>
    <attribute>
      <idAttribute src="HDSKOL_0001:ElevAnm" id="Elev Anm." unique="true"/>
    </attribute>
    <attribute>
      <idAttribute src="HDSKOL_0001:FAnm" id="Elev Adr. Anm." unique="false"/>
    </attribute>
    </attributeContent>
    <!-- sendtoVehicle and sendtooperator values according to the 4000 -->
    <manualDescriptionContent sendtoVehicle="true" sendtoOperator="false" sendtoInvoice="false"
vehicleConfirmation="true" manualText="#ELEV 2#Elev Adr. Anm.#Elev Anm."/>
  </content>
</contents>
</node>
<!-- nodeType="1804" => "destination" -->
<node nodeSeqno="2" nodeType="1804">
  <addressNode addressName="SKOLGATAN 1" postalNo="43651">
    <manualDescriptionAddress sendtoInvoice="false" sendtoVehicle="true" sendtoOperator="true"
manualText="#L:08:30#SKOLGATAN 1#PN: 43651#Adr.Anm" vehicleConfirmation="true"/>
    <attributeAddress>
      <attribute>
        <idAttribute src="HDSKOL_0001:AdrAnm" id="Adr.Anm" unique="true"/>
      </attribute>
    </attributeAddress>
  </addressNode>
  <timesNode>
    <!-- timeType="2101" => scheduledtime -->
    <time timeZone="1" time="2013-12-09T08:30:00" timeType="2101"/>
  </timesNode>
  <contents>
    <!-- contentType="1001" => traveller -->
    <content contentType="1001" name="Elev 1">
      <attributeContent>
      <attribute>
        <idAttribute src="HDSKOL_0001:ElevAnm" id="Elev Anm." unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="HDSKOL_0001:TAnm" id="Elev Adr. Anm." unique="false"/>
      </attribute>
      </attributeContent>
      <!-- sendtoVehicle and sendtooperator values according to the 4000 -->
      <manualDescriptionContent sendtoVehicle="true" sendtoOperator="false" sendtoInvoice="false"
vehicleConfirmation="true" manualText="#ELEV 1#Elev Adr. Anm.#Elev Anm."/>
    </content>
    <content contentType="1001" name="Elev 2">
      <attributeContent>
      <attribute>
        <idAttribute src="HDSKOL_0001:ElevAnm" id="Elev Anm." unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="HDSKOL_0001:TAnm" id="Elev Adr. Anm." unique="false"/>
      </attribute>
    </content>
  </contents>
</node>

```

```

        </attribute>
    </attributeContent>
    <!-- sendtoVehicle and sendtooperator values according to the 4000 -->
    <manualDescriptionContent sendtoVehicle="true" sendtoOperator="false" sendtoInvoice="false"
vehicleConfirmation="true" manualText="#ELEV 2#Elev Adr. Anm.#Elev Anm." />
</content>
</contents>
</node>
</scheduleElementRoute>
</scheduleElement>
</scheduleElements>
</orderTemplate>
</msg>
</SUTI>

```

SUTI Message 4001	Kommentar
<pre> <orgSender name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgSender> </pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:MsgId"
<pre> <orgReceiver name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgReceiver> </pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
msgType="4001"	SUTI nod <msg/>
msgName="RequestedTrafficInformation"	SUTI nod <msg/>
<pre> <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/> </pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
orderTemplateName="A1"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:TurIdNr" id="238"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>

Kommentar:

Notera för att ta fram hela körordertexten för ordern skall följande attribut värden adderas.

```

<node nodeSeqno="1"
  <manualDescriptionAddress manualText="#H:08:00#Frångatan 1#PN: 12345#Adr.Anm" />
  <manualDescriptionContent manualText="#ELEV 1#Elev Adr. Anm.#Elev Anm." />
</node>

<node nodeSeqno="2"
  <manualDescriptionAddress manualText="#H:08:15#ABC gatan 1#PN: 41476#Adr.Anm" />
  <manualDescriptionContent manualText="#ELEV 2#Elev Adr. Anm.#Elev Anm." />
</node>

<node nodeSeqno="2"
  <manualDescriptionAddress manualText="#L:08:30#SKOLGATAN 1#PN: 43651#Adr.Anm" />
  <manualDescriptionContent manualText="#ELEV 1#Elev Adr. Anm.#Elev Anm." />
  <manualDescriptionContent manualText="#ELEV 2#Elev Adr. Anm.#Elev Anm." />
</node>

```


2.2.9 SUTI message Pickup Confirmation 4010

```
<?xml version="1.0" encoding="UTF-8"?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="4010" msgName="PickupConfirmation">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <referencesTo>
      <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
      <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true"/>
    </referencesTo>
    <manualDescriptionMsg sendtoInvoice="false" sendtoVehicle="false" sendtoOperator="false"
manualText="AltLayout=N" vehicleConfirmation="false"/>
  </msg>
</SUTI>
```

SUTI Message 4010	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurlIdNr"
msgType="4010"	SUTI nod <msg/>
msgName="PickupConfirmation"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
sendtoVehicle="true" sendtoOperator="false"	SUTI nod <manualDescriptionMsg/>
sendtoVehicle="false" sendtoOperator="true"	SUTI nod <manualDescriptionMsg/>
manualText="AltLayout=N"	SUTI nod <manualDescriptionMsg/>
Saknas	Behövs inte då vi alltid skall svara

2.2.10 SUTI message Order reject request 2005

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2005" msgName="OrderRejectRequest">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true"/>
      <orderTemplateCalendar/>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1">
          <scheduleElementDate date="2013-08-20"/>
          <scheduleElementReferencesTo>
            <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
            <idOrder src="PLANET_0001:BestNr" id="486" unique="false"/>
            <idDriver src="PLANET_0001:OperId" id="BAL" unique="true"/>
          </scheduleElementReferencesTo>
        </scheduleElement>
      </scheduleElements>
    </orderTemplate>
  </msg>
</SUTI>
```

SUTI Message 2005	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurlIdNr"
msgType="2005"	SUTI nod <msg/>
msgName="OrderRejectRequest"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
src="PLANET_0001:BestNr" id="486"	SUTI nod <idOrder/>
date="2013-12-09"	SUTI nod <scheduledElementOrderDate/>
src="PLANET_0001:OperId" id="BAL"	SUTI nod <idDriver/>

2.2.11 SUTI message Orderstatus 2531

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2531" msgName="orderStatus">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true" orderTemplateName="A1"/>
      <orderTemplateCalendar/>
      <orderTemplateAgreement>
        <idAgreement src="HDSKOL_0001:GruppKundId" id="HD-SKOL" unique="true"/>
      </orderTemplateAgreement>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1">
          <scheduleElementDate date="2013-08-20"/>
          <scheduleElementReferencesTo>
            <idOrder src="PLANET_0001:BestNr" id="486" unique="false"/>
          </scheduleElementReferencesTo>
        </scheduleElement>
      </scheduleElements>
    </orderTemplate>
  </msg>
</SUTI>
```

SUTI Message 2531	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. ex. src="HDSKOL_0001:GruppKundId"
msgType="2531"	SUTI nod <code><msg/></code>
msgName="orderStatus"	SUTI nod <code><msg/></code>
<pre><idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="PLANET_0001:BestNr" id="90181"	
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <code><idOrderTemplate/></code>
orderTemplateName="A1"	SUTI nod <code><idOrderTemplate/></code>
src="HDSKOL_0001:GruppKundId" id="HD-SKOL"	SUTI nod <code><idAgreement/></code> GruppKundId = Grupptillstånd, fakturamottagare
date="2013-08-20"	SUTI nod <code><scheduleElementDate/></code> För fasta dagar anges inte datum utan noden <code><scheduleElementWeekday/></code> används där den angivna dagen har värdet = "true" För icke fasta dagar används noden <code><scheduleElementDate/></code> och datumet anges som

	date="2013-08-20"
--	-------------------

Kommentar:

HD-skol svar på msg 2531 är msg 2800 med `scheduleElementFunction="Update"` eller `scheduleElementFunction="Delete"` beroende på om ordern finns i HD-skol.

2.2.12 SUTI message RequestForOrderInformation 6810

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="6810" msgName="requestForOrderInformation">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <scheduleElementOrderList>
      <scheduleElementOrder>
        <scheduleElementOrderReferencesTo>
          <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
          <idOrder src="PLANET_0001:BestNr" id="91181" unique="false"/>
          <idOrderTemplate src="HDSKOL_0001:TurlidNr" id="238" unique="true"/>
        </scheduleElementOrderReferencesTo>
        <scheduledElementOrderDate date="2013-08-20"/>
      </scheduleElementOrder>
    </scheduleElementOrderList>
  </msg>
</SUTI>
```

SUTI Message 6810	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurlidNr"
msgType="6810"	SUTI nod <msg/>
msgName="requestForOrderInformation"	SUTI nod <msg/>
<pre><idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>
src="HDSKOL_0001:TurlidNr" id="238"	SUTI nod <idOrder/>
date="2013-08-20"	<p>SUTI nod <scheduleElementDate/></p> <p>För fasta dagar anges inte datum utan noden <scheduleElementWeekday/> används där den angivna dagen har värdet = "true"</p> <p>För icke fasta dagar används noden <scheduleElementDate/> och datumet anges som date="2013-08-20"</p>
src="PLANET_0001:BestNr" id="91181"	SUTI nod <idOrder/>

2.2.13 SUTI message RequestedOrderInformation 6800

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="6800" msgName="requestedOrderInformation">
    <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurIdNr" id="238" unique="true" orderTemplateName="A1"/>
      <orderTemplateCalendar/>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1">
          <scheduleElementReferencesTo>
            <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
            <idOrder src="PLANET_0001:BestNr" id="486" unique="false"/>
            <!-- idRejectReason skickas vid eventulla fel -->
            <idRejectReason src="PLANET_0001:Txt" id="Eventuell fel text" unique="false"/>
          </scheduleElementReferencesTo>
          <scheduleElementRoute>
            <!-- nodeType="1803" => "pickup" -->
            <node nodeSeqno="1" nodeType="1803">
              <addressNode addressName="VILLAGATAN 3">
                <idZone src="HDSKOL_0001:HOMr" id="TRC002" unique="true" />
              </addressNode>
              <timesNode>
                <!-- timeType="2101" => scheduledtime -->
                <time timeZone="1" time="2013-08-20T08:30:00" timeType="2101"/>
              </timesNode>
              <contents>
                <!-- contentType="1001" => traveller -->
                <content contentType="1001" name="KALLE SVENSSON">
                  <idContent src="HDSKOL_0001:KId" id="010101-1234" unique="true"/>
                  <attributeContent>
                    <attribute>
                      <idAttribute src="HDSKOL_0001:Bet" id="TRSKOL" unique="true"/>
                    </attribute>
                    <attribute>
                      <idAttribute src="HDSKOL_0001:KostSt" id="TRSKO1" unique="true"/>
                    </attribute>
                    <attribute>
                      <idAttribute src="HDSKOL_0001:Proc" id="47.0" unique="true"/>
                    </attribute>
                    <attribute>
                      <idAttribute src="HDSKOL_0001:Bom" id="N" unique="true"/>
                    </attribute>
                    </attributeContent>
                  </content>
                </contents>
              </node>
            <!-- nodeType="1803" => "pickup" -->
            <node nodeSeqno="2" nodeType="1803">
              <addressNode addressName="VILLAGATAN 18">
                <idZone src="HDSKOL_0001:HOMr" id="TRC002" unique="true" />
              </addressNode>
              <timesNode>
                <!-- timeType="2101" => scheduledtime -->
                <time timeZone="1" time="2013-08-20T08:32:00" timeType="2101"/>
              </timesNode>
            </node>
          </scheduleElementRoute>
        </scheduleElement>
      </scheduleElements>
    </orderTemplate>
  </msg>
</SUTI>

```

```

<contents>
  <!-- contentType="1001" => traveller -->
  <content contentType="1001" name="GRETA ERIKSSON">
    <idContent src="HDSKOL_0001:KId" id="020202-1234" unique="true"/>
    <attributeContent>
      <attribute>
        <idAttribute src="HDSKOL_0001:Bet" id="TRSKOL" unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="HDSKOL_0001:KostSt" id="TRSKO1" unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="HDSKOL_0001:Proc" id="53.0" unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="HDSKOL_0001:Bom" id="N" unique="true"/>
      </attribute>
    </attributeContent>
  </content>
</contents>
</node>
<!-- nodeType="1804" => "destination" -->
<node nodeSeqno="3" nodeType="1804">
  <addressNode addressName="TROLLSKOLAN">
    <idZone src="HDSKOL_0001:L0mr" id="TRV018" unique="true" />
  </addressNode>
  <timesNode>
    <!-- timeType="2101" => scheduledtime -->
    <time timeZone="1" time="2013-08-20T08:45:00" timeType="2101"/>
  </timesNode>
  <contents>
    <!-- contentType="1001" => traveller -->
    <content contentType="1001" name="KALLE SVENSSON">
      <idContent src="HDSKOL_0001:KId" id="010101-1234" unique="true"/>
      <attributeContent>
        <attribute>
          <idAttribute src="HDSKOL_0001:Bet" id="TRSKOL" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="HDSKOL_0001:KostSt" id="TRSKO1" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="HDSKOL_0001:Proc" id="47.0" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="HDSKOL_0001:Bom" id="N" unique="true"/>
        </attribute>
      </attributeContent>
    </content>
    <!-- contentType="1001" => traveller -->
    <content contentType="1001" name="GRETA ERIKSSON">
      <idContent src="HDSKOL_0001:KId" id="020202-1234" unique="true"/>
      <attributeContent>
        <attribute>
          <idAttribute src="HDSKOL_0001:Bet" id="TRSKOL" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="HDSKOL_0001:KostSt" id="TRSKO1" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="HDSKOL_0001:Proc" id="53.0" unique="true"/>
        </attribute>
        <attribute>

```

```

    <idAttribute src="HDSKOL_0001:Bom" id="N" unique="true"/>
    </attribute>
  </attributeContent>
</content>
</contents>
</node>
</scheduleElementRoute>
</scheduleElement>
</scheduleElements>
</orderTemplate>
</msg>
</SUTI>

```

SUTI Message 6800	Kommentar
<pre> <orgSender name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgSender> </pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:Bet"
<pre> <orgReceiver name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgReceiver> </pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr"
msgType="6800"	SUTI nod <msg/>
msgName="requestedOrderInformation"	SUTI nod <msg/>
<pre> <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/> </pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
orderTemplateName="A1"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>
src="PLANET_0001:BestNr" id="486"	SUTI nod <idOrder/>
	Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/> Om svaret är OK skickas inte noden <idRejectReason/>
	Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/> Om svaret är OK skickas inte noden <idRejectReason/>
src="HDSKOL_0001:KId" id="010101-1234"	SUTI nod <idContent/>
name="KALLE SVENSSON"	SUTI nod <content/>
time="2013-08-20T08:30:00"	SUTI nod <time/> HTid eller LTid bestäms av nodeType
addressName="VILLAGATAN 3"	SUTI nod <addressNode/> HAdr eller LAdr bestäms av nodeType
src="HDSKOL_0001:HOmr" id="TRC002"	SUTI nod <idZone/>
time="2013-08-20T08:45:00"	SUTI nod <time/> HTid eller LTid bestäms av nodeType
addressName="TROLLSKOLAN"	SUTI nod <addressNode/> HAdr eller LAdr bestäms

	av nodeType
src="HDSKOL_0001:Bet" id="TRSKOL"	SUTI nod <idAttribute/>
src="HDSKOL_0001:KostSt" id="TRSKO1"	SUTI nod <idAttribute/>
src="HDSKOL_0001:Proc" id="53.0"	SUTI nod <idAttribute/>
src="HDSKOL_0001:Bom" id="N"	SUTI nod <idAttribute/>

2.2.14 SUTI message RequestGeograhpyinformation 1050

To be implemented later

2.2.15 SUTI message TransferGeograhpyinformation 1051

To be implemented later

2.2.16 SUTI message Keepalive 7000

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="7000" msgName="Keepalive">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
  </msg>
</SUTI>
```

SUTI Message 7000	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges.
msgType="7000"	SUTI nod <msg/>
msgName="Keepalive"	SUTI nod <msg/>
<pre><idMsg src="HDSKOL_0001 :MsgId" id="20131125NNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken

2.2.17 SUTI message Keepalive 7001

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="7001" msgName="KeepAliveConfirmation">
    <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
    <referencesTo>
      <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
      <!-- idRejectReason skickas vid eventuella fel -->
      <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/>
    </referencesTo>
  </msg>
</SUTI>
```

SUTI Message 7001	Kommentar
<pre><orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender></pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre><orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver></pre>	Information om vem som tar emot meddelandet. Attribut name och id anges.
msgType="7001"	SUTI nod <msg/>
msgName="KeepAliveConfirmation"	SUTI nod <msg/>
<pre><idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/></pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
	<p>Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/></p> <p>Om svaret är OK skickas inte noden <idRejectReason/></p>
	<p>Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/></p> <p>Om svaret är OK skickas inte noden <idRejectReason/></p>

2.2.18 SUTI message RequestForOrderInfo 2540

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2540" msgName="RequestForOrderInfo">
    <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurIdNr" id="238" unique="true" orderTemplateName="A1"/>
      <orderTemplateCalendar/>
      <scheduleElements>
        <scheduleElement scheduleElementSequenceNbr="1">
          <!-- Fasta dagar anges ej med datum utan som nedan -->
          <scheduleElementWeekday monday="true"/>
          <scheduleElementReferencesTo>
            <idOrder src="PLANET_0001:BestNr" id="3412" unique="false"/>
            <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
          </scheduleElementReferencesTo>
          <scheduleElementRoute>
            <!-- nodeType="1803" => "pickup" -->
            <node nodeSeqno="1" nodeType="1803">
              <addressNode addressName="ABC-GATAN 5">
                <idZone src="HDSKOL_0001:Omr" id="OMR918" unique="true" />
              </addressNode>
              <timesNode>
                <!-- Fast måndag = 1899-12-31, Fast tisdag = 1900-01-01 osv. -->
                <time timeZone="1" time="1899-12-31T08:20:00" timeType="2101"/>
              </timesNode>
              <contents>
                <!-- contentType="1001" => traveller -->
                <content contentType="1001" name="empty">
                  <resourceContent>
                    <vehicle>
                      <capacity>
                        <seats noOfSeats="5"/>
                      </capacity>
                    </vehicle>
                  </resourceContent>
                </content>
              </contents>
            </node>
            <!-- nodeType="1804" => "destination" -->
            <node nodeSeqno="2" nodeType="1804">
              <addressNode addressName="SKOLGATAN 23">
                <idZone src="HDSKOL_0001:Omr" id="OMR311" unique="true" />
              </addressNode>
              <timesNode>
                <time timeZone="1" time="1899-12-31T08:55:00" timeType="2101"/>
              </timesNode>
              <contents>
                <!-- contentType="1001" => traveller -->
                <content contentType="1001" name="empty">
                  <resourceContent>
                    <vehicle>
                      <capacity>
                        <seats noOfSeats="5"/>
                      </capacity>
                    </vehicle>
                  </resourceContent>
                </content>
              </contents>
            </node>
          </scheduleElementRoute>
        </scheduleElement>
      </scheduleElements>
    </orderTemplate>
  </msg>
</SUTI>

```

```

    </resourceContent>
  </content>
</contents>
</node>
</scheduleElementRoute>
</scheduleElement>
</scheduleElements>
</orderTemplate>
</msg>
</SUTI>

```

SUTI Message 2540	Kommentar
<pre> <orgSender name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgSender> </pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:MsgId"
<pre> <orgReceiver name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgReceiver> </pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:TurlIdNr"
msgType="2540"	SUTI nod <msg/>
msgName="RequestForOrderInfo"	SUTI nod <msg/>
<pre> <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNN" unique="true"/> </pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
monday="true"	<p>SUTI nod <scheduleElementWeekday/></p> <p>För fasta dagar anges inte datum utan noden <scheduleElementWeekday/> används där den angivna dagen har värdet = "true"</p> <p>För icke fasta dagar används noden <scheduleElementDate/> och datumet anges som date="2013-08-20"</p>
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>
src="PLANET_0001:BestNr" id="3412"	SUTI nod <idOrder/>
orderTemplateName="A1"	SUTI nod <idOrderTemplate/>
noOfSeats="5"	SUTI nod <seats/>
addressName="ABC-GATAN 5"	SUTI nod <addressNode/>
src="HDSKOL_0001:Omr" id="OMR918"	SUTI nod <idZone/>
time="1899-12-31T08:20:00"	<p>SUTI nod <time/></p> <p>Attributet time skall anges som datum + tid har löst detta genom att ange fast måndag = 1899-12-31 fast tisdag = 1900-01-01 fast onsdag = 1900-01-02 fast torsdag = 1900-01-03 fast fredag = 1900-01-04 fast lördag = 1900-01-05 fast söndag = 1900-01-06</p>
addressName="SKOLGATAN 23"	SUTI nod <addressNode/>
src="HDSKOL_0001:Omr" id="OMR311"	SUTI nod <idZone/>
time="1899-12-31T08:55:00"	<p>SUTI nod <time/></p> <p>Attributet time skall anges som datum + tid har löst</p>

	detta genom att ange fast måndag = 1899-12-31 fast tisdag = 1900-01-01 fast onsdag = 1900-01-02 fast torsdag = 1900-01-03 fast fredag = 1900-01-04 fast lördag = 1900-01-05 fast söndag = 1900-01-06
--	--

2.2.19 SUTI message OrderInfo 2541

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<SUTI>
  <orgSender name="HDSKOL">
    <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="PLANET">
    <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="2541" msgName="RequestForOrderInfo">
    <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
    <referencesTo>
      <!-- msgid that this message replies to -->
      <idMsg src="PLANET_0001:MsgId" id="20131125NNNNNNNNN" unique="true"/>
      <!-- idRejectReason skickas vid eventulla fel -->
      <idRejectReason src="PLANET_0001:Txt" id="Eventuell fel text" unique="false"/>
    </referencesTo>
    <orderTemplate>
      <idOrderTemplate src="HDSKOL_0001:TurlIdNr" id="238" unique="true" orderTemplateName="A1"/>
    </orderTemplateCalendar/>
    <scheduleElements>
      <scheduleElement scheduleElementSequenceNbr="1">
        <!-- Fasta dagar anges ej med datum utan som nedan -->
        <scheduleElementWeekday monday="true"/>
        <scheduleElementReferencesTo>
          <idOrder src="PLANET_0001:BestNr" id="3412" unique="false"/>
          <idOrder src="HDSKOL_0001:HD_OrderNr" id="27641" unique="true"/>
        </scheduleElementReferencesTo>
        <scheduleElementRoute>
          <!-- nodeType="1803" => "pickup" -->
          <node nodeSeqno="1" nodeType="1803">
            <addressNode addressName="ABC-GATAN 5">
              <idZone src="HDSKOL_0001:Omr" id="OMR918" unique="true" />
            </addressNode>
            <timesNode>
              <!-- Fast måndag = 1899-12-31, Fast tisdag = 1900-01-01 osv. -->
              <time timeZone="1" time="1899-12-31T08:20:00" timeType="2101"/>
            </timesNode>
            <contents>
              <!-- contentType="1001" => traveller -->
              <content contentType="1001" name="">
                <resourceContent>
                  <vehicle>
                    <capacity>
                      <seats noOfSeats="5"/>
                    </capacity>
                  </vehicle>
                </resourceContent>
              </content>
            </contents>
          </node>
          <!-- nodeType="1804" => "destination" -->
          <node nodeSeqno="2" nodeType="1804">
            <addressNode addressName="SKOLGATAN 23">
              <idZone src="HDSKOL_0001:Omr" id="OMR311" unique="true" />
            </addressNode>
            <timesNode>
              <time timeZone="1" time="1899-12-31T08:55:00" timeType="2101"/>
            </timesNode>
            <contents>
              <!-- contentType="1001" => traveller -->
              <content contentType="1001" name="">

```

```

    <resourceContent>
      <vehicle>
        <capacity>
          <seats noOfSeats="5"/>
        </capacity>
      </vehicle>
    </resourceContent>
  </content>
</contents>
</node>
</scheduleElementRoute>
</scheduleElement>
</scheduleElements>
</orderTemplate>
</msg>
</SUTI>

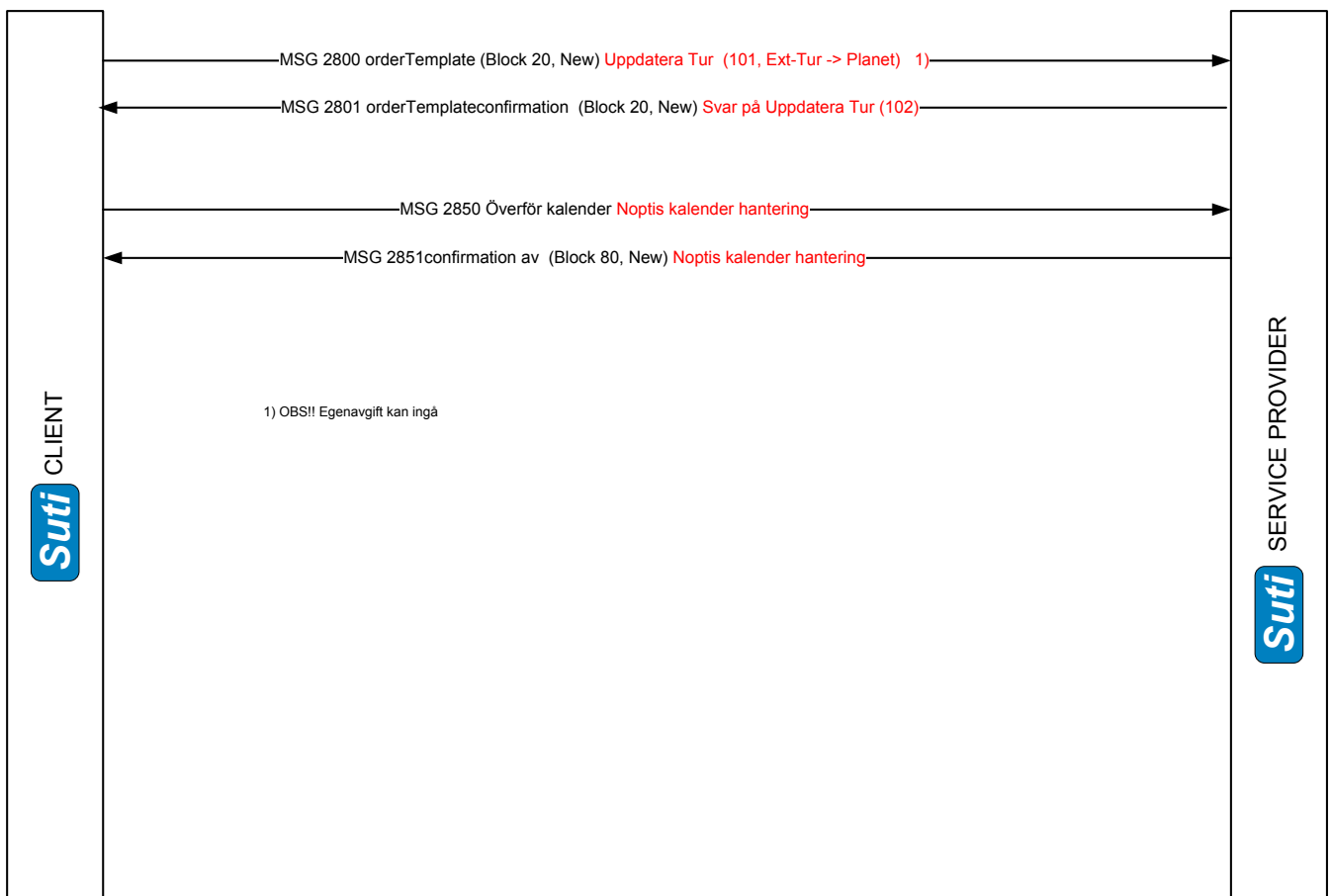
```

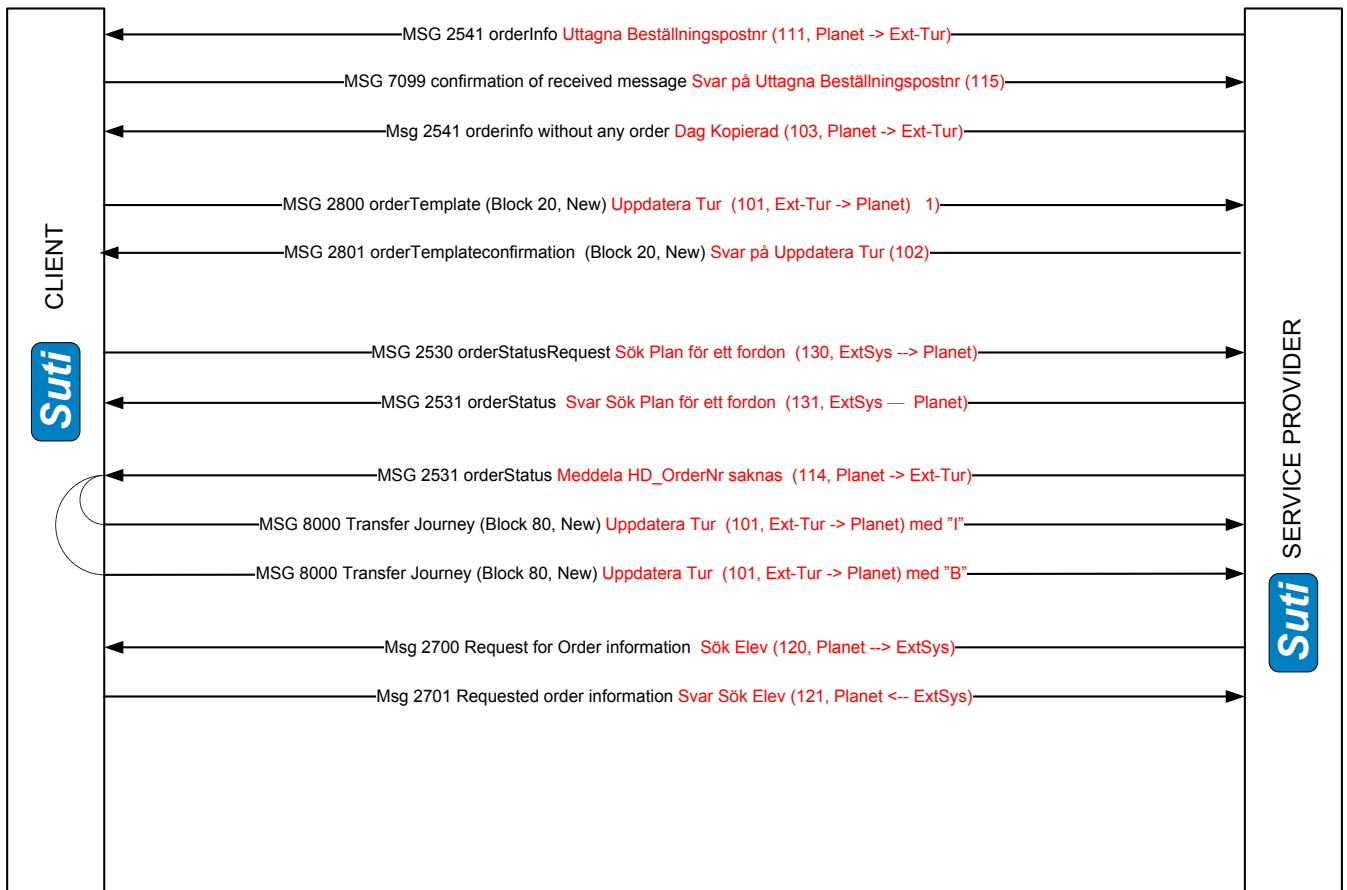
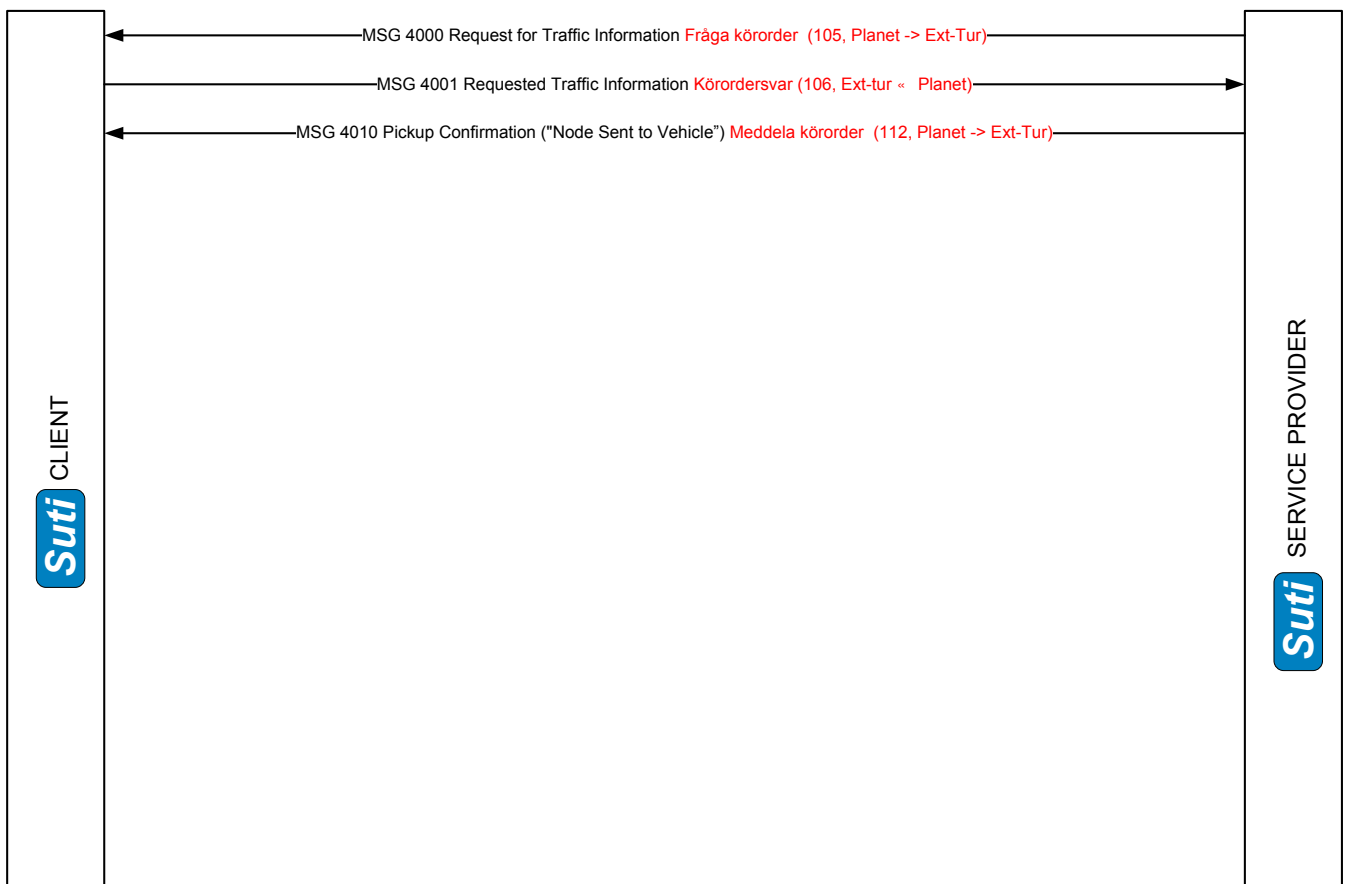
SUTI Message 2541	Kommentar
<pre> <orgSender name="HDSKOL"> <idOrg src="SUTI:idLink" id="HDSKOL_0001" unique="true"/> </orgSender> </pre>	Information om vem som skickar meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="HDSKOL_0001:MsgId"
<pre> <orgReceiver name="PLANET"> <idOrg src="SUTI:idLink" id="PLANET_0001" unique="true"/> </orgReceiver> </pre>	Information om vem som tar emot meddelandet. Attribut name och id anges. Attributet id används sedan i src för att tala om vem som "äger" parametern ex. src="PLANET_0001:BestNr"
msgType="2541"	SUTI nod <msg/>
msgName="OrderInfo"	SUTI nod <msg/>
<pre> <idMsg src="HDSKOL_0001:MsgId" id="20131125NNNNNNNN" unique="true"/> </pre>	Unikt id på aktuellt meddelande, anges som datum + nummer på 8 tecken
	<p>Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/></p> <p>Om svaret är OK skickas inte noden <idRejectReason/></p>
	<p>Om svaret INTE är OK skickas <idRejectReason src="HDSKOL_0001:Txt" id="Eventuell fel text" unique="false"/></p> <p>Om svaret är OK skickas inte noden <idRejectReason/></p>
monday="true"	<p>SUTI nod <scheduleElementWeekday/></p> <p>För fasta dagar anges inte datum utan noden <scheduleElementWeekday/> används där den angivna dagen har värdet = "true"</p> <p>För icke fasta dagar används noden <scheduleElementDate/> och datumet anges som date="2013-08-20"</p>
src="HDSKOL_0001:TurlIdNr" id="238"	SUTI nod <idOrderTemplate/>
src="HDSKOL_0001:HD_OrderNr" id="27641"	SUTI nod <idOrder/>
src="PLANET_0001:BestNr" id="3412"	SUTI nod <idOrder/>
orderTemplateName="A1"	SUTI nod <idOrderTemplate/>
noOfSeats="5"	SUTI nod <seats/>

addressName="ABC-GATAN 5"	SUTI nod <addressNode/>
src="HDSKOL_0001:Omr" id="OMR918"	SUTI nod <idZone/>
time="1899-12-31T08:20:00"	SUTI nod <time/> Attributet time skall anges som datum + tid har löst detta genom att ange fast måndag = 1899-12-31 fast tisdag = 1900-01-01 fast onsdag = 1900-01-02 fast torsdag = 1900-01-03 fast fredag = 1900-01-04 fast lördag = 1900-01-05 fast söndag = 1900-01-06
addressName="SKOLGATAN 23"	SUTI nod <addressNode/>
src="HDSKOL_0001:Omr" id="OMR311"	SUTI nod <idZone/>
time="1899-12-31T08:55:00"	SUTI nod <time/> Attributet time skall anges som datum + tid har löst detta genom att ange fast måndag = 1899-12-31 fast tisdag = 1900-01-01 fast onsdag = 1900-01-02 fast torsdag = 1900-01-03 fast fredag = 1900-01-04 fast lördag = 1900-01-05 fast söndag = 1900-01-06

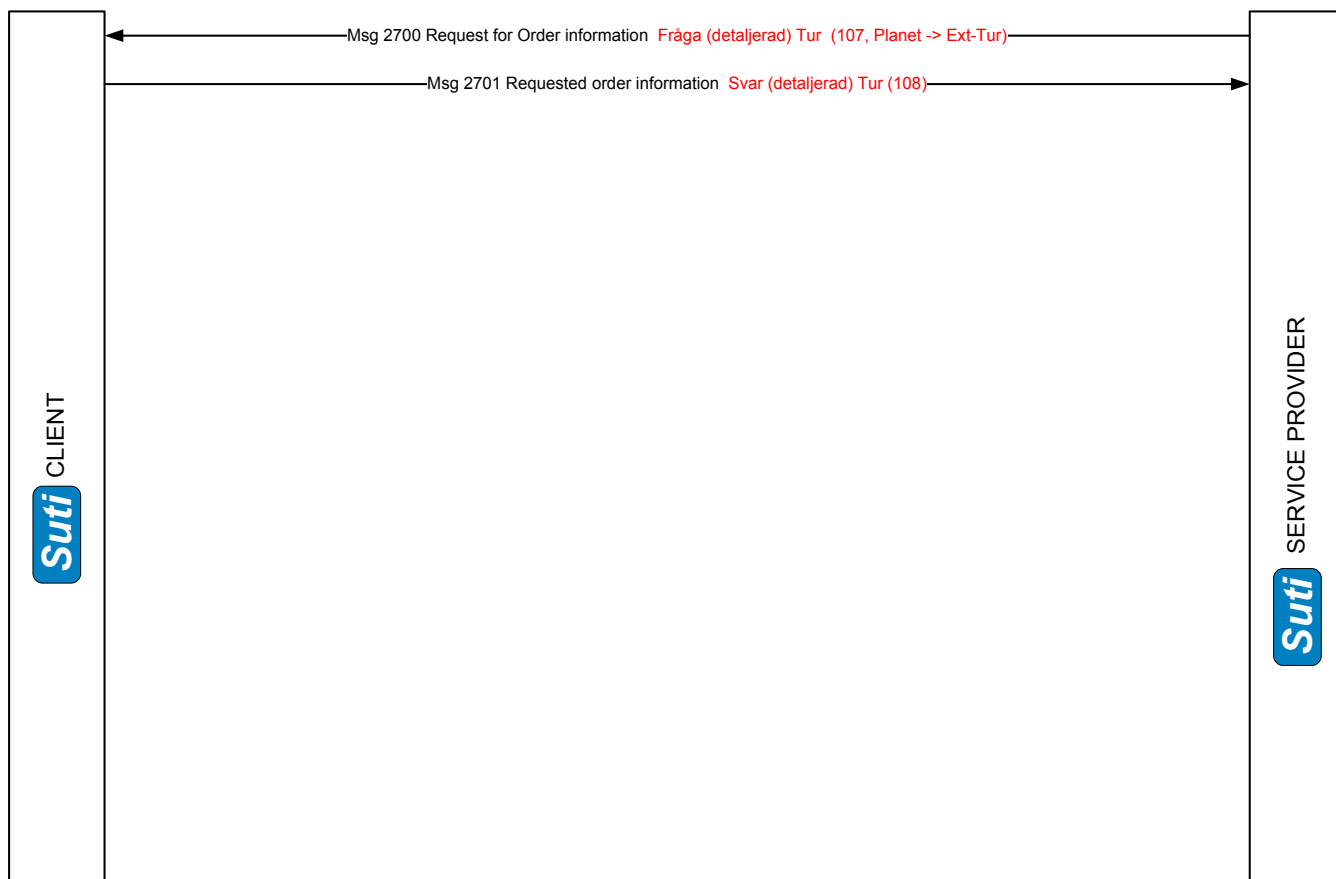
2.2.20 Flowcharts

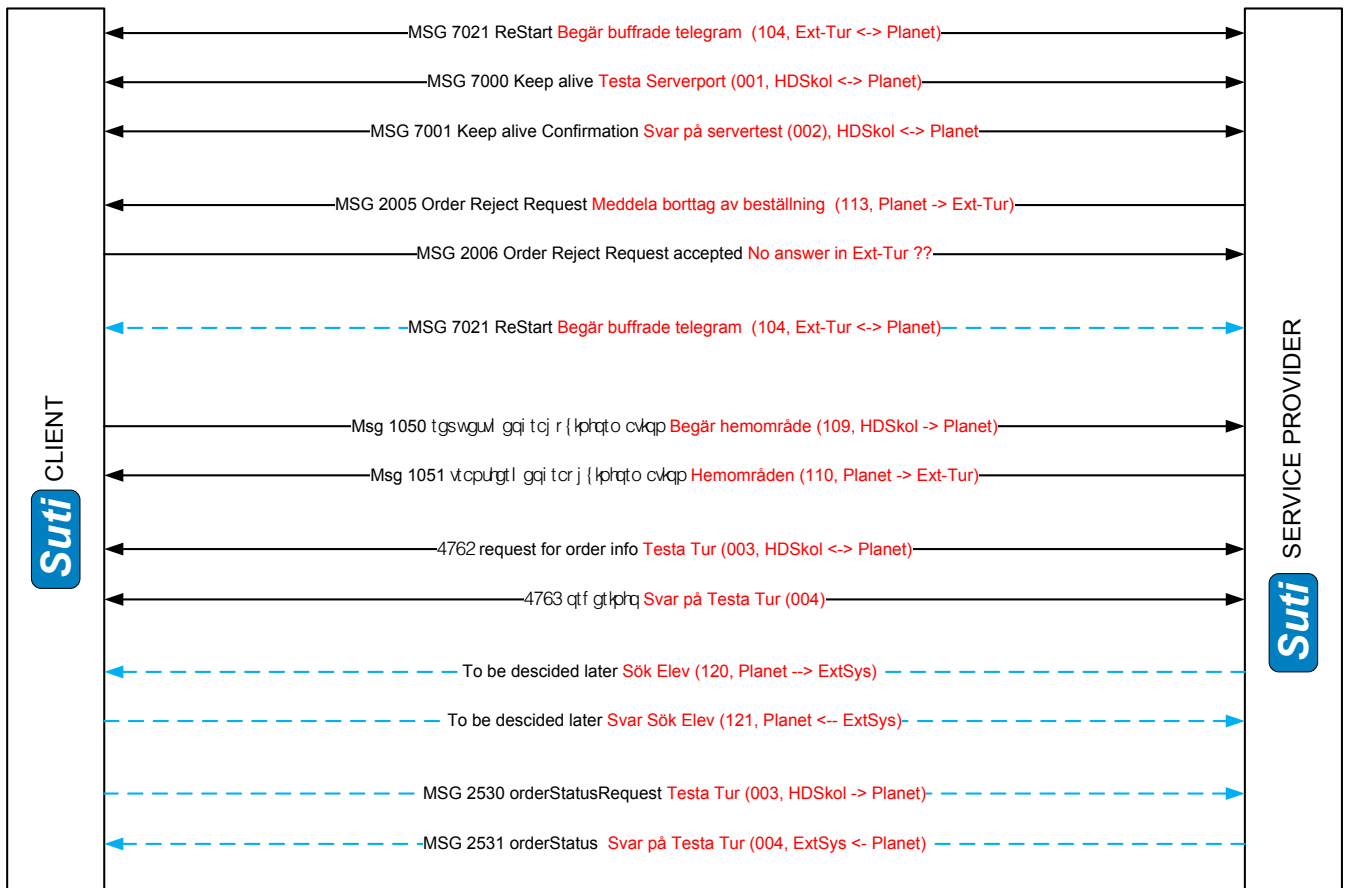
Förbokning Schema



Aktuella Förbokningar**Körorder**

Fakturaunderlag



Övrigt**2.2.21 Ej implementerade delar**

Sök elev telegram 120

Ej implementerat

Ersätts av SUTI message ?

Ej implementerat

Sök plan för ett fordon telegram 130

Ej implementerat

Ersätts av SUTI message ?

2.3 SUTI flagstops

På SUTI konferencen i februar 2010 og i september 2010 diskuterede vi de nye telegrammer 2900 Authorization request og 2901 Authorization accept. Telegrammerne er beskrevet i dokumentet V2_3_EconomicalTransactions.pdf.

Trapeze vil gerne anvende disse telegrammer til en flagstop logik, og det kræver at der tilføjes et content element som beskriver kunden i Msg2900. Desuden er det nødvendigt med en ny subnodeType ved navn flagstop.

2.3.1 Flagstop forløb

En kunde kommer ind i en taxi og kører sit smartcard igennem en kortlæser i taxien. Kortnummeret sendes til et client system som svarer tilbage med info om kunden og accept for om kunden kan køre en tur. Info skal bestå af kundens navn, adresse og saldo på kundens konto.

Hvis kunden ikke findes i client systemet eller kunden ikke må køre, svares der tilbage med en afvisning og en årsag til afvisningen.

Hvis kunden er godkendt til at køre en tur, samler taxien kunden op og sender de geografiske koordinater, kundenr og smartcardId til client systemet. Det svarer til at sende en enkelt node til client systemet, og noden udføres med det samme.

Når kunden senere sættes af, sendes node 2 som er kundens destination. I client systemet kan de to noder nu samles til en booking og prisen på turen kan bogføres på kundens konto.

2.3.2 Flagstops i SUTI

2.3.2.1 Authorization

Flagstop forløbet består af to dele. Den første del er en autorisation, hvor en kunde godkendes til at køre en tur. Her kan vi anvende Msg2900 Authorization Request.

Indholdet i telegrammet skal være smartcardId (et kortnr). Det placeres under referencesTo i idAuthorization:

```
<referencesTo>
  <idAuthorization id="12345678" src="trapezesoftware_Skånetrafiken_0001:smartCardId"></idAuthorization>
</referencesTo>
```

Msg2900 besvares med Msg2901 Authorization Accept. Til dette telegram skal der tilføjes et content element (SUTI/Msg/AuthorizationAccept/Content), så man kan angive en kundes navn og øvrige oplysninger som hører hjemme under content. Kundens hjemmeadresse og saldoen på kundens konto kan angives under manualDescriptionContent.

2.3.2.2 Bookingen

Den anden halvdel af flagstoppet er en booking som sammensættes af to nodes. Den første node dannes ud fra den geografiske position, hvor kunden stiger ind i bilen, og den anden node er adressen, hvor kunden sættes af. Pickup noden kan angives i order elementet i Msg2900 Authorization request, og i subnodeType angives det at det drejer sig om et flagstop:

```
<node nodeSeqno="1" nodeType="pickup" subnodeType="flagstop">
```

Efter at bilen modtager en Authorization Accept, sendes en Msg4010 Event Confirmation som bekræfter at kunden nu er i bilen og er klar til at køre.

Dropoff noden sendes fra bilen til clientsystemet, når kunden stiger ud af bilen. Det er en Msg4010 Event Confirmation hændelse igen med flagstop som subNodeType:

```
<pickupConfirmation eventType="passengerdropped">
  <nodeConfirmed nodeType="destination" nodeSeqno="2" subnodeType="flagstop">
```

Når clientsystemet har modtaget flagstop destination noden, sender clientsystemet Msg4020 End of order. Og dermed er kommunikationen vedrørende flagstoppet slut.

Hvis der i Msg2901 Authorization request er angivet deliveryNote under process elementet:

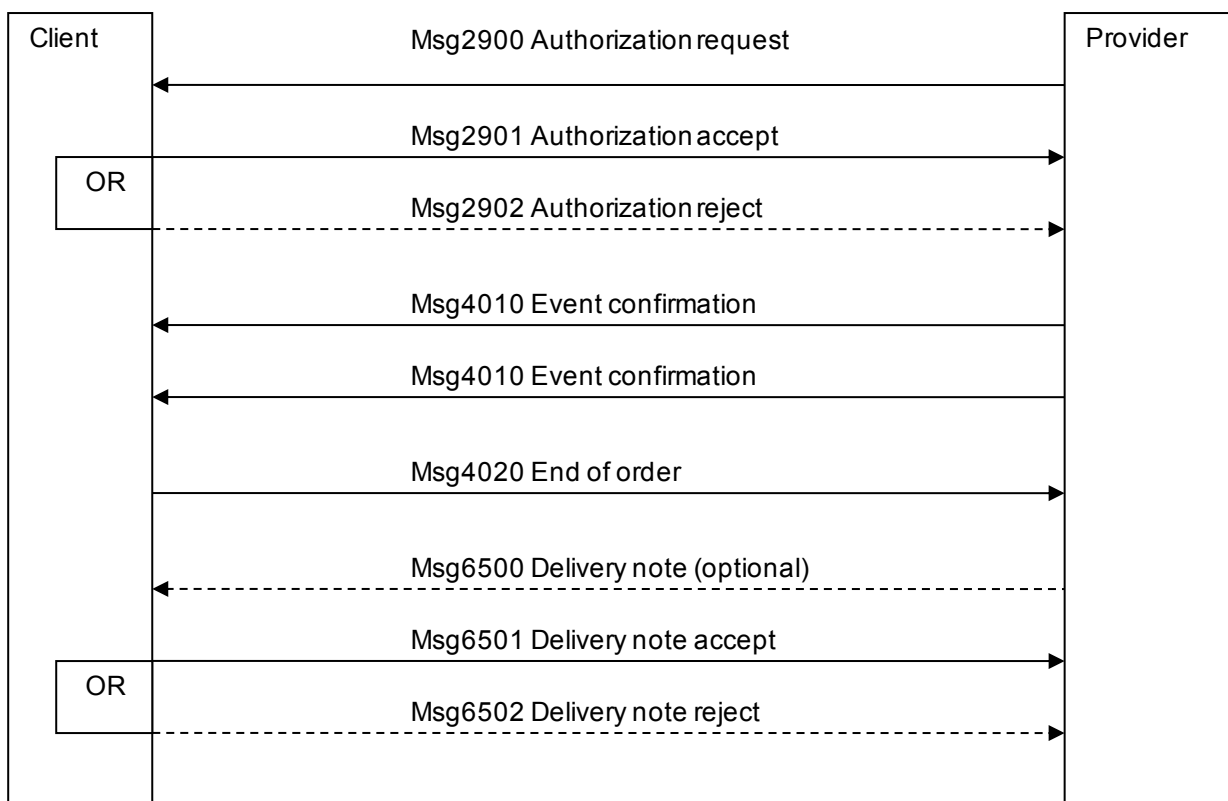
```
<process allowRouting="false" trafficControl="true" dispatchResponsible="client" report="false" preorderedVehicle="false"
dispatch="true" manualDispatch="false" deliveryNote="true"></process>
```

Så udvides flagstop kommunikationen med en Msg6500 deliveryNote og et tilhørende svar.

Se eksempler på telegrammer i filerne:

SUTI_Msg2900 Authorization request.xml
 SUTI_Msg2901 Authorization Accept.xml
 Msg4010_EventConfirmation_pickup.xml
 Msg4010_EventConfirmation_destination.xml

Her er en skitse af message flow:



2.4 Delivery Note and associated messages

2.4.1 How to request a Msg 6500 Delivery Note

A Msg 6500 can be requested in the following 3 ways.

2.4.1.1 Request for Msg 6500 by using process element in a Msg 2000

A Msg 6500 can be requested by using the attribute deliveryNoteRequest in Msg 2000 Order in the process element.

```
<process manualDispatch="false" dispatch="true" dispatchResponsible="provider" trafficControl="false"
report="false" preorderedVehicle="false" allowRouting="false" deliveryNoteRequest="true"/>
```

2.4.1.2 Request for Msg 6500 by using process element in a Msg 2901

A Msg 6500 can be requested by using the attribute deliveryNoteRequest in Msg 2901 Order Authorization accept in the process element.

```
<process manualDispatch="false" dispatch="true" dispatchResponsible="provider" trafficControl="false"
report="false" preorderedVehicle="false" allowRouting="false" deliveryNoteRequest="true"/>
```

2.4.1.3 Request for Msg 6500 by sending a Msg 6510

A Msg 6500 can be requested by sending a MSG 6510 deliveryNoteRequest to the provider.

```
<?xml version="1.0" encoding="UTF-8"?>
<SUTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SUTI_MessageXSD_2_4_X_Inprogress.xsd">
  <orgSender name="Skånetrafiken">
    <idOrg src="SUTI:idlink" id="trapezesoftware_Skånetrafiken_0003" unique="true"/>
  </orgSender>
  <orgReceiver name="TaxiBilSyd">
    <idOrg src="SUTI:idlink" id="Holmedal_TaxiBilSyd_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="6510" msgName="deliveryNoteRequest">
    <idMsg src="Trapezesoftware_Skånetrafiken_0003:MsgId" id="2011011312345678" unique="true"/>
    <referencesTo>
      <idMsg src="Holmedal_TaxiBilSyd_0001:MSGID" id="2011011315096979" unique="true"/>
      <idVehicle src="Holmedal_TaxiBilSyd_0001:VehicleNumber" id="1808" unique="true"/>
    </referencesTo>
  </msg>
</SUTI>
```

2.4.1.4 Responding to a Msg 6510 with a Msg 6511

If the client requests a Msg 6500 by using a Msg 6510 the provider has the option to answer by sending a Msg 6511 deliveryNoteRequestReject. A Msg 6511 shall be sent immediately after a Msg 6510 has been received.

```
<?xml version="1.0" encoding="UTF-8"?>
<SUTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SUTI_MessageXSD_2_4_X_Inprogress.xsd">
  <orgSender name="TaxiBilSyd">
    <idOrg src="SUTI:idlink" id="Holmedal_TaxiBilSyd_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="Skånetrafiken">
    <idOrg src="SUTI:idlink" id="trapezesoftware_Skånetrafiken_0003" unique="true"/>
  </orgReceiver>
  <msg msgType="6511" msgName="deliveryNoteRequestReject">
    <idMsg src="Holmedal_TaxiBilSyd_0001:MSGID" id="2011011315096980" unique="true"/>
    <referencesTo>
      <idMsg src="trapezesoftware_Skånetrafiken_0003:MsgId" id="2011011312345678" unique="true"/>
      <idVehicle src="Holmedal_TaxiBilSyd_0001:VehicleNumber" id="1808" unique="true"/>
    </referencesTo>
  </msg>
</SUTI>
```

2.4.2 Sending a Msg 6500 Delivery Note

If requested a Msg 6500 shall be sent within 1 business day. This time limit can be altered if a mutual agreement between Client and Provider exists.

```
<?xml version="1.0" encoding="UTF-8"?>
<SUTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SUTI_MessageXSD_2_4_X_Inprogress.xsd">
  <orgSender name="TaxiBilSyd">
    <idOrg src="SUTI:idlink" id="Holmedal_TaxiBilSyd_0001" unique="true"/>
  </orgSender>
  <orgReceiver name="Skånetrafiken">
    <idOrg src="SUTI:idlink" id="trapezesoftware_Skånetrafiken_0003" unique="true"/>
  </orgReceiver>
  <msg msgType="6500" msgName="deliveryNote">
    <idMsg src="Holmedal_TaxiBilSyd_0001:MSGID" id="2011011315096979" unique="true"/>
    <deliveryNote>
      <idReceipt id="12345" src="trapezesoftware_Skånetrafiken_0003:FakturaNr"></idReceipt>
      <eventReportDeliveryNote>
        <event eventType="pickup">
          <subOrderEvent>
            <idOrder id="12345" src="trapezesoftware_Skånetrafiken_0003:BookingId"></idOrder>
          </subOrderEvent>
          <eventContent contentType="companion" name="unknown">
            <resourceContent>
              <vehicle>
                <capacity>
                  <seats noOfSeats="2" noOfItems="2"></seats>
                </capacity>
              </vehicle>
            </resourceContent>
          </eventContent>
        </event>
      </eventReportDeliveryNote>
      <summaryReportDeliveryNote orderEnded="true"></summaryReportDeliveryNote>
      <economyReportDeliveryNote>
        <!--Commentary: <economyReportDeliveryNote> WITHOUT nodeSeqno means that this is reported TOTAL-->
        <formOfPayment>
          <payment paymentType="account" amount="125.00">
            <idCurrency src="ISO 4217" id="SEK" unique="false"/>
          </payment>
        </formOfPayment>
      </economyReportDeliveryNote>
    </deliveryNote>
  </msg>
</SUTI>
```

```

        <vatAmountSpecification>
          <totalVatAmount amountVat="25" amountInclVat="125" distance="23" duration="32">
            <idAmount id="Total" src="Holmedal_TaxiBilSyd_0001:AmountId"></idAmount>
          </totalVatAmount>
        </vatAmountSpecification>
      </payment>
      <payment paymentType="socialservicefee" amount="50"></payment>
    </formOfPayment>
  </economyReportDeliveryNote>
  <economyReportDeliveryNote nodeSeqno="2">
<!--Commentary: <economyReportDeliveryNote> WITH nodeSeqno means that this is presented a NOD. Note!
Accounting should be done in the node where the traveller leaves the vehicle -->
    <formOfPayment>
      <payment paymentType="socialservicefee" amount="25.00">
        <idCurrency src="ISO 4217" id="SEK" unique="false"/>
      </payment>
    </formOfPayment>
  </economyReportDeliveryNote>
  <economyReportDeliveryNote nodeSeqno="4">
<!--Commentary: <economyReportDeliveryNote> WITH nodeSeqno means that this is presented a NOD. Note!
Accounting should be done in the node where the traveller leaves the vehicle -->
    <formOfPayment>
      <payment paymentType="socialservicefee" amount="25.00">
        <idCurrency src="ISO 4217" id="SEK" unique="false"/>
      </payment>
    </formOfPayment>
  </economyReportDeliveryNote>
</deliveryNote>
</msg>
</SUTI>

```

2.4.3 Answering a Msg 6500

A Msg 6500 shall be answered within 1 business day. It's possible to answer with Msg 6501 deliveryNoteAccept, Msg 6502 deliveryNoteReject or Msg 6503 deliveryNoteReject.

2.4.3.1 Msg 6501 deliveryNoteAccept

Msg 6501 shall be sent if the client accepts the information included in Msg 6500.

```

<?xml version="1.0" encoding="UTF-8"?>
<SUTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SUTI_MessageXSD_2_4_X_Inprogress.xsd">
  <orgSender name="Skånetrafiken">
    <idOrg src="SUTI:idlink" id="trapezesoftware_Skånetrafiken_0003" unique="true"/>
  </orgSender>
  <orgReceiver name="TaxiBilSyd">
    <idOrg src="SUTI:idlink" id="Holmedal_TaxiBilSyd_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="6501" msgName="deliveryNoteAccept">
    <idMsg src="rapezesoftware_Skånetrafiken_0003:MsgId" id="2011011312345678" unique="true"/>
    <referencesTo>
      <idMsg src="Holmedal_TaxiBilSyd_0001:MSGID" id="2011011315096979" unique="true"/>
      <idVehicle src="Holmedal_TaxiBilSyd_0001:VehicleNumber" id="1808" unique="true"/>
    </referencesTo>
  </msg>
</SUTI>

```


2.4.3.2 Msg 6502 deliveryNoteReject

Msg 6501 shall be sent if the client doesn't accept the information included in Msg 6500.

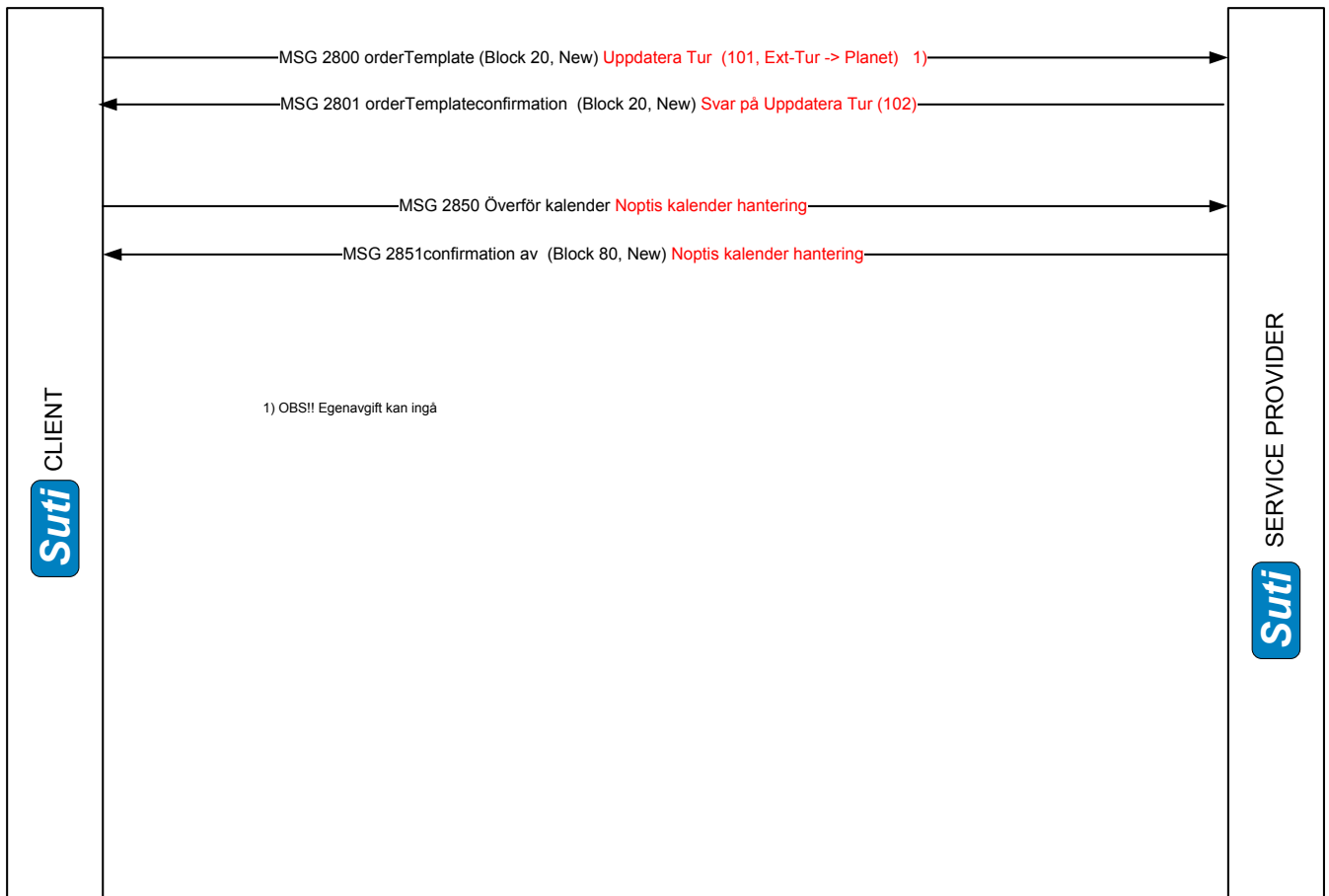
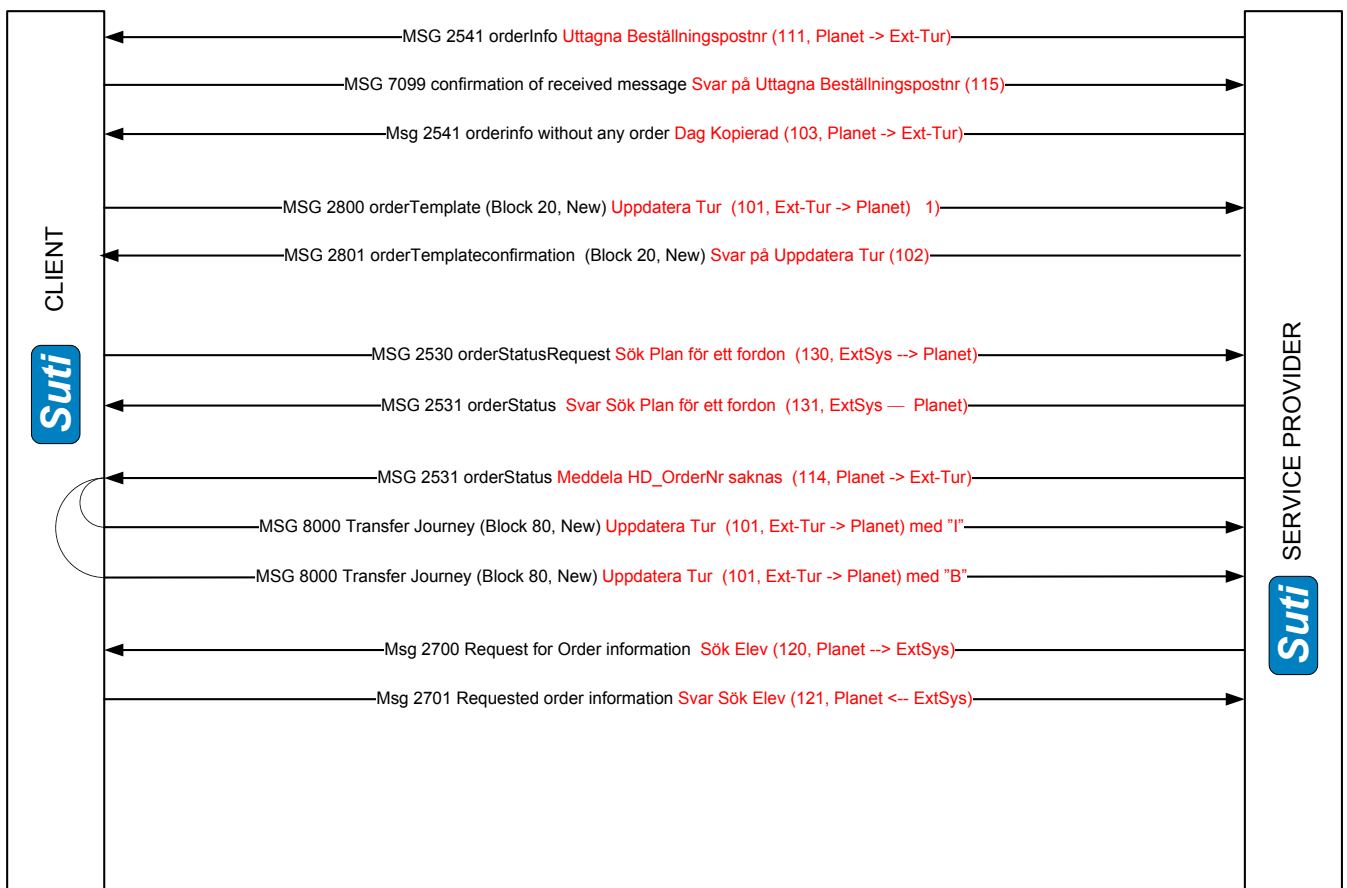
```
<?xml version="1.0" encoding="UTF-8"?>
<SUTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SUTI_MessageXSD_2_4_X_Inprogress.xsd">
  <orgSender name="Skånetrafiken">
    <idOrg src="SUTI:idlink" id="trapezesoftware_Skånetrafiken_0003" unique="true"/>
  </orgSender>
  <orgReceiver name="TaxiBilSyd">
    <idOrg src="SUTI:idlink" id="Holmedal_TaxiBilSyd_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="6502" msgName="deliveryNoteReject">
    <idMsg src="rapezesoftware_Skånetrafiken_0003:MsgId" id="2011011312345678" unique="true"/>
    <referencesTo>
      <idMsg src="Holmedal_TaxiBilSyd_0001:MSGID" id="2011011315096979" unique="true"/>
      <idVehicle src="Holmedal_TaxiBilSyd_0001:VehicleNumber" id="1808" unique="true"/>
      <idRejectReason id="Turen blev ikke kørt." src="Holmedal_TaxiBilSyd_0001:RejectReason"/>
    </referencesTo>
  </msg>
</SUTI>
```

2.4.3.3 Msg 6503 deliveryNoteWait

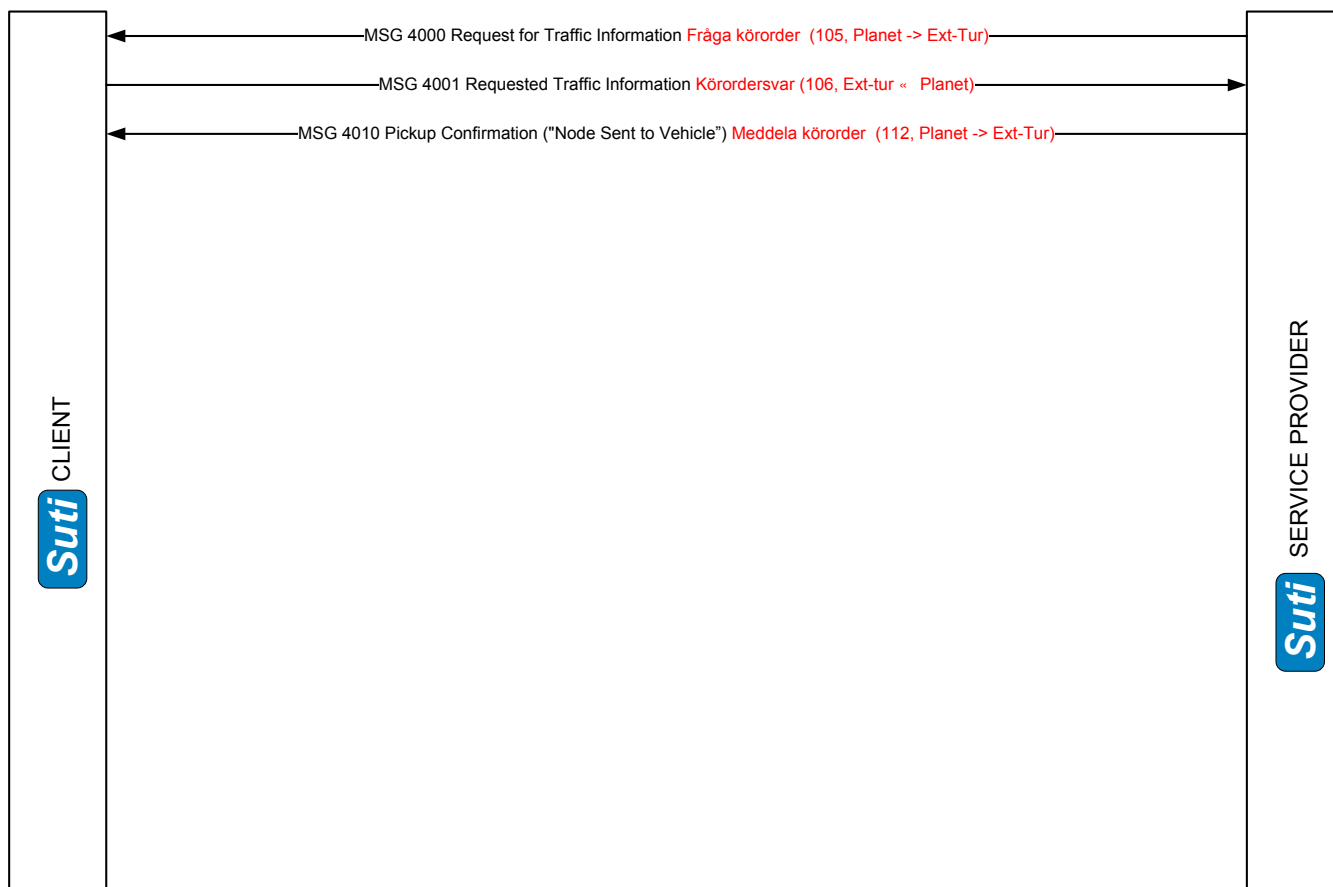
Msg 6501 shall be sent if the client can't answer a msg 6500. The reason for this may include a sample of the message will occur.

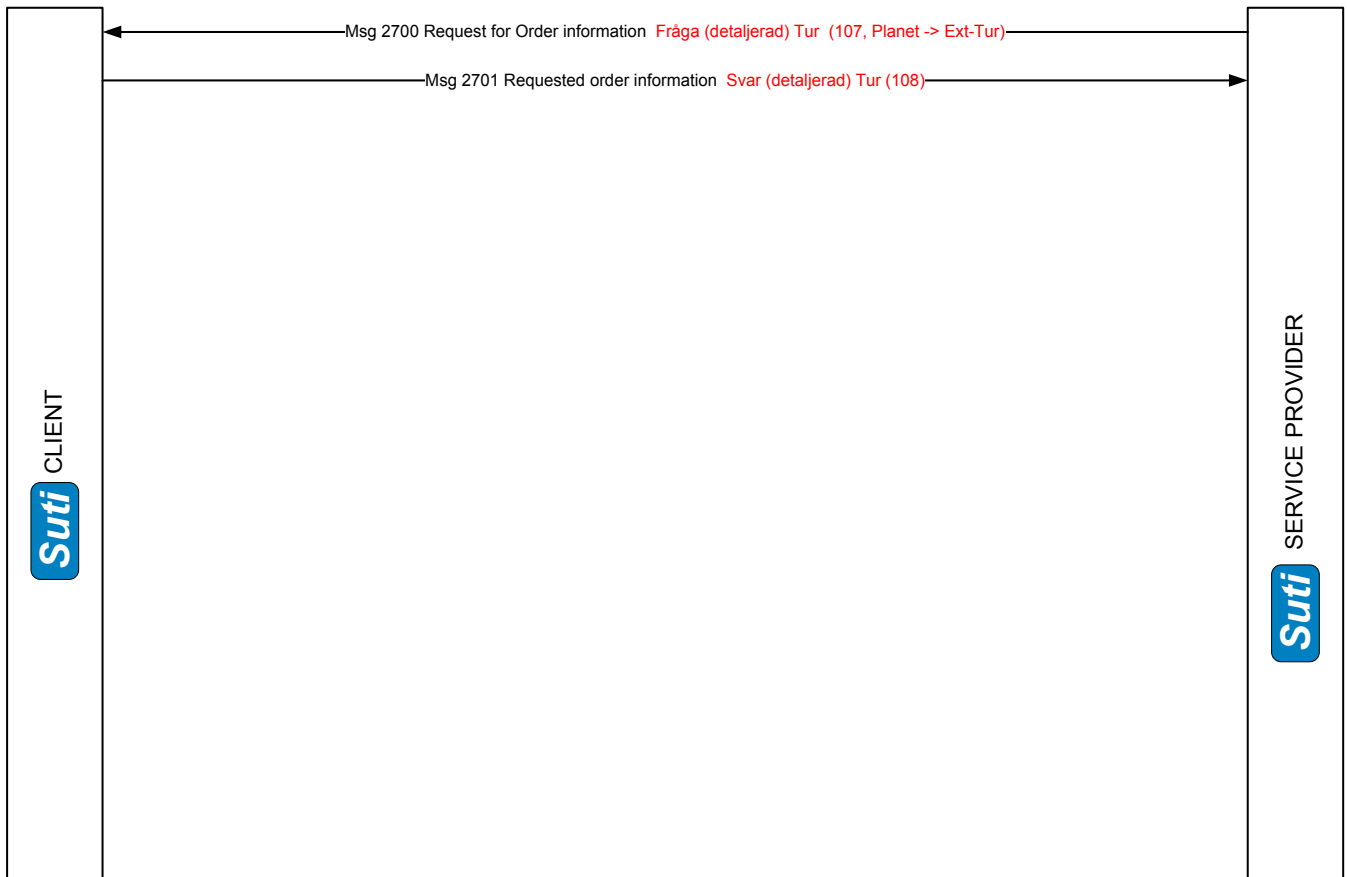
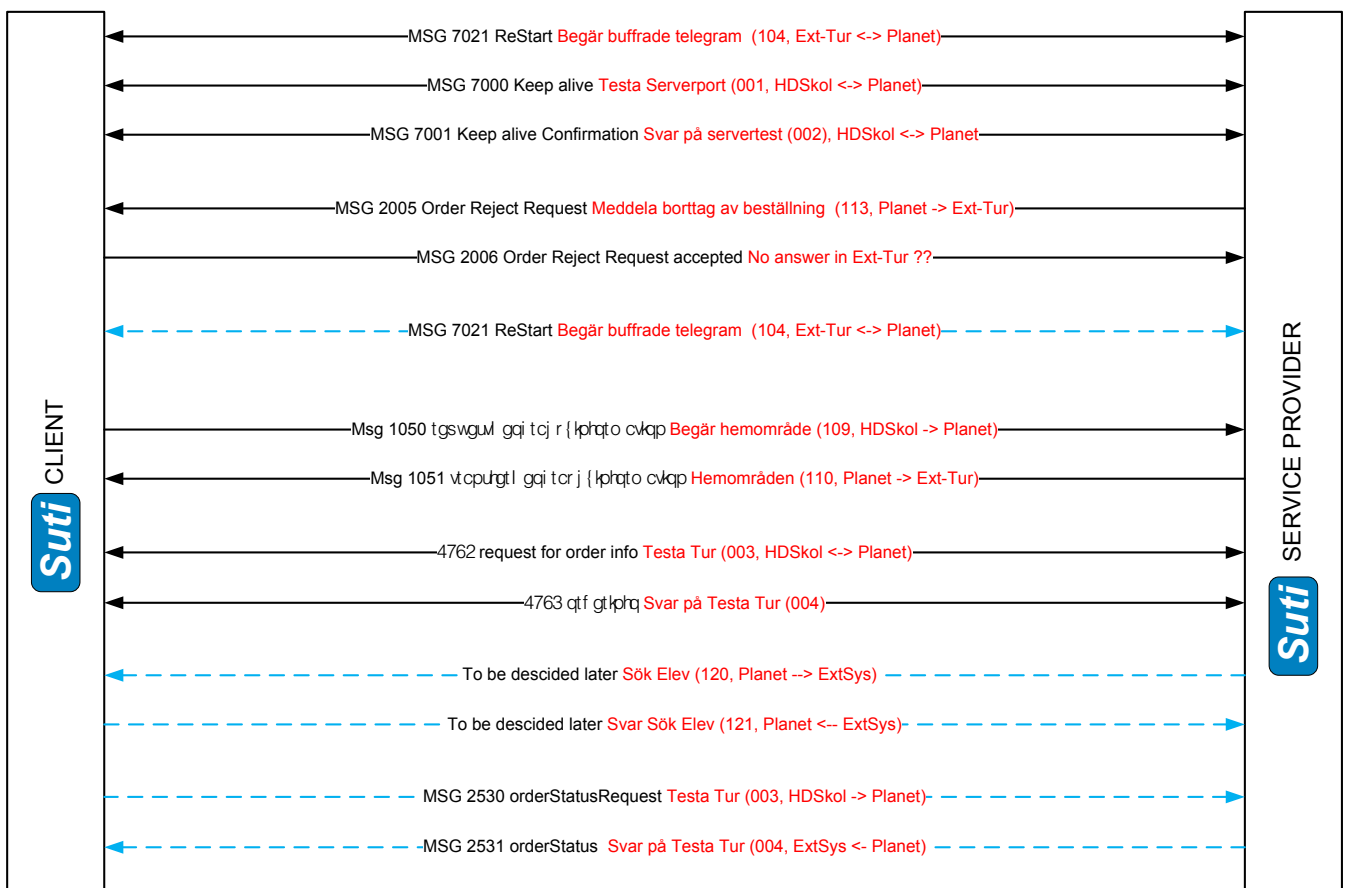
If Msg 6503 is used, a Msg 6501 or Msg 6502 shall be sent within 2 business days after a msg 6503 has been sent. If the client doesn't send a Msg 6501 or Msg 6502 within 2 business days, the provider shall consider the Msg as Approved. This time limit can be altered if a mutual agreement between Client and Provider exists.

```
<?xml version="1.0" encoding="UTF-8"?>
<SUTI xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SUTI_MessageXSD_2_4_X_Inprogress.xsd">
  <orgSender name="Skånetrafiken">
    <idOrg src="SUTI:idlink" id="trapezesoftware_Skånetrafiken_0003" unique="true"/>
  </orgSender>
  <orgReceiver name="TaxiBilSyd">
    <idOrg src="SUTI:idlink" id="Holmedal_TaxiBilSyd_0001" unique="true"/>
  </orgReceiver>
  <msg msgType="6503" msgName="deliveryNoteWait">
    <idMsg src="rapezesoftware_Skånetrafiken_0003:MsgId" id="2011011312345678" unique="true"/>
    <referencesTo>
      <idMsg src="Holmedal_TaxiBilSyd_0001:MSGID" id="2011011315096979" unique="true"/>
      <idVehicle src="Holmedal_TaxiBilSyd_0001:VehicleNumber" id="1808" unique="true"/>
    </referencesTo>
  </msg>
</SUTI>
```

Förbokning Schema**Aktuella Förbokningar**

Körorder



Fakturaunderlag**Övrigt**

2.5 Accounting

2.5.1 Basic Accounting from Provider to Client

2.5.1.1 General

This type of accounting is called Basic accounting since it is called by the Provider as a request for settlement of transports as it is the basic principle for settlement. The accounting is based on an agreement between Client and Provider.

All accounting in SUTI is based on settlement of **tours**. A tour is a transport with a defined vehicle, a defined driver with a start and end. The transport has normally been ordered in advance by the Client and the transport can contain one or more travelers (or other content). The tour is defined in the vehicle as one economic transaction.

The process is a continuous process and most probable on a daily basis.

The process has four clearly differentiated sub processes:

1. Daily process (Invoicing).

The Provider sends an accounting message containing one or several tours. Each tour can contain several suborders. The Client will accept the message and will start an automatic process to either accept or decline the invoice for each tour. Each decision is documented in a response message that shall contain exactly the same tours stating the decision for each tour. All accepted tours can be accounted by the Provider and the Provider shall expect a payment for the accepted tours.

For each tour where the accounting is not accepted there should always be a reason that is documented in an errorCode. The responsibility to further process the tour is now back to the Provider. Normally the Provider corrects the tour and will then resend the tour in a later message (most often together with new tours).

There is an option for the Client to decide that it will not make an automatic decision. In such a case the tour will be sent to a separate process called Reconsideration (see below). In such a case the tour will be neither set as accepted nor declined (rejected). It will be set as validate and the Client has then the responsibility to return with a later decision where it is either accepted or rejected (in a specific type of message).

2. Reconsideration

The process can either be initiated by Client as described above or by Provider if the Provider does not approve with a reject for a tour. This process is normally a process with more manual treatment. Therefore the use of it should be restricted by both parties.

3. Revaluation

This process is initiated by Client. It contains a change of decision for a tour. Normally this process is initiated because new information regarding a tour has been revealed later than at first acceptance. The message of revaluation should also be used for dealing with fines.

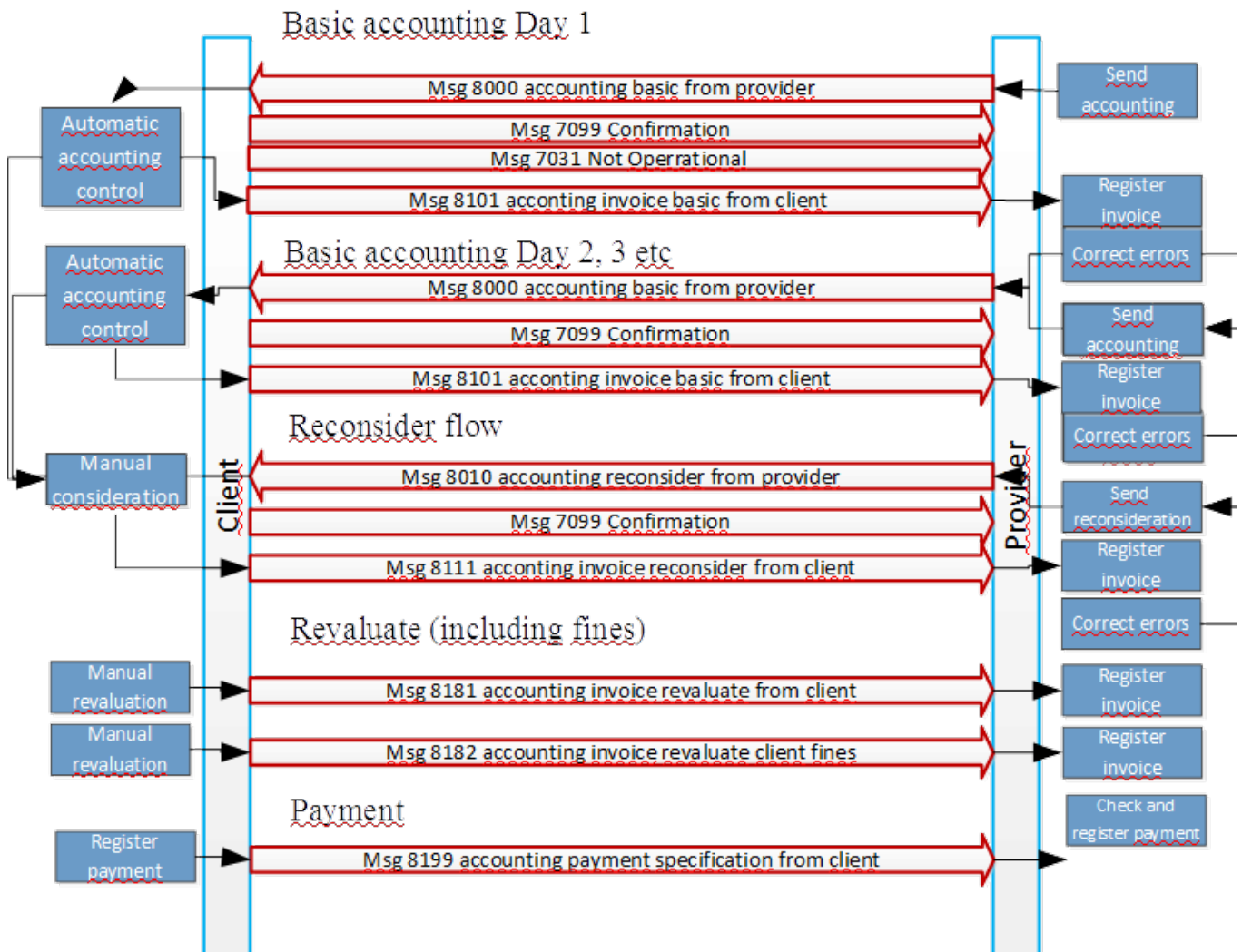
In case that the process of Fines is separate process by the client, the descisions regarding fines can also be sent in an own message, where the identification may differ.

4. Payment

The Client informs the Provider about the content of a specific payment. The message will only contain information witch messages (invoices) that is settled in a payment.

Decisions are always made exclusively by Client. A tour can be resent in many different message as long as it is not accepted. Sending a tour after is has been accepted should always be rejected unless it is the Revaluation process.

2.5.1.2 Flow chart



Observe that msg 7099 above has been specified only in a few of the different interactions. However it can be used in all interactions and it should be used as a technical acknowledgement if the processes are more batch oriented and the message are sent in a more online fashion.

2.5.1.3 Information content

The processes can easily be considered as sending files. If it is handled in such a way each file should only contain one message. The orgSender and orgReceiver should have been agreed in advance between the parties as for other SUTI communications.

Following general instructions regarding the content of the SUTI-elements should be followed in order for the communication to be considered as good use of SUTI.

<msg>

The attribute msgType decides in accordance with Flow chart above what type of message ("file") that is sent. The specification of use is described in document Messages. Please note that msg 8000 shall only contain tours not yet accepted. To send an already accepted tour in msg 8000 shall result in error from client. On the other hand may a non-accepted (rejected) tour be sent many times (until it's accepted or the Provider decides to treat the tour different).

A tour that is not accepted may also be sent in msg 8010. This is the msg that shall be used when the Provider considers that a reject is false or if the Provider has noted the tour for manual treatment. The process should be handled with restrictions as it normally results in a manual treatment and will therefore be more expensive to

handle.

Following considerations should also be noted in msg-element

- idMsg, it holds the "file-identity." and should be kept and used in response
- infoTimeStamp shall exist and shall hold the date of the "youngest" tour in the msg. E.g. if file is sent daily it keeps the date of all the tours. Tours younger than the date shall be rejected with an error.
- referencesTo, shall hold the same value in an idMsg-element as the idMsg msg that it is a response to (as for other SUTI-msg's. Please note that as well as a 7099-msg as corresponding 8101-msg may hold this value, if the 7099-msg is used.
- account-element is mandatory

<account>

The msg will hold all remaining information in an account-element consisting of 5 elements

- an id idAccounting, where the value shall be decided by client. Only tours referring to this value of accounttype shall exist in msg.
- none, one or many tours inside the element tour
- the elements orgClient and orgProvider that shall hold necessary information regarding the Client and the Provider. The information shall hold the same type of quality as is requested in an invoicing process.
- a summary of the economic information for all tours presented inside the element economyReportAccounting. The element shall only consist of one element formOfPayment, which can hold several elements of type payment according to following rules
- msgs 8000 and 8010 from Provider hold only one payment, that presents the requested total amount. This total shall be the sum of the payment for each tour. If the client finds that this is not true it shall return a msg 7030 Syntax Error instead of a msg 8101.
- the response msg from client 8101, 8111, 8181 and 8182 should contain two payment. One payment for the accepted tours and one for the rejected tours
- the msg 8199 contain one payment for each invoice paid

Each payment from provider including an accepted amount shall have an element idEkinfo having a src="XX:InvoiceId" value="invoiceno of client" put in relevant information for each accounting. This id (or those id's) will then also appear in msg 8199. XX is a SUTI-code for the client.

<tour> (tur)

This element is described in detail for each tour by provider while the client may only respond with referenceToTour, statusTour and economyReportTour for a tour. Each tour in a msg from provider shall be present in the response from the client. If not the provider shall send a msg 7030 as a response. (the msg is not presented in the Flow chart but is of course always a possible response option in all communications in SUTI)

The following is true regarding the responsibility to proceed in the invoicing process for a tour in Basic accounting.

- if the tour is rejected or not ready for accounting – provider has the
- if the tour is accepted – only client can revalidate the tour
- if the tour is in the Reconsideration process, i.e. set to validate by client or sent for reconsideration by provider – the responsibility is up to client to either accept or reject.

<referenceToTour>

This element is mandatory for each tour. If the tour has been submitted as an order by client it should contain the value of the idOrder that the tour has been sent as from client. The tour must have the value of the idOrder from the providers system. If the tour has been constructed in the providers system the values of the client's idOrder must be present in suborderTour. The entire accounting is based on the value of the idOrder from the providers system. In case that the tour is constructed in the providers system and an idOrder is not present at start of accounting for a tour the client must add such an idOrder in the response. The combination of those two idOrder values is a unique combination in the accounting process.

It is the responsibility of the client and the provider to store the values of the identity and secure that they are not used in other tours in an accounting communication.

In case that the idOrder of the client is not known by Provider due to technical or other problems, the Provider shall include the identity but set the value to zero (0).

<statusTour>

This element has three elements

1. idActivity, that is an element where src="SUTI:idActivity" and the only values that must be used is listed in SUTI attribute list for idActivity. Following values (or their corresponding number) can be used
 - invoice set by provider for a non-accepted tour
 - accepted set by client when accounting for a tour is accepted
 - rejected set by client when the accounting for a tour is rejected. The element idError must be present
 - validate set by client when a tour is held for revalidation
 - revalidated set by client in the Revalidation process
2. idError of type id where src="SUTI:idError" and any of the existing values in SUTI attribute list for idError can be used
3. manualDescriptionStatus, This is an element that can be used by provider and client to submit text regarding the tour. A possible use is for provider to submit more information in the msg 8010 (reconsideration). The element can be used instead of an email-conversation. If it is used it should contain an infoTimeStamp to inform of the date and time the text was entered.

<economyReportTour>

This is the element that contains the economic information regarding the tour. At least one payment must be presented and it shall have the paymenttype with the value =1303 (account). It shall contain at least one vatAmountSpecification stating the value of the value added tax..

<suborderTour>

The recommendation is that the provider shall at least have one suborderTour for each tour. Inside this element the events of the suborder will be found. All information regarding the traveler (or other content) is also found i.e. requisition number and payment for social fee. It will also state the actual values of node-timing.

The addresses and times should always be the actual known from the providers system. The corresponding planned information it is assumed that the client knows from its own ordering system. All node addresses shall contain gps-positioning Only start and stop event must have other address-information regarding streets etc.

If there is an idOrder from client this information must be present.

To get a complete picture of the tour the information regarding the events must be collected from all suborders inside the tour. This mission is the responsibility of the client in order to get this complete picture.

Following events must be present in 8000 or 8010 msg's.

1. start

Must include an eventContent. All present information regarding the content shall be present in this node e.g. a requisition number presented in an idContent. Payment of type socialfee shall be included in a payment in economyContent. If the content has other attributes these shall be present in attributeContent. This event will hold actual values of pick-up address and pick-up time in eventAddress and eventTime. In case of several suborders some provider systems may only have one start event present. In such a case the content and start information for other suborders must be found in a passengerinvehicle event
2. stop

Contains actual delivery information for content. The same procedure as for start is possible. Stop may be replaced by passengerdropped at intermediate events.
3. acceptOrder

This event contains actual information of the time and position when the driver accepted the tour-mission. In case there are several suborders this event shall either be the same for all suborders or only be present in first suborder

These three events are best placed under eventTour, since they concern the tour and are not specific to any content.

The provider system shall present all node events that exists without any restrictions. Following are examples of such events.

1. navigation
In case provider been ordered to have a navigation node.
2. vehicleatnode
To state when vehicle was on node
3. passengerinvehicle
Notification of information when a traveler has been registered as present in vehicle.
4. passengerdropped
Notification of information when traveler is registered as left the vehicle
5. noshow
To inform that traveler could not be found on pick-up address.

<resourceTour>

Information regarding vehicle and driver information is presented in this element

Regarding the vehicle the following elements shall be present

- idVehicle, this information shall be presented at least once with information regarding the internal number from the provider. However in some accounting other appearances may be necessary e.g. the national number (with src="NO:idVehicle" (or corresponding with value= national licencenumber) or like in opgjör in Norway where the value of löyvenr must also be present (src "NO:LöyveNummer"). In AttributesVehicle, should be presented the actual attributes that the provider has assigned the vehicle
- the elements taximeterType and taximeterSoftware should contain actual information regarding the taximeter

Regarding the driver following information shall be presented

- idDriver
- any attributeDriver
- the name of the driver in element name

<calculationTour>

This element consists of one or more elements of type calculationFare, each gives information regarding collected items from the taximeter or from the provider system.

Each calculationFare has a calculationType. Naming and source for this information is set by the provider but must be submitted to the client before first accounting. The client has the responsibility to map this information to useable items inside it system. The client must have accepted a new value before it may appear in an accounting. For each calculationFare there are either a fixed amount (fixedPriceAmount) or a taximeterAmount. For taximeterAmount only values existing on the Nordic market can be presented. The provider shall without restrictions deliver such detailed taximeter information from its Please not that any start or stop information for the taximeter values is not stored in today's system and cannot be presented

<summaryReportTour>

Used to give information regarding totals for duration and distance for the tour in the elements durationStart och distanceStart.

<eventTour>

There should only be three elements of this type. They hold information on the accept, start and stop of the tour. For further information about events see general discussion eventSubOrder above.

2.5.2 Direct accounting from client to provider

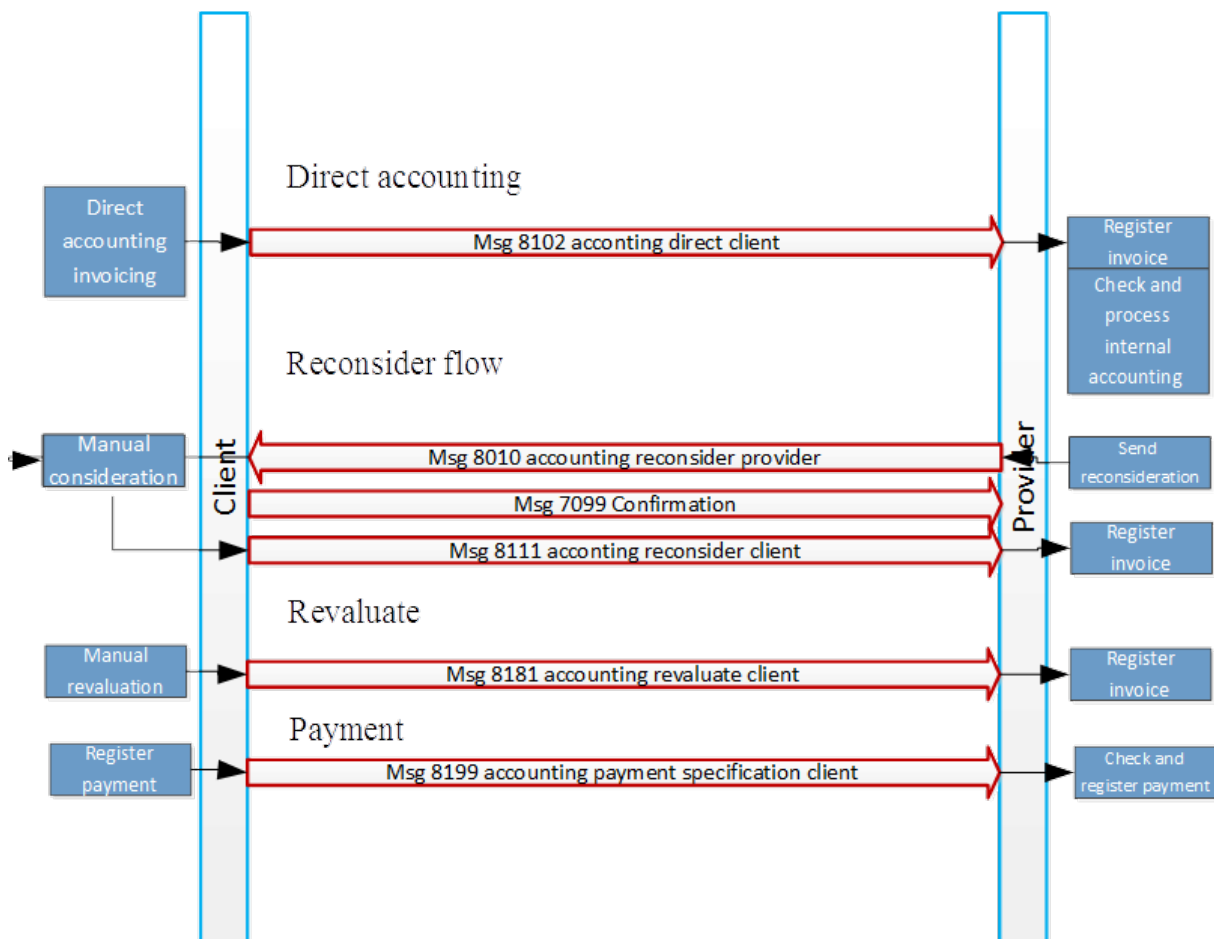
2.5.2.1 General

This accounting will be presented in more detail at launch of first such system. However the process has been designed and shall contain the structure you will find in Flow chart below.

It is based on the situation where the client has already sent information regarding the tour and also economic information in its ordering system. The process is always initiated by client. Normally such a msg 8102 will be a periodically msg containing all tours for a certain time period.

The information content may also differ from what has been described in Basic accounting, since it is based on information inside the client system.

2.5.2.2 Flow chart



2.6 Exchanging resource information

2.6.1 General

The most common task of communication using the SUTI protocol is the transferring of different types of order information regarding one or more transports and the settlement (accounting) of such matters.

However using the SUTI protocol you can also exchange different type of general information between a Client and a Provider regardless of any order. This may mostly be the case as preparations for an order process. Such types of messages uses messagetypes 1xxx described and defined in chapter 1 of Messages.

In this chapter you will find some use cases regarding such message flows described a little deeper.

2.6.2 Resource request and response

The most simple form of such requests are the messages 100x which will result in one or many 101X messages.

By such a request the client will ask for the specification (description) of one or many vehicles and its corresponding driver. The respond is always the current specification of the vehicle and the actual driver.

The request messages all have the same general description with these options

- Msg 1000 request the description of a single vehicle having the idVehicle in referencesTo
- Msg 1001 request the descriptions of all vehicles in an agreement (link) having no idVehicle in referencesTo
- Msg 1002 requesting the descriptions of all active vehicles for a Provider

These types of messages must be agreed in advance between client and provider.

The responses are for 1000 one 1010 and for the other as many 1011 or 1012 as there are available vehicles to describe. Each message 101x only holds one vehicle and the id of the vehicle is present in referencesTo of the message.

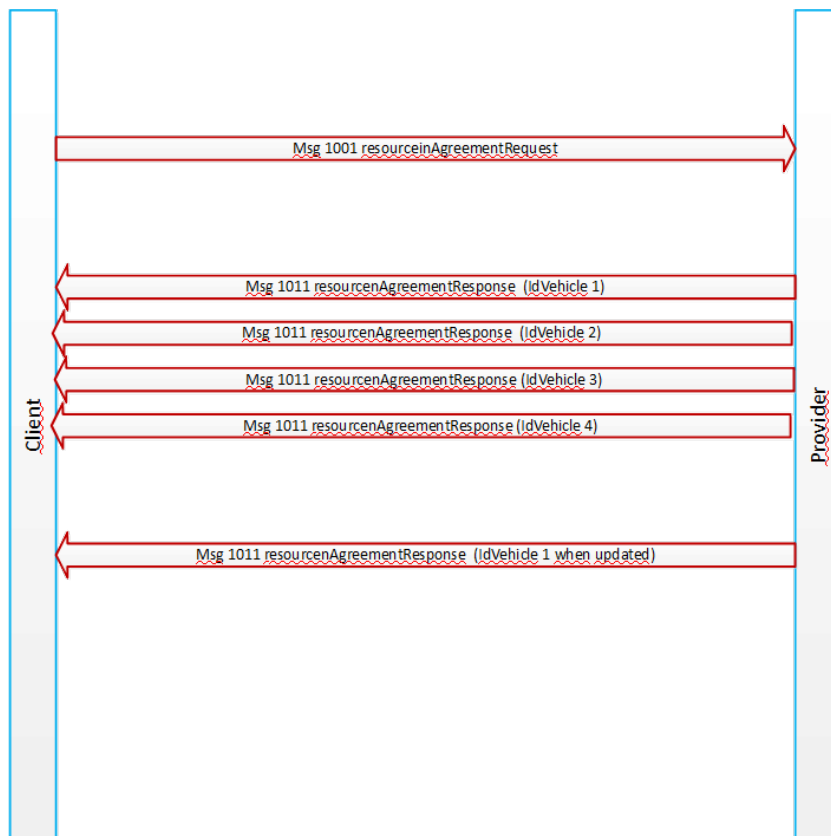
The provider may also request to be informed of any changes i.e. change of configuration of vehicle or change of driver for vehicle. This is done by setting the attribute updateRequest to true. This will mean that the Provider will send a msg 1010 (or maybe 1012 or 1013) whenever the change is documented in the Provider system.

There is no way of ending such an information flow. If the vehicle no longer has a driver, i.e. is not in action the 1010 may contain no driver.

In order to be certain about the current status of all vehicles with updateRequest = true, the 1001 or 1002 is suggested to be sent with some regular interval like once a day (preferably when communication volumes are low).

Even if the Client has set updateRequest to true in a 1001 or 1002 the Client may ask for a single vehicle by using 1000. However a msg 1000 should never hold updateRequest = true. If set the Provider may ignore such setting.

A typical flow using 1001 may be like this. Observe any implementation using 1010 instead of 1011 as a response may be acceptable.



2.6.3 bulkLocation request and response

The Client and the provider may want to exchange information regarding locationData (most often gps-positions) of vehicles. For vehicles in order there are msg 5020 and 5021 to request and respond to such information.

This communication may however tend to be large in amount of data and messages. There may also be a need to have information of other vehicles than those in a specific order.

The answer to that is the use of msg 1100, 1101 or 1102 with the corresponding responses 1110, 1111 and 1112.

- Msg 1100 to request for all vehicles in an geographic area represented by a gps position and a distance
- Msg 1101 to request all vehicles for a current agreement (can be used instead of 5020, 5021).
- Msg 1102 to request all vehicles known to the Provider system

For 1101 and 1102 the Client may also ask to be informed regularly for a certain timeperiod thus providing the Client with continuously information of locationData for vehicles. The 1100 is just a single request.

There may be complementary restrictions that the Client may request. Thus combinations with statusTaximeter will restrict to only vehicles having that setting of statusTaximeter will occur in response. Combinations with resource in request will reduce the response to only vehicles satisfying the resource description will be included in responses.

The msg 1101 can not however have any complementary restrictions. If there are such present in a message the Provider may ignore them.

The flow of 110x and 111x is very similar to the flow of 100x and 101x described in previous chapter. There are however one difference.

The response messages may be large in size and the Provider may want to split the response for a bulk on several messages. Pls refer to chapter 3.2.x for details of how this will happen. There you will also find the details of how a bulkLocation is defined.

3. Use of information elements

3.1 General descriptions

3.1.1 General use of information elements

Understanding and using of information has two aspects. One is obviously sending and receiving of a message. When, why and how to send messages to communicate information is described in chapter 2.

To understand and be able to fully use and treat a message the content must also be understood. This chapter will therefore concentrate on the content.

When building the SUTI standard it has been the ambition to reuse information elements in different situations and successively build an own standard of each information element.

This chapter is therefore dominated by a description of vital information elements in chapter 3.2. The list is prospected to be successively enlarged.

There are however two different type of information elements that has had its own documentation in the SUTI standard. It is the id-elements and resource descriptions. These are of special interest. You will therefore find them deeply described in 3.1.2. and 3.1.3 below.

SUTI encourage use of international standards as far as it is possible. The use of the XML-structure has therefore been essential. As far as it is possible also use of national and international sources is encouraged.

3.1.2 ID (Identification elements) Structures and Usage in SUTI

3.1.2.1 Background

In SUTI there are a large number of different type instances that use Identification elements. There is a general ID-element (**idType**) that assure that all instances has the same structure. The parts are

Attribute source

This attribute is a description of the organisation and/or part of the organisation that keeps the original content of all possible values for the id. How to set **source** is described below.

Attribute id

This attribute hold one possible value from the source set above.

Attribute unique

This attribute has two possible values:

“Yes” tells that the **id** always uniquely identifies the same object.

“No” tells that several objects can have the same value. This may occur when the **id** is reused over time or that the **id** is applicable to several instances. One example of this may be that the id of a transport containing several orders is sent with the order. The **id** is not unique to the order.

There are many **idType**'s among the SUTI elements. We will not describe these here. We refer to the schema and the message-description for that. There are however one specific element **SUTId** whose usage is described in detail below. If the usage of this element is not in correspondence with what is stated there the link can never be approved as a SUTI-link.

3.1.2.3 Rules for usage of attribute source

General sources

The value of the attribute **source** for the same source should obviously be the same wherever its used. In principle general type of sources like ISO, IATA etc should be preferred. In SUTI we have at present identified following general sources.

Links to referenced ISO standards

Topic	Standard	Link
Currency	ISO 4217	http://www.bsi-global.com/British_Standards/currency/index.xalter
Language	ISO 639-2	http://www.loc.gov/standards/iso639-2/langcodes.html
Country	ISO 3166-1	http://www.iso.org
Time and Timezones	ISO 8601	http://www.iso.org

For the attributes for description of different types of time we strongly emphasize use of “**ISO 8601**” (<http://www.iso.org>), **even if these attributes are not of idType**

Member sources

The most vital idTypes are **idOrg**, used for **orgSender**, **orgReceiver** and **orgProvider**. Here you will find the description of a member or a partner to a member. These id's should have a source “SUTI:idLink” if the values are set as the values of the link (“holmedal_hbgtaxi_001”) or be set with the structure “SUTI:idMember” (e.g. **source** “SUTI:holmedal” if the **id** is like “holmedal_hbgtaxi”). For description of **idLink** see full description below.

Each member in SUTI has its unique **idMember**, which is set when the member has been approved and paid. Each member keeps its own directory of its partners and systems.

A lot of sources derives from sources within the members organisation or partners to the member. Some of these are very specific to the member e.g. **idAgreement** and **idProduct**. There are no national or international sources to describe these id's. Therefore the source must be some source by the member. A very natural way of expressing the source is “idMember:idAgreement” and “idMember:idProduct”, e.g. “flygtaxi:idAgreement” and “flygtaxi:idProduct”. Unfortunately today the structure is not exactly like this. As long as the source is given like “idMember:somethingUniqueWithinTheMember” it will do until further, like “flygtaxi:Product”.

There are other examples of these type of sources which could be given by more than one **source** e.g. **idVehicle** and **idDriver**. These type of id's could be described in two different ways, either from an internal source or from a general source. For idVehicle the **id** can be either the national registration number “ABC 123” or the internal number used by the provider “456”.

When using the internal id the **source** should be with the stucture “idMember:idPartner:idVehicle”, e.g. “holmedal_hbgtaxi:idVehicle”. Very many today use a structure like “holmedal_hbgtaxi_001:Vehiclno” for this purpose. There is a use of the “ValueOfAn_idLink:Vehiclno”. This is not incorrect but is in excess. The risk that there would be another source with the identical value is minimal.

When using national id's these should refer to a national registry. In case these are a part of an international standardization unit the source should obviously be set as for general sources described above. In case these registries is just hold by a national organisation which could change over time we recommend use of more describing sources like “SE:idVehicle”, i.e. “National code ISO 3166-1:idVehicle”. SUTI will never allow any members to have idMember conflicting with the ISO 3166-1. Nor must not any use of such a shortening other than representing a national authority be set in conflict with ISO 3166-1. Thus this forms a good base for having these type of sources unique and the same for a long timeperiod. In Norway the same source is therefore is set as “NO:idVehicle”.

When setting the source to national sources for organisations or persons similarly use of “SE:idOrg” and “SE:idPerson” can be used.

3.1.2.5 SUTI Id Description

This section explains how to construct a complete SUTI Id and how to register a new SUTI link.

A SUTI Id is an important part of the SUTI link and the communication performed through the link. The SUTI Id is used to uniquely identify the Client and the Provider in a SUTI communication and give the Client system and the Provider system a tool to make sure that the other side is the correct side and it is the correct link that is used. This is very important if a system is using several links to one Client/Provider or to different Clients/Providers.

A SUTI-Id is built from 3 different parts:

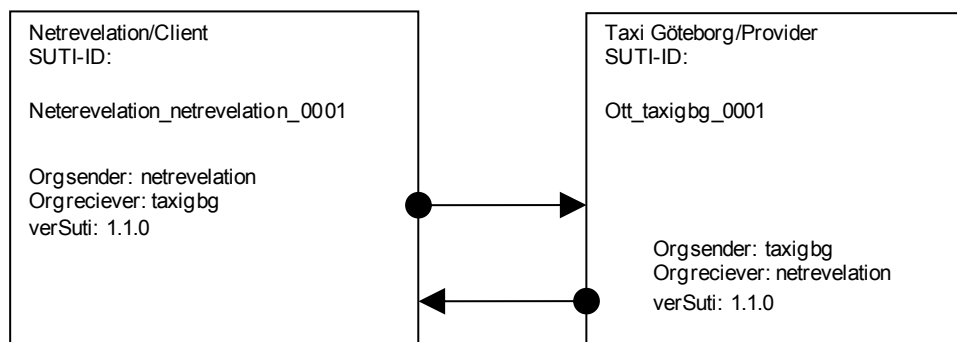
1. The Id of the System supplier. This can be found in the document Memberids located at the SUTI member pages (<http://member.suti.se>). If the System Supplier is a new member of SUTI and haven't had their Id distributed or issued, contact the Technical committee. If the System Supplier isn't a member of SUTI an Id can not be issued until a the Supplier has become a member.
2. The Id of System owner (a partner to the member or a member itself). If the System Supplier is a new member of SUTI and haven't had their Id distributed or issued, contact the Technical committee. If the System Supplier isn't a member of SUTI an Id can not be issued until a the Supplier has become a member.
3. An Id that uniquely identifies this link at the Client's or the Provider's site.

These three are put together like this:

Id systemsupplier_Id System owner_unique Id

Example:

A SUTI link between a Netrevelation system at Netrevelation and an OTT system at Taxi Göteborg could look like this:



3.1.3 Resource description

3.1.3.1 GENERAL

One of the most basic functions that must be standardized to get a SUTI-link to work, are how we describe the resources used for the transportation. If the Client doesn't work with the same description of a vehicle that the Provider use, it's impossible to send an order from the Client to the Provider with any possibility to provide a resource that is needed for the order. This document contains a way to describe resources needed to provide a transportation service.

Two major resources are used in order to provide a transportation service, a Vehicle and a Driver. This is the resources described in this document.

3.1.3.1.1 VEHICLE GENERAL

A vehicle has two sets of descriptions, Capacity and Attributes. Capacity describes the size of the vehicle, the number of passengers in the vehicle and attributes specifying a seat. Attributes specify functions and properties of the vehicle.

3.1.3.1.1.1 VEHICLE CAPACITY GENERAL

Many vehicles have the possibility to let the driver change the internal configuration very easily. This poses a problem if we shall be able to handle an order that demands a configuration that a vehicle can't carry out in the normal configuration, but with very little effort, the driver can change the configuration of the vehicle to fit the demands in the order. We have a need for a description of the alternative configuration. This need for an alternative description gets even more important if we deal with orders involving travelers in wheelchairs or in stretchers.

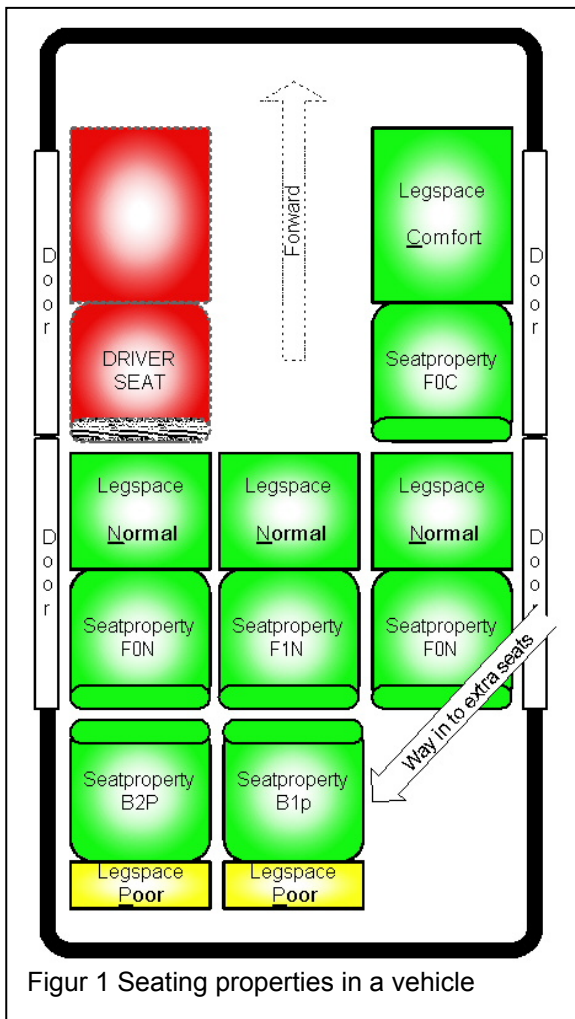
Many of these vehicles have many combinations of seating and luggage space or seating and wheelchair space. A description of a vehicle that handles all the combinations of seat and space will be very extensive and too difficult to handle for both Client and Provider systems. To solve this problem we will deal with the possibility to have three descriptions of a vehicle: Normal, Alternative and Stretcher description. Normal will be the configuration a vehicle normally has. Alternative will be the configuration if all possibilities are used. Stretcher configuration will be the configuration that can handle a stretcher.

3.1.3.1.2 DRIVER GENERAL

A driver has only one set of descriptions: Attributes. Attributes of the driver specify skills and other properties of the driver.

3.1.3.2 SEATING PROPERTIES in VEHICLE

Three properties are used: Direction, Access and Legspace. A property for a seat can be F0C. The first character is the Direction "F", the second character is Access "0" and the third character is Legspace "C".



3.1.3.2.1

DIRECTION

Indicates if the traveler will be facing Forward ("F") or Backward ("B") while seated in this seat.

3.1.3.2.2

ACCESS

Indicates how many sets the traveler has to pass over in order to reach his/her seat. This is indicated by a number from zero and up.

If it is a seat placed at a door in figure 1 the access property will be "0". If it's the middle seat in the back in figure 1 the access property will be "1". The left extra seat in figure 1 has the access property "2" as the traveler have to pass over both the right back seat and the right extra seat.

3.1.3.2.3

LEGSPEACE

Indicates the Legspace for the traveler while seated.

Comfort ("C") = Good space, may be the seat beside the driver.

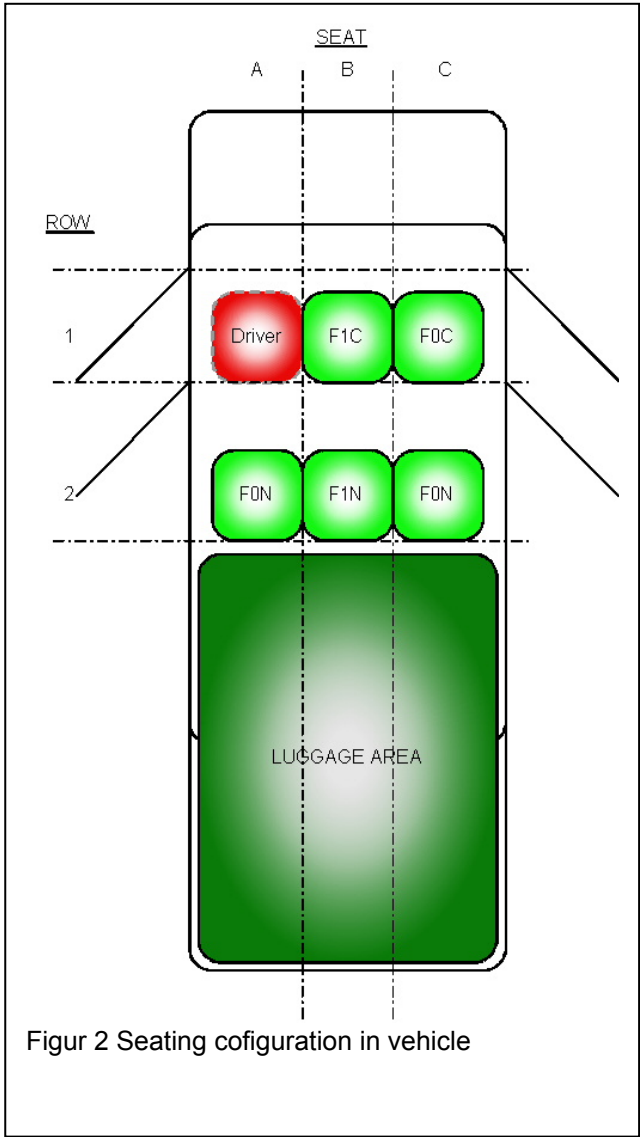
Normal ("N") = Normal space, may be the seat behind the driver in a normal car.

Poor ("P") = Very little legspace, may be an extra seat placed behind the normal back seat.

3.1.3.3 SEATING POSITION in VEHICLE

In certain orders the Client has a need to specify a certain seat in a vehicle. If the traveler has the need to sit in one of the front seat, the Client system must specify this to the Provider system in the order.

In order to facilitate this we have adopted a system derived from airline booking. The position of the seat is specified by a row and a place in that row. In order to specify a certain seat the Client system tells the Provider system in the order that a certain traveler shall be seated in a certain seat. This is done under the node section of the order message in which the pickup point of this traveler is specified. Here the client specify seat 1B, which mean that the traveler shall be seated next to the driver in the front seat.



3.1.3.3.1 ROW

Row number is counted from the front toward the back of the vehicle. The first row is numbered 1, the second 2 and so on through the vehicle. This makes it possible to describe vehicles reaching from taxis through buses up to airplanes.

In row number 1 the seat reserved for the driver are not specified, to avoid booking this seat for passenger.

3.1.3.3.2 PLACE IN A ROW

Place are counted from left to right facing in a forward direction of the vehicle. Numbers are A,B,C and so on.

Place 1C in figure 2 will have the property F0C.

3.1.3.4 ATTRIBUTES on VEHICLE

Attributes on a vehicle specify functions and properties of the vehicle. A vehicle can have certain equipment required for special assignments or may be sanitized for travelers with allergies.

These attributes are managed by SUTI and kept in a public list for members to use.

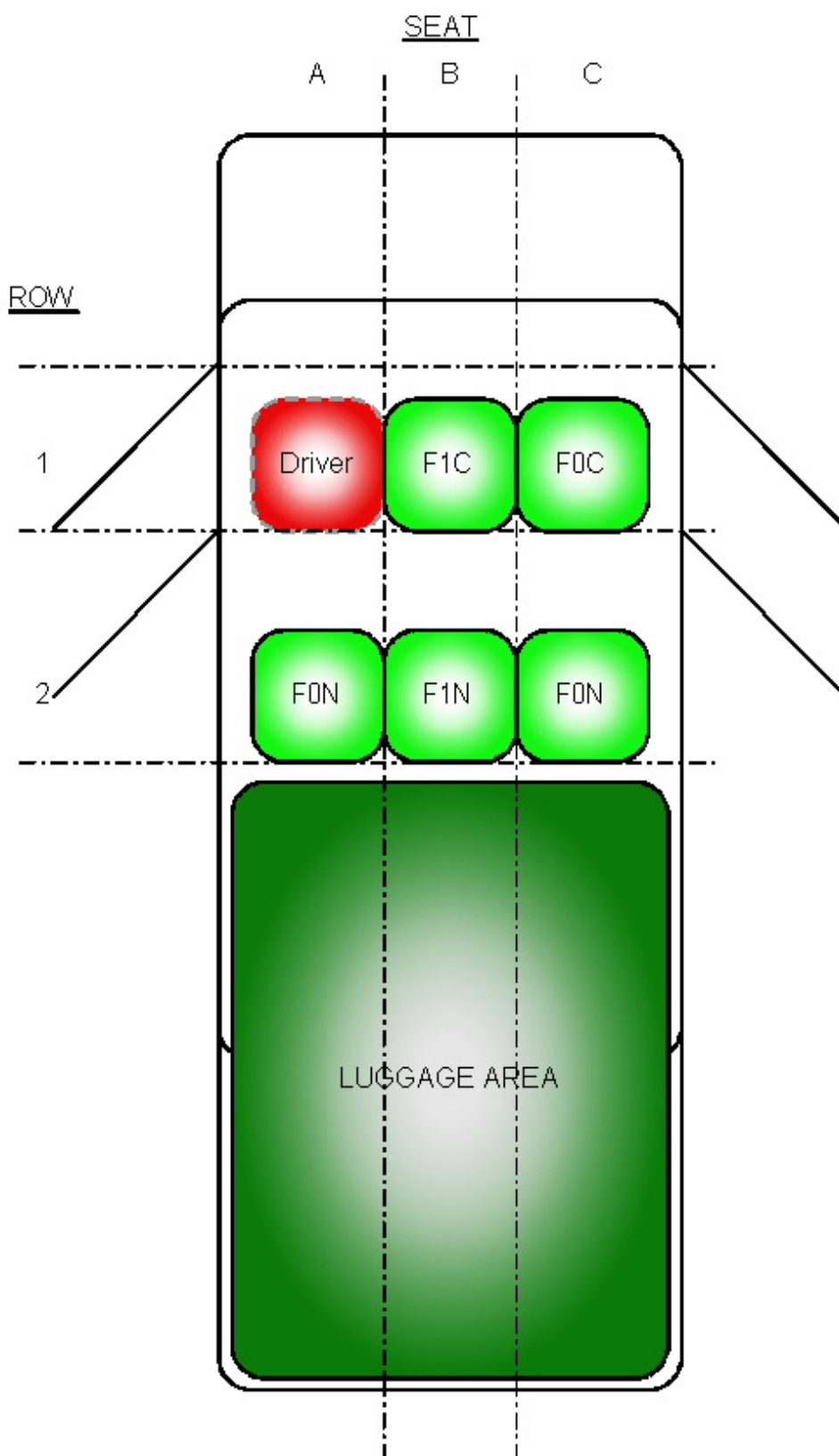
3.1.3.5 ATTRIBUTES on DRIVER

Attributes on a driver can be special skills or education that the driver possesses. He may talk a certain language, he may have knowledge about first aid or CPR. Another attribute on a driver is if he/she has special security clearances for certain places such as airports, harbors or other restricted areas.

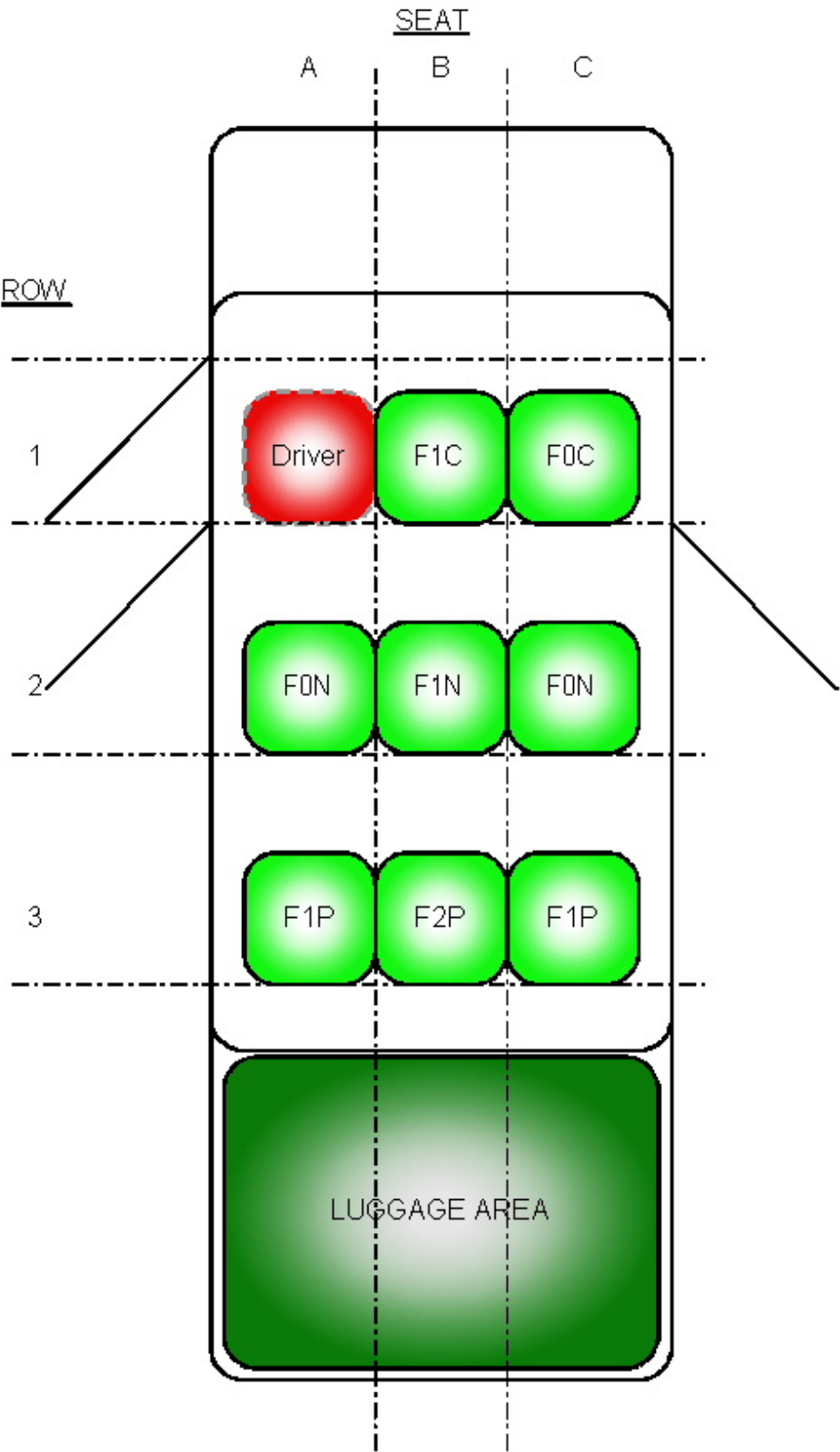
These attributes are managed by SUTI and kept in a public list for members to use.

3.1.3.6 EXAMPLE of CONFIGURATION

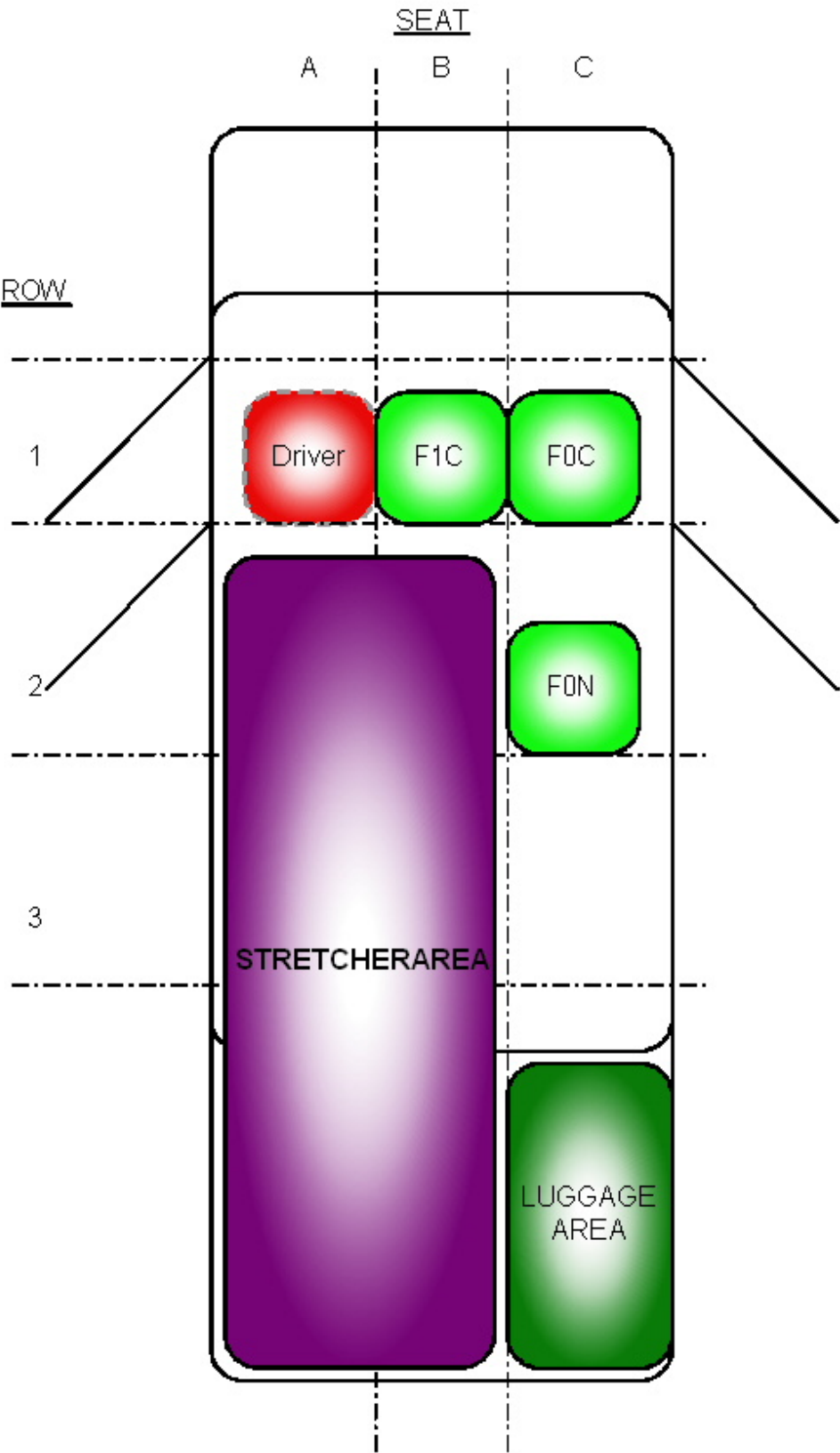
3.1.3.6.1 NORMAL CONFIGURATION



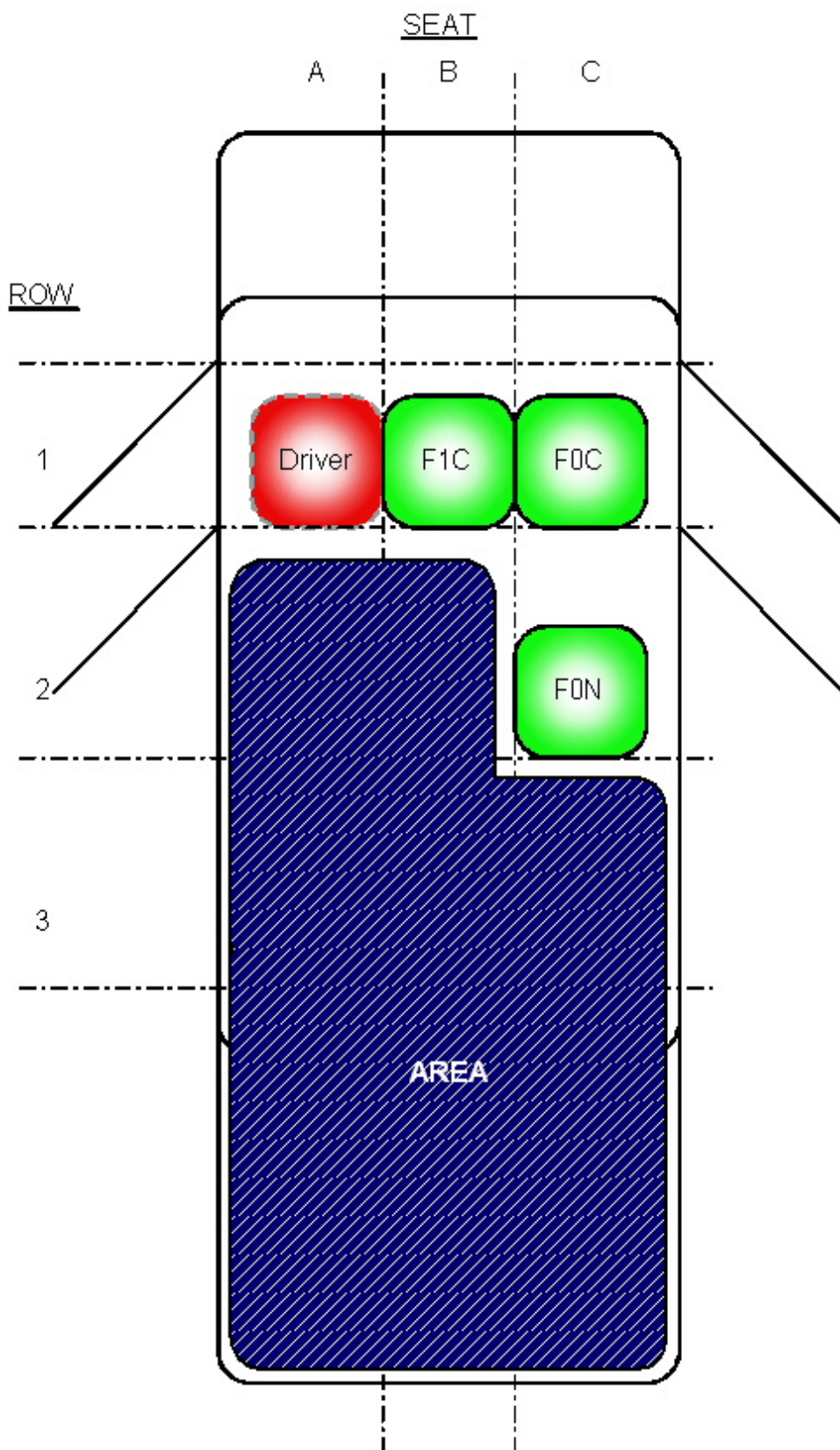
3.1.3.6.2 ALTERNATIVE CONFIGURATION



3.1.3.6.3 SPECIAL CONFIGURATION STRETCHER



3.1.3.6.4 SPECIAL CONFIGURATION WHEELCHAIR



3.1.4 Content

3.1.5 Addresses (addressNode)

Addresses are mainly used in addressNode to indicate Pickup, Drop off and/or Via, a point along the travelled route where to perform an action. The action in Via can be to drive through a certain point or giving the traveller a possibility to do something at a certain point between Pickup and Drop off.

To describe an address SUTI uses xml type "addressType" which include different ways to describe an address. It's built of a set of attributes and a set of elements. The addressType contains both information in the form of a normal address eg "Short lane 7B; 12345; Big City" and geographic information in the form of a zone or a position. It's highly recommended that geographic location with a position (latitude/longitude) is used every time an address is passed between Client and Provider.

The information in an address node is most vital for the operation between The Client and The Provider. The Client has the responsibility to provide the information with such a quality of the information that the Provider can fulfill the order without unnecessary treatment. Generally the Provider should automatically be able to do the following operations:

- Find suitable vehicle
- Allow driver/vehicle with information necessary to carry through without any further information including navigating to an address.

The Provider must have enough information in order to fulfill other obligations such as price setting and route planning etc.

It's the Providers responsibility to check that the information is acceptable. In case the information is not acceptable the provider should always be able to reject the order returning reject reason 2515 addressinfo not sufficient

3.1.5.1 Minimum requirements

At least one of either street in combination with streetNo and streetNoLetter or addressName shall be used. If addressName is used without street it must be a generally known name eg "Södersjukhuset" or "Arlanda Airport".

A route description shall always be placed in manualDescriptionAddress, never in any other field. The fields street, streetNo and streetNoLetter shall be validated against an address database at the Client or an external database like Google or similar. The responsible part is the part that originally enters an address into the Client system or a pre system to the Client system

3.1.5.2 Attributes in addressType

Name	Description
addressName	Used to give the possibility to give the name of a hospital, a restaurant etc eg "City Hospital". This field shall never contain the complete address.
street	The official name of the street eg "Big street"
streetNo	Number of the street eg "67"
streetNoLetter	Sub number of street number. The address "Big street 67" has four entrances, 67A, 67B, 67C and 67D. In this attribute you will find "A" for the address "Big street 67A".
location	Normally the postal address name. or even smaller area names eg "Stadshagen".
community	Normally the wider area for several locations. For example swedish "kommun".
postalNo	Zip code eg "197 35"
country	Country eg "Sverige"
mapPage	Page containing the address in a map eg "123A4" <i>NOTE!! If this is used please inform The technical Committee about this.</i>

3.1.5.3 Elements in addressType

Name	Description
geographicLocation	Geographic information containing latitude and longitude and other information needed to describe the position of this address.
idAddressname	This brings the possibility to specify an identity for the organization at the address. The source and the use should be described in a self declaration.
idStreet	This brings the possibility to refer to national or commercial map data.
idPostalCode	This brings the possibility to refer to a nationally defined postal code.
idCommunity	This brings the possibility to refer to a national id for municipality.
idCountry	Id identifying Country. Source (src) shall refer to ISO 3166-1
idZone	Id of a used Zone or zoneset. Source (src) shall refer to either a general zone defined between sender and receiver or to a zoneset used by one or both of the systems.
idMap	Used map. Source (src) shall always refer to used map.
attributeAddress	Attributes of addressNode. Mainly an attribute indicating a special need for a specific traveller at this specific address. But may be used in other type of contents such as parcel. Attributes can be found in SUTI's attributelist or in specific lists used by Client or Provider. Source shall always refer to the owner of the attribute list.
manualDescriptionAddress	Used to provide a further clarification of an address eg "Våning 1"

3.1.6 Geographical location (geographicLocation)

Geographical location is used to locate and describe in a relative exact way a specific point in in a city or in the countryside. The points can be addresses (Kungsgatan 1), institutions (Södersjukhuset), country addresses (postlåda 1432) or generally known places (Arlanda Airport).

SUTI uses the format WGS-84 (Iso standard: ISO 6709:2008) with decimal degrees.

`<geographicLocation lat="58.39193" long="13.42448" precision="100" typeOfCoordinate="WGS-84" />`

3.1.6.1 Resolution WGS-84 decimal degrees

Decimal degrees (55.123459)	Resolution
1	100 km
0.01	1 km
0.001 *	100 m
0.0001 *	10 m
0.00001 *	1 m

*) A resolution of better than 100 meters is preferred, at least three decimals (0.001 decimal degrees).

3.1.6.2 Attributes and elements in geographicLocation

Name	Description
zone	A description of a geographic area that has to be agreed upon between Client and Provider.
typeOfCoordinate	Shall always be WGS-84
lat	Latitude in WGS-84 and decimal degrees eg 54.737834
long	Longitude in WGS-84 and decimal degrees eg 9.622380
precision	Precision shall be from GPS-unit in vehicle and measured in meter. This is the deviation in position measured by the GPS-unit.
speed	Speed shall be from GPS-unit in vehicle and measured in meter per second.
speedSource	Speed measured at GPS or VEHICLE.
direction	Direction shall be from GPS-unit in vehicle and measured in degrees (0 - 360 degrees).
height	Height shall be from GPS-unit in vehicle and measured in meter. May be measured in meter with decimals.

deviationSpeed	Deviation in speed
vehicleDistance (element)	Used then Client ask provider for suitable vehicles for a possible mission. Note that it is not an order (msg 2000) but only a resource request.

3.1.6.3 Minimum requirements geographicLocation

A position must be specified for each address or navigation point so that it can be displayed in map or navigation software for the driver.

3.1.7 Economy

3.2 List of Information elements

3.2.1 Agreement

3.2.1.1 General

```
<agreement name="FLINK">
  <idAgreement src="Flygtaxi:AGREEMENT" id="FLYGTAXI" unique="true" />
  <product>
    <idProduct src="Flygtaxi:PRODUCT" id="TAXI" unique="true" />
  </product>
</agreement>
```

If idAgreement must be unique for each connected SUTI-link the source must refer to the complete link Id of this link ([flygtaxi_flinksthlm_0001](#)) and the range that contains agreementId at the client ([AGREEMENT](#)). But if idAgreement is unique within the Flygtaxi organization it's possible to use the organizations SUTI id ([Flygtaxi](#)) and the range that contains agreementId at the client ([AGREEMENT](#)).

The same apply to idProducts. If idProducts must be unique for each connected SUTI-link the source must refer to the complete link Id of this link ([flygtaxi_flinksthlm_0001](#)) and the range that contains productId at the client ([PRODUCT](#)). But if idProducts is unique within the Flygtaxi organization it's possible to use the organizations SUTI id ([Flygtaxi](#)) and the range that contains productId at the client ([PRODUCT](#)).

3.2.1.2 Minimum requirements

3.2.2 idMsg

3.2.2.1 General

```
<idMsg src="flygtaxi_flinksthlm_0001:MSGID" id="2010081802070189" unique="true" />
```

If idMsg must be unique for each connected SUTI-link the source must refer to the complete link Id of this link ([flygtaxi_flinksthlm_0001](#)) and the range that contains messageId at the part that creates and sends the message ([MSGID](#)). But if idMsg is unique within the Flygtaxi organization it's possible to use the organizations SUTI id ([Flygtaxi](#)) and the range that contains messageId at the part that creates and sends the message ([MSGID](#)).

3.2.2.2 Minimum requirements

3.2.3 idOrder

3.2.3.1 General

```
<idOrder src="flygtaxi_flinksthlm_0001:MISSIONID" id="173438" unique="true" />
```

If idOrder must be unique for each connected SUTI-link the source must refer to the complete link Id of this link ([flygtaxi_flinksthlm_0001](#)) and the range that contains orderId at the client ([MISSIONID](#)). But if idOrder is unique within the Flygtaxi organization it's possible to use the organizations SUTI id ([Flygtaxi](#)) and the range that contains orderId at the client ([MISSIONID](#)).

3.2.3.2 Minimum requirements

3.2.4 infoTimeStamp

3.2.4.1 General

```
<infoTimeStamp>
  <time timeAccuracy="1" timeZone="1" time="2010-08-18T08:02:26.29" timeType="actual"/>
</infoTimeStamp>
```

infoTimeStamp shall be used to timestamp the information that is sent in the message. It allows the provider for example to send a time with msg 5021 Requested Location telling at which time the location was valid. The

timetype used shall be actual as it's the actual time the information was valid.

3.2.4.2 Minimum requirements

3.2.5 msgTimeStamp

3.2.5.1 General

```
<msgTimeStamp>
  <time timeAccuracy="1" timeZone="1" time="2010-08-18T08:02:26.29" timeType="actual"/>
</msgTimeStamp>
```

msgTimeStamp shall be used to timestamp the sender's creation of the message. The time type used shall be actual as it's the actual time the message arrived.

3.2.5.2 Minimum requirements

3.2.6 orgReceiver

3.2.6.1 General

```
<orgReceiver name="Stockholm Transfer">
  <idOrg src="SUTI:idLink" id="tsys_flink_0001" unique="true" />
</orgReceiver>
```

The orgReceiver is the part that receives a certain message. This means that both Client and Provider can be orgReceiver depending on the message in question.

3.2.6.2 Minimum requirements

3.2.7 orgSender

3.2.7.1 General

```
<orgSender name="Flink Stockholm">
  <idOrg src="SUTI:idLink" id="flygtaxi_flinksthlm_0001" unique="true" />
</orgSender>
```

The orgSender is the part that sends a certain message. This means that both Client and Provider can be orgSender depending on the message in question.

3.2.7.2 Minimum requirements

3.2.8 referencesTo

3.2.8.1 General

```
<referencesTo>
  <idOrder src="Holmedal_VejbyTaxi_0001:Bookings" id="117691-0" />
  <idOrder src="trapezesoftware_Skånetrafiken_0002:BookingId" id="3872661" />
  <idMsg src="trapezesoftware_Skånetrafiken_0002:MsgId" id="2009082411641113" />
</referencesTo>
```

In a message that is being sent between Client and Provider referencesTo is used to indicate which order and which message that the sent message refers to. It's a very good practice to use both the Clients idOrder and the Providers idOrder and at least the idMsg of the message that is being answered with this message. Another use of referencesTo is to indicate that an order already has been sent to another Provider before it was sent to the Provider that just received it.

3.2.8.2 Minimum requirements

It is important that all SUTI systems comply with the use of <referencesTo> tag. The documentation for SUTI describes that at least these telegrams must use the tag <referencesTo>:

1020, 1021, 1022, 1023, 1024, 1025, 2000, 2001, 2002, 2003, 2010, 2011, 2012, 2013, 2030, 2031, 2032, 2050, 3xxx, 4xxx, 5xxx, 6xxx

As a minimum, both Provider and Client in SUTI always include these elements in <referencesTo> tagged:

<idOrder>	Reference for Provider SUTI idOrder
<idOrder>	Reference Client SUTI idOrder
<idMsg>	Reference to the idMsg being answered

Has msg2030 (Order Forwarded) been used, one can further refer to <idOrderForward> from all systems that have received the order.

ReferencesTo tag may include several other elements. Use several elements when there is a need to specify information about the implementing party. Thus, telegrams 3xxx, 4xxx, 5xxx, 6xxx shall contain a reference to the Provider car which carry out the order via <idVehicle> element.

<idVehicle>	Reference for Provider defined idVehicle
<idDriver>	Reference for Provider defined idDriver (tas eventuellt bort som minimum req.

Note that both the Client and Provider must include all references in its communications.

Client system must take note of idOrder and idVehicle from Provider system, and always include it with its own elements of <referencesTo>!

Using elements <idRejectReason> or <idOrderStatus> there must be documentation of the values of the element can contain, and use only the values SUTI specification document.

<idRejectReason>	Use of msg2002 (Order Reject).
<idOrderStatus>	Use of msg2531 (Order Status).

(see, for example, referencesTo_example.xml)

3.2.9 Event types

3.2.9.1 General

3.2.9.2 Minimum requirements

3.2.9.2.1 Event Type: 1701 invehicle (alt passengerinvehicle)

Used in Msg	4010
Relation to	Node for pickup.
Description	This event is used to signal that the content has entered the vehicle at the pickup point. The signal shall be sent without delay after its triggered. Sent from Provider to Client.
Triggered by	Driver signals that the content has entered the vehicle by pressing a button or reading some type of id card presented by the content.
Used Timetype	Actual
Other demands	

3.2.9.2.2 Event Type: 1702 exitvehicle (alt passengerdropped)

Used in Msg	4010
Relation to	Node for drop off.
Description	This event is used to signal that the content has left the vehicle at the end point. The signal shall be sent without delay after its triggered. Sent from Provider to Client.
Triggered by	Driver signals that the content has left the vehicle by pressing a button or reading some type of id card presented by the content.

Used Timetype	Actual
Other demands	

3.2.9.2.3 Event Type: 1703 noshow

Used in Msg	4010
Relation to	Node for pickup.
Description	This event is used to signal that no content showed up.
Triggered by	Driver signals this by pressing a button or other device.
Used Timetype	Actual
Other demands	The behavior for the driver must be regulated either in a Self Declaration or a specific tender.

3.2.9.2.4 Event Type: 1704 parcelinvehicle

This eventtype is equal to 1701 invehicle and we recommend the use of 1701 invehicle.

3.2.9.2.5 Event Type: 1705 parceldropped

This eventtype is equal to 1702 exitvehicle and we recommend the use of 1702 exitvehicle.

3.2.9.2.6 Event Type: 1706 actiondone

Use atnode och exitnode instead of action done?

Used in Msg	4010
Relation to	Node for action
Description	
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.7 Event Type: 1707 navigationdone

Use atnode och exitnode instead of navigation done?

Used in Msg	
Relation to	
Description	
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.8 Event Type: 1708 cancelatdoor

To be deleted! Use noshow instead or request a cancelation from provider!

Used in Msg	
Relation to	
Description	
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.9 Event Type: 1709 vehicleatnode

Used in Msg	4010
-------------	------

Relation to	All types of nodes.
Description	Signals that the vehicle has arrived at the node or is estimated to arrive at the node.
Triggered by	If the vehicle is at the node the driver signal this by using a button or other device in the vehicle. If it's an estimated arrival at node either the driver or the provider system can send the signal. For the moment there are no standard how the signal shall be sent and who will send it.
Used Timetype	Actual, Estimated
Other demands	

3.2.9.2.10 **Event Type: 1710 infotocontent**

Used in Msg	4010 or 4040
Relation to	Node for pickup
Description	Signals that the Provider or Client has sent information to the content. The actual text sent is present in manualDescriptionContent.
Triggered by	Provider system 4010 – Client system 4040
Used Timetype	Actual time
Other demands	

3.2.9.2.11 **Event Type: 1711 dispatchconfirmationsent**

Used in Msg	4010/4040
Relation to	Node
Description	Preferably used by Client (4040) to inform Provider that it has informed the content which vehicle that has been confirmed for the order. Sent only for pickup node of content
Triggered by	Client
Used Timetype	Actual
Other demands	

3.2.9.2.12 **Event Type: 1712 delayconfirmationsent**

Used in Msg	4010/4040
Relation to	Node
Description	Preferably used by Client (4040) to inform Provider that it has informed the content that vehicle is delayed for vehicle at node. Sent only for pickup node of content. We recommend not to use this message but instead use 1709 vehicle at node with an estimated time.
Triggered by	Client
Used Timetype	Estimated
Other demands	

3.2.9.2.13 **Event Type: 1713 arrivalconfirmationsent**

Used in Msg	4010/4040
Relation to	Node
Description	Preferably used by Client (4040) to inform Provider that it has informed the content that vehicle is at node. Sent only for pickup node of content
Triggered by	Client
Used Timetype	Estimated
Other demands	

3.2.9.2.14 Event Type: 1714 start

Used in Msg	80xx
Relation to	Tour
Description	Tour has started
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.15 Event Type: 1715 stop

Used in Msg	80xx
Relation to	Tour
Description	Tour has stopped
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.16 Event Type: 1716 acceptOrder

Used in Msg	80xx
Relation to	Tour accepted
Description	
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.17 Event Type: 1717 vehicleExitNode

Used in Msg	4010
Relation to	All types of nodes.
Description	A possibility to use at dropoff after registration of 1702 exitVehicle if there is a delay e.g. depending of assistance to content. Possibility is also to use 1706 action done.
Triggered by	
Used Timetype	
Other demands	

3.2.9.2.18 Event Type: 1718 contentNoshow

Used in Msg	4040
Relation to	Node for pickup.
Description	This event is used to signal that passenger or parcel will not appear at node
Triggered by	Client system have got the information from passenger or parcel handling agent and triggers the event
Used Timetype	Actual
Other demands	The Client should describe in a Self Declaration how it expects the Provider to act if this message is sent

3.2.9.2.19 Event Type: 1719 contentAtNode

Used in Msg	4040
Relation to	Node for pickup.
Description	This event is used to signal that passenger or parcel is at node
Triggered by	Client system have got the information from passenger or parcel handling agent and triggers the event
Used Timetype	Actual
Other demands	The Client should describe in a Self Declaration how it expects the Provider to act if this message is sent

3.2.10 idPayment/idEkInfo

An example of use of idPayment and idEkInfo regarding the use of payment references.

<economyOrder>

```

    <price priceResponsible="provider" fixedPrice="true"
    vatIncluded="true" price="390" />
    <formOfPayment>
    <payment paymentType="account">
    <idPayment src="Providerbolaget:CustomerId" id="Kund123456" />
    <idEkInfo src="Providerbolaget:Costcenter" id="XXXXXX" unique="true" />
    </payment>
    </formOfPayment>
</economyOrder>

```

3.2.11 locationList

This is a list of vehicles and its last known position. The locationList is a part of a bulkLocationList where the aim has been to communicate positions of a large number of vehicles at the same time maybe frequently repeated. Thus the standard has minimized the number of character that will be transferred in each message.

In first edition SUTI has set a standard for a locationListType="1" opening up the future possibility of adding other formats.

In locationListType="1" each item in the array of locations will be separated by brackets () and each items inside the brackets will be separated by semicolon giving a structure (value 1;value 2;etc;value n). The following values can be present in this littype:

Value 1 idVehicle, the source is assumed to be the linked, thus having the values from the provider

Value 2 time of observation, should be set as integer holding seconds since 1970 UTC

Value 3 statusTaxameter, holding any of the values N, F, O or B

N = Not in service

F = Free (ready for order)

O = Occupied (in order)

B = In service but set for temporarily Break

Value 4 lat, the latitude coordinate expressed in wgs84 decimal degrees

Value 5 long, the longitude coordinate expressed in wgs84 decimal degrees

Value 6 speed, expressed as an integer in km/h

Value 7 direction, holding integer values 0-359 as an angel, 0=north

An example of an observation could thus be (123;1513728000;O;59.339176;17.991666;15;210) for the vehicle having id=123.

In the locationList the observations will be given in sequence giving the structure

<locationList> (.....) (.....)(.....) </locationList> for the observations of three vehicles.

In the messages holding observations (111X) the amount of data may be large. In order not to exceed sizes for messages the provider has the option of separating the response into separate messages. This can be solved by using attributes in bulkLocationResponse.

Attribute Values/Example

isCompleet Set to True if the response holds all observations, set to False if the observations

msgSeqNo The sequence number of a split msg

msgCount The total number of msg to fulfill the response

3.2.12 Ratings

From 2018 SUTI has introduced the possibility to communicate ratings between Client and Provider (both ways,

since Provider may act to collect data on behalf of Client).

The ratings may be sent for an order or for a vehicle or a driver depending on the information present in ReferencesTo. Since the ratings most often is dependant on a rating from a traveler, the message may appear at any moment. It may also be sent as a rating average for a longer time period. In such case the rating will hold ratingTimeFrom and ratingTimeTo.

SUTI has set a standard for ratings sent. We strongly recommend Clients and Providers to adopt to the SUTI standard in order to bring ratings to something useful for Partners when developing the quality.

The standard is twofold. A rating may consist of several ratings. Each rating is defined for certain type of rating allowing following types of rating

- Overall rating, meaning just one overall rating covering a rating of all aspects
- Vehicle appearance, meaning how the vehicle appeared to the traveler. Covers all aspects like clean outside, clean inside, damages, noise etc
- Driver appearance, meaning how the driver appeared to the traveler, Covers all aspects except driving that has an own type
- Driving appearance, meaning how the driving appeared to the traveler.
- Punctuality, meaning how the traveler rated timing in accordance with pick-up and drop-off

For each of these rating types there is a SUTI-attribute, pls refer to this list. When using any of these attributes the source should be "SUTI:idAttributesRating".

Other sources may then be introduced, thus opening up for communication of other ratings than SUTI-ratings. Before doing so we recommend that the Client or Provider contacts SUTI technical committee. The rating types may be developed based on good arguments.

The second part of the SUTI standard is the ratingValue.

The ratingValue is allowed to be a decimal value ≥ 1 and ≤ 5 having 5 as best possible rating and 1 as worst possible rating. Thus ratings having different scales may fairly easily be converted to a SUTI accepted value and averages over time may also be possible.

The judgement behind a rating may vary in wording but normally some of the following expression are probably used

- Close to 5, very good, excellent, etc
- Close to 4, good, above expectations
- Close to 3, fair, acceptable, according to expectations
- Close to 2, bad, had shortcomings, below expectations
- Close to 1, very bad, lousy, far below expectations
- Exactly 0. no rating available

In addition the ratings may have a comment both for each ratingType but also one general covering all ratings.

In the header of ratings there is a valueCnt providing the possibility to include the number of ratings that an average value is based on. For a single rating the value should be one (1) even if it includes different ratingTypes.

There is a possibility to send the rating having valueCnt = 0 if no rating was given i.e. a traveler denied rating.

3.2.13 Order Status

The following values of Order Status shall at least be returned in case the Client sends an Order Status Request (msg 2530) and the Provider answers with a msg 2531 Order Status

Status

- Cancelled, meaning that the ordernumber is known but the order is set as cancelled
- Confirmed, The order is known and accepted. Order is waiting for dispatching time to happen

- Dispatching, the dispatching time has come and the Provider system is searching a suitable vehicle for the order
- Dispatch, a vehicle has accepted the order but the taxameter is not yet started
- Started, the vehicle has now started the vehicle equipment (taxameter) for the order
- Completed, the vehicle equipment for the order is closed normally for the order (trip, driverSession) and a receipt is created
- No-show, The pick up was unsuccessful and a no-show receipt is created. To be noted is that such a status only can occur if every pick-up (usually only one) is unsuccessful, The status may be started even if a single node has a no-show on a multinoded order.
- None, the order is not known in the Provider system

3.2.14 idAction

In message 4100 there is an idAction and SUTI has expressed a source of these values inside stateCodes. As of today two different type of actions are described:

- signalling that noVehicle or noContent is registered on node although they are either reported as event or scheduled time has been reached
- requesting or deciding on activities to connect passenger and driver.

Use cases for these actions are described in the monitoring flows in chapter 4.4 below.

4. Examples of Self Declaration Flows

4.1 Objectives of this chapter

SUTI has been developed for many years. It is therefore possible to use the protocol in a numerous different situations. It is of course also one of the main success factors, being able to adopt.

Based on the experiences during all these years there has on the same time a wish to get the communication more and more standardized in order to bring down cost to implement the communications between Client and Provider.

The work with Use Cases has been the answer to this, starting with some simple Order Flows in chapter 2 and working out Use Cases for small specific areas of communication also in chapter 2 and standardizing use of elements in chapter 3.

The work continued by defining and standardizing of Self Declarations in 2013 (see SUTI Document Relations). Since then many Self Declarations have been issued by many Clients.

In this chapter we go a little further the path of standardization by providing some good examples of complete flows that can (or should) be part of any Self Declaration.

In order to do so we have to build some general precautions for each Flow. These are also set forth before each Flow.

There is not an ambition to give examples for everything that might be part of Client-Provider communication. When defining some later flows the descriptions and examples already in chapter 2 are extensive like chapters 2.2- 2.5 going through repetitive Orders, Flagstop, Delivery Note and Accounting.

What we will focus on is the different Order Flows and Monitoring Flows.

The Order Flows concentrate on handling of a complete Order while the Monitoring Flows concentrate on situations for a node. These Monitoring Flows are not very different depending on the Order Flows but are more dependant on the types of Nodes and the situation for the Node.

The three main Order Flows that we focus on are

- Trip Order Flow, having few passengers and few nodes in a limited time frame. No or simple updates.
- Node by Node Order Flow, where the Client submit just one node at a time
- Session Order Flow with possibility for more passengers over a longer time period (Multinode orders)

Since the dispatching is different for each of above Order Flow will also contain Dispatch Flow for each of them

For the Monitoring Flows (on a single node) we have concentrated on

- Pick-up at a simple address, Prebooked
- Pick-up at a simple address, ASAP
- Pick-up at a Hub area, Prebooked
- Drop-off

For each of the use cases it is understood that the passenger has a smartphone that may at least communicate with the system of the Client by SMS. Other communication methods by app or webpage may also be possible.

In all use cases it is assumed that the Client holds the responsibility for communication with the passenger and the Provider holds the responsibility for the communication with vehicle and driver. The vehicle has some kind of equipment that continuously communicates with the Provider system exchanging positions and enables datacommunication.

4.2 Order Flows

4.2.1 Trip Order Flow

This flow is in accordance with the Typical flow briefly described in 2.1.1.2 above.

4.2.1.1 Supported Messages

All message have the format of XML (encoding UTF-8) and support the SUTI format described in SUTI version 2015. For more detail information about the supported message see attached example files.

Message	Description
MSG 2000	Order
MSG 2001	Order Confirmation
MSG 2005	Order Reject Request
MSG 2006	Order Reject Request Accepted
MSG 2007	Order Reject Request Reject
MSG 2010	Order Cancellation
MSG 2011	Order Cancellation Confirmation
MSG 3003	Dispatch Confirmation
MSG 4010	Pickup Confirmation (Event confirmation)
MSG 4031	No Contact with Vehicle
MSG 5000	Message to Vehicle
MSG 5001	Message to Vehicle Confirmation
MSG 5010	Message to Client from Vehicle
MSG 5011	Message to Client from Vehicle Confirmation
MSG 5020	Request for location
MSG 5021	Requested location
MSG 7000	Keep Alive
MSG 7001	Keep Alive Confirmation
MSG 7030	Syntax Error
MSG 7031	Not Operational

4.2.1.2 Flow order

4.2.1.2.1 Dispatch

The Client sends a 2000 message (Order), the message contains two nodes one pickup and the associated drop off.

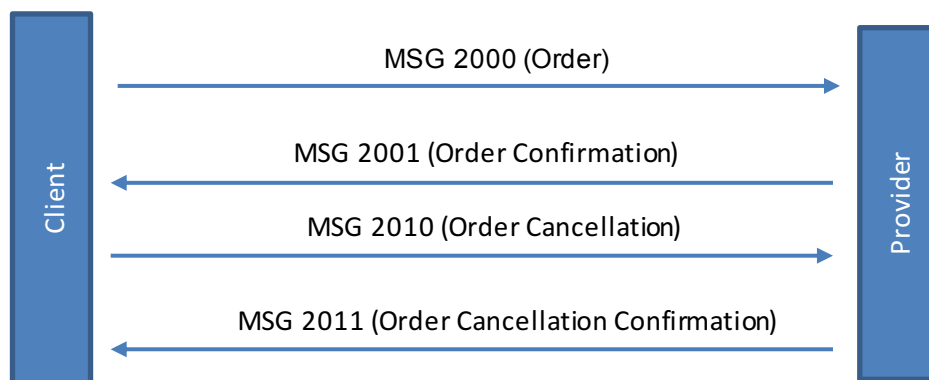
When the provider receives a 2000 message the provider responds with a 2001 message (Order Accept) with no delay, message 2002 (Order Reject) is not supported.

The 2000 message is also used when a trip is altered. When a trip is altered the client sends a 2010 message (Order Cancellation) and the provider responds with a 2011 message (Order Cancellation Confirmation). When the previously trip is cancelled a new 2000 message (containing the updated information) is sent to the provider.

The message 2010 (Order Cancellation) is initiated by the client. When the provider receives a 2010 message the provider responds with a 2011 message (Order Cancellation Confirmation).

A 2010 message is used by the client to cancel a previously sent 2000 message. All non-performed nodes will be canceled.

If the provider have accept an order with message 2001 and for some reason was not able to carry out the order the provider must manually contact the client. If the client accept the withdrawn of the order the client sends a message 2010 otherwise the order shall be executed.



4.2.1.2.2 Dispatch using "anbudssområde"



4.2.1.3 Messages

All messages are XML files following the SUTI format described in SUTI version 2015

4.2.1.3.1 OrgSender and orgReceiver nodes

These nodes are found in all communications. It is important to remember that orgSender is always the sender of the message and orgReceiver is always the receiver of the message.

```

<orgSender name="PASS">
  <idOrg src="SUTI:link" id="trapeze_pass_0001" unique="true"/>
</orgSender>
<orgReceiver name="HDCOM">
  <idOrg src="SUTI:link" id="holmedal_hdcom_0001" unique="true"/>
</orgReceiver>
  
```

4.2.1.3.2 MSG 2000 Order

MSG 2000 is sent from the Client to the Provider. It indicates the start of a run of the circulation. After receiving telegram 2000 the Provider sends a 2001 order confirmation to the Client.

Client action

Make the order readable by the Provider.

Make the order understandable by the Provider.

Send the order in due time giving the Provider time enough to process the order and dispatch necessary resources.

Ensure the order contain all needed information to the Provider.

Provider action

Ensure the order is received in full.

Ensure the order is generated in the provider system in full.

Immediately respond MSG 2000 with MSG 2001.

Note that each node has an identifier ("trapeze_pass_0001:idSubOrder") to be returned as a reference to the node in the telegram 4010.

```

<msg msgType="2000" msgName="Order">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  
```

```

<order>
  <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
  <agreement>
    <idAgreement src="trapeze_pass_0001:idAgreement" id="123456"/>
  </agreement>
  <process allowRouting="false" trafficControl="true" pickupconfirmation="standard" orderAlteration="false"
dispatchResponsible="provider" report="false" preorderedVehicle="false" dispatch="true" manualDispatch="false"/>
  <resourceOrder>
    <vehicle>
      <idVehicle src="trapeze_pass_0001:idVehicle" id="51006P"/>
      <capacity>
        <seats noOfSeats="4"/>
      </capacity>
      <attributesVehicle>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1628" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1611" unique="true"/>
        </attribute>
      </attributesVehicle>
    </vehicle>
    <driver>
      <attributesDriver>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
        </attribute>
      </attributesDriver>
    </driver>
  </resourceOrder>
  <route>
    <node nodeSeqno="1" nodeType="pickup">
      <addressNode addressName="" street="Baravägen" streetNo="1" streetNoLetter="" community="Lund"
postalNo="22240">
        <geographicLocation typeOfCoordinate="WGS-84" lat="55.718095" long="13.190738" precision="6"/>
      </addressNode>
      <timesNode>
        <time timeType="estimatedtime" time="2015-08-10T08:00:18"/>
        <time timeType="scheduledtime" time="2015-08-10T08:00:00"/>
      </timesNode>
      <contents>
        <content contentType="traveller" name="John Doe">
          <idContent src="trapeze_pass_0001:ContentId" id="9999"/>
          <attributeContent>
            <attribute>
              <idAttribute src="SUTI:idAttribute" id="1001" unique="true"/>
            </attribute>
          </attributeContent>
          <economyContent>
            <formOfPayment>
              <payment amount="0" paymentType="prepaidsocialfee"/>
            </formOfPayment>
          </economyContent>
          <contactInfosContent>
            <contactInfo contactType="phone" contactInfo="0431-74700"/>
            <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
          </contactInfosContent>
          <subOrderContent>
            <idOrder src="trapeze_pass_0001:idSubOrder" id="11183750" unique="true"/>
          </subOrderContent>
        </content>
      </contents>
    </node>
    <node nodeSeqno="2" nodeType="destination">
      <addressNode addressName="" street="Kalkstensvägen" streetNo="14" streetNoLetter="" community="Lund"
postalNo="22240">
        <geographicLocation typeOfCoordinate="WGS-84" lat="55.690147" long="13.215330" precision="6"/>
      </addressNode>
    </node>
  </route>
</order>

```

```

</addressNode>
<timesNode>
  <time timeType="estimatedtime" time="2015-08-10T08:07:18"/>
  <time timeType="scheduledtime" time="2015-08-10T08:07:00"/>
</timesNode>
<contents>
  <content contentType="traveller" name="John Doe">
    <idContent src="trapeze_pass_0001:ContentId" id="9999"/>
    <attributeContent>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="1001" unique="true"/>
      </attribute>
    </attributeContent>
    <economyContent>
      <formOfPayment>
        <payment amount="0" paymentType="prepaidsocialfee"/>
      </formOfPayment>
    </economyContent>
    <contactInfosContent>
      <contactInfo contactType="phone" contactInfo="0431-74700"/>
      <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
    </contactInfosContent>
    <subOrderContent>
      <idOrder src="trapeze_pass_0001:idSubOrder" id="11183750" unique="true"/>
    </subOrderContent>
  </content>
</contents>
</node>
</route>
</order>
</msg>

```

4.2.1.3.3 **MSG 2001 Order confirmation**

MSG 2001 is sent from the Provider to the Client as a response to a received MSG 2000. It indicates that the order is received, comply with the referred agreement and will be carried out.

Client action

Perform necessary updates to indicate that the order is accepted by the Provider.
Ensure that the order don't get send to another Provider.

Provider action

Give the order an identity or booking number.
Inform the Client about the assigned identity or booking identity.

```

<msg msgType="2001" msgName="Order confirmation">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  </referencesTo>
</msg>

```

4.2.1.3.4 MSG 2005 Order Reject Request

MSG 2005 is a request for a rejection of an order. It is sent by the provider to the client to request to reject an order that has been accepted in an earlier stages.

Client action

Check if it's possible to make a rejection and if it's possible make necessary updates in the client system to ensure that the order has been rejected by the provider.

Provider action

Be prepared to receive an answer to the request.

```
<msg msgType="2005" msgName="Order Reject Request">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  </referencesTo>
</msg>
```

4.2.1.3.5 MSG 2006 Order Reject Request Accepted

MSG 2006 is a response of MSG 2005. It is sent by the client to inform the provider that the request for rejection in MSG 2005 is accepted and will be carried out.

Client action

Make necessary updates in the client system to ensure that the order is rejected.

Provider action

Make necessary updates in the provider system to ensure that the order is rejected.

```
<msg msgType="2006" msgName="Order Reject Request accepted">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795236" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  </referencesTo>
</msg>
```

4.2.1.3.6 MSG 2007 Order Reject Request Reject

MSG 2007 is a response of MSG 2005. It is sent by the Client to inform the Provider that the requested rejection in MSG 2005 is rejected and the order shall be carried out by the Provider.

Client action

Make necessary updates in the client system to ensure that the order is not rejected and will be carried out by the Provider.

Provider action

Make necessary updates in the provider system to ensure that the order will be carried out.

```
<msg msgType="2007" msgName="Order Reject Request reject">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795237" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  </referencesTo>
</msg>
```

4.2.1.3.7 MSG 2010 Order Cancellation Request

MSG 2010 is a request from the Client to cancel a previous sent and accepted MSG 2000. The Provider is expected to try to cancel the order as a consequence of this message.

Client action

Ensure that the order id of the order in the cancellation request is correct.

Provider action

Immediately process the cancellation request.

Inform the client that cancellation is carried out without consequence by sending MSG 2011.

```
<msg msgType="2010" msgName="Order Cancellation">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121409309255" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  </referencesTo>
</msg>
```

4.2.1.3.8 **MSG 2011 Order cancellation accepted**

MSG 2011 is a response of MSG 2010. It is sent by the Provider to inform the Client that the cancellation in MSG 2010 is accepted and will be carried out without further consequences.

Client action

Make necessary updates in the client system to ensure that the order is cancelled.

Provider action

Make necessary updates in the provider system to ensure that the order is cancelled and will not be carried out without further consequence.

```
<msg msgType="2011" msgName="Order Cancellation accepted">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409309256" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121409309255" unique="true"/>
  </referencesTo>
</msg>
```

4.2.2 Node by Node Flow

This flow is in accordance with the Node by Node flow briefly described in 2.1.1.3 above.

Compared to the Trip Node Flow this type of order is built up by Client sending one node. The nodes are sent in the order that the driver request them and the Client accept the nodes to be submitted. The whole order is a circulation of a vehicle for a longer timeperiod.

As a consequence the order flow is simpler but instead you will find a sequence of submitting new nodes.

4.2.2.1 Flow of start of new circulation:

The Client sends a 2000 message (Order), the message contains two nodes the first pickup of the run and as a destination node the time of SchemaVehicleEndTime (end time of the run).

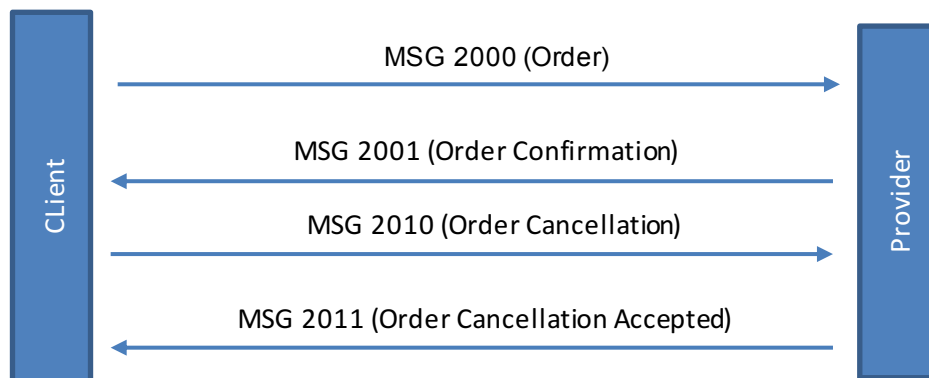
When the provider receives a 2000 message the provider response with a 2001 message (OrderAccept).

The message 2010 (Order Cancellation) is initiated by the client. When the provider receives a 2010 message the provider response with a 2011 message (Order Cancellation Confirmation).

A 2010 message is used by the client to cancel a previously sent 2000 message. All non-performed nodes will be canceled.

If the provider have accept an order with message 2001 and for some reason was not able to carry out the order the provider must manually contact the client. If the client accept the withdrawn of the order the client sends a message 2010 otherwise the order shall be executed.

A comment is that some systems also use the msg 1920 ResourceAllocation to start the circulation in stead of msg 2000. In such cases the response should either be 1921 Accept or 1922 Reject.



4.2.2.2 Flow provider wants next node:

When the provider wants the next node of the run (next pickup, drop off) the provider sends a 4000 message (Request for Traffic Information). When the client receives a 4000 message the client responds with a 4001 message (Requested traffic information) or a 4020 message (Order end).

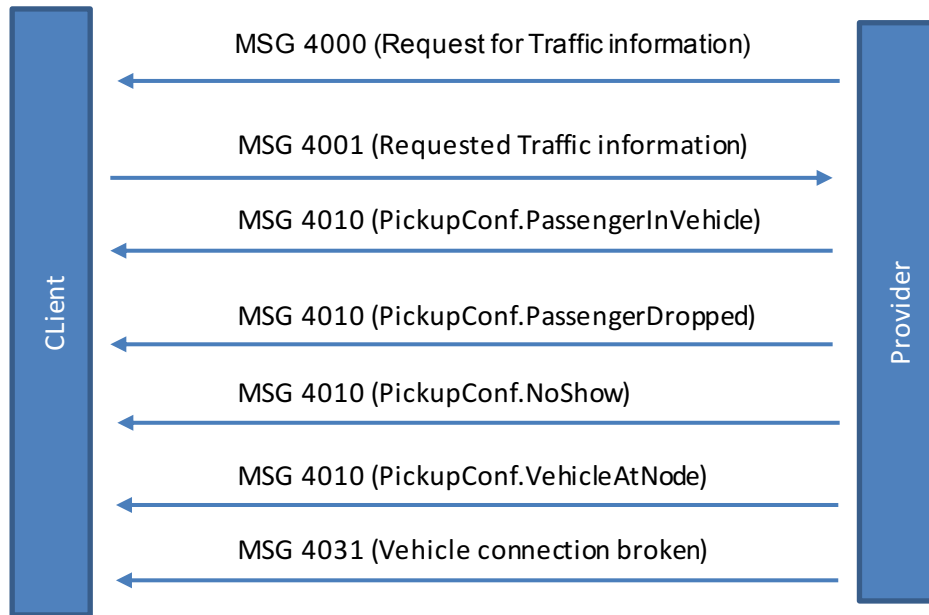
The 4020 message is send to the provider if there is no more nodes on the current run.

When the provider has selected a vehicle hence the client has received a 3003 message the provider can send a message 4010 to the client.

Supported event types in 4010 message

EventType	Description
PassengerInVehicle (1701)	Passenger in vehicle
PassengerDropped (1702)	Passenger dropped
NoShow (1703)	When the passenger is not on the agreed location
VehicleAtNode (1709)	When the vehicle arrives at the agreed location

In case where the vehicle is out of range (no contact) the provider can send a 4031 message to the client.



4.2.2.3 Messages

All messages are XML files following the SUTI format described in SUTI version 2015

4.2.2.3.1 orgSender and orgReceiver nodes

These nodes are found in all communications. It is important to remember that orgSender is always the sender of the message and orgReceiver is always the receiver of the message.

```

<orgSender name="PASS">
  <idOrg src="SUTI:alink" id="trapeze_pass_0001" unique="true"/>
</orgSender>
<orgReceiver name="HDCOM">
  <idOrg src="SUTI:alink" id="holmedal_hdcom_0001" unique="true"/>
</orgReceiver>
  
```

4.2.2.3.2 MSG 2000 Order

MSG 2000 is sent from the Client to the Provider. It indicates the start of a run of the circulation. After receiving telegram 2000 the Provider sends a 2001 order confirmation to the Client.

Note that MSG 2000 only includes the first pickup node and the schema vehicles end time as the destination node.

```

<msg msgType="2000" msgName="Order">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  <order>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <agreement>
      <idAgreement src="trapeze_pass_0001:idAgreement" id="123456"/>
    </agreement>
    <process allowRouting="false" trafficControl="true" pickupconfirmation="standard" orderAlteration="false"
    dispatchResponsible="provider" report="false" preorderedVehicle="false" dispatch="true" manualDispatch="false"/>
    <resourceOrder>
      <vehicle>
        <idVehicle src="trapeze_pass_0001:idVehicle" id="51006P" unique="true"/>
        <capacity>
          <seats noOfSeats="4"/>
        </capacity>
      </vehicle>
    </resourceOrder>
  </order>
</msg>
  
```

```

    <attributesVehicle>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="1628" unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="1611" unique="true"/>
      </attribute>
    </attributesVehicle>
  </vehicle>
  <driver>
    <attributesDriver>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
      </attribute>
    </attributesDriver>
  </driver>
</resourceOrder>
<route>
  <node nodeSeqno="1" nodeType="pickup">
    <addressNode addressName="" street="Baravägen" streetNo="1" streetNoLetter="" community="Lund"
postalNo="22240">
      <geographicLocation typeOfCoordinate="WGS-84" lat="55.718095" long="13.190738" precision="6"/>
    </addressNode>
    <timesNode>
      <time timeType="estimatedtime" time="2015-08-10T08:00:18"/>
      <time timeType="scheduledtime" time="2015-08-10T08:00:00"/>
    </timesNode>
    <contents>
      <content contentType="traveller" name="John Doe">
        <attributeContent>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="1001" unique="true"/>
          </attribute>
        </attributeContent>
        <economyContent>
          <formOfPayment>
            <payment amount="0" paymentType="prepaidsocialfee"/>
          </formOfPayment>
        </economyContent>
        <contactInfosContent>
          <contactInfo contactType="phone" contactInfo="0431-74700"/>
          <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
        </contactInfosContent>
        <subOrderContent>
          <!-- Used in telegram 4010-->
          <idOrder src="trapeze_pass_0001:idSubOrder" id="11183743" unique="true"/>
        </subOrderContent>
      </content>
    </contents>
  </node>
  <node nodeSeqno="2" nodeType="destination">
    <addressNode/>
    <timesNode>
      <time timeType="scheduledtime" time="2015-08-10T16:32:26"/>
    </timesNode>
  </node>
</route>
</order>
</msg>

```


4.2.2.3.3 MSG 2001 Order confirmation

MSG 2001 is sent from the Provider to the Client as a response to a received MSG 2000. It indicates that the order is received, comply with the referred agreement and will be carried out.

```
<msg msgType="2001" msgName="Order confirmation">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  </referencesTo>
</msg>
```

4.2.2.3.4 MSG 2010 Order Cancellation Request

MSG 2010 is a request from the Client to cancel a previous sent and accepted MSG 2000. The Provider is expected to try to cancel the order as a consequence of this message.

```
<msg msgType="2010" msgName="Order Cancellation">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121409309255" unique="true"/>
  <referencesTo>
    <idOrder src="trapeze_pass_0001:idOrder" id="15121403501" unique="true"/>
  </referencesTo>
</msg>
```

4.2.2.3.5 MSG 2011 Order cancellation accepted

MSG 2011 is a response of MSG 2010. It is sent by the Provider to inform the Client that the cancellation in MSG 2010 is accepted and will be carried out without further consequences

```
<msg msgType="2011" msgName="Order Cancellation accepted">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409309256" unique="true"/>
  <referencesTo>
    <idOrder src="trapeze_pass_0001:idOrder" id="15121403501" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121409309255" unique="true"/>
  </referencesTo>
</msg>
```

4.2.2.3.6 MSG 3003 Dispatch confirmation

The Provider use MSG 3003 to inform the Client that the suggested and accepted vehicle has accepted the order and will perform it.

```
<msg msgType="3003" msgName="Dispatch confirmation">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015122213393416" unique="true"/>
  <referencesTo>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742"/>
    <idVehicle src="holmedal_hdcom_0001:idRegistrationNumber" id="ABC123" unique="true"/>
  </referencesTo>
  <resourceDispatch>
    <vehicle>
      <idVehicle src="trapeze_pass_0001:idVehicle" id="51006P" unique="true"/>
      <idVehicle src="holmedal_hdcom_0001:idVehicle" id="3110" unique="true"/>
      <capacity>
        <seats noOfSeats="4"/>
      </capacity>
      <attributesVehicle>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1628" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1611" unique="true"/>
        </attribute>
      </attributesVehicle>
    </vehicle>
  </resourceDispatch>
</msg>
```

```

</vehicle>
<manualDescriptionResource sendtoInvoice="false" vehicleConfirmation="false" sendtoOperator="true"
sendtoVehicle="false" manualText="">
  <idActionText src="holmedal_hdcom_0001:driverEndTime" id="2015-12-23T00:36:29"/>
</manualDescriptionResource>
<driver>
  <attributesDriver>
    <attribute>
      <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
    </attribute>
    <attribute>
      <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
    </attribute>
  </attributesDriver>
  <idDriver src="holmedal_hdcom_0001:idDriver" id="123456" unique="true"/>
  <contactInfoDriver>
    <contactInfo contactType="phone" contactInfo="043174700"/>
    <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
  </contactInfoDriver>
</driver>
<vehiclestartLocation typeOfCoordinate="WGS-84" precision="6" lat="56.047010" long="12.703940"/>
</resourceDispatch>
</msg>

```

4.2.2.3.7 MSG 4000 Request for traffic information

MSG 4000 is sent from the Provider/vehicle to the Client as a request for information about an ongoing order. For example a vehicle requests the next pickup point or requests traffic information.

```

<msg msgType="4000" msgName="Request for Traffic Information">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409309255" unique="true"/>
  <infoTimeStamp>
    <time timeType="actual" time="2015-12-14T09:27:53"/>
  </infoTimeStamp>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
  </referencesTo>
  <resourceDispatch>
    <vehicle>
      <idVehicle src="trapeze_pass_0001:idVehicle" id="51006P" unique="true"/>
      <idVehicle src="holmedal_hdcom_0001:idVehicle" id="3110" unique="true"/>
      <capacity>
        <seats noOfSeats="4"/>
      </capacity>
      <attributesVehicle>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1628" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1611" unique="true"/>
        </attribute>
      </attributesVehicle>
      <vehicleCurrentLocation typeOfCoordinate="WGS-84" lat="57.925737" long="12.524865" precision="6"/>
    </vehicle>
    <driver>
      <attributesDriver>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
        </attribute>
        <attribute>
          <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
        </attribute>
      </attributesDriver>
    </driver>
  </resourceDispatch>
</msg>

```

4.2.2.3.8 MSG 4001 Requested traffic information

MSG 4001 is sent from the Client to the Provider in response of a MSG 4000. MSG 4001 contains the requested information, alternative information or informing that no information is available at the moment.

```
<msg msgType="4001" msgName="Requested Traffic Information">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121409016233"/>
  <referencesTo>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409309255"/>
  </referencesTo>
  <order>
    <idOrder src="trapeze_pass_0001:idOrder" id="15121403501" unique="true"/>
    <resourceOrder>
      <vehicle>
        <idVehicle src="trapeze_pass_0001:idVehicle" id="51006P" unique="true"/>
        <capacity>
          <seats noOfSeats="4"/>
        </capacity>
        <attributesVehicle>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="1628" unique="true"/>
          </attribute>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="1611" unique="true"/>
          </attribute>
        </attributesVehicle>
      </vehicle>
      <driver>
        <attributesDriver>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
          </attribute>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
          </attribute>
        </attributesDriver>
      </driver>
    </resourceOrder>
    <route>
      <node nodeSeqno="1" nodeType="destination">
        <addressNode addressName="ICA Maxi" street="Baravägen" streetNo="1" streetNoLetter="" community="Lund"
postalNo="22240">
          <geographicLocation typeOfCoordinate="WGS-84" lat="55.718095" long="13.190738" precision="6"/>
        </addressNode>
        <timesNode>
          <time timeType="scheduledtime" dwellTime="3" time="2015-08-10T08:00:18"/>
          <time timeType="estimatedtime" dwellTime="3" time="2015-08-10T08:00:18"/>
        </timesNode>
        <contents>
          <content contentType="traveller" name=" John Doe">
            <attributeContent>
              <attribute>
                <idAttribute src="SUTI:idAttribute" id="1001" unique="true"/>
              </attribute>
            </attributeContent>
            <economyContent>
              <formOfPayment>
                <payment amount="0" paymentType="prepaidsocialfee"/>
              </formOfPayment>
            </economyContent>
            <contactInfosContent>
              <contactInfo contactType="phone" contactInfo="0431-74700"/>
              <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
            </contactInfosContent>
            <subOrderContent>
              <idOrder src="trapeze_pass_0001:idSubOrder" id="11183752" unique="true"/>
            </subOrderContent>
          </content>
        </contents>
      </node>
    </route>
  </order>
</msg>
```

```
</content>  
</contents>  
</node>  
</route>  
</order>  
</msg>
```

4.2.3 Order by Order Flow

This flow is in accordance with the Order by Order flow briefly described in 2.1.1.6 above.

4.2.3.1 Flow order

There are two types of dispatch models Dispatch and Dispatch using anbudsområde.

4.2.3.2 Dispatch

The Client sends a 2000 message (Order), the message contains all nodes in a single trip.
The 2000 message can contain several pickups, drop off and action nodes.

When the provider receives a 2000 message the provider responds with a 2001 message (Order Accept) with no delay, message 2002 (Order Reject) is only implemented that we log that the provider sends it.
The message 2010 (Order Cancellation) is initiated by the client. When the provider receives a 2010 message the provider responds with a 2011 message (Order Cancellation Accepted).
A 2010 message is used by the client to cancel a previously sent 2000 message. All non-performed nodes will be canceled.

4.2.3.2.1 Provider cancelation

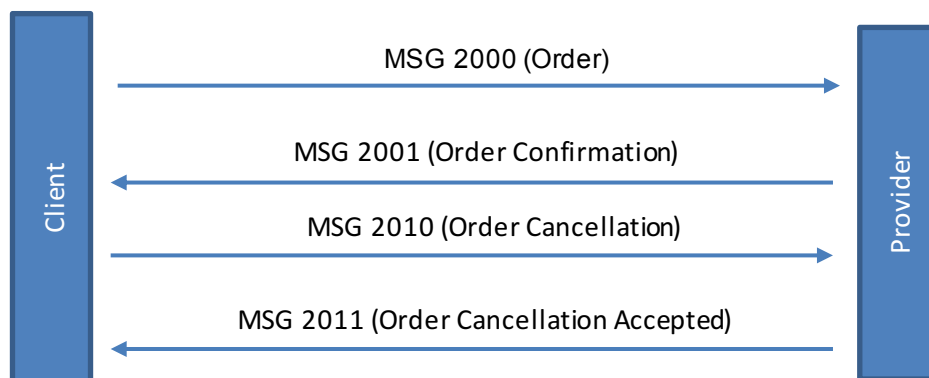
If the provider has accepted an order with message 2001 and for some reason was not able to carry out the order the provider must manually contact the client. If the client accepts the withdrawn of the order the client sends a message 2010 otherwise the order shall be executed.

4.2.3.2.2 Order alteration

The Client send a new updated order with the same IdOrder as the previously sent order and with the process flag orderAlteration = true.

The 2000 message has a referencesTo section that contains known ids that the Client has received from the Providers 2001 and 3003 messages.

The referencesTo section contains references from both Client and Provider side.



4.2.3.3 Dispatch using "anbudsområde"

The Client sends a 2000 message (Order), the message contains all nodes in a single trip.
The 2000 message can contain several pickups, drop off and action nodes.

When the provider receives a 2000 message the provider responds with a 2001 message (Order Accept) with no delay, message 2002 (Order Reject) is only implemented that we log that the provider sends it.
The message 2010 (Order Cancellation) is initiated by the client. When the provider receives a 2010 message the provider responds with a 2011 message (Order Cancellation Accepted).
A 2010 message is used by the client to cancel a previously sent 2000 message. All non-performed nodes will be canceled.

4.2.3.3.1 Provider cancelation

If the provider has accepted an order with message 2001 and for some reason was not able to carry out the order the provider can send message 2005 (Order Reject Request). Depending on whether the provider is last candidate that can receive the order the client response will be.

If the provider is NOT, the last candidate

1. Client sends message 2006 (Reject Request Accepted) to the provider.
2. Client sends message 2010 (Order Cancellation) to the provider.
3. Provider confirms with message 2011 (Order Cancellation Confirmation).
4. Client send the order to the next candidate.

If the provider is the last candidate

1. Client sends message 2007 (Order Reject Request Rejected).

If the provider has received message 2007 (Order Reject Request Rejected) and for some reason was not able to carry out the order the provider must manually contact the client.

4.2.3.3.2 Order alteration

The Client send a new updated order with the same IdOrder as the previously sent order and with the process flag orderAlteration = true.

The 2000 message has a referencesTo section that contains known ids that the Client has received from the Providers 2001 and 3003 messages.

The referencesTo section contains references from both Client and Provider side.



4.2.3.4 Messages

All messages are XML files following the SUTI format described in SUTI version 2015

4.2.3.4.1 OrgSender and orgReceiver nodes

These nodes are found in all communications. It is important to remember that orgSender is always the sender of

the message and orgReceiver is always the receiver of the message.

```
<orgSender name="PASS">
  <idOrg src="SUTI:link" id="trapeze_pass_0001" unique="true"/>
</orgSender>
<orgReceiver name="HDCOM">
  <idOrg src="SUTI:link" id="holmedal_hdcom_0001" unique="true"/>
</orgReceiver>
```

4.2.3.4.2 MSG 2000 Order

MSG 2000 is sent from the Client to the Provider. It indicates the start of a run of the circulation. After receiving telegram 2000 the Provider sends a 2001 order confirmation to the Client.

Client action

Make the order readable by the Provider.

Make the order understandable by the Provider.

Send the order in due time giving the Provider time enough to process the order and dispatch necessary resources.

Ensure the order contain all needed information to the Provider.

Provider action

Ensure the order is received in full.

Ensure the order is generated in the provider system in full.

Immediately respond MSG 2000 with MSG 2001.

Note that each node has an identifier ("trapeze_pass_0001:idSubOrder") to be returned as a reference to the node in the telegram 4010.

```
<msg msgType="2000" msgName="Order">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  <order>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <agreement>
      <idAgreement src="trapeze_pass_0001:idAgreement" id="123456"/>
    </agreement>
    <process allowRouting="false" trafficControl="true" pickupconfirmation="standard" orderAlteration="false"
dispatchResponsible="provider" report="false" preorderedVehicle="false" dispatch="true" manualDispatch="false"/>
    <resourceOrder>
      <vehicle>
        <idVehicle src="trapeze_pass_0001:idVehicle" id="51006P"/>
      </vehicle>
    </resourceOrder>
    <route>
      <node nodeSeqno="1" nodeType="pickup">
        <addressNode addressName="" street="Baravägen" streetNo="1" streetNoLetter="" community="Lund"
postalNo="22240">
          <geographicLocation typeOfCoordinate="WGS-84" lat="55.718095" long="13.190738" precision="6"/>
        </addressNode>
        <timesNode>
          <time timeType="estimatedtime" time="2015-08-10T08:00:18"/>
          <time timeType="scheduledtime" time="2015-08-10T08:00:00"/>
        </timesNode>
        <contents>
          <content contentType="traveller" name="John Doe">
            <idContent src="trapeze_pass_0001:ContentId" id="5319397" unique="true"/>
            <attributeContent>
              <attribute>
                <idAttribute src="SUTI:idAttribute" id="1001" unique="true"/>
              </attribute>
            </attributeContent>
            <economyContent>
              <formOfPayment>
                <payment amount="0" paymentType="prepaidsocialfee"/>
              </formOfPayment>
            </economyContent>
            <resourceContent>
              <vehicle>
                <capacity>
                  <seats noOfSeats="1" noOfItems="1"/>
                </capacity>
              </vehicle>
            </resourceContent>
          </content>
        </contents>
      </node>
    </route>
  </order>
</msg>
```

```

    <attributesVehicle>
      <attribute>
        <idAttribute src="SUTI:idLink" id="1628" unique="true"/>
      </attribute>
    </attributesVehicle>
  </vehicle>
  <driver>
    <attributesDriver>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
      </attribute>
      <attribute>
        <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
      </attribute>
    </attributesDriver>
  </driver>
</resourceContent>
<contactInfosContent>
  <contactInfo contactType="phone" contactInfo="0431-74700"/>
  <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
</contactInfosContent>
<subOrderContent>
  <idOrder src="trapeze_pass_0001:idSubOrder" id="11183750" unique="true"/>
</subOrderContent>
</content>
</contents>
</node>
<node nodeSeqno="2" nodeType="destination">
  <addressNode addressName="" street="Kalkstensvägen" streetNo="14" streetNoLetter="" community="Lund"
postalNo="22240">
    <geographicLocation typeOfCoordinate="WGS-84" lat="55.690147" long="13.215330" precision="6"/>
  </addressNode>
  <timesNode>
    <time timeType="estimatedtime" time="2015-08-10T08:07:18"/>
    <time timeType="scheduledtime" time="2015-08-10T08:07:00"/>
  </timesNode>
  <contents>
    <content contentType="traveller" name="John Doe">
      <idContent src="trapeze_pass_0001:ContentId" id="5319397" unique="true"/>
      <attributeContent>
        <attribute>
          <idAttribute src="SUTI:idLink" id="1001" unique="true"/>
        </attribute>
      </attributeContent>
      <economyContent>
        <formOfPayment>
          <payment amount="0" paymentType="prepaidsocialfee"/>
        </formOfPayment>
      </economyContent>
      <subOrderContent>
        <idOrder src="trapeze_pass_0001:idSubOrder" id="11183750" unique="true"/>
      </subOrderContent>
    </content>
  </contents>
</node>
<node nodeSeqno="3" nodeType="action" subnodeType="break">
  <addressNode addressName="" street="" streetNoLetter="" community="" postalNo="">
    <geographicLocation typeOfCoordinate="WGS-84" lat="55.718095" long="13.190738" precision="6"/>
  </addressNode>
  <timesNode>
    <time timeType="estimatedtime" time="2015-08-10T08:00:18"/>
    <time timeType="scheduledtime" time="2015-08-10T08:00:00"/>
  </timesNode>
  <contents/>
</node>
<node nodeSeqno="4" nodeType="pickup">
  <addressNode addressName="" street="Vikingavägen" streetNo="17" streetNoLetter="" community="Lund"
postalNo="22240">
    <geographicLocation typeOfCoordinate="WGS-84" lat="55.695565" long="13.242720" precision="6"/>
  </addressNode>

```



```

<timesNode>
  <time timeType="estimatedtime" time="2015-08-10T08:12:18"/>
  <time timeType="scheduledtime" time="2015-08-10T08:12:00"/>
</timesNode>
<contents>
  <content contentType="traveller" name="Jane Doe">
    <idContent src="trapeze_pass_0001:ContentId" id="446699" unique="true"/>
    <attributeContent>
      <attribute>
        <idAttribute src="SUTI:idLink" id="1001" unique="true"/>
      </attribute>
    </attributeContent>
    <economyContent>
      <formOfPayment>
        <payment amount="0" paymentType="prepaidsocialfee"/>
      </formOfPayment>
    </economyContent>
    <resourceContent>
      <vehicle>
        <capacity>
          <seats noOfSeats="1" noOfItems="1"/>
        </capacity>
        <attributesVehicle>
          <attribute>
            <idAttribute src="SUTI:idLink" id="1611" unique="true"/>
          </attribute>
        </attributesVehicle>
      </vehicle>
      <driver>
        <attributesDriver>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="1510" unique="true"/>
          </attribute>
          <attribute>
            <idAttribute src="SUTI:idAttribute" id="2313" unique="true"/>
          </attribute>
        </attributesDriver>
      </driver>
    </resourceContent>
    <contactInfosContent>
      <contactInfo contactType="phone" contactInfo="0431-74700"/>
      <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
    </contactInfosContent>
    <subOrderContent>
      <idOrder src="trapeze_pass_0001:idSubOrder" id="11183751" unique="true"/>
    </subOrderContent>
  </content>
</contents>
</node>
<node nodeSeqno="5" nodeType="destination">
  <addressNode addressName="" street="Latinlinjen" streetNo="11" streetNoLetter="" community="Lund"
postalNo="22240">
    <geographicLocation typeOfCoordinate="WGS-84" lat="55.708449" long="13.244673" precision="6"/>
  </addressNode>
  <timesNode>
    <time timeType="estimatedtime" time="2015-08-10T08:20:18"/>
    <time timeType="scheduledtime" time="2015-08-10T08:20:00"/>
  </timesNode>
  <contents>
    <content contentType="traveller" name="Jane Doe">
      <idContent src="trapeze_pass_0001:ContentId" id="446699" unique="true"/>
      <attributeContent>
        <attribute>
          <idAttribute src="SUTI:idLink" id="1001" unique="true"/>
        </attribute>
      </attributeContent>
      <economyContent>
        <formOfPayment>
          <payment amount="0" paymentType="prepaidsocialfee"/>
        </formOfPayment>
      </economyContent>
    </content>
  </contents>
</node>

```

```

    </economyContent>
    <subOrderContent>
      <idOrder src="trapeze_pass_0001:idSubOrder" id="11183751" unique="true"/>
    </subOrderContent>
  </content>
</contents>
</node>
</route>
</order>
</msg>

```

4.2.3.4.3 MSG 2000 Order alteration

MSG 2000 with process flag orderAlteration = true is sent from the Client to the Provider when the order needs to be updated. After receiving telegram 2000 the Provider sends a 2001 order confirmation to the Client.

Client action

Make the order readable by the Provider.

Make the order understandable by the Provider.

Send the order in due time giving the Provider time enough to process the order and dispatch necessary resources.

Ensure the order contain all needed information to the Provider.

Provider action

Ensure the order is received in full.

Ensure the order is generated in the provider system in full.

Immediately respond MSG 2000 with MSG 2001.

Note that each node has an identifier ("trapeze_pass_0001:idSubOrder") to be returned as a reference to the node in the telegram 4010.

```

<msg msgType="2000" msgName="Order">
  <idMsg src="trapeze_pass_0001:MsgId" id="2017032316255334" unique="true"/>
  <referencesTo>
    <idOrder src="trapeze_pass_0001:idOrder" id="71410951" unique="true"/>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="32170" unique="true"/>
    <idMsg src="trapeze_pass_0001:MsgId" id="2017032316255333" unique="true"/>
    <idVehicle src="trapeze_pass_0001:VehicleId" id="51006P" unique="true"/>
    <idVehicle src="holmedal_hdcom_0001:VehicleId" id="9996" unique="true"/>
  </referencesTo>
  <order>
    <idOrder src="trapeze_pass_0001:idOrder" id="71420951" unique="true"/>
    <agreement>
      <idAgreement src="trapeze_pass_0001:idAgreement" id="123456"/>
    </agreement>
    <process allowRouting="false" trafficControl="true" pickupconfirmation="standard" orderAlteration="true"
dispatchResponsible="provider" report="false" preorderedVehicle="false" dispatch="true" manualDispatch="false"/>
    <resourceOrder>
      <vehicle>
        <idVehicle src="trapeze_pass_0001:VehicleId" id="51006P" unique="true"/>
      </vehicle>
    </resourceOrder>
    <route>
      <node nodeSeqno="1" nodeType="pickup">
        <addressNode addressName="" street="Baravägen" streetNo="1" streetNoLetter="" community="Lund"
postalNo="22240">
          <geographicLocation typeOfCoordinate="WGS-84" lat="55.718095" long="13.190738" precision="6"/>
        </addressNode>
        <timesNode>
          <time timeType="scheduledtime" time="2015-08-10T08:00:00"/>
          <time timeType="estimatedtime" time="2015-08-10T08:00:18"/>
        </timesNode>
        <contents>
          <content contentType="traveller" name="John Doe">
            <idContent src="trapeze_pass_0001:ContentId" id="5319397" unique="true"/>
          </attributeContent>

```

```

    <attribute>
      <idAttribute src="SUTI:idAttribute" id="1001" unique="true"/>
    </attribute>
  </attributeContent>
  <economyContent>
    <formOfPayment>
      <payment amount="0" paymentType="prepaidsocialfee"/>
    </formOfPayment>
  </economyContent>
  <resourceContent>
    <vehicle>
      <capacity>
        <seats noOfSeats="1" noOfItems="1"/>
      </capacity>
      <attributesVehicle>
        <attribute>
          <idAttribute src="SUTI:idLink" id="1628" unique="true"/>
        </attribute>
      </attributesVehicle>
    </vehicle>
  </resourceContent>
  <contactInfosContent>
    <contactInfo contactType="phone" contactInfo="0431-74700"/>
    <contactInfo contactType="mail" contactInfo="info@holmedal.se"/>
  </contactInfosContent>
  <subOrderContent>
    <idOrder src="trapeze_pass_0001:idSubOrder" id="11183750" unique="true"/>
  </subOrderContent>
</content>
</contents>
</node>
<node nodeSeqno="2" nodeType="destination">
  <addressNode addressName="" street="Kalkstensvägen" streetNo="14" streetNoLetter="" community="Lund"
postalNo="22240">
    <geographicLocation typeOfCoordinate="WGS-84" lat="55.690147" long="13.215330" precision="6"/>
  </addressNode>
  <timesNode>
    <time timeType="estimatedtime" time="2015-08-10T08:07:18"/>
    <time timeType="scheduledtime" time="2015-08-10T08:07:00"/>
  </timesNode>
  <contents>
    <content contentType="traveller" name="John Doe">
      <idContent src="trapeze_pass_0001:ContentId" id="5319397" unique="true"/>
      <attributeContent>
        <attribute>
          <idAttribute src="SUTI:idLink" id="1001" unique="true"/>
        </attribute>
      </attributeContent>
      <economyContent>
        <formOfPayment>
          <payment amount="0" paymentType="prepaidsocialfee"/>
        </formOfPayment>
      </economyContent>
      <subOrderContent>
        <idOrder src="trapeze_pass_0001:idSubOrder" id="11183750" unique="true"/>
      </subOrderContent>
    </content>
  </contents>
</node>
</route>
</order>
</msg>

```

4.2.3.4.4 MSG 2001 Order confirmation

MSG 2001 is sent from the Provider to the Client as a response to a received MSG 2000. It indicates that the order is received, comply with the referred agreement and will be carried out.

Client action

Perform necessary updates to indicate that the order is accepted by the Provider.
Ensure that the order don't get send to another Provider.

Provider action

Give the order an identity or booking number.
Inform the Client about the assigned identity or booking identity.

```
<msg msgType="2001" msgName="Order confirmation">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  </referencesTo>
</msg>
```

4.2.3.4.5 MSG 2005 Order Reject Request

MSG 2005 is a request for a rejection of an order. It is sent by the provider to the client to request to reject an order that has been accepted in an earlier stages.

Client action

Check if it's possible to make a rejection and if it's possible make necessary updates in the client system to ensure that the order has been rejected by the provider.

Provider action

Be prepared to receive an answer to the request.

```
<msg msgType="2005" msgName="Order Reject Request">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121809865469" unique="true"/>
  </referencesTo>
</msg>
```

4.2.3.4.6 MSG 2006 Order Reject Request Accepted

MSG 2006 is a response of MSG 2005. It is sent by the client to inform the provider that the request for rejection in MSG 2005 is accepted and will be carried out.

Client action

Make necessary updates in the client system to ensure that the order is rejected.

Provider action

Make necessary updates in the provider system to ensure that the order is rejected.

```
<msg msgType="2006" msgName="Order Reject Request accepted">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795236" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  </referencesTo>
</msg>
```

4.2.3.4.7 MSG 2007 Order Reject Request Reject

MSG 2007 is a response of MSG 2005. It is sent by the Client to inform the Provider that the requested rejection in MSG 2005 is rejected and the order shall be carried out by the Provider.

Client action

Make necessary updates in the client system to ensure that the order is not rejected and will be carried out by the

Provider.

Provider action

Make necessary updates in the provider system to ensure that the order will be carried out.

```
<msg msgType="2007" msgName="Order Reject Request reject">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795237" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  </referencesTo>
</msg>
```

4.2.3.4.8 MSG 2010 Order Cancellation Request

MSG 2010 is a request from the Client to cancel a previous sent and accepted MSG 2000. The Provider is expected to try to cancel the order as a consequence of this message.

Client action

Ensure that the order id of the order in the cancellation request is correct.

Provider action

Immediately process the cancellation request.

Inform the client that cancellation is carried out without consequence by sending MSG 2011.

```
<msg msgType="2010" msgName="Order Cancellation">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121409309255" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409312773" unique="true"/>
  </referencesTo>
</msg>
```

4.2.3.4.9 MSG 2011 Order cancellation accepted

MSG 2011 is a response of MSG 2010. It is sent by the Provider to inform the Client that the cancellation in MSG 2010 is accepted and will be carried out without further consequences.

Client action

Make necessary updates in the client system to ensure that the order is cancelled.

Provider action

Make necessary updates in the provider system to ensure that the order is cancelled and will not be carried out without further consequence.

```
<msg msgType="2011" msgName="Order Cancellation accepted">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121409309256" unique="true"/>
  <referencesTo>
    <idOrder src="holmedal_hdcom_0001:idOrder" id="3640266-0" unique="true"/>
    <idOrder src="trapeze_pass_0001:idOrder" id="11183742" unique="true"/>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121409309255" unique="true"/>
  </referencesTo>
</msg>
```

4.2.4 Session Order Flow

This flow is in accordance with the driverSession flow briefly described in 2.1.1.7 above.

4.3 Assignment of vehicle flows (Dispatch)

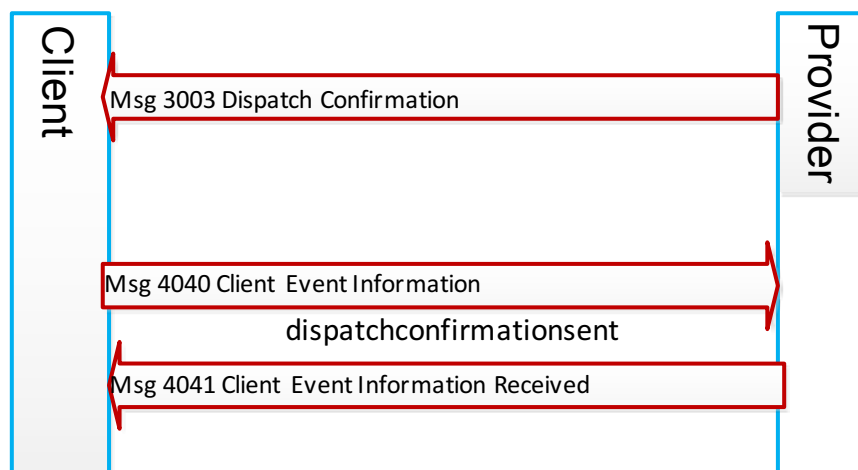
4.3.1 Normal Flow

The Provider has set a time-stamp on the order when to start dispatching. In most case the Client will always accept the chosen vehicle and no use of msg 3000, 3001 or 3002 is necessary. In the Node by Node Flow it is common that the use of 3000, 3001 and 3002 will be used.

When the Provider has got an accept from driver of a suitable vehicle the Provider will send msg 3003 to inform that the Driver in the the assigned vehicle has accepted the order.

Depending on setup the Client will now expect coordinates to include the vehicle in bulkLocations from next communication of such coordinate or will start sending 5020 to start receive 5021.

If the Client informs the passenger of the chosen vehicle and starts showing vehicle positions in the app of the Client, the Client will send a msg to vehicle 4040-event dispatchConfirmationSent The Provider system will immediately return with msg 4041 and it is assumed that the driver will be informed. See below for flow and examples.



Conditions for actions by Client

The Client may inform the passenger of the chosen vehicle and may also display the position of the vehicle at request from passenger or if the passenger uses the App of the Client. Such information and display will not take place until the Client is confident that there will not be a change in vehicle assignment.

The normal procedure is that no information to passenger will take place unless

- There are location information that shows that the vehicle has moved in direction to pick-up location (distance has diminished)
- Time is close to pick-up

4.3.2 Irregularity Flow 1: The suggested driver may not accept the order.

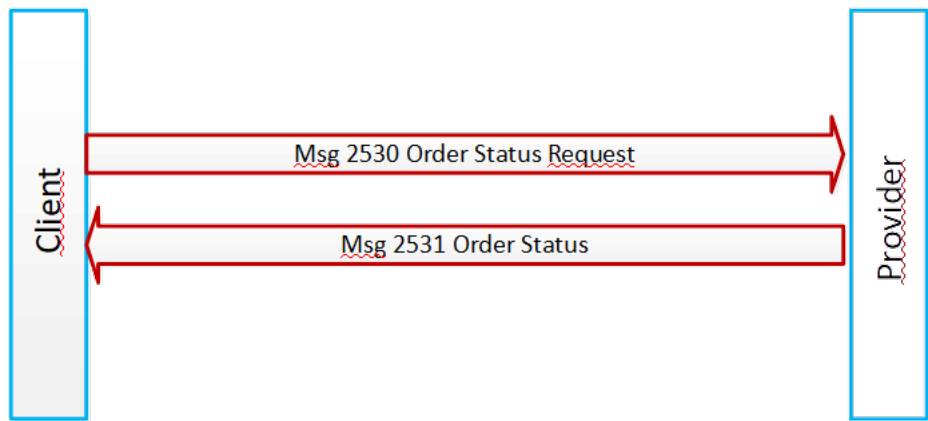
The provider will then send a new msg 3003 and the procedure in picture above will be repeated.

4.3.3 Irregularity Flow 2: Late Dispatching from Provider.

The Client has an expected time for arrival of msg 3000. The time has been passed and no 3000 msg has arrived

The client will send an Order Status Request (msg 2530) and expects an Order status in response (msg 2531). See below for flow and examples.

For further actions depending on answer see below under estimated time for vehicle at node.



4.4 Monitoring Flows (Traffic Control)

4.4.1 Prerequisites general

For each use case in this section it is understood that the Client has sent an order (2000 msg) and the Provider has accepted the order (2001). It is also understood that any negotiating on choice of vehicle is out of scope and not a part of the different situations.

As the Order Flow shows the Client will ask for bulkLocations, thus no communication using 5020, 5021 are shown below. However it is an ordinary Flow of the Client system to use 5020 in different situation and will expect 5021 as response.

The Monitoring Flows will vary depending on the type of the pickup point. The different flows will be added based on comparison with the most common, pick up at an ordinary (simple) address.

The Monitoring Flows will also vary depending on if it is a preboking or an ASAP-order. As with the different pickup points the Flows will majorly be compared with the standard of the prebooked passenger.

4.4.2 Monitor Flow Pick up at a simple address for a preebooked passenger (Standard)

The pick-up takes place at a simple address. The monitoring will be divided into three different stages

- Estimation of vehicle at node
- Vehicle at node
- Passenger in vehicle

4.4.2.1 Estimation of time for vehicle at node (Stage 1)

4.4.2.1.1 Normal Flow

When the system of the Client has decided that it has a trusted assignment of vehicle it will calculate an estimated time of vehicle at node. In case this estimation is not later than scheduled pick-up time, No action will be taken and estimated time given to passenger will be the same as pick-up time. The normal situation is thus no Flow.

There is always the possibility for either the Provider system or the driver to estimate and send information about estimated time for vehicle at node. Message 4010 should be used for that holding an Estimated Time.

4.4.2.1.2 Irregularity Flow 1: The Client has due to vehicle position estimated a late Passenger pickup

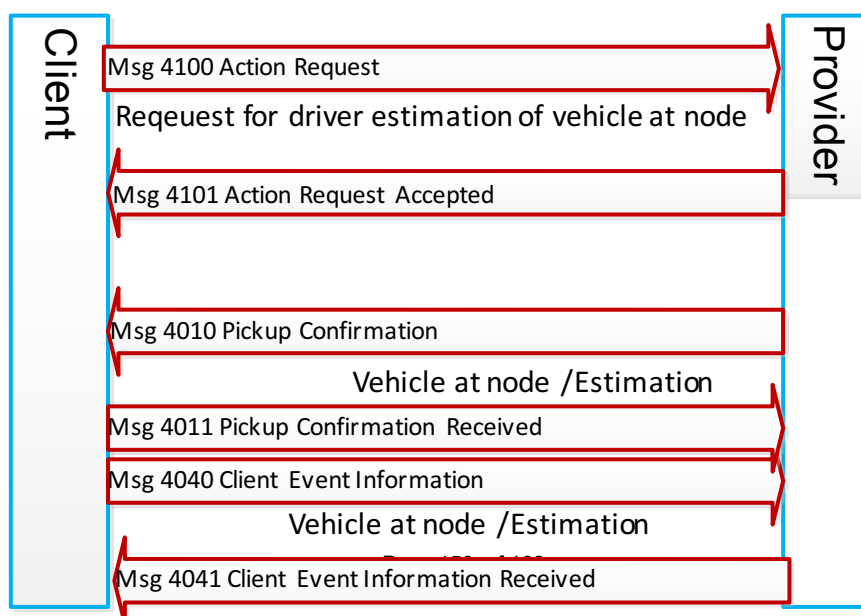
The Client will inform the passenger that there may be a late arrival of vehicle. If the estimated remaining time until vehicle is expected at node is greater than x minutes it will also request of an estimation from Driver.

A process starting with Client sending a msg 4100 idAction=50101 (Request estimated time vehicle at Node) will take place. The request will also hold the estimated time of the Client. The request is accepted by Provider (msg 4101) and is sent to driver.

The driver will calculate the time and inform Provider system. The Provider system will send 4010 "vehicle at node" with the estimated time. The Provider system may make the calculation without involving driver.

When Client has got the estimation it will inform the passenger but will also send a msg 4040 "vehicle at node" having the estimated time. It is assumed that the Provider will inform driver.

If there are no driver estimation within reasonable time the Client will anyhow inform passenger of the estimation from the Client system and send a msg 4040 "vehicle at node" having the estimated time of the Client. It is assumed that the Provider will inform driver. See below for flow and examples.



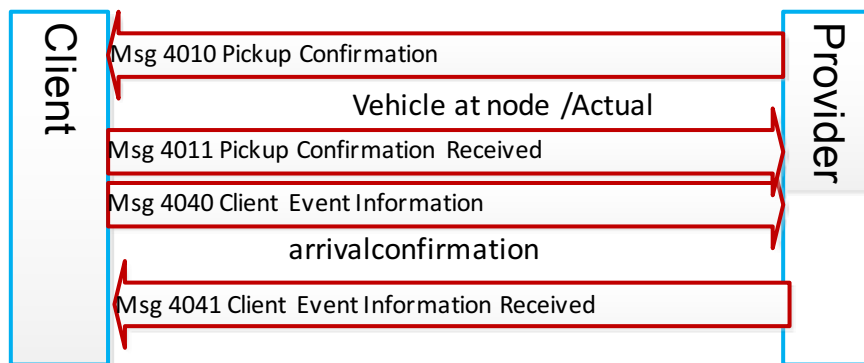
4.4.2.2 Vehicle at node (Stage 2)

4.4.2.2.1 Normal Flow

The vehicle will arrive at node. The driver will inform the Provider that the vehicle is at node. The Provider will immediately send msg 4010 “vehicle at node” with an actual time. The Client will accept by sending a msg 4011.

The Client will also inform the passenger and send a msg 4040 ArrivalConfirmation when that has happened.

See below for flow and examples.

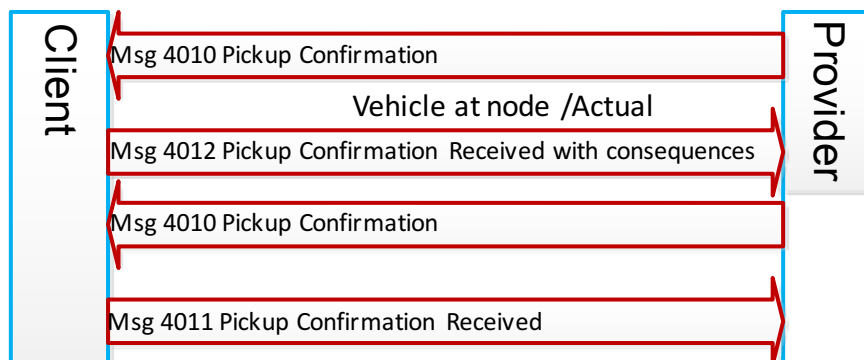


4.4.2.2.2 Irregularity Flow 1: Client decides not to accept vehicle at node

The reason for doing this is either

- Vehicle is not standing still or
- Vehicle is too far from the coordinates of the pick-up address

Client is aware of that for both measurements there may be some measurement errors and uncertainties. The Client will hold some tolerances for both of the measurements, In some circumstances the errors may be so large that they are outside these tolerances, Therefore the Client will accept a second 4010 msg. See below for flow and examples.



However if Client still believe that there is some error with the msg 4010, the 4011 may be followed by a 5000 with a request to confirm reading. The msg may for instance state that some later investigation will occur or may ask for a driver comment expected to be sent in a 5010 as soon as possible.

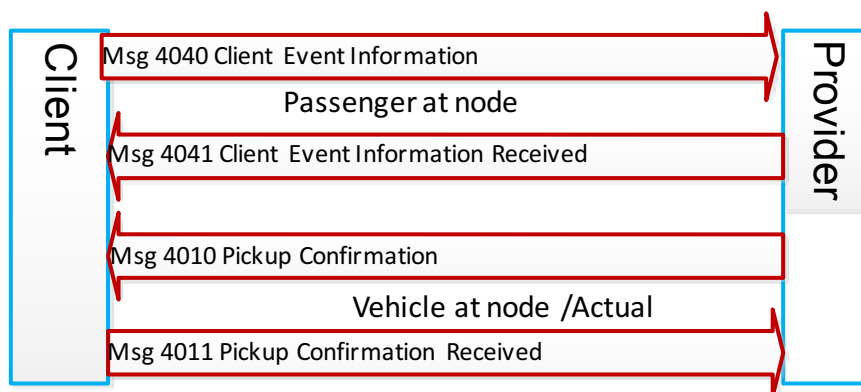
As with the Normal Flow the Client will inform Passenger and also send a Msg 5000 to Vehicle to confirm that Passenger is informed.

4.4.2.2.3 Irregularity Flow 2: The passenger has notified the Client that he/she is at node

The Client will send a msg 4040 – “passenger at node” containing actual time and potentially the coordinates of the passenger. The client has a procedure to ask the client for reveal of location when the passenger has reported at node. If the client determines that the coordinates is reliable they will be presented in the 4040 msg.

The Provider will accept by sending a 4041 and at the same time accept the coordinates as a precision of pick-up address. In case such coordinates are presented the provider shall act to inform the driver. If coordinates exists they will be held prior to text address that will else be of first priority for decision of pick-up location.

In case Client sends a 4040 “passenger at node” after the Provider has sent 4010 “vehicle at node” it is also a request that the driver may move the vehicle and resend 4010 vehicle at node. Even if the vehicle is not moved there should be a new 4010 sent.



4.4.2.2.4 Irregularity Flow 3: The driver notifies that vehicle is not at the correct pick-up address.

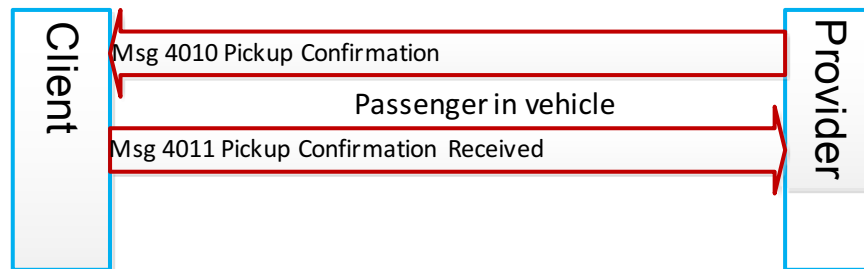
The driver will move the vehicle and send a new 4010. If the new 4010 is sent after an accept 4011 the Client will act as if the 4010 is a new 4010 msg and the same procedures as above is applicable. If it is sent after a 4012 the answer will be a 4011 maybe followed by a 5000 as described above.

The Flow here is a repeat of Normal Flow above.

4.4.2.3 Passenger in vehicle (Stage 3)

4.4.2.3.1 Normal Flow

The driver and the passenger has identified each other and msg 4010 – Passenger in vehicle is sent. The client always responds with a 4011.



4.4.2.3.2 Irregularity Flow 1: Client does not accept the identification of the passenger

If the Client requires a validation of passenger identification there might be a situation where the Client does not accept the actual identification.

When this happens the Client sends a 4012 (received with consequences) instead of 4011.

This is the only reason the Client may send a 4012. The 4012 is also a request for the driver to send a new 4010 containing the correct id.

In case no correct id is available the driver shall hold until Client has contacted passenger. In case the Client decides not to interrupt the pick-up process a 4011 will be sent. Whatever situation occurs after a 4012 the client will send a 5000 to inform driver.

4.4.2.3.3 Irregularity Flow 2: Client has not got a Vehicle at node prior to Passenger in vehicle

The Client is aware of that the reporting of node events is dependant of driver actions in the Vehicle. Driver may do some initial error and want to change. Here we will comment some potential errors and what the actions may be.

The Normal Flow will always be that there is a Vehicle at node before Passenger in Vehicle. However if the Passenger in Vehicle is sent first it will be accepted as a simultaneous Vehicle at node.

In case that the driver has sent the wrong 4010 msg the driver may alter the actions and send new 4010 msgs.

In case a 4010 Vehicle at node is sent after a Passenger in Vehicle it is assumed that there is a manual error and the previously accepted Passenger in Vehicle will be discarded.

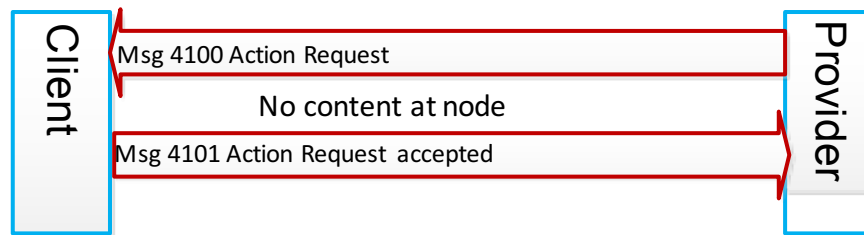
Last accepted Msg 4010 of both types will be used as the final results.

Observe that the vehicle must not leave the node unless the Client has accepted Passenger in Vehicle. The only reason may be that communication is broken with vehicle. In case the Provider is aware of such a communication failure the provider shall send 4031 stating that communication with vehicle is broken. If the breakdown last for more than 5 minutes the msg 4031 shall be repeated.

4.4.2.3.4 Irregularity Flow 3: The driver signals that no content is at node

This situation may occur either if the Client has sent Passenger at node or the actual time is close to estimated or scheduled pick-up time and the driver cannot identify the passenger.

The alarm takes place with a 4100 ActionRequest holding SUTI idAction=50102, meaning No content at node. The Client acknowledges with 4101.



The Client has now a pamphlet of actions depending on situation. The first action is always to inform passenger of the alarm either by App or SMS.

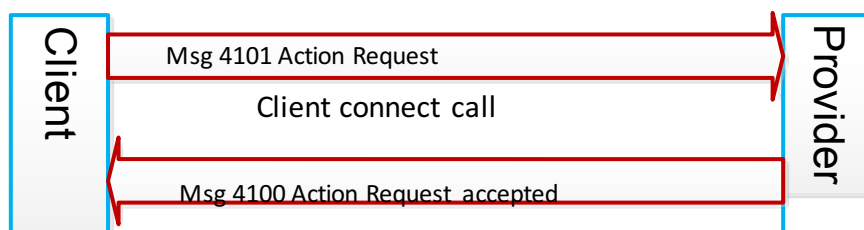
Situations and Flows when Client have no system (App) contact with passenger

If the Client has no possibility to interact with the passenger it will take the action to connect the passenger and the driver. There are several different methods to do so.

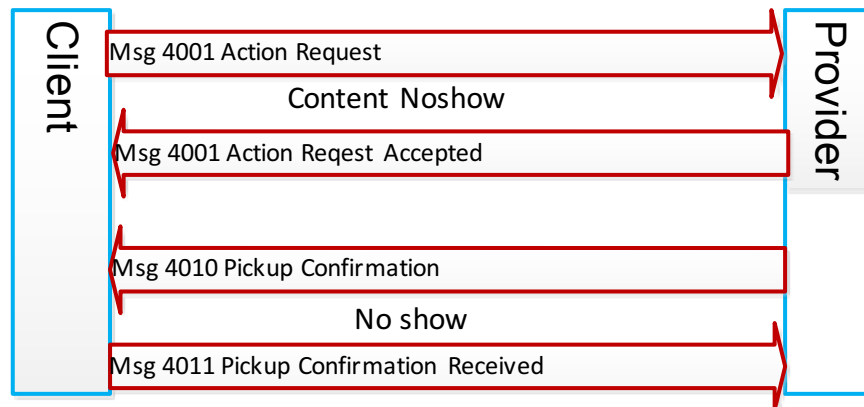
- No calling from either driver or passenger. The Client will contact the passenger and the driver. The Client will connect the two.(idAction=50111)
- No calling from either driver or passenger. The Provider will contact the passenger and the driver. The Provider will connect the two. .(idAction=50112)
- The passenger will make a direct call to the driver. The telephonnumber should be present in msg 3003. .(idAction=50113)
- The driver will make a direct call to the passenger. The Client will inform the driver of the telphonenumber to the passenger. .(idAction=50114)

In the cases of driver making a call the Provider may have implemented a service for an operator to call instead of the driver. For each of this different cases there is a sequence of msgs sent. See an example flow where Client organizes a call.

Generally the procedure to decide for the action to connect passenger and driver is using the msg 4100 using different idAction. Both the Provider and the Client can initiate such a process. The msg will hold the idAction describing used method. The other part can accept or deny using 4101 or 4102.



If however all actions taken does not result in a connection between driver and passenger the driver has no other possibility to decide that there is a no-show situation. The driver informs the Provider and Provider/driver will send 4010 – No show.The client will send 4011 to acknowledge. I is also a possibility that Client may have stimulate the No-show by sending 4001 as shown below.



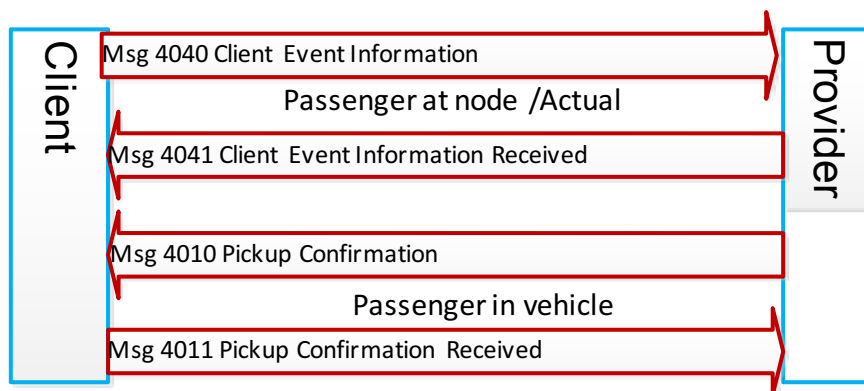
If 4010 – No show is sent while passenger are in contact with Client in an App such message may be answered with 4012 – Pickup confirmation received with consequences.

Pls note that No-show for a node is automatically a No-show for all nodes for the same content.

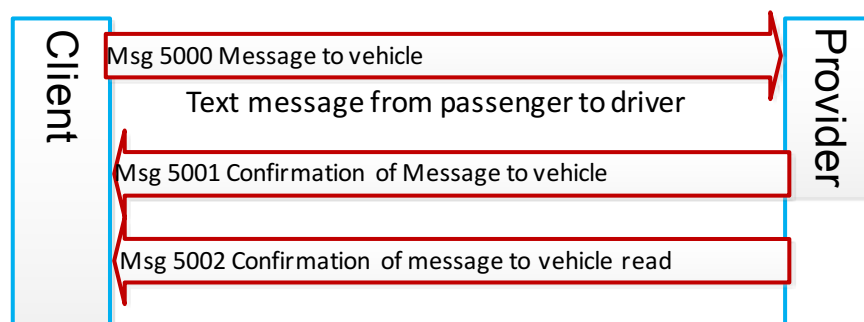
Situations and Flows when Client system are in contact with passenger

If the Client has the possibility to interact with the passenger there may be several options available.

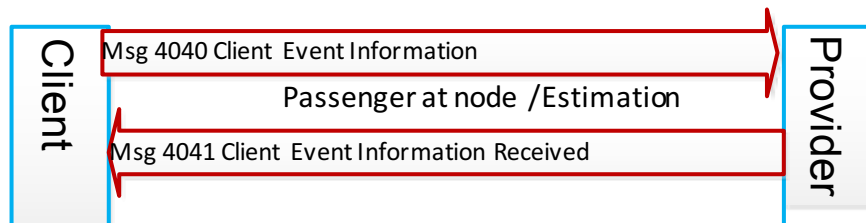
- The passenger may be at node. In this case the Client will send 4040 Passenger at Node, returning back the responsibility to the Provider. Eventually the driver and the passenger will locate each other and 4010 Passenger in vehicle is sent



- If the Client have had the possibility to ask the passenger for a close-by sign for the location of the passenger this description of the sign may also be forwarded to driver in a 5000 msg.



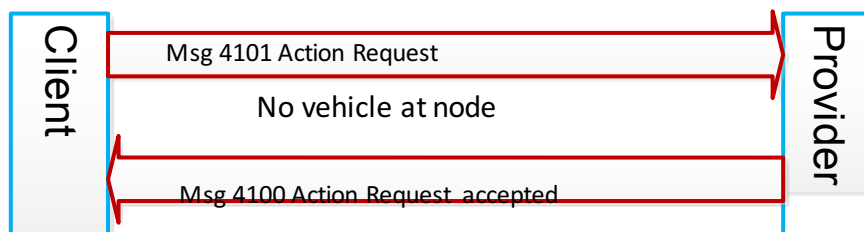
- The passenger may be late, and the driver is requested to wait for further information. The client will send 4040 with a new estimated time. The message is also a request for the driver to wait.



- The passenger may provide a general message. Will be forwarded as a 5000. See above
- The passenger will ask for driver contact. 5000 will be sent. See above to establish call.

4.4.2.3.5 **Irregularity 4: The passenger signals that no vehicle is at address**

Before this message it is assumed that the passenger is aware of any late vehicle as described in estimation of Vehicle at node. Thus the signal comes when the passenger has been notified that the vehicle is at node. The Client will inform the Provider with 4040 Passenger at node bringing the responsibility over to the Provider.



The normal reaction on Provider side is to inform driver. The most reasonable action from Provider is an action to establish contact between driver and passenger described in irregularity 3 above.

4.4.2.3.6 **Irregularity 5: The passenger or client decides to interrupt node**

This is not normally handled as a part of the monitoring process. See for sections regarding update of order. A method may be cancel node. There is a possibility to use No-show as is described in Contnt Noshow above.

4.5 Technical Flows

4.5.1 MSG 7000 Keep alive

MSG 7000 is sent either by the client or by the provider as a check that the other system is alive.

```
<msg msgType="7000" msgName="Keep alive">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795234" unique="true"/>
</msg>
```

4.5.2 MSG 7001 Keep alive confirmation

MSG 7001 is sent as a response of a MSG 7000 received from the other side.

```
<msg msgType="7001" msgName="Keep alive confirmation">
  <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121414795235" unique="true"/>
  <referencesTo>
    <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795234" unique="true"/>
  </referencesTo>
</msg>
```

4.5.3 MSG 7030 Syntax Error

MSG 7030 is sent by the system that has received a message that don't exists, don't have a message id.

```
<msg msgType="7030" msgName="Syntax error">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795235" unique="true"/>
  <referencesTo>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121414795234" unique="true"/>
  </referencesTo>
  <!--The error found during the processing of message, can be sent by Client or the Provider -->
  <manualDescriptionMsg manualText="Error message text" sendtoInvoice="false" sendtoOperator="false"
sendtoVehicle="false" vehicleConfirmation="false"/>
</msg>
```

4.5.4 MSG 7031 Not operation

MSG 7031 is sent as a response of a message that can't be responded to or when the system is down for maintenance.

```
<msg msgType="7031" msgName="Not operational">
  <idMsg src="trapeze_pass_0001:idMsg" id="2015121414795235" unique="true"/>
  <referencesTo>
    <idMsg src="holmedal_hdcom_0001:idMsg" id="2015121414795234" unique="true"/>
  </referencesTo>
  <manualDescriptionMsg manualText="Error message text" sendtoInvoice="false" sendtoOperator="false"
sendtoVehicle="false" vehicleConfirmation="false"/>
</msg>
```


5. Alterations

5.1 Rev 1 2011-06-23

Chapter 3 Msg 6500 and associated messages added.

5.2 Rev 2 2012-03-02

Chapter 2 Use Cases added.

Earlier chapter 2 and 3 moved to chapter 3 Specific Use Cases in detail

5.3 Rev 3 2012-03-16

Section 2.1 Order flow altered. Headers General, Recommended solution, Flow chart and Use of referencesTo has been added.

Order irregularities Client and order irregularities Provider has been altered to Order events Client and Order events Provider.

Sections for Best Practice have been added. These sections have been moved from document SUTI_2012_Messages.doc.

5.4 Rev 4 Version 2013 2012-10-30

Section 2.1.6 to 2.1.9 added.

Section 2.2.4 added.

Section 3.8 extended.

Section 3.9 and 3.10 added.

5.5 Rev 5 Version 2013 2013-03-04

Flow charts added in section 2.

5.6 Rev 6 Version 2013.1 2013-05-29

Chapter 2.3.2 Order alterations has been updated.

5.7 Rev 7 Version 2014.1 2015-01-09

Chapter 2.5 Oppgør added.

5.8 Rev 8 Version 2015-2015-04-13

Alterations moved to end of document.

Chapter 1, Use Cases general Description and Use, added.

Section 5.1 Flagstops moved to 2.3.

Section 5.2 Delivery note and associated messages moved to 2.4

Section 2.5 Accounting added.

5.9 Rev 9 Version 2015. 2015-09-08

Document ID_Structures has been inserted in Use Cases.

Document LinksToISOStandards has been inserted in Use Cases.

Document ResourceDescription has been inserted in Use Cases.

Description of EventTypes has been added to Use Cases.
Chapter 3 has been altered.

5.10 Rev 10 Version 2016. 2016-02-26

Chapter 2.1.1.12 Order Forwarded has been added.
Chapter 3.2.8.2 has been altered. Reference to idOrderForward has been added.
Chapter 2.5 has been altered. Message 8182 has been added.

5.11 Rev 11 Version 2017. 2017-05-15

Chapter 2.1.1.4 Corrected in flow.
Chapter 2.1.1.6 Altered to order by order and added.
Chapter 2.1.1.7 Altered to driverSession and added
Chapter 2.1.4.2 Node events replaced by new information reflecting change in SUTI 2017
Chapter 2.1.5 Special processes for Order by order and driverSession added.

5.12 Rev 12 Version 2018. 2017-10-02

Chapter 3.1.4 Content added.
Chapter 3.1.5 Addresses added.
Chapter 3.1.6 Economy added.

5.13 Rev 13 Version 2018 2018-01-16

Chapter 3.2.9 addressNode has been moved into chapter 3.1.5 Addresses.
Chapter 3.2.10 addressNode has been moved into chapter 3.1.6 Geographical Location.
New chapter 3.2.9 Event types

5.14 Rev 14 Version 2019

Chapter 1 addition of some general text., e.g. descriptions of some basic terms
Chapter 4 (old Technical considerations) has been moved to new chapter 1
Chapter 4 /new) holds Examples of Self Declaration Flows
Chapter 3.2.10 0 – 3.2.13 added (idPayment/Ekinfo, locationList, Ratings, Order status)
Chapter 3.2.9 has new eventtypes 1718, 1719, 1720

5.15 Rev 15 Version 2021

Chapter 1.4 Techniques used for describing SUTI (XML and JSON) is new (earlier chapters moved)
Chapter 1.5 Communication alternatives for SUTI is enlarged introducing a new recommendation and specification for SUTI Communication using HTTP Post on each side.