

# SUTI Use Cases



## **TABLE of CONTENT**

<b>1.</b>	<b><i>Alterations</i></b>	<b>4</b>
1.1	Rev 1 2011-06-23	4
1.2	Rev 2 2012-03-02	4
1.3	Rev 3 2012-03-16	4
1.4	Rev 4 Version 2013 2012-10-30	4
1.5	Rev 5 Version 2013 2013-03-04	4
1.6	Rev 6 Version 2013 2013-03-04	4
<b>2.</b>	<b><i>Use cases</i></b>	<b>5</b>
2.1	<b>Order flow</b>	<b>5</b>
2.1.1	Basic flow	5
2.1.2	Typical flow	6
2.1.3	Extensive flow	8
2.1.4	Extensive flow with traffic control	10
2.1.5	Multiple client order combined into a shared provider order	12
2.1.6	Resource allocation followed by traffic control	14
2.1.7	Resource order followed by traffic control	14
2.1.8	Order Trading	14
2.1.9	Development of order	15
2.2	<b>Additional processes</b>	<b>16</b>
2.2.1	Order status	16
2.2.2	Alteration in dispatch with dispatch reject	17
2.2.3	Alteration in dispatch with new dispatch proposal	18
2.2.4	Resource allocation	19
2.2.4.1	The requested resource is not available	19
2.2.4.2	The requested resource is available	19
2.3	<b>Order events Client</b>	<b>20</b>
2.3.1	Order/node cancellation	20
2.3.1.1	Order/node cancelation accept	20
2.3.1.2	Order/node cancellation accepted with consequence	20
2.3.1.3	Order/node cancellation reject	20
2.3.2	Order alteration	20
2.3.3	Late order	21
2.3.3.1	The hard way orderReject (Client is responsible for the order)	21
2.3.3.2	The softer way pickup confirmation (Provider is responsible for the order)	21
2.4	<b>Order events Provider</b>	<b>22</b>
2.4.1	Order reject	22
2.4.2	Order reject request	22
2.4.3	Node events	23
2.4.3.1	Pickup confirmation using estimated time	23
2.4.3.2	Pickup confirmation using vehicleataddress	23
2.4.3.3	Pickup confirmation using passengerinvehicle	23
2.4.4	Request for additional or corrected node information	24
<b>3.</b>	<b><i>Use of information elements</i></b>	<b>25</b>
3.1	<b>Agreement</b>	<b>25</b>
3.1.1	General	25
3.1.2	Minimum requirements	25
3.2	<b>idMsg</b>	<b>25</b>
3.2.1	General	25

3.2.2	Minimum requirements	25
<b>3.3</b>	<b>idOrder</b>	<b>25</b>
3.3.1	General	25
3.3.2	Minimum requirements	25
<b>3.4</b>	<b>infoTimeStamp</b>	<b>25</b>
3.4.1	General	25
3.4.2	Minimum requirements	26
<b>3.5</b>	<b>msgTimeStamp</b>	<b>26</b>
3.5.1	General	26
3.5.2	Minimum requirements	26
<b>3.6</b>	<b>orgReceiver</b>	<b>26</b>
3.6.1	General	26
3.6.2	Minimum requirements	26
<b>3.7</b>	<b>orgSender</b>	<b>26</b>
3.7.1	General	26
3.7.2	Minimum requirements	26
<b>3.8</b>	<b>referencesTo</b>	<b>26</b>
3.8.1	General	26
3.8.2	Minimum requirements	26
<b>3.9</b>	<b>addressNode</b>	<b>27</b>
3.9.1	General	27
3.9.2	Minimum requirements	28
<b>3.10</b>	<b>geographicLocation</b>	<b>28</b>
3.10.1	General	28
3.10.2	Minimum requirements	28
<b>4.</b>	<b><i>Technical issues</i></b>	<b>28</b>
<b>4.1</b>	<b>Communication</b>	<b>28</b>
<b>5.</b>	<b><i>Specific use cases in detail</i></b>	<b>29</b>
<b>5.1</b>	<b>SUTI flagstops</b>	<b>29</b>
5.1.1	Flagstop forløb	29
5.1.2	Flagstops i SUTI	29
5.1.2.1	Authorization	29
5.1.2.2	Bookingen	29
<b>5.2</b>	<b>Delivery Note and associated messages</b>	<b>31</b>
5.2.1	How to request a Msg 6500 Delivery Note	31
5.2.1.1	Request for Msg 6500 by using process element in a Msg 2000	31
5.2.1.2	Request for Msg 6500 by using process element in a Msg 2901	31
5.2.1.3	Request for Msg 6500 by sending a Msg 6510	31
5.2.1.4	Responding to a Msg 6510 with a Msg 6511	31
5.2.2	Sending a Msg 6500 Delivery Note	32
5.2.3	Answering a Msg 6500	33
5.2.3.1	Msg 6501 deliveryNoteAccept	33
5.2.3.2	Msg 6502 deliveryNoteReject	33
5.2.3.3	Msg 6503 deliveryNoteWait	33

## **1. Alterations**

### **1.1 Rev 1 2011-06-23**

Chapter 3 Msg 6500 and associated messages added.

### **1.2 Rev 2 2012-03-02**

Chapter 2 Use Cases added.

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

### **1.3 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.

### **1.4 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.

### **1.5 Rev 5 Version 2013 2013-03-04**

Flow charts added in section 2.

### **1.6 Rev 6 Version 2013 2013-03-04**

Section 2.1.8 added.

## 2. Use cases

### 2.1 Order flow

#### 2.1.1 Basic flow

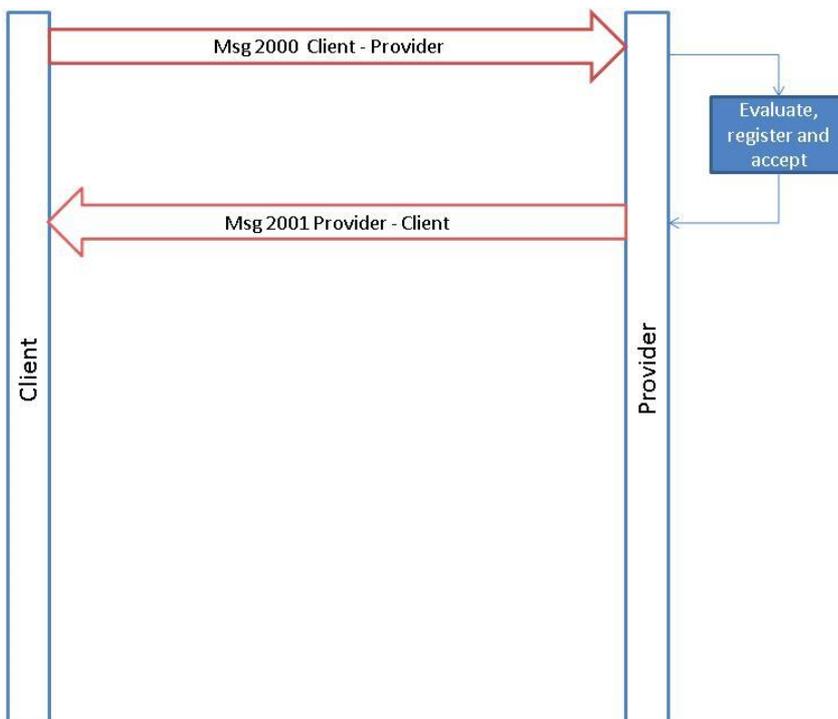
##### General

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



## 2.1.2 Typical flow

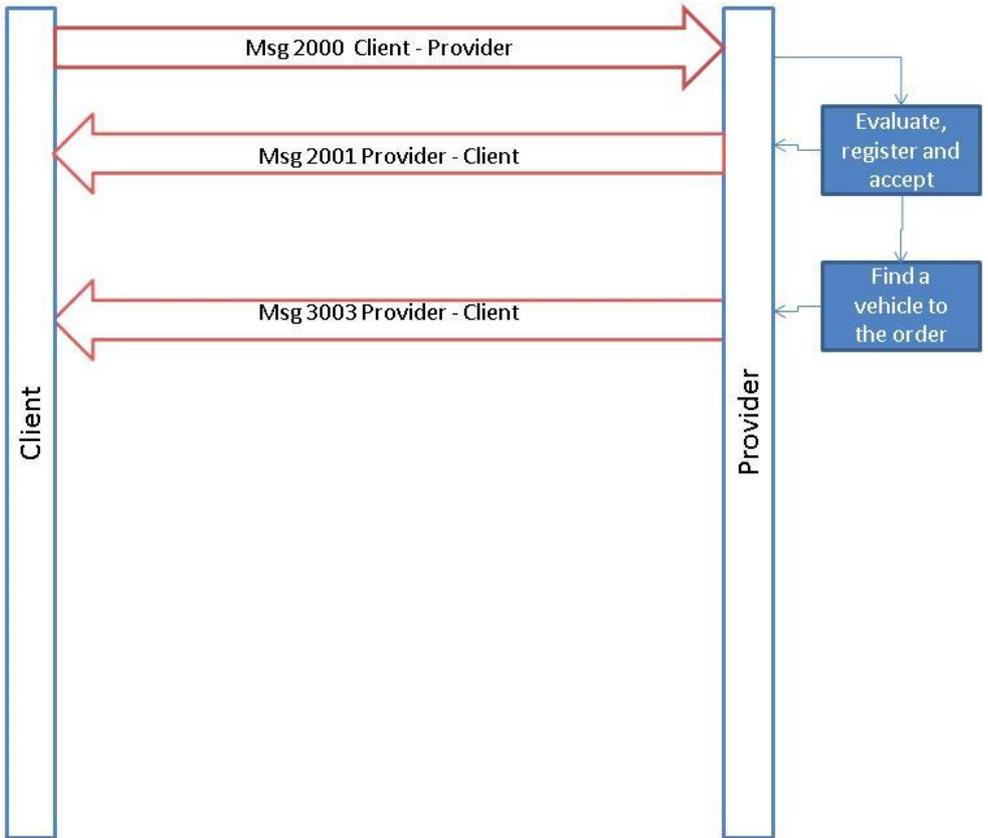
### General

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

	<b>Client</b>	<b>Provider</b>	<b>Comment</b>
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**



### 2.1.3 Extensive flow

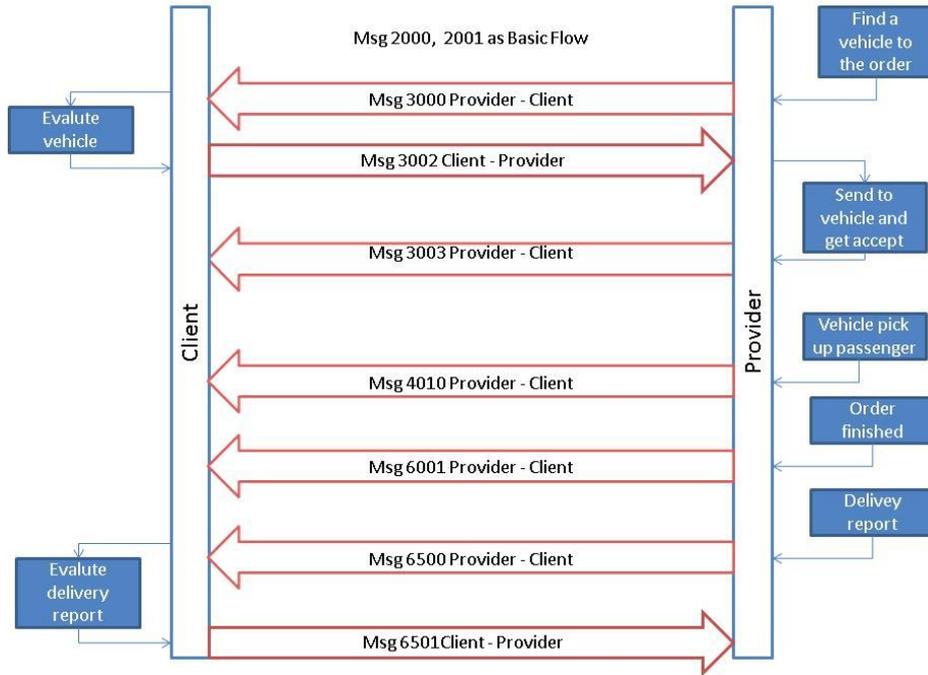
#### General

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

	<b>Client</b>	<b>Provider</b>	<b>Comment</b>
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**



## 2.1.4 Extensive flow with traffic control

### 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.

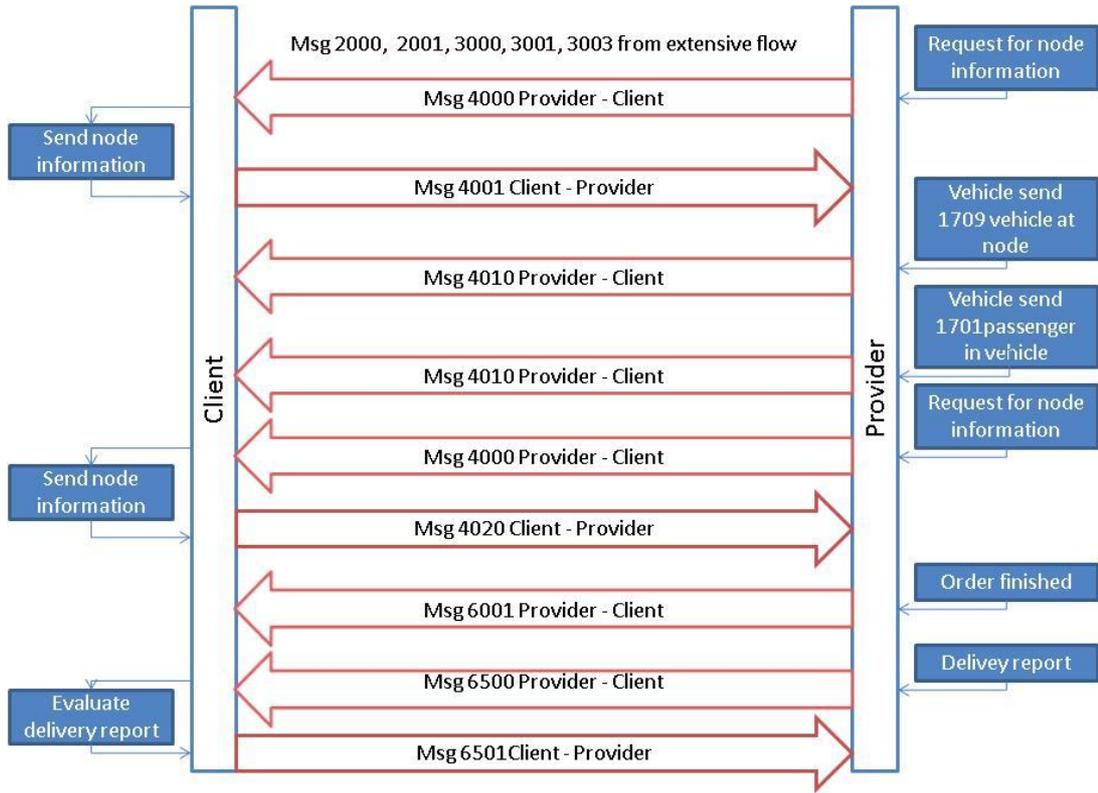
The client must always accept the suggested resources before starting the process to pick-up first node.

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 MSG4020		
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**



## 2.1.5 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.

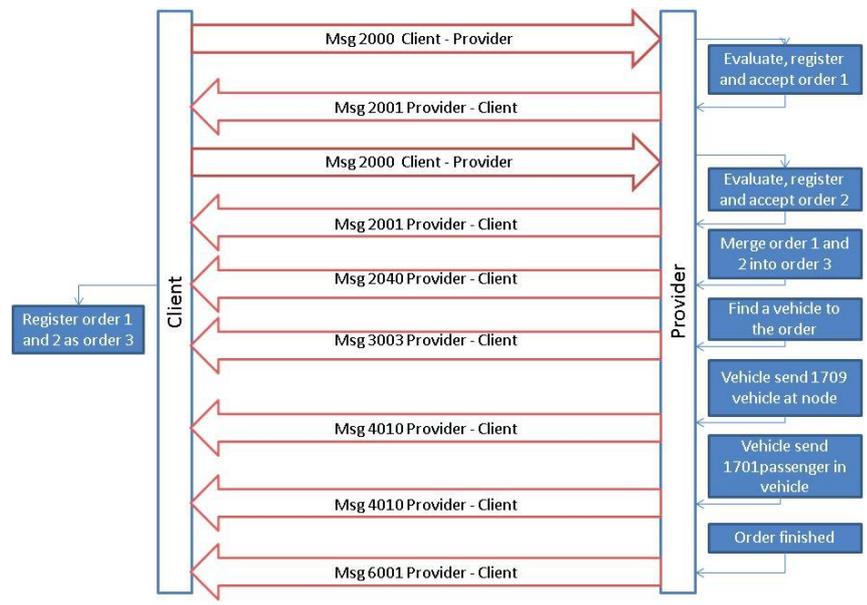
### Recommended Solution

	<b>Client</b>	<b>Provider</b>	<b>Comment</b>
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**



### 2.1.6 Resource allocation followed by traffic control

To be added. Dynamic traffic control for a hub area.

### 2.1.7 Resource order followed by traffic control

To be added. Dynamic traffic control for a hub area.

### 2.1.8 Order Trading

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.

	<b>Client</b>	<b>Provider 1</b>	Provider 2	<b>Comment</b>
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 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.

### 2.1.9 Development of order

To be added.

## 2.2 Additional processes

### 2.2.1 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:

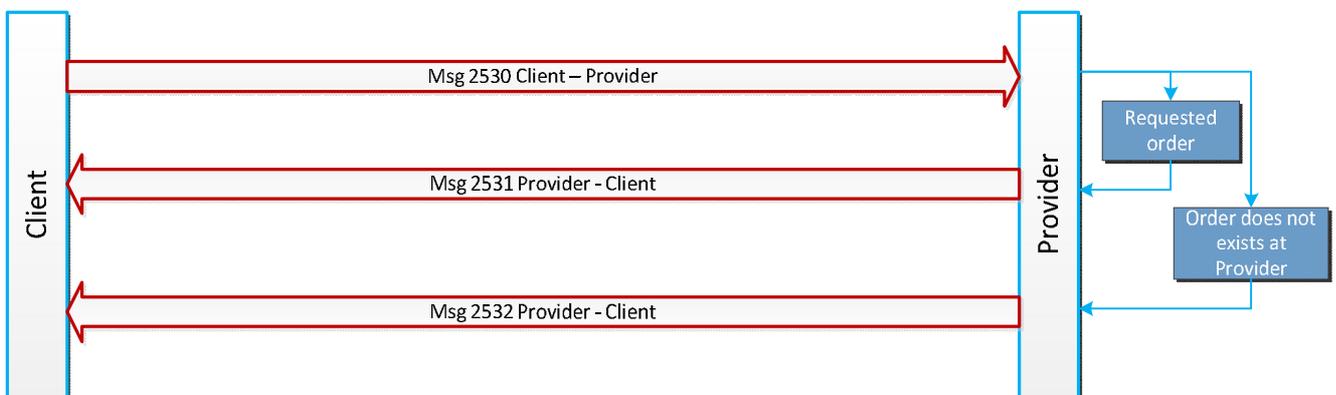
- the clients has not received appropriate messages in due time and need to secure its information status on an order
- the client has received a request from a traveller on a node and wants to secure its information status on an order
- the client has had a failure and wants to secure its information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



## 2.2.2 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.

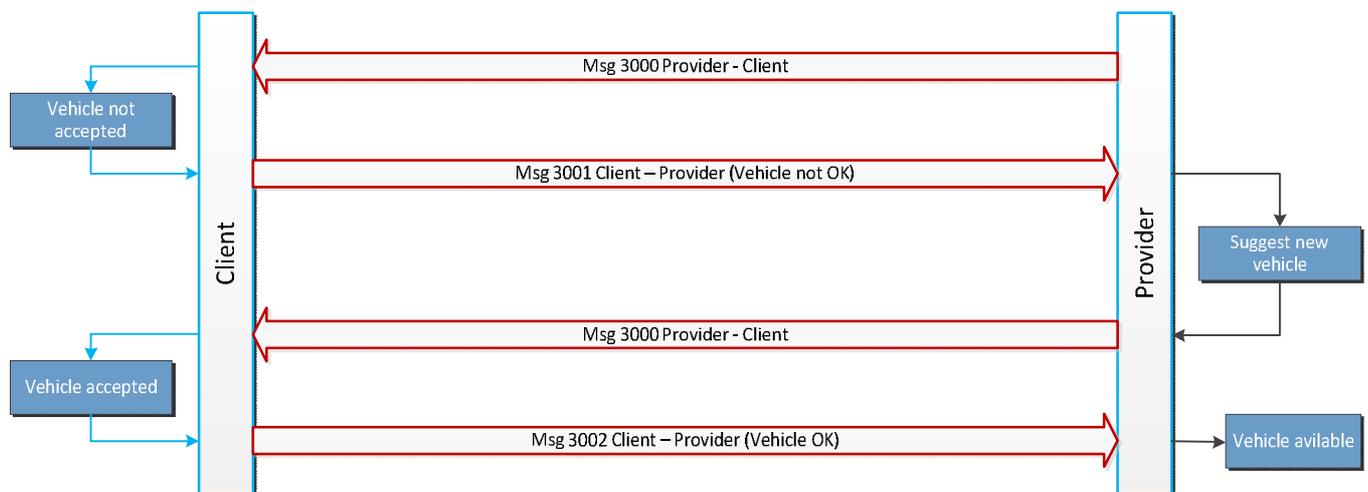
As seen below the provider initiate by sending its proposed resources. The client must answer immediately with either msg 3002, as described in the flow-chapter, or with 3001 in case the suggested resource is not accepted. If the answer is 3001, the provider should send a new alternative suggestion with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





### 2.2.4 Resource allocation

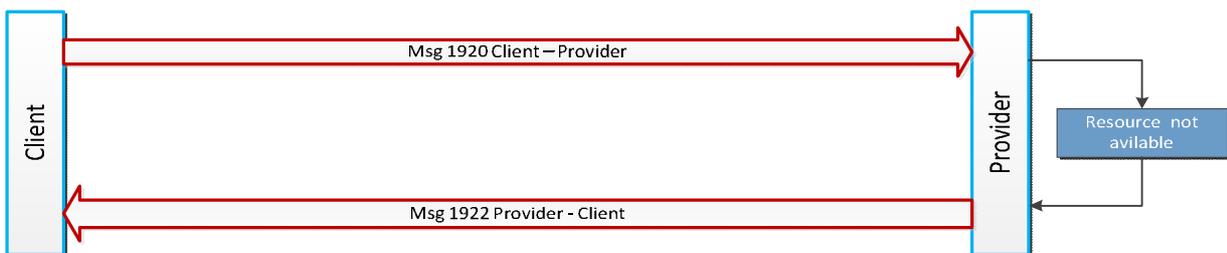
#### General

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.

#### 2.2.4.1 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



#### 2.2.4.2 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



## 2.3 Order events Client

### 2.3.1 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.

#### 2.3.1.1 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 cancellationConsequence 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.

#### 2.3.1.2 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 cancellationConsequence 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

#### 2.3.1.3 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.

### 2.3.2 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. 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 cancellation followed by a new order. 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 cancellation 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.*

### 2.3.3 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.

#### 2.3.3.1 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 asptime or send a new scheduledtime within an acceptable timeframe.

#### 2.3.3.2 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.

## 2.4 Order events Provider

### 2.4.1 Order reject

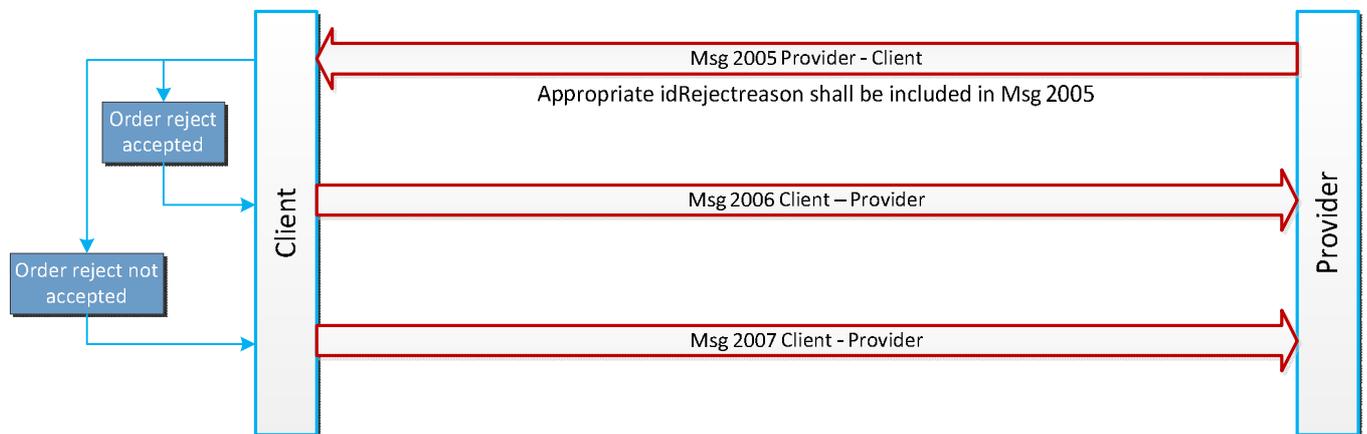
An order reject shall be sent without any delay and the reasons for an order reject shall be agreed upon in the agreement and no order accept shall be sent. If the reasons for reject aren't specified in the agreement every order reject shall include a reject reason.

### 2.4.2 Order reject request

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

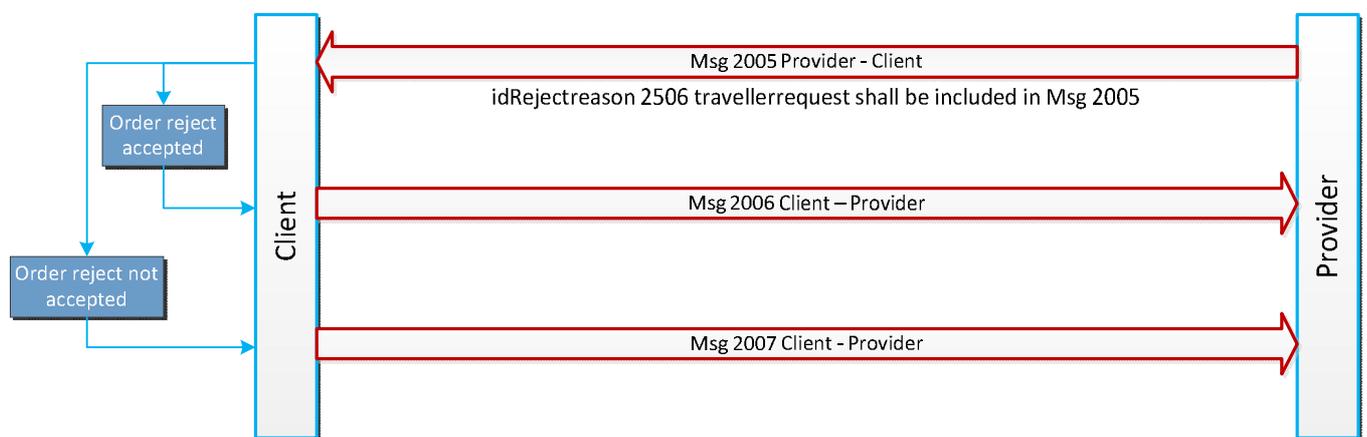
Sending the order reject request is done then the provider wishes that should not to any part be handled by the provider. The reason for this can be lack of capacity. If this is the case the message should contain idrejectreason 2504 nocapacity. The Client should without no delay answer with any of the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

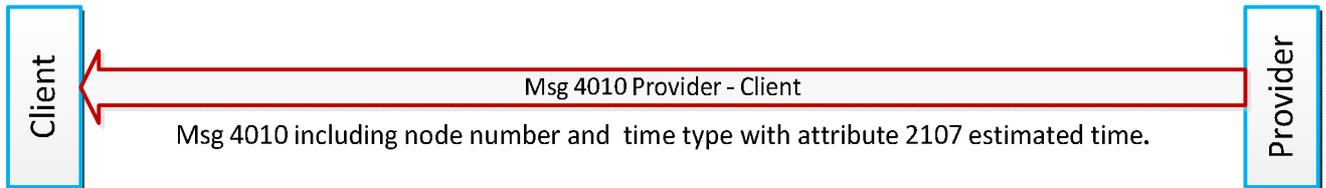


### 2.4.3 Node events

#### 2.4.3.1 Pickup confirmation using estimated time

	Client	Provider	Comment
n		Sends MSG4010	Shall include the node number and a timetype containing attribute 2107 estimatedtime.

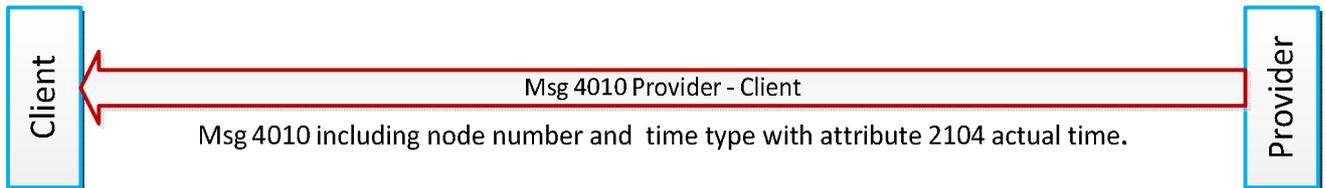
**Flow chart**



#### 2.4.3.2 Pickup confirmation using vehicleataddress

	Client	Provider	Comment
n		Sends MSG4010	Shall include the node number and a timetype containing attribute 2104 actualtime.

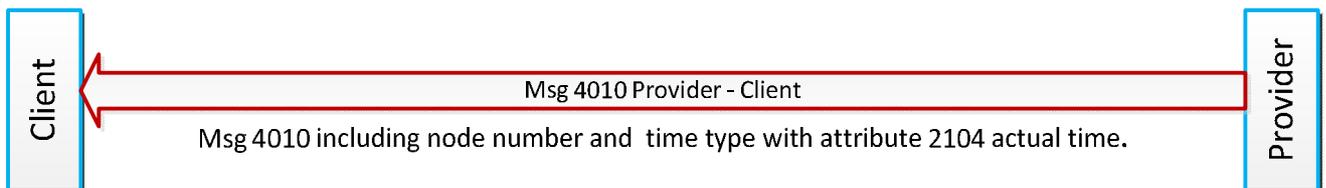
**Flow chart**



#### 2.4.3.3 Pickup confirmation using passengerinvehicle

	Client	Provider	Comment
n		Sends MSG4010	Shall include the node number and a timetype containing attribute 2104 actualtime.

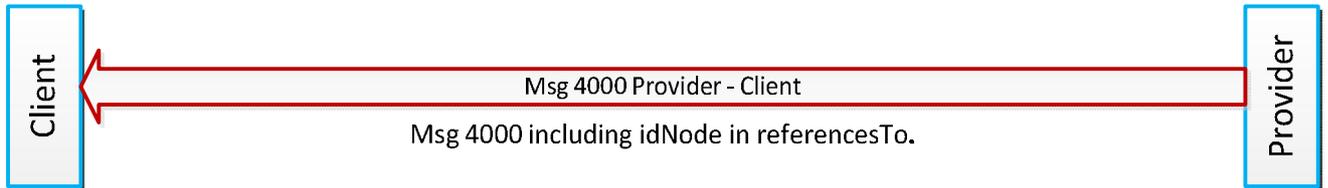
**Flow chart**



### 2.4.4 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



### 3. Use of information elements

Listed in alphabetical order.

#### 3.1 Agreement

##### 3.1.1 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](#)).

##### 3.1.2 Minimum requirements

#### 3.2 idMsg

##### 3.2.1 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 Minimum requirements

#### 3.3 idOrder

##### 3.3.1 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](#)).

##### 3.3.2 Minimum requirements

#### 3.4 infoTimeStamp

##### 3.4.1 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.

### 3.4.2 Minimum requirements

## 3.5 msgTimeStamp

### 3.5.1 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.

### 3.5.2 Minimum requirements

## 3.6 orgReceiver

### 3.6.1 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.

### 3.6.2 Minimum requirements

## 3.7 orgSender

### 3.7.1 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.

### 3.7.2 Minimum requirements

## 3.8 referencesTo

### 3.8.1 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 references To is to indicate that an order already has been sent to another Provider before it was sent to the Provider that just received it.

### 3.8.2 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, 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 <idOrder> 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)

### 3.9 addressNode

#### 3.9.1 General

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 addressinfonotsufficient

Below you find the structure of the address node including some good examples.

addressName	Name of company or institution eg "Flygtaxi"
street	Street name eg "Drottningholmsvägen"
streetNo	Street number eg "37"
streetNoLetter	Sub number of a streetnumber eg "A"
location	Name of the city or part of the city eg "Stadshagen"
community	The municipality eg "Stockholms kommun"
postalNo	Postalcode eg "112 42"
country	The country eg "Sverige"
mapPage	Page containing the address in a map eg "123A4"
manualDescriptionAddress	Used to provide a further clarification of an address eg "Våning 1"

### 3.9.2 Minimum requirements

At least one of either street in combination with streetNo and streetNoLetter or addressName shall be used.

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

## 3.10 geographicLocation

### 3.10.1 General

Iso standard: ISO 6709:2008

Resolution:

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).

### 3.10.2 Minimum requirements

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.

The format used shall be WGS-84 with decimal degrees.

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

## 4. Technical issues

### 4.1 Communication

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

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 this message is part of the set of SUTI messages that this system handle.

## 5. Specific use cases in detail

### 5.1 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.

#### 5.1.1 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.

#### 5.1.2 Flagstops i SUTI

##### 5.1.2.1 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.

##### 5.1.2.2 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:

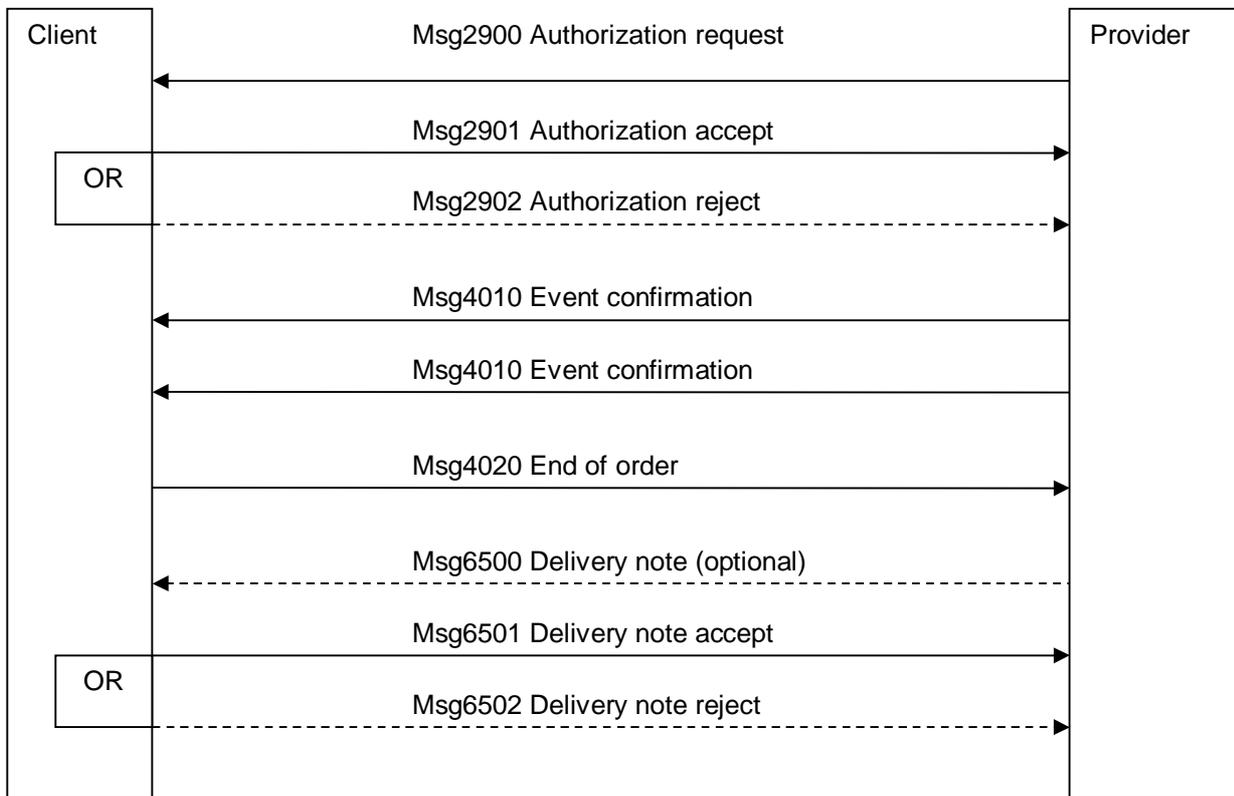
```
<process allowRouting="false" trafficControl="true" dispatchResponsible="client" report="false" preorderedVehicle="false" dispatch="true" manualDispatch="false" deliveryNote="true"></process>
```

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:



## 5.2 Delivery Note and associated messages

### 5.2.1 How to request a Msg 6500 Delivery Note

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

#### 5.2.1.1 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"/>
```

#### 5.2.1.2 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"/>
```

#### 5.2.1.3 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>
```

#### 5.2.1.4 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>
```

```

    </referencesTo>
  </msg>
</SUTI>

```

## 5.2.2 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"/>
            <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>

```

### 5.2.3 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.

#### 5.2.3.1 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>

```

#### 5.2.3.2 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>

```

#### 5.2.3.3 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>
```