eDG Transport Document WebServices description

17/12/2020

Document updated to chapter 3.2

Table of contents

1 PURPOSE OF THE DOCUMENT

This document describes all the web-services and the methods needed to be implemented for the deployment of the Architecture defined in the Memorandum of Understanding in order to allow usage of electronic of dangerous goods transport documents as defined in the ADR/ADN/RID regulation.

This architecture is issued from Telematic Working Group of the Joint meeting (WP15 of UNECE)

These webservices are closely linked with the eDG Transport Document UML model and its XML Schema.

2 PREREQUISITE

Use of webservices requires prior registration:

- ***of TP1 with each other***
- ***of TP2 with a TP1***

To register as a new TP1 it is mandatory to communicate the public key of the certificate of the new TP1 to the other TP1 already certified, members of the global network constituting the architecture for eDG transport document.

This public key is used by the TP1s to identify the sender of messages (query and answers to another TP1) and ensure recording and non-repudiation.

Registration with a TP1 and communication of the public key are also mandatory for any new TP2.

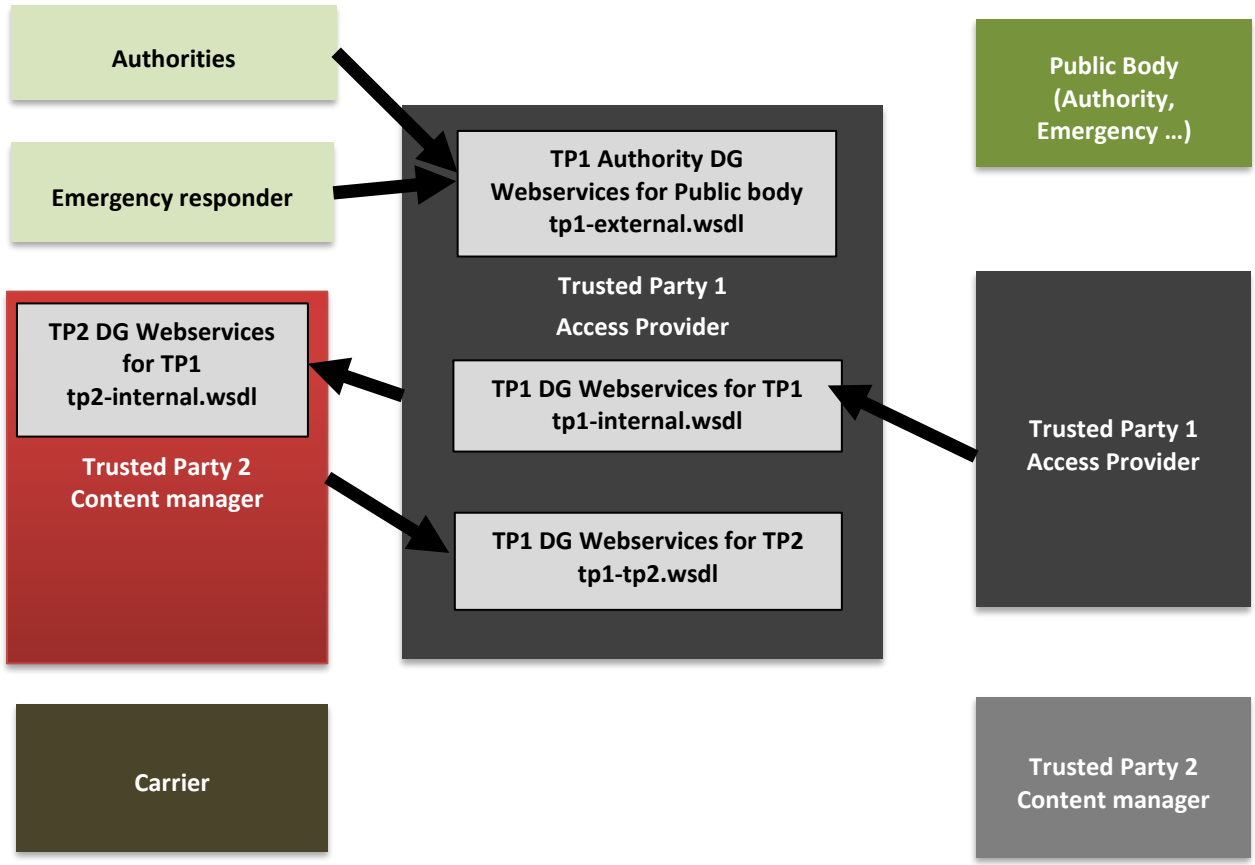
This principle applies to exchanges between TP1 and authorities as well.

When a public service registers to a TP1, this TP1 shall verify the type of this public service (emergency responders, enforcement bodies, traffic management bodies). Thus the TP1 may identify and communicate the nature of the query.

3 DIFFERENTS TYPE OF WEBSERVICES

Different webservices are the following:

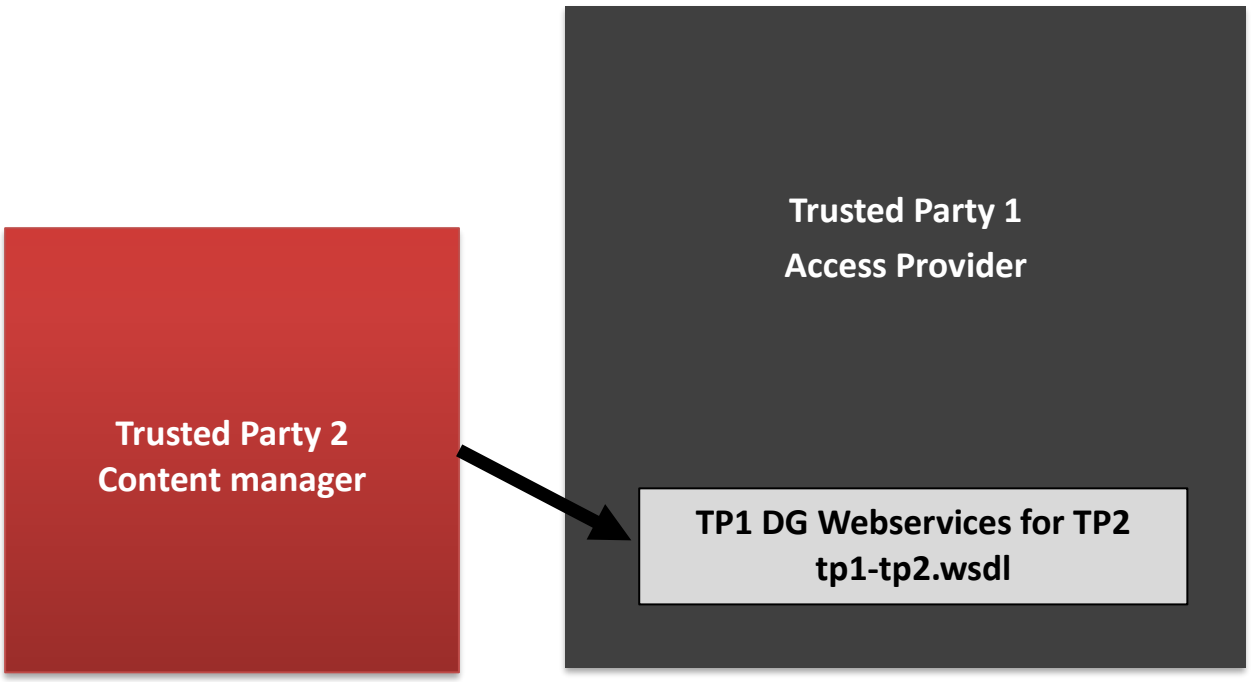
1. ***TP1 Internal Webservices: Webservices for communications from a TP1 to another TP1***
2. ***TP1 – TP2 Webservices: Webservices for communication from a TP2 to aTP1***
3. ***TP1 Authority Webservices: Webservices for communication between public bodies and TP1***
4. ***TP2 Internal Webservices: Webservices for communication from a TP1 to a TP2***



Note : The split in 4 different webservices allows better security by implementation of different access point depending the type of actors (TP1, TP2 or public bodies).

3.1 TP1 DG WEBSERVICES FOR TP2

Name of the webservice is « TP1-TP2Services » in the file tp1-tp2.wsdl



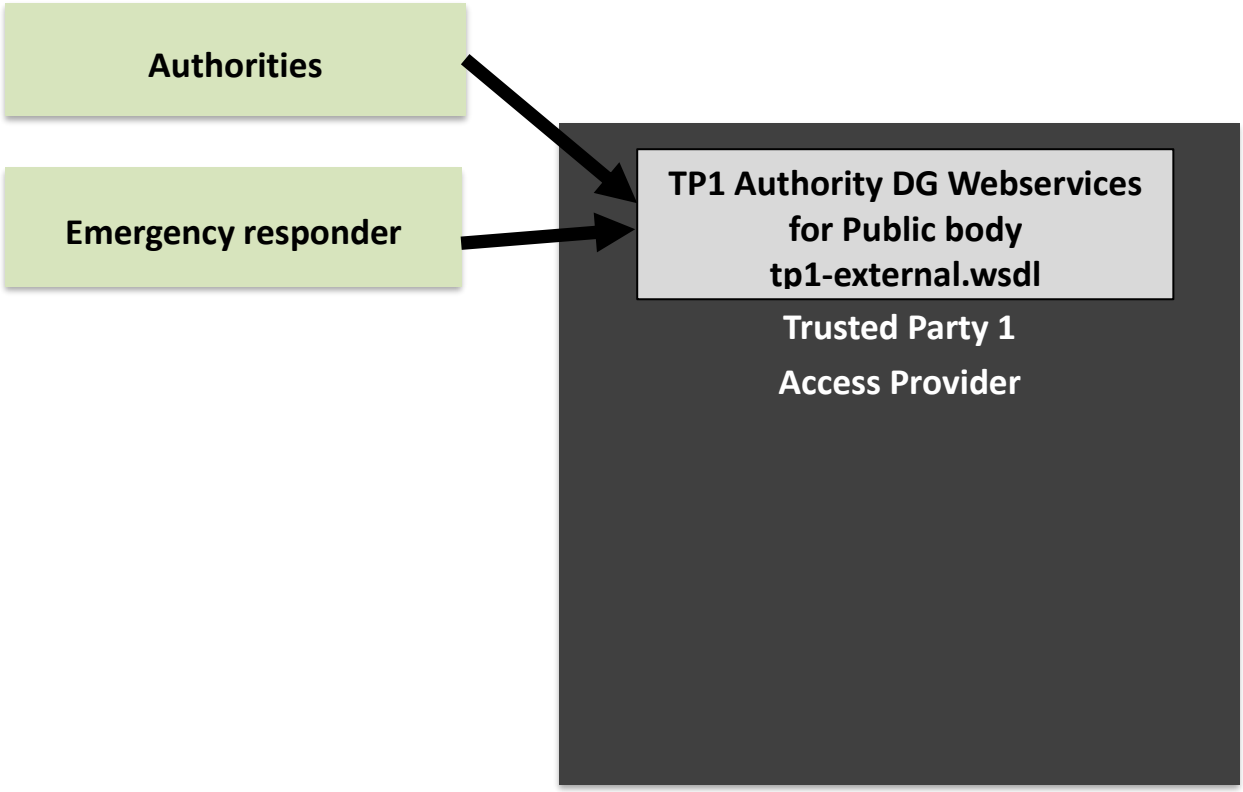
Availables methods are:

1. ***handleLoadedVehicle : Indicate if the journey (vehicle for road transport) has started or not.***
2. ***saveTrackingDataVehicle : Allows to send dynamic information to TP1 (eg positionning).***
3. ***sendTP2RegistrationRequest : Allows for a TP2 to ask for registration to a TP1.***

Note: When using the methods « handleLoadedVehicle » and « saveTrackingDataVehicle » the TP1 shall verify that the sender of the query is a TP2 by using the public key.

3.2 TP1 AUTHORITY DG WEBSERVICES FOR PUBLIC BODY

Name of Webservices is « TP1AuthorityServices » in the file tp1-external.wsdl



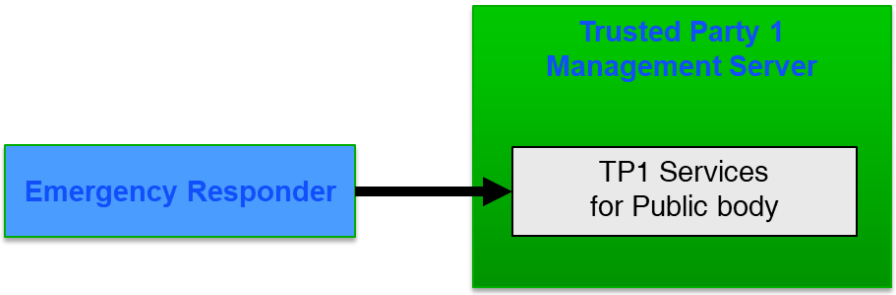
Available methods are:

1. **getDGTDocument: Allows an external system to ask transport documents for a vehicle**
2. ***getArchiveJourneyList: Allows an external system from authority to ask for archived data***
3. ***getTransportUnitByArea: Allows to get identification of vehicle in the vicinity of an area (for instance an accident area) Allows an external system from emergency responder to ask the list of transport units located inside circle.***
4. ***getDynamicInformation: Allows an external system from public body (road operator of traffic manager by instance) to ask dynamic information related to a transport unit and its goods***
5. ***sendPublicServiceRegistrationRequest: Allows to implement automatic connection of a public service***

Note: When using these methods the TP1 shall verify that the sender of the query is a public service by using the public key.

3.3 TP1 WEBSERVICES FOR OTHER TP1

Name of webservice is « TP1InternalWebServices ».



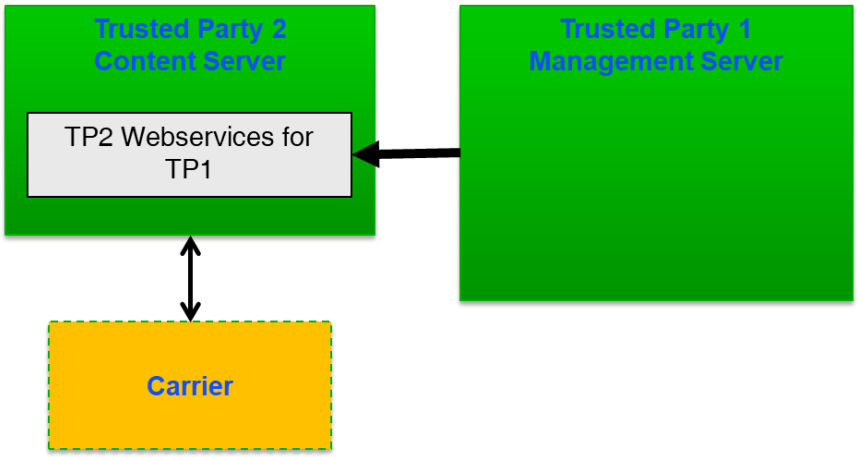
Available methods are:

6. ***getDGTDocument :Allows an external system to ask transport documents for a vehicle***
7. ***getTransportUnitByArea : Allows to get identification of vehicle in the vicinity of an area (for instance an accident area)***
8. ***getDynamicInformation : Allows to send dynamic information concerning a vehicle.***

Note: When using the methods « getDGTDocument », « getTransportUnitByArea» and «getDynamicInformation » the TP1 shall verify that the sender of the query is a public service of the type emergency responder by using the public key.

3.4 TP2 WEBSERVICES FORTP1

Name of webservice is « TP2InternalWebServices ».



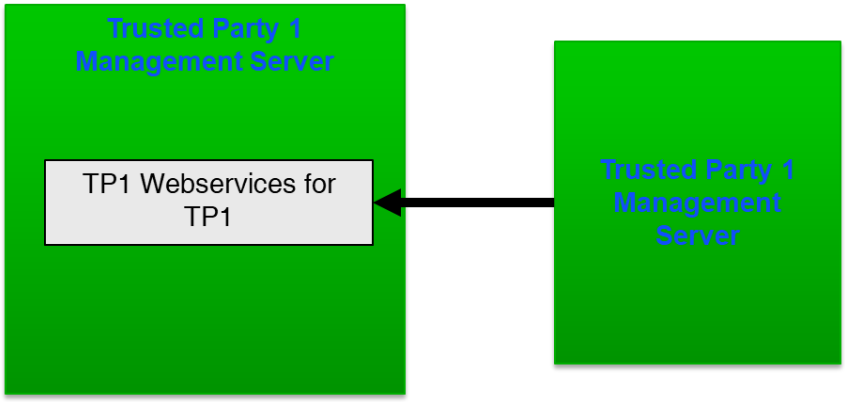
Available methods are:

9. ***getDGTDocument : Allows a TP1 to ask for the transport document for a vehicle communicating the reason for it***
10. ***getArchiveJourneyList : Allows the TP1 to get archived data***

Note: The TP2 shall verify that the sender of the request is a TP1 where he is registered by using the public key.

3.5 TP1 WEBSERVICES FOR TP1

Name of webservice is « TP1InternalWebServices ».



Available methods are:

11. ***getDGTDocument : Allows a TP1 to ask for the transport document for a vehicle communicating the reason for it.***
12. ***getArchiveJourneyList : Allows the TP1 to get archived data***

13. ***getTransportUnitByArea : Allows to get identification of vehicle in the vicinity of an area (for instance an accident area) and communicate the reason for request***
14. ***getDynamicInformation : Allows to send dynamic information concerning a vehicle.***

Note: The TP1 shall verify that the sender of the request is a TP1 by using the public key

4 TP1 INTERNAL SERVICES

4.1 MANDATORY METHODS

4.1.1 handleLoadedVehicle

This method indicates that a journey has begun or has ended.

boolean handleLoadedVehicle(eu.datex2.schema._2._2_0.DGCarryingVehicle dgCarryingVehicle, boolean activation) throws java.rmi.RemoteException

Entry parameters

eu.datex2.schema._2._2_0.DGCarryingVehicle dgCarryingVehicle: objet « DGCarryingVehicle » where « DGTransportUnit » is mandatory and indicates which Transport Unit is involved for the journey. « DGTransportUnit » is the list of iD to identify the each transport unit composing the ”convoy”. For road transport, at least VIN and Plate Number of the tractor and the trailers hanged together. For water way, the ID of the vessel and the containers loaded on it…
boolean activation: If the value is true then the transport is active that means the transport Unit is loaded with dangerous goods and electronic document are available if needed. If the value is false then the transport is inactive. Electronic document are no more available but archives are done for the three next months.

Exit parameters

boolean : « TRUE » indicates that activation or deactivation is done without error. “FALSE” indicates that activation or deactivation is not done due to an error and the method must be invoked until TRUE.

Constraints

This request must be transmitted to only one TP1 and must take into account all the documents related to the transport unit. In certain case the transport unit is a list of transport units (Tractor and its trailers, Wagon and its containers …). Thus, at a time, each transport unit (or group of linked transport units) is only registered on one and only one TP2 and consequently on one and only one TP1.

Request example

```
<?xml version="1.0" encoding="utf-8"?><soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
  <handleLoadedVehicle xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
   <ns1:dgCarryingVehicle extensionName="GeotransMD" extensionVersion="0.2.3"
id="ID_UNIQUE1" modelBaseVersion="2" xmlns:ns1="http://datex2.eu/schema/2/2_0">
    <ns1:dgTransportUnit xsi:type="ns1:RoadVehicle">
     <ns1:vehicleExtension>
      <ns1:vehicleRegistrationPlateIdentifier>BB333DD</ns1:vehicleRegistrationPlat
eIdentifier>
Adding Country of the Car      <ns1: vehicleCountryOfOrigin>fr</ns1:
vehicleCountryOfOrigin>
      <ns1:vehicleCharacteristics>
       <ns1:vehicleType>lorry</ns1:vehicleType>
      </ns1:vehicleCharacteristics>
     </ns1:vehicleExtension>
```

```
     <ns1:vehicleExtension>
      <ns1:vehicleRegistrationPlateIdentifier>BB444DD</ns1:vehicleRegistrationPlat
eIdentifier>
      <ns1:vehicleCharacteristics>
       <ns1:vehicleType>trailer</ns1:vehicleType>
      </ns1:vehicleCharacteristics>
     </ns1:vehicleExtension>
    </ns1:dgTransportUnit>
    <ns1:dgPublication lang="fr">
     <ns1:publicationTime xsi:type="ns1:DateTime">2015-07-
20T11:57:54+02:00</ns1:publicationTime>
     <ns1:publicationCreator>
      <ns1:country>fr</ns1:country>
      <ns1:nationalIdentifier>publicationCreator</ns1:nationalIdentifier>
     </ns1:publicationCreator>
    </ns1:dgPublication>
    <ns1:dgExchange>
     <ns1:keepAlive>true</ns1:keepAlive>
     <ns1:response>acknowledge</ns1:response>
     <ns1:supplierIdentification>
      <ns1:country>fr</ns1:country>
      <ns1:nationalIdentifier>supplier</ns1:nationalIdentifier>
     </ns1:supplierIdentification>
    </ns1:dgExchange>
   </ns1:dgCarryingVehicle>
   <activation>true</activation>
  </handleLoadedVehicle>
</soapenv:Body>
</soapenv:Envelope>
```

4.1.2 getDGTDocument

This method is for a TP1 which requests DGT documents to other(s) TP1 for a known transport unit. It must give the reason of the request in order that the transport company can know if the request has origin emergency services or enforcement bodies or others allowed bodies.

eu.datex2.schema._2._2_0.DGCarryingVehicle getDGTDocument(java.lang.String idTransportUnit, java.lang.String countryCode, com.geotransmd.schema._1_0.RequestReasonEnum requestReason) throws java.rmi.RemoteException

Entry Parameters

java.lang.String idTransportUnit : One o the iD of one of the transport units composing the convoy
java.lang.String countryCode : Country code of the transport unit based on ISO 3166-1 (ex : FR)
com.geotransmd.schema._1_0.RequestReasonEnum requestReason : Reason of the request from the predefine list emergencyServices, enforcementBodies, customs, infrastructureManagers, trafficManagers, statisticsProducers, preventiveSafety.
Name of the requesting authority
Country of the requesting authority

Exit parameters

eu.datex2.schema._2._2_0.DGCarryingVehicle : Transport documents issued from the object « DGFloder » out of « contains ».

Constraints

Issued from request of the public service, this request is transmited by the TP1 which is able to associate the reason of the request.
On the basis of this reason, TP2 will transmit all or part of the data taking into account the regulatory obligations. For example, in case of request for statistical purposes, the TP2 may limit the transmission of personal data such as the names of people, their phone number, part of the address limiting to the town or city.
If the request is issued from emergency services (requestReason is valued with emergencyServices), the TP2 must transmit all the stored data including the last location if given by the transport company.
All content of this xml file (included optional classes and attributes) must be transmitted by all TP1 involved in the data exchange to emergency responder which has requested the data.
When reason are emergencyServices or enforcementBodies, TP2 has to transmit all the data given by the transport company.
For the other reasons, which allow added services, TP2 can transmit only a part of the data and especially, to guaranty privacy, hide names and phone numbers, and also part of geographical addresses.

Request example

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
    <getDGTDocument>
       <idTransportUnit>5T</idTransportUnit>
       <countryCode>FR</countryCode>
       <requestReason>emergencyServices</requestReason>
    </getDGTDocument>
</soapenv:Body>
</soapenv:Envelope>
```

4.1.3 getArchiveJourneyList

This methods is a request from one TP1 to another one for getting the list of the journey done by a transport unit between two dates within the last three months. The TP1 requester knows which public body is the original requester and can give the reason of the request according to the status of this public body given in the registration process.

com.geotransmd.schema._1_0.Journey[] getArchiveJourneyList(java.lang.String idTransportUnit, java.lang.String countryCode, com.geotransmd.schema._1_0.RequestReasonEnum requestReason, java.util.Calendar startTime, java.util.Calendar endTime) throws java.rmi.RemoteException

Entry parameters

java.lang.String idTransportUnit : one of the iD of the transport unit
java.lang.String countryCode : Country code (ex : FR)
com.geotransmd.schema._1_0.RequestReasonEnum requestReason : Reason of the request (emergencyServices, enforcementBodies, customs, infrastructureManagers, trafficManagers, statisticsProducers, preventiveSafety)
java.util.Calendar startTime : date to begin the research
java.util.Calendar endTime : date to finish the research

Exit parameters

com.geotransmd.schema._1_0.Journey[] : list of the journey (Date/Time of active period) of the transport unit . If the list is too long the response could be a time out. In this case, the request must be done a second time with a shortest period.

Constraints

No Constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <soapenv:Body>
            <getArchiveJourneyList
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
                  <idTransportUnit>5T</idTransportUnit>
                  <countryCode>FR</countryCode>
                  <requestReason>emergencyServices</requestReason>
                  <startTime xsi:type="ns1:DateTime"
xmlns:ns1="http://datex2.eu/schema/2/2_0">2017-11-27T00:00:00+01:00</startTime>
                  <endTime xsi:type="ns2:DateTime"
xmlns:ns2="http://datex2.eu/schema/2/2_0">2017-12-01T00:00:00+01:00</endTime>
            </getArchiveJourneyList>
      </soapenv:Body>
</soapenv:Envelope>
```

4.1.4 getTransportUnitByArea

This method gets the list of ID of actives transport units located in a circle which define the area of request. This area is an entry parameter as well as the reason of request. This area is circle defined with its centre (WGS84 coordinates) and a distance in meter for the radius.
This method is for emergency situation when an accident occurs and iD of the transport unit involved is not available.
The TP1 has to transmit the request to all TP1 and all TP2, compile the responses that are list of iD and transmit the response to the emergency responder which has invoked the method. To manage that the TP1 do not need to have implemented geographical services.

java.lang.String[] getTransportUnitByArea(float x, float y, int radius, com.geotransmd.schema._1_0.RequestReasonEnum requestReason) throws java.rmi.RemoteException

Entry parameters

float x: x coordinate of the centre of the circle (WGS84 longitude)
float y: y coordinate of the centre of the circle (WGS84 latitude)
int radius: radius of the circle in meter
com.geotransmd.schema._1_0.RequestReasonEnum requestReason : reason of the request (emergencyServices, enforcementBodies, customs, infrastructureManagers, trafficManagers, statisticsProducers, preventiveSafety)

Exit parameters

java.lang.String[] : list of the iD of the transport units in the area

Constraints

No constraints

Request example

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

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <soapenv:Body>
            <getTransportUnitByArea
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
                  <x>1.5612</x><y>-0.49848</y>
                  <radius>150</radius>
                  <requestReason>emergencyServices</requestReason>
            </getTransportUnitByArea>
      </soapenv:Body>
</soapenv:Envelope>
```

4.2 OPTIONAL METHODS

4.2.1 saveTrackingDataVehicle

This method allows to send the transport unit location and, if available, other dynamic parameters (alarms and/or events) from TP2 to TP1.
This method allows to implement added services for authorities or public service like green line at customs, control access measures in define areas, extra authorisation to overpass traffic restrictions.

boolean saveTrackingDataVehicle(eu.datex2.schema._2._2_0.DGCarryingVehicle position) throws java.rmi.RemoteException

Entry parameters

eu.datex2.schema._2._2_0.DGCarryingVehicle position : object « DGCarryingVehicle » where « DGTransportUnit » is mantatory, associated with « DGVehicleInformationDynamic » (mainly « PositioningInformation ») and « DGLoadInformationDynamic » (depending transmitted alarms and events)

The TP2 sender is identified by TP1 thanks to its public key of its certificate.

Exit parameters

boolean : « TRUE » indicates that data transmission is done without error. “FALSE” indicates that data transmission is not done due to an error and the method must be invoked until TRUE.

Constraints

This request is invoked by the TP2 on the TP1 where the journey has been activated.
By using this method, TP2 and TP1 ensure a higher level of security in the data exchange because, only iD of Transport unit and position are transmitted in the same time without description of the dangerous goods. Dangerous goods description will be given by invocation of getDGTDocument method with other reason than emergency services.

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <soapenv:Body>
```

```
            <saveTrackingDataVehicle
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
                  <ns1:dgCarryingVehicle id="b669bd75-ffff-4313-8772-
3d138e970c4e" modelBaseVersion="2" xmlns:ns1="http://datex2.eu/schema/2/2_0">
                        <ns1:dgVehicleInformationDynamic>
                              <ns1:dgtDateTime>
                                    <ns1:currentDateTime
xsi:type="ns1:DateTime">2017-12-01T00:00:00+01:00</ns1:currentDateTime>
                              </ns1:dgtDateTime>
                              <ns1:positioningInformation>
                                    <ns1:speed>0.0</ns1:speed>
                                    <ns1:pointByCoordinatesExtended>
                                          <ns1:pointCoordinates>
                                                <ns1:latitude>-
0.49848</ns1:latitude>

      <ns1:longitude>1.5612</ns1:longitude>
                                          </ns1:pointCoordinates>
                                    </ns1:pointByCoordinatesExtended>
                              </ns1:positioningInformation>
                              <ns1:dgTransportUnit xsi:type="ns1:RoadVehicle">
                                    <ns1:vehicleExtension>

      <ns1:vehicleRegistrationPlateIdentifier>5T</ns1:vehicleRegistrationPlateIden
tifier>
                                          <ns1:vehicleCharacteristics>

      <ns1:vehicleType>lorry</ns1:vehicleType>
                                          </ns1:vehicleCharacteristics>
                                    </ns1:vehicleExtension>
                                    <ns1:vehicleExtension>

      <ns1:vehicleRegistrationPlateIdentifier>1564897</ns1:vehicleRegistrationPlat
eIdentifier>
                                          <ns1:vehicleCharacteristics>

      <ns1:vehicleType>trailer</ns1:vehicleType>
                                          </ns1:vehicleCharacteristics>
                                    </ns1:vehicleExtension>
                              </ns1:dgTransportUnit>
                        </ns1:dgVehicleInformationDynamic>
                        <ns1:dgPublication lang="fr">
                              <ns1:publicationTime xsi:type="ns1:DateTime">2017-
12-01T16:21:32+01:00</ns1:publicationTime>
                              <ns1:publicationCreator>
                                    <ns1:country>fr</ns1:country>

      <ns1:nationalIdentifier>publicationCreator</ns1:nationalIdentifier>
                              </ns1:publicationCreator>
                        </ns1:dgPublication>
                        <ns1:dgExchange>

      <ns1:clientIdentification>TESTSTP1TP2</ns1:clientIdentification>
                              <ns1:keepAlive>true</ns1:keepAlive>
                              <ns1:response>acknowledge</ns1:response>
                              <ns1:supplierIdentification>
```

```
                                        <ns1:country>fr</ns1:country>

        <ns1:nationalIdentifier>supplier</ns1:nationalIdentifier>
                                    </ns1:supplierIdentification>
                            </ns1:dgExchange>
                        </ns1:dgCarryingVehicle>
                    </saveTrackingDataVehicle>
        </soapenv:Body>
</soapenv:Envelope>
```

4.3 SENDTP2REGISTRATIONREQUEST

This method allows that a TP2 requests a registration on a TP1.

boolean sendTP2RegistrationRequest(com.geotransmd.schema._1_0.TP2 tp2) throws java.rmi.RemoteException, com.geotransmd.schema._1_0.ErrorMessageException

Entry parameters :

com.geotransmd.schema._1_0.TP2 tp2 : TP2 description:

- URL : TP2 entry point for the TP1
- Public key of TP2 certificate
- TP2 Company name
- Contact name, mail and phone number of the responsible person

Exit parameters

boolean : « TRUE » indicates that data transmission is done without error. “FALSE” indicates that data transmission is not done due to an error and the method must be invoked until TRUE.

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
        <soapenv:Body>
                <sendTP2RegistrationRequestRequestMessage
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
                        <tp2>
                                <ns1:id xmlns:ns1="http://geotransmd.com/schema/1_0">id</ns1:id>
                                <ns2:nom
xmlns:ns2="http://geotransmd.com/schema/1_0">nom</ns2:nom>
                                <ns3:postalAddress xmlns:ns3="http://geotransmd.com/schema/1_0">
                                        <ns3:streetAddress1>streetAddress1</ns3:streetAddress1>
                                        <ns3:streetAddress2>streetAddress2</ns3:streetAddress2>
                                        <ns3:postalCode>postalCode</ns3:postalCode>
                                        <ns3:locality>locality</ns3:locality>
                                        <ns3:country>country</ns3:country>
                                </ns3:postalAddress>
                                <ns4:contact xmlns:ns4="http://geotransmd.com/schema/1_0">
                                        <ns4:contactName>contactName</ns4:contactName>
                                        <ns4:contactAddress>contactAddress</ns4:contactAddress>
```

```
        <ns4:contactTelephoneNumber>contactTelephoneNumber</ns4:contactTelephoneNumber>
                                                <ns4:contactEmail>contactEmail</ns4:contactEmail>

        <ns4:contactResponsibility>contactResponsibility</ns4:contactResponsibility>

        <ns4:additionalContactInformation>additionalContactInformation</ns4:additionalContactInfor
mation>
                                        </ns4:contact>
                                        <ns5:url
xmlns:ns5="http://geotransmd.com/schema/1_0">url</ns5:url>
                                        <ns6:login
xmlns:ns6="http://geotransmd.com/schema/1_0">login</ns6:login>
                                        <ns7:password
xmlns:ns7="http://geotransmd.com/schema/1_0">password</ns7:password>
                                        <ns8:certificat
xmlns:ns8="http://geotransmd.com/schema/1_0">certificat</ns8:certificat>
                                </tp2>
                        </sendTP2RegistrationRequestRequestMessage>
        </soapenv:Body>
</soapenv:Envelope>
```

5 TP1 EXTERNAL SERVICES

Identification and authentication among systems (TP1, TP2 and external IT systems from public bodies) is done by certificates.

5.1 MANDATORY METHODS

5.1.1 getDGTDocument

This method allows request of DG Transport document of an identified transport unit by an external IT system registered as public body to the TP1.

eu.datex2.schema._2._2_0.DGCarryingVehicle getDGTDocument(java.lang.String idTransportUnit, java.lang.String countryCode) throws java.rmi.RemoteException, com.geotransmd.schema._1_0.ErrorMessageException

Entry Parameters

java.lang.String idTransportUnit : iD of the transport Unit
java.lang.String countryCode : Country code (ex : FR)

Paramètres en sortie

eu.datex2.schema._2._2_0.DGCarryingVehicle : DG Transport document (object « DGFloder » in « contains »)
com.geotransmd.schema._1_0.ErrorMessageException : Error occurred during data processing

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
	<soapenv:Body>
		<getDGTDocument>
			<idTransportUnit>5T</idTransportUnit>
			<countryCode>FR</countryCode>
		</getDGTDocument>
	</soapenv:Body>
</soapenv:Envelope>
```

5.1.2 getArchiveJourneyList

This method is a request from one external system to the TP1 where it is registered for getting the list of the journey done by a transport unit between two dates within the last three months.

com.geotransmd.schema._1_0.Journey[] getArchiveJourneyList(java.lang.String idTransportUnit, java.lang.String countryCode, java.util.Calendar startTime, java.util.Calendar endTime) throws java.rmi.RemoteException, com.geotransmd.schema._1_0.ErrorMessageException

Entry parameters

java.lang.String idTransportUnit : ID of the transport unit
java.lang.String countryCode : Country code (ex : FR)
java.util.Calendar startTime : date to begin the research
java.util.Calendar endTime : date to finish the research

Exit parameters

com.geotransmd.schema._1_0.Journey[] : list of the journey (Date/Time of active period) of the transport unit . If the list is too long the response could be a time out. In this case, the request must be done a second time with a shortest period.
com.geotransmd.schema._1_0.ErrorMessageException : Error occurred during data processing

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <soapenv:Body>
            <getArchiveJourneyList
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
                  <idTransportUnit>5T</idTransportUnit>
                  < countryCode>FR</countryCode>
                  <startTime xsi:type="ns1:DateTime"
xmlns:ns1="http://datex2.eu/schema/2/2_0">2017-11-27T00:00:00+01:00</startTime>
                  <endTime xsi:type="ns2:DateTime"
xmlns:ns2="http://datex2.eu/schema/2/2_0">2017-12-01T00:00:00+01:00</endTime>
            </getArchiveJourneyList>
      </soapenv:Body>
</soapenv:Envelope>
```

5.1.3 getTransportUnitByArea

This method get the list of ID of actives transport units located in a circle which define the area of request. This area is an entry parameter as well as the reason of request. This area is circle defined with its centre (WGS84 coordinates) and a distance in meter for the radius.
This method is normally for emergency situation when an accident occurs and iD of the transport unit involved is not available.
If TP1 and public bodies develop added services based on geographical position and TP2 and Transport companies agree to use these added services, then this method can be invoked by public bodies.

java.lang.String[] getTransportUnitByArea(float x, float y, int radius,
com.geotransmd.schema._1_0.RequestReasonEnum requestReason) throws
java.rmi.RemoteException

Entry parameters

float x: x coordinate of the centre of the circle (WGS84 longitude)
float y: y coordinate of the centre of the circle (WGS84 latitude)
int radius: radius of the circle in meter

Exit parameters

java.lang.String[] : list of the iD of the transport units in the area

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
	<soapenv:Body>
		<getTransportUnitByArea
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
			<x>1.5612</x><y>-0.49848</y>
			<radius>150</radius>
		</getTransportUnitByArea>
	</soapenv:Body>
</soapenv:Envelope>
```

5.2 OPTIONAL METHODS

5.2.1 getDynamicInformation

This method allows the request of dynamic information (Location by instance) of an identified transport unit by a public body to the TP1 on which it is registered.

DGCarryingVehicle getDynamicInformation(String idTransportUnit) throws RemoteException

Entry parameters

String idTransportUnit : Id of the transport unit

Exit parameters

DGCarryingVehicle : send the object « DGCarryingVehicle » in which « DGTransportUnit », « DGVehicleInformationDynamic » and « DGLoadInformationDynamic » are instanced.

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
	<soapenv:Body>
		<getDynamicInformation
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
			<idTransportUnit>5T</idTransportUnit>
		</getDynamicInformation>
	</soapenv:Body>
</soapenv:Envelope>
```

5.2.2 sendPublicServiceRegistrationRequest

This method allows that a public body requests a registration on a TP1.

boolean sendPublicServiceRegistrationRequest(com.geotransmd.schema._1_0.ServicePublic servicePublic) throws java.rmi.RemoteException, com.geotransmd.schema._1_0.ErrorMessageException

Entry parameters

com.geotransmd.schema._1_0.ServicePublic servicePublic : Public service description:

- Public key of public bodies certificate
- Public body address (street, postal code, locality)
- Public body name
- Contact name, mail and phone number of the responsible person
- Actor type: competent authority (include also infrastructure manager), emergency responders, enforcement bodies, security bodies.

Exit parameters

boolean : « TRUE » indicates that data transmission is done without error. “FALSE” indicates that data transmission is not done due to an error and the method must be invoked until TRUE.

Constraints

No constraint

Comment from France

It could be interesting to introduce the area of competence of the public service in order to avoid the risk that due to some input error a public service receive information from a transport unit located outside of its geographical area. Eg police from Ukraine request information from a vehicle and receive information from another one driving in Italy due to a mistake in the identification plate number. In this case with the competence area, TP2 would have been able to give a response vehicle outside of your area.

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
        <soapenv:Body>
                <sendPublicServiceRegistrationRequest
xmlns="http://geotransmd.com/wsdl/TP1Services/1_0">
                        <servicePublic>
                                <ns1:id xmlns:ns1="http://geotransmd.com/schema/1_0">id</ns1:id>
                                <ns2:nom
xmlns:ns2="http://geotransmd.com/schema/1_0">nom</ns2:nom>
                                <ns3:postalAddress xmlns:ns3="http://geotransmd.com/schema/1_0">
                                        <ns3:streetAddress1>streetAddress1</ns3:streetAddress1>
                                        <ns3:streetAddress2>streetAddress2</ns3:streetAddress2>
                                        <ns3:postalCode>postalCode</ns3:postalCode>
                                        <ns3:locality>locality</ns3:locality>
                                        <ns3:country>country</ns3:country>
                                </ns3:postalAddress>
                                <ns4:contact xmlns:ns4="http://geotransmd.com/schema/1_0">
```

```
                                        <ns4:contactName>contactName</ns4:contactName>
                                        <ns4:contactAddress>contactAddress</ns4:contactAddress>

        <ns4:contactTelephoneNumber>contactTelephoneNumber</ns4:contactTelephoneNumber>
                                        <ns4:contactEmail>contactEmail</ns4:contactEmail>

        <ns4:contactResponsibility>contactResponsibility</ns4:contactResponsibility>

        <ns4:additionalContactInformation>additionalContactInformation</ns4:additionalContactInfor
mation>
                                    </ns4:contact>
                                    <ns5:actorType
xmlns:ns5="http://geotransmd.com/schema/1_0">competentAuthority</ns5:actorType>
                                </servicePublic>
                            </sendPublicServiceRegistrationRequest>
        </soapenv:Body>
</soapenv:Envelope>
```

6 TP2INTERNALSERVICES

Identification and authentication among systems (TP1, TP2 and external IT systems from public bodies) is done by certificates.

6.1 MANDATORY METHODS

6.1.1 getDGTDocument

This method allows a TP1 to request DG tansport documents for an identified transport unit with the reason of the request.

eu.datex2.schema._2._2_0.DGCarryingVehicle getDGTDocument(java.lang.String idTransportUnit, java.lang.String countryCode, com.geotransmd.schema._1_0.RequestReasonEnum requestReason) throws java.rmi.RemoteException

Entry parameters

java.lang.String idTransportUnit : iD of the transport unit
java.lang.String countryCode : Countruy code (ex : FR)
com.geotransmd.schema._1_0.RequestReasonEnum requestReason : Request reason (emergencyServices, enforcementBodies, customs, infrastructureManagers, trafficManagers, statisticsProducers, preventiveSafety)

Exit parameters

eu.datex2.schema._2._2_0.DGCarryingVehicle Transport document (object « DGFloder » in « contains »)

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <soapenv:Body>
            <getDGTDocument xmlns="http://geotransmd.com/wsdl/TP2Services/1_0">
                  <idTransportUnit>7478RT33</idTransportUnit>
                  <countryCode>FR</countryCode>
                  <requestReason>emergencyServices</requestReason>
            </getDGTDocument>
      </soapenv:Body>
</soapenv:Envelope>
```

6.2 SAVETP2OK

This method is the acknowledgment from TP1 to TP2 that the registration is done and that the TP2-TP1 is ready for operation.

boolean saveTP2OK(boolean saved, java.lang.String idTp2, java.lang.String commentaire, java.lang.String login, java.lang.String password) throws java.rmi.RemoteException, com.geotransmd.schema._1_0.ErrorMessageException {

Paramètres en entrée :

boolean saved : « TRUE » if the registration request is validated
java.lang.String idTp2 : idGiven to the TP2
java.lang.String commentaire : Coments of the registration request
java.lang.String login : login for access to TP1 (not in use)
java.lang.String password : password for access to TP1 (not utilisé actuin useellement)

Paramètres en sortie

boolean : « TRUE » f OK
com.geotransmd.schema._1_0.ErrorMessageException : error occured during data processing

Constraints

No constraint

Request example

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
	<soapenv:Body>
		<saveTP2OK xmlns="http://geotransmd.com/wsdl/TP2Services/1_0">
			<save>true</save>
			<idTP2>idTp2</idTP2>
			<comment>commentaire</comment>
			<login>login</login>
			<password>password</password>
		</saveTP2OK>
	</soapenv:Body>
</soapenv:Envelope>
```

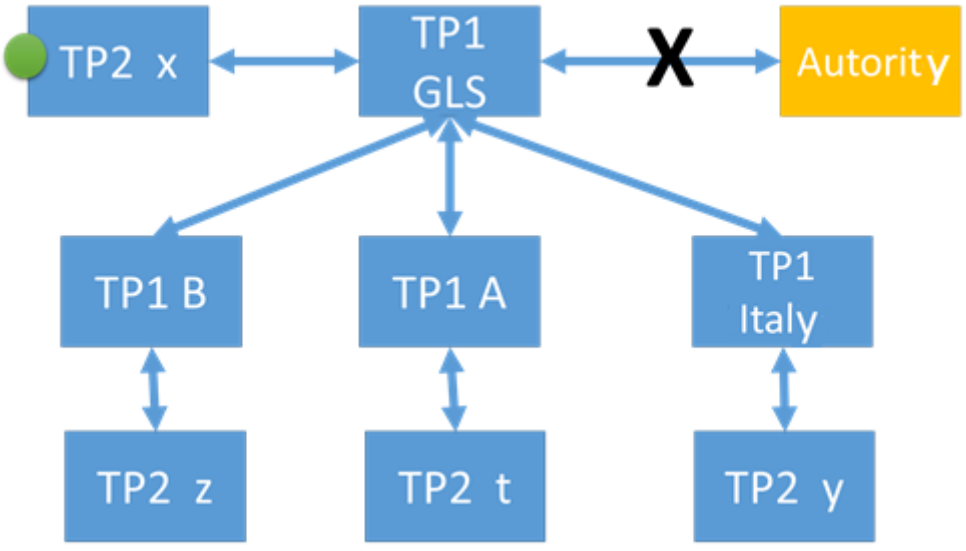
7 ERROR MANAGEMENT

7.1 CONNEXION ERROR TO TP1 BY A PUBLIC SERVICE

This type of error may be caused by the following problems:

15. ***"No 3G/4G network": Error message managed only at the level of the public service software***
16. ***"Problem with the internet network connecting TP1 and public service": Error message managed only at the level of the public service software***
17. ***"Server failure" or "Stopped server": Error message managed at the level of the public service software but also error could be transmitted by the TP1 IT system***
18. ***"Computer error on the TP1": Web service exception sent back to client. The public service software tracks the error for further analysis. Error message managed at the level of the public service software but also error could be transmitted by the TP1 IT system***

No document is given.



The green dot indicates the TP2 that relates to the transport document.

7.2 GEOFENCING DENIAL DOCUMENT NOT SEND

The request arrives at TP2 but TP2 do not send back the eDGT document because the requesting authority in not in charge of the geaographical area where the vehicule is.

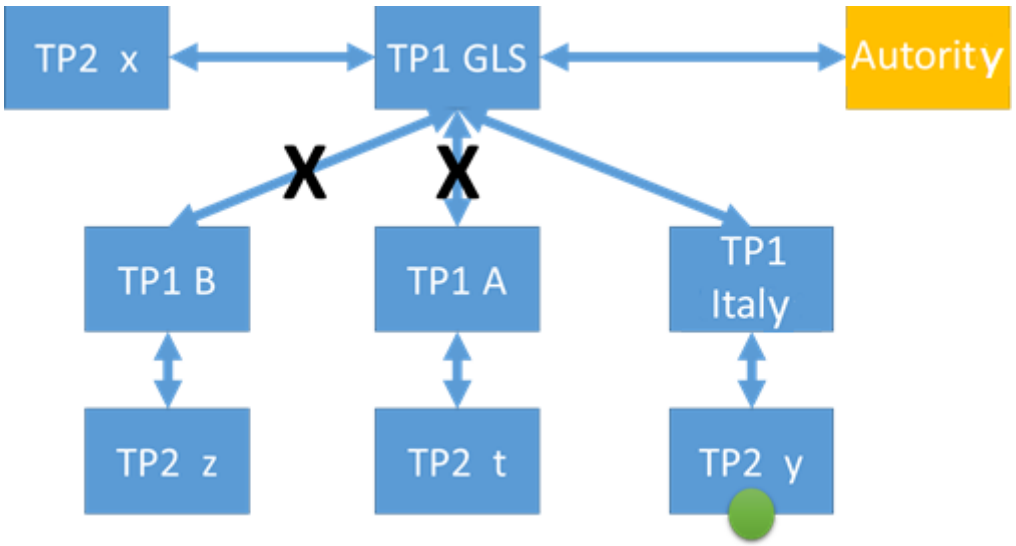
7.3 CONNEXION ERROR BETWEEN TP1S

7.3.1 A transport document is sent back

One or more TP1 cannot be reached but one TP1 sends a document. The TP1 that has sent the query forwards the transport document and stores the error messages sent by the unreachable or failed TP1

TP1 may be unreachable for the following reasons:

19. ***Internet network problem***
20. ***Stopped or failed server***
21. ***Software bug or computer error in one or several TP1***
22. ***GEOFENCING DENIAL The request arrives at TP2 but TP2 do not send back the eDGT document because the requesting authority in not in charge of the geaographical area where the vehicule is.***
23.



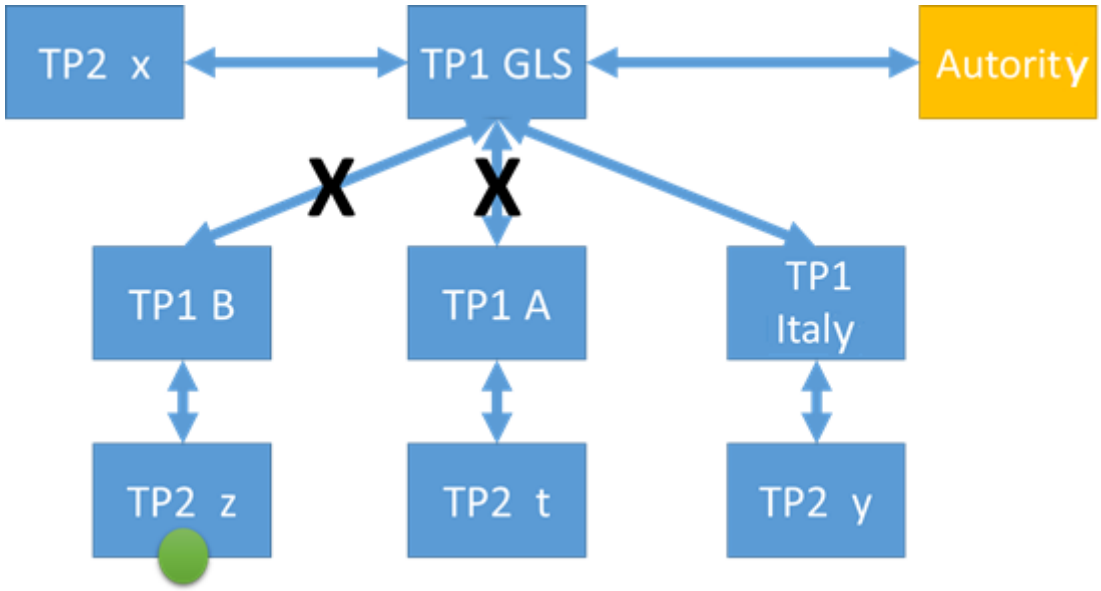
The green dot indicates the TP2 that relates to the transport document.

7.3.2 No transport document is sent back

One or more TP1 cannot be reached and no transport document has been sent back. The TP1 who requested the data sends an exception message with the error code "No transport document but some TP1 didn't answer" and records the errors messages from the failed or unreachable TP1s.

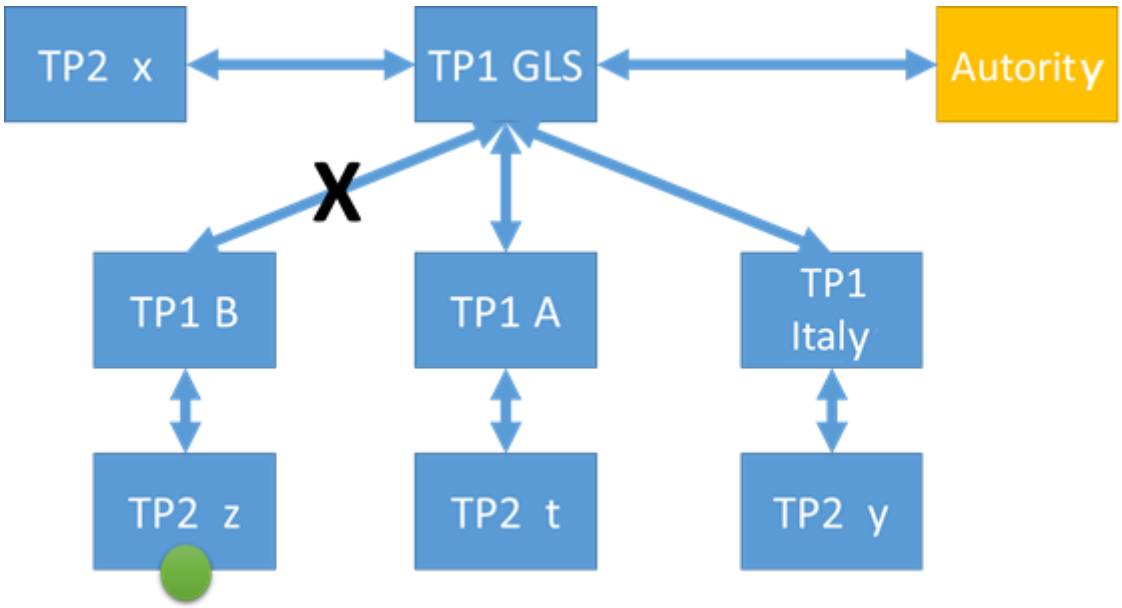
TP1 may be unreachable for the following reasons:

24. ***Internet network problem***
25. ***stopped or failed server***
26. ***computer error in one or several TP1***
27. ***Different version of the eDgt protocole implemented***
28. ***GEOFENCING DENIAL The request arrives at TP2 but TP2 do not send back the eDGT document because the requesting authority in not in charge of the geaographical area where the vehicle is.***



The green dot indicates the TP2 that relates to the transport document.

Other case:



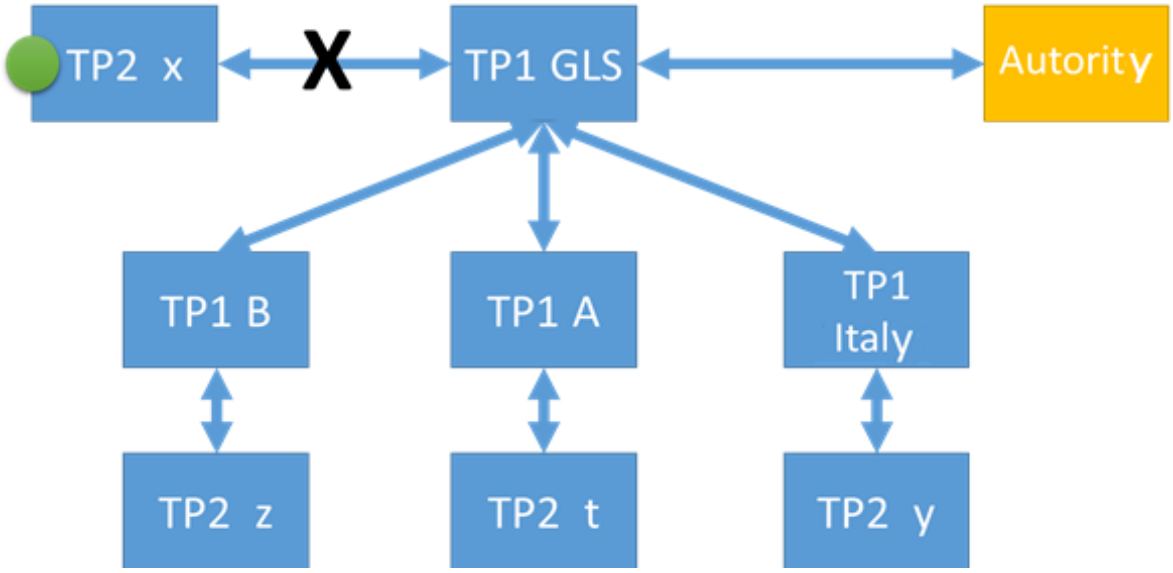
The green dot indicates the TP2 that relates to the transport document.

7.4 CONNEXION ERROR BETWEEN TP1 AND TP2

The TP2 that manages the transport document does not answer to the query of the TP1. The TP1 that sent the query sends an exception message with the error code « Transport Unit known but no response from TP2 » and records the error messages sent back by the TP2.

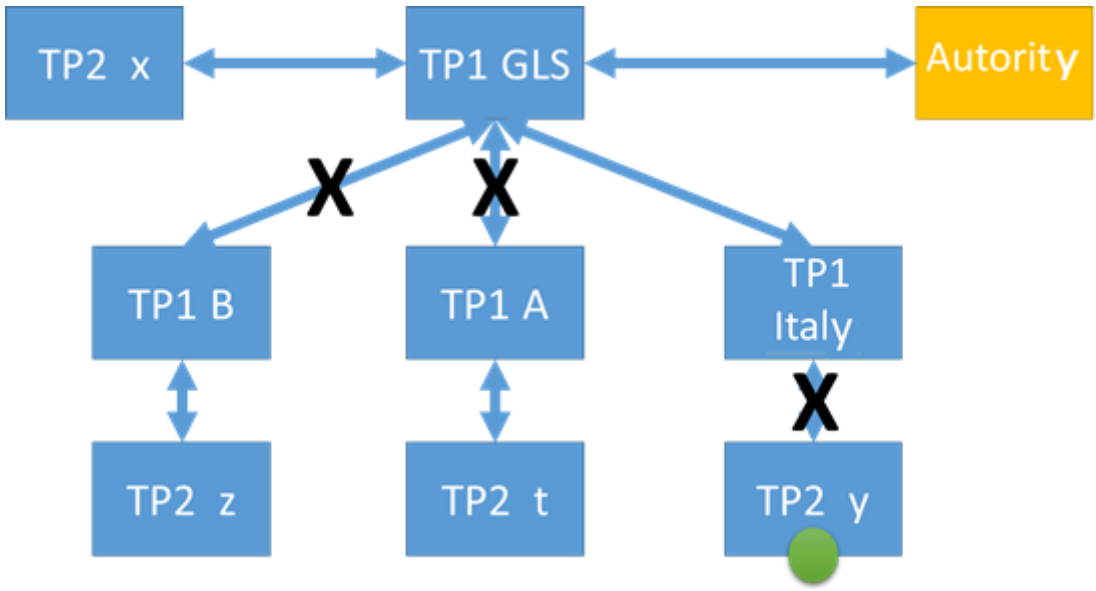
This type of error may be caused by the following problems:

29. ***Internet network problem between TP1 and TP2***
30. ***TP2 server failed or stopped***
31. ***Software bug or computer error with TP2***



The green dot indicates the TP2 that relates to the transport document.

Other case:



The green dot indicates the TP2 that relates to the transport document.

The TP1 that has sent the query sends back the exception message sent by the TP1 where the unreachable TP2 is registered.

7.5 ERROR NAMING

7.5.1 General error response issued from TP1 or TP2

This error is like the timeout.1
<enumeration value="unknownReason"/>
The requested TP1 responses an exception and not a valid response by example the version of XSD are not the same between the 2 systems or invalid XML.
<enumeration value="processingError"/>

7.5.2 Functional response from TP1 to a request from another TP1

The requested Transport Unit is not in the list of transport unit
<enumeration value="unknownTransportUnit"/>
If the invocated method is getDGTDocument , then error message indicates that the transport unit is not active now.
If the invocated method is getArchiveJourneyList, then error message indicates that the transport unit was never active during the requested period.
If the invocated method is getDynamicInformation, then error message indicates that the transport unit is not active now.

~~The requested Transport Unit is known but it is not active at the time of the request~~
~~<enumeration value="notActiveTransportUnit"/>~~

7.5.3 Functional response from TP1 to a request from a public body

The requested TP1 responses an incomplete information
<enumeration value="incompleteInformation"/>
If the public body do not transmit parameters
~~Functional response from TP2~~
~~<enumeration value="unknownCarrier"/>~~

~~The requested Transport Unit is not in the list of transport unit which have done a transport (List for archive)~~
~~<enumeration value="inactiveTransport"/>~~

7.5.4 Response given by a TP2

This response is given by a TP2 which do not want to transmit information because:

- the Publicbody which requests as no right to do it because the transport unit is not in its geographical area of competence
  <enumeration value="transportNotInQueryRegion"/>
- the dynamic information is not available, or the carrier don't want to share dynamic information with the requesting public body
  <enumeration value="noDynamicInformationAvailable"/>

[1] If all requested TP1s are in timeout, then it could be that the requester TP1 encounters network failure happened after receiving the request from the public body.

7.5.5 Response translated by the TP1 to be considered by the Publicbody

If the response do not contain a valid eDGI corresponding to the requested transport unit, then the TP1 transmits to the public body:

<enumeration value="unkownTransportUnit”/> when all TP1s respond “unkownTransportUnit”

In this case the Carrier could be fined

<enumeration value="invalidResponseFromTP1"/> when a least one TP1 is “unknownReason”

In this case, the architecture is responsible and carrier not.

or if the reason is due to the responsible TP2

<enumeration value="invalidResponseFromTP2"/>

In this case the carrier should be fined

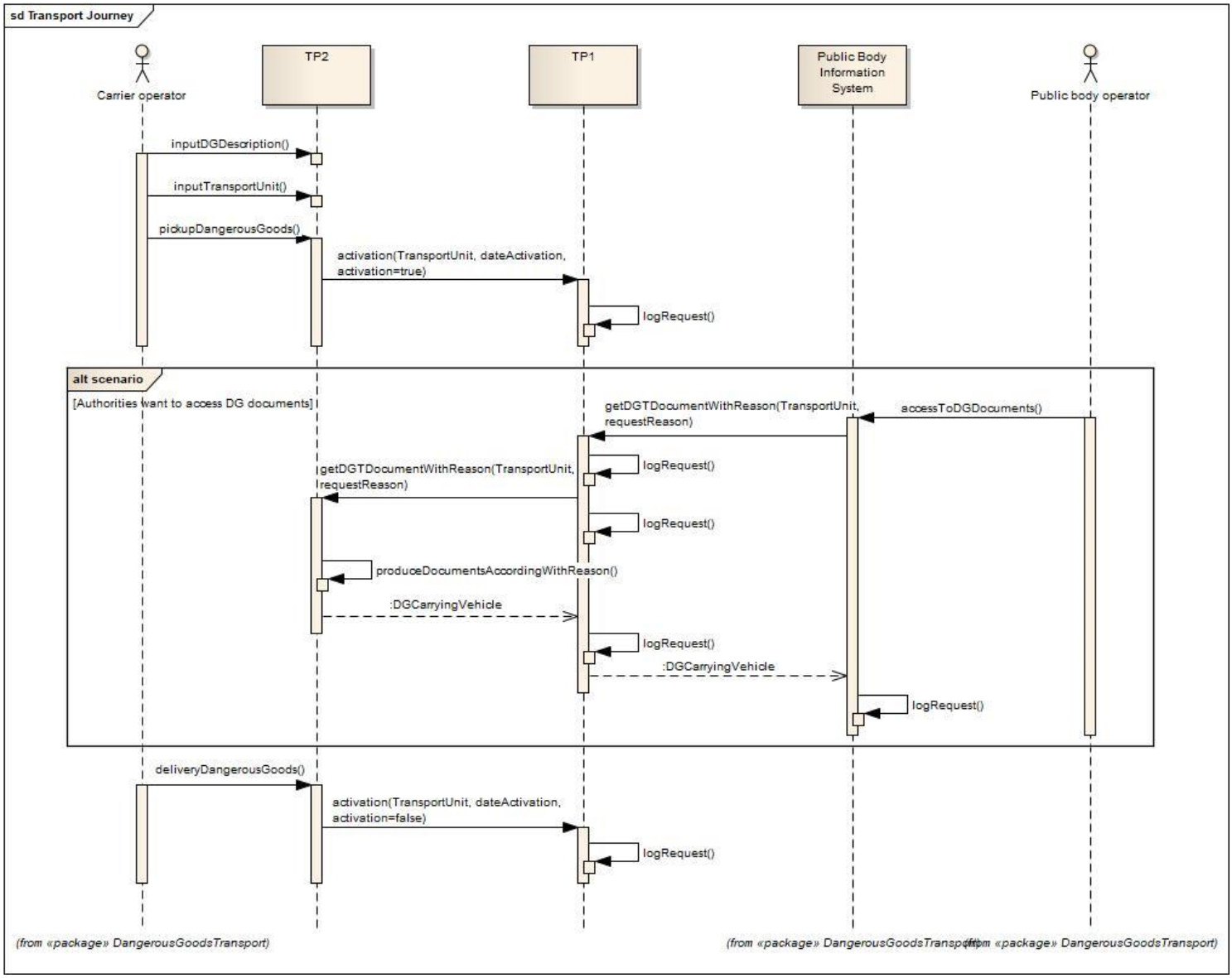
or <enumeration value="transportNotInQueryRegion"/>

In this case the public body is requesting information about a transport which is out of its territory.

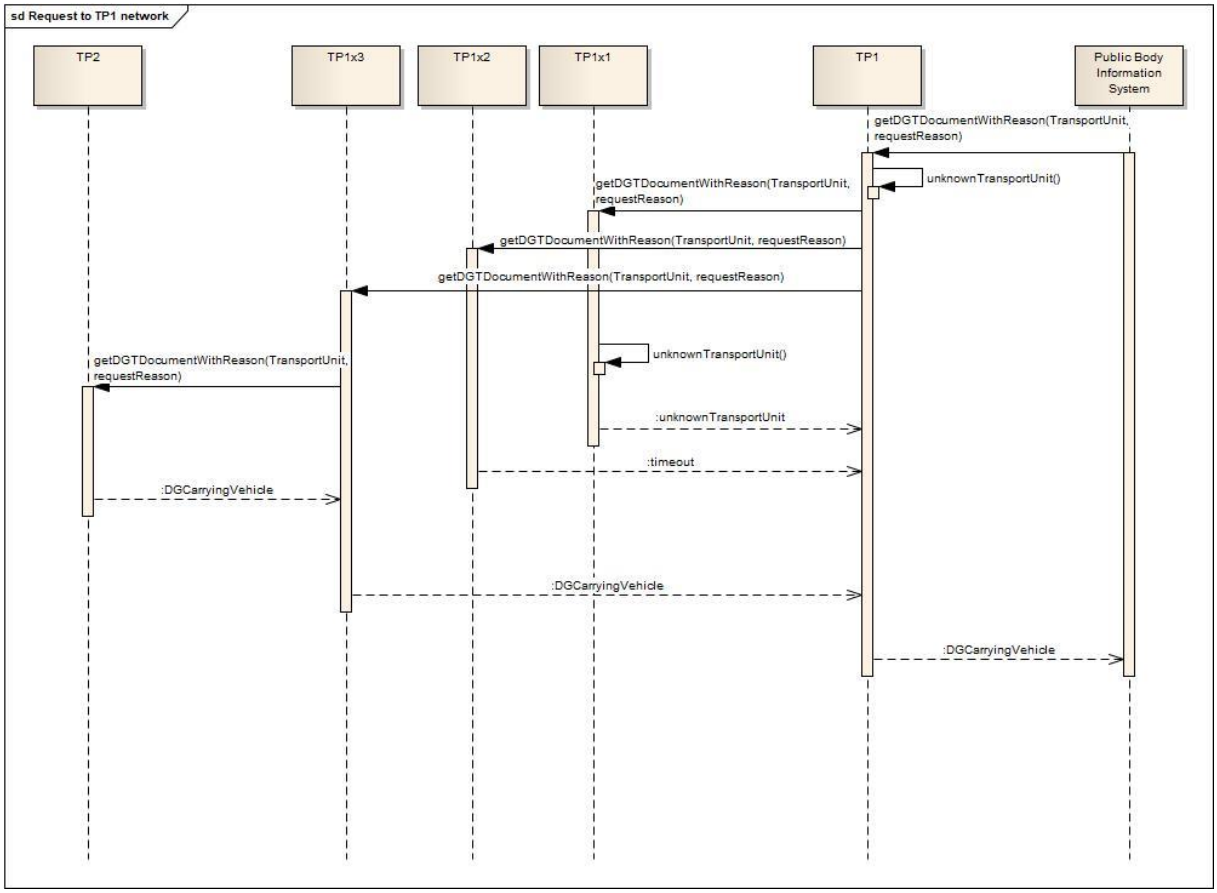
If the public body is in front of the transport unit, that means that the location given by the On board Unit is false and the carrier should be fined

8 SEQUENCE DIAGRAM EXAMPLES

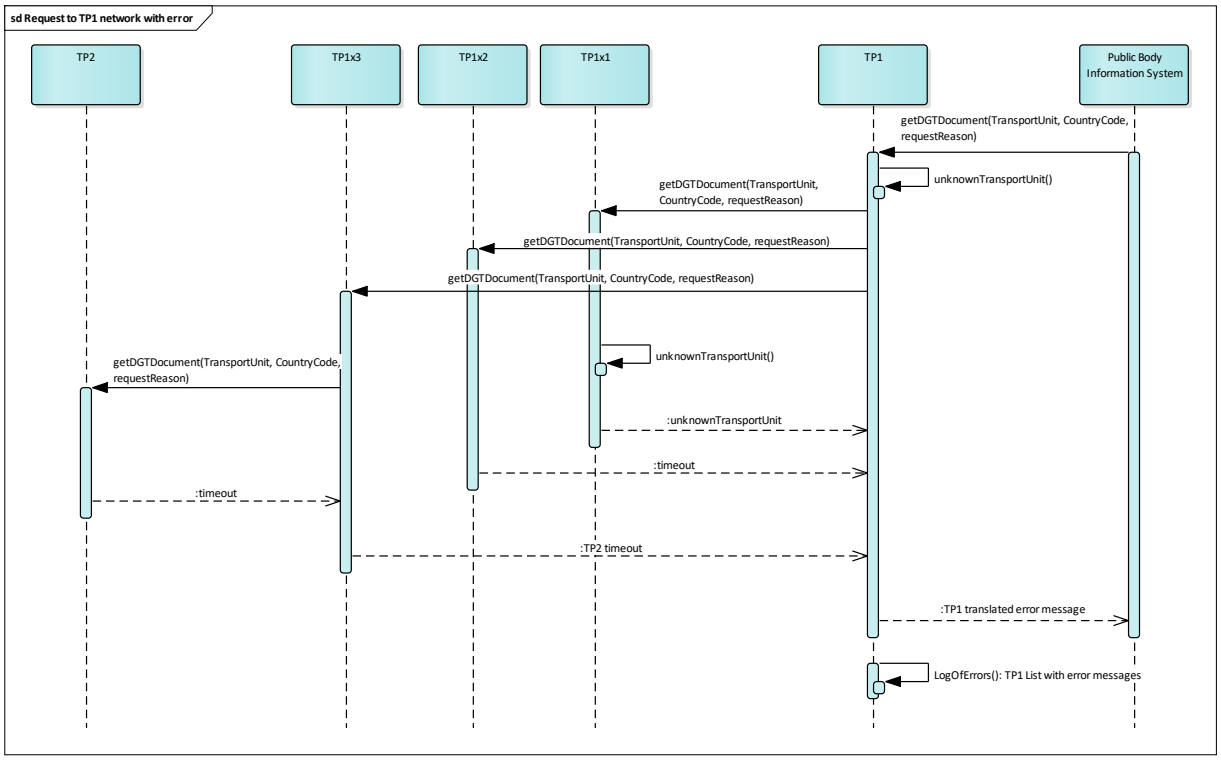
8.1 TRANSPORT JOURNEY WITH DOCUMENT REQUEST



8.2 DOCUMENT REQUEST THROUGH TP1 NETWORK



8.3 DOCUMENT REQUEST THROUGH TP1 NETWORK WITH ERROR



Examples

Report on a good case, only one TP1 queried

TP1 Identification	ExtendedDgtDenyReason	DocsRecvd	TP2 Identification
tp1.novacom.fr	-	true	tp2.schenker.com

TP1 translated error message: none (empty)

Report on an unknown transport

Identification	ExtendedDgtDenyReason	DocsRecvd	TP2 Identification
tp1.novacom.fr	unknownTransportUnit	false	
tp1.gbk.de	unknownTransportUnit	false	
tp1.ricardo.co.uk	unknownTransportUnit	false	

TP1 translated error message: unknownTransportUnit

Report on a failure

Identification	ExtendedDgtDenyReason	DocsRecvd	TP2 Identification
tp1.novacom.fr	invalidResponseFromTP1	false	
tp1.gbk.de	unknownTransportUnit	false	
tp1.ricardo.co.uk	unknownTransportUnit	false	

TP1 translated error message: invalidResponseFromTP1

8.4 TRANSPORT UNIT BY GEOGRAPHIC AREA FOR EMERGENCY RESPONDERS

To be done in the future

8.5 DYNAMIC INFORMATION REQUESTED BY PUBLIC BODY FOR "GREEN LINE"

To be done in the future