Functional and Technical Integration of SAP Outbound OEM Product into Partner Product

SAP Networked Logistics Hub 1.0

Version 1.6
February 2017
# History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Name</th>
<th>Change/Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014-11-27</td>
<td>1.0</td>
<td>SAP Networked Logistics Hub 1.0</td>
<td>First version</td>
</tr>
<tr>
<td>2015-02-24</td>
<td>1.1</td>
<td>SAP Networked Logistics Hub 1.0 FP01</td>
<td>Second version</td>
</tr>
<tr>
<td>2015-06-01</td>
<td>1.2</td>
<td>SAP Networked Logistics Hubs 1.0 FP02</td>
<td>Third version</td>
</tr>
<tr>
<td>2015-17-12</td>
<td>1.3</td>
<td>SAP Networked Logistics Hub 1.0 FP03</td>
<td>Fourth version</td>
</tr>
<tr>
<td>2016-02-26</td>
<td>1.4</td>
<td>SAP Networked Logistics Hub 1.0 FP04</td>
<td>Fifth version</td>
</tr>
<tr>
<td>2016-08-12</td>
<td>1.5</td>
<td>SAP Networked Logistics Hub 1.0 FP05</td>
<td>Sixth version</td>
</tr>
<tr>
<td>2017-02-10</td>
<td>1.6</td>
<td>SAP Networked Logistics Hub 1.0 FP06</td>
<td>Seventh version</td>
</tr>
</tbody>
</table>

**Updates:**
Payload details moved to *Payload Information* document
## Table of Contents

1. **Abstract** .................................................................................................................. 5
2. **Overview** .................................................................................................................. 6
   2.1 SAP Networked Logistics Hub Integration with Telematics Data Broker .................. 6
   2.2 SAP Networked Logistics Hub Integration with Traffic Information Services .......... 6
   2.3 SAP Networked Logistics Hub Integration with Order Management Systems .......... 7
3. **Interface Details** ...................................................................................................... 8
   3.1 Conventions ............................................................................................................. 8
   3.2 SAP Networked Logistics Hub Integration with Telematics Data Broker ............... 9
      3.2.1 Telematic Data Broker: Customer Management ............................................. 11
      3.2.2 Telematic Data Broker: Device Management ................................................ 13
      3.2.3 Geofence Replication ....................................................................................... 14
      3.2.4 Container Terminal Replication ...................................................................... 19
      3.2.5 Availability of Automatic Status of Container Terminal ................................. 23
      3.2.6 Address Resolution/Geocoding ...................................................................... 25
   3.3 Receiving Device Positions in SAP Networked Logistics Hub ............................... 27
      3.3.1 Sending Messages from SAP Networked Logistics Hub through TDB ............ 27
      3.3.2 Receiving Messages Driver/Device through TDB in SAP Networked Logistics Hub ................................................................................................................. 27
      3.3.3 Usage Log ......................................................................................................... 28
   3.4 SAP Networked Logistics Hub and Traffic Information Service ........................... 28
      3.4.1 SAP Networked Logistics Hub Integration with Telematic Data Broker (TDB) ......................................................................................................................... 28
      3.4.2 SAP Networked Logistics Hub Integration with HERE Map .......................... 28
   3.5 SAP Networked Logistics Hub Integration – Order Management (Tour Integration in SAP Networked Logistics Hub) .................................................. 29
      3.5.1 Order Replication from Order Management System to SAP Networked Logistics Hub through Telematics Data Broker .................................................. 31
      3.5.2 Order Replication SAP Networked Logistics Hub to Telematics Data Broker .... 32
      3.5.3 Event Notifications Telematics Data Broker to SAP Networked Logistics Hub ............................................................................................................... 32
      3.5.4 Sending Tour Status SAP Networked Logistics Hub to Telematics Data Broker ............................................................................................................... 33
   3.6 SAP Networked Logistics Hub Integration with SAP Yard Logistics .................. 33
      3.6.1 OData Entity Data Model ................................................................................. 34
      3.6.2 Metadata Url .................................................................................................... 35
      3.6.3 System Query Options ..................................................................................... 35
      3.6.4 Key Services .................................................................................................... 35
   3.7 Interface Data Type .................................................................................................. 36
      3.7.1 Address ............................................................................................................ 36
      3.7.2 Communication ............................................................................................... 37
      3.7.3 Geofence ......................................................................................................... 37
      3.7.4 Object .............................................................................................................. 38
      3.7.5 Geometry ....................................................................................................... 38
      3.7.6 Contact ........................................................................................................... 38
      3.7.7 Message ......................................................................................................... 39
      3.7.8 Communication Party ..................................................................................... 40
      3.7.9 Multilingual Text ............................................................................................ 40
      3.7.10 LocationBasedEvent ..................................................................................... 40
      3.7.11 Stop Type Values .......................................................................................... 40
4. **Deployment and Operations** ................................................................................... 42
   4.1 Deployed Component Structure and Deployment Options .................................... 42
   4.2 System Landscape ................................................................................................. 42
5. **Restrictions and Hints** ........................................................................................... 43

## Table of Figures

Copyright SAP AG 2017
Figure 1: Overview of SAP Networked Logistics Hub ................................................................. 5
Figure 2: Freight Forwarder and Telematics Data Service Provider (SAP Networked Logistics Hub Data Model) 10
Figure 3: Freight Forwarder Registration at Telematics Data Broker ........................................ 11
Figure 4: Geofences with Gates ................................................................................................ 15
Figure 5: Geofence Example 1 .................................................................................................. 16
Figure 6: Geofence Example 2 .................................................................................................. 18
Figure 7: Order Management Integration .................................................................................. 30
Figure 8: Tour Data Model ....................................................................................................... 31
Figure 9: Tour Events ................................................................................................................. 33
Figure 10: SAP Networked Logistics Hub Software Component Structure ............................. 42
Figure 11: Typical Production System Landscape ................................................................... 42
1 Abstract

SAP Networked Logistics Hub is a cloud service, enabling a business web of business partners interacting in the port areas.

SAP Networked Logistics Hub is targeted at the logistic hub operators (initially sea ports), freight forwarders (transportation service providers on road), container terminal operators, and parking space operators.

Figure 1: Overview of SAP Networked Logistics Hub

The SAP Networked Logistics Hub runs as a public cloud solution on the SAP HANA Cloud platform written as a native HANA application.

SAP Networked Logistics Hub interfaces are http based, adopting the REST paradigm.
2 Overview

The SAP Networked Logistics Hub integrates with the following types of services and applications:

- Telematics data services / Integration services connecting to telematics data services
- Traffic information services
- Order management applications (in a broader sense)
- Marketplace (operated by the logistic hub/hub operator)

Potentially, the SAP Networked Logistics Hub can integrate with

- Terminal Operator Information Systems/integration services
- Commercial services (parking space booking …)

The following sections provide an overview of the peer systems and respective interfaces. Each of the integration is described in detail.

2.1 SAP Networked Logistics Hub Integration with Telematics Data Broker

The SAP Networked Logistics Hub version 1.0 obtains vehicle location information via the Telematics Data Broker (TDB), which has the following capabilities:

- Provide vehicle, i.e. device positions by integrating with several telematics services and by providing an own mobile solution.
- Enable text based communication with vehicles and devices, e.g. for exchanging traffic and order related messages.
- Provide a service calculating vehicle arrival times (called estimated time of arrival service or ETA)

The integration points between the SAP Networked Logistics Hub and Telematics Data Broker Service are:

- Freight forwarder registration
- Freight forwarder telematics account definition at the telematics data broker
- Vehicle/device registration
- Notify device locations
- Exchange text messages (device specific, geo fence based)
- Replicate geo-fence
- Arrival time service

2.2 SAP Networked Logistics Hub Integration with Traffic Information Services

The SAP Networked Logistics Hub adopts DATEX II standard as consumer for receiving traffic information. The data sources that are integrated via telematics data provider (as integration service provider) are the ADAC traffic information and hub infrastructure related traffic information provided by the Hamburg port Administration (HPA) originating in a system called Hub Manager.

The SAP Networked Logistics Hub receive traffic-related information and share them with vehicles as location based messages.

The DATEX II standard is based on the location code lists and event code lists. The content is not part of the standard.
In SAP Networked Logistics Hub 1.0, an API for importing location and event code lists are not provided currently. Instead, SAP obtained the following versions of the BAST code lists:

- location code list version 12
- event code list version DE 4.01

The SAP Networked Logistics Hub also provides a UI and possibly an API for maintaining location codes for private location code lists and event code lists – as maintained by the hub manager.

### 2.3 SAP Networked Logistics Hub Integration with Order Management Systems

- Transport order/task list import into SAP Networked Logistics Hub
- Transport order/task list synchronization with SAP Networked Logistics Hub
- Notification of status update driver to order management system
- Notification of approximation and estimated time of arrival

The integration of external order management system with the SAP Networked Logistics Hub is not elaborated in this document. This document is more based on integration of tours with Telematics Information Services.
3 Interface Details

3.1 Conventions

Note: We adopt Representational State Transfer (REST)\(^1\) as the default interface paradigm. Exceptions are made where web service based standard interfaces are available such as the DATEX II for traffic/incident related communication interfaces.

Conventions:

- Related entities share a common base bath such as \texttt{/api/}
- An entity name is used in its plural form, e.g.\texttt{/api/customers/}
- We adopt lower camel case notation such as the vehicleLocation, telematicDataProvider

There are the following standard methods – not all of them need to be available for a given entity:

- GET \texttt{/api/customers/}
- GET \texttt{/api/customers/<customerID>}
- POST \texttt{/api/customers/}
- PUT \texttt{/api/customers/}
- PUT \texttt{/api/customers/<customerID>}
- DELETE \texttt{/api/customers/}
- DELETE \texttt{/api/customers/<customerID>}

Where \texttt{<customerID>} needs to be replaced with the actual server side customer ID.

We use server side identifiers. This means that the client need to retain the mapping of an own internal ID to the server's ID. With some communication parties, the role of client and server depends on the integration scenario. The client may pass its own identifier as the standard property \texttt{externalId} to the server to allow proper reverse invocation.

To avoid duplicate, creation of entity instances POST should implement idempotent behavior. For this purpose, we use the attribute \texttt{externalId} as client defined external ID for identifying entity instance in POST methods.

Example:
The SAP Networked Logistics Hub creates an object (=mobile device) entity on the Connected Car service for the customer with ID 1234. For this purpose, the POST method for customer 1234 resource \texttt{object} is invoked:

\begin{verbatim}
POST ./base/customers/1234/objects/
Payload:
{
    "externalId": "O1234-0001"
    ...
}
\end{verbatim}

\(^1\) Compare http://en.wikipedia.org/wiki/REST
Here the property reference with value O1234-0001 is the SAP Networked Logistics Hub internal ID of a SAP Networked Logistics Hub:device entity. The externalId is an SAP Networked Logistics Hub identifier that is repeatedly used in case of communication problems until telematics data broker returns a response indicating a successful creation of the object.

The telematics data broker then returns a customer payload comprising the default property `internalId`:

Response:

```json
{
  "internalId": "OT1-abcd",
  "externalId": "O1234-0001",
  ...
}
```

To update this object, SAP Networked Logistics Hub then invokes

PUT ./api/customers/1234/objects/OT1-abcd

Whereas, when telematics data broker invoke SAP Networked Logistics Hub to notify of an individual device (= ConnectedCar::Object) location, it refer to the same entity via the reference

POST ./devices/O1234-0001/locations

Identifier handling:

- As a general principal, the client side internal ID is passed as `externalId` property on the entity.
- When a new entity instance is created on the server side, the entity is returned in the response. The identifier is then returned in property `internalId`.
- For business partner identifiers, a property called `standardId` is introduced where the value of the standard identifier by convention is the DUNS number of business partner.

### 3.2 SAP Networked Logistics Hub Integration with Telematics Data Broker

The SAP Networked Logistics Hub version 1.0 obtains the vehicle location information via TDB, which has the following capabilities:

- Provide vehicle, i.e. device positions by integrating with several telematics services and by providing own mobile solution.
- Enable text-based communication with vehicles and devices, e.g. for exchanging traffic and order related messages.
- Provide a service calculating vehicle arrival times (called Estimated Time of Arrival service or ETA).

In SAP Networked Logistics Hub, the telematics data provider acts as telematics data broker that acquire the vehicle locations on-behalf of a freight forwarder, and then share the location information (and more) with SAP Networked Logistics Hub.

In SAP Networked Logistics Hub, the freight forwarders are so-called business partners/organizations. Before SAP Networked Logistics Hub can obtain vehicle or device locations from the telematics data
broker (TDB), SAP Networked Logistics Hub must register the business partner with the TDB as customer. A freight forwarder can use SAP Networked Logistics Hub to track the freight forwarder’s vehicle locations. For this, respective on-board units or mobile devices must be registered at the TDB under the respective customer.

Often freight forwarders have a fleet with vehicles of different manufacturers such as the MAN, Daimler, and Volvo. The vehicles’ on-board units are then connected to different telematics data services such as the MAN telematics and Daimler Fleetboard. In addition, the telematics data provider operates a vendor independent service through a mobile solution called Logiweb.

A freight forwarder with vehicles from several vendors can see all telematics enabled vehicles in SAP Networked Logistics Hub by allowing a broker obtain location data from all the respective services. A device has an assigned telematics data broker and an assigned telematics data service account.

![Figure 2: Freight Forwarder and Telematics Data Service Provider (SAP Networked Logistics Hub Data Model)](image)

Example

- FFWD1 is freight forwarder using SAP Networked Logistics Hub
- MAN Telematics, Daimler Fleetboard, and Logiweb are all telematics data services

A typical registration flow is as follows:

- SAP Networked Logistics Hub registers FFWD1 as customer on the telematics data broker
- SAP Networked Logistics Hub calls telematics data provider wallet maintenance on the telematics data broker
  - User of FFWD1 maintains FFWD1@Fleetboard telematics data provider account in the wallet
  - User of FFWD1 maintains FFWD1@MAN Telematics telematics data provider account in the wallet
- SAP Networked Logistics Hub calls the telematics connected car to create device FFWD1 MAN Truck 1 referring to the FFWD1@MAN Telematics telematics data provider account
- SAP Networked Logistics Hub calls the telematics data provider connected car to create device FFWD1 MAN Truck 2 referring to the FFWD1@MAN Telematics telematics data provider account
- SAP Networked Logistics Hub calls the telematics data broker to create device FFWD1 Daimler Truck 1 referring to the FFWD1@Fleetboard telematics data provider account.

From then on, the telematics data provider push the periodic updates of device locations to SAP Networked Logistics Hub.

The following figure depicts the registration sequence:

![Diagram of registration sequence]

**Figure 3: Freight Forwarder Registration at Telematics Data Broker**

### 3.2.1 Telematic Data Broker: Customer Management

The SAP Networked Logistics Hub invokes this API to query, create, update, and delete the freight forwarder as a customer.
3.2.1.1 Method Overview

The following methods are provided by the telematics data broker:

- GET /services-ext/scl/customers/
- GET /services-ext/scl/customers/<customerID>
- PUT /services-ext/scl/customers/<customerID>
- DELETE /services-ext/scl/customers/<customerID>

The customer is created indirectly via the customerRegistration entity.

- POST /services-ext/scl/customerRegistrations/

3.2.1.2 Register New Customer on Telematic Data Broker Service

The purpose of this method is registration of a freight forwarder at a telematics data broker.

**Pre-condition**: The customer is created on SAP Networked Logistics Hub and not successfully created on the telematics data provider side.

**Post-condition:**

- The customer is created on the telematics data provider account. The customer onboarding on the telematics data broker starts. (The customer account is not activated immediately).
- The SAP Networked Logistics Hub has received the external ID of the customer on the telematics data provider.

For details on payload, refer section 1.1.1.1 Register New Customer on Telematics Data Broker Service in the Payload Information document.

3.2.1.3 Telematic Data Providers (TDP)

The SAP Networked Logistics Hub invokes this API to maintain Wallet at TDB and query device list for individual freight forwarder as customer.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

3.2.1.4 Lists of TDPs agreed by Telematic Data Broker

The agreed list of Telematic Data Providers that is supported by TDB is as given below:
### 3.2.1.5 TDP Account Management – Wallet Maintenance

The purpose of this method is to get Wallet Maintenance URL for a given customer registered with the SAP Networked Logistics Hub/TDB. The UI of SAP Networked Logistics Hub helps the user to open the given Telematic Data Broker wallet URL. Opening the URL returned using this method helps the user of the customer to maintain his credentials for individual Telematics Data Provider account

**Pre-condition:** The customer is created on the TDB.
Check: The customer is activated (No assumption).

For details on payload, refer section 1.1.1.2 in the *Payload Information* document.

### 3.2.1.6 Getting List of Devices for given Telematics Data Provider

When a truck is created in SAP Networked Logistics Hub, it must be associated with the appropriate mobile device or a telematics device (onboard device). Once the TDB wallet is maintained, it is possible to get the list of devices for or from a given Telematics Data Provider, as owned by the customer.

**Pre-condition:** The customer is created on the TDB. Check: The customer is activated (No assumptions). The wallet is maintained for the given telematics data provider.

For details on payload, refer section 1.1.1.3 in the *Payload Information* document.

### 3.2.2 Telematic Data Broker: Device Management

The SAP Networked Logistics Hub invokes this API to query, create, update, and delete devices for the individual freight forwarder as a customer.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>
3.2.2.1 Method Overview
The telematics data broker provides the following methods:

- POST /services-ext/scl/customers/<customerID>/objects
- PUT /services-ext/scl/customers/<customerID>/objects/<deviceID>
- DELETE /services-ext/scl/customers/<customerID>/objects/<deviceID>

3.2.2.2 Device Registration
Once a vehicle or truck is created in SAP Networked Logistics Hub with an appropriate device assignment (onboard device/mobile device), then TDB start receiving device positions from the device. Subsequently, the TDB posts the device positions to SAP Networked Logistics Hub. For TDB to recognize a telematics device or mobile device, corresponding trackable object has to be stored in TDB system.

For details on payload, refer section 1.1.2.1 in the Payload Information document.

3.2.2.3 Change Device Status
Once a device is registered in TDB, the status of the device can be changed to Inactive or Active. The TDB tracks the devices positions of an active device only.

For details on payload, refer section 1.1.2.2 in the Payload Information document.

3.2.2.4 Delete Device
Once a device is registered in TDB, if the device is no longer in use, it can be deleted in the TDB.

URL:
DELETE /services-ext/scl/customers/<customerID>/objects/<deviceID>

Explanation:
- In the URI, the <customerId> is the customer internal ID of the TDB.
- internalId is the ID of the device in TDB. The SAP Networked Logistics Hub maintains a mapping between the SAP Networked Logistics Hub-ID and TDB-ID.

3.2.3 Geofence Replication
The SAP Networked Logistics Hub invokes this API to create, update, and delete geofences for the Hub Manager/Freight Forwarder/Parking Space Operator/r/Enterprise Logistics Hub as a customer.
An important use case for geo-fences is sending of location-based messages to the vehicles. For example, a driver is informed of a traffic situation when approaching a location. This is realized by associating traffic messages to geo-fences that are sent to a device when it enters the geo-fence. The directional semantics are enabled by stacking the geo-fences. An event approaching from the south is enabled by defining an event entering the geo-fence 1 when already in geo-fence 2.

Figure 4: Geofences with Gates

The geo-fences are defined as polygons where edges are points encoded as geo-locations with latitude and longitude coordinates.

The following picture shows a Google Map view near the SAP headquarters, Walldorf (Germany). An area is marked using a polygon (filled white area) and a pushpin is placed near the center of the parking space.
Figure 5: Geofence Example 1

The following content is from a kml file, defining a placemark **SAP Parkplatz Walldorf (Freifläche)** with a polygonal geo-fence specification (here indicated as element **LinearRing**).

```xml
<Placemark>
  <name>SAP Parkplatz Walldorf (Freifläche)</name>
  <styleUrl>#m_ylw-pushpin0</styleUrl>
  <Polygon>
    <tessellate>1</tessellate>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          8.643338264304115,49.29617059580877,0
          8.644797742981407,49.29523172357361,0
          8.646156589974394,49.29622993923193,0
          8.645479113706948,49.29660129256203,0
          8.643632294320909,49.29636867090505,0
          8.643338264304115,49.29617059580877,0
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>
```
We use Geojson\(^2\) for representing the geo-fences. Geo-fences are represented as polygons. The positions of the edges are represented as longitude; latitude coordinates using WGS84 as the reference coordinate system. This is also the default in Geojson, which also defines the coordinate order as longitude and latitude (whereas GPS systems often use latitude, longitude!)

The above linear ring is represented as feature of type *Polygon*:

```json
{  "type": "Polygon",  "coordinates": [   [ // longitude, latitude   [8.643338264304115, 49.29617059580877],   [8.644797742981407, 49.29523172357361],   [8.646156589974394, 49.29622993923193],   [8.645479113706948, 49.2960129256203],   [8.643338264304115, 49.29617059580877] ] ]}
```

The corresponding geo-fence in SAP Networked Logistics Hub is defined as follows:

```json
{   "externalId":"uuid1",   "internalId":"GF1",   "reference":"SAP-GF1",   "name":"SAP Parkplatz Walldorf",   "type":"1",   "area":{      "type":"Polygon",      "coordinates":[       [8.643338264304115, 49.29617059580877],       [8.644797742981407, 49.29523172357361],       [8.646156589974394, 49.29622993923193],       [8.645479113706948, 49.2960129256203],       [8.643338264304115, 49.29617059580877] ] ]}
```

The geo-fence has attributes *identification* and *description* of the geo-fence. Type 1 refers to a simple geo-gence and type 2 to a geo-fence with *gates* for handling the directional semantics.

**Note:** *ExternalId* is not part of the geo-fence data. All external ID are IDs of the SAP Networked Logistics Hub system. All internal IDs are IDs of the telematics system.

---

\(^2\) See [http://wiki.geojson.org/GeoJSON_draft_version_6](http://wiki.geojson.org/GeoJSON_draft_version_6)
The following example shows the same geo-fence (SAP Parkplatz Walldorf) with one gate geo-fence, highlighted as green area.

![Image of SAP Parkplatz Walldorf with geofence highlighted]

**Figure 6: Geofence Example 2**

The corresponding JSON representation is as follows:

```json
{
    "internalId":"GF2",
    "externalId":"SAP-GF2",
    "name":"SAP Parkplatz Walldorf - with 1 Gate",
    "type":"2",
    "area":{
        "type":"Polygon",
        "coordinates": [ [8.64338264304115,49.29617059580877], [8.644797742981407,49.29523172357361], [8.646156589974394,49.29622993923193], [8.645479113706948,49.29660129256203], [8.64338264304115,49.29617059580877] ]
    },
    "gates": [ {
        "internalId":"GF2-gate1",
        "externalId":"SAP-GF2-gate1",
        "name":"West Gate",
        "type":"1",
        "area":{
            "type":"Polygon",
            "coordinates": [ [8.64462157476677,49.29515190896578], [8.644954785317921,49.29537391641546], [8.643715681972512,49.29619173425356] ]
        }
    ]
}
```
3.2.3.1 Method Overview

- **POST** /services-ext/scl/customers/<customerId>/geofences/
- **PUT** /services-ext/scl/customers/<customerId>/geofences/<geofenceId>
- **DELETE** /services-ext/scl/customers/<customerId>/geofences/<geofenceId>

The POST and PUT methods use the data type, **Geofence**. The data type, *definition* provided further in the document also has an example payload.

3.2.3.2 Create Geofence

**POST** /services-ext/scl/customers/<customerId>/geofences/

The purpose of this method is to create a new instance of a geofence defined for a customer.

*Request payload data type*: **Geofence**

*Response payload data type*: **Geofence**

3.2.3.3 Change Geofence

**POST** /services-ext/scl/customers/<customerId>/geofences/<geofenceId>

The purpose of this method is to change an existing geofence defined for a customer.

*Request payload data type*: **Geofence**

*Response payload data type*: **Geofence**

3.2.3.4 Delete Geofence

**DELETE** /services-ext/scl/customers/<customerId>/geofences/<geofenceId>

The purpose of this method is to delete an instance of an existing geofence for a customer.

3.2.4 Container Terminal Replication

The SAP Networked Logistics Hub invokes this API to create, edit, and delete container terminals for the Container Terminal Operator as a customer. An important use case is defining a container terminal as a point of interest in the *Traffic Status* application depending on the area of business.

The difference between a container terminal and depot: container depot deals with empty containers whereas a container terminal deals with loaded containers and the order to be serviced.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
</tbody>
</table>
3.2.4.1  Get Terminal Types
GET /services-ext/spl/terminalTypes

Response Content Type: application/json; charset=UTF-8

Response:
[
{
   "terminalType": "HCS1-proxy-name",
   "description": "HCS"
},
{
   "terminalType": "EG1-proxy-name",
   "description": "Eruogate"
}
]

3.2.4.2  Get Terminals of Terminal Type
GET /services-ext/spl/terminalTypes/<terminalType>/terminals

Response Content Type: application/json; charset=UTF-8

Response:
[
{
   "terminalType": "terminal-type",
   "terminalId": "terminal-id-number",
   "description": "terminal-description"
}
]

3.2.4.3  Create Point of Interest
POST /services-ext/spl/customers/<customerId>/pois

Request Content Type: application/json; charset=UTF-8

Request Payload:
{
   "externalId": "scl-id",
   "name": "POI1",
   "address1": "address1",
   "address2": "address2",
   "city": "Spb",
   "state": "RU",
}
"country": "Russia",
"position": {
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      125.6,
      10.1
  ]
  }
},
"posManual": true,
"categories": [ 
  "category1"
],
"standardId": "HCS1{59236278-FC0A-4F7C-8E2A-19057B29A1DE}"}

→ Here the category identifies the POI as terminal

Response Content Type: application/json;charset=UTF-8

Response Header: redirection URL  
https://<telematics-data-broker-cloud-hostname>/services-ext/poi/pois/HCS1{59236278-FC0A-4F7C-8E2A-19057B29A1DE}

Response Data:

{
  "externalId": "scl-id",
  "internalId": "P1234",
  "name": "POI1",
  "address1": "address1",
  "address2": "address2",
  "city": "Spb",
  "state": "RU",
  "country": "Russia",
  "position": {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [
        125.6,
        10.1
      ]
    }
  }
}
},
  "posManual": true,
  "categories": [
    "category1"
  ],
  "standardId": "HCS1{59236278-FC0A-4F7C-8E2A-19057B29A1DE}"
}

3.2.4.4 Update Point of Interest

**PUT** /services-ext/spl/customers/<customerId>/pois/<connected-car-poi-standard-id>

**Request Content Type:** application/json;charset=UTF-8

**Request Payload:**

```json
{
  "externalId": "scl-id",
  "name": "POI1",
  "address1": "address1",
  "address2": "address2",
  "city": "Spb",
  "state": "RU",
  "country": "Russia",
  "position": {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [125.6, 10.1]
    }
  },
  "posManual": true,
  "categories": [
    "category1"
  ],
  "standardId": "HCS1{59236278-FC0A-4F7C-8E2A-19057B29A1DE}"
}
```

→ Here the category identifies the POI as terminal

**Response Content Type:** application/json;charset=UTF-8

**Response Header:** redirection URL
Response Data:
{
    "externalId": "scl-id",
    "internalId": "P1234",
    "name": "POI1",
    "address1": "address1",
    "address2": "address2",
    "city": "Spb",
    "state": "RU",
    "country": "Russia",
    "position": {
        "type": "Feature",
        "geometry": {
            "type": "Point",
            "coordinates": [
                125.6,
                10.1
            ]
        }
    },
    "posManual": true,
    "categories": [
        "category1"
    ],
    "standardId": "HCS1{59236278-FC0A-4F7C-8E2A-19057B29A1DE}"
}

3.2.4.5 Delete Point of Interest
DELETE /services-ext/spl/customers/<customerId>/pois/<connected-car.poi.standard-id>

3.2.5 Availability of Automatic Status of Container Terminal

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

GET /services-ext/spl/terminals/status

Response Content Type: application/xml;charset=UTF-8
Response Header:
The mapping is as follows:
1: free
2: partially blocked
3: blocked

### 3.2.6 Address Resolution

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

#### 3.2.6.1 Geocoding

Returns a geo position information out of a given address information specified either in detail or in a search query representation

Sample requests:

- [https://ws41.caritc.de/services-ext/geoCoder/geoloc?provider=osm&query=Offenbachstra%C3%9Fe%203%2CNeu-Ulm%2CGermany&limitCountries=de%2Cjp&language=en](https://ws41.caritc.de/services-ext/geoCoder/geoloc?provider=osm&query=Offenbachstra%C3%9Fe%203%2CNeu-Ulm%2CGermany&limitCountries=de%2Cjp&language=en)

- [https://ws41.caritc.de/services-ext/geoCoder/geoloc?provider=osm&houseNumber=3&street=Offenbachstra%C3%9Fe&city=Neu-](https://ws41.caritc.de/services-ext/geoCoder/geoloc?provider=osm&houseNumber=3&street=Offenbachstra%C3%9Fe&city=Neu-)
Ulm&county=&state=Bavaria&country=Germany&postalCode=89231&limitCountries=de%2Cjp&language=en

Sample response:
HTTP/1.1 200 OK
{
    "lat": 48.4029402,
    "lon": 10.0181523
}

3.2.6.2 Reverse Geocoding

Returns an address information to a given geo position information.
Following is the address resolution service format

Sample requests:
- https://ws41.caritc.de/services-ext/geoCoder/geoloc?provider=osm&query=Offenbachstr%C3%A4%C3%9Fe%203%2CNeu-Ulm%2CGermany&limitCountries=de%2Cjp&language=en
- https://ws41.caritc.de/services-ext/geoCoder/geoloc?provider=osm&houseNumber=3&street=Offenbachstr%C3%A4%C3%9Fe&city=Neu-Ulm&county=&state=Bavaria&country=Germany&PostalCode=89231&limitCountries=de%2Cjp&language=en

Sample response:
HTTP/1.1 200 OK
{
    "lat": 48.4029402,
    "lon": 10.0181523
}

Sample request:

Sample response:
HTTP/1.1 200 OK
{
    "latitude": 48.40048195,
    "longitude": 9.98337900958972,
    "displayName": "Hauptbahnhof, Bahnhofplatz, Fischerviertel, Weststadt, Arbeitersiedlung Untere Bleiche, Ulm, Regierungsbezirk Tübingen, Baden-Württemberg, 89073, Deutschland",
    "address": {
        "houseNumber": "",
        "road": "Bahnhofplatz",
        "suburb": "Weststadt",
        "city": "Ulm",
        "county": ""
    }
}
3.3 Receiving Device Positions in SAP Networked Logistics Hub

The TDB invokes this SAP Networked Logistics Hub API to publish the device position information.

<table>
<thead>
<tr>
<th>Provider</th>
<th>SAP Networked Logistics Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Telematic Data Broker</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

For details on payload, refer section 1.2 in the *Payload Information* document.

### 3.3.1 Sending Messages from SAP Networked Logistics Hub through TDB

The SAP Networked Logistics Hub invokes this TDB API to send messages to the drivers or geofences for Hub Manager/Freight Forwarder/Parking Space Operator/Enterprise Logistics Hub as the customer.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
<tr>
<td>Payload data format</td>
<td>The POST method uses data type <strong>Message</strong>.</td>
</tr>
</tbody>
</table>

### 3.3.1.1 Sending Messages to Device

For details on payload, refer section 1.2.1.1 in the *Payload Information* document.

### 3.3.1.2 Sending Messages to Geofences

For details on payload, refer section 1.2.1.2 in the *Payload Information* document.

### 3.3.2 Receiving Messages Driver/Device through TDB in SAP Networked Logistics Hub

The TDB invokes this SAP Networked Logistics Hub API to send messages from the drivers/devices to the freight forwarder as the customer.
3.3.3 Usage Log

The TDB invokes this SAP Networked Logistics Hub API to get SAP Networked Logistics Hub/Enterprise Logistics Hub usage log for a single freight forwarder as the customer.

For details on payload, refer section 1.2.2 in the Payload Information document.

<table>
<thead>
<tr>
<th>Provider</th>
<th>SAP Networked Logistics Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Telematic Data Broker</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
<tr>
<td>Payload data format</td>
<td>The POST method uses the data type, Message</td>
</tr>
</tbody>
</table>

3.4 SAP Networked Logistics Hub and Traffic Information Service

3.4.1 SAP Networked Logistics Hub Integration with Telematic Data Broker (TDB)

The SAP Networked Logistics Hub invokes this API to receive Information about the active traffic Information from the Hub Manager system of Hamburg Port Administration and ADAC.

For payload details, refer 1.3.1 in the Payload Information document.

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http</td>
</tr>
<tr>
<td>Data representation</td>
<td>XML – corresponds to DATEXII format as explained above</td>
</tr>
</tbody>
</table>

3.4.2 SAP Networked Logistics Hub Integration with HERE Map

The SAP Networked Logistics Hub invokes this API to receive Information about the active traffic Information from HERE Map. The use of the traffic API is to obtain traffic incident data for a specific area.

For payload details, refer 1.3.1 in the Payload Information document.
### SAP Networked Logistics Hub Integration – Order Management (Tour Integration in SAP Networked Logistics Hub)

The following figure shows an overview of the components used in the tour management:

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic</th>
<th>Data Representation</th>
<th>Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
<td>XML, JSON</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td>https</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**URL:**

https://traffic.cit.api.here.com/traffic/6.0/incidents

**Example 1:**

?app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}

**Example 2**

https://traffic.cit.api.here.com/traffic/6.0/incidents.json
?app_id={YOUR_APP_ID}
&app_code={YOUR_APP_CODE}
&quadkey=12020330
The freight forwarders or logistic service providers (LSP) generally use a transport order management solution for managing tours. The tours that are ready for execution can be replicated to SAP Networked Logistics Hub. The tour data may be complete with respect to driver and vehicle or incomplete if a LSP operates in a taxi mode, i.e. the assignment of driver to a tour is done late based on the driver/vehicle location.

The dispatcher of a freight forwarder can use SAP Networked Logistics Hub to assign a tour to a vehicle/driver. A tour becomes visible to a driver or vehicle once the tour has been replicated to the SAP Networked Logistics Hub and the assignment to driver/vehicle is done. The SAP Networked Logistics Hub then dispatches the tour information to the respective device(s) through TDB, which then exposes them to the mobile devices or on-board units.

A dispatcher or driver marks a tour as active. The SAP Networked Logistics Hub then activates calculation of estimated time of arrival on the telematics data broker. As the tour comprises all required location information and waiting times as tour stops, the estimated time of arrival is calculated per tour stop and based on the tour assigned to the device.

The following figure depicts the SAP Networked Logistics Hub tour data model:
A tour or freight order is defined as a sequence of stops a driver must take with load and unload items and relevant details such as container numbers. The loaded/unloaded goods are defined in the item by referring to a freight unit. The tour item comprises freight unit details as part of the tour data.

The freight forwarder owns a tour. It can be pre-assigned to a planned vehicle or pool. When the tour is activated it must be assigned to exactly one vehicle (pending: or driver) acting as the “servicing vehicle.” Once this is done, the order is dispatched to the mobile device.

An order management system can propose events on the tour, stop, and item level. This can then be triggered by the driver/device and relayed to the order management system via the SAP Networked Logistics Hub.

The following same tour provides an example for picking up a container at the SAP Head Quarters, Walldorf and dropping it off in SAP Service Center, Rot. Each location is identified by an address and must be complemented by a geo-location (required for the ETA service).

### 3.5.1 Order Replication from Order Management System to SAP Networked Logistics Hub through Telematics Data Broker

<table>
<thead>
<tr>
<th>Provider</th>
<th>Order Management System via Telematics Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
</tbody>
</table>
3.5.2 Order Replication SAP Networked Logistics Hub to Telematics Data Broker

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http</td>
</tr>
<tr>
<td>Data representation</td>
<td>XML – corresponds to DATEXII format as explained above</td>
</tr>
</tbody>
</table>

3.5.2.1 Method Overview

POST /services-ext/scl/customers/{customerId}/tours/
PUT /services-ext/scl/customers/{customerId}/tours/{tourInternalId}/status/
DELETE /services-ext/scl/customers/{customerId}/tours/{tourInternalId}/

Compare above example for reference.

3.5.3 Event Notifications Telematics Data Broker to SAP Networked Logistics Hub

<table>
<thead>
<tr>
<th>Provider</th>
<th>SAP Networked Logistics Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Telematic Data Broker</td>
</tr>
<tr>
<td>Protocol</td>
<td>http</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

For details on payload, refer section 1.4.2 in the Payload Information document.

3.5.3.1 Receiving Tour Events

URL: /sap/scl/xs/integration/t1/inbound/{t-systems-customer-id}/tours/{t-systems-tour-id}/events

POST Request data:
{
  "data": [
    {
      "scope": 1,
      "eventCode": "com.sap.scl.Accept",
      "utcTime": "2014-02-20T13:10Z"
    },
    {
      "scope": 2,
      "tourStopInternalId": "T1-Tour208-Stop-ID#1",
      "eventCode": "com.sap.scl.LoadingBegin",
      "utcTime": "2014-02-20T16:10Z"
    }
  ]
}
Response:
As long as there is no error, there is no response. If there is any error, you will get Error [] object in the response.

The picture below shows typical status transitions than can take place when events arrive at SAP Networked Logistics Hub.

**Figure 9: Tour Events**

### 3.5.4 Sending Tour Status SAP Networked Logistics Hub to Telematics Data Broker

<table>
<thead>
<tr>
<th>Provider</th>
<th>Telematic Data Broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Networked Logistics Hub</td>
</tr>
<tr>
<td>Protocol</td>
<td>http</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

For details on payload, refer section 1.4.3 in the *Payload Information* document.

### 3.6 SAP Networked Logistics Hub Integration with SAP Yard Logistics

Generally, any logistic hub uses a software solution that simplifies and streamlines truck, container, and rail yard management. SNLH, a solution capable of producing track and trace data when integrated with SAP yard logistics enables efficient operations within the logistics yard.
The following diagram illustrates the integration scenario:

Figure 10: Integration Scenario

To integrate the expected time of arrival of vehicles and other vehicle details, SAP Networked Logistics Hub provides oData based RESTful APIs. The yard logistics invokes these APIs to fetch the details and update the yard document(s), as required.

<table>
<thead>
<tr>
<th>Provider</th>
<th>SAP Networked Logistics Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>SAP Yard Logistics</td>
</tr>
<tr>
<td>Protocol</td>
<td>http/REST</td>
</tr>
<tr>
<td>Data representation</td>
<td>JSON</td>
</tr>
</tbody>
</table>

3.6.1 OData Entity Data Model
An enterprise logistics hub represents the business entities that are responsible for either manufacturing or stocking and shipping of either raw materials or shipped goods. In SAP Networked Logistics Hub, the primary purpose is to ensure speed processing and movement of containers either carrying inbound or outbound orders. Entities are generally a person or partnership or business with which addresses or locations can be associated. In the context of SNLH, the entities are checkpoint, loading station, weigh bridge etc.

### 3.6.2 Metadata Url
- [https://<host>/snlhcore/v1/odata.svc/$metadata](https://<host>/snlhcore/v1/odata.svc/$metadata)

For more information, refer the OData Service Metadata Information document.

### 3.6.3 System Query Options
The following are the system query options:
- `$filter`
- `$orderby`
- `$top`
- `$skip`
- `$format`
- `$select`
- `$inlinecount`

### 3.6.4 Key Services

**Enterprise**
Hubs
- https://<host>/snlhcore/v1/odata.svc/Hubs?$format=json

Vehicles

Tours
- https://<host>/snlhcore/v1/odata.svc/Tours?$format=json

Stops
- https://<host>/snlhcore/v1/odata.svc/Stops?$format=json

For more information, refer section 3.7.11 Stop Type Values

Items
- https://<host>/snlhcore/v1/odata.svc/Items?$format=json

Entity
- https://<host>/snlhcore/v1/odata.svc/Entity?$format=json

Hubs within an Enterprise
- https://<host>/snlhcore/v1/odata.svc/Enterprises('<EnterpriseUUID>')/Hubs?$format=json

Hub Entities within Hub
- https://<host>/snlhcore/v1/odata.svc/Hubs('<HubUUID>')/EntityDetails?$format=json

Inbound Tours to a Hub
- https://<host>/snlhcore/v1/odata.svc/Hubs('<HubUUID>')/TourDetails?$format=json

Vehicle within Hub
- https://<host>/snlhcore/v1/odata.svc/Hubs('<HubUUID>')/VehicleDetails?$format=json

Tour stops
- https://<host>/odata.svc/Tours('<TourUUID>')/StopDetails?$format=json

Stop Items
- https://<host>/v1/odata.svc/Stop('<StopUUID>')/ItemDetails?$format=json

### 3.7 Interface Data Type

#### 3.7.1 Address

```java
public class Address {
    String street;
}```
3.7.2 Communication

public class Communication {
    public String emailAddress;
    public String phone;
    public String fax;
    public String webURI;
}

3.7.3 Geofence

public class Geofence {
    public String externalId;
    public String internalId;
    public String name;
    public String type;
    public Geometry area;
    public Geofence[] gates;
}

Example: Simple (polygonal) Geofence:
{
    "transactionId": "uuid1",
    "internalId": "GF1",
    "externalId": "SAP-GF1",
    "name": "SAP Parkplatz Walldorf",
    "type": "1",
    "area": {
        "type": "Polygon",
        "coordinates": [
            [8.64338264304115, 49.29617059580877],
            [8.644797742981407, 49.29523172357361],
            [8.64615689974394, 49.2962299323193],
            [8.645479113706948, 49.29660129256203],
            [8.64338264304115, 49.29617059580877]
        ]
    }
}

Example: Geofence with one Gate:
{
    "internalId": "GF2",
    "externalId": "SAP-GF2",
}
"name":"Entering SAP Parkplatz Walldorf",
"type":"2",
"area": {
    "type":"Polygon",
    "coordinates": [
        [8.64338264304115, 49.29617059580877],
        [8.644797742981407, 49.29523172357361],
        [8.646156589974394, 49.29622993923193],
        [8.645479113706948, 49.29660129256203],
        [8.643338264304115, 49.29617059580877]
    ]
},
"gates": [
    "type":"Polygon",
    "coordinates": [
        [8.64462157476677, 49.29515190896578],
        [8.644954785317921, 49.29537391641546],
        [8.643715681972512, 49.29619173425356],
        [8.643430461688542, 49.2953839952883],
        [8.64462157476677, 49.29515190896578]
    ]
]

3.7.4 Object

Example
{
    "internalId":"TBD-internal-id",
    "externalId":"FFWD1-at-logiweb",
    "telematicDataProviderType":"Logiweb",
    "description":"FFWD1 Logiweb account",
}

3.7.5 Geometry

public class Geometry {
    public String type;
    public double[][] coordinates;
}

3.7.6 Contact

public class Contact {
    String internalId;
    String externalId;
}
String function;
String formOfAddress;
String givenName;
String surnamePrefix;
String surname;
Boolean main;

Example
{
  "externalId": "SAP-P1",
  "function": "Architect",
  "formOfAddress": "Mr.",
  "givenName": "freightforwarderadmin",
  "surname": "freightforwarderadmin",
  "main": true,
  "communication": {
    "emailAddress": "freightforwarderadmin@carriercompany.com",
    "phone": "+49 6227 747474",
    "fax": "+49 6227 757575",
    "webURI": "www.carrier.com"
  }
}

3.7.7 Message

public class Message {
  public String internalId;
  public String externalId; // client’s ID of the message when the message is created via POST
  public String replaceMessageId; // internalId of a message that shall be replaced
  public String inReplyToMessageId; // future use: the message I of a preceding message
                                      // to which a reply is sent
  public String originalMessageId; // future use: the message ID of the first message in a
                                     // thread (copy / forward set this ID if empty and keep it
                                     // if filled)
  public CommunicationParty[] recipients; // a list of receivers which are either a device
                                          // or a business partner (such as person or
                                          // organization)
  public CommunicationParty sender; // the sender identification, either a device
                                      // or a business partner
  public LocationBasedEvent[] triggers; // a list of location based event definitions
                                          // (typically entering of a geofence) that trigger
                                          // the delivery of the message to a device
                                          // that raises such an event
  public String serviceId;
  public String[] subServiceIds;
  public String expirationDateTime; // timestamp without millisecond
  public MultilingualText[] texts; // message text - details see there
  public String uri; // typically a www address where more information can be found
  public int priority; // a priority where 1 means very high to 4 or 5 = low // tbd
  public String source; // a String indicating the origination of the message,
                        // e.g. "ADAC", "Hamburg Port Authority"
}
3.7.8 Communication Party

```java
public class CommunicationParty {
    public static final String COMMUNICATION_PARTY_TYPE_DEVICE = "1";
    public static final String COMMUNICATION_PARTY_TYPE_BUSINESS_PARTNER = "2";
    public static final String COMMUNICATION_PARTY_TYPE_BUSINESS_EMAIL = "3";
    public static final String COMMUNICATION_PARTY_TYPE_GROUP = "4";
    // in conjunction with deviceGroupId or allDevices
    public String communicationPartyType;

    public class businessPartner {
        public String internalId;
    }
}
```

3.7.9 Multilingual Text

```java
public class MultilingualText {
    // LanguageCode is based on the W3C "built-in" data type XSD:language.
    // The language code of LanguageCode is represented according to IETF RFC 3066.
    public static final String CONTENT_TYPE_TEXT_ASCII = "text/ascii"; // default
    public static final String CONTENT_TYPE_TEXT_HTML = "text/html"; // future use
    String language; // language is optional if a list comprises only one element,
    // language is unknown and no translation will be provided
    String text;
    String contentType; // "text/ascii" or "text/html"
    // if a text in HTML is provided the same text SHOULD also be provided as
    // text/ascii because many devices do not support HTML
    // String charSet; // optional, must be utf-8 or empty - charset is utf-8 by default
    Boolean original; // optional. In a list one entry may be marked as original
    // text where others are translations thereof
}
```

3.7.10 LocationBasedEvent

```java
{
    public static final TYPE_GEOFENCE = "1"; // Type geofence
    String type;
    String geofenceId;
    String gateId;
}
```

3.7.11 Stop Type Values

<table>
<thead>
<tr>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC0003</td>
<td>Container Terminal</td>
</tr>
<tr>
<td>LC0005</td>
<td>Stop</td>
</tr>
<tr>
<td>LC0007</td>
<td>Container Depot</td>
</tr>
<tr>
<td>LC0009</td>
<td>WareHouse</td>
</tr>
<tr>
<td>LC0011</td>
<td>Production Site</td>
</tr>
<tr>
<td>LC0013</td>
<td>User Defined</td>
</tr>
<tr>
<td>LC0014</td>
<td>Loading Station</td>
</tr>
<tr>
<td>LC0015</td>
<td>Check Point</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>LC0016</td>
<td>Dangerous Goods</td>
</tr>
</tbody>
</table>

- Container Terminal - LC0003
- Stop - LC0005
- Container Depot - LC0007
- WareHouse - LC0009
- Production Site - LC0011
- User Defined - LC0013
- Loading Station - LC0014
- Check Point - LC0015
- Dangerous Goods - LC0016
4 Deployment and Operations

1.1 Deployed Component Structure and Deployment Options

![Diagram of SAP Networked Logistics Hub](image)

**Figure 10: SAP Networked Logistics Hub Software Component Structure**

1.2 System Landscape

![Diagram of Typical Production System Landscape](image)

**Figure 11: Typical Production System Landscape**
5 Restrictions and Hints

For information about restrictions, refer RIN 2094629.