Open Communication Interface for Road Traffic Control Systems
Offene Schnittstellen für die Straßenverkehrstechnik

OCIT Outstations Car Roadside Unit (RSU)

OCIT-O_Car_Data_V1.0_D02

OCIT Developer Group (ODG)

OCIT Outstations Car
Roadside Unit

Document: OCIT-O_Car_Data_V1.0_D02

Issued by: OCIT Developer Group (ODG)

Contact: www.ocit.org

Table of contents

Document history

Version Status	Distribu-tion list	Date	Comment
V1.0_D01	ODG internal	2016-01-10	Draft for OCIT-O AG5
		2016-02-04	Own sat object instead of MW task
		2016-02-04	MAP data should be transmitted as XML block object. Decoding takes place via associated xsd
V1.0_D02		2016-02-11	Changes accepted by AG5
V1.0_E01	PUBLIC	2016-03-30	ODG Homepage
V1.0_D02	ODG internal	2017-12-28	4.3 Risk message, new
		2018-01-30	Beta version (draft status)
		2018-02-02	Correction: avt-stoye
		2018-03-12	4.3.1, 4.3.3. Correction of the documentation based on xml Specifications: KD V3.0
V1.0 A01	PUBLIC	2018-03-15	For OCIT-O Car V1.0 ODG Homepage

Specifications

The **OCIT Outstations configuration document OCIT-O KD V3.0** contains an overview of all the specifications whose copyrights are managed by the ODG and arranges versions and revision levels according to:

- associated specifications of the interface "OCIT outstations for traffic signal controllers" with reference to the corresponding OCIT-C specifications
- gives information on the use of the transmission profiles and
- provides an overview of packages of specifications for interfaces for the use of which a nominal fee is required by ODG

The current issue of the document is published on www.ocit.org.

1 Introduction

All of the functions for the OCIT-O Car interface relevant between a control center are defined in this document. The outstandingly new function OCIT-O Car V1.0 is the possibility of using Car2X data from a roadside unit (RSU) for traffic detection.

1.1 Supported functions

The OCIT-O Car interface for traffic control in this version is based on the aforementioned reference specifications.

An OCIT-O Car interface can use different transmission profiles that are set in the optional definitions.

It is not mandatory that equipment operated on OCIT-O Car support all the functions defined in the reference specifications. They only support those functions that are necessary for the relevant purpose and design.

So, for example, control data for a traffic signal controller or layouts are not supported. The unavailability of a feature called up by the control central must bring about a recognizable response (return code) in the roadside unit.

The specifications of the OCIT-O Car interface version 1.0 are compatible with control centers with OCIT-C version 2.0

Functions in OCIT-O Car V1.0:

- Status information
- PT prioritization
- Risk messages

2 Devices and system functions

In this section you will find definitions that are necessary for operating an RSU.

The following definitions also apply to a traffic signal controller with RSAP.

2.1 RSU with OCIT-O Car interface

Due to the time behavior of the OCIT Outstations protocol, OCIT-O Car RSUs are intended specifically for use in systems with a decentralized structure. They can detect and provide traffic information values. They have the following characteristic properties:

- They have powerful microprocessors that locally carry out the processing of Car2X data.
- They have accurate clocks whose time is used for labeling events.
- OCIT-O Car is not suitable for operating a TSC.

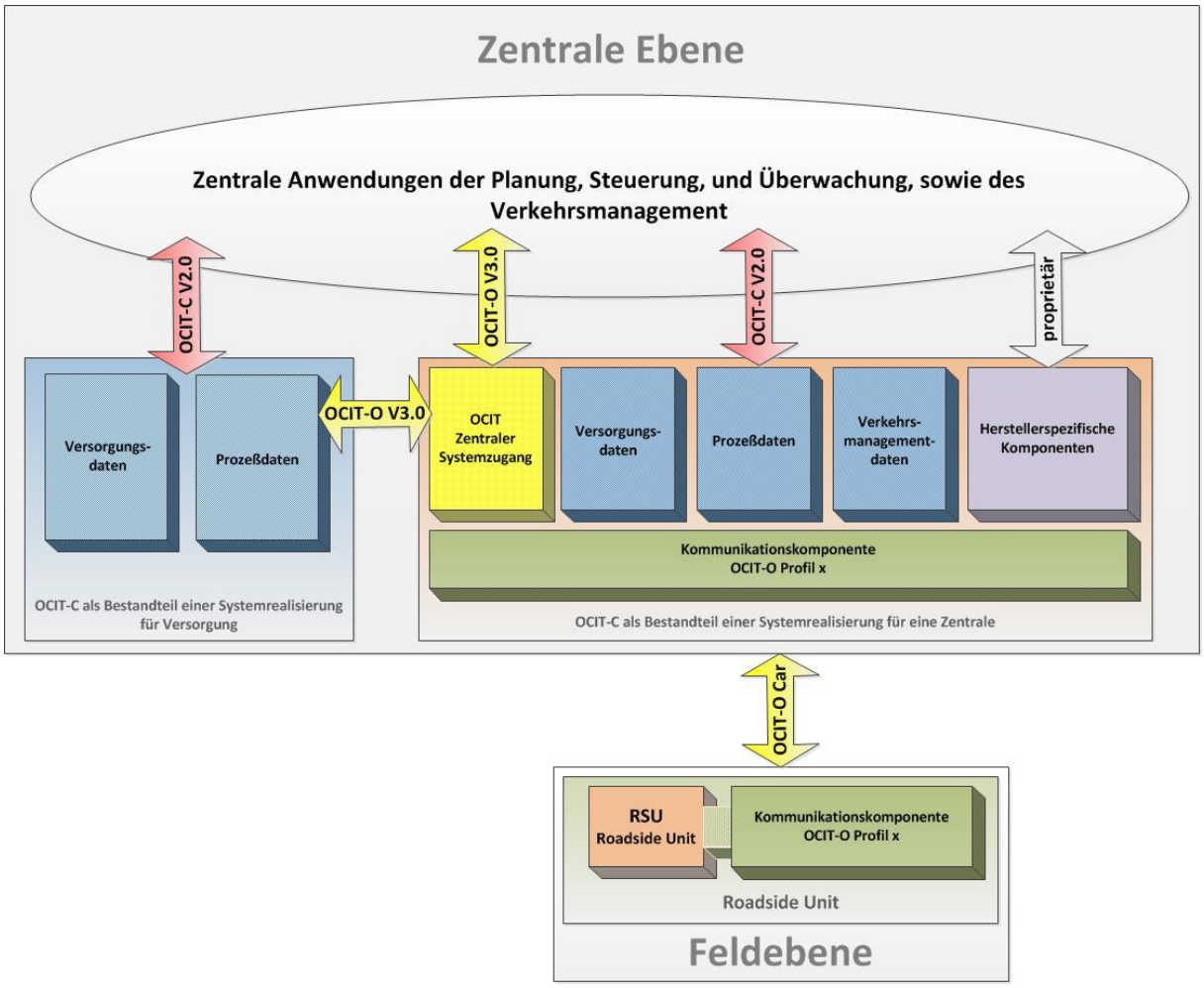


Figure 1 Roadside Unit with interface OCIT-O Car

2.2 Traffic signal controller with RSAP

The traffic signal controllers with a roadside access point (RSAP) connect to the OCIT control center via an OCIT-O V3.0 interface.

2.3 Transmission profile

For transmitting data from OCIT-O Car, only profile 3 or profile 4 is to be used.

3 Object definitions

Detailed descriptions of the data and methods of the objects can be found in the file OCIT-O_Car.xml associated with this document. Details that were not listed in the following object descriptions for reasons of simplicity can only be found there.

3.1 Archives

Selected operating data are collected in the archives of the RSUs. There are several archives in each controller. The data from the archives can be read out by the control center or via system access tools. Additionally, the control center can request data archived by the controller that are in certain positions or data that were recorded at particular times. During normal operation the archived data are collected by the control center upon occurrence of particular events. In the case of occurrence of such an event the controller sends an event telegram (does not contain the data) to the control center, which in turn can request individual or multiple data from the archives. Event telegrams can be triggered:

- when a set fill level of the archive has been reached,
- when entering certain variable values,
- when changing the target address for the event telegrams.

The archives of the controllers can be parameterized by the control center during operation. The following can be defined: size, type of tasks, events that lead to event telegrams, stop and enable collection of data, reset.

A detailed description of how to handle messages and measurement values can be found in the document OCIT-O Outstations basic functions for field devices (OCIT-O_Basis V3.0).

Archives defined for RSUs are described in section 3.2.

3.2 Archives of the RSU

The following archives are predefined in OCIT-O Car for each RSU:

- The **standard message archive (1)** contains messages regarding faults and other messages: OCIT main message + secondary message + message degree. The tasks for this are pre-defined and cannot be changed. The data stored in the archive are preserved after switching off the power supply. The messages in which Basis is included.
  - A **syslog archive (2)** for syslog messages (with text) and manufacturer-specific messages that are kept in persistent storage. The archive is available even in the basic configuration. The archive size is adjusted by the manufacturer to the other archives available in the controller. The data stored in the archive are preserved after switching off the power supply.
  - A **service system access archive (3)** for tasks concerning system access ways.
  - A **status archive (5)** for saving the operating status (RSUStateMsg messages). The operating statuses are collected each time the operating status changes. Any change in the operating status generates a RSUState entry in the status list. The tasks for this are pre-defined and cannot be changed. The data stored in the archive are preserved after switching off the power supply.
  - A **PT arcive (33)** for CAM-R09 telegrams
  - A **CAM archive (37)** for the status information of the vehicles
  - A **DENM archive (38)** for the risk messages

The minimum sizes of archives of OCIT-compatible RSUs are listed in the document Function Level OCIT-Outstations Version 3.0 for Traffic Signal Controllers (OCIT-O_V3.0_Funktionsspiegel).

3.2.1 Properties of the lists

List number	1 Standard message archive	2 Syslog	3 Service	5 Status	33 PT	37 CAM	38 DENM
Creating task possible?	No	Yes	Yes	No	Yes	Yes	Yes
Start/stop/reset the list possible?	No	Yes	Yes	No	Yes	Yes	Yes
Suspend/Unsuspend the list possible?	No	Yes	Yes	No	Yes	Yes	Yes
Pre-assigned persistence (None, Task, Tasks & Buffer)	Tasks & Buffer	Tasks & Buffer	Tasks & Buffer	Tasks & Buffer	None	None	None
Selection of persistence possible	No	No	No	No	Manufacturer-dependent	Manufacturer-dependent	Manufacturer-dependent
Pre-occupied state of the list (start, stop, suspend)	Start	Stop	Stop	Start	Stop	Stop	Stop
OverwriteOnFull active?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Size of the buffers changeable?	No	No	No	No	Yes	Yes	Yes
Predefined tasks (after reset)	0:Message task I 100:Message task W 2:Message task E 3:Message task F	0:Message task I	0:Message task I	RSUStateMsg 0:Message task I	0:Message task I	0:Message task I	0:Message task I

4 Car-2-X Communication

This section of the document describes all the objects for the OCIT-C Car interface between a control center and a Car2X-compliant roadside unit (RSU). The RSU can be part of a traffic signal system (TSC with RSAP) or be operated as a standalone unit.

The functions are based on the following Car2X messages

Cooperative Awareness Message (CAM)
Decentralized Environmental Notification Message (DENM)

This way the OCIT-O Car interface supports risk messages, quality assurance, statistics and messages from floating car data as well as prioritization of PT and special vehicles.

4.1 Status information

The data from the Car2X – communication should be used for a quality analysis and quality assurance of the intersections.

The vehicles send the status information from several sensors to the RSU via the CAM messages. Many data, such as the outside temperature, status of the brake booster or the angular position of the steering wheel are of no interest here. The data which are of interest for the traffic engineers are as follows:

- Time and date
- Position
- Speed
- Vehicle direction
- Vehicle type

The following values are provided for central analysis of the vehicle data (each in the area of the RSU):

- Average speed
- Travel time

4.1.1 Task MWAuftragSingleCar

The value MVAuftragSingleCarExt is used to display a vehicle within an road section. The generated frames contain data which describe the start and end point of the route and the travel time of the vehicle. The data are calculated from the CAM messages that were sent to the RSU. If the task is set, a frame is saved for each vehicle in the RSU area.

The task MWAuftragSingleCar is derived from the task object and generates MWAuftragFrameSingleCar frame types.

MWAuftragSingleCar (100:413)

MWAuftragSingleCar		
METHOD	**Name**	**Description**
119, 120, 121, 122	**ActivateEvent,** **AddElement,** **Start,** **Stop**	See seciton "Message and measurement value processes" in the "OCIT-O Basis" document.

Structure of the event frame (MWAuftragFrameSingleCar):

Name	**Data type**	**Comment**
StartPosition.Latitude	LONG Min=-900 000 000 Max=900 000 000 Nullval=900 000 001	Geolocation of the vehicle when entering the road section in 1/10 map scale
StartPosition.Longitude	LONG Min=-1 800 000 000	Geolocation of the vehicle when entering the road section in 1/10 map scale

	Max=1 800 000 000 Nullval=1 800 000 001	
EndPosition.Latitude	LONG Min=-900 000 000 Max=900 000 000 Nullval=900 000 001	Geolocation of the vehicle when leaving the road section
EndPosition.Longitude	LONG Min=-1 800 000 000 Max=1 800 000 000 Nullval=1 800 000 001	Geolocation of the vehicle when leaving the road section
StationType	UBYTE ENUM	Type of the vehicle (vehicle class) **(100:103 STATION_TYPE)**
TravelTime	USHORT Nullval=0xffff	Vehicle travel time
AverageSpeed	UBYTE Nullval=0xff	Average speed

Vehicle class classification acc. to (StationType)

Value	**Description**
0	Unknown, none of the other categories
1	Pedestrian
2	Bicycle
3	Moped
4	Motorbike
5	Passenger car
6	Bus
7	Van
8	Truck
9	Semi-trailer truck
10	Special vehicles
11	Tram

4.2 PT prioritization

Public transport vehicles and special response vehicles are able to request a prioritization using the Car2X communication CAM telegrams. These request telegrams can be registered With OCIT-O Car.

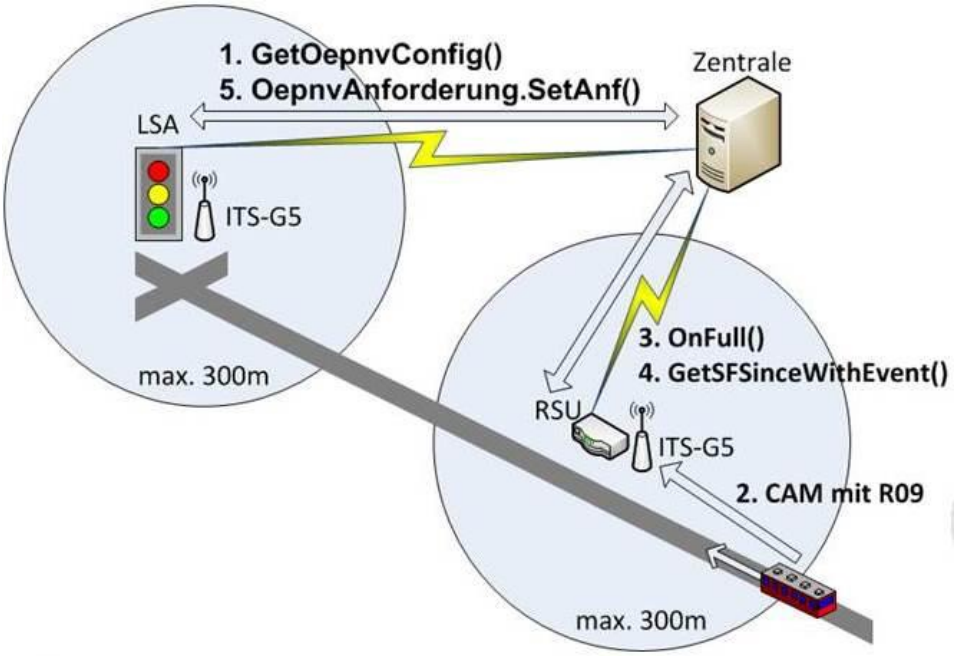


Figure 2: PT prioritization

4.2.1 Task for CAM-R09 telegrams

The task **MwAuftragCamR09 (100:1411)** for cam messages with R09 containers is the same in terms of selection for the task for R09 telegrams and only returns an advanced data set.

MwAuftragCamR09 is derived from MwAuftragR09.

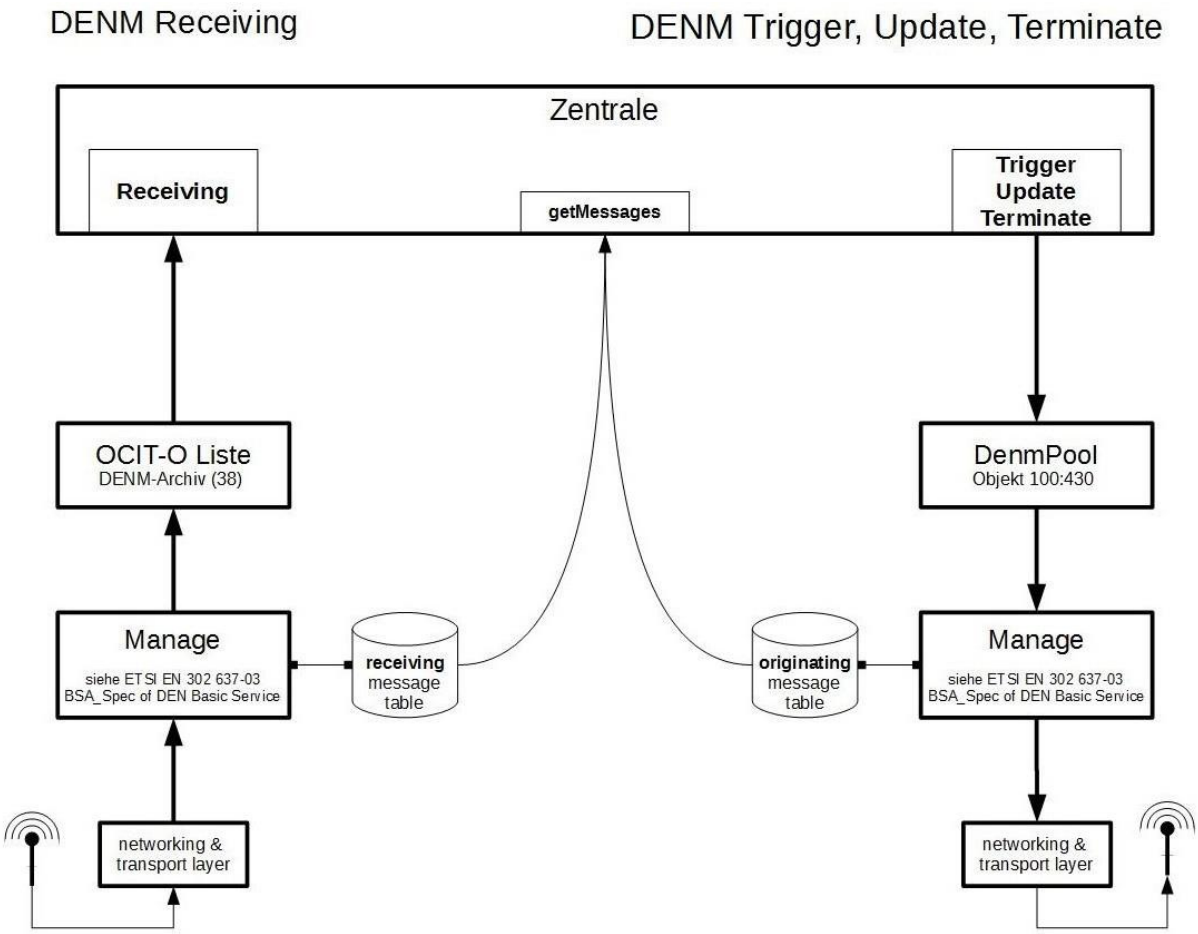
The advanced data structure looks as follows:

Name	**Abbreviations**	**Data type**	**Value range**	**Comments**
Day (Issue date)	DD	UBYTE	1 - 31	Date created / time
Month (Issue date)	MO	UBYTE	1 -12	Date created / time
Year (Issue date)	YY	UBYTE	0 - 99	Date created / time
Stunde	HH	UBYTE	0...23	Date created / time

Name	Abbrevi-ations	Data type	Value range	Comments
(Erstelldatum)				
Minute (Erstelldatum)	MM	UBYTE	0...59	Date created / time
Sekunde (Erstelldatum)	SS	UBYTE	0...59	Date created / time
Meldepunktnummer	MPN	LONG	1 - 2^24	5 characters in the telegram
Liniennummer	LLL	USHORT	0 - 999	3 characters in the telegram
Kursnummer	KK	UBYTE	0 - 99	2 characters in the telegram
Routennummer	RRR	USHORT	0 - 999	3 characters in the telegram
Priorität	P	UBYTE	0 - 7	1 character in the telegram
Zuglänge	Z	UBYTE	0 - 7	1 character in the telegram
Richtung Hand	H	UBYTE	0 - 3	1 character in the telegram; manual request by the driver (e.g. via the key switch at a stop)
Fahrplanabw (Sek)	FAHRP	SHORT (signed short)	-3599 bis 3599	"Schedule situation" Deviation from schedule as in the received R09 telegram.
StationID		ULONG		ID of the unit, from which the event was sent
StationType		UBYTE ENUM		Vehicle type of the unit, from which the event was sent **(100:103 STATION_TYPE)**
Position.Latitude		LONG		Geolocation of the vehicle when sending the event in 1/10 map scale
Position.Longitude		LONG		Geolocation of the vehicle when sending the event in 1/10 map scale
EmbarkationStatus		BOOL		Embarkation status (whether doors are open or not)

4.3 Risk messages

Risk messages can be triggered at the control center level, as well as at the field level, and can be distributed across the entire system. Generally, risk messages are used to increase the safety of traffic participants, to reduce CO2 emissions, and for statistical purposes.



4.3.1 Task for risk messages (DENM)

The task **MWAuftragDENM (100:440)** is used for detecting risk messages that have been received by the vehicles (ITS stations) within the range of an RSU. The frames generated contain all the data of a DENM.

Managing DENMs is described in detail in "ETSI EN 302 637-03". It is also defined there, when receiving DENMs, how you can identify whether they are new messages or just repeats. Moreover, things like the validity period and scheduling are described there.

The task MWAuftragDenm is defined from MWAuftragExtern and has the following task elements.

4.3.2 Management of risk messages (DENM)

The **object DenmPool (100:430)** provides methods with which the DENMs can be added (trigger), updated (update) and terminated (terminate). What's more, the method "getMessages" makes it possible to read out the DENMs from the message tables.

The term "message table" is based on the ETSI documentation and describes the buffers for the generated ("originating") and received ("receiving") DEMNs. (ETSI recommends storage in different tables here.)

ETSI also defines the optional "forwarding" function in addition to "originating" and "receiving". This function is currently not defined in OCIT-O.

4.3.3 Object DenmPool

DenmPool (100:430)

DenmPool		
METHOD	**Name**	**Description**
120	triggerMessage	Generates a new DENM for a newly detected event.
	Input parameters	
	Message	DENM for a newly detected event.
	TransmissionControl	Additional data to control the transfer
	Output parameters	
	RetCode	OK: is returned if the message was able to be added. EXISTS_ALREADY: is returned if the message cannot be added PARAM_INVALID: is returned if the message was not given in full.
121	updateMessage	Generates an update DENM for an update of the event.
	Input parameters	
	Message	DENM for an update of the event.
	TransmissionControl	Additional data to control the transfer
	Output parameters	
	RetCode	OK: is returned if the message was able to be updated. NOT_POSSIBLE: is returned if the message was not found PARAM_INVALID: is returned if the message was not given in full.

DenmPool		
METHOD	**Name**	**Description**
122	terminateMessage	Generates a cancellation or negation DENM for termination of the event.
	Input parameters	
	Message	DENM for termination of the event. (Normally without situation, location and alacarte container)
	TransmissionControl	Additional data to control the transfer
	Output parameters	
	RetCode	OK: is returned if the message was able to be removed.
125	getMessages	Provides all messages from the given table (originating, receiving, etc.)
	Input parameters	
	Type	Type of the message table (originating, receiving, ...)
	Output parameters	
	RetCode	OK: function has been executed correctly. PARAM_INVALID: wrong table-type.
	Messages.Anzahl	Number of the following USHORT messages
	Messages	Messages
	RetCode	OK: function has been executed correctly. PARAM_INVALID: wrong table-type.

References

ETSI TR 102 638:	"Basic Set of Applications; Definitions". Europe.
ETSI TS 102 637-3:	"Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service". Europe.
ETSI EN 302 637-03:	"Specifications of Decentralized Environmental Notification Basic Service (DENM)". Europe.

Figures

Glossary

The content of this section includes technical terms that refer to the context of this document. Terms present in all OCIT documents can be found in the document "OCIT – O Glossary" and "OCIT-O_System_V3.0".

OCIT-O_Car_Data_V1.0_D02