



**Intelligent Transport Systems (ITS);  
Vehicular Communications;  
Basic Set of Applications;  
Local Dynamic Map (LDM)**

---

**Reference**

DEN/ITS-0010005

---

**Keywords**

application, ITS, management, user

**ETSI**

---

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2014.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM®** and the GSM logo are Trade Marks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	5
Foreword.....	5
Modal verbs terminology.....	5
Introduction .....	5
1 Scope .....	7
2 References .....	7
2.1 Normative references .....	7
2.2 Informative references.....	7
3 Definitions and abbreviations.....	8
3.1 Definitions .....	8
3.2 Abbreviations .....	8
4 General description of a LDM.....	9
4.1 Functionality provided by the LDM.....	9
5 LDM functional specification .....	9
5.1 LDM requirements .....	9
5.1.1 LDM functional requirements.....	9
5.1.2 LDM other requirements .....	10
5.2 The LDM within the ITS-S communication architecture.....	10
5.3 LDM functional architecture .....	11
5.3.1 Function FN.LDM.1 - LDM Service .....	11
5.3.2 Function FN.LDM.2 - LDM Maintenance.....	12
5.4 Interfaces of the LDM .....	12
5.4.1 Interface IF.LDM.1 - management layer .....	12
5.4.2 Interface IF.LDM.2 - security layer.....	13
5.4.3 Interface IF.LDM.3 - LDM Data Providers.....	13
5.4.4 Interface IF.LDM.4 - LDM Data Consumers .....	13
6 LDM Interfaces .....	14
6.1 Interface IF.LDM.2 to the security layer.....	14
6.1.1 Authorization .....	14
6.1.1.1 Authorize messages.....	14
6.1.2 Revocation .....	15
6.1.2.1 RevokeAuthorization messages .....	15
6.2 Interface IF.LDM.3 to LDM Data Providers.....	16
6.2.1 Registration.....	16
6.2.1.1 RegisterDataProvider messages .....	16
6.2.2 Deregistration .....	17
6.2.2.1 DeregisterDataProvider messages.....	17
6.2.2.2 RevokeDataProviderRegistration message .....	17
6.2.3 Maintenance of Provider data .....	18
6.2.3.1 AddProviderdata messages .....	18
6.2.3.2 UpdateProviderdata messages.....	19
6.2.3.3 DeleteProviderdata messages.....	19
6.3 Interface IF.LDM.4 to LDM Data Consumers .....	20
6.3.1 Registration.....	20
6.3.1.1 RegisterDataConsumer messages .....	20
6.3.2 Deregistration .....	21
6.3.2.1 DeregisterDataConsumer messages .....	21
6.3.2.2 RevokeDataConsumerRegistration message.....	22
6.3.3 Data Request.....	22
6.3.3.1 RequestDataobjects messages .....	22
6.3.4 Subscription .....	23

6.3.4.1	SubscribeDataConsumer messages .....	23
6.3.5	Cancel subscription.....	25
6.3.5.1	UnsubscribeDataConsumer messages .....	25
<b>Annex A (normative):</b>	<b>Data request filtering.....</b>	<b>26</b>
A.1	Minimal syntax of data request filtering .....	26
A.1.1	Filter .....	26
A.1.2	Filter Statement .....	26
A.1.2.1	Comparison Operator.....	27
A.1.2.2	Reference Value.....	27
A.1.3	Logical operators for combining Filter Statements .....	27
A.2	Ordering Data Request Results .....	28
<b>Annex B (normative):</b>	<b>ITS LDM Interface messages specified in ASN.1 .....</b>	<b>29</b>
<b>Annex C (informative):</b>	<b>Bibliography.....</b>	<b>35</b>
History .....		36

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This final draft European Standard (EN) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS), and is now submitted for the Vote phase of the ETSI standards EN Approval Procedure.

The present document is a single part deliverable.

<b>Proposed national transposition dates</b>	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	18 months after doa

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

## Introduction

In cooperative Intelligent Transport Systems (ITS), the Local Dynamic Map (LDM) is a key facility supporting various ITS applications by maintaining the information on objects influencing or being part of ITS. The Local Dynamic Map therefore is relevant to the development of technical standards and specifications in order to ensure deployment and interoperability of cooperative systems and services described in the EC's ICT Standardization Work Programme [i.7].

The LDM is a facility within the ITS station facilities layer as defined in the ITS communication architecture given in EN 302 665 [i.1]. Cooperative Awareness Messages (CAMs) as defined in EN 302 637-2 [4] and Decentralized Environmental Notification Messages (DENMs) as defined in EN 302 637-3 [5] are important sources of data for the LDM.

Moreover the LDM will support the Basic Set of Applications (BSA) outlined in TS 102 637-1 [i.2] by providing plausible authorized, area related information in a time relevant manner. The BSA provides the application specific requirements for the LDM.

The following applications from the BSA are considered:

- Driving assistance - Cooperative awareness.
- Driving assistance - Road Hazard Signalling (see TS 101 539-1 [i.3]).
- Speed management.
- Cooperative navigation Location based services.
- Community services.
- ITS station life cycle management.

---

# 1 Scope

The present document defines functional behaviour associated with a Local Dynamic Map (LDM) for usage in an ITS station unit (ITS-SU). It specifies functions and interfaces supported by a LDM. These functions and interfaces provide secure access to the LDM to manage LDM data objects stored in a LDM. It defines LDM data objects for safety-related and Vehicle to Vehicle (V2V)-related applications.

---

## 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

### 2.1 Normative references

The following referenced documents are necessary for the application of the present document.

- [1] ETSI TS 102 860 (V1.1.1) (2011-05): "Intelligent Transport Systems (ITS); Classification and management of ITS application objects".
- [2] ISO/IEC 8824-1:2008: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".
- [3] ETSI TS 102 894-2: "Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary".
- [4] ETSI EN 302 637-2 (V1.3.0) (2013-08): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service".
- [5] ETSI EN 302 637-3 (V1.2.0) (2013-08): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service".

### 2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] ETSI EN 302 665 (V1.1.1) (2010-09): "Intelligent Transport Systems (ITS); Communications Architecture".
- [i.2] ETSI TS 102 637-1 (V1.1.1) (2010-09): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 1: Functional Requirements".
- [i.3] ETSI TS 101 539-1 (V1.1.1) (2013-08): "Intelligent Transport Systems (ITS); V2X Applications; Part 1: Road Hazard Signalling (RHS) application requirements specification".
- [i.4] ETSI TS 102 723-5 (V1.1.1) (2012-11): "Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 5: Interface between management entity and facilities layer".

- [i.5] ETSI TR 102 863 (V1.1.1) (2011-06): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization".
- [i.6] ISO/IEC 19505-2:2012(E): "Information technology - Object Management Group Unified Modeling Language (OMG UML), Superstructure".
- [i.7] European Commission: "2010-2013 ICT Standardisation Work Programme for industrial innovation", 2nd update - 2012.

NOTE: Available at: [http://ec.europa.eu/enterprise/sectors/ict/files/ict-policies/2010-2013\\_ict\\_standardisation\\_work\\_programme\\_2nd\\_update\\_en.pdf](http://ec.europa.eu/enterprise/sectors/ict/files/ict-policies/2010-2013_ict_standardisation_work_programme_2nd_update_en.pdf).

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

**area of interest:** geographical area specified by data consumer limiting the LDM to satisfying the data consumers' subsequent requests for information only from data originating within that area

**LDM area of maintenance:** geographical area specified by the LDM for LDM maintenance

**LDM data consumer:** facility or an application that is authorized to request data from the LDM

**LDM data object:** object with attributes that can be accessed by the LDM Interfaces

**LDM data object identifier:** unique identifier within the LDM for a LDM Data Object added by a LDM Data Provider

**LDM data provider:** facility or an application that is authorized to provide the data to the LDM

**Local Dynamic Map (LDM):** facilities layer data store for storing LDM Data Objects that are timestamped and location referenced

### 3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN	Abstract Syntax Notation
BSA	Basic Set of Applications
CA	Co-operative Awareness
CAM	Co-operative Awareness Message
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Message
FA-SAP	Facilities/Applications Service Access Point
ICRW	Intersection Collision Risk Warning
ICT	Information and Communication Technology
ITS	Intelligent Transport System
ITS-AID	ITS Application Identifier
ITS-S	Intelligent Transport System Station
ITS-SU	Intelligent Transport System Station Unit
LCRW	Longitudinal Collision Risk Warning
LDM	Local Dynamic Map
MF-SAP	Management/Facilities Service Access Point
NF-SAP	Networking & Transport/Facilities Service Access Point
RHS	Road Hazard Signalling
SF-SAP	Security Facilities - Service Access Point
UML	Unified Modelling Language
V2V	Vehicle to Vehicle



---

## 4 General description of a LDM

A Local Dynamic Map (LDM) is a facility in cooperative Intelligent Transport Systems (ITS). It supports ITS applications by maintaining information on objects influencing or influenced by road traffic. ITS applications require information on moving objects such as vehicles nearby or on stationary objects such as traffic road signs. Information required by, or useful to active applications, can be maintained in a LDM.

The LDM is a conceptual data store located within an ITS-S as outlined in EN 302 665 [i.1] containing information which is relevant to the operation of ITS applications and related road safety and traffic efficiency. Data can be received from a range of different sources such as vehicles, infrastructure units, traffic centres, personal ITS stations, and on-board sensors and applications. The LDM offers mechanisms to grant secure access to the data that it holds. For example, the LDM can provide information on the surrounding vehicles and Road Side Units to any authorized application that requests it.

The information stored in the LDM can be accessed in the form of objects called LDM Data Objects. LDM Data Objects are provided from for example basic services for ITS Message Sets such as those defined in EN 302 637-2 [4] and EN 302 637-3 [5]. LDM Data Objects can be composed of sub objects, similar to the hierarchical structure of data frames in messages, and the objects contain attributes representing data elements from TS 102 894-2 [3]. Information on a vehicle or road side ITS-S for example is provided by a cooperative awareness basic service as defined in EN 302 637-2 [4] and is accessed from the LDM as a LDM Data Object with sub-objects representing the information from the CAM Basic Container. Information on an event for example is provided by a distributed environmental notification basic service as defined in EN 302 637-3 [5] and is accessed from the LDM as a LDM Data Object with sub-objects for the situation, location and a la carte containers.

The LDM can also store LDM Data Objects from applications and other facilities. For example, the LDM may maintain information on the ITS-S it is part of.

The LDM does not modify the data provided by LDM Data Providers. No permanent, static information is required to be stored in the LDM.

### 4.1 Functionality provided by the LDM

The basic functionality of the LDM is to provide a repository of information for facilities and applications. Facilities such as the CA and DEN basic services can store information into the LDM. Applications can retrieve information from and store information into the LDM. Additional functionality of the LDM includes:

- Registration/Deregistration of facilities and applications as LDM Data Providers/sinks to the LDM via the security layer (authorization) (see clause 6.1.1).
- Subscribe/Unsubscribe for notifications (see clause 6.3.4).
- Information retention by applying rules, e.g. based on time and/or location (see clause 5.3.2).
- Prioritization of requests (see clause 6.3.3).

---

## 5 LDM functional specification

### 5.1 LDM requirements

#### 5.1.1 LDM functional requirements

A LDM may communicate with other entities within the ITS-S architecture outlined in EN 302 665 [i.1] in order to:

- receive incoming information such as decoded CAMs in accordance with EN 302 637-2 [4] and DENMs in accordance with EN 302 637-3 [5];
- store and protect information according to constraints of time and LDM Area of Maintenance;

- provide information to authorized applications as requested:
  - by means of a subscription/notification method; or
  - by means of queries including spatial queries;
- prioritize data requests;
- store and protect LDM Data Objects so that it can be shared with applications;
- provide a mechanism for facilities and applications to register and deregister as LDM Data Providers;
- provide a mechanism for applications to register and deregister as LDM Data Consumers;
- ensure data access by LDM Data Providers and LDM Data Consumers is authorized.

### 5.1.2 LDM other requirements

In addition to the functional requirements listed in clause 5.1.1, a LDM may be constrained by a range of other requirements such as reliability (system maturity, fault tolerance and restorability) and scalability. However, within communications systems such requirements are normally considered to be related to procurement and, consequently, are not specified in the present document.

## 5.2 The LDM within the ITS-S communication architecture

The LDM collects, qualifies (ensures that it is valid and from an authorized source) and stores data received from other ITS-Ss. The LDM may also collect, qualify and store information from other sources such as traffic information providers, or from its own sensors and applications.

As shown in Figure 1, the LDM receives data from other ITS-Ss through a common interface which is available to all message services such as CA and DEN within the ITS-S Facilities layer. Information is exchanged with other services or applications by invoking functions located at the FA-SAP as outlined in EN 302 665 [i.1]. Security and management permissions are provided by functions which are located at the SF-SAP and the MF-SAP respectively.

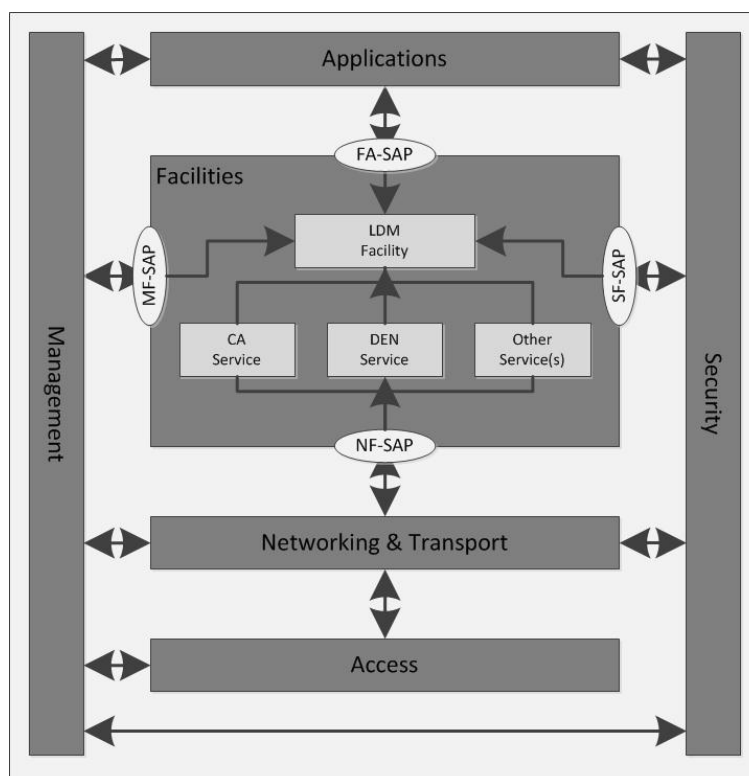
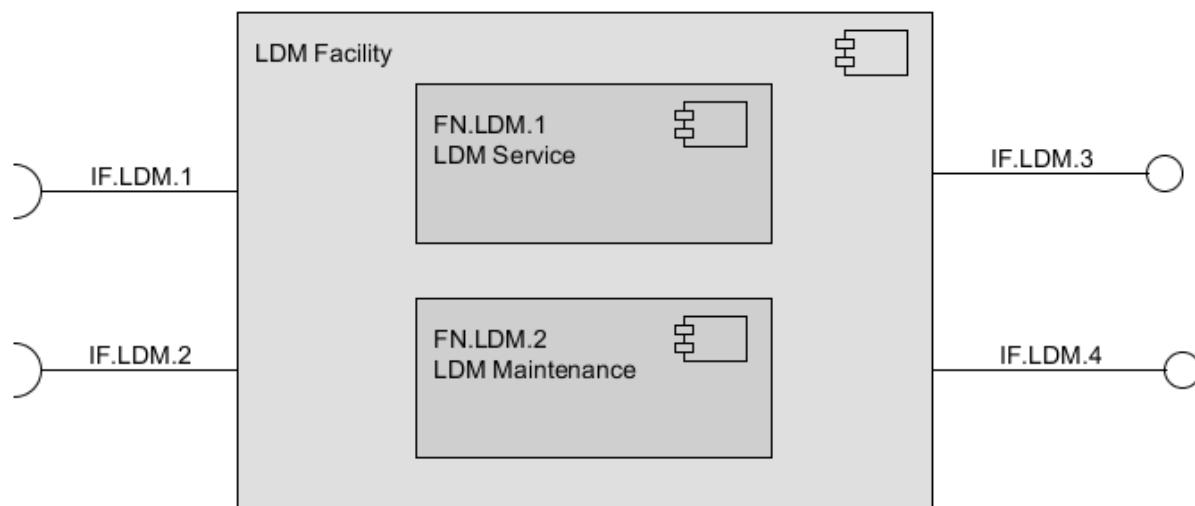


Figure 1: LDM and logical interfaces

## 5.3 LDM functional architecture

The rationale for and guidance on standardization of the LDM outlined in TR 102 863 [i.5] specify two main components of the LDM; the LDM Maintenance component and the LDM Service component (see clause 5.3.1). Figure 2 shows these two components of the LDM and its main interfaces in a Unified Modelling Language (UML) component diagram in accordance with ISO/IEC 19505-2 [i.6]. The interfaces are separated into those required by the LDM (IF.LDM.1 and IF.LDM.2) and those exposed to other facilities and applications (IF.LDM.3 and IF.LDM.4).



**Figure 2: LDM Functional architecture**

The LDM shall provide the functions defined in Table 1 and illustrated in Figure 2.

**Table 1: LDM Functions**

Function	Description
FN.LDM.1	The LDM Service component (see clause 5.3.1) is responsible for providing functionalities to authorized LDM Data Providers for LDM data manipulation (such as adding new data, modifying existing data, delete existing data), direct access to data (query data) and a publish/subscribe mechanism for data access by LDM Data Consumers. It also provides registration and deregistration functionalities to LDM Data Providers and LDM Data Consumers.
FN.LDM.2	The LDM Maintenance component (see clause 5.3.2) is responsible for storing and maintaining the data and its integrity as well as for the garbage collection of persistent data held within the LDM.

### 5.3.1 Function FN.LDM.1 - LDM Service

The LDM is connected to authorized LDM Data Providers and LDM Data Consumers. LDM Data Providers provide information to the LDM which makes these data available to LDM Data Consumers. The LDM offers three different types of interfaces:

- a transaction interface for LDM Data Providers, where a transaction describes a sequence of LDM Data Object exchanges between a LDM Data Provider and the LDM (see clause 6.2.3);
- a query interface for LDM Data Consumers (see clause 6.3.3); and
- a publish/subscribe interface for LDM Data Consumers (see clause 6.3.4).

The LDM shall:

- provide a mechanism for facilities to register and deregister as LDM Data Providers;
- provide a mechanism for applications to register and deregister as LDM Data Providers or LDM Data Consumers;
- verify the authorization of LDM Data Providers and LDM Data Consumers prior to data access.

### 5.3.2 Function FN.LDM.2 - LDM Maintenance

The LDM shall maintain all LDM Data Objects received from registered and authorized LDM Data Providers during their time validity and within the LDM Area of Maintenance of the LDM.

The LDM considers a LDM Data Object to be valid during the time period starting on the timestamp of the LDM Data Object and for the duration of the time validity period. A LDM Data Provider specifies the timestamp and the time validity of every LDM Data Object it provides to the LDM:

- The timestamp is specified upon adding or updating the LDM Data Object (see clause 6.2.3).
- The default time validity for all LDM Data Objects is specified upon registration (see clause 6.2.1). The default time validity is replaced by the time validity specified for a specific LDM Data Object upon adding or updating the LDM Data Object (see clause 6.2.3).

The LDM considers a LDM Data Object to be valid if the location of the LDM Data Object intersects with the LDM Area of Maintenance. A LDM Data Provider specifies the location upon adding or updating a LDM Data Object (see clause 6.2.3). The LDM Area of Maintenance is a geographical area defined by the LDM, which can be defined relative to the momentary location of the host ITS-S.

## 5.4 Interfaces of the LDM

The LDM interfaces are identified in Figure 2 and specified in Table 2. Table 2 consists of the following 5 columns:

- Interface ID - providing the identifier of the interface described.
- Interface Type - describing the type of interface, with provided (P): interface is realized by the LDM and offered to its clients, required (R): interface is needed by the LDM to perform an action but realized by another component.
- Component connected - name of the component interacting with the LDM.
- Message type - type of message exchanged via the interface.
- Direction - describing the message flow, with IN: message received by the LDM, OUT: message provided by the LDM.

**Table 2: LDM Interfaces**

Interface ID	Interface Type	Component connected	Message Type	Direction
IF.LDM.1	R	Management layer	MF-SAP	IN and OUT
IF.LDM.2	R	Security layer	SF-SAP	IN and OUT
IF.LDM.3	P	LDM Data Providers	CAM, DENM and other	IN
IF.LDM.4	P	LDM Data Consumers	CAM, DENM and other	OUT

NOTE: This is a non-exclusive list which may be extended in the future.

### 5.4.1 Interface IF.LDM.1 - management layer

The interface IF.LDM.1 to the ITS Management layer is described in TS 102 723-5 [i.4].

### 5.4.2 Interface IF.LDM.2 - security layer

The LDM shall provide an interface IF.LDM.2 for the exchange of information with the ITS Security layer as described in EN 302 665 [i.1] in order to verify the authorization of an ITS application or facility to access or modify specific LDM Data Objects within the LDM.

The ITS security layer will exchange information with the LDM across interface IF.LDM.2 in order to revoke the authorization of a previously authorized ITS LDM Data Provider and LDM Data Consumer.

### 5.4.3 Interface IF.LDM.3 - LDM Data Providers

The LDM shall provide an interface IF.LDM.3 to enable an application or facility to register as a LDM Data Provider and, subsequently, to send LDM Data Objects to the LDM.

A LDM Data Provider shall register with the LDM before the LDM accepts LDM Data Objects from the LDM Data Provider. The LDM shall request the security layer to check if the LDM Data Provider is authorized using the message sequence specified in clause 6.1.1.1 across interface IF.LDM.2 (clause 5.4.2). The LDM shall confirm the success of the authorization to the LDM Data Provider.

The LDM shall at least support the exchange of LDM Data Objects, sub-objects and attributes derived from frames, sub-frames and elements such as defined in the Common Data Dictionary as specified in TS 102 894-2 [3]. Further details on LDM Data Objects are out of scope of the present document.

While the LDM Data Provider is registered, the LDM shall provide access to LDM Data Objects for which the LDM Data Provider is authorized.

When the authorization is revoked by the services of the security layer then the LDM shall deny further access to the LDM Data Objects.

A LDM Data Provider may deregister itself from the LDM after which it shall no longer have access to LDM Data Objects.

A LDM Data Provider shall provide a timestamp and location for LDM maintenance purposes with each LDM Data Object sent to the LDM.

The LDM may store LDM Data Objects identified in Table 3 if offered by an authorized LDM Data Provider. The LDM may update parts of the LDM Data Objects if offered by an authorized LDM Data Provider.

**Table 3: LDM data objects**

Message type	Reference	Data Object Description
CAM	EN 302 637-2 [4]	All data objects and attributes from a CAM
DENM	EN 302 637-3 [5]	All data objects and attributes from a DENM

NOTE: This is a non-exclusive list which may be extended in the future.

### 5.4.4 Interface IF.LDM.4 - LDM Data Consumers

The LDM shall provide an interface IF.LDM.4 to enable an ITS application or facility to register as a LDM Data Consumer and to access data in the LDM.

A LDM Data Consumer shall register with the LDM before the LDM provides access to the LDM Data Objects. The LDM requests the security layer to check if the LDM Data Consumer is authorized using the message sequence specified in clause 6.1.1.1 across interface IF.LDM2 (see clause 5.4.2). The LDM checks the Area of Interest against its LDM Area of Maintenance. The LDM shall confirm the success of the registration to the LDM Data Consumer.

While the LDM Data Consumer is registered, the LDM shall grant access to LDM Data Objects for which the LDM Data Consumer is authorized.

When the authorization is revoked by the security services then the LDM shall inform the LDM Data Consumer that its registration is revoked and shall deny further access to the LDM Data Objects.

A LDM Data Consumer may deregister itself from the LDM after which it shall no longer have access to LDM Data Objects.

The LDM shall support the exchange of data as defined in the Common Data Dictionary as specified in TS 102 894-2 [3].

The LDM shall provide mechanisms for filtering the LDM Data Objects to be returned upon request from the LDM Data Consumer. These filtering mechanisms are:

- 1) A querying mechanism for an immediate single data request.
- 2) A publish/subscribe mechanism for the continuous return of data which shall support the following:
  - a) Event driven data request according to the given filter.
  - b) Periodic data request according to a given time interval and filter.
  - c) Composite event driven data request according to a given filter or time interval.

The filtering mechanisms contain a filter on one or more attributes of the requested LDM Data Objects. A filter on a single attribute of a LDM Data Object compares the attribute value against a reference value (see clause A.1.2.2).

Only LDM Data Objects that meet the specified filtering criteria and are within the defined Area of Interest shall be returned to the LDM Data Consumer by the LDM.

The response to a data request shall be a list of zero or more requested LDM Data Objects. In the case of the publish/subscribe mechanism a LDM Data Consumer receives a response to the same request every time the subscription criteria are matched.

The LDM shall support the prioritization of processing data requests.

---

## 6 LDM Interfaces

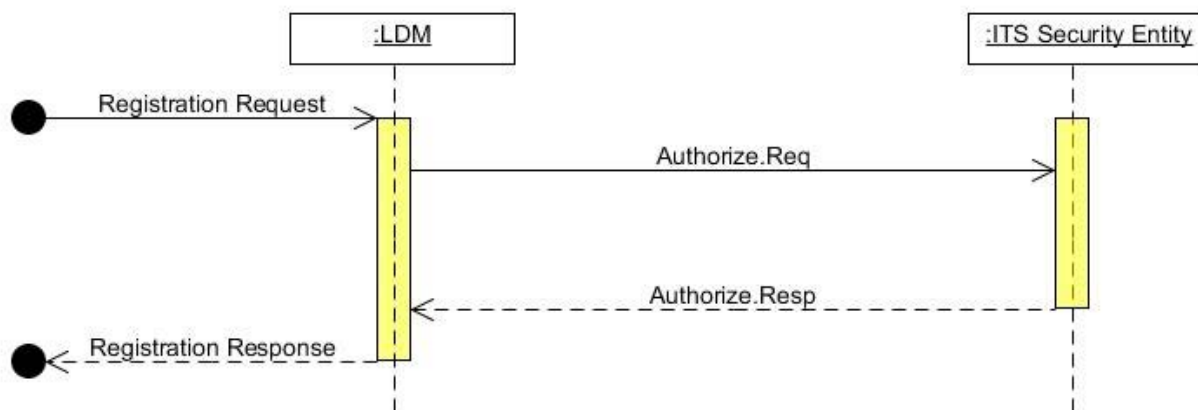
The interfaces to LDM Data Providers, LDM Data Consumers, and to the security and management layers are defined here as messages in the information flow. These messages can also be considered as the data part of the service primitives to the AF-SAP, MF-SAP and SF-SAP, and need to be extended with the source and destination addresses. This clause specifies the minimal functionality of the LDM interfaces.

### 6.1 Interface IF.LDM.2 to the security layer

#### 6.1.1 Authorization

##### 6.1.1.1 Authorize messages

When the LDM receives a Registration request from a LDM Data Provider or a LDM Data Consumer (see clauses 6.2.1 and 6.3.1) it shall send an Authorize request message to the ITS Security layer across the interface IF.LDM.2 to verify if the LDM Data Provider or LDM Data Consumer is authorized for access to LDM Data Objects. Figure 3 shows the message sequence with the authorization request message (*Authorize.Req*) and the response message (*Authorize.Resp*) to confirm the successful or unsuccessful authorization. The Registration Request and Response messages are defined in clauses 6.2.1.1 and 6.3.1.1 for the LDM Data Provider and Consumers respectively.



**Figure 3: LDM Data Provider or Consumer authorization message sequence**

The content of the Authorize request and response messages in this information flow are specified at a functional level as in Table 4. The request and response messages are specified at a syntactical level in annex B.

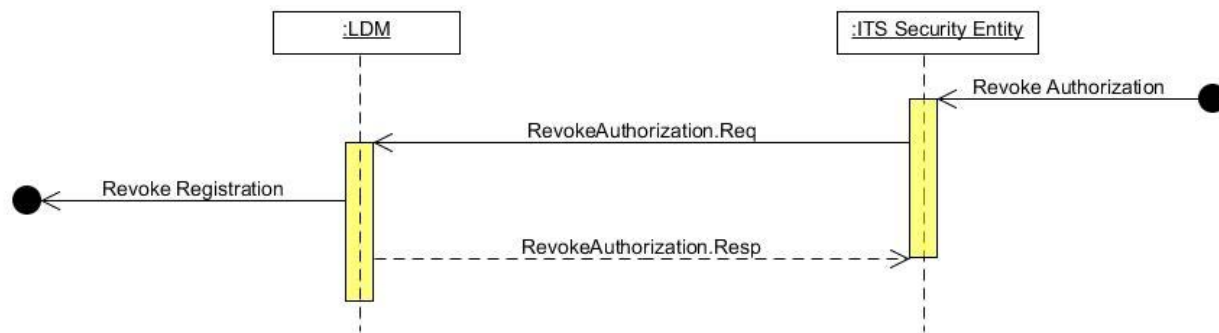
**Table 4: Contents of Authorize Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Permissions for which access is requested/granted	Permissions are defined as one or more root LDM Data Object types (or classes) that can be accessed from the LDM. Typical permissions are the root classes of LDM Data Objects decoded from ITS message sets from Table 3 and TS 102 894-2 [3]	Mandatory	Optional (see note)
Result	Indication of result of the request: <ul style="list-style-type: none"> <li>• Successful</li> <li>• Fail: Invalid ITS-AID</li> <li>• Fail: Unable to authenticate application</li> <li>• Fail: Application not authorized for requested permissions</li> </ul>		Mandatory
NOTE:	Mandatory if the Result parameter is set to "Successful" and the list of permissions, for which authorization is granted, is different from that requested.		

## 6.1.2 Revocation

### 6.1.2.1 RevokeAuthorization messages

The RevokeAuthorization message shall be sent by the ITS Security Layer to the LDM across the interface IF.LDM.2 to inform the LDM that authorization to permit a particular LDM Data Provider or LDM Data Consumer to have access to information in the LDM has been revoked. Figure 4 shows the message sequence with the revocation request message (`RevokeAuthorization.Reg`) and the response message (`RevokeAuthorization.Resp`) to confirm the successful or unsuccessful revocation. The Revoke Registration message is defined in clauses 6.2.2.2 and 6.3.2.2 for the LDM Data Provider and Consumers respectively.



**Figure 4: Authorization revocation message sequence**

The content of the RevokeAuthorization request and response messages are specified at a functional level as in Table 5. The request and response messages are specified at a syntactical level in annex B.

**Table 5: Contents of RevokeAuthorization Information Flows**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Reason	Indication of the reason why authorization is revoked for the application: <ul style="list-style-type: none"> <li>Registration of the application or service has been revoked by the Registration Authority</li> <li>Period of authorization has expired</li> </ul>	Mandatory	
Revocation acknowledgement	Indication that the revocation has been received and processed		Mandatory

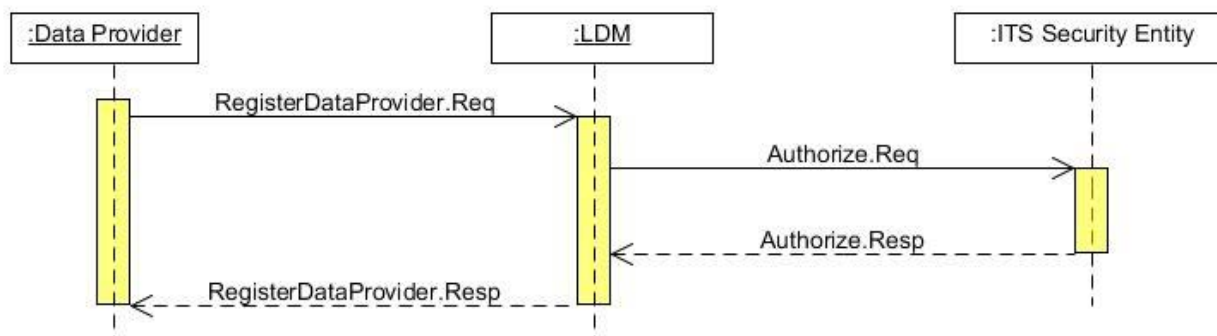
## 6.2 Interface IF.LDM.3 to LDM Data Providers

### 6.2.1 Registration

#### 6.2.1.1 RegisterDataProvider messages

A LDM Data Provider shall send a RegisterDataProvider request message to the LDM across the interface IF.LDM.3 to register for access to LDM Data Objects. Figure 5 shows the message sequence with the registration request message (`RegisterDataProvider.Reg`) and the response message (`RegisterDataProvider.Resp`) to confirm the successful or unsuccessful registration. The LDM uses the Authorize message across interface IF.LDM.2 to request the verification of the authorization from the security layer (see clause 6.1.1.1).

**NOTE:** If a LDM Data Provider is already registered with the LDM it is deregistered without notification before the new registration request is processed.



**Figure 5: LDM Data Provider registration message sequence**



The content of the RegisterDataProvider request and response messages are specified at a functional level as in Table 6. The request and response messages are specified at a syntactical level in annex B.

**Table 6: Contents of RegisterDataProvider Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Permissions for which access is requested/granted	Permissions are defined as one or more root LDM Data Object types (or classes) that can be provided as the root classes of LDM Data Objects decoded from ITS message sets from Table 3 and TS 102 894-2 [3]	Mandatory	Optional (see note)
Time validity	Default time validity of provided data	Mandatory	
Result	Indication of result of the registration request: <ul style="list-style-type: none"> <li>Accepted</li> <li>Rejected</li> </ul>		Mandatory
NOTE:	Mandatory if the Result parameter is set to the value "Accepted" and the list of permissions for which authorization is granted is different from that requested.		

## 6.2.2 Deregistration

### 6.2.2.1 DeregisterDataProvider messages

A LDM Data Provider shall send a DeregisterDataProvider request message to the LDM to deregister itself as a LDM Data Provider from the LDM. Figure 6 shows the message sequence for deregistration with the request message (DeregisterDataProvider.Reg) and the response message (DeregisterDataProvider.Resp) to confirm the deregistration.



**Figure 6: LDM Data Provider deregistration message sequence**

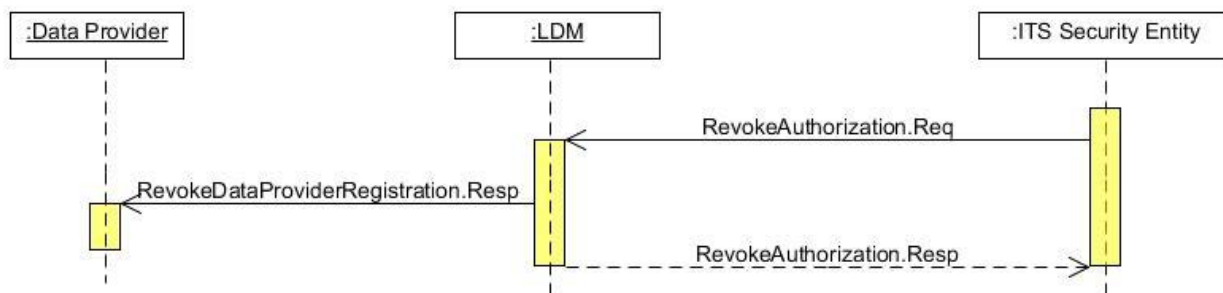
The content of the DeregisterDataProvider request and response messages flow are specified at a functional level as in Table 7. The request and response messages are specified at a syntactical level in annex B.

**Table 7: Contents of DeregisterDataProvider Information flows**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Deregistration acknowledgement	Indication that the deregistration message has been received and processed.		Mandatory

### 6.2.2.2 RevokeDataProviderRegistration message

When the authorization of a LDM Data Provider is revoked by the security layer (see clause 6.1.2.1) then the LDM shall send a RevokeDataProviderRegistration response message to the LDM Data Provider to inform that its registration is terminated and further access to the LDM Data Objects will be denied. Figure 7 shows the message sequence with the response message (RevokeDataProviderRegistration.Resp) to inform the LDM Data Provider that its registration has been revoked.



**Figure 7: LDM Data Provider revocation message sequence**

The content of the RevokeDataProviderRegistration response message are specified at a functional level as in Table 8. The request and response messages are specified at a syntactical level in annex B.

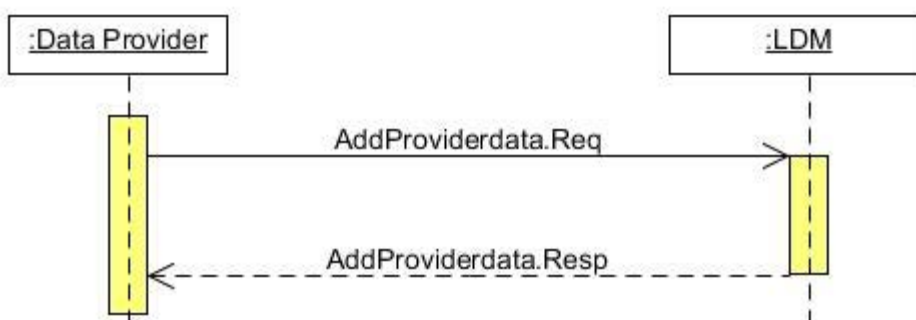
**Table 8: Contents of RevokeAuthorization Information flows**

Parameter	Content	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory

## 6.2.3 Maintenance of Provider data

### 6.2.3.1 AddProviderdata messages

A LDM Data Provider shall send an AddProviderdata request message to the LDM across interface IF.LDM.3 to add a new LDM Data Object in the LDM. Figure 8 shows the message sequence with the request message (AddProviderData .Req) and the response message (AddProviderData .Resp) to confirm the successful or unsuccessful request to add a new LDM Data Object.



**Figure 8: AddProviderdata message sequence**

The content of the AddProviderdata request and response messages are specified at a functional level as in Table 9. The request and response messages are specified at a syntactical level in annex B.

**Table 9: Contents of AddProviderdata Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Timestamp	Timestamp of the Data Object	Mandatory	
Location	Location of the Data Object	Mandatory	
Data Object	New Data Object to be added	Mandatory	
Time validity	Time validity of the Data Object	Optional	
Data Object identifier	Unique identifier of the Data Object to be added		Mandatory (see note)

NOTE: The LDM shall return "unsuccessful" when the Data Object cannot be added to the data store.

### 6.2.3.2 UpdateProviderdata messages

A LDM Data Provider shall send an UpdateProviderdata request message to the LDM across interface IF.LDM.3 to update an existing LDM Data Object in the LDM. Figure 9 shows the message sequence with the update request message (UpdateProviderdata.Reg) and the response message (UpdateProviderdata.Resp) to confirm the successful or unsuccessful request to update a LDM Data Object.



**Figure 9: Update provider data message sequence**

The content of the UpdateProviderdata request and response messages are specified at a functional level as in Table 10. The request and response messages are specified at a syntactical level in annex B.

**Table 10: Contents of UpdateProviderdata Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Data Object identifier	Identifier of the Data Object to be updated	Mandatory	Mandatory
Timestamp	Timestamp of the Data Object	Mandatory	
Location	Location of the Data Object	Mandatory	
Data Object	Updated Data Object	Mandatory	
Time validity	Time validity of the Data Object	Optional	
Result	Indication of result of the update request: <ul style="list-style-type: none"> <li>Update successful</li> <li>Update unsuccessful because Data Object Identifier does not exist</li> <li>Update unsuccessful because Data Object type is inconsistent with the existing Data Object with this Data Object Identifier</li> </ul>		Mandatory

### 6.2.3.3 DeleteProviderdata messages

A LDM Data Provider shall send a DeleteProviderdata request message to the LDM across interface IF.LDM.3 to delete an existing LDM Data Object in the LDM. Figure 10 shows the message sequence with the delete request message (DeleteProviderdata.Reg) and the response message (DeleteProviderdata.Resp) to confirm the request to delete a LDM Data Object.



**Figure 10: Delete provider data message sequence**

The content of the DeleteProviderdata response and request messages are specified at a functional level as in Table 11. The request and response messages are specified at a syntactical level in annex B.

If a timestamp is provided then the LDM shall delete only the LDM Data Object from the history having this specific timestamp or being older.

**Table 11: Contents of DeleteProviderdata Information Flow**

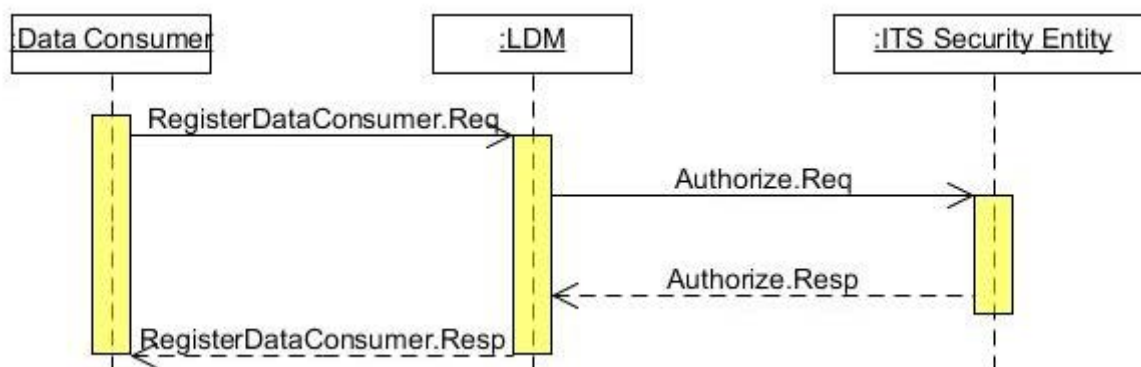
Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Data Object identifier	Identifier of the Data Object to be deleted	Mandatory	Mandatory
Timestamp	Timestamp indicating the minimum age of a Data Object (part) to be deleted	Optional	
Result	Indication of result of the update request: <ul style="list-style-type: none"> <li>• Deletion successful</li> <li>• Deletion unsuccessful</li> </ul>		Mandatory

## 6.3 Interface IF.LDM.4 to LDM Data Consumers

### 6.3.1 Registration

#### 6.3.1.1 RegisterDataConsumer messages

A LDM Data Consumer shall send a RegisterDataConsumer request message to the LDM across the interface IF.LDM.4 to register for access to LDM Data Objects. Figure 11 shows the message sequence with the register request message (`RegisterDataConsumer.Reg`) and the response message (`RegisterDataConsumer.Resp`) to confirm the successful or unsuccessful registration. The LDM uses the Authorization message across interface IF.LDM.2 to request the verification of the authorization from the security layer (see clause 6.1.1.1).



**Figure 11: RegisterDataConsumer message sequence**

The content of the RegisterDataConsumer request and response messages are specified at a functional level as in Table 12. The request and response messages are specified at a syntactical level in annex B.

**Table 12: Contents of RegisterDataConsumer Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Permissions for which access is requested/granted	Permissions are defined as one or more root LDM Data Object types (or classes) that can be requested from the LDM such as the root classes of LDM Data Objects decoded from ITS message sets from Table 3 and TS 102 894-2 [3]	Mandatory	Optional (see note)
Area of Interest	The reference position of the Area of Interest is the ITS station location. It can have any of the shapes as defined in TS 102 894-2 [3], within the definition of the LDM Area of Maintenance	Optional	
Result	Indication of result of the registration request: <ul style="list-style-type: none"> <li>Accepted; Area of Interest covered by LDM Area of Maintenance</li> <li>Accepted; Area of Interest extends beyond LDM Area of Maintenance</li> <li>Rejected</li> </ul>		Mandatory
NOTE:	Mandatory if the Result parameter is set to the value "Accepted; Area of Interest covered by LDM Area of Maintenance" or "Accepted; Area of Interest extends beyond LDM Area of Maintenance" and the list of permissions for which authorization is granted is different from that requested.		

## 6.3.2 Deregistration

### 6.3.2.1 DeregisterDataConsumer messages

A LDM Data Consumer shall send a DeregisterDataConsumer request message to the LDM across the interface IF.LDM.4 to deregister itself as a LDM Data Consumer from the LDM. Figure 12 shows the message sequence for deregistration with the request message (DeregisterDataConsumer.Reg) and the response message (DeregisterDataConsumer.Resp) to confirm the deregistration.

**Figure 12: DeregisterDataConsumer message sequence**

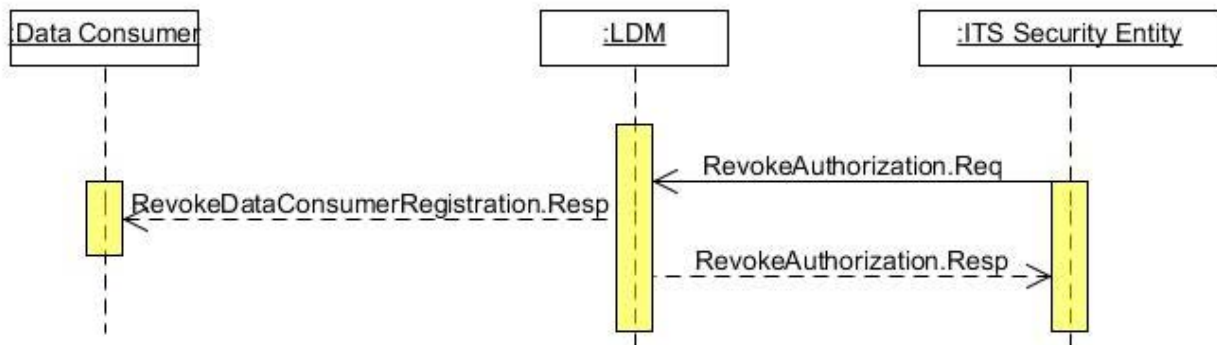
The content of the DeregisterDataConsumer request and response messages are specified at a functional level as in Table 13. The request and response messages are specified at a syntactical level in annex B.

**Table 13: Contents of DeregisterDataConsumer Information Flows**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Deregistration acknowledgement	Indication that the deregistration message has been received and processed		Mandatory

### 6.3.2.2 RevokeDataConsumerRegistration message

The LDM shall send a `RevokeDataConsumerRegistration` response message to a LDM Data Consumer across interface IF.LDM.4 to inform the LDM Data Consumer that its registration is terminated and further access to the LDM Data Objects will be denied. Figure 13 shows the message sequence with the response message (`RevokeDataConsumerRegistration.Resp`) to inform the LDM Data Consumer that its registration has been revoked.



**Figure 13: RevokeDataConsumerRegistration message sequence**

The content of the `RevokeDataConsumerRegistration` response message are specified at a functional level as in Table 14. The request and response messages are specified at a syntactical level in annex B.

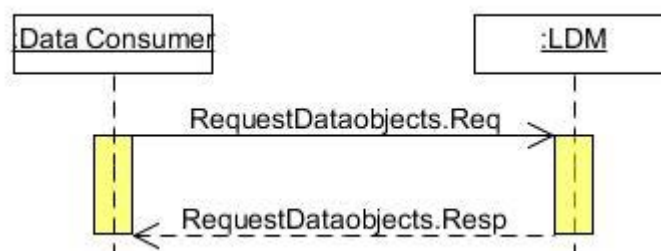
**Table 14: Contents of RevokeDataConsumerRegistration Information Flow**

Parameter	Content	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory

### 6.3.3 Data Request

#### 6.3.3.1 RequestDataobjects messages

A LDM Data Consumer shall send a `RequestDataobjects` request message to the LDM across the interface IF.LDM.4 to requesting LDM Data Objects from the LDM according to the given request parameters. Figure 14 shows the message sequence with the request message (`RequestDataobjects.Reg`) and the response message (`RequestDataobjects.Resp`) with the result of the successful or unsuccessful LDM Data Objects request.



**Figure 14: RequestDataobjects message sequence**

The content of the `RequestDataobjects` request and response messages are specified at a functional level as in Table 15. The request and response messages are specified at a syntactical level in annex B.

**Table 15: Contents of RequestDataobjects Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Data Object type	A Data Type Identifier for a Data Object Type such as defined in TS 102 894-2 [3]	Mandatory	Optional
Priority	Priority for processing which is lower or equal to the maximum authorized priority	Optional	
Filter	Filter criteria see clause A.1.	Optional	
Order	Indication of the natural ordering of the requested Data Objects, defining the attributes to be used and the ordering direction. The ordering direction can have two possible values: <ul style="list-style-type: none"> <li>• Ascending; or</li> <li>• Descending</li> </ul>	Optional	
Requested Data	List of zero or more requested Data Objects that match the filter, originate within the Area of Interest and are in the requested order		Optional (see note 1)
Result	Indication of result: <ul style="list-style-type: none"> <li>• Successful</li> <li>• Fail: Invalid ITS-AID</li> <li>• Fail: Invalid Data Object type</li> <li>• Fail: Invalid Priority</li> <li>• Fail: Invalid Filter</li> <li>• Fail: Invalid Order</li> </ul>		Mandatory
ErrorMessage	Error message describing the error of the request failure, e.g. due to an error in the definition of priority, the filter, or ordering of the requested Data Object Type		Optional (see note 2)
NOTE 1: Mandatory if the Result parameter is set to the value "Successful".			
NOTE 2: Mandatory if the Result parameter is not set to the value "Successful".			

A filter is defined as a set of one or more filter statements (see clause A.1). A filter statement comprises an attribute of the requested LDM Data Object, a comparison operator and a reference value to compare against. It evaluates to either TRUE or FALSE.

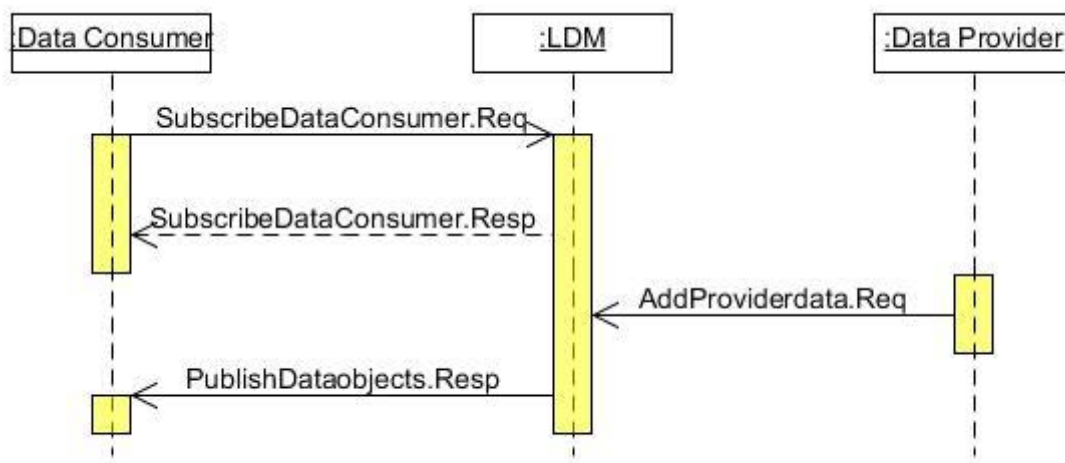
## 6.3.4 Subscription

### 6.3.4.1 SubscribeDataConsumer messages

A LDM Data Consumer shall send a SubscribeDataConsumer request message to the LDM across the interface IF.LDM.4 to subscribe to new LDM Data Objects in the LDM.

The LDM shall send a PublishDataobjects response message to a LDM Data Consumer across interface IF.LDM.4 whenever the information to which the LDM Data Consumer subscribes is added to the LDM or updated.

Figure 15 shows an example of a message sequence with the request message (`SubscribeDataConsumer.Req`) and the response message (`SubscribeDataConsumer.Resp`) with the result of the for a successful or unsuccessful subscription, and resultant publication message (`PublishDataobjects.Resp`).



**Figure 15: LDM Data Consumer subscription and result publication message sequence**

The content of the SubscribeDataConsumer request and response messages are specified at a functional level as in Table 16. The content of the PublishDataobjects response message are specified at a functional level as in Table 17. The request and response messages are specified at a syntactical level in annex B.

**Table 16: Contents of SubscribeDataConsumer Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Data Object type	A Data Type Identifier for a Data Object Type such as defined in TS 102 894-2 [3]	Mandatory	
Priority	Priority for processing which is lower or equal to the maximum authorized priority	Optional	
Filter	Filter criteria as defined in Table 15 in clause 6.3.3.1	Optional	
Notification Interval	Indication of a time period	Optional	
Multiplicity	Indication of minimum number of Data Objects in Result	Optional	
Order	Indication of ordering the requested Data Objects. Defining the attributes to be used and the ordering direction	Optional	
Subscription Identifier	Unique Identifier defined by the LDM to Identify this subscription		Optional (see note 1)
Result	Indication of result of the subscription request: <ul style="list-style-type: none"> <li>• Successful</li> <li>• Fail: Invalid ITS-AID</li> <li>• Fail: Invalid Data Object type</li> <li>• Fail: Invalid Priority</li> <li>• Fail: Invalid Filter</li> <li>• Fail: Invalid Notification Interval</li> <li>• Fail: Invalid Multiplicity</li> <li>• Fail: Invalid Order</li> </ul>		Mandatory
ErrorMessage	Error message describing the error of the subscription failure, e.g. due to an error in the definition of priority, the filter, or ordering of the requested Data Object Type		Optional (see note 2)

NOTE 1: Mandatory if the Result parameter is set to the value "Successful".  
NOTE 2: Mandatory if the Result parameter is not set to the value "Successful".

**Table 17: Contents of PublishDataobjects Information Flow**

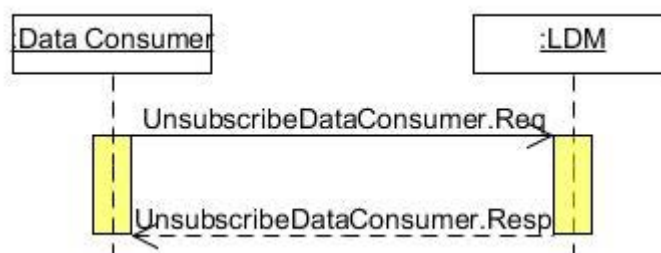
Parameter	Content	Response
Subscription Identifier	Subscription Identifier as received during subscription	Mandatory
Requested Data	Requested data as specified in Table 15	Mandatory



## 6.3.5 Cancel subscription

### 6.3.5.1 UnsubscribeDataConsumer messages

A LDM Data Consumer shall send an UnsubscribeDataConsumer request message to the LDM across the interface IF.LDM.4 to unsubscribe from the notification service of the LDM. Figure 16 shows the message sequence with the request message (UnsubscribeDataConsumer.Reg) and the response message (UnsubscribeDataConsumer.Resp) to confirm the successful or unsuccessful subscription cancellation.



**Figure 16: UnsubscribeDataConsumer message sequence**

The content of the UnsubscribeDataConsumer request and response messages are specified at a functional level as in Table 18. The request and response messages are specified at a syntactical level in annex B.

**Table 18: Contents of UnsubscribeDataConsumer Information Flow**

Parameter	Content	Request	Response
ITS Application Identifier	ITS-AID as defined in TS 102 860 [1]	Mandatory	Mandatory
Subscription Identifier	Subscription Identifier as received during subscription (see Table 17 in clause 6.3.4.1)	Mandatory	
Unsubscribe acknowledgement	Indication that the unsubscribe message has been received and processed		Mandatory

---

## Annex A (normative): Data request filtering

### A.1 Minimal syntax of data request filtering

The Data Request Filter restricts the data returned by a Data Request (see clause 6.3.3) or a Subscription (see clause 6.3.4). Every LDM Data Object that matches the filter and originates within the Area of Interest is returned in the result set.

A filter is defined as a set of one or more filter statements. A filter statement comprises an attribute of the requested LDM Data Object, a comparison operator and a reference value to compare against. A filter statement evaluates to either TRUE or FALSE. Filter statements can be combined using the logical AND and OR operators.

#### A.1.1 Filter

A filter consists of a set of one or more filter statements, combined with logical operators.

Filter = <FilterStatement> [<logical operator> <FilterStatement>]\*

#### A.1.2 Filter Statement

A filter statement shall comprise the following tuple:

- attribute identifier;
- comparison operator; and
- reference value.

A filter statement comprises = <DataObject.Attribute> <ComparisonOperator> <ReferenceValue>

### A.1.2.1 Comparison Operator

**Table A.1: Definition of comparison operators for filtering**

Operator	Description	Examples
Equal	Equals (==)	Return results where the <i>CauseCode</i> of a DENM is 2 ("accident"): Filter: <code>causeCode == 2</code>
Notequal	Does not equal (!=)	Return results where the <i>CauseCode</i> of a DENM is not 0 ("unknown"): Filter: <code>causeCode != 0</code>
Gt	Greater than (>)	Return results where the <i>ReferencePosition.longitude</i> of a DENM is greater than 72 566 045: Filter: <code>referencePosition.longitude &gt; 72 566 045</code>
Lt	Less than (<)	Return results where the <i>ReferencePosition.latitude</i> of a DENM is less than 23 039 568: Filter: <code>referencePosition.latitude &lt; 23 039 568</code>
Gte	Greater than or equal to (>=)	Return results where the <i>ReferencePosition.longitude</i> of a DENM is greater or equal than 72 566 045: Filter: <code>referencePosition.longitude &gt;= 72 566 045</code>
Lte	Less than or equal to (<=)	Return results where the <i>ReferencePosition.latitude</i> of a DENM is smaller or equal than 23 039 568: Filter: <code>referencePosition.latitude &lt;= 23 039 568</code>
Like	Like (=~), the value contains a specified substring	Return results where the MESSAGE.ATTRIBUTE is containing the substring 'myCar': Filter: <code>referencePosition.streetName =~ 'myAddress'</code>
Notlike	Not Like (!~), the value does not contain a specified substring	Return results where the MESSAGE.ATTRIBUTE is not containing the substring 'myCar': Filter: <code>referencePosition.streetName !~ 'myCar'</code>

### A.1.2.2 Reference Value

The Reference Value is used for comparison and shall be of the same type as the DataObjectAttribute type it is compared to. It is defined as an absolute value.

### A.1.3 Logical operators for combining Filter Statements

Filter Statements can be combined in a filter using OR and AND boolean logic. The OR operator takes precedence over the AND operator.

**Table A.2: Definition of logical operators for filtering**

Operator	Description	Examples
Or	Logical OR (  )	Return results where the <i>CauseCodeType</i> of a DENM is either 2 ("accident") or 3 ("roadworks"): Filter: <code>causeCode == 2    causeCode==3</code>
And	Logical AND (&&)	Return results where the <i>CauseCodeType</i> of a DENM is 2 ("accident") and <i>SubCauseCodeType</i> is 2 ("heavy accident"): Filter: <code>causeCode == 2 &amp;&amp; subCauseCode==2</code>

**EXAMPLE:** A more complex example is to find all DENMs which are either "accidents" or "roadworks" within an area specified by upper left corner and lower right corner of a rectangle:

Filter: (*causeCode* equal 2 or *causeCode* equal 3) and (*referencePosition.latitude* gt 43 600 336 and *referencePosition.latitude* lt 43 606 551) and (*referencePosition.longitude* gt 7 062 578 and *referencePosition.longitude* lt 7 072 878).

---

## A.2 Ordering Data Request Results

The order of a Data Request Result is defined by providing a list of tuples. Each tuple is defined as:

Order = <DataObject.Attribute> <Ordering direction>

Where <DataObject.Attribute> identifies an attribute of the requested Data Object type (Table 15) and <Order direction> defines whether the ordering direction is ascending or descending, having either the value ASC (for ascending) or DESC (for descending). The tuples are performed successively, according to the position in the list of tuples, i.e. the first tuple is performed before the second, etc.

## Annex B (normative): ITS LDM Interface messages specified in ASN.1

The present annex defines ASN.1 types and values for the LDM interface defined in clause 6. The ASN.1 modules in this annex are in accordance to ISO/IEC 8824-1 [2].

```

ITSldm { itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wgl(1) itsldmapi(2863)
operation(0) version0(0) }

DEFINITIONS AUTOMATIC TAGS ::= BEGIN

IMPORTS

ITSaid FROM ITSaid0v0 { itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg2(2) itsaid(2860)
operation(0) version0(0)}

ItsPduHeader, CauseCode, Speed, StationID, TimestampIts, ReferencePosition, Longitude, Latitude FROM
ITS-Container { itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wgl (1) ts (102894) cdd
(2) version (1) }

CAM FROM CAM-PDU-Descriptions { itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wgl (1)
en (302637) cam (2) version (1) }

DENM, RelevanceDistance, RelevanceTrafficDirection FROM DENM-PDU-Descriptions { itu-t (0)
identified-organization (4) etsi (0) itsDomain (5) wgl (1) en (302637) denm (1) version (1) }

; --End of IMPORTS

-- IF.LDM.2 to security layer

-- Authorize Information Flow
AuthorizeReq ::= SEQUENCE {
    applicationId    ITSaid,
    accessPermissions PermissionsList
} -- End of AuthorizeReq

AuthorizeResp ::= SEQUENCE {
    applicationId    ITSaid,
    accessPermissions PermissionsList OPTIONAL,
    result           AuthorizationResult
} -- End of AuthorizeResp

-- RevokeAuthorization Information Flows
RevokeAuthorizationReq ::= SEQUENCE {
    applicationId    ITSaid,
    reason           RevocationReason
} -- End of RevokeAuthorizationReq

RevokeAuthorizationResp ::= SEQUENCE {
    applicationId    ITSaid,
    result           RevocationResult
} -- End of RevokeAuthorizationResp

PermissionsList ::= SEQUENCE OF Permission

Permission ::= DataObjectType

AuthorizationResult ::= ENUMERATED {
    successful (0),
    invalidITS-AID (1),
    authenticationFailure (2),
    applicationNotAuthorized (3)
} -- End of AuthorizationResult

RevocationReason ::= ENUMERATED {
    registrationRevokedByRegistrationAuthority (0),
    registrationPeriodExpired (1)
} -- End of RevocationReason

RevocationResult ::= ENUMERATED {
    successful (0),
    invalidITS-AID (1),

```

```

    unknownITS-AID (2)
} -- End of RevocationResult

-- IF.LDM.3 to data Provider

-- RegisterDataProvider Information Flow
RegisterDataProviderReq ::= SEQUENCE {
    applicationId    ITSaid,
    accessPermissions    PermissionsList,
    timeValidity      TimeValidity
} -- End of RegisterDataProviderReq

RegisterDataProviderResp ::= SEQUENCE {
    applicationId    ITSaid,
    accessPermissions    PermissionsList OPTIONAL, -- Mandatory if the Result parameter is set
to the value "Accepted" and the list of permissions for which authorization is granted is different
from that requested
    result          RegisterDataProviderResult
} -- End of RegisterDataProviderResp

-- DeregisterDataProvider Information Flow
DeregisterDataProviderReq ::= SEQUENCE {
    applicationId    ITSaid
} -- End of DeregisterDataProviderReq

DeregisterDataProviderResp ::= SEQUENCE {
    applicationId    ITSaid,
    result          DeregisterDataProviderAck
} -- End of DeregisterDataProviderResp

-- RevokeAuthorization Information Flow
RevokeDataProviderRegistrationResp ::= SEQUENCE {
    applicationId    ITSaid
} -- End of RevokeDataProviderRegistrationResp

-- AddProviderdata Information Flow
AddDataProviderReq ::= SEQUENCE {
    applicationId    ITSaid,
    timeStamp        TimestampIts,
    location         Location,
    dataObject       DataContainer, -- null value when the Data Object cannot be added to the data
store
    timeValidity     TimeValidity OPTIONAL
} -- End of AddDataProviderReq

AddDataProviderResp ::= SEQUENCE {
    applicationId    ITSaid,
    dataObjectID     DataObjectID -- null value when the Data Object cannot be added to the data
store
} -- End of AddDataProviderResp

-- UpdateProviderdata Information Flow
UpdateDataProviderReq ::= SEQUENCE {
    applicationId    ITSaid,
    dataObjectID     DataObjectID,
    timeStamp        TimestampIts,
    location         Location,
    dataObject       DataContainer,
    timeValidity     TimeValidity OPTIONAL
} -- End of UpdateDataProviderReq

UpdateDataProviderResp ::= SEQUENCE {
    applicationId    ITSaid,
    dataObjectID     DataObjectID,
    result          UpdateDataProviderResult
} -- End of UpdateDataProviderResp

-- DeleteProviderdata Information Flow
DeleteProviderdataReq ::= SEQUENCE {
    applicationId    ITSaid,
    dataObjectID     DataObjectID,
    timeStamp        TimestampIts OPTIONAL
} -- End of DeleteProviderdataReq

DeleteProviderdataProviderResp ::= SEQUENCE {
    applicationId    ITSaid,
    dataObjectID     DataObjectID,

```

```

    result          DeleteDataProviderResult
} -- End of DeleteProviderdataResp

RegisterDataProviderResult ::= ENUMERATED {
    accepted (0),
    rejected (1)
} -- End of RegisterDataProviderResult

DeregisterDataProviderAck ::= ENUMERATED {
    accepted (0),
    rejected (1)
} -- End of DeregisterDataProviderAck

UpdateDataProviderResult ::= ENUMERATED {
    succeed (0),
    unknownDataObjectID (1),
    inconsistentDataObjectType (2)
} -- End of UpdateDataProviderResult

DeleteDataProviderResult ::= ENUMERATED {
    succeed (0),
    failed (1)
} -- End of DeleteDataProviderResult

TimeValidity ::= INTEGER { oneMilliSec(1) }

Location ::= CHOICE {
    referencePosition ReferencePosition, -- Location of the Data Object
    referenceArea      ReferenceArea
}

DataObjectID ::= UID -- Unique Identifier

-- Data Object description
DataContainer ::= CHOICE {
    cam      CAM,
    denm     DENM,
    ...
} -- End of DataContainer
-- Any other Data Object such as defined in the Common

Data Directory
} -- End of DataContainer

-- IF.LDM.4 to data Consumer

-- RegisterDataConsumer Information Flow
RegisterDataConsumerReq ::= SEQUENCE {
    applicationId   ITSaid,
    accessPermissions PermissionsList,
    areaOfInterest AreaOfInterest OPTIONAL
} -- End of RegisterDataConsumerReq

RegisterDataConsumerResp ::= SEQUENCE {
    applicationId   ITSaid,
    accessPermissions PermissionsList OPTIONAL, -- Mandatory if the Result parameter is set to
    result          RegisterDataConsumerResult
} -- End of RegisterDataConsumerResp

-- DeregisterDataConsumer Information Flows
DeregisterDataConsumerReq ::= SEQUENCE {
    applicationId ITSaid
} -- End of DeregisterDataConsumerReq

DeregisterDataConsumerResp ::= SEQUENCE {
    applicationId ITSaid,
    ack           DeregisterDataConsumerAck
} -- End of DeregisterDataConsumerResp

-- RevokeDataConsumerRegistration Information Flow
RevokeDataConsumerRegistrationResp ::= SEQUENCE {
    applicationId ITSaid
} -- End of RevokeDataConsumerRegistrationResp

-- RequestDataobjects Information Flow
RequestDataobjectsReq ::= SEQUENCE {
    applicationId ITSaid,

```

```

    dataObjectType  DataObjectType,
    priority        UserPriority OPTIONAL,
    order           Order OPTIONAL,
    filter          Filter OPTIONAL
} -- End of RequestDataobjectsReq

RequestDataobjectsResp ::= SEQUENCE {
    applicationId  ITSaid,
    dataObjectType DataObjectType OPTIONAL,
    requestedData  RequestedDataObjects OPTIONAL, -- Mandatory if the Result parameter is set to
the value "Successful"
    result         RequestedDataObjectsResult,
    errorMessage  ErrorMessage
} -- End of RequestDataobjectsResp

-- SubscribeDataConsumer Information Flow
SubscribeDataobjectsReq ::= SEQUENCE {
    applicationId  ITSaid,
    dataObjectType DataObjectType,
    priority        UserPriority OPTIONAL,
    filter          Filter OPTIONAL,
    notifyTime     TimestampIts OPTIONAL,
    multiplicity    Multiplicity OPTIONAL,
    order           Order OPTIONAL
} -- End of SubscribeDataobjectsReq

SubscribeDataobjectsResp ::= SEQUENCE {
    applicationId  ITSaid,
    subscriptionId SubscriptionId OPTIONAL, -- Mandatory if the Result parameter is set to the
value "Successful"
    result         SubscribeDataobjectsResult,
    errorMessage  ErrorMessage
} -- End of SubscribeDataobjectsResp

-- PublishDataobjects Information Flow
PublishDataobjects ::= SEQUENCE {
    subscriptionId SubscriptionId ,
    requestedData  RequestedDataObjects
} -- End of PublishDataobjects

-- UnsubscribeDataConsumer Information Flow
UnsubscribeDataobjectsReq ::= SEQUENCE {
    applicationId  ITSaid,
    subscriptionId SubscriptionId
} -- End of UnsubscribeDataobjectsReq

UnsubscribeDataobjectsResp ::= SEQUENCE {
    applicationId  ITSaid,
    result         UnsubscribeDataobjectsResult
} -- End of UnsubscribeDataobjectsResp

RegisterDataConsumerResult ::= ENUMERATED {
    accepted (0), -- Area of interest covered by LDM Area of Maintenance
    warning (1), -- Area of interest extends beyond LDM Area of Maintenance
    rejected (2)
} -- End of RegisterDataConsumerResult

DeregisterDataConsumerAck ::= ENUMERATED {
    succeed (0),
    failed (1)
} -- End of DeregisterDataConsumerAck

SubscriptionId ::= INTEGER (0..65535)

Multiplicity ::= INTEGER (0 .. 255)

RequestedDataObjectsResult ::= ENUMERATED {
    successful (0),
    invalidITSaid (1),
    invalidDataObjectType(2),
    invalidPriority (3),
    invalidFilter (4),
    invalidOrder (5)
} -- End of RequestedDataObjectsResult

SubscribeDataobjectsResult ::= ENUMERATED {
    successful (0),
    invalidITSaid (1),

```



```

    invalidDataObjectType(2),
    invalidPriority (3),
    invalidFilter (4),
    invalidNotificationInterval (5),
    invalidMultiplicity (6),
    invalidOrder (7)
} -- End of SubscribeDataobjectsResult

UID ::= INTEGER

ReferenceArea ::= CHOICE {
    geometricArea      GeometricArea,
    relevanceArea      RelevanceArea
}

GeometricArea ::= CHOICE {
    circle      Circle,
    rectangle   Rectangle,
    ellipse     Ellipse
} -- End of GeometricArea

Circle ::= SEQUENCE { -- reference EN 302 931
    radius Distance
} -- End of Circle

Rectangle ::= SEQUENCE { -- reference EN 302 931
    aSemiAxis Distance,
    bSemiAxis Distance,
    azimuthAngle Direction -- aSemiAxis directionality
} -- End of Rectangle

Ellipse ::= SEQUENCE { -- reference EN 302 931
    aSemiAxis Distance,
    bSemiAxis Distance,
    azimuthAngle Direction -- aSemiAxis directionality
} -- End of Ellipse

Direction ::= INTEGER{north(0), east(7200), south(14400), west(21600)} (0..28799)

Distance ::= INTEGER (0 .. 65535) -- multiples of 1.0m

RelevanceArea ::= SEQUENCE { -- reference EN 302637_3
    relevanceDistance      RelevanceDistance,
    relevanceTrafficDirection RelevanceTrafficDirection
}

--the reference position of the AreaOfInterest is the host station location
AreaOfInterest ::= GeometricArea

DataObjectType ::= ENUMERATED {
    cam (0),
    denm (1),
    ... -- Any other Data Object Type such as defined in the Common
} -- End of DataObjectType

UserPriority ::= INTEGER (0..255)

Order ::= SEQUENCE OF OrderTuple

OrderTuple ::= SEQUENCE {
    attribute      Attribute,
    orderingDirection OrderingDirection
} -- End of OrderTuple

OrderingDirection ::= ENUMERATED {
    ascending (0),
    descending (1)
} -- End of OrderingDirection

RequestedDataObjects ::= SET OF DataContainer

UnsubscribeDataobjectsResult ::= ENUMERATED {
    accepted (0),
    rejected (1)
} -- End of UnsubscribeDataobjectsResult

ErrorMessage ::= IA5String (SIZE(0 .. 65535))

```

```

-- Filtering

Filter ::= CHOICE {
    filterStatement FilterStatement,
    filterExp       FilterExp
} -- End of Filter

FilterExp ::= SEQUENCE {
    filter1      Filter,
    logicalOperator LogicalOperators,
    filter2      Filter
} -- End of FilterExp

FilterStatement ::= SEQUENCE {
    attribute      Attribute,
    operator       ComparisonOperators,
    refValue       ReferenceValue
} -- End of FilterStatement

ComparisonOperators ::= ENUMERATED {
    equal (0),
    notequal (1),
    gt (2),
    lt (3),
    gte (4),
    lte (5),
    like (6),
    notlike (7)
} -- End of ComparisonOperators

LogicalOperators ::= ENUMERATED {
    and (0),
    or (1)
} -- End of LogicalOperators

-- Attribute is a Data Element from the Common Data Dictionary ts (102894-2) cdd
Attribute ::= OCTET STRING (SIZE(0..65535))

ReferenceValue ::= CHOICE {
    boolValue BOOLEAN,           -- Basic types
    sbyteValue INTEGER (0 .. 127), -- 7 bits
    byteValue INTEGER (0 .. 255), -- 8 bits
    shortValue INTEGER (0 .. 32765), -- 16 bits
    intValue INTEGER,
    octsValue OCTET STRING,
    bitsValue BIT STRING,
    strValue IA5String (SIZE(0 .. 65535)),
    causeValue CauseCode,       -- ITS types
    speedValue Speed,
    stationIDValue StationID,
    ...
} -- End of ReferenceValue

END -- End of ITSldm

```

---

## Annex C (informative): Bibliography

- ETSI TS 101 539-2: "Intelligent Transport Systems (ITS); V2V Application; Part 2: Intersection Collision Risk Warning (ICRW) Application Specification".
- ETSI TS 101 539-3: "Intelligent Transport Systems (ITS); V2V Application; Part 3: Longitudinal Collision Risk Warning (LCRW) Application Specification".
- ETSI TS 102 894-1: "Intelligent Transport Systems (ITS); User and Applications requirements; Part 1: Facility layer structure, functional requirements and specifications".
- OpenLR White Paper Version 1.3 (05.2010): "An open standard for encoding, transmitting and decoding location references in digital maps".
- ISO/IEC 8825-2:2008: "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)".

---

## History

<b>Document history</b>				
V1.0.0	January 2014	EN Approval Procedure	AP 20140522:	2014-01-22 to 2014-05-22
V1.1.0	July 2014	Vote	V 2040920:	2014-07-22 to 2014-09-22