INTERNATIONAL STANDARD

ISO 12855

First edition
2012-02-15

Electronic fee collection — Information exchange between service provision and toll charging

Perception du télépéage — Échange d'informations entre la prestation de service et la perception du péage

Reference number
ISO 12855:2012(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 12855 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*, in collaboration with Technical Committee CEN/TC 278, *Road transport and traffic telematics*.

Introduction

The widespread use of tolling also requires provisions for users of vehicles that are circulating through many different toll domains. Users should be offered a single contract for driving a vehicle through various toll domains. Where those vehicles require a form of on-board equipment (OBE) this should be interoperable with the toll systems in the various toll domains. In Europe, for example, this need has been officially recognized and legislation on interoperability has already been adopted (see Directive 2004/52/EC). There is both a commercial and economic justification in respect to the OBE and the toll systems for standards enabling interoperability.

The system architecture defined in ISO 17573 is the basis for all standards that relate to tolling systems in the toll domain. From this system architecture standard, other standards have consistently reused

— common definitions of terms and concepts and basic system functionalities and structure,

— common terminology, and

— identified interfaces that are or need to be defined.

ISO 17573 uses ISO/IEC 10746-3 for the description of the architecture.

The following figure shows the scope of the group of electronic fee collection (EFC) related standards based upon the architecture standard.

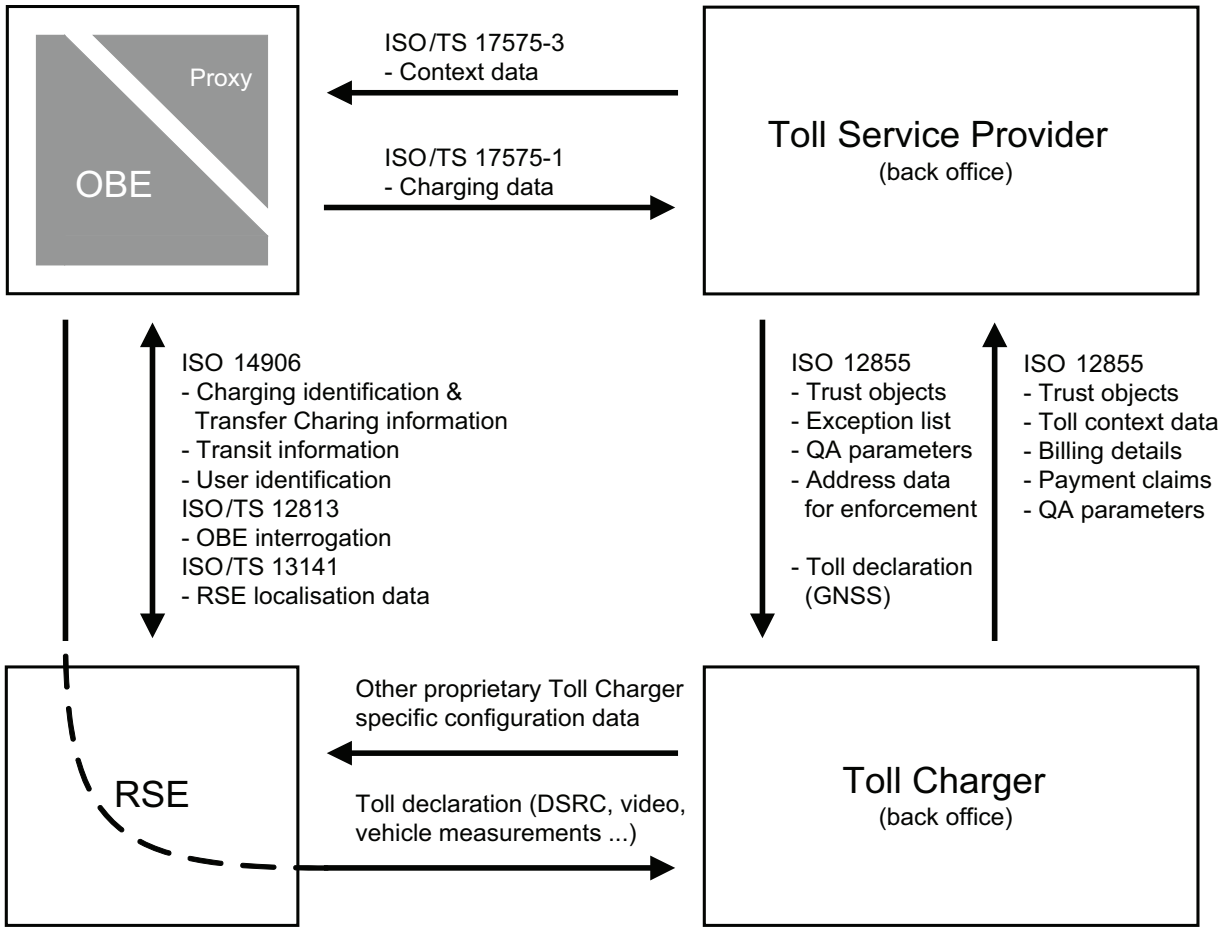


Figure 1 — Scope of EFC related standards

A given transport service for a given vehicle is fully identified by one or several toll declarations, made available to the Toll Charger. Toll declarations have to be made available according to the rules of the toll regime of the toll domain.

The amount due for a given transport service used by a vehicle liable to toll is concluded by the Toll Charger (TC) with the use of toll declarations (as described above) and calculation is made according to the rules of the toll regime (formula, tariff tables, specific situations rules, traffic conditions, etc.).

The information above, associated with a given transport service, is named billing details; for a given transport service, the billing details are referring to one or several toll declarations.

Depending on the toll regime, billing details are elaborated with information collected by the Toll Charger and/or the relevant Toll Service Provider (TSP); they are concluded by the toll charger.

The Toll Charger elaborates and makes the payment claims (or toll payment claims) available to each Toll Service Provider, according to the bilateral agreements it has with each Toll Service Provider, referring to billing details. These payment claims include an amount due taking into account any specific commercial conditions applicable to a vehicle, a fleet of vehicles or a given Toll Service Provider.

This International Standard identifies and specifies the set of messages exchanged between two actors in the roles of Toll Service Provider and Toll Charger as defined in ISO 17573. To specify these interfaces, this International Standard uses the enterprise description of the toll environment, and the interactions defined between the named classes of roles, as defined in ISO 17573. This allows for a complete specification of the data that is transferred between those identified entities. In addition to that, a number of computational interfaces are identified, where interactions in terms of sequences of messages are defined.

Electronic fee collection — Information exchange between service provision and toll charging

1 Scope

This International Standard specifies

— the interfaces between electronic fee collection (EFC) systems for vehicle related transport services, e.g. road user charging, parking and access control; it does not cover interfaces for EFC systems for public transport; an EFC system can include any EFC system, e.g. also systems automatically reading licence plate numbers of vehicles passing a toll point;

— an exchange of information between the central equipment of the two roles of service provision and toll charging, e.g.

  — charging related data (toll declarations, billing details),

  — administrative data, and

  — confirmation data;

— transfer mechanisms and supporting functions;

— information objects, data syntax and semantics;

— examples of data interchanges.

This International Standard supports any toll service and any technology used for charging.

It is defined as a toolbox standard of transactions and messages which can be used for the assigned purpose. The detailed definitions of mandatory and optional elements in a real implementation are defined elsewhere. It does not define all communication sequences, communication stacks and timings.

The scope of this International Standard is illustrated in Figure 2. The data types and associated coding related to the data elements described in Clause 6 are defined in Annex A, using the abstract syntax notation one (ASN.1) according to ISO/IEC 8824-1.

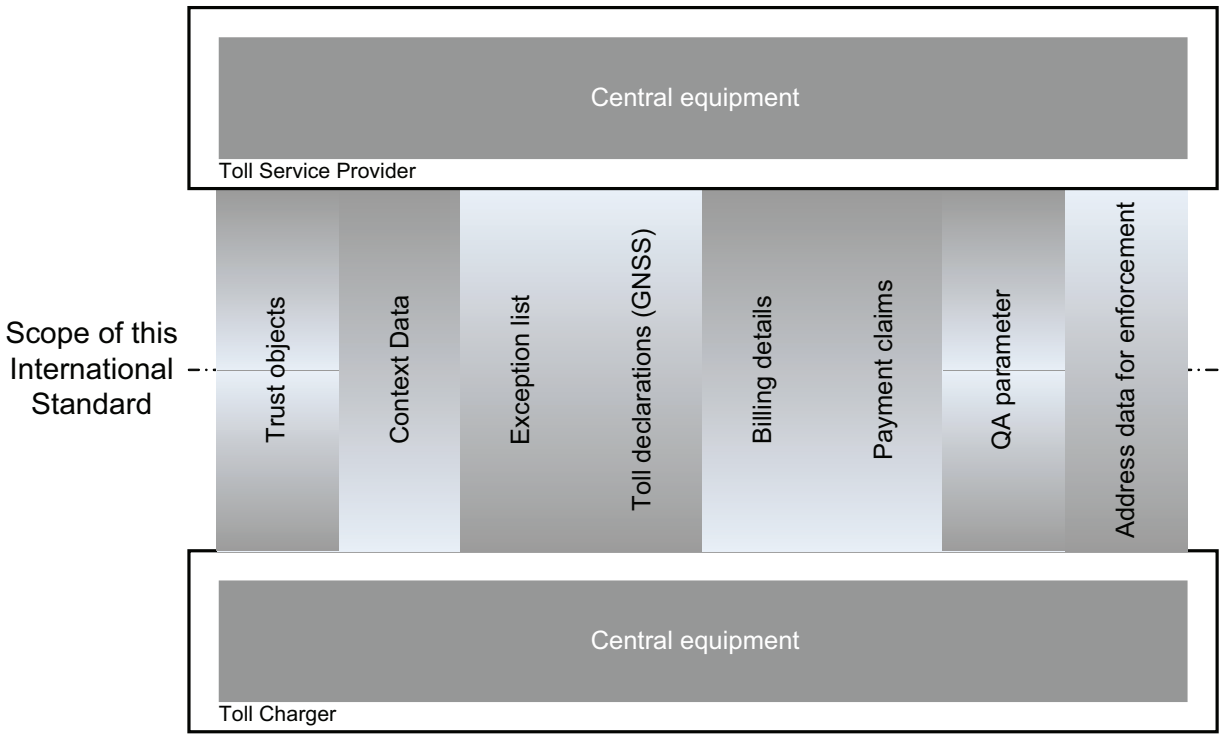


Figure 2 — Scope of this International Standard

Any communication between Toll Charger and/or Toll Service Provider with any other involved party is outside the scope of this International Standard. Any communication between elements of the Toll Charger and the Toll Service Provider which is not part of the back office communication is outside the scope of this International Standard.

The processes regarding the payments and exchanges of fiscal, commercial or legal accounting documents are outside the scope of this International Standard.

The definitions of service communication channels, protocols and service primitive to actually transfer the messages are outside the scope of this International Standard.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 17573, *Electronic fee collection — System architecture for vehicle-related tolling*

ISO 14906, *Electronic fee collection — Application interface definition for dedicated short-range communication*

ISO/TS 17575-1, *Electronic fee collection — Application interface definition for autonomous systems — Part 1: Charging*

ISO/TS 17575-3, *Electronic fee collection — Application interface definition for autonomous systems — Part 3: Context data*

ISO/TS 17575-4, *Electronic fee collection — Application interface definition for autonomous systems — Part 4: Roaming*

ISO/IEC 9646-7, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 7: Implementation Conformance Statements*

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 8825-4, *Information technology — ASN.1 encoding rules: XML Encoding Rules (XER)*

ISO 639-1, *Codes for the representation of names of languages — Part 1: Alpha-2 code*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
billing detail
for a given transport service, all necessary data required to determine and/or verify the amount due for the service user

NOTE 1 If the data is accepted by both the Toll Charger and the Toll Service Provider then it is called a concluded billing detail which can be used to issue a payment claim.

NOTE 2 For a given transport service, the billing detail is referring to one or several valid toll declaration(s). A valid billing detail has to fulfil formal requirements, including security requirements, agreed between the Toll Service Provider and the Toll Charger.

3.2
charge report
data structure transmitted from the front end to the Back End to report road usage data and supplementary related information

NOTE In 2009/750/EC charge report is referred to as “toll declaration”.

3.3
charging data
toll relevant data produced by the on-board equipment and sent to the Toll Service Provider's back-office systems

3.4
computational specification
decomposition of a system into objects performing individual functions and interacting at well defined interfaces

3.5
context data
information defined by the responsible Toll Charger necessary to establish the toll due for circulating a vehicle on a particular toll domain and to conclude the toll transaction

[ISO 17573, definition 3.1]

3.6
customer
person or legal entity that uses the service of a Toll Service Provider

[ISO 17573, definition 3.2]

NOTE Depending on the local situation, the customer can be the owner, lessor, lessee, keeper, (fleet) operator, holder of the vehicle's registration certificate, driver of the vehicle, or any other third person.

3.7
driver
person who drives a vehicle

[ISO 17573, definition 3.3]

NOTE The driver is assumed to operate (use/serve) the on-board equipment (e.g. the setting of the number of axles).

3.8
electronic fee collection
EFC
toll charging by electronic means via a wireless interface

NOTE 1 Adapted from ISO 17573, definition 3.4.

NOTE 2 The actual payment (collection of the fee) may take place outside the toll system.

3.9
enforcement
process of compelling observance of a law, regulation, etc.

[ISO 17573, definition 3.5]

NOTE In this context: the process of compelling observance of a toll regime.

3.10
interface
abstraction of the behaviour of an object that consists of a subset of the interactions of that object together with a set of constraints on when they may occur

[ISO/IEC 10746-2]

3.11
interoperability
ability of systems to provide services to, and accept services from, other systems and to use the services so exchanged to enable them to operate effectively together

[ISO 17573, definition 3.7]

NOTE For tolling, interoperability aims at enabling a vehicle to drive through various toll domains while having only one on-board equipment operating under one contract with a Toll Service Provider.

3.12
on-board equipment
OBE
equipment fitted within or on the outside of a vehicle and used for toll purposes

[ISO 17573, definition 3.9]

NOTE The OBE does not need to include payment means.

3.13
one(s) liable for toll
person(s) or legal entities liable to pay toll under the operation of a toll regime

[ISO 17573, definition 3.10]

NOTE A toll regime can designate more than one person to be (jointly and severally) liable for paying the toll.

3.14
payment claim
recurring statement referring to concluded billing details made available to the Toll Service Provider by the Toll Charger who indicated and justified the amount due

NOTE The payment claim is used by the Toll Service Provider to issue financial objects to its customers (e.g. invoices on behalf of the Toll Charger). A given toll payment claim is referring to billing details and takes into account any specific commercial conditions applicable to a vehicle, a fleet of vehicles, a customer of a Toll Service Provider and/or a Toll Service Provider. A valid “payment claim” has to fulfil formal requirements, including security requirements, agreed between the Toll Service Provider and the Toll Charger.

3.15
roadside equipment
RSE
equipment located along the road transport network, for the purpose of communication and data exchanges with on-board equipment

[ISO 14906, definition 3.1]

3.16
service user
see **user** (3.29)

3.17
tariff scheme
set of rules to determine the fee due for a vehicle in a toll domain for a tolled object at a certain day and time

[ISO 17573, definition 3.14]

EXAMPLE A table that shows the fee for various classes of vehicles.

3.18
toll
charge, tax, fee, or duty in connection with using a vehicle within a toll domain

[ISO 17573, definition 3.15]

NOTE The definition is a generalization of the classic definition of a toll as “a charge, a tax, or a duty for permission to pass a barrier or to proceed along a road, over a bridge, etc.”. The definition above also includes fees regarded as an (administrative) obligation, e.g. a tax or a duty.

3.19
Toll Charger
TC
legal entity charging toll for using a vehicle within a toll domain

[ISO 17573, definition 3.16]

NOTE In other documents the terms “operator” or “toll operator” can be used.

3.20
toll declaration
statement to a Toll Charger that confirms the presence of a vehicle in a toll domain in a format agreed between the Toll Service Provider and the Toll Charger

[ISO 17573, definition 3.17]

NOTE A valid toll declaration has to fulfil formal requirements, including security requirements, agreed between the Toll Service Provider and the Toll Charger.

3.21
toll domain
area or part of a road network where a toll regime is applied

[ISO 17573, definition 3.18]

3.22
toll point
location within a toll domain where the on-board equipment has to issue a toll declaration

[ISO 17573, definition 3.19]

EXAMPLE A part of a toll plaza for electronic fee collector.

3.23
toll regime
set of rules, including enforcement rules, governing the collection of a toll in a toll domain

[ISO 17573, definition 3.20]

3.24
toll service
service enabling users having only one contract and one set of on-board equipment to use a vehicle in one or more toll domains

[ISO 17573, definition 3.22]

3.25
Toll Service Provider
TSP
legal entity providing to its customers toll services on one or more toll domains for one or more classes of vehicles

[ISO 17573, definition 3.23]

NOTE 1 In other documents the terms “issuer” or “contract issuer” can be used.

NOTE 2 The Toll Service Provider can provide the on-board equipment or can provide only a magnetic card or a smart card to be used with on-board equipment provided by a third party.

NOTE 3 The Toll Service Provider is responsible for the operation (functioning) of the on-board equipment.

3.26
toll system
off-board equipment and possible other provisions used by a Toll Charger for the collection of toll for vehicles

NOTE 1 The on-board equipment is excluded from the definition. On-board equipment should be part of any toll system for which it may be used.

NOTE 2 The actual payment (collection of the fee) may take place outside the toll system.

3.27
tolled object
distinguished part of a toll domain for which one or more tariff schemata apply

EXAMPLE An area, all public roads within an area, a bridge, a zone, or a stretch of a road (network).

3.28
trust object
information object that is exchanged between entities to ensure mutual trust

EXAMPLE An electronic signature or an electronic certificate.

3.29
user
customer of a Toll Service Provider, one liable for toll, the owner of the vehicle, a fleet operator, a driver etc. depending on the context

[ISO 17573, definition 3.26]

4 Symbols and abbreviated terms

ADU	Application Data Unit
ANPR	Automatic Number Plate Reading
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit (ISO 14906)
CCC	Compliance Check Communication (ISO/TS 12813)
CRL	Certificate Revocation List
DSRC	Dedicated Short Range Communication (ISO 14906)
EFC	Electronic Fee Collection (ISO 17573)
GNSS	Global Navigation Satellite System
ICS	Implementation Conformance Statement
IEC	International Electrotechnical Commission
IUT	Implementation Under Test
ITU	International Telecommunication Union
LPN	Licence Plate Number
OBE	On-Board Equipment (ISO 14906)
OBU	On-Board Unit
OCSP	Online Certificate Status Protocol
OSI	Open Systems Interconnection
PAN	Personal Account Number (ISO 14906)
PICS	Protocol Implementation Conformance Statement
QA	Quality Assurance
RSE	Roadside Equipment (ISO 14906)
SLA	Service Level Agreement
SU	Service User
SUT	System Under Test (ISO 14907-1)
TC	Toll Charger
TSP	Toll Service Provider

5 Architectural concept

5.1 Main roles in the Toll Charging environment

This International Standard is built upon ISO 17573. ISO 17573 defines the four main roles shown in Figure 3.

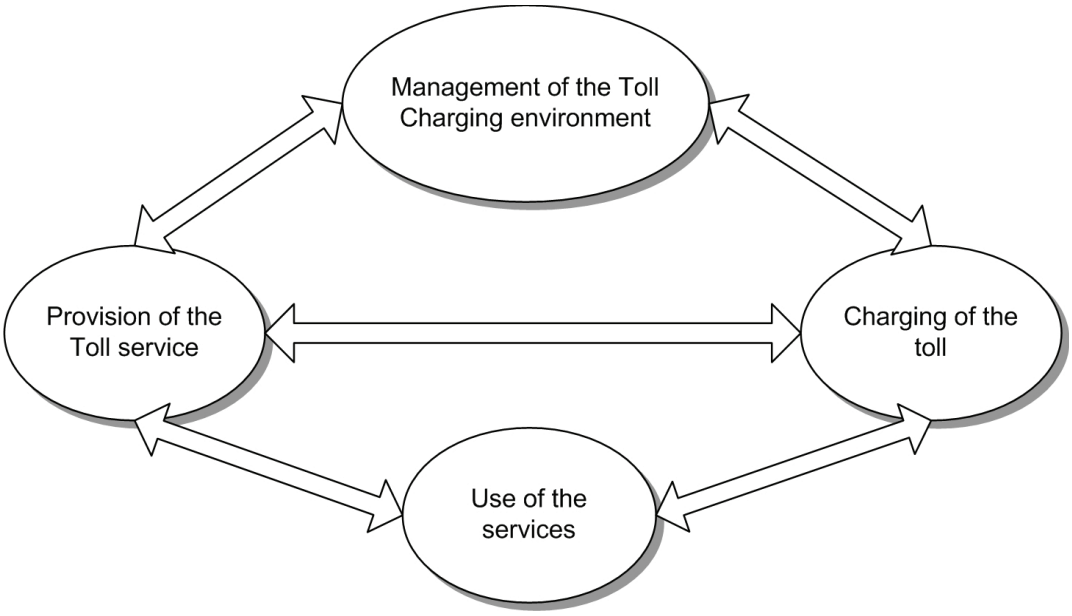


Figure 3 — Roles in the Toll Charging environment

Information exchanges are agreed upon between Toll Charger and Service Provider also taking into account privacy regulations. The information exchanges needed by the Toll Charger and the Toll Service Provider to perform their roles are described in this clause.

5.2 Information exchange between Toll Charging and Provision

5.2.1 General

The information exchange between the Service Provision and the Toll Charging roles supports the provision of the following functionalities, which are all based on the EFC system behaviour definitions in ISO 17573:

— Exchange Trust Objects

— Originating and providing EFC context data

— Manage Exception list

— Report Toll declarations

— Report Billing details

— Claim payment for service usage

— Exchange Enforcement data

— Exchange Quality assurance parameters

This International Standard leaves implementers the freedom of defining suitable protocol procedures, i.e. for complex transactions, hence it only defines:

— A basic interaction protocol (request – response) for information exchange

— Basic protocol mechanisms, to be used to build more complex protocol procedures

— The semantics and the format of the messages that are exchanged

The following subclauses describe the functionalities listed above.

5.2.2 Basic interaction protocol

Information exchanges happen by means of Application Protocol Data Unit (APDU) transfers.

Some APDU transfers need to be acknowledged. When this happens, related protocol procedures are specified. This International Standard defines no provisions for complex transfers (transactions), i.e. APDU transfers that cover several APDUs. Instead, this International Standard defines basic protocol mechanisms, to be used by implementations that need to identify transactions.

5.2.3 Basic protocol mechanisms

5.2.3.1 General approach

This International Standard provides the following basic protocol mechanisms, which shall be implemented to exchange information between the Toll Service Provider's and the Toll Charger's central equipment. These basic protocol mechanisms consist of:

— An identification schema for the messages that are exchanged.

— A generic interaction (i.e. not related to any specific functionality) that allows requesting a specific information exchange from the counterpart. This interaction is provided by the "Request" message.

— A generic acknowledge mechanism (i.e. not related to any specific functionality) that allows acknowledging a specific interaction. This mechanism is provided by the "Acknowledge" message.

— A generic status mechanism (i.e. not related to any specific functionality) that allows providing status information for a specific interchange. This mechanism is provided by the "Status" message.

By means of the above mechanisms, an implementation can build more complex protocol procedures, including rollback, recovery, checkpointing or restart.

This International Standard does not specify timings and retry procedures for acknowledgements. Timeouts can be defined as agreements between Toll Charger and Toll Service Provider to cover the case of missing acknowledgments. To handle any lost messages a timeout system can be implemented.

5.2.3.2 Identification schema

Each interaction is performed by means of one or more message exchanges. Each message shall contain a unique identifier in the namespace of the originator of the message. The combination of originator identifier and message identifier ensures that all messages are uniquely identified.

5.2.3.3 Request message

The Request message may be used to:

— Alert the counterpart that one is ready to accept any kind of information exchange.

— Inform the counterpart that one is ready to accept a specific type of message, by indicating the type of message one is ready to accept.

— Request the counterpart to re-issue a specific message, by indicating the type and the identifier of the message.

— Request for information identified by the type and message content.

5.2.3.4 Acknowledge message

The “Acknowledge” message is used whenever a specific message in an information exchange needs to be confirmed. The “Acknowledge” message indicates the specific message to be acknowledged by specifying its identifier. It may additionally carry an indication of a positive or negative acknowledgment.

5.2.3.5 Status message

The “Status” message is used to provide the counterpart general status information on the interface or inform about status on previously transferred information. It may be used to:

— Provide general information on the status of the interface.

— Alert the counterpart that some previously provided information becomes invalid without any new information being currently available.

— Alert the counterpart that the previous information contained an error and has to be recalled.

5.2.4 Exchange Trust Objects

The “Exchange Trust Objects” functionality is derived from the EFC system behaviours “Adding (or excluding) a new Toll Charger” and “Adding (or excluding) a new Service Provider”. Actions performed when executing the above behaviours shall exchange Trust Objects to be used in order to secure their bilateral communication. The functionality may also be used whenever an entity sees the need to update its own Trust Objects or another entity may ask for the update of an already existing Trust Object.

NOTE Examples of Trust Objects are:

— Asymmetric public keys

— Certificates

— Symmetric Keys

— Certificate Revocation Lists

The Exchange Trust Objects functionality can be used by either the Toll Service Provider or Toll Charger.

The initiator of the exchange can request the sending of Trust Objects from the receiver by sending a “Request” message. The “Request” message may contain an optional indicator that specifies already issued Trust Objects. When no indicator is specified, the Trust Objects to be transferred shall be the current ones.

After receiving a “Request” message that asks for Trust Objects, the requested Trust Objects are generated (or retrieved) by the recipient of the request and sent to the requesting party who shall acknowledge the reception by issuing an “Acknowledge” message.

After acknowledgement, the exchanged Trust Objects shall be considered immediately valid unless they contained a validity starting date. The validity period in this last case starts from the time indicated in the Trust Objects.

5.2.5 Originating and providing EFC context data

The “Originating and providing EFC context data” functionality is derived from ISO 17573 as defined in the EFC system behaviour “Adding, modifying or closing a toll regime”.

The usage of the “Originating and providing EFC context data” functionality may be used by a Toll Charger when any change of a toll domain or toll regime occurs, including the start of a new toll domain, by issuing an “EFC Context Data” message.

Any Toll Service Provider may request from any Toll Charger at any time for any reason to send the current or any previous version of the toll context data for a toll domain under its responsibility. This operation is performed by means of a “Request” message.

Reception of an “EFC Context Data” message shall be acknowledged by means of an “Acknowledge” message.

The information describing a toll regime uses one or more sets of EFC Context Data. It is defined through its tolled objects and the rules associated with them. These data elements are defined in detail in ISO/TS 17575-3 and extended in this International Standard for DSRC systems and other extensions.

Other toll regime properties to be configured by the Toll Charger are the interrelations a toll regime may have in relation to others. These rules and configuration parameters are defined in ISO/TS 17575-4 and are addressed within this International Standard as Toll Context interrelations.

5.2.6 Manage Exception list

5.2.6.1 General

The “Manage Exception list” functionality originates from ISO 17573 defined in the EFC system behaviours “Collecting toll information – User billing” and “Collecting charging information (autonomous systems)”.

NOTE 1 To avoid the term of blacklist, which has a different meaning in various existing EFC systems, and to include another list with a similar meaning [e.g. grey list or black list of Personal Account Numbers (PAN, as defined in ISO 14906), list of blocked license plates, white list, etc.], this International Standard uses the term “Exception list” to summarize all possibilities of limiting the usability of an OBE or giving information on the special handling of an OBE in a toll regime. Other standards may still use differing terms, but they are all included in the term “Exception list”.

NOTE 2 The conditions and the periods of time of the acceptance of an OBE within a toll regime are limited, by putting it on the Exception list or removing it. This is solely the responsibility of the Toll Service Provider that issued the OBE. Any information sufficient for the identification of a specific vehicle or OBE by the Toll Charger (e.g. OBE ID, PAN, license plate) may be included in the Exception list as agreed between TC and TSP.

5.2.6.2 Exception list entry requested by a Toll Charger

The “Manage Exception list” functionality may be used by a Toll Charger when it registers violations by a specific Service User or wrong technical behaviour by a specific OBE.

NOTE 1 In this case the Toll Charger may issue a “Report Abnormal OBE” message to request the inclusion of this OBE in the Exception list.

The Toll Service Provider shall acknowledge the decision to include the OBE on the Exception list by means of an “Acknowledge” message.

NOTE 2 This may be due to non-conforming behaviour of the Service User or of a malfunctioning OBE or others.

5.2.6.3 Exception list entry decided by the Toll Service Provider

A Toll Service Provider can unilaterally add, modify or delete items in its Exception list.

The functionality is performed by issuing the “Exception List” message. This offers the Toll Service Provider the opportunity to provide any information about an OBE of a Service User to the Toll Charger.

NOTE 1 The message can include among others one of the following reasons:

— Toll Service Provider has terminated its support/responsibility for a vehicle/OBE

— An OBE was lost or stolen

— The Toll Service Provider has started/accepted its support/responsibility for a vehicle/OBE

— The Toll Charger is informed about the commercial conditions to apply to an OBE (e.g. discount for a group of vehicles)

NOTE 2 The Exception list can be used to provide additional information on a vehicle/OBE for a toll regime (e.g. specific commercial conditions) and/or limit or restrict the acceptance of an OBE within a toll regime operated via the road infrastructure of a Toll Charger, where an exchange of data between Toll Service Provider and Toll Charger is needed.

Upon reception the Toll Charger may semantically check the received Exception list. If an error is detected in the Exception list, the whole Exception list shall be disputed by sending an “Acknowledge” message indicating the detected error.

The Toll Service Provider may then rectify the problem and transmit a new Exception list. Until a valid Exception list is transmitted, the last correct list remains active in the systems of the Toll Charger.

If a new valid Exception list is received by the Toll Charger it shall be acknowledged by sending an “Acknowledge” message.

5.2.7 Report Toll declarations

The “Report Toll declarations” functionality originates from ISO 17573 defined in the EFC system behaviour “Collecting charging information (autonomous systems)”.

NOTE 1 The Charging data generated by an OBE is used to report a Service User entering, moving around in or leaving a toll domain. A Service usage statement with an amount due can be made either by a single tolled object or by a combination of several tolled objects. Any Service usage is reported as Charging data through an exchange of data between an OBE/proxy (Front End system) and the central equipment (Back End system) managed by a Toll Service Provider. This interface between Front End and Toll Service Provider is specified in ISO/TS 17575 Parts 1 to 4 and is not covered by this International Standard.

The gathered Charging data shall be collected in the central system of the Toll Service Provider. If the Toll Service Provider needs to enrich the Charging data in its central equipment, it may do so before sending it as Toll declarations to the Toll Charger who offered the transport service. This optional possibility to enrich the Charging data enables the concept of shared user data, where only limited information may be included in the OBE, while the rest is held centrally at the issuing Toll Service Provider.

The Toll declarations shall contain all information required by the Toll Charger to calculate the amount due for the use of a toll domain or verify the calculation done by the Toll Service Provider. Details about configuration parameters for Charge reports are defined by the Toll Charger in the EFC context data.

A toll declaration shall be reported to the Toll Charger operating a toll domain, by means of a “Toll Declaration” message.

NOTE 2 The toll declarations may be delivered periodically in an agreed frequency (e.g. weekly, daily, hourly, in real time …) or upon triggering the delivery and with the quantity of information agreed for a toll regime.

The Toll Charger shall acknowledge the received “Toll Declaration” message by indicating for each toll declaration whether it is accepted or not by means of an “Acknowledge” message. If a toll declaration is not accepted, this has to be handled in a dispute phase, which is a process and thus outside the scope of this International Standard.

If the Toll Charger detects a contradiction between the Toll Declarations provided by the Toll Service Provider and its own data (e.g. CCC data, LPN reading, etc.), it may ask the Toll Service Provider for additional information about the provided Toll Declarations for this specific vehicle or Service User. The Toll Charger

may send a “Retrieve specific Toll Declarations” message to the Toll Service Provider to provide any detailed Toll Declarations for a specific Service User and/or a specific period of time.

NOTE 3 This enables the Toll Charger to receive only daily summation records of the use of its toll domain from the Toll Service Provider to produce its Billing details in normal operation without any detailed knowledge about each segment passed by an OBE. When the Toll Charger detects a contradiction in the provided high level Toll Declarations during comparison with the CCC data it recorded, it may ask the Toll Service Provider for detailed Toll Declarations for this specific vehicle or Service User.

5.2.8 Report Billing details

Depending on its toll system a Toll Charger either acquires the toll declarations directly from the RSE it operates (e.g. DSRC-based systems) or via the Toll Service Provider's central equipment (e.g. autonomous systems), which are then used for the generation of Billing details.

A single Billing detail may refer to one or several Toll declarations. The generation of Billing details is based on the requirements defined for the toll regime. Thus a Billing detail may be built out of:

— an elementary usage of a transport service (e.g. regarding the toll for a road section)

— several usages of a transport service within a given period of time (a day, a week, a month …)

— several elementary usages of a transport service within a given journey

If some relevant information is missing to build a Billing detail out of Toll declarations, it may be requested from the central equipment of the Toll Service Provider.

NOTE 1 This optional possibility to enrich the Toll declarations enables the concept of shared user data, where only limited information may be included in the OBE, while the rest is held centrally at the issuing Toll Service Provider.

NOTE 2 The Toll Charger and Toll Service Provider can agree to aggregate the generated Billing details to reduce the number of lines to be processed in their bookkeeping systems for handling the payment claim between Toll Charger and Toll Service Provider. This can be achieved by one of the following measures among others:

— After generating the Billing details the Toll Charger aggregates any Billing details prior to claiming the payment if this is bilaterally agreed. A unique identifier (Reference number) for each aggregate is generated during aggregation and associated with all Billing details in order to link them to the derived payment claim. By this, the Toll Service Provider is always able to check the consistency of the payment claim with the Billing details they stem from.

— The Toll Charger and Toll Service Provider can agree to use the same aggregation process and aggregate the Billing details independently from each other for the representation on the invoice in their own central equipment to avoid any rounding differences.

The generated Billing details shall be reported by means of a “Billing Detail” message.

The Toll Service Provider may check the completeness and the conformity of the provided Billing details for a given service usage. Therefore the Billing details may reference any Toll declarations directly acquired by the Toll Charger and thus provide information to the Toll Service Provider to check their authenticity [checking the Authenticator(s)].

The Toll Service Provider shall acknowledge the received “Billing Detail” message by indicating for each Billing detail whether it is accepted or not by means of an “Acknowledge” message.

5.2.9 Claim payment for service usage

The Claim payment for service usage functionality is derived from ISO 17573 defined in the EFC system behaviour “Claiming tolls”.

The claim payment phase starts once the Billing details have been agreed between the Toll Charger and the Toll Service Provider (see 5.2.8). The payment claim is based upon these agreed and possibly aggregated Billing details.

If any specific commercial conditions were agreed (to be applied to the payment claim of a Service User) this may be included directly as discount in the original payment claim or it may be included at a later time in a separate payment claim as a credit note (e.g. when billing details of a whole month define the discount level of a scaled discount scheme).

Any additional claims for other services which do not directly derive from the bilaterally agreed Billing details (e.g. belated payment, reimbursement, penalties ...) may be included as well.

The payment claim shall contain all the legal and fiscal information required according to each toll regime, to be used for invoicing between the Toll Charger, the Toll Service Provider and the Service User.

NOTE 1 The payment claim may be used by the Toll Service Provider to generate invoices to its customers (Service Users) on behalf of the relevant Toll Charger or for its own account (if the Toll Service Provider is reselling the toll).

The payment claim shall be put forward to the Toll Service Provider by means of a “Payment Claim” message.

The Toll Service Provider shall acknowledge the received “Payment Claim” message by indicating for each payment claim whether it is accepted or not by means of an “Acknowledge” message.

NOTE 2 Possible reasons for dispute, among others, are: Inconsistency of payment claim with Billing details or non-conformance of Specific Commercial conditions.

5.2.10 Exchange Enforcement data

5.2.10.1 General

The “Exchange Enforcement data” functionality originates from ISO 17573 defined in the EFC system behaviour “Exceptions detection – User and OBE compliance checking”.

The “Exchange Enforcement data” functionality may be used each time when a Toll Charger needs additional information for a compliance checking process. The compliance checking process is concentrated on:

— Verifying if the Service User fulfils its obligation to co-operate

— Gathering facts required for later performance monitoring and/or SLA evaluation

A following off-line process performed by the Toll Charger may:

— Issue toll violation tickets for the Service Users if the Service User was responsible for the wrong toll declaration

— Compare roadside observations with Toll declarations received from the Toll Service Provider and update the performance monitoring parameters with these results

Even if the details on how to perform compliance checking/enforcement Back End processes are left to the Toll Chargers, some basic information exchange shall be supported on both sides.

5.2.10.2 Retrieve User Details

The “Retrieve User Details” message may be used by a Toll Charger to request from Toll Service Provider(s) whether they have a contractual relationship with a given Service User identified in the message and/or to provide additional details on a Service User by specifying the requested details.

This message can either be sent to a specific Toll Service Provider or broadcasted to a group or all Toll Service Providers.

5.2.10.3 Provide User Details

After receiving a “Retrieve User Details” message the Toll Service Provider shall answer the request with a “Provide User Details” message, thus either confirming a contractual relationship or declining it.

As a minimum the Toll Service Provider shall confirm the existence of a contractual relationship with a Service User by providing one of the following data:

— Personal Account Number (PAN, as defined in ISO 14906)

— Contract serial number

— License plate number

— On-Board Equipment Identification (OBE ID)

If additional details on the Service User were requested in the “Retrieve User Details” message the Toll Service Provider shall deliver them.

NOTE The delivery of User Details may be subject to local data protection regulations.

5.2.10.4 Retrieve and Report CCC Event

The Toll Service Provider may send a “Retrieve specific CCC Event” message to the Toll Charger to enquire about the details for any CCC Events for a specific Service User and/or a specific period of time.

The Toll Charger shall answer to a received “Retrieve specific CCC Event” message with a “Report CCC Event” message or may send the “Report CCC Event” message of its own accord to inform the Toll Service Provider about any new CCC Event.

The Toll Service Provider shall acknowledge any “Report CCC Event” message by means of an “Acknowledge” message.

5.2.11 Exchange Quality assurance parameters

The Report Quality assurance parameters functionality is derived from ISO 17573 defined in the EFC system behaviour “Defining rules and monitoring operations”.

In ISO 17573 the Report Quality assurance parameters functionality is used to report and monitor the EFC activities of Toll Chargers and Toll Service Providers. In this International Standard it is used additionally to exchange any information necessary to monitor the quality of operations between the Toll Charger and the Toll Service Provider. One of these quality assurance parameters may be:

— Detection rate: As it is important for both the Toll Charger and the Toll Service Provider that the Detection rate (i.e. the percentage of detected OBE of a given Toll Service Provider in a given toll regime of a Toll Charger) of the Toll Service Provider's OBEs in a Toll regime of a Toll Charger be as high as possible, this may need to be constantly monitored by an SLA level.

Types and ranges of QA parameters are agreed upon between Toll Chargers and Toll Service Providers with bilateral agreements.

The exchange of Quality assurance parameters depends on the initiator of the exchange:

— The Toll Charger may send a Quality assurance parameter by issuing a “Report QA” message.

— The Toll Service Provider may request an update of a Quality assurance parameter by sending a “Request” message. The Toll Charger shall respond to the request by sending a “Report QA” message.

— The Toll Service Provider may send a Quality assurance parameter by issuing a “Report QA” message.

— The Toll Charger may request an update of a Quality assurance parameter by sending a “Request” message. The Toll Service Provider shall respond to the request by sending a “Report QA” message.

The receiver of a “Report QA” message shall acknowledge it by means of an “Acknowledge” message.

6 Computational specification

6.1 Overview

This clause specifies the computational objects for the information exchanges between Toll Chargers and Toll Service Providers. Each computational object is identified by:

— A name

— The function(s) it performs

— The Application Protocol Data Units (messages) it is able to generate or accept

For the sake of simplicity, the functionalities identified in previous clauses correspond to computational objects here. This means that the functions performed by each object are already described in previous clauses. A given computational object can be instantiated for the Toll Service Provider or for the Toll Charger role, with the limitations and permissions stated as rules in the previous clauses. Also, each computational object is modelled as having one interface where to exchange messages. By no means shall it be intended that computational objects herein described correspond to real processes or components in a real implementation.

Table 1 lists the defined computational objects. For each object, the relevant interface messages are listed, together with the limitations, permissions and obligations for each role derived by the rules defined in previous clauses.

While a real implementation can choose which computational objects are supported, if any of the computational objects listed in Table 1 are implemented, the defined rules applicable to the Toll Service Provider and the Toll Charger shall be implemented.

The rules are as follows:

— May initiate: An entity is able and allowed to initiate a message exchange

— Shall initiate: An entity is able and has to initiate a message exchange

— Shall be able to receive: An entity shall be able to receive this kind of message

— May respond: An entity is able and allowed to respond to a received Request message

— Shall respond: An entity is able and has to respond to a received Request message

The supplier of an implementation according to this International Standard shall complete the proforma protocol implementation conformance statement as defined in Annex B.

Table 1 — Computational objects and messages

Computational object	Message	Toll Service Provider rules	Toll Charger rules
Exchange Trust Objects	Request	May initiate, shall be able to receive	May initiate, shall be able to receive
	Trust Objects	May initiate, shall be able to receive	May initiate, shall be able to receive
	Acknowledge	Shall initiate, shall be able to receive	Shall initiate, shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Originating and providing EFC context data	Request	May initiate	Shall be able to receive
	EFC context data	Shall be able to receive	May initiate, may respond
	Acknowledge	Shall initiate	Shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Manage Exception List	Request	Shall be able to receive	May initiate
	Exception List	Shall initiate	Shall be able to receive
	Acknowledge (Exception list)	Shall be able to receive	Shall initiate
	Report abnormal OBE	Shall be able to receive	May initiate
	Acknowledge (Report abnormal OBE)	Shall initiate	Shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Toll Declaration	Retrieve specific Toll Declaration(s)	Shall be able to receive	May initiate
	Request	Shall be able to receive	May initiate
	Toll Declaration	Shall initiate, shall respond	Shall be able to receive
	Acknowledge	Shall be able to receive	Shall initiate
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive

Table 1 (*continued*)

Computational object	Message	Toll Service Provider rules	Toll Charger rules
Report Billing Details	Request	May initiate	Shall be able to receive
	Billing Details	Shall be able to receive	Shall initiate, shall respond
	Acknowledge	Shall initiate	Shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Claim Payment	Request	May initiate	Shall be able to receive
	Payment Claim	Shall be able to receive	Shall initiate, shall respond
	Acknowledge	Shall initiate	Shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Exchange Enforcement Data	Retrieve specific User Details	Shall be able to receive	May initiate
	Provide User Details	May initiate, shall respond	Shall be able to receive
	Retrieve specific CCC Event	May initiate	Shall be able to receive
	Report CCC Event	Shall be able to receive	May/Shall initiate
	Acknowledge (Report CCC Event)	Shall be able to receive	May initiate, shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Exchange Quality Assurance Parameters	Request	May initiate, shall be able to receive	May initiate, shall be able to receive
	Report QA	May initiate, shall be able to receive	May initiate, shall be able to receive
	Acknowledge	Shall initiate, shall be able to receive	Shall initiate, shall be able to receive
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive

The offering of an interface by an actor may require different behaviours depending on the activities to be performed with the exchange of information.

Interface messages summarized in Table 1 correspond to Application Protocol Data Units in the OSI Application Layer model.

The message content is formed by the following fields:

1) An Application Protocol Control Information (APCI) field
2) An Application Data Unit (ADU) field

Each message content shall contain one APCI field and one ADU field. Additionally the message contains one authenticator, placed outside of the message contents but inside the message.

The InfoExchange APDU comprises the following elements:

1) An infoExchangeContent field

2) An optional infoExchangeAuthenticator field

Table 2 indicates the fields in the infoExchange APDU which is defined in Annex A.

Table 2 — InfoExchange APDU fields

Field name	Data type	m/o
infoExchangeContent	InfoExchangeContent	m
infoExchangeAuthenticator	MessageAuthenticator	o

Each infoExchangeContent field shall contain one APCI field and one ADU field as specified in the InfoExchangeContent data type.

Table 3 indicates the fields in the InfoExchangeContent data type which is defined in Annex A.

Table 3 — InfoExchangeContent Data type fields

Field name	Data type	m/o
apci	ApciFields	m
adu	SEQUENCE OF AduContent	m

One adu field can contain one or more structures of the same data type, and is specified by an ASN.1 type SEQUENCE OF AduContent.

The following subclauses describe the APCI and the ADUs for the defined Application Protocol Data Units (Interface Messages).

6.2 Application Protocol Data Units

6.2.1 General

Any message (InfoExchange) may be signed by calculating a message authenticator over the combination of the APCI fields and the messages (ADUs). The resulting bit-string will be included as infoExchangeAuthenticator in the resulting message.

The infoExchangeAuthenticator is a dynamic object, and will be generated each time a new message is issued, as, for example, the APCI field messageDate is changed also for a resent message.

6.2.2 Application Protocol Control Information

The Application Protocol Control Information for all APDUs shall consist of the fields described in Table 4. For each field, an indication on whether the field is mandatory (m) or User option (o) is given.

Table 4 — APCI fields

Field name	Data type	m/o
messageOriginator	Provider	m
informationSenderID	Provider	m
informationRecipientID	Provider	m
contextID	RegimeID	o
messageIdentifier	MessageIdentifier	m
relatedMessageID	RelatedMessageID	o
aduType	AduType	m
numberOfStructs	NumberOfADUStruct	m
messageDate	GeneralizedTime	m

The messageOriginator shall be the identifier of the entity responsible for the content of the message. That may or need not be the Information Sender.

The informationSenderID identifies the entity sending the message. This entity shall be addressed when responding to this message.

The informationRecipientID shall be the identifier of the entity responsible for processing the content of the message.

The contextID specifies the EFC context for which this information is relevant. That may be one or more EFC contexts operated by the Toll Charger. The contextID shall be unique in the namespace of the Toll Charger.

The messageIdentifier is a unique identifier for each given Message Originator that is assigned by the Message Originator. It is used to either acknowledge or request retransmission of a previously issued message.

The relatedMessageId is an optional field that can be used to indicate that the message is a response to a previously received message from the MessageOriginator.

The aduType specifies the type of ADU contained within the message.

The numberofStruct indicates the number of repeating data structures contained in the ADU.

The messageDate is the date and time when the message was generated by the MessageOriginator.

6.2.3 Application Data Unit

The Application Data Unit (ADU) contains one or more data structures of the same type.

The ADUs listed in Table 5 are defined in this International Standard.

Table 5 — ADU Overview

ADU	Description
requestADU	Generic Request
ackADU	Generic Acknowledge
statusADU	Generic Status
trustObjectADU	Send Trust Objects
eFCContextDataADU	Send EFC Context Data
exceptionListADU	Send Exception List
reportAbnormalOBEADU	Report Abnormal OBE
retrieveTollDeclarationADU	Retrieve Specific Toll Declarations
tollDeclarationADU	Toll Declaration
billingDetailsADU	Billing Details
paymentClaimADU	Payment Claim
reportQAADU	Report QA
retrieveUserDetailsADU	Retrieve Specific User Details
userDetailsADU	Provide User Details
retrieveCCCEventADU	Retrieve Specific CCC Event
reportCCCEventADU	Report CCC Event

The specific format of these data structures are specified in the subclauses that follow.

6.3 RequestADU data structure

The requestADU is made of one data structure of type RequestADU. It is used to indicate to the recipient of the Request message the general readiness to receive, or the specific request to receive a specific data structure.

Table 6 indicates the fields in a RequestADU data structure which are defined in Annex A.

Table 6 — RequestADU fields

Field name	Data type	m/o
requestedADUType	AduType	o
aDUIdentifier	messageIdentifier	o
numberOfADUStructs	SEQUENCE OF NumberOfADUStruct	o

All fields in the RequestADU data structure are optional. If no fields are specified, the RequestADU message indicates a general readiness to receive.

If the requestedADUType field is specified, an ADU of the indicated data type is expected. It is used to indicate readiness to receive a new structure of the indicated type.

If the requestedADUType field is specified, the aDUIdentifier may be specified.

If the aDUIdentifier is specified, an ADU of the given type, which was previously transmitted and indicated by the aDUIdentifier field, is requested to be re-transmitted.

If the aDUIdentifier field is specified, a numberOfADUStructs field may be specified. If a numberOfADUStructs field is specified as n, the request is to re-transmit a previously transmitted data structure of the same type and with the same identifier, containing only the nth instance.

EXAMPLE A number of data structures of the type BillingDetails have been transferred in a previous data exchange. The receiver of that data structure needs to get one specific BillingDetails data structure, not the whole lot. It then issues a Request message that specifies the BillingDetails data type, the identifier of the previously received message, and the number that identifies the specific data structure in that message.

6.4 AcknowledgeADU data structure

The acknowledgeADU is made of one data structure of the AckADU data type. The data structure is used to indicate that a specific data structure has been received. It can optionally indicate acceptance or rejection of the previously received data structure, or of the whole message that transported the data structure.

Table 7 indicates the fields in an AckADU data structure which are defined in Annex A.

Table 7 — AckADU fields

Field name	Data type	m/o
aDUIdentifier	MessageIdentifier	m
numberOfADUStructs	SEQUENCE OF NumberOfADUStruct	o
ackCode	MessageReasonCode	m

aduCode	SEQUENCE OF SEQUENCE {
	numberOfADUStruct NumberOfADUStruct,
	aDUReasonCode ADUReasonCode
	}
		o

The aDUIdentifier field indicates the identifier of the message containing the data structure(s) being acknowledged.

The numberOfADUStructs field may optionally indicate the data structure in the message being explicitly acknowledged. A value of 0 specifies the whole message.

The ackCode field is a field that contains the reason for accepting or rejecting the identified data structure or message.

The aduCode is an optional field that can be used to respond to multiple instances of ADUs in the message being responded to.

EXAMPLE A number of data structures of the type BillingDetails have been transferred in a previous data exchange. The receiver of that data structure accepts one specific BillingDetails, and rejects all others. To do so, it issues an Acknowledge message identifying the message that contains as many Acknowledge data structures as the BillingDetails data structures previously received, indicating for each of them the reason for acceptance or rejection. If all Billing Details are accepted, the acceptance of the message is sufficient.

6.5 StatusADU data structure

The statusADU is made up of one data structure of the StatusADU data type. It can be used to indicate to the recipient that previously sent messages are obsolete (e.g. to support rollback scenarios) or to indicate the inability to receive messages.

Table 8 indicates the fields of the StatusADU data structure which are defined in Annex A.

Table 8 — StatusADU fields

Field name	Data type	m/o
generalStatusCode	GeneralStatusCode	m

messageStatusCode	SEQUENCE OF {
	message MessageIdentifier
	reasonOfADUStruct SEQUENCE OF SEQUENCE {
	numberOfADUStruct NumberOfADUStruct,
	reason MessageReasonCode
	}
	}
		o

The GeneralStatusCode is used to indicate readiness to receive messages.

The messageStatusCode is an optional field which can be used to provide message-specific Reason codes by ADU for messages that contain multiple ADUs.

6.6 TrustObjectsADU data structure

The Trust Objects Message can be used either by the Toll Charger or the Toll Service Provider to provide Trust Objects. The message can be sent either as a response to a Request message sent by the other party or initiated as a push message if new versions of Trust Objects are available.

The trustObjectsADU is made of one or more data structures of the TrustObjectADU data type. Each data structure may contain one or more trust objects.

Table 9 indicates the fields in a TrustObjectsADU data structure which are defined in Annex A.

Table 9 — TrustObjectADU fields

Field name	Data type	m/o
trustObjectID	INTEGER	m
startValidity	GeneralizedTime	o
endValidity	GeneralizedTime	o
trustObjectStatus	TrustObjectStatus	m
typeOfTrustObject	ListofTrustObjectTypes	m
purposesOfTrustObject	SEQUENCE OF ListOfTrustObjectPurposes	m
trustObject	TrustObjectCode	m

The trustObjectID field indicates the identifier for the Trust Object.

The startValidity is an optional field which can be used to indicate when the Trust Object is valid from. If this is not completed then it shall be assumed that it is valid with immediate effect.

The endValidity is an optional field which can be used to indicate the expiry date for a Trust Object. If this is not completed then it shall be assumed that there is no expiry data.

The trustObjectStatus field is an enumerated field used to indicate the status of the specified Trust Object.

The typeOfTrustObject field is an enumerated field used to specify the type of Trust Object.

The purposesOfTrustObject field is an enumerated field used to indicate the intended purpose of the Trust Object.

The trustObjectfield contains the definition of the Trust Object.

6.7 EFCContextDataADU data structure

The message EFC context data can be used by the Toll Charger either as a response to the Request message sent by the Toll Service Provider or as a push message if new versions of EFC context attributes are available.

The eFCContextDataADU is made of one data structure of either the gnssContext data type or the dsrcContext data type.

Table 10 indicates the fields in the gnssContext data structure which are defined in Annex A.

Table 10 — gnssContext fields

Field name	Data type	m/o
contextInterrelations	RoamingRules	o
regimeContextData	SEQUENCE OF SEQUENCE { contextId EntityId, iso175753ADU Iso17575-3AduBody, gnssGDFLayout SectionTollingLayoutGDF, feeModifiers FeeModifiers }	 m m o o

The contextInterrelations field is an optional field which can be used to define the interrelations with other toll regimes. These rules and configuration parameters are defined in ISO/TS 17575-4.

The regimeContextData field is used to provide details of the GNSS context which will be used by the Toll Service Provider to determine the behaviour of the Front End.

Table 11 indicates the fields in the dsrcContext data structure which are defined in Annex A.

Table 11 — dsrcContext fields

Field name	Data type	m/o

regimeContextData	SEQUENCE {
	iso175753ADU Iso17575-3AduBody,
	feeModifiers FeeModifiers OPTIONAL
	}
		
		m
		o

The regimeContextData field is used to provide details of the DSRC context which will be used by the Toll Service Provider to determine the behaviour of the OBE.

6.8 ExceptionListADU data structure

The Exception List message can be used by the Toll Service Provider to provide either as a response to a request for a specific version of an Exception List by a Toll Charger or as a push message if an updated Exception List is available.

The exceptionListADU is made up of a single data structure of the ExceptionListADU data type.

Table 12 indicates the fields in the ExceptionListADU data structure which are defined in Annex A.

Table 12 — ExceptionList fields

Field name	Data type	m/o
exceptionListversion	ExceptionListVersion	m
exceptionListType	ExceptionListType	m
exceptionValidityStart	GeneralizedTime	o
exceptionValidityEnd	GeneralizedTime	o
exceptionListEntries	SEQUENCE OF ExceptionListEntry	m

The exceptionListVersion field shall contain a version number which will be incremented with each version of the exception list.

The exceptionListType field is an enumerated field which is used to indicate the type of exception list (black list, white list, grey list …). This International Standard does not define the semantics of a white, black or grey list. The values of the exceptionListType are just options to differentiate lists. It is up to bilateral agreements or further specifications to define the exact semantics for each list type, e.g. for EETS.

The exceptionValidityStart is an optional field which, if populated, can be used to specify a point in time in the future from which the exception list is valid. If this is not completed then it shall be assumed that it is valid with immediate effect.

The exceptionValidityEnd field is an optional field which, if populated, can be used to specify and expiry date and time for the exception list.

The exceptionListEntries field is structured field which contains the complete list of entries for the Toll Service Provider's exception list and includes for each entry userId, blockType, reasonCode and dateAndTime.

6.9 ReportAbnormalOBEADU data structure

The Report Abnormal OBE message can be used by the Toll Charger either in a response to a request from the Toll Service Provider or as a push message when abnormal OBE behaviour is detected.

The reportAbnormalOBEADU is made up of one data structure of the ReportAbnormalOBEADU data type.

Table 13 indicates the fields of the ReportAbnormalOBE data structure which are defined in Annex A.

Table 13 — ReportAbnormalOBEADU fields

Field name	Data type	m/o
userID	UserId	m
dateandTime	GeneralizedTime	m
abnormalOBEReasonCode	SEQUENCE OF AbnormalOBEReasonCode	m

The userID field shall contain the UserID recorded by the Toll Charger's equipment.

The dateandTime field shall contain the date and time of the detection of the abnormal OBE behaviour.

The abnormalOBEReasonCode shall contain the coding of the reasons for the Abnormal OBE report determined by the Toll Charger.

6.10 RetrieveTollDeclarationADU data structure

The Retrieve Specific Toll Declarations message can be used to query a Toll Service Provider for Toll declarations for a specific Service User in a specified level of detail defined in the Charge report configuration of the EFC Context data.

The retrieveTollDeclarationADU is made up of one data structure of the RetrieveTollDeclarationADU Data Type.

Table 14 indicates the fields of the RetrieveTollDeclarationADU data structure which are defined in Annex A.

Table 14 — RetrieveTollDeclarationADU fields

Field name	Data type	m/o
userId	UserId	o
startTime	GeneralizedTime	o
endTime	GeneralizedTime	o

The userId field is an optional field used to specify the UserId for which the Toll Declarations are requested. If it is not specified then the response shall include all Toll Declarations for all UserIds linked to the Service Provider.

The startTime field is an optional field which can be used to specify a start of a period for which Toll Declarations are requested. If it is not specified then the response shall include all historic Toll Declarations.

The endTime field is an optional field which can be used to specify the end of a period for which Toll Declarations are requested. If it is not specified then the response shall include all Toll Declarations up to the time the request message was generated.

6.11 Toll DeclarationADU data structure

The Toll Declaration message can be used by both the Toll Service Provider and the Toll Charger either in response to a general Request to receive unsent Toll Declarations, a Retrieve specific Toll Transactions message to retrieve Toll declarations for a specific vehicle or as a push message when Toll Declarations are available to be transferred.

The tollDeclarationADU is made up of one data structure of the TollDeclarationADU Data Type.

Table 15 indicates the fields of the Toll Declaration data structure which are defined in Annex A.

Table 15 — TollDeclarationADU fields

Field name	Data type	m/o
tollDeclarationId	TollDeclarationId	m
gnssTollDeclaration	SEQUENCE OF ChargeReport	m

The tollDeclarationId field contains the identifier for the Toll Declaration which is unique within the generating entity.

The tollDeclaration field is a sequence of ChargeReport which is imported from ISO/TS 17575-1 according to the format specified in the EFC Context Data which details the data required for the Toll Declaration.

6.12 BillingDetailsADU data structure

The Billing Details message can be used by both the Toll Service Provider and the Toll Charger either in response to a request to receive unsent Toll Declarations or as a push message when Billing Details are available.

The billingDetailsADU is made up of one data structure of the BillingDetailsADU Data Type.

Table 16 indicates the fields of the BillingDetailsADU data structure which are defined in Annex A.

Table 16 — BillingDetailsADU fields

Field name	Data type	m/o

billingDetailsId	SEQUENCE {
	issuerId Provider,
	claimId INTEGER
	}
		m

tollChargerId	Provider	m
contextId	RegimeId	m
userId	UserId	o
period	Period	m
billingDetailAmount	Amount	m

usageDetails	SEQUENCE {
	contextName UTF8String,
	appliedUserClass UTF8String,
	perDeclaredVehicleClasses SEQUENCE OF SEQUENCE{
	declaredVehicleClass UTF8String,
	perUsedTimeClasses SEQUENCE OF SEQUENCE{
	appliedTimeClass UTF8String,
	costCenter UTF8String OPTIONAL,
	usageList SEQUENCE OF UsageDetail
	}
	}
	}
		o

refTollDeclaration	SEQUENCE OF TollDeclarationId	o
associatedEventData	SEQUENCE OF AssociatedEventData	o

The billingDetailsId field shall contain the unique (within the Toll Charge) identifier for the Billing Detail.

The tollChargerId field shall contain the unique identifier for the Toll Charger.

The contexId field shall contain the Toll charge specific Regime identifier.

The optional userId field may contain the userId of the Service User the Billing Detail shall be charged to. If the userId is not stated, the Toll Service Provider may get this information out of the referenced Toll declaration.

The billingDetailAmount field shall contain the details of the fee, currency and VAT rate.

The usageDetails field can be used to provide information for the invoice to the Service User.

The refTollDeclaration field can be used to provide a linkage to Toll Declarations which are associated with the Billing Details.

The associatedEventData field can be used to attach the raw event data from which the billing detail has been generated and may include cccRecord, imageRecord, aNPRRecord, classificationRecord, operatorRecord, dsrcData.

Table 17 indicates the fields of the cccRecord data structure which are defined in Annex A.

Table 17 — cccRecord fields

Field name	Data type	m/o
timeOfEvent	GeneralizedTime	o
locationOfEvent	Location	o
cccMessages	SEQUENCE OF DSRCData	m
initiatedActions	SEQUENCE OF InitiatedAction	m

The timeOfEvent Field shall contain the Date and Time of the CCCRecord.

The locationOfEvent field shall contain the Toll Charger coding of the location of where the CCCRecord was generated.

The cccMessages field shall contain the raw DSRC data for the CCCRecord.

The initiatedActions field shall contain the details of the actions that have been initiated by the Toll Charger as a result of the CCCRecord.

Table 18 indicates the fields of the imageRecord data structure which are defined in Annex A.

Table 18 — imageRecord fields

Field name	Data type	m/o
imageRecordId	RecordId	m
imageDateTime	GeneralizedTime	m

imageLocation	SEQUENCE {
	cameraOwner Provider,
	cameraId INTEGER(0..65535)
	}
		m

imageReference	INTEGER	o
imageData	BIT STRING	m
subRecordAuthenticator	SubRecordAuthenticator	o

The imageRecordID field shall contain the unique Toll Charger identifier for the ImageRecord.

The imageDateTime field shall contain the date and time that the image record was generated.

The imageLocation field shall contain the Toll Charger coding of the location of where the ImageRecord was generated.

The imageReference field shall contain the Toll Charger unique identifier for the Image.

The imageData field shall contain the JPEG image Base 64 binary encoded.

The subRecordAuthenticator field shall contain an authenticator calculated over the content of the imageRecord.

Table 19 indicates the fields of the aNPRRecord data structure which are defined in Annex A.

Table 19 — aNPRRecord fields

Field name	Data type	m/o
anprRecordId	RecordId	m

associatedimages	SEQUENCE OF SEQUENCE {
	anprImage RecordId,
	contextImage RecordId
	}
		m

imageDateTime	GeneralizedTime	m

imageLocation	SEQUENCE {
	cameraOwner Provider,
	cameraId INTEGER(0..65535)
	}
		m

determinedVRM	SEQUENCE { anprResult OCTET STRING (SIZE(14)), anprConfidence INTEGER(0..100), secondaryAnprResult OCTET STRING(SIZE(14)) OPTIONAL, manualResult OCTET STRING (SIZE(14)) OPTIONAL, operatorId INTEGER(0..65535) },	m
vehicleDetails	SEQUENCE { vehicleMake UTF8String OPTIONAL, vehicleModel UTF8String OPTIONAL, vehicleColour UTF8String OPTIONAL },	m
subRecordAuthenticator	SubRecordAuthenticator	o

The anprRecordID field shall contain the unique Toll Charger identifier for the anprRecord.

The associatedImages field shall contain the references to the associated images from which the anprRecord was generated.

The imageDateTime field shall contain the date and time that the image record was generated.

The imageLocation field shall contain the Toll Charger coding of the location of where the Image Record was generated.

The determinedVRM field shall contain the ANPR result(s).

The vehicleDetails field shall contain the determined vehicle details if available.

The subRecordAuthenticator field shall contain an authenticator calculated over the content of the aNPRRecord.

Table 20 indicates the fields of the classificationRecord data structure which are defined in Annex A.

Table 20 — classificationRecord fields

Field name	Data type	m/o
classificationRecordID	RecordId	m
derivedLocalClass	INTEGER (0..255)	m

measuredParameters	SEQUENCE {
	measurementTime GeneralizedTime,
	measuredClass INTEGER(0..255) OPTIONAL,
	measuredVehicleLength INTEGER(0..65535) OPTIONAL,
	measuredVehicleWidth INTEGER(0..65535) OPTIONAL,
	measuredVehicleHeight INTEGER(0..65535) OPTIONAL,
	measuredHeightAbove1stAxle INTEGER(0..65535) OPTIONAL,
	measuredVehicleAxles INTEGER(0..15) OPTIONAL,
	measuredTrailerAxles INTEGER(0..7) OPTIONAL,
	measuredTrailerPresence BOOLEAN OPTIONAL,
	...
	}
		m

subRecordAuthenticator	SubRecordAuthenticator	o

The classificationRecordID field shall contain the unique Toll Charger identifier for the classificationRecord.

The derivedLocalClass field shall contain the Toll Charger specific code for the derived vehicle class.

The measureParameters field shall contain the raw measurements of the vehicle parameters recorded by the classification system.

The subRecordAuthenticator field shall contain an authenticator calculated over the content of the classificationRecord.

Table 21 indicates the fields of the operatorRecord data structure which are defined in Annex A.

Table 21 — operatorRecord fields

Field name	Data type	m/o
operatorRecordId	RecordId	m

operatorData	SEQUENCE {
	operatorTime GeneralizedTime,
	operatorClass INTEGER(0..255) OPTIONAL,
	operatorId INTEGER(0..65535) OPTIONAL,
	operatorVRM OCTET STRING (SIZE(14)) OPTIONAL,
	operatorPAN UTF8String OPTIONAL,
	operatorOBUId UTF8String OPTIONAL,
	machineReadPAN UTF8String OPTIONAL,
	machineReadOBUId UTF8String OPTIONAL,
	...
	}
		m

subRecordAuthenticator	SubRecordAuthenticator	o

The operatorRecordID field shall contain the unique Toll Charger identifier for the operatorRecord.

The operatorData field shall contain the information collected and recorded by the Operator.

The subRecordAuthenticator field shall contain an authenticator calculated over the content of the operatorRecord.

Table 22 indicates the fields of the dsrcData data structure which are defined in Annex A.

Table 22 — dsrcData fields

Field name	Data type	m/o
dsrcRSEData	DSRCRSEData	o
dsrcAttributesRead	SEQUENCE OF Attributes	o
dsrcAttributesWritten	SEQUENCE OF Attributes	o
dsrcAttrAuth	DSRCAttrAuth	o

The dsrcRSEData field shall contain the data recorded by the RSE during the DSRC Transaction as specified in Annex A.

The dsrcAttributesRead field shall contain the details of the attributes read from the OBE during the DSRC transaction.

The dsrcAttributesWritten field shall contain the details of the attributes written to the OBE during the DSRC transaction.

The dsrcAttrAuth field shall contain the information that is required to verify the authenticators received during the DSRC Transaction as specified in Annex A.

6.13 PaymentClaimADU data structure

The Payment Claim Message can be used by the Toll Charger either in response to the Request Message sent by the Toll Service Provider or as a push message when new Payment Claims are available.

The paymentClaimADU is made up of one data structure of the PaymentClaimADU data type.

Table 23 indicates the fields of the PaymentClaimADU data structure which are defined in Annex A.

Table 23 — PaymentClaimADU fields

Field name	Data type	m/o
paymentClaimId	INTEGER	m
startDateTime	GeneralizedTime	m
endDateTime	GeneralizedTime	o
userId	UserId	o
paymentClaimAmount	Amount	m
paymentClaimStatus	PaymentClaimStatus	m
typeOfGoods	TypeOfGoods	o
associatedBillingDetails	SEQUENCE OF RecordId	o

The paymentClaimId field shall contain the unique (within the originating entity) identifier for the Payment Claim.

The startDateTime field specifies the start of the period for which the Payment Claim relates.

The endDateTime field can be used to specify the end of the period for which the Payment Claim relates. If this is not populated then it shall be assumed to be the time that the message was generated.

The userId field can be used to provide User specific Payment Claims.

The paymentClaimAmount field contains the details of the net amount, currency and VAT for the payment claim.

The paymentClaimStatus field is used to indicate the version and status of the Payment Claim.

The typeOfGoods field is an optional enumerated field which can be used to indicate the type of goods for the payment claim including: toll, discount, credit note, penalty, processing fee.

The associatedBillingDetails field is used to associate previously exchanged Billing Details to the Payment Claim.

6.14 RetrieveUserDetailsADU data structure

The Retrieve User Details Message can be used by the Toll Charger to send a request to a Toll Service Provider to declare if he has a contractual relationship with one or more Service Users and/or to provide details for the referenced Service Users.

The retrieveUserDetailsADU is made up of one data structure of the RetrieveUserDetailsADU data type.

Table 24 indicates the fields of the RetrieveUserDetailsADU data structure which are defined in Annex A.

Table 24 — RetrieveUserDetailsADU fields

Field name	Data type	m/o
userId	UserId	m
listOfParametersRequested	SEQUENCE OF UserParameterRequest	o
userDetailsRequestReason	UserDetailsRequestReason	o

The userId field is used to specify the UserId for which the additional details are requested.

The listOfParametersRequested field is an optional field used to specify which additional parameters, if any, are required. If this is not populated then the response shall contain all available details.

The userDetailsRequestReason field is an optional field used to specify the reason for the request.

6.15 ProvideUserDetailsADU data structure

The Provide User Details Message is used by a Toll Service Provider in response to a Retrieve User Details Message from a Toll Charger.

The provideUserDetailsADU is made up of one data structure of the ProvideUserDetailsADU data type.

Table 25 indicates the fields of the ProvideUserDetailsADU data structure which are defined in Annex A.

Table 25 — ProvideUserDetailsADU fields

Field name	Data type	m/o
originaluserIdRequest	UserId	m
userId	UserId	m
statusFlag	UserStatus	o
listOfUserParameters	SEQUENCE OF UserParameterResponse	o

The originaluserIdRequest field contains the UserId specified in the associated request message.

The userId field contains the UserId contained in the corresponding response message.

The statusFlag field is an optional enumerated field which can be used to indicate details of the current status associated with the UserId.

The listOfUserParameters field is an optional field which can be used to provide the response to the requested parameters.

Table 26 indicates the fields of the UserParameterResponse data structure which are defined in Annex A.

Table 26 — UserParameterResponse fields

Field name	Data type	m/o
userParameterResponse	UserParameterResponseDetails	m
userParameterStatus	UserParameterStatus	m

The userParameterResponse field contains the response to the requested information and the userParameterStatus is an optional field which is used to provide an explanation of why the requested User Parameter has not be provided.

6.16 RetrieveCCCEventADU data structure

The Retrieve Specific CCC Event message can be used to query a Toll Charger for information on CCC Events for a specific Service User.

The retrieveCCCEventADU is made up of one data structure of the RetrieveCCCEventADU Data Type.

Table 27 indicates the fields of the RetrieveCCCEventADU data structure which are defined in Annex A.

Table 27 — RetrieveCCCEventADU fields

Field name	Data type	m/o
userId	UserId	o
startTime	GeneralizedTime	o
endTime	GeneralizedTime	o

The userId field is an optional field used to specify the UserId for which the CCC Events are requested. If it is not specified then the response shall include all historic CCC Events for all UserIds associated with the requesting Toll Service Provider.

The startTime field is an optional field which can be used to specify a start of a period for which CCC Events are requested. If it is not specified then the response shall include all historic CCC Events for the specified UserId or for all UserIds associated with the requesting Toll Service Provider.

The endTime field is an optional field which can be used to specify the end of a period for which CCC Events are requested. If it is not specified then the response shall include all CCC Events for the specified UserId or for all UserIds associated with the requesting Toll Service Provider up to the time the request message was generated.

6.17 ReportCCCEventADU data structure

The Report CCC Event Message can be used by the Toll Charger either in response to the Request Message sent by the Toll Service Provider or as a push message when new CCC Events are available.

The reportCCCEventADU is made up of a sequence of the data structure CCCEvent data type.

Table 28 indicates the fields of the CCCEvent data structure which are defined in Annex A.

Table 28 — CCCEvent fields

Field name	Data type	m/o
timeOfEvent	GeneralizedTime	o
locationOfEvent	Location	o
cccMessages	SEQUENCE OF DSRCData	m
initiatedActions	SEQUENCE OF InitiatiedAction	m

The timeOfEvent field can be used to specify the time of the CCC Event as determined by the Toll Charge Equipment.

The locationOfEvent field can be used to provide details of the location of the CCC Event.

The cccMessages field contains the record(s) of the CCC Event(s) from the Toll Charger equipment.

The initiatedActions field provides details of the actions that have been instigated by the Toll Charger as a result of the specified CCC Event.

6.18 Report QA data structure

The Report QA Message can be used by the Toll Charger either in response to the Request Message sent by the Toll Service Provider or as a push message when new QA Reports are available.

The reportQAADU is made up of one data structure of the ReportQAADU data type.

Table 29 indicates the fields of the ReportQAADU data structure which are defined in Annex A.

Table 29 — ReportQAADU fields

Field name	Data type	m/o
qualityParameterID	INTEGER	m
qualityParameterName	UTF8String	o
qualityParameterValue	REAL	o
qualityParameterStatus	QualityParameterStatus	m

The qualityParameterID field is used to specify the type of Quality Parameter from a bi-laterally agreed list between the toll service provider and the toll charger.

The qualityParameterName field is an optional field which can be used to provide a text description for the Quality Parameter.

The qualityParameterValue field is an optional field which can be used to provide a value for the Quality Parameter.

The qualityParameterStatus field is used to specify the status of the Quality Parameter.

7 Transfer mechanisms and supporting functions

7.1 Transfer mechanisms

The messages specified in Clause 6 of this International Standard shall be exchanged between the central equipments of Toll Service Providers and Toll Chargers making use of a variety of applicable communication means. These communication means shall be used applying a common set of communication services for all applicable communication stacks and media.

If the Application protocol used for the information transfer is character-oriented, then the XML Encoding Rules (XER) according to ISO/IEC 8825-4 shall be used to encode data.

7.2 Supporting functions

7.2.1 Communication Services

This International Standard provides the means of a request, an acknowledge and a status message. This International Standard does not prescribe any sequence of messages or timing. Within the limits of the provisions of Clause 6 the communicating parties are free to select their communications means.

7.2.2 Message Authenticators

Authentication of messages is done by using the infoexchange authenticator attribute in the APDU. It shall be computed over the attributes of the header and the payload of the message itself.

The concrete definition of the Message Authenticators is outside the scope of this International Standard and shall be part of an EFC security framework standard.

Annex A
(normative)

Data type specifications

The data types and associated coding related to the data elements described in Clause 6 are defined using the Abstract Syntax Notation One (ASN.1) technique according to ISO/IEC 8824-1.

```
EFCInfoExchange    {iso standard 12855 modules(0) data(1) version(1)}
DEFINITIONS AUTOMATIC TAGS
::= BEGIN
EXPORTS ALL;

IMPORTS
-- Imports all relevent existing data definitions from relevent standards
-- ISO 14906 (EFC modules)

ResultOp, StationType, Attributes, GetInstanceRq, GetInstanceRs,
ContractSerialNumber, PaymentMeans, VehicleClass, VehicleDimensions,
VehicleAxles, VehicleWeightLimits, VehicleSpecificCharacteristics,
DescriptiveCharacteristics, EngineCharacteristics, EnvironmentalCharacteristics,
FutureCharacteristics, EquipmentStatus, ReceiptData, EFC-ContextMark,
LPN, Provider, PaymentFee, DateAndTime, DateCompact, PaymentMeans,
PersonalAccountNumber,GetStampedRq, GetStampedRs
FROM EfcModule {iso standard 14906 modules(0) efc(0) version(1)}

VehicleWeightLaden,VehicleIdentificationNumber,VehicleLicencePlateNumber,
VehicleAuthenticator,ValidityOfContract,ReceiptContract,PurseBalance,PayUnit,
PaymentMeansUnit,PaymentMeansBalance,DriverCharacteristics,ContractValidity
FROM EfcModule {iso standard 14906 modules(0) efc(0) version(1)}

EquipmentOBUId, ApplicationContextMark, ObeConfiguration, Time, VST, SessionLocation
FROM DSRCData  {iso standard 14906 modules(0) dsrc(1) version(1)}

ObeId, Distance, ChargeReport
FROM ChargingModule {iso standard 17575 modules(0) efc(0) version(1)}

Iso17575-3AduBody, TariffClassId, EntityId, ChargeObjectId, CordonEntryLocation,
CordonExitLocation, ChargeUnit, Point, TimeClassId, VersionAndValidity,
TollContextOverview,
TariffTable, TariffClassDefinition, LocalVehicleClassDefinition, TimeClassDefinition,
UserClassDefinition, AbsolutePointCoordinates
FROM ContextDataModule {iso standard 175753 modules (0) efc (0) version (1)}

-- ISO 14816

CountryCode, IssuerIdentifier
FROM AVIAEINumberingAndDataStructures {iso(1) standard(0) iso14816(14816) }

-- ISO 17575-4

RoamingRules
FROM RoamingModule {iso standard 175754 modules(0) efc(0) version(1)};


InfoExchange ::= SEQUENCE {
    infoExchangeContent       InfoExchangeContent,
    infoExchangeAuthenticator MessageAuthenticator OPTIONAL
        -- over the single attribute InfoExchangeContent
    }
```

```
InfoExchangeContent ::= SEQUENCE {
    apci    ApciFields,    -- Header
    adu     SEQUENCE OF AduContent
        -- Sequence of individual ADUs of the same type
    }

ApciFields ::= SEQUENCE {
    messageOriginator     Provider,  -- Source of the content of the message
    informationSenderID   Provider,  -- Sender of this content
    informationrecipientIDProvider,
        -- if 'NullValue' allows Broadcast to all connected entities
    contextID             RegimeID OPTIONAL,
    messageIdentifier     MessageIdentifier,
        -- Shall be unique together with messageOriginator
    relatedMessageId      RelatedMessageId OPTIONAL,
    aduType               AduType,
        -- Type of ADU, shall be unique for the message
    numberofStructs       NumberOfADUStruct,
    messageDate           GeneralizedTime
    }

AduContent ::= CHOICE {
    requestADU                [1] RequestADU,
    ackADU                    [2] AckADU,
    trustObjectADU            [3] TrustObjectADU,
    efcContextDataADU         [4] EFCContextDataADU,
    exceptionListADU          [5] ExceptionListADU,
    reportAbnormalOBEADU      [6] ReportAbnormalOBEADU,
    tollDeclarationADU        [7] TollDeclarationADU,
    billingDetailsADU         [8] BillingDetailsADU,
    paymentClaimADU           [9] PaymentClaimADU,
    reportQAADU               [10]    ReportQAADU,
    statusADU                 [11]    StatusADU,
    retrieveUserDetailsADU    [12]    RetrieveUserDetailsADU,
    provideUserDetailsADU     [13]    ProvideUserDetailsADU,
    reportCCCEventADU         [14]    ReportCCCEventADU,
    retrieveTollDeclarationADU    [15]    RetrieveTollDeclarationADU,
    retrieveCCCEventADU       [16]    RetrieveCCCEventADU
    }

-- Level 2 definitions

RelatedMessageId ::= SEQUENCE {
    messageOriginator  Provider,
    messageIdentifier  MessageIdentifier
    }

RequestADU::= SEQUENCE {
    requestedADUType   AduType OPTIONAL,
    aDUIdentifier      MessageIdentifier OPTIONAL,
    numberOfADUStructs SEQUENCE OF NumberOfADUStruct OPTIONAL
    }

AckADU ::= SEQUENCE {
    aDUIdentifier          MessageIdentifier,
    numberOfADUStructs     SEQUENCE OF NumberOfADUStruct OPTIONAL,
    ackCode                MessageReasonCode,
    aduCode SEQUENCE OF SEQUENCE {
        numberOfADUStruct  NumberOfADUStruct,
            -- set to 0 if applies to all structs
        aDUReasonCode      ADUReasonCode
        } OPTIONAL
    }

StatusADU ::= SEQUENCE {
    generalStatusCode  GeneralStatusCode,
    messageStatusCode  SEQUENCE {
        message            MessageIdentifier,
```

```
        reasonOfADUStruct  SEQUENCE OF SEQUENCE {
            numberOfADUStruct  NumberOfADUStruct,
            reason             MessageReasonCode
            }
        } OPTIONAL
    }

TrustObjectADU ::= SEQUENCE {
    trustObjectID          INTEGER,
    startValidity          GeneralizedTime OPTIONAL,
    endValidity            GeneralizedTime OPTIONAL,
    trustObjectStatus      TrustObjectStatus,
    typeOfTrustObject      ListOfTrustObjectTypes,
    purposesOfTrustObject  SEQUENCE OF ListOfTrustObjectPurposes,
    eFCContextMark         EFC-ContextMark OPTIONAL,
    keyReference           INTEGER OPTIONAL,
        -- can be used e.g. for the keyref of EN15509
    trustObject            TrustObjectCode,
    ...
    }

ReportQAADU ::= SEQUENCE {
    qualityParameterID        INTEGER,
        -- bilaterally agreed between TSP and TC
    qualityParameterName      UTF8String OPTIONAL,
        -- printable name of the parameter
    qualityParameterValue     REAL OPTIONAL,
    qualityParameterStatus    QualityParameterStatus
    }

ExceptionListADU ::= SEQUENCE {
    exceptionListversion      ExceptionListVersion,
    exceptionListType         ExceptionListType,
    exceptionValidityStart    GeneralizedTime OPTIONAL,
    exceptionValidityEnd      GeneralizedTime OPTIONAL,
    exceptionListEntries      SEQUENCE OF ExceptionListEntry
    }

ReportAbnormalOBEADU ::= SEQUENCE {
    userId                 UserId,
    dateAndTime            GeneralizedTime,
    abnormalOBEReasonCode  SEQUENCE OF AbnormalOBEReasonCode
    }

PaymentClaimADU ::= SEQUENCE {
    paymentClaimId             INTEGER,
    startDateTime              GeneralizedTime,
    endDateTime                GeneralizedTime OPTIONAL,
    userId                     UserId OPTIONAL,
    paymentClaimAmount         Amount,
    paymentClaimStatus         PaymentClaimStatus,
    typeOfGoods                TypeOfGoods OPTIONAL,
    associatedBillingDetails   SEQUENCE OF RelatedMessageId OPTIONAL
        -- billingDetailsId
    }

EFCContextDataADU::= CHOICE {
    gnssContext[1] SEQUENCE {
        contextInterrelations RoamingRules OPTIONAL,
        regimeContextData     SEQUENCE OF SEQUENCE {
            iso175753ADU      Iso17575-3AduBody,
            gnssGDFLayout     SectionTollingLayoutGDF OPTIONAL,
                -- overrides ContextLayout in 17575 if present
            feeModifiers      FeeModifiers OPTIONAL
            }
        },
```

```
    dsrcContext[2] SEQUENCE {
        regimeContextData SEQUENCE {
            iso175753ADU       Iso17575-3AduBody,
            feeModifiers       FeeModifiers OPTIONAL
            }
        }
    }

RetrieveTollDeclarationADU ::= SEQUENCE OF RetrieveTollDeclarations

RetrieveUserDetailsADU ::= SEQUENCE OF RetrieveUserDetails

RetrieveCCCEventADU ::= SEQUENCE OF RetrieveCCCEvents

ProvideUserDetailsADU ::= SEQUENCE OF ProvideUserDetails

ReportCCCEventADU ::= SEQUENCE OF CCCEvent

--------------------

RetrieveTollDeclarations ::= SEQUENCE {
    userID     UserId OPTIONAL,
    startTime  GeneralizedTime OPTIONAL,
    endTime    GeneralizedTime OPTIONAL
    }

TollDeclarationADU ::= SEQUENCE {
    tollDeclarationId     TollDeclarationId,
    gnssTollDeclaration   SEQUENCE OF ChargeReport
    }

BillingDetailsADU ::= SEQUENCE {
    billingDetailsId SEQUENCE {
        issuerId          Provider,
        claimId           INTEGER
        },
    tollChargerId         Provider,
    contextId             RegimeID,
    userId                UserId OPTIONAL,
    period                Period,
    billingDetailsAmount  Amount,
    usageDetails SEQUENCE {
        contextName               UTF8String,
        appliedUserClass          UTF8String,
        perDeclaredVehicleClasses     SEQUENCE OF SEQUENCE {
            declaredVehicleClass  UTF8String,
            perUsedTimeClasses    SEQUENCE OF SEQUENCE {
                appliedTimeClass  UTF8String,
                costCenter        UTF8String OPTIONAL,
                usageList         SEQUENCE OF UsageDetail
                }
            }
        } OPTIONAL,
    refTollDeclaration    SEQUENCE OF TollDeclarationId OPTIONAL,
    associatedEventData   SEQUENCE OF AssociatedEventData OPTIONAL
    }

AssociatedEventData ::= SEQUENCE {
    cCCRecord             CCCEvent OPTIONAL,
    imageRecord           ImageRecord OPTIONAL,
    aNPRRecord            ANPRRecord OPTIONAL,
    classificationRecord  ClassificationRecord OPTIONAL,
    operatorRecord        OperatorRecord OPTIONAL,
    dsrcData              DSRCData OPTIONAL
    }
```

```
UsageDetail ::=     CHOICE {
    forSectionedRoads      [1] SEQUENCE {
            qualifier CHOICE {
                howOften           [1] INTEGER,
                entranceTime       [2] GeneralizedTime
                } OPTIONAL,
            entranceChargeObject          ChargeObjectId, -- from ISO 17575-3
            entranceChargeObjectName      UTF8String OPTIONAL,
            intermediateSections          SEQUENCE OF SEQUENCE {
                intermediateSection       ChargeObjectId, -- from ISO 17575-3
                intermediateSectionName   UTF8String OPTIONAL,
                intermediateSectionTime   GeneralizedTime OPTIONAL
                } OPTIONAL,
            exitQualifier CHOICE {
                exitChargeObjectStruct    [1] SEQUENCE {
                    exitChargeObject      ChargeObjectId, -- from ISO 17575-3
                    exitChargeObjectName  UTF8String OPTIONAL,
                    exitTime              GeneralizedTime OPTIONAL
                    },
                singleChargeObject        [2] BOOLEAN
                },
            fee                    Amount OPTIONAL,
            feeQualifier           FeeQualifier OPTIONAL
        },
    forTravellingInArea    [2] SEQUENCE {
        areaDisplayName               UTF8String,
        accumulatedDistance           INTEGER, -- Length in Meter,
        beginOfAccumulation           GeneralizedTime OPTIONAL,
        endOfAccumulation          GeneralizedTime OPTIONAL,
        fee                        Amount OPTIONAL,
        feeQualifier               FeeQualifier OPTIONAL
        },
    forStayedInArea        [3] SEQUENCE {
        areaDisplayName               UTF8String,
        qualifier CHOICE {
            howOften               [1] INTEGER,
            entranceTime           [2] GeneralizedTime
            } OPTIONAL,
        stayedDuration             INTEGER,   -- in number of charge units
        chargeUnit                 ChargeUnit,-- from ISO 17575-3
        fee                        Amount OPTIONAL,
        feeQualifier               FeeQualifier OPTIONAL
        },
    forCordonCrossings     [4] SEQUENCE {
        qualifier CHOICE {
            howOften               [1] INTEGER,
            entranceTime           [2] GeneralizedTime
            } OPTIONAL,
        entranceChargeObject       CordonEntryLocation,  -- from ISO 17575-3
        entranceChargeObjectName   UTF8String OPTIONAL,
        exitChargeObject           CordonExitLocation OPTIONAL,
        exitChargeObjectName       UTF8String OPTIONAL,
        exitTime                   GeneralizedTime OPTIONAL,
        fee                        Amount OPTIONAL,
        feeQualifier               FeeQualifier OPTIONAL
        },
    freeTextDetail         [5] SEQUENCE {
        textLanguage               LanguageID,
        textDetail                 UTF8String,
        fee                        Amount OPTIONAL,
        feeQualifier               FeeQualifier OPTIONAL
        }
    }

FeeQualifier ::= ENUMERATED {
    standardCharge     (0),
    tollSubstitute     (1),
    belatedPayment     (2),
```

```
    reimbursement      (3),
    refForFutureUse1   (4),
    refForFutureUse2   (5),
    refForFutureUse3   (6),
    refForFutureUse4   (7),
    refForFutureUse5   (8),
    refForPrivateUse1  (100),
    refForPrivateUse2  (101),
    refForPrivateUse3  (102),
    refForPrivateUse4  (103),
    refForPrivateUse5  (104)
    }

LanguageID ::= UTF8String -- two letter language code according to ISO 639-1

-- Level 3 definitions

-- General ADU definitions

AduType ::= ENUMERATED {
    requestADU                (1),
    ackADU                    (2),
    trustObjectADU            (3),
    efcContextDataADU         (4),
    exceptionListADU          (5),
    reportAbnormalOBEADU      (6),
    tollDeclarationADU        (7),
    billingDetailsADU         (8),
    paymentClaimADU           (9),
    reportQAADU               (10),
    statusADU                 (11),
    retrieveUserDetailsADU    (12),
    provideUserDetailsADU     (13),
    reportCCCEventADU         (14),
    retrieveTollDeclarationADU    (15),
    retrieveCCCEvents         (16)
    }

GeneralStatusCode ::= ENUMERATED {
    notReadyToReceive  (1),
    readyToReceive     (2),
    refForFutureUse1   (3),
    refForFutureUse2   (4),
    refForFutureUse3   (5),
    refForFutureUse4   (6),
    refForFutureUse5   (7),
    refForPrivateUse1  (100),
    refForPrivateUse2  (101),
    refForPrivateUse3  (102),
    refForPrivateUse4  (103),
    refForPrivateUse5  (104)
    }

MessageReasonCode ::= ENUMERATED {
    -- provides status information for the whole message and is ADU type independent
    obsoleteADU        (1),    -- ADU is no longer valid or to be used
    aDUOK              (2),    -- Message was accepted
    aDUNotOK           (3),    -- Message rejected
    refForFutureUse1   (4),
    refForFutureUse2   (5),
    refForFutureUse3   (6),
    refForFutureUse4   (7),
    refForFutureUse5   (8),
    refForPrivateUse1  (100),
    refForPrivateUse2  (101),
    refForPrivateUse3  (102),
    refForPrivateUse4  (103),
    refForPrivateUse5  (104)
    }
```

```
ADUReasonCode ::= INTEGER {
    -- provides the option for including an ADU type dependent status code
    trustObjectExpired        (1),
    trustObjectUnreadable     (2),
    claimRejectedByTSP        (3),
    claimApprovedByTSP        (4),
    reportQAAccepted          (5),
    reportQANotAccepted       (6),
    paymentGuaranteeAccepted  (7)
    -- 8-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

--Toll declaration and billing details definitions

Amount ::= SEQUENCE {
    netPaymentFee  PaymentFee,-- from ISO 14906
    vATrate        INTEGER,   -- in 0.01%
    }

Period ::= SEQUENCE {
    startDate  GeneralizedTime,
    endDate    GeneralizedTime
    }

ImageRecord ::= SEQUENCE {
    imageRecordId             RecordId,
    imageDateTime             GeneralizedTime,
    imageLocation SEQUENCE {
        cameraOwner    Provider,
        cameraId       INTEGER(0..65535)
        },
    imageReference            INTEGER OPTIONAL,
    imageData                 BIT STRING,
    subRecordAuthenticator    SubRecordAuthenticator OPTIONAL
    }

ANPRRecord ::= SEQUENCE   {
    anprRecordId   RecordId,
    associatedimages SEQUENCE OF SEQUENCE {
        anprImage          RecordId,
        contextImage       RecordId
        },
    imageDateTime  GeneralizedTime,
    imageLocation SEQUENCE{
        cameraOwner        Provider,
        cameraId           INTEGER(0..65535)
        },
    determinedVRM SEQUENCE {
        anprResult             OCTET STRING (SIZE(14)),
        anprConfidence         INTEGER(0..100),
        secondaryAnprResult    OCTET STRING (SIZE(14)) OPTIONAL,
        manualResult           OCTET STRING (SIZE(14)) OPTIONAL,
        operatorId             INTEGER(0..65535)
        },
    vehicleDetails SEQUENCE {
        vehicleMake        UTF8String OPTIONAL,
        vehicleModel       UTF8String OPTIONAL,
        vehicleColour      UTF8String OPTIONAL
        },
    subRecordAuthenticator SubRecordAuthenticator OPTIONAL
    }

ClassificationRecord ::= SEQUENCE {
    classificationRecordId RecordId,
    derivedLocalClass      INTEGER (0..255),
    measuredParameters SEQUENCE {
```

```
        measurementTime              GeneralizedTime,
        measuredClass                INTEGER(0..255) OPTIONAL,
        measuredVehicleLength        INTEGER(0..65535) OPTIONAL,
        measuredVehicleWidth         INTEGER(0..65535) OPTIONAL,
        measuredVehicleHeight        INTEGER(0..65535) OPTIONAL,
        measuredHeightAbove1stAxle       INTEGER(0..65535) OPTIONAL,
        measuredVehicleAxles         INTEGER(0..15) OPTIONAL,
        measuredTrailerAxles         INTEGER(0..7) OPTIONAL,
        measuredTrailerPresence      BOOLEAN OPTIONAL,
        ...
        },
    subRecordAuthenticator SubRecordAuthenticator OPTIONAL
    }

OperatorRecord ::= SEQUENCE {
    operatorRecordId      RecordId,
    operatorData SEQUENCE {
        operatorTime      GeneralizedTime,
        operatorClass     INTEGER(0..255) OPTIONAL,
        operatorId        INTEGER(0..65535) OPTIONAL,
        operatorVRM       OCTET STRING (SIZE(14)) OPTIONAL,
        operatorPAN       UTF8String OPTIONAL,
        operatorOBUId     UTF8String OPTIONAL,
        machineReadPAN    UTF8String OPTIONAL,
        machineReadOBUId  UTF8String OPTIONAL,
        ...
        },
    subRecordAuthenticator SubRecordAuthenticator OPTIONAL
    }

-- Enforcement Support definitions

RetrieveUserDetails ::= SEQUENCE {
    userId                   UserId,
    listOfParametersRequested SEQUENCE OF UserParameterRequest OPTIONAL,
    userDetailsRequestReason  UserDetailsRequestReason OPTIONAL
    }

UserStatus ::= INTEGER {
    -- Provides general information on the status of the current status of the user
    noContractualRelation            (0), -- no contract with TSP (user unknown)
    standardPrivateUserContract      (1), -- contract as private user
    standardCommercialUserContract   (2), -- contract as a commercial user
    tempExceptionListed              (3),
        -- put on exception list, expect to remove it soon
        -- (e.g. because of insufficient credit on the account)
    permanentExceptionListed         (4), -- put permanently on exception list
    contractClosed                   (5), -- contract with the user has ended
    noStatus                         (6)  -- no status information available
    -- 7-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

UserParameterRequest ::= INTEGER {
    userPostalAddress            (0),
    contractSerialNumber         (1),
    contractValidity             (2),
    driverCharacteristics        (3),
    eFC-ContextMark              (4),
    environmentalCharacteristics (5),
    engineCharacteristics        (6),
    equipmentOBUId               (7),
    equipmentStatus              (8),
    paymentMeans                 (9),
    paymentMeansBalance          (10),
    paymentMeansUnit             (11),
    payUnit                      (12),
    personalAccountNumber        (13),
```

```
    provider                    (14),
    receiptContract             (15),
    validityOfContract          (16),
    vehicleAuthenticator        (17),
    vehicleClass                (18),
    vehicleDimensions           (19),
    vehicleLicencePlateNumber   (20),
    vehicleIdentificationNumber (21),
    vehicleWeightLaden          (22),
    vehicleWeightLimits         (23)
    -- 24-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

UserDetailsRequestReason ::= INTEGER {
    generalInterest             (0),
    allowingCalculatingFee      (1),
    incompleteCCCreadout        (2),
    forwardingEnforcementTicket (3)
    -- 4-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

ProvideUserDetails ::=SEQUENCE {
    originaluserIdRequest UserId, -- in the format of the request
    userId                UserId,
    statusFlag            UserStatus OPTIONAL,
    listOfUserParameters  SEQUENCE OF UserParameterResponse OPTIONAL
    }

UserParameterResponse ::= SEQUENCE {
    userParameterResponse UserParameterResponseDetails,
    userParameterStatus   UserParameterStatus
    }

UserParameterResponseDetails ::= CHOICE {
    userPostalAddress           [1] UserPostalAddress,
    contractSerialNumber        [2] ContractSerialNumber,
    contractValidity            [3] ContractValidity,
    driverCharacteristics       [4] DriverCharacteristics,
    environmentalCharacteristics [5] EnvironmentalCharacteristics,
    engineCharacteristics       [6] EngineCharacteristics,
    equipmentOBUId              [7] EquipmentOBUId,
    equipmentStatus             [8] EquipmentStatus,
    paymentMeans                [9] PaymentMeans,
    paymentMeansBalance         [10]   PaymentMeansBalance,
    paymentMeansUnit            [11]   PaymentMeansUnit,
    payUnit                     [12]   PayUnit,
    personalAccountNumber       [13]   PersonalAccountNumber,
    purseBalance                [14]   PurseBalance,
    receiptContract             [15]   ReceiptContract,
    validityOfContract          [16]   ValidityOfContract,
    vehicleAuthenticator        [17]   VehicleAuthenticator,
    vehicleClass                [18]   VehicleClass,
    vehicleLicencePlateNumber   [19]   VehicleLicencePlateNumber,
    vehicleIdentificationNumber [20]   VehicleIdentificationNumber,
    vehicleWeightLaden          [21]   VehicleWeightLaden,
    vehicleWeightLimits         [22]   VehicleWeightLimits,
    ...
    }

UserPostalAddress ::= SEQUENCE {
    name       PrintableString (SIZE (1..60)),
    company    PrintableString (SIZE (1..60)) OPTIONAL,
    street     PrintableString (SIZE (1..60)) OPTIONAL,
    postcode   PrintableString (SIZE (6)),
    town       PrintableString (SIZE (1..60)) OPTIONAL,
    country    PrintableString (SIZE (1..40)),
    ...
    }
```

```
UserParameterStatus ::=   INTEGER {
    -- For every attribute of the user details response it can be indicated
    -- why a requested detail could not be provided
    userParameterAvailable       (0),
-- requested attribute is sent
    notAvailable                 (1),
-- requested attribute is not available
    localPrivacyRules            (2),
        -- requested attribute cannot be sent due to privacy regulations
    missingContractualAgreement  (3)
        -- requested attribute cannot be sent due to a missing contractual agreement
    -- 4-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

RetrieveCCCEvents ::= SEQUENCE {
    userID     UserId  OPTIONAL,
    startTime  GeneralizedTime OPTIONAL,
    endTime    GeneralizedTime OPTIONAL
    }

CCCEvent ::= SEQUENCE {
    userID             UserId OPTIONAL,
    timeOfEvent        GeneralizedTime OPTIONAL,
    locationOfEvent    Location OPTIONAL,
    cccMessages        SEQUENCE OF DSRCData, -- need to be CCC message containers
    initiatedActions   SEQUENCE OF InitiatedAction
    }

InitiatedAction ::=   INTEGER {
-- The attribute indicates to the TSP what kind of action has been performed
-- by the TC as a result of the CCC data
    vehicleWasStopped     (0),
        -- Vehicle was stopped following the Compliance Check Communication Result
    violationCaseIndicated(1),
        -- The Compliance Check Communication Result indicated that a violation
-- may have occurred.
    evidenceDataGathered  (2),
        -- The Compliance Check Communication and associated has been retained
-- as evidence of violation
    putOnTSPExceptionList (3),
        -- As a result of the Compliance Check Communication a request has been
-- made to add the UserID to the Toll Service Provider's Exception List
    putOnTCExceptionList  (4)
        -- As a result of the Compliance Check Communication the UserId has been
-- added to the Toll Charger Exception List
    -- 5-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

Location ::= SEQUENCE {
    positionOfLocation    Point OPTIONAL,
    location SEQUENCE {
        locationProviderId Provider,
        locationID         INTEGER
        } OPTIONAL
    }

DSRCData ::= SEQUENCE {
    dsrcRSEData            DSRCRSEData OPTIONAL,
    dsrcAttributesRead     SEQUENCE OF Attributes OPTIONAL,
    dsrcAttributesWritten  SEQUENCE OF Attributes OPTIONAL,
    dsrcAttrAuth           DSRCAttrAuth OPTIONAL,
    chargeObjectId         ChargeObjectId OPTIONAL
    }
```

```
DSRCRSEData::= SEQUENCE {
    transactionId          INTEGER,
    rSEDateTime            GeneralizedTime,
    transactionResult      ResultOp,
    transactionStatus      TransactionStatus,
    exceptionListMatch     ExceptionListEntry OPTIONAL,
    tariffID               TariffClassId OPTIONAL,
    fee                    Amount OPTIONAL,
    feeQualifier           FeeQualifier OPTIONAL,
    tollStationID          INTEGER OPTIONAL,
    laneDirectionNumber    SessionLocation OPTIONAL,
    typeOfTransaction      StationType OPTIONAL,
    vstData                ObeConfiguration OPTIONAL,
    ...
    }

DSRCAttrAuth ::= SEQUENCE OF SEQUENCE {
    attrOrigEncoding   BIT STRING, -- the bit string encoding of the AttributeList
    rndRSE             OCTET STRING,
    keyRef             INTEGER,
    authCode           OCTET STRING,
    result             AuthCheckResult
    }

TransactionStatus ::= INTEGER {
    completed                  (1),
    abortedAfterFirstPhase     (2),
    abortedAfterSecondPhase    (3)
    -- 4-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

AuthCheckResult ::= INTEGER {
    notChecked (1),
    oK         (2),
    nOK        (3)
    -- 4-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

-- Trust Objects

ListOfTrustObjectTypes ::= INTEGER {
    certificate           (0),
    symmetricKey          (1),
    cRL                   (2),
    cDP                   (3),
    oCSP                  (4),
    encryptedSymmetricKey (5)
    -- 6-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

ListOfTrustObjectPurposes ::= INTEGER {
    -- used for defining the intended usage of the key
    trustObjects       (0),
        -- Trust Object with an unspecified purpose
    dSRCCharging       (1),
        -- Trust Object to be used to validate authenticators from
        -- DSRC Charging Applications
    dSRCAC             (2),
        -- Trust Object to be used to calculate Access Credentials
        -- for DSRC Charging Applications
    oBEInterrogation   (3),
        -- Trust Object to be used to validate authenticators received
        -- during a Compliance Check Communication
```

```
    oBEInterrogationAC (4),
        -- Trust Obejct to be used to calculate Access Credentials
        -- for Compliance Check Communication
    sIGExceptionList   (5),
        -- Trust Object to be used in the validation of authenticity
        -- of received Exception Lists
    sIGContextData     (6),
        -- Trust Object to be used in the validation of authenticity
        -- of received Context Data
    sIGBillingDetails  (7),
        -- Trust Object to be used in the validation of authenticity
        -- of received Billing Details
    sIGFiscalObjects   (8),
        -- Trust Object to be used in the validation of authenticity
        -- of received Fiscal Objects
    sIGCommunication   (9),
        -- Trust Object to be used in the validation of authenticity
        -- of received messages using the infoExchangeAuthenticator
    eNCCommunication   (10)
        -- Trust Object to be used to decrypt received messages
    -- 11-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

TrustObjectStatus ::= INTEGER {
    valid      (0),
    expired    (1)
    -- 2-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

-- Quality Parameters

QualityParameterStatus ::= INTEGER {
    accordingToAgreedMeasuremtMethodology(0),
    measuredAsShortTermSample             (1),
    longTermAverage                       (2),
    singleWorstValue                      (3),
    trendWarning                          (4),
    initiateActions                       (5),
    escalationWarning                     (6),
    escalationWasActivated                (7)
    -- 8-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

-- Exception Lists

ExceptionListType ::= INTEGER {
    blacklist                    (1),
    whitelist                    (2),
    greylist                     (3),
    BlackListIncrementalUpdate   (4),
    WhiteListIncrementalUpdate   (5),
    GreyListIncrementalUpdate    (6)
    -- 7-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

ExceptionListEntry ::= SEQUENCE {
    userId         UserId,
    blockType      ExceptionListBlockType,
    reasonCode     ExceptionListReasonType,
    dateAndTime    GeneralizedTime
    }
```

```
ExceptionListBlockType ::= ENUMERATED {
    -- which kind of limitations shall result in putting this
    -- service user on the exception list
    allApplications                     (0),
    locallyBlocked                      (1),
    blockedForSchemesRequiringOdometer  (2)
    -- 3-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

ExceptionListReasonType ::= ENUMERATED {
    -- The reason code indicates why a user has been put on the exception list
    notToBeDisclosed                (0),
    obeDeactivated                  (1),
    obeIsStolen                     (2),
    temporaryTechnicalProblem       (3),
    suspicionOnTechnicalManipulation (4),
    latePayment                     (5), -- commercial conditions
    noPayment                       (6), -- commercial conditions
    contractHolderInsolvent         (7),
    normalUser                      (8)  -- to support whitelists
    -- 9-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

AbnormalOBEReasonCode ::= INTEGER {
    reasonNotToBeDisclosed      (0),
    obeIsDefect                 (1),
    obeIsNotWorkingProperly     (2),
    userShowsFraudBehaviour     (3),
    userShowsViolatingBehaviour (4)
    -- 5-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

-- Payment claim

PaymentClaimStatus ::= INTEGER {
    firstVersion          (0),
    firstAmendedVersion   (1),
    amendedVersion        (2)
    -- 3-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

TypeOfGoods::= INTEGER {
    toll           (0),
    discount       (1),
    creditnote     (2),
    penalty        (3),
    processingFee  (4)
    -- 5-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

-- EFC Context Data

ChargeLocation ::= SEQUENCE {
    description    UTF8String OPTIONAL,
    chargePointId  LocationOfStation,
    chargePoint    AbsolutePointCoordinates OPTIONAL,
    cpLocation     SessionLocation,
    eventType      SessionType
    }
```

```
LocationOfStation ::= INTEGER

SessionType::= INTEGER {  -- Type of session
    accessClosedSystem (0),
    exitClosedSystem   (1),
    openSystem         (2),
    checkPoint         (3)
    -- 4-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

-- GDF Toll Context layout option for GNSS based systems

SectionTollingLayoutGDF ::= SEQUENCE {
    efcLayer           EFClayer,
    ...
    }

EFClayer ::= SEQUENCE {
    eFCLayerId             INTEGER, -- unique within the toll regime
    tollContextName        UTF8String OPTIONAL,
    chargeObjects          SEQUENCE OF GDFChargeObject,
    referencedGDFsource    GDFSource,
    eFClayerVersion        VersionAndValidity
    }

GDFChargeObject ::= SEQUENCE {
    chargeObjectId             ChargeObjectId,
    roadOperatorId             Provider OPTIONAL,
    applicableLocationClass    INTEGER OPTIONAL,
    applicableTimeClasses      SEQUENCE OF TimeClassId OPTIONAL,
    tollRelevantLength         Distance,
    equivalentMeasuredLength   Distance OPTIONAL,
    tollRoad                   TollRoad,
    liabilityRules CHOICE {
        tollPoint      [1] GDFReference, -- check if point has been passed
        minTollPath    [2] SEQUENCE {
            firstPoint     GDFReference,
            secondPoint    GDFReference  -- check if point has been passed
            },
        minimumUsage   [3] INTEGER (1 .. 1000) -- in 0.1 %
        }
    }

TollRoad ::= SEQUENCE {
    efcNodeFrom    EFCnodeId, -- unique within the toll regime
    efcLink SEQUENCE {
        roadElementsTowardChargePoint SEQUENCE OF RoadElementsTowardChargePoint,
        junctionIdTo                  GDFReference
        },
    efcNnodeTo     EFCnodeId -- unique within the toll regime
    }

RoadElementsTowardChargePoint ::= SEQUENCE {
    junctionIdFrom     GDFReference,
    gDFroadElement     GDFReference
    }

GDFSource ::= SEQUENCE {
    dataProviderId Provider,
    typeId             INTEGER,
    versionNumberId    VersionNumberId
    }

GDFReference ::= SEQUENCE {
    datasetID      DataSetId,
    layerId        LayerId,
    sectionId      SectionId,
    objectId       ObjectId
    }
```

```
EFCnodeId ::= SEQUENCE {
    eFContextSpecificId   INTEGER, -- unique within the toll regime
    sectionName        UTF8String OPTIONAL,
    gDFspecificId      GDFReference OPTIONAL
    }

FeeModifiers ::= SEQUENCE {
    typeOfUsage            SEQUENCE OF ExemptedVehicleClasses OPTIONAL,
    volumeDiscounts        SEQUENCE OF SEQUENCE {
        timeInterval       TimeInterval OPTIONAL,
        forVehicleClasses  SEQUENCE OF ForVehicleClass OPTIONAL
        } OPTIONAL
    }

ExemptedVehicleClasses ::= INTEGER {
    handicappedPeople     (1),
    military              (2),
    police                (3),
    roadMaintenance       (4),
    circusTruck           (5),
    mobileShopTruck       (6),
    truckCarryingMilk     (7),
    truckCarryingTimber   (8),
    publicTransportBus    (9)
    -- 10-100 for future CEN and ISO use
    -- 101-255 for private use
    }

TimeInterval ::= INTEGER {
    day    (1),
    week   (2),
    month  (3),
    year   (4)
    -- 5-100 reserved for future CEN and ISO use
    -- 101-255 reserved for private use
    }

ForVehicleClass ::=   SEQUENCE { --- definition of volume related discounts
    vehicleClass       INTEGER,
    activationFee      SEQUENCE {
        amount         Amount,
        timeInterval   TimeInterval OPTIONAL
        } OPTIONAL,
    minAmountOfFee     Amount OPTIONAL,
    numberOfTrips      SEQUENCE {
        fromChargeObject          ChargeObjectId,
        toChargeObject            ChargeObjectId,
        perTimeClass              BOOLEAN OPTIONAL,
        pervehicleClass           BOOLEAN OPTIONAL,
        numberOfTripsDiscounts    SEQUENCE OF NumberOfTripsDiscount
        } OPTIONAL
    }

NumberOfTripsDiscount ::= SEQUENCE {
    minNumberOfTrips          INTEGER,
    discount                  INTEGER, -- in 0,1%
    onlyForAmountAboveLimit   BOOLEAN OPTIONAL
    }

-- General definitions

UserId ::= SEQUENCE {
    pan                   PersonalAccountNumber OPTIONAL, -- ISO 14906
    contractSerialNumber  ContractSerialNumber OPTIONAL,  -- ISO 14906
    licensePlateNumber    LPN OPTIONAL,
    obeID                 ObeId OPTIONAL                   -- ISO 17575-1
    }
```

```
RecordId ::= SEQUENCE {
    -- Message independent identifier for identifying records
    -- from other sources (e.g. enforcement operator)
    providerId Provider OPTIONAL,
    recordType RecordType OPTIONAL,
    uniqueId   INTEGER
    }

RecordType ::= ENUMERATED {
    cCCRecord             (1),
    imageRecord           (2),
    aNPRRecord            (3),
    classificationRecord  (4),
    operatorRecord        (5),
    dsrcData              (6)
    }

TollDeclarationId ::= SEQUENCE {
    issuerId       Provider,
    declarationId  INTEGER
    }

-- Level 4 definitions
-- May be imported from other modules later

DataSetId ::= INTEGER

LayerId ::= INTEGER

SectionId ::= INTEGER

ObjectId ::= INTEGER

VersionNumberId ::= INTEGER

NumberOfADUStruct ::= INTEGER

MessageAuthenticator ::= BIT STRING

SubRecordAuthenticator ::= BIT STRING

MessageIdentifier ::= INTEGER

RegimeID ::= INTEGER (0..65535)

TrustObjectCode ::= BIT STRING
    -- structure of the bit string to be defined by another standard

ExceptionListVersion ::= INTEGER

END
```

Annex B
(normative)

Protocol Implementation Conformance Statement

B.1 Guidance for completing the PICS proforma

B.1.1 Purposes and structure

The purpose of this PICS proforma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in this International Standard may provide information about the implementation in a standardized manner.

The PICS proforma is subdivided into clauses for the following categories of information:

— guidance for completing the PICS proforma;

— identification of the implementation;

— identification of the protocol;

— global statement of conformance;

— PICS proforma tables.

B.1.2 Abbreviations and conventions

The PICS proforma contained in this annex is comprised of information in tabular form in accordance with the guidelines presented in ISO/IEC 9646-7.

— Item column

The item column contains a number which identifies the item in the table.

— Item description column

The item description column describes in free text each respective item (e.g. parameters, timers, etc.). It implicitly means “is <item description> supported by the implementation?”.

— Status column

The following notations, defined in ISO/IEC 9646-7 are used for the status column:

m	mandatory - the capability is required to be supported.
o	optional - the capability may be supported or not.
n/a	not applicable - in the given context, it is impossible to use the capability.
x	prohibited (excluded) - there is a requirement not to use this capability in the given context.
o.i	qualified optional - for mutually exclusive or selectable options from a set. “i” is an integer which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table.
ci	conditional - the requirement on the capability (“m”, “o”, “x” or “n/a”) depends on the support of other optional or conditional items. “i” is an integer identifying a unique conditional status expression which is defined immediately following the table.

— Reference column

The reference column makes reference to this International Standard, except where explicitly stated otherwise.

— Support column

The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7, are used for the support column:

Y or y	supported by the implementation.
N or n	not supported by the implementation.
N/A, n/a or -	no answer required (allowed only if the status is n/a, directly or after evaluation of a conditional status).

NOTE As stated in ISO/IEC 9646-7, support for a received PDU requires the ability to parse all valid parameters of that PDU. Supporting a PDU while having no ability to parse a valid parameter is non-conformant. Support for a parameter on a PDU means that the semantics of that parameter are supported.

— Values allowed column

The values allowed column contains the type, the list, the range, or the length of values allowed. The following notations are used:

- range of values: <min value> .. <max value>
  example: 5 .. 20

- list of values: <value1>, <value2>, ..., <valueN>
  example: 2 ,4 ,6 ,8, 9
  example: '1101'B, '1011'B, '1111'B
  example: '0A'H, '34'H, '2F'H

- list of named values: <name1>(<val1>), <name2>(<val2>), ..., <nameN>(<valN>)
  example: reject(1), accept(2)

- length: size (<min size> .. <max size>)
  example: size (1 .. 8)

— Values supported column

The values supported column shall be filled in by the supplier of the implementation. In this column, the values or the ranges of values supported by the implementation shall be indicated.

— References to items

For each possible item answer (answer in the support column) within the PICS proforma, a unique reference exists, used, for example, in the conditional expressions. It is defined as the table identifier, followed by a solidus character "/", followed by the item number in the table. If there is more than one support column in a table, the columns are discriminated by letters (a, b, etc.), respectively.

EXAMPLE 1 5/4a is the reference to the first answer (i.e. contained in the first support column) of item 4 in Table B.5 of Annex B.

EXAMPLE 2 6/3b is the reference to the second answer (i.e. contained in the second support column) of item 3 in Table B.6 of Annex B.

— Prerequisite line

A prerequisite line takes the form: Prerequisite: <predicate>.

A prerequisite line after a clause or table title indicates that the whole clause or the whole table is not required to be completed if the predicate is FALSE.

B.1.3 Instructions for completing the PICS proforma

The supplier of the implementation shall complete the PICS proforma in each of the spaces provided. In particular, an explicit answer shall be entered in each of the support or supported column boxes provided, using the notation described previously.

If necessary, the supplier may provide additional comments in space at the bottom of the tables or separately.

B.2 Identification of the implementation

Identification of the Implementation Under Test (IUT) and the system in which it resides [the System Under Test (SUT)] shall be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information shall both be filled in if they are different.

A person who can answer queries regarding information supplied in the PICS shall be named as the contact person.

B.2.1 Date of the statement

.....

B.2.2 Implementation Under Test (IUT) identification

IUT name:

.....

.....

IUT version:

.....

B.2.3 System Under Test (SUT) identification

SUT name:

.....

.....

Hardware configuration:

.....

.....

.....

Operating system:

.....

B.2.4 Product supplier

Name:

.....

Address:

.....

.....

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

.....

B.2.5 Applicant (if different from product supplier)

Name:

.....

Address:

.....

.....

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

B.2.6 PICS contact person

(A person to contact if there are any queries concerning the content of the PICS)

Name:

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

.....

B.3 Identification of the protocol

This PICS proforma applies to the following standard:

ISO 12855 “Electronic fee collection — Information exchange between service provision and toll charging”.

B.4 Global statement of conformance

Are all mandatory capabilities implemented? (Yes/No) .....

NOTE Answering “No” to this question indicates non-conformance to the protocol specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming, on pages attached to the PICS proforma.

B.5 Roles

Table B.1 — Roles

Item	Supported role	Reference	Status	Support (Y/N)
1	Toll Charger	5.1	o.1	
2	Toll Service Provider	5.1	o.1	

o.1: it is mandatory to support at least one of these options.

B.6 Functionalities

Table B.2 — Functionalities

Item	Supported types	Reference	Status	Support (Y/N)
1	Basic protocol mechanisms	5.2.3	m	
2	Exchange Trust Objects	5.2.4	o	
3	Originating and providing EFC context data	5.2.5	o	
4	Manage exception lists	5.2.6	o	
5	Report Toll declarations	5.2.7	o	
6	Report Billing details	5.2.8	o	
7	Claim payment for service usage	5.2.9	o	
8	Exchange Enforcement data	5.2.10	o	
9	Exchange Quality assurance parameters	5.2.11	o	

B.7 Protocol data units

Table B.3 — InfoExchange APDU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	infoExchangeContent	6.1	m		6.1	m	
2	infoExchangeAuthenticator	6.1	o		6.1	m	

Table B.4 — Application Protocol Control Information

Item	Field name	Reference	Status	Support (Y/N)
1	messageOriginator	6.2.1	m	
2	informationSenderID	6.2.1	m	
3	informationRecipientID	6.2.1	m	
4	contextID	6.2.1	o	
5	messageIdentifier	6.2.1	m	
6	relatedMessageID	6.2.1	o	
7	aduType	6.2.1	m	
8	numberOfStructs	6.2.1	m	
9	messageDate	6.2.1	m	

Table B.5 — Application Data Units

Item	ADU	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	RequestADU	6.1, 6.3	o		6.1, 6.3	m	
2	AckADU	6.1, 6.4	m		6.1, 6.4	m	
3	StatusADU	6.1, 6.5	o		6.1, 6.5	m	
4	TrustObjectADU	6.1, 6.6	c.1		6.1, 6.6	c.2	
5	EFCContextDataADU	6.1, 6.7	c.3		6.1, 6.7	c.4	
6	ExceptionListADU	6.1, 6.8	c.5		6.1, 6.8	c.6	
7	ReportAbnormalOBEADU	6.1, 6.9	c.7		6.1, 6.9	c.8	
8	RetrieveTollDeclarationADU	6.1, 6.10	c.9		6.1, 6.10	c.10	
9	TollDeclarationADU	6.1, 6.11	c.11		6.1, 6.11	c.12	
10	BillingDetailsADU	6.1, 6.12	c.13		6.1, 6.12	c.14	
11	PaymentClaimADU	6.1, 6.13	c.15		6.1, 6.13	c.16	
12	RetrieveUserDetailsADU	6.1, 6.14	c.17		6.1, 6.14	c.18	
13	UserDetailsADU	6.1, 6.15	c.19		6.1, 6.15	c.20	
14	RetrieveCCCEventADU	6.1, 6.16	c.21		6.1, 6.16	c.22	
15	ReportCCCEventADU	6.1, 6.17	c.23		6.1, 6.17	c.24	
16	ReportQAADU	6.1, 6.18	c.25		6.1, 6.18	c.26	

c.1: IF (Table B.2/2) THEN o ELSE n/a.

c.2: IF (Table B.2/2) THEN m ELSE n/a.

c.3: IF (Table B.1/1 AND Table B.2/3) THEN o ELSE n/a.

c.4: IF (Table B.1/2 AND Table B.2/3) THEN m ELSE n/a.

c.5: IF (Table B.1/2 AND Table B.2/4) THEN o ELSE n/a.

c.6: IF (Table B.1/1 AND Table B.2/4) THEN m ELSE n/a.

c.7: IF (Table B.1/1 AND Table B.2/4) THEN o ELSE n/a.

c.8: IF (Table B.1/2 AND Table B.2/4) THEN m ELSE n/a.

c.9: IF (Table B.1/1 AND Table B.2/5) THEN o ELSE n/a.

c.10: IF (Table B.1/2 AND Table B.2/5) THEN m ELSE n/a.

c.11: IF (Table B.1/2 AND Table B.2/5) THEN o ELSE n/a.

c.12: IF (Table B.1/1 AND Table B.2/5) THEN m ELSE n/a.

c.13: IF (Table B.1/1 AND Table B.2/6) THEN o ELSE n/a.

c.14: IF (Table B.1/2 AND Table B.2/6) THEN m ELSE n/a.

c.15: IF (Table B.1/1 AND Table B.2/7) THEN o ELSE n/a.

c.16: IF (Table B.1/2 AND Table B.2/7) THEN m ELSE n/a.

c.17: IF (Table B.1/1 AND Table B.2/8) THEN o ELSE n/a.

c.18: IF (Table B.1/2 AND Table B.2/8) THEN m ELSE n/a.

c.19: IF (Table B.1/2 AND Table B.2/8) THEN m ELSE n/a.

c.20: IF (Table B.1/1 AND Table B.2/8) THEN m ELSE n/a.

c.21: IF (Table B.1/2 AND Table B.2/8) THEN o ELSE n/a.

c.22: IF (Table B.1/1 AND Table B.2/8) THEN m ELSE n/a.

c.23: IF (Table B.1/1 AND Table B.2/8) THEN o ELSE n/a.

c.24: IF (Table B.1/2 AND Table B.2/8) THEN m ELSE n/a.

c.25: IF (Table B.1/1 AND Table B.2/9) THEN o ELSE n/a.

c.26: IF (Table B.1/2 AND Table B.2/9) THEN m ELSE n/a.

Table B.6 — Request ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	requestedADUType	6.3	o		6.3	m	
2	ADUIdentifier	6.3	o		6.3	m	
3	numberOfADUStructs	6.3	o		6.3	m	

Table B.7 — Ack ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	ADUIdentifier	6.4	m		6.4	m	
2	numberOfADUStructs	6.4	o		6.4	m	
3	ackCode	6.4	m		6.4	m	
4	aduCode	6.4	o		6.4	m	

Table B.8 — Status ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	generalStatusCode	6.5	m		6.5	m	
2	messageStatusCode	6.5	o		6.5	m	

Table B.9 — TrustObjects ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	trustObjectID	6.6	c.1		6.6	c.1	
2	startValidity	6.6	c.2		6.6	c.1	
3	endValidity	6.6	c.2		6.6	c.1	
4	trustObjectStatus	6.6	c.1		6.6	c.1	
5	typeOfTrustObject	6.6	c.1		6.6	c.1	
6	purposesOfTrustObject	6.6	c.1		6.6	c.1	
7	trustObject	6.6	c.1		6.6	c.1	

c.1: IF (Table B.2/4) THEN m ELSE n/a.

c.2: IF (Table B.2/4) THEN o ELSE n/a.

Table B.10 — EFC Context type support

Item	Supported type	Reference	Status	Support (Y/N)
1	DSRC Tolling	6.7	o.1	
2	Autonomous tolling	6.7	o.1	

o.1: It is mandatory to support at least one of these options.

Table B.11 — EFCContextData ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	contextInterrelations	6.7	c.4		6.7	c.3	
2	contextId	6.7	c.3		6.7	c.3	
3	Iso175753ADU	6.7	c.3		6.7	c.3	
4	gnssGDFLayout	6.7	c.4		6.7	c.3	
5	feeModifiers	6.7	c.4		6.7	c.3	
6	tollContextOverview	6.7	c.2		6.7	c.1	
7	tariffTable	6.7	c.2		6.7	c.1	
8	localVehicleClassDefinition	6.7	c.2		6.7	c.1	
9	timeClassDefinition	6.7	c.2		6.7	c.1	
10	userClassDefinition	6.7	c.2		6.7	c.1	
11	chargeLocations	6.7	c.2		6.7	c.1	

c.1: IF (Table B.5/5 AND Table B.10/1) THEN m ELSE n/a.

c.2: IF (Table B.5/5 AND Table B.10/1) THEN o ELSE n/a.

c.3: IF (Table B.5/5 AND Table B.10/2) THEN m ELSE n/a.

c.4: IF (Table B.5/5 AND Table B.10/2) THEN o ELSE n/a.

Table B.12 — ExceptionList ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	exceptionListVersion	6.8	c.1		6.8	c.1	
2	exceptionListType	6.8	c.1		6.8	c.1	
3	exceptionValidityStart	6.8	c.2		6.8	c.1	
4	exceptionValidityEnd	6.8	c.2		6.8	c.1	
5	exceptionListEntries	6.8	c.1		6.8	c.1	

c.1: IF (Table B.5/6) THEN m ELSE n/a.

c.2: IF (Table B.5/6) THEN o ELSE n/a.

Table B.13 — ReportAbnormalOBE ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	userID	6.9	c.1		6.9	c.1	
2	dateandTime	6.9	c.1		6.9	c.1	
3	abnormalOBEReasonCode	6.9	c.1		6.9	c.1	

c.1: IF (Table B.5/7) THEN m ELSE n/a.

Table B.14 — RetrieveTollDeclaration ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	userID	6.10	c.2		6.10	c.1	
2	startTime	6.10	c.2		6.10	c.1	
3	endTime	6.10	c.2		6.10	c.1	

c.1: IF (Table B.5/8) THEN m ELSE n/a.

c.2: IF (Table B.5/8) THEN o ELSE n/a.

Table B.15 — TollDeclaration ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	tollDeclarationId	6.11	c.1		6.11	c.1	
2	gnssTollDeclaration	6.11	c.1		6.11	c.1	

c.1: IF (Table B.5/9) THEN m ELSE n/a.

Table B.16 — BillingDetails ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	billingDetailsId	6.12	c.1		6.12	c.1	
2	contextId	6.12	c.1		6.12	c.1	
3	userId	6.12	c.2		6.12	c.1	
4	period	6.12	c.1		6.12	c.1	
5	billingDetailAmount	6.12	c.1		6.12	c.1	
6	usageDetails	6.12	c.2		6.12	c.1	
7	refTollDeclaration	6.12	c.2		6.12	c.1	
8	associatedEventData	6.12	c.2		6.12	c.1	

c.1: IF (Table B.5/10) THEN m ELSE n/a.

c.2: IF (Table B.5/10) THEN o ELSE n/a.

Table B.17 — PaymentClaim ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	paymentClaimId	6.13	c.1		6.13	c.1	
2	startDateTime	6.13	c.1		6.13	c.1	
3	endDateTime	6.13	c.2		6.13	c.1	
4	userId	6.13	c.2		6.13	c.1	
5	paymentClaimAmount	6.13	c.1		6.13	c.1	
6	paymentClaimStatus	6.13	c.1		6.13	c.1	
7	typeOfGoods	6.13	c.2		6.13	c.1	
8	associatedBillingDetails	6.13	c.1		6.13	c.1	

c.1: IF (Table B.5/11) THEN m ELSE n/a.

c.2: IF (Table B.5/11) THEN o ELSE n/a.

Table B.18 — RetrieveUserDetails ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	paymentClaimId	6.14	c.1		6.14	c.1	
2	startDateTime	6.14	c.2		6.14	c.1	
3	endDateTime	6.14	c.2		6.14	c.1	

c.1: IF (Table B.5/12) THEN m ELSE n/a.

c.2: IF (Table B.5/12) THEN o ELSE n/a.

Table B.19 — ProvideUserDetails ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	originaluserIdRequest	6.15	c.1		6.15	c.1	
2	userId	6.15	c.1		6.15	c.1	
3	statusFlag	6.15	c.2		6.15	c.1	
4	listOfUserParameters	6.15	c.2		6.15	c.1	

c.1: IF (Table B.5/13) THEN m ELSE n/a.

c.2: IF (Table B.5/13) THEN o ELSE n/a.

Table B.20 — RetrieveCCEvent ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	userId	6.16	c.2		6.16	c.1	
2	startTime	6.16	c.2		6.16	c.1	
3	endTime	6.16	c.2		6.16	c.1	

c.1: IF (Table B.5/14) THEN m ELSE n/a.

c.2: IF (Table B.5/14) THEN o ELSE n/a.

Table B.21 — ReportCCEvent ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	timeOfEvent	6.17	c.2		6.17	c.1	
2	locationOfEvent	6.17	c.2		6.17	c.1	
3	cccMessages	6.17	c.1		6.17	c.1	
4	initiatedActions	6.17	c.1		6.17	c.1	

c.1: IF (Table B.5/15) THEN m ELSE n/a.

c.2: IF (Table B.5/15) THEN o ELSE n/a.

Table B.22 — ReportQA ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	qualityParameterID	6.18	c.1		6.18	c.1	
2	qualityParameterName	6.18	c.2		6.18	c.1	
3	qualityParameterValue	6.18	c.2		6.18	c.1	
4	qualityParameterStatus	6.18	c.1		6.18	c.1	

c.1: IF (Table B.5/16) THEN m ELSE n/a.

c.2: IF (Table B.5/16) THEN o ELSE n/a.

Annex C
(informative)

How to use road network data attributes coded in GDF format

C.1 General

ISO 14825 defines, among other things, how to code digital road maps. This International Standard is widely used to exchange information between data provider and data user. Several companies are providing digital maps according to this International Standard; however, the used identifier and names of the internal data elements are defined using proprietarily identifier.

It is anticipated that the existing digital road maps will provide a valuable base defining a toll road network within a general road network. Using a full road network data base would cover the need of proxies allowing tolled object detection algorithms to evaluate more than the minimum knowledge of the toll road network. This covers also the need of smart clients, especially if their functionality is combined with vehicle navigation application using a full navigation road data set inside the vehicle. Even if it is out of the scope of this International Standard, it may also be used by Road Operators or Toll Chargers to inform partner entities on how local rules should be applied. In this case the quality of the underlying digital road map should fulfil the requirements of the tolling application. This mainly focuses on actuality and completeness.

However, ISO 14825 does not cover toll relevant attributes even if the simple qualifier, that a certain road is a toll road, is supported. Therefore this International Standard defines an add-on to be used in conjunction with a clearly identified standard digital map coded in GDF. This add-on is defined in an ASN.1 coded data structure defining the toll relevant attributes in the GDF view of an additional EFC layer. Within this EFC layer, references to the digital map are used, feeding a tolled object evaluation algorithm with sufficient topologic data.

This annex provides some guidelines on how to use references to GDF road data files defining context data for tolling.

In all cases the specific underlying digital road network data base should be of a quality sufficient for toll applications.

C.2 A short introduction to GDF coded road maps

As defined in ISO 14825 digital road maps are defined mainly in 3 levels. The highest level 2 (for roads) defines the road network from the view of a driver finding the way to a destination. Roads are defined just as a link between intersections. Curves in the road are neglected and even complex structures as a highway crossing are defined just as an intersection. This view is sufficient and optimal for navigation.

In the level 1 of the GDF road map layer, the roads are split into road elements connecting junctions at both sides of the road element. A road element is defined as the smallest element of a road having a single set of properties. Road elements are defined as straight lines even if the actual road has some curves. There are no geographic coordinates defined at this level. Just references from the level 1 junctions to the level 0 may provide geographic locations. All the road elements provide the conceivable trajectories a vehicle can use when moving. This includes small areas like parking areas or industry yards.

Finally, the level 0 finally defines the topology. This is done by defining all relevant nodes accurate with geographic coordinates and links between them without defining their properties. Links between these nodes may be shaped to better fit the reality.

All these nodes may be referenced by defined junctions at level 1. And all road elements interconnecting junctions at level 1 may be referenced from level 2. With that the GDF standard provides a very practical view on road networks that may be re-used for defining sectioned toll roads.

C.3 The EFC layer as add-on to standard GDF coded road map data

In general the EFC layer defines all the EFC specific attributes required in a Front End, recognizing a sectioned toll road tolled object and selecting the tolled object specific parameters required for calculating the fee. The toll road network is defined as a sequence of tolled objects. This network uses common node identifiers at both sides of a toll road segment, allowing for the reconstruction of the consecutive logical order of tolled objects.

NOTE This may be used in the Front End recognizing “missing” tolled objects if the timing supports the assumption that the tolled object in between should had been used. According to the level 2 definition of GDF the EFC node is a logical location without a relation to any topology or geographic coordinates. However, the toll road is defined as a link between two EFC nodes described as a sequence of road elements according to the level 1 definition of GDF. These road elements have junctions (nodes) at both sides of each of the road elements and with this there is a relation of a junction by a reference to a node in level 0 of the digital map. These level 0 nodes provide the geographic coordinates and the link to the “real world”.

Herewith two consecutive toll road sections may be connected to the same EFC node using different level 1 junctions and with that different geographic coordinates as defined in level 0. With this an EFC node may cover more than one geographic location even if this is not defined explicitly in the EFC layer.

A Front End evaluation algorithm may recognize this and extract a link between these junctions using level 1 (purely GDF) definitions. With this both the ends of toll road sections are connected in any trajectory the digital map provides.

This mechanism may also be used for deciding on the use of a tolled object if other roads are close by. Here the logical connection supports the assumption that a tolled object may only have been used if the full vehicle trajectory follows a logical connection toward the tolled object.

The relation between the EFC layer and the road network layers as defined in ISO 14825 is illustrated in Figure C.1. It can be seen that two consecutive EFC links are not necessarily connected together. In complex intersections providing more than one trajectory reaching the entrance or leaving the exit, the “area” of multiple trajectories should be associated with the complex structure of an EFC node. Any movement within this area between the end of the last and the beginning of the next toll link does not cause any charging on its own.

Within this International Standard the definition of the GDF standard should not be repeated. Also the exchange of GDF data sets between Front End and Back End is seen as outside the scope. However, the definitions of ISO/TS 17575-2 can be applied for GDF type data transactions.

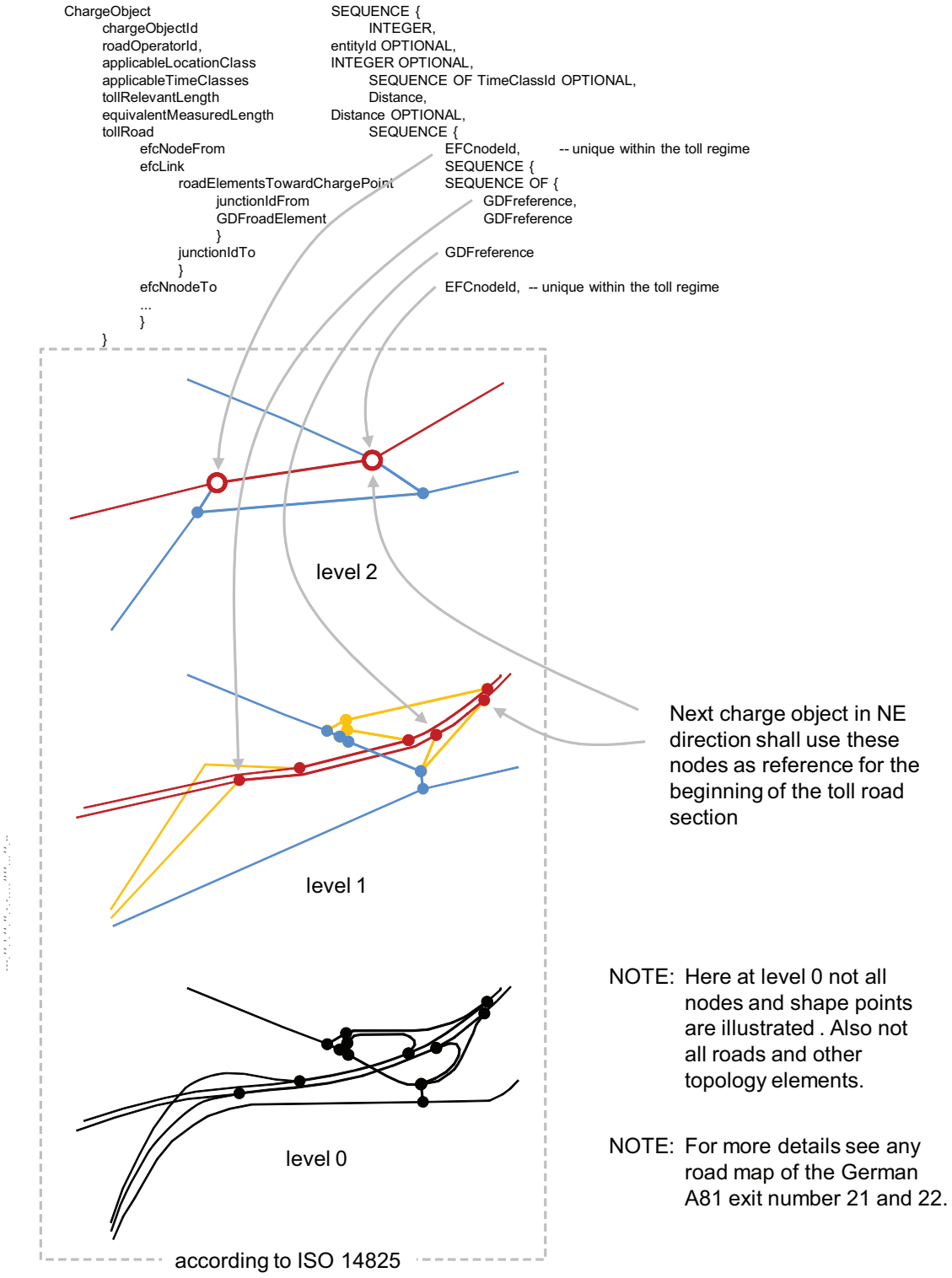


Figure C.1 — The relation between the EFC and ISO 14825 compatible road network layer

Annex D
(informative)

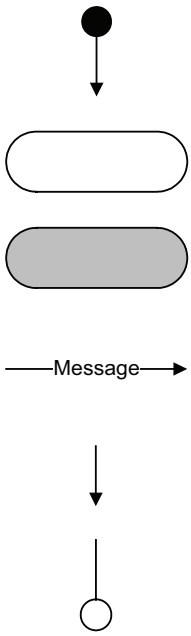
Example enforcement process applying standardized message exchanges

D.1 General

The following example illustrates the use of the standardized functionalities of the interface for the exchange of enforcement data.

D.2 Symbols

In the described processes the following symbols are used.



Solid circles represent a starting point of a data exchange

White rounded corner boxes indicate mandatory responsibilities and related activities within the processing of data

Grey rounded corner boxes indicate optional responsibilities and related activities within the processing of data

Horizontal arrows indicate information exchanged between roles as activities performed within responsibilities. The name of the message is attached to the arrow.

Vertical arrows represent execution steps within the data exchange

Empty circles represent an ending point of a data exchange

D.3 Process Phases

The handling of a non-conformant activity during a compliance check is divided into five different phases.

D.3.1 Phase I: Perform compliance check and request missing toll declarations

A non-conformant activity is established when no payment or a reduced payment for the usage of a toll regime is recorded during a compliance check. This compliance check may be performed by CCC, video equipment or by any other means.

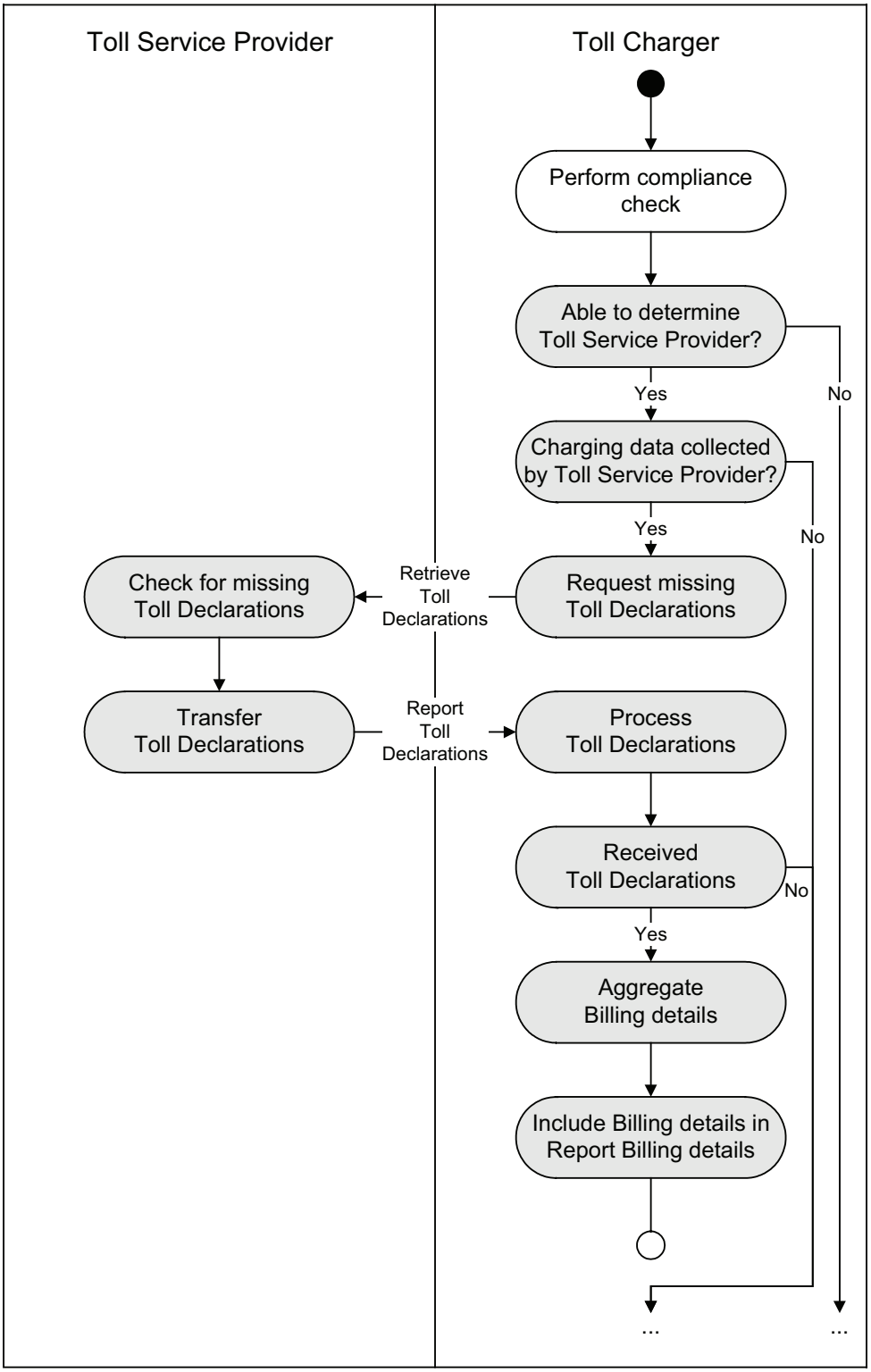


Figure D.1 — Phase I: Perform compliance check and request missing toll declarations

If non-conformant activity is detected and the Toll Service Provider is not known (e.g. no OBE in the vehicle) the process will be continued with phase II.

If the Toll Charger is unable to identify any Toll Service Provider who wants to handle the enforcement case, he has to issue the Enforcement Notice on its own without any further exchange of information via this interface. He has to try to obtain the information about the registered keeper of the vehicle from another source in phase IV.

If non-conformant activity is detected and the Toll Service Provider is known (e.g. blocked OBE in the vehicle, missing toll declarations for an equipped vehicle …) and the charging data for a toll regime was collected by a Toll Service Provider the Toll Charger can ask the Toll Service Provider to transmit any missing or not transferred toll declarations for this case of non-conformant activity. He issues a “Retrieve specific Toll Declarations” message to provide any recorded toll declarations (e.g. license plate, date/time, category …) for the non-conformant activity.

If the Toll Service Provider confirms the existence of missing Billing details for a given compliance check, he transmits all requested missing toll declarations. This information is then aggregated at the Toll Charger together with the normal toll declarations and included into the Report Billing details process. If the delivered toll declarations are sufficient to clarify the case of non-conformant activity the case can be closed.

D.3.2 Phase II: Build inferred Billing details

If the Toll Service Provider declines the existence of missing toll declarations or the Charging data was originally recorded on the road infrastructure of the Toll Charger, the Toll Charger can try to build inferred toll declarations. If this is possible, he may aggregate them into Billing details and include them into a request to enrich these inferred Billing details. The Toll Service Provider checks whether he accepts the inferred Billing details and acknowledges or disputes them. If the Toll Service Provider does not decline these inferred Billing details, the case of non-conformant activity can be closed.

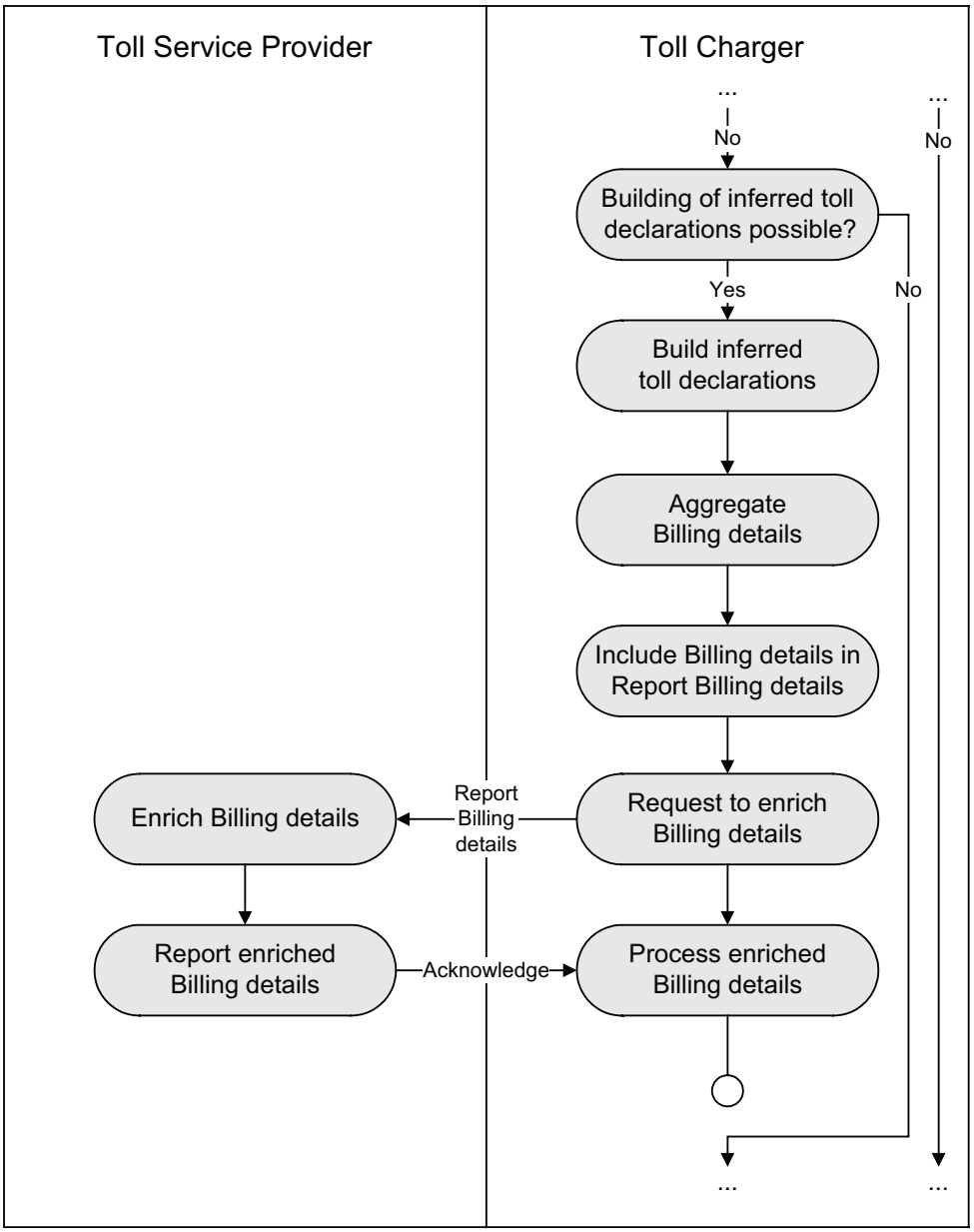


Figure D.2 — Phase II: Build inferred Billing details

D.3.3 Phase III: Request Payment guarantee

If the generation of inferred Billing details is not possible or the inferred Billing details are declined by the Toll Service Provider, the Toll Charger may request the handling of the payment for the non-conformant activity under a payment guarantee from the Toll Service Provider. This is typically handled outside a formal interface.

If the Toll Service Provider accepts the handling of the case of non-conformant activity under a payment guarantee, the charge associated with it is included as a Billing detail in the payment process. The case of non-conformant activity can be closed.

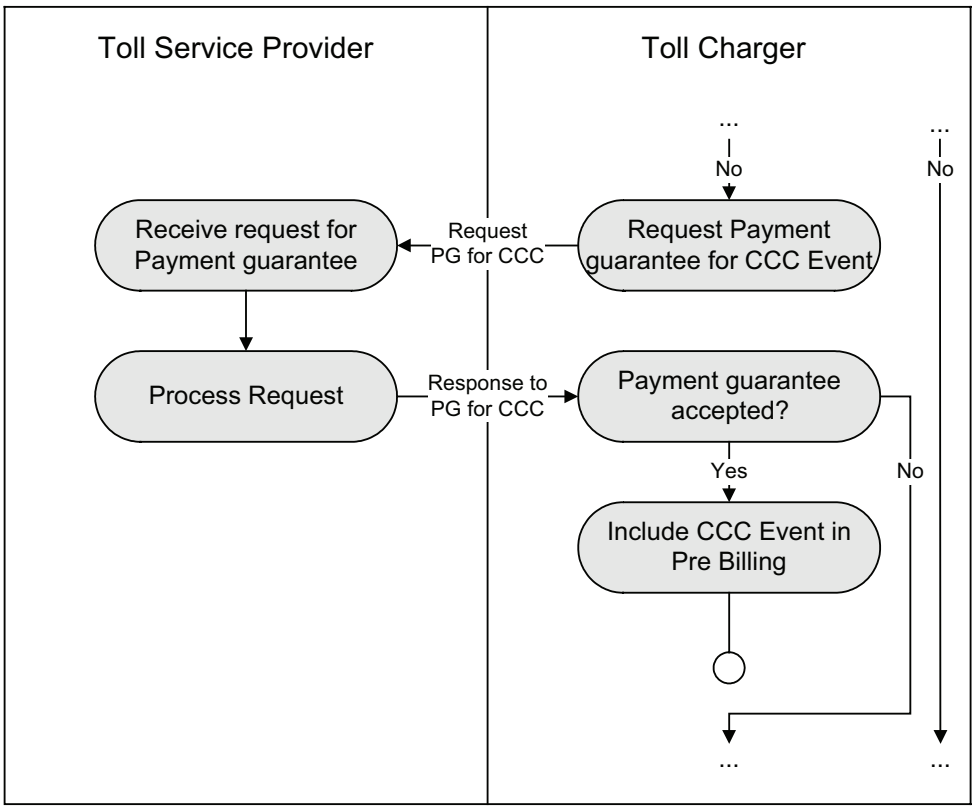


Figure D.3 — Phase III: Request Payment guarantee

D.3.4 Phase IV: Request Service User address details

The Toll Charger can retrieve the Service User's address details of the account holder from the Toll Service Provider by issuing a “Retrieve User Details” message to prepare the Enforcement Notice on its own if the Toll Service Provider declines the handling of the payment for the case of non-conformant activity under a payment guarantee.

The Toll Service Provider can only send any address details to the Toll Charger if they agreed on this process and if the Toll Service Provider is allowed to share it from a legal perspective (e.g. local data protection regulations). If he is not allowed to pass on this information or he does not have any current address information (e.g. no customer anymore), the Toll Service Provider may decline the request. In this case the Toll Charger has finally to try to obtain the information about the registered keeper of the vehicle from another source (e.g. vehicle registry).

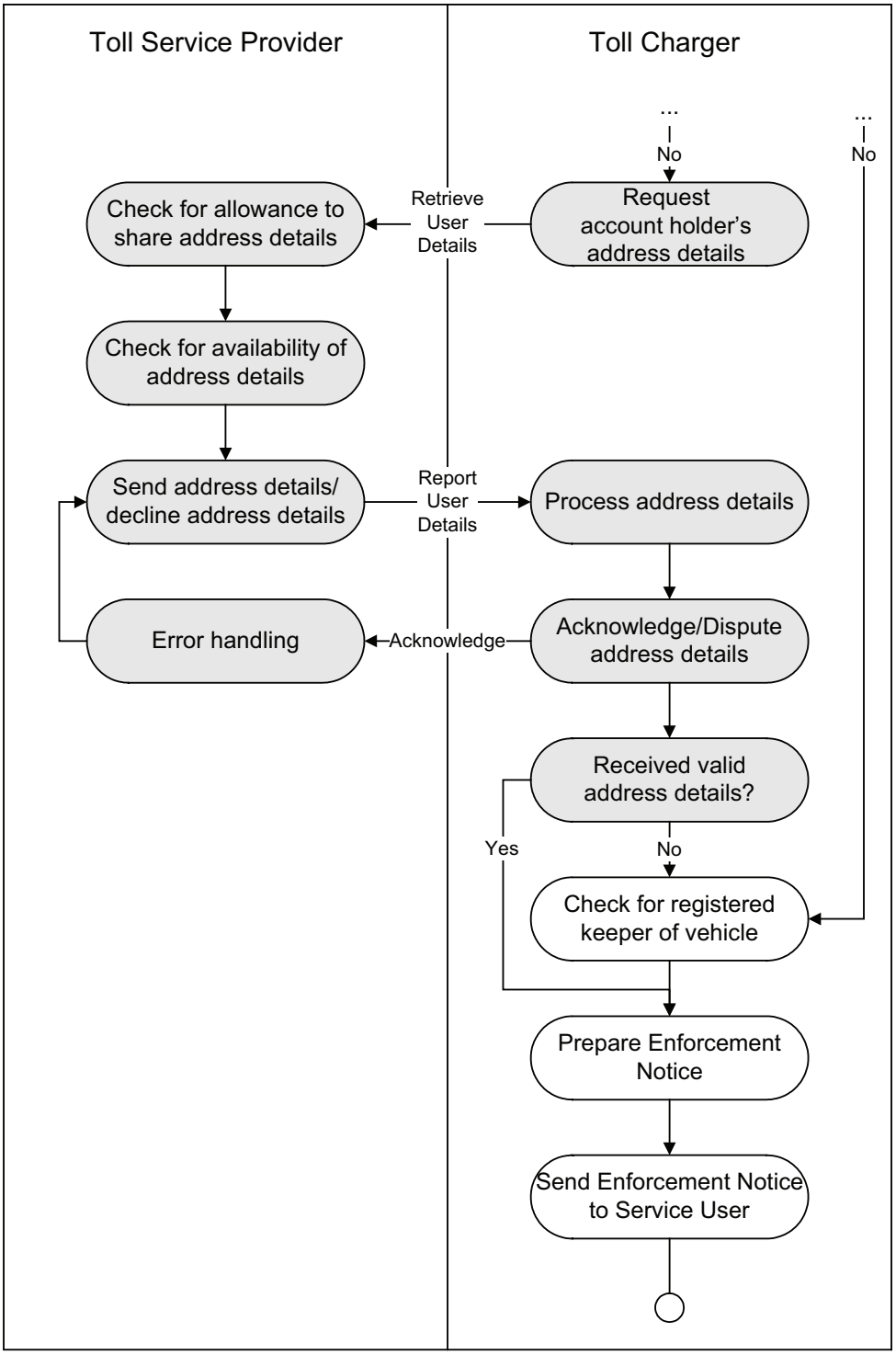


Figure D.4 — Phase IV: Request Service User address details

With enough information about the Service User, the Enforcement Notice is prepared and sent to the responsible Service User.

Annex E
(informative)

Data flow in a toll domain

The following diagram shows a typical data flow in a toll domain between vehicle/OBE, Toll Charger, Toll Service Provider and Service User.

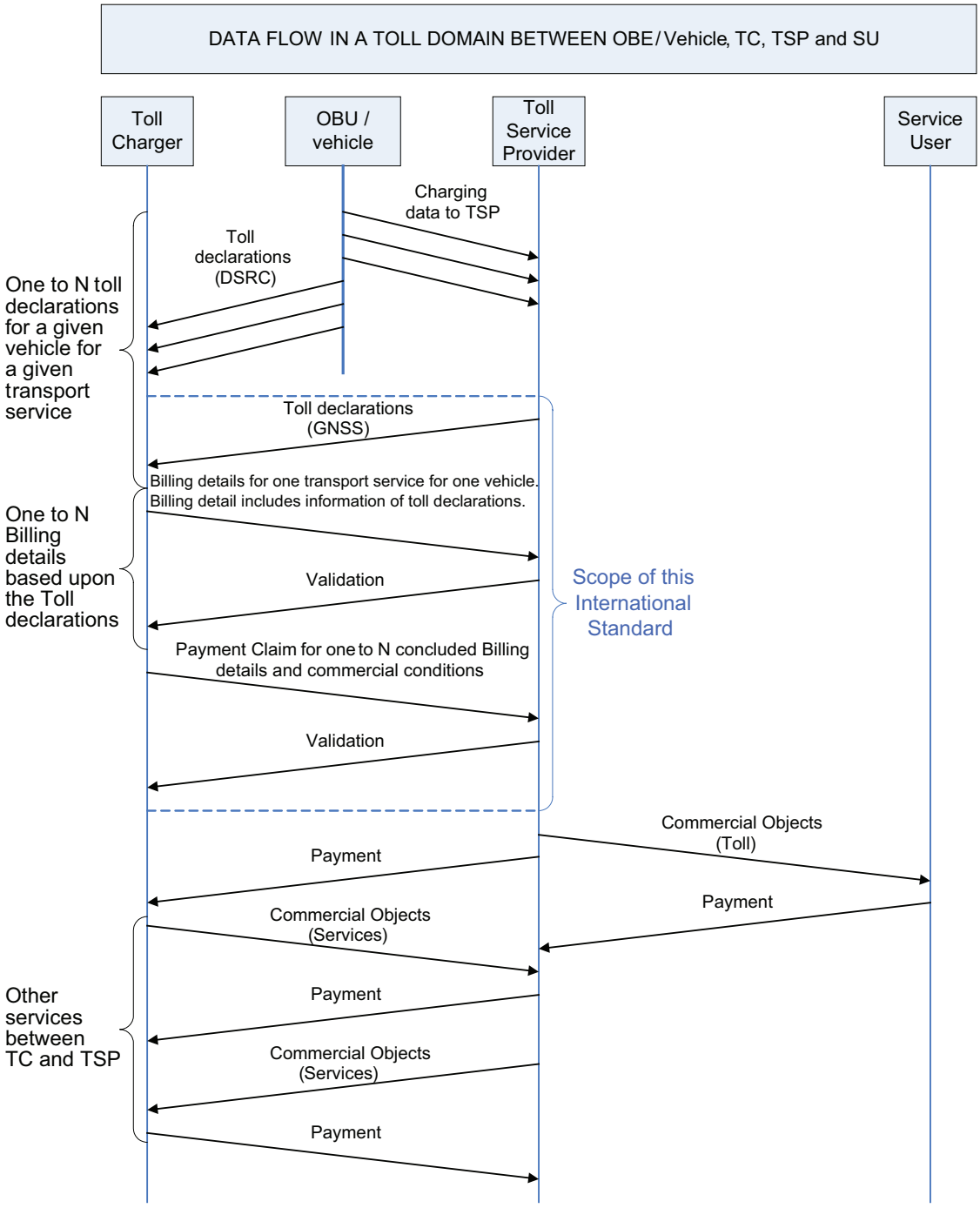


Figure E.1 — Data flow in a toll domain

In GNSS-based tolling, toll relevant data produced by the Front-End systems and sent to the Toll Service Provider's back-office systems is called Charging Data. Charging Data, with or without further processing, is transferred via back-office communication to the Toll Charger, as a Toll declaration.

In DSRC-based tolling as well as in GNSS-based tolling, Toll relevant data produced by the DSRC module of an OBE, and/or any other information (measurements, video, pictures …), collected by the RSE and sent to the central equipment of a Toll Charger is called Toll declaration.

Independent of the type of toll regime (GNSS based tolling or DSRC based tolling), a given transport service is fully defined (according to the rules of the Toll regime) by one or several Toll declarations.

This information associated with a given transport service, completed by the amount due calculated according to the rules of the toll regime, is called a Billing detail. A Billing detail is exchanged and confirmed between Toll Service Provider and Toll Charger and is the ultimate basis for any claim from the Toll Charger to the Toll Service Provider and subsequently from the Toll Service Provider to the Service User.

For a given Transport Service for a given vehicle:

— in GNSS based tolling, there may be Toll declarations available from the TSP and/or supportive data from measurement/verification of the vehicle made by the TC (such as CCC transaction, transit information or any other kind of information collection);

— in DSRC based tolling, there may be toll declarations available from the OBE via the RSE and/or supportive data from measurement/verification of the vehicle made by the TC (such as CCC transaction, transit information or any other kind of information collection).

Bibliography

[1] ISO/TS 12813, *Electronic fee collection — Compliance check communication for autonomous systems*

[2] ISO/TS 13141, *Electronic fee collection — Location augmentation communication for autonomous systems*

[3] ISO 14825, *Intelligent transport systems — Geographic Data Files (GDF) — GDF5.0*

[4] ITU-T Recommendation X.207 (1994) | ISO/IEC 9545:1994*, Information technology — Open Systems Interconnection — Application Layer structure*

[5] ITU-T Recommendation X.217 (1995) | ISO/IEC 8649:1996, *Information technology — Open Systems Interconnection — Service Definition for the Association Control Service Element*

[6] ITU-T Rec. X.901 | ISO/IEC 10746-1, *Information technology — Open distributed processing — Reference model: Overview*

[7] ITU-T Rec. X.902 | ISO/IEC 10746-2, *Information technology — Open Distributed Processing — Reference model: Foundations*

[8] ITU-T Rec. X.903 | ISO/IEC 10746-3, *Information technology — Open distributed processing — Reference Model: Architecture*

[9] ITU-T Rec. X.904 | ISO/IEC 10746-4, *Information technology — Open distributed processing — Reference Model: Architectural semantics*

[10] ITU-T Rec. X.911 | ISO/IEC 15414*, Information technology — Open Distributed Processing — Reference model: Enterprise language*

[11] ITU-T Rec X.920 | ISO/IEC 14750, *Information technology — Open Distributed Processing — Interface Definition Language*

[12] ISO/TS 14907-1, *Electronic fee collection — Test procedures for user and fixed equipment — Part 1: Description of test procedures*

[13] Directive 2004/52/EC of the European Parliament and of the Council on interoperability of electronic road toll systems in the community

[14] Decision 2009/750/EC, Commission Decision of 6 October 2009 on the definition of the European Electronic Toll Service and its technical elements

[15] ISO/TS 17575-2, *Electronic fee collection — Application interface definition for autonomous systems — Part 2: Communication and connection to the lower layers*

ICS 03.220.20; 35.240.60

Price based on 75 pages