INTERNATIONAL STANDARD

ISO 12855

Second edition
2015-12-15

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 12855:2015(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 204, *Intelligent transport systems.*

This second edition cancels and replaces the first edition (ISO 12855:2012), which has been technically revised. The following changes have been made:

— adding new Application Data Units (ADUs) due to comments received from National Bodies;

— aligning the ASN.1 data definitions with the current versions of EN 14906, and ISO 17575 (all parts);

— moving the ASN.1 module from Annex A to an external text file, in a format that can be processed by ASN.1 compilers;

— clarifying the semantics of parameters in ADUs;

— aligning the structure of all major clauses in a consistent manner to improve readability.

Introduction

The widespread use of tolling requires provisions for users of vehicles that circulate 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 and Decision 2009/750/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. With respect to ISO 17573, this International Standard

— adopts its definitions of terms and concepts and basic system functionalities and structure,

— uses its terminology, and

— specifies the interfaces therein identified.

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

Figure 1 shows the scope of the group of electronic fee collection (EFC) related standards based upon the architecture standard.

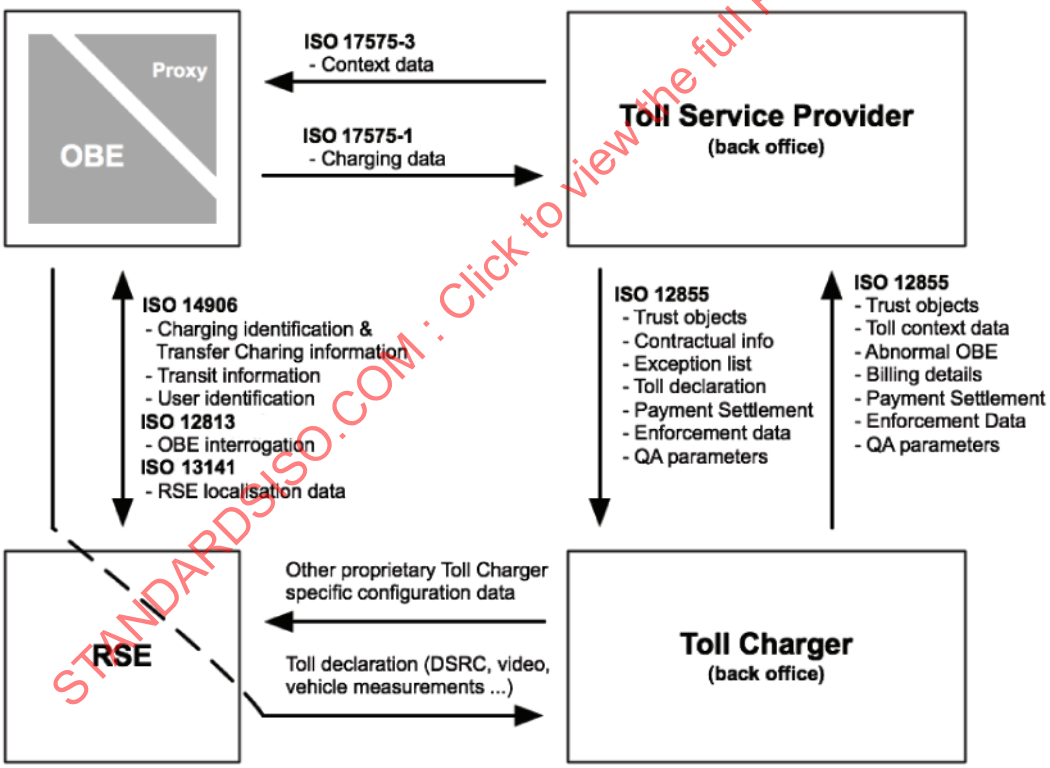

Figure 1 — Scope of EFC related standards

A given transport service for a given vehicle is fully identified by one or several toll declarations made available to the Toll Charger (TC). 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 finalized by the 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.). That means that the

TC has the authority to decide on the amount due, even if it decides to assign the Toll Service Provider (TSP) the task to calculate the amount due.

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 TC and/or the relevant TSP; they are finalized by the TC.

The TC elaborates and makes the payment claims (or toll payment claims) available to each TSP, according to the bilateral agreements it has with each TSP, 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 TSP.

This International Standard defines a set of interactions in support of technical interoperability between back-office systems of TCs and TSPs. The EFC-service and the EFC System model on which this International Standard is based is defined in ISO 17573.

This International Standard does not provide a full solution for interoperability, and it does not define other parts of the EFC system, other services, other technologies and non-technical elements of interoperability.

The development of a common European Electronic Toll Service (EETS) as a part of the European EFC Directive (2004/52/EC) also calls for the definition of an interoperable EFC service. This International Standard provides for effective support for the work on the definition of EETS. Details on the usage of this International Standard for the EETS are provided in Annex F.

This International Standard identifies and specifies the set of Application Protocol Data Units 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, a number of computational interfaces are identified and interactions in terms of sequences of Application Protocol Data Units 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. including systems that automatically read 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 (see Annex C and Annex D), and

— an example on how to use this International Standard for the European Electronic Tolling Service (EETS) (see Annex F).

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

It is defined as a toolbox standard of transactions and Application Protocol Data Units (APDUs), 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.

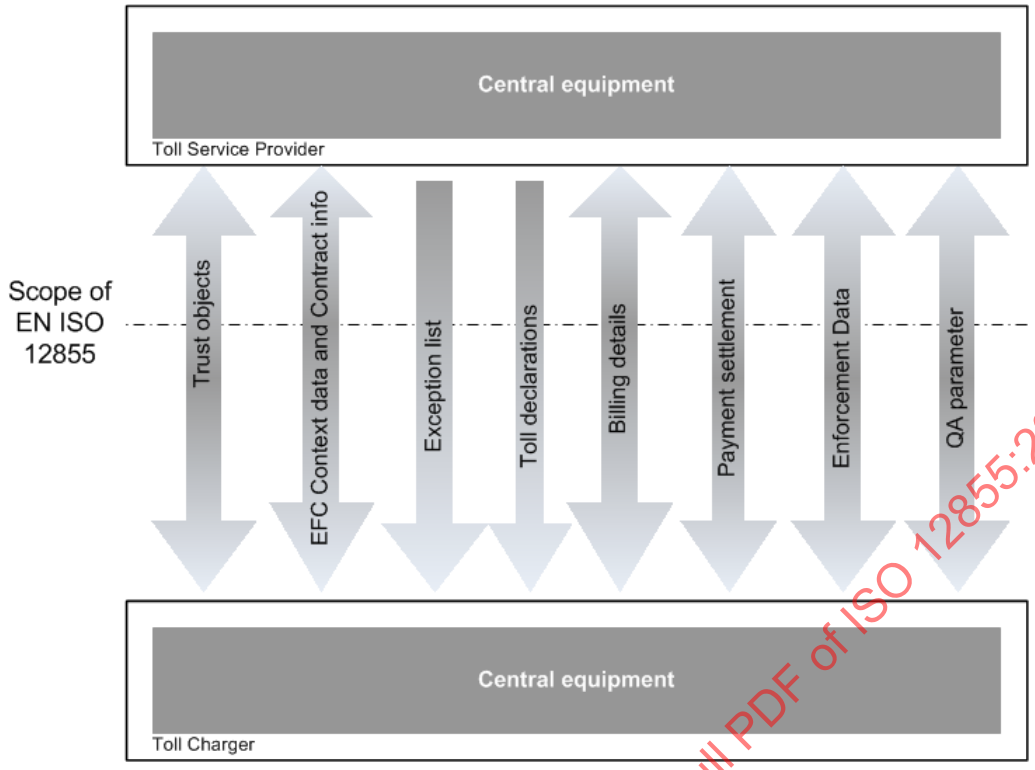

Figure 2 — Scope of this International Standard

This International Standard is not applicable to

— any communication between Toll Charger (TC) or Toll Service Provider (TSP) with any other involved party,

— any communication between elements of the TC and the TSP that is not part of the back office communication,

— processes regarding payments and exchanges of fiscal, commercial or legal accounting documents, and

— definitions of service communication channels, protocols and service primitives to transfer the APDUs.

2 Normative references

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

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

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

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/IEC 9594-8:2014, *Information technology — Open Systems Interconnection — The Directory — Part 8: Public-key and attribute certificate frameworks*

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

ISO/IEC 9797-2:2011, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*

ISO/IEC 10118-3, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

ISO/IEC 11770-3, *Information technology — Security techniques — Key management —Part 3: Mechanisms using asymmetric techniques*

ISO/IEC 14888-2:2008, *Information technology — Security techniques — Digital signatures with appendix — Part 2: Integer factorization based mechanisms*

ISO 14906:2011/Amd1:2015, *Electronic fee collection — Application interface definition for dedicated short-range communication*

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

ISO 17575-1:—[1)], *Electronic fee collection — Application interface definition for autonomous systems — Part 1: Charging*

ISO 17575-3:—[1)], *Electronic fee collection — Application interface definition for autonomous systems — Part 3: Context data*

ISO/IEC 18033-2:2006, *Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers*

ISO/TS 19299:2015, *Electronic fee collection — Security framework*

IETF RFC 2634, *Enhanced Security Services for S/MIME, June 1999*

IETF RFC 4347, *Datagram Transport Layer Security, April 2006*

IETF RFC 5035, *Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility, August 2007*

IETF RFC 5246, *The Transport Layer Security (TLS) Protocol, August 2008*

IETF RFC 5746, *Transport Layer Security (TLS) Renegotiation Indication Extension, February 2010*

IETF RFC 6040, *Tunnelling of Explicit Congestion Notification, February 2013*

3 Terms and definitions

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

3.1
billing detail
information needed to determine or verify the amount due for the usage of a given service

Note 1 to entry: If the data are accepted by both the Toll Charger and the Toll Service Provider then the term used is “concluded billing detail”, which can be used to issue a payment claim.

Note 2 to entry: For a given transport service, the billing detail refers 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.

1) To be published.

3.2
black list
list of users for which the service provider denies contractual responsibility

Note 1 to entry: The service provider in the context of this International Standard is the Toll Service Provider (TSP).

3.3
charge report
information containing road usage and related information originated at the Front End

Note 1 to entry: In European Decision 2009/750/EC, a charge report is referred to as a "toll declaration".

[SOURCE: ISO 17575-1:—, 3.6]

3.4
charging data
relevant data on the usage of a certain service

3.5
computational specification
decomposition of a system into objects performing individual functions and interacting at well-defined interfaces

3.6
electronic fee collection
EFC
fee collection by electronic means

3.7
enforcement
measures or actions performed to achieve compliance with laws, regulations or rules

Note 1 to entry: In this context: the process of compelling observance of a toll regime.

3.8
interoperability
ability of systems to exchange information and to make mutual use of the information that has been exchanged

Note 1 to entry: 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.

[SOURCE: ISO/IEC TR 10000-1:1998, 3.2.1, modified.]

3.9
on-board equipment
OBE
all required equipment on-board a vehicle for performing required *EFC* (3.6) functions and communication services

3.10
payment claim
recurring statement referring to concluded *billing details* (3.1) made available to the payer by the payee indicating and justifying the amount due

Note 1 to entry: 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 refers to *billing details* (3.1) 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.

3.11
roadside equipment
RSE
equipment located along the road, ether fixed or mobile

3.12
tariff scheme
set of rules to determine the fee due for a vehicle within a *toll domain* (3.17)

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

3.13
toll
charge, tax or duty levied in relation with using a vehicle in a *toll domain* (3.17)

Note 1 to entry: 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.

3.14
toll charger
TC
entity which levies *toll* (3.13) for the use of vehicles in a *toll domain* (3.17)

Note 1 to entry: In other documents the terms "operator" or "toll operator" can be used.

[SOURCE: ISO 17573:2010, 3.16, modified — "legal" has been deleted from before "entity" and "the use of" has been added.]

3.15
toll context data
information defined by the responsible *Toll Charger* (3.14) necessary to establish the *toll* (3.13) due for using a vehicle on a particular toll context and to conclude the toll transaction

[SOURCE: ISO 12855:2015, 3.15]

3.16
toll declaration
statement to declare the usage of a given *toll* (3.13) service to a *Toll Charger* (3.14)

Note 1 to entry: A valid toll declaration has to fulfil formal requirements, including security requirements, agreed between the *Toll Service Provider* (3.19) and the *Toll Charger* (3.14).

[SOURCE: ISO/TS 19299:2015, 3.44]

3.17
toll domain
area or a part of a road network where a certain *toll regime* (3.18) is applied

[SOURCE: ISO 17573:2010, 3.18, modified — "certain" has been added.]

3.18
toll regime
set of rules, including *enforcement* (3.7) rules, governing the collection of *toll* (3.13) in a *toll domain* (3.17)

[SOURCE: ISO 17573:2010, 3.20]

3.19
toll service provider
TSP
entity providing *toll* (3.13) services in one or more *toll domains* (3.17)

Note 1 to entry: In other documents the terms "issuer" or "contract issuer" can be used.

Note 2 to entry: The *Toll Service Provider* (3.19) can provide the *on-board equipment* (3.9) or can provide only a magnetic card or a smart card to be used with *on-board equipment* (3.9) provided by a third party.

Note 3 to entry: The *Toll Service Provider* (3.19) is responsible for the operation (functioning) of the *on-board equipment* (3.9).

[SOURCE: ISO 17573:2010, 3.23, modified — the definition has been condensed.]

3.20
trust object
information object that is exchanged between entities to ensure mutual trust

EXAMPLE An electronic signature or an electronic certificate.

[SOURCE: ISO 17573:2010, 3.28]

3.21
white list
list of users for which the service provider accepts contractual responsibility

Note 1 to entry: The service provider in the context of this International Standard is the *Toll Service Provider* (3.19).

Note 2 to entry: An entry on a white list is independent of entries on a *black list* (3.2).

4 Symbols and abbreviated terms

ADU	Application data unit (ISO 14906)
ANPR	Automatic Number Plate Reading
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit (ISO 14906)
CCC	Compliance Check Communication (ISO 12813)
CRL	Certificate revocation list
DSRC	Dedicated short-range communication (ISO 14906)
DTLS	Datagram Transport Layer Security
EFC	Electronic Fee Collection (ISO 17573)
GNSS	Global Navigation Satellite System
HTTPS	Hyper-Text Transfer Protocol Secure
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)
QA	Quality Assurance
RSA	Rivest, Shamir and Adleman (ISO/TS 19299)
RSE	Roadside Equipment (ISO 14906)
SLA	Service Level Agreement
SU	Service User
SUT	System Under Test (ISO/TS 14907-1)
TC	Toll Charger
TLS	Transport Layer Security
TSP	Toll Service Provider
VRM	Vehicle Registration Mark

NOTE RSA is an algorithm for public-key cryptography, also referred to as asymmetrical cryptographic technique.

5 Architectural concepts and information exchanges

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.

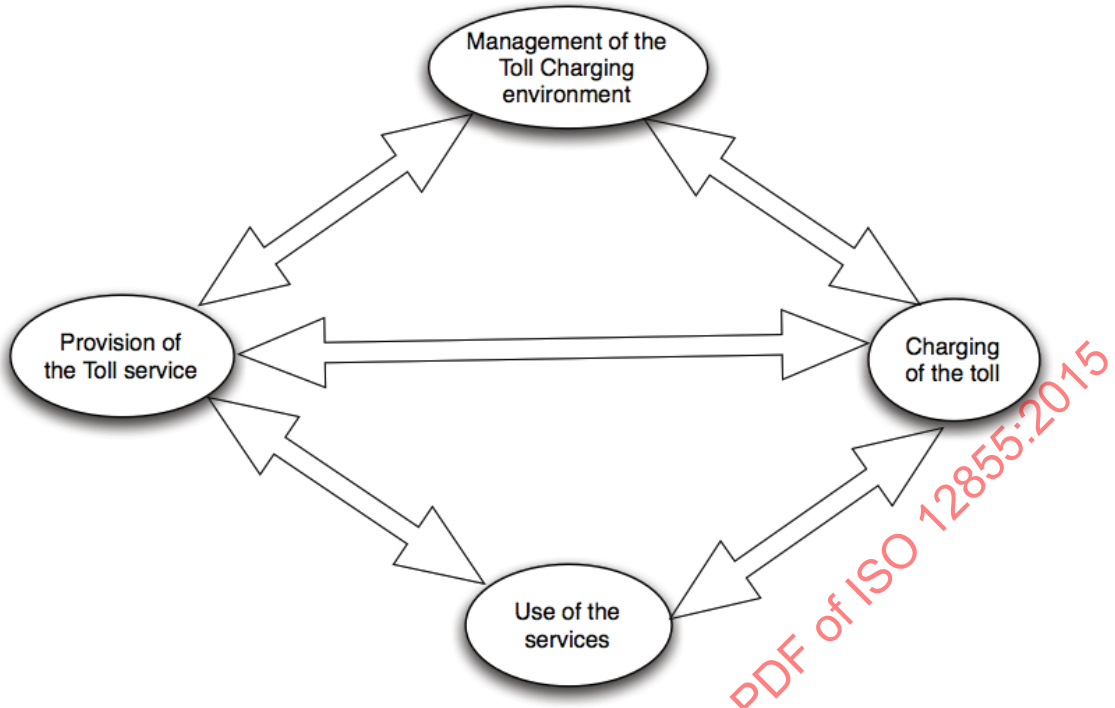

Figure 3 — Roles in the toll charging environment

Information exchanges are agreed upon between Toll Charger and Service Provider, 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.

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 functionalities that are based on the EFC system behaviour definitions in ISO 17573.

These functionalities are listed below, in the order they are given in Clauses 5 and 6:

— basic protocol mechanisms;

— exchange trust objects;

— exchange of EFC context data and contractual information;

— manage exception list;

— report toll declarations;

— report billing details;

— payment settlement:

  — claim payment for service usage;

  — announce payments;

— exchange enforcement data (includes user details);

— exchange quality assurance parameters.

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

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

— basic protocol mechanisms, to be used to build more complex protocol procedures, and

— the semantics and the format of the APDUs that are exchanged.

These functionalities are described in 5.2.2 to 5.2.11.

5.2.2 Basic interaction protocol

Information exchanges are performed by means of 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 define and perform 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 APDUs 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" ADU,

— a generic acknowledge mechanism (i.e. not related to any specific functionality) that allows acknowledging a specific interaction. The "ack" ADU provides this mechanism, and

— an optional 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" ADU.

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 acknowledgements. To handle any lost APDUs a timeout system can be implemented.

5.2.3.2 Identification schema

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

5.2.3.3 Request functionality

The "request" functionality shall 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 ADU, by indicating the type of ADU one is ready to accept,

— request the counterpart re-issue a specific APDU, by indicating the type and the identifier of the APDU, and

— request information identified by type and ADU content.

5.2.3.4 Acknowledgement functionality

The "acknowledgement" functionality shall be used to inform the counterpart that a specific ADU has been received correctly, or errors have been detected, or it is not supported.

5.2.3.5 Status functionality

The "status" functionality can be used to provide the counterpart with general status information on the interface or inform it about a 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, and

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

5.2.4 Exchange trust objects functionality

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 and certificate revocation lists.

Either the Toll Service Provider or Toll Charger can use the exchange trust objects functionality.

The initiator of the exchange can request the receiver to send trust objects by means of a "request" ADU. The "request" ADU may contain an optional indicator that specifies already issued trust objects to be resent. When no indicator is specified, the trust objects to be transferred shall be the current ones.

After receiving a "request" ADU that asks for trust objects, the requested trust objects are newly generated (or retrieved) by the recipient of the request and sent to the requesting party who shall acknowledge the reception by issuing an "ack" ADU.

If either party decides at any time to update any own trust objects, it can deliver them to any counterpart without waiting for a "request" ADU.

The receiver of trust objects shall acknowledge the reception by issuing an "ack" ADU.

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 shall start from the time indicated in the trust objects.

5.2.5 Exchange of EFC context data and contractual information functionality

5.2.5.1 Originating and providing EFC context data functionality

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 "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" ADU.

Any Toll Service Provider may request from any Toll Charger at any time and 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" ADU.

Reception of an "EFC context data" ADU shall be acknowledged by means of an "acknowledge" ADU.

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. EFC Context Data elements are defined in detail in ISO 17575-3, and are extended in this International Standard for DSRC-based and other tolling systems.

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 17575-3 and are addressed within this International Standard as Toll Context interrelations.

5.2.5.2 Contractual information functionality

The contractual information functionality allows a Toll Service Provider to deliver to the Toll Charger any type of user contract related information that is stored in the OBEs, among which the personal account number (PAN), the supported security level and the key set to be used in the calculation of DSRC authenticators. This functionality allows the Toll Charger to compare data that is stored in the OBEs with information received by the Toll Service Provider.

5.2.6 Manage exception list functionality

5.2.6.1 General

The "manage exception list" functionality originates from ISO 17573 and is defined in the EFC system behaviours "collecting toll information – user billing" and "collecting charging information (autonomous systems)".

NOTE 1 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 can use different terms, but they are all included in the term "exception list".

NOTE 2 The conditions and the periods of time that an OBE can be accepted 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, licence plate) can be included in the exception list as agreed between TC and TSP.

5.2.6.2 Exception list entry requested by a Toll Charger

A Toll Charger may use the "manage exception list" functionality 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 can issue a "report abnormal OBE" ADU to request the inclusion of this OBE in the exception list.

The Toll Service Provider shall acknowledge the inclusion of the OBE on the exception list by means of an ack ADU.

NOTE 2 This can 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" ADU. 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 ADU can include, among others, one of the following reasons:

– the 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.

NOTE 3 The exception list can be used to provide information on the country of registration of a vehicle and a VAT ID number to allow a Toll Charger the correct application of VAT for ferry operations, where this information is needed for handling of reverse charge VAT in the EU.

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 ack ADU 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 ack ADU.

5.2.7 Report toll declarations functionality

The "report toll declarations" functionality originates from ISO 17573 and is 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 17575-1 ISO 17575-2 and ISO 17575-3 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 to verify the calculation done by the Toll Service Provider. The Toll Charger in the EFC context data defines details about configuration parameters for toll declarations.

A toll declaration shall be reported to the Toll Charger operating a toll domain, by means of a "toll declaration" ADU.

NOTE 2 The toll declarations can 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" ADU by indicating for each toll declaration whether it is accepted or not by means of an "ack" ADU. If a toll declaration is not accepted, this has to be handled in a dispute phase, which is 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. Compliance Check Communication (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 "request" ADU 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 and 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 can ask the Toll Service Provider for detailed toll declarations for this specific vehicle or Service User.

5.2.8 Report billing details functionality

Depending on its toll system a Toll Charger either acquires the toll declarations directly from the roadside equipment (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 consist 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 …), or

— 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 is included in the OBE, while the rest is held centrally at the issuing Toll Service Provider.

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. To avoid any rounding differences between the VAT declarations for the local tax authorities of the Toll Charger and the VAT shown on the invoices to the Service Users (SUs), the Toll Charger and the Toll Service Provider shall apply the same aggregation methods for the exchanged and acknowledged aggregated billing details, which need to be bilaterally agreed. Consequently, rounding rules are not defined in this International Standard.

NOTE 2 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 method, the Toll Service Provider is always able to check the consistency of the payment claim with the billing details from which they stem;

– 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.

See informative Annex E for an example of one possible solution to deal with the aggregation of billing details.

The generated billing details shall be reported by means of a "billing detail" ADU. In a DSRC environment, this ADU is always generated by the Toll Charger, as it collects the toll declarations directly from its RSE. In a GNSS environment, this ADU may be either generated by the Toll Charger or the Toll Service Provider depending on the operational model of the Toll Charger.

NOTE 3 The operational model of the Toll Charger in a GNSS environment could require it to generate any billing details and keep complete control over the usage of its toll domain. This is typically the case for privately operated Toll Chargers or road concessionaires. If a government body conducts the role of Toll Charger, on the other hand, it may only want to do spot checks to avoid the operational workload of being a Toll Charger in a GNSS environment. In this case, the Toll Charger may require the Toll Service Provider to generate the billing details ADU to be sent to it.

The recipient 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 and thus provide information to the Toll Service Provider to check their authenticity.

The recipient shall acknowledge the received "billing detail" ADU by indicating for each billing detail whether it is accepted or not by means of an "ack" ADU.

5.2.9 Payment settlement functionality

5.2.9.1 Claim payment for service usage

The claim payment for service usage functionality is derived from ISO 17573 and is 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 a 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 for a whole month define the discount level of a scaled discount scheme).

Any additional claims for other services that do not directly derive from the bilaterally agreed billing details (e.g. overdue 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 can be used by the Toll Service Provider to generate invoices for its customers (Service Users) on behalf of the relevant Toll Charger or for its own account (if the Toll Service Provider is buying the toll from the Toll Charger and reselling it to the Service User).

The payment claim shall be put forward to the Toll Service Provider by means of a "payment claim" ADU.

The Toll Service Provider shall acknowledge the received "payment claim" ADU by indicating for each payment claim whether it is accepted or not by means of an "ack" ADU.

NOTE 2 Possible reasons for dispute, among others, are: inconsistency of payment claim with billing details or non-conformance of specific commercial conditions.

5.2.9.2 Payment announcement

The payment announcement for service usage functionality orientates itself on the payment claim functionality but is initiated by the Toll Service Provider or on request from the Toll Charger.

After billing details have been agreed between the Toll Charger and the Toll Service Provider, the Toll Service Provider sends a payment announcement to the Toll Charger informing it about an upcoming

payment. This payment announcement may include references to billing details, toll declarations or even single elements of a charge report.

The Toll Charger shall acknowledge the received "payment announcement" ADU by indicating for each payment announcement whether it is accepted or not by means of an "ack" ADU.

5.2.10 Exchange enforcement data functionality

5.2.10.1 General

The "exchange enforcement data" functionality originates from ISO 17573 and is defined in the EFC system behaviour "exceptions detection – user and OBE compliance checking".

The "exchange enforcement data" functionality may be used each time 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, and

— 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 Service Users if the Service User was responsible for the wrong toll declaration, and

— 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 functionality

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

This ADU 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 "request" ADU requiring user details, the Toll Service Provider may answer with a "provide user details" ADU, thus confirming a contractual relationship, or with an "ack" ADU with a specific reason code to decline any contractual relationship.

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 14906:2011/Amd1:2015);

— contract serial number;

— licence plate number;

— on-board equipment identification (OBE ID).

The Toll Service Provider shall provide any additional details on the Service User as specified in the "request" ADU to the best of its ability.

NOTE The delivery of user details can be subject to local data protection regulations.

The Toll Charger shall acknowledge any received “provide user details” ADU by means of an “ack” ADU.

5.2.10.4 Retrieve and report CCC event

The Toll Service Provider may send a “request” ADU 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.

In this case, the Toll Charger shall answer to the received “request” ADU with a “report CCC event” ADU or it may send the “report CCC event” ADU of its own accord to inform the Toll Service Provider about any new CCC event.

The Toll Service Provider shall acknowledge any received “report CCC event” ADU by means of an “ack” ADU.

5.2.10.5 List of users

The list of users functionality provides support to the Toll Charger in its enforcement activities and is used to detect potential fraud in particular.

The Toll Charger can specify in a “request” ADU to the Toll Service Provider a specific userId (e.g. a licence plate number) and can request information on all other userIds (e.g. licence plate numbers) that belong to the same Service User (e.g. haulier).

5.2.11 Exchange quality assurance parameters functionality

The report quality assurance parameters functionality is derived from ISO 17573 and is 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 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). As it is important for both the Toll Charger and the Toll Service Provider that the detection rate 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” ADU;

— the Toll Service Provider may request an update of a quality assurance parameter by sending a “request” ADU. The Toll Charger shall respond to the request by sending a “report QA” ADU;

— the Toll Service Provider may send a quality assurance parameter by issuing a “report QA” ADU;

— the Toll Charger may request an update of a quality assurance parameter by sending a “request” ADU. The Toll Service Provider shall respond to the request by sending a “report QA” ADU.

The receiver of a “REPORT QA” ADU shall acknowledge it by means of an “ack” ADU.

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, and

— the Application Data Units (ADUs) 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. The fact that computational objects are modelled here as each having one logical interface for interactions does not put any limits on the number of real interfaces offered in an implementation of this International Standard.

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

An implementation conforming to this International Standard shall declare which of the computational objects listed in Table 1 are supported, by completing the proforma protocol implementation conformance statement as defined in Annex B. For each computational object the implementer declares to support, and depending on the implementation's role (Toll Charger or Toll Service Provider), the rules stated in Table 1 shall be implemented, with the following meanings:

— may initiate: an entity is able and allowed to initiate an interaction;

— shall initiate: an entity is able and has to initiate an interaction;

— shall be able to receive: an entity shall be able to accept this kind of interaction. This means that responses to the related ADUs cannot bear an "ADU unknown" reason code.

Table 1 — Computational objects and interactions

Computational object	Interaction	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

Table 1 *(continued)*

Computational object	Interaction	Toll Service Provider rules	Toll Charger rules
Originating and providing EFC context data and contractual information	Request (EFC context data)	May initiate	Shall be able to receive
	Request (contractual information)	Shall be able to receive	May initiate
	EFC context data	Shall be able to receive	May initiate
	Contractual information	May initiate	Shall be able to receive
	Acknowledge (EFC context data)	Shall initiate	Shall be able to receive
	Acknowledge (contractual information)	Shall be able to receive	Shall initiate
	Status	May initiate, shall be able to receive	May initiate, shall be able to receive
Manage exception list	Request (exception list)	Shall be able to receive	May initiate
	Request (abnormal OBE)	May initiate	Shall be able to receive
	Exception list	May initiate	Shall be able to receive
	Report abnormal OBE	Shall be able to receive	May initiate
	Acknowledge (exception list)	Shall be able to receive	Shall 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	Request	Shall be able to receive	May initiate
	Toll declaration	May initiate	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
Report billing details	Request	May initiate, shall be able to receive	May initiate, shall be able to receive
	Billing details	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
Payment settlement	Request (payment claim)	May initiate	Shall be able to receive
	Request (payment announcement)	Shall be able to receive	May initiate
	Payment claim	Shall be able to receive	May initiate
	Payment announcement	May initiate	Shall be able to receive
	Acknowledge (payment claim)	Shall initiate	Shall be able to receive
	Acknowledge (payment announcement)	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	Interaction	Toll Service Provider rules	Toll Charger rules
Exchange enforcement data	Request (user details)	Shall be able to receive	May initiate
	Request (CCC event)	May initiate	Shall be able to receive
	Request (list of users)	Shall be able to receive	May initiate
	Provide user details	May initiate	Shall be able to receive
	Report CCC event	Shall be able to receive	May initiate
	List of Users	May initiate	Shall be able to receive
	Acknowledge (user details)	Shall be able to receive	Shall initiate
	Acknowledge (report CCC event)	Shall initiate	Shall be able to receive
	Acknowledge (list of users)	Shall be able to receive	Shall initiate
	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 interactions summarized in Table 1 are performed by exchanging OSI application layer APDUs. They are described in detail in the following sections and formally defined in Annex A. In case of any conflict between this description and the definition in Annex A, the definition in Annex A takes precedence.

6.2 Application Protocol Data Units

6.2.1 General

The APDU content is formally defined in Annex A by the ASN.1 type InfoExchange, which is made of the following elements:

1) a mandatory **infoExchangeContent** field;

2) an optional **infoExchangeAuthenticator** field.

Table 2 describes the fields in the infoExchange APDU.

Table 2 — InfoExchange APDU fields

Field name	Data type/Data description	m/o
infoExchangeContent	Actual content of the APDU For types and semantics, see further on in Clause 6	m
infoExchangeAuthenticator	Record (ASN.1 SEQUENCE) made of two fields: authenticatorEfc, and ackAuthenticatorEfc	o

Each infoExchangeContent field shall contain one Application Protocol Control Information (APCI) field and one ADUs field as specified in the InfoExchangeContent data type (see Table 3).

Any APDU (InfoExchange) may optionally be signed by calculating an **infoExchangeAuthenticator** over the **infoExchangeContents** field. The calculated authenticator shall be included as **infoExchangeAuthenticator** in the resulting APDU. Semantics of the two fields of the **infoExchangeAuthenticator**, as well as the algorithms used to calculate them, are specified in 7.3.

The **infoExchangeAuthenticator** is a dynamic object and shall be generated each time a new APDU is issued. The infoExchangeAuthenticator can be calculated by either the APDU originator or the information sender (see 6.2.2).

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/Data description	m/o
apci	Record (ASN.1 SEQUENCE) containing Protocol Control Information For a detailed description see 6.2.2	m
adus	Contains one or more Application Data Units of the same type, as a choice among requestAdus, ackAdus, trustObjectAdus, efcContextDataADUs, exceptionListADUs, reportAbnormalOBEADUs, tollDeclarationADUs, billingDetailsADU, paymentClaimADUs, reportQAADUs, statusADUs, reportCCCEventADUs, provideUserIdListADUs, paymentAnnouncementADUs and contractIssuerListADUs For detailed descriptions see Clause 6.3 to Clause 6.18	m

A top level view of the structure of InfoExchangeContent data type and its fields is given in Figure 4.

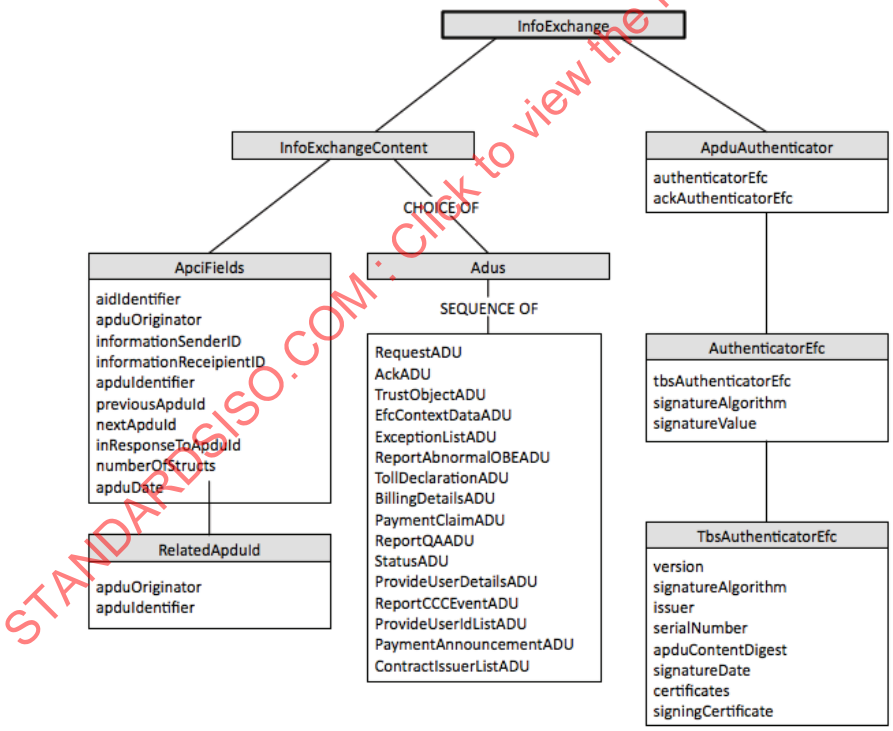

Figure 4 — InfoExchange data type structure

The clauses that follow describe the APCI and the ADUs for the defined APDUs.

APDUs are described in the following terms:

1. tables that list the parameters that make up the APDUs. For each table, the parameter names are listed, together with an indication of their fields and formats and whether the parameter is

mandatory (m) or optional (o). The parameter description is indicative and is formally specified in Annex A. In cases where there are discrepancies, the formal definition in Annex A takes precedence;

2. additional semantics definition, and, given as text after each table, usage of each parameter.

6.2.2 Application Protocol Control Information

The APCI for all APDUs shall consist of the fields described in Table 4.

Table 4 — APCI fields

Field name	Data type/Data description	m/o
aidIdentifier	INTEGER on two octets identifying the protocol version	m
apduOriginator	Entity identifier (ASN.1 Provider type) of the APDU contents generator	m
informationSenderID	Entity identifier (ASN.1 Provider type) of the APDU sender	m
informationRecipientID	Entity identifier (ASN.1 Provider type) of the APDU receiver	m
apduIdentifier	INTEGER that identifies this APDU	m
previousApduId	INTEGER that identifies a different APDU	o
nextApduId	INTEGER that identifies a different APDU	o
inResponseToApduId	INTEGER that identifies a different APDU	o
apduDate	Time stamp formatted as ASN.1 GeneralizedTime type	m

aidIdentifier shall identify the Protocol Version number. **aidIdentifier** = 0 shall be used to indicate this International Standard version.

apduOriginator shall be the identifier of the entity originating the contents of the APDU, and is an entity of type Toll Charger or Service Provider. That may or may not be the Information Sender.

informationSenderID shall identify the entity actually sending the APDU. It may be used to identify a technical entity supporting the apduOriginator in the task of transferring information. It may contain the same identifier as the **apduOriginator** field, when the originator and the sender are the same entity.

informationRecipientID shall be the identifier of the addressed entity responsible for processing the content of the APDU and is an entity of type Toll Charger or Service Provider. In some APDUs a null value is admitted for this field, with the meaning that the ADU(s) herein contained are intended for all possible recipients of the APDU.

apduIdentifier shall be a unique identifier for each given apduOriginator that is assigned by the apduOriginator. It is used to either acknowledge or request retransmission of a previously issued APDU.

previousApduId can optionally indicate that the APDU is related to a previously sent APDU from the apduOriginator. The field contains the identification of the APDU in terms of apduOriginator and apduIdentifier. This field, together with the nextApduId field, can be used to identify chains of APDUs.

nextApduId can optionally indicate that the APDU is related to a next to be sent APDU from the apduOriginator. The field contains the identification of the APDU in terms of apduOriginator and apduIdentifier. This field, together with the previousApduId field, can be used to identify chains of APDUs.

inResponseToApduId can optionally indicate that the APDU is either a response to a request or an ACK. It may be needed to identify a chain of APDUs to be processed in a workflow. The field contains the identification of the APDU in terms of apduOriginator and apduIdentifier.

apduDate shall specify a time stamp for the APDU.

6.2.3 Application Data Units

The Application Data Units (ADUs field) shall contain one or more data structures of the same type.

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

Table 5 — ADUs overview

Functionality	ADU type name	Description
Basic	RequestADU	Generic request
	AckADU	Generic acknowledge
	StatusADU	Generic status
Exchange trust objects	TrustObjectADU	Send trust objects
Exchange EFC context data and contractual information	EfcContextDataADU	Send EFC context data
	ContractIssuerListADU	Provide contractual information on users
Manage exception lists	ExceptionListADU	Send exception list
	ReportAbnormalOBEADU	Report abnormal OBE
Report toll declarations	TollDeclarationADU	Toll declaration
Report billing details	BillingDetailsADU	Billing details
Payment settlement	PaymentClaimADU	Payment claim
	PaymentAnnouncementADU	Payment announcement
Exchange enforcement data	ProvideUserDetailsADU	Provide user details
	ReportCCCEventADU	Report CCC event
	ProvideUserIdListADU	Provide a user list
Exchange quality assurance parameters	ReportQAADU	Report QA

These data structures are specified in 6.3 to 6.18.

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 ADU the general readiness to receive, or the specific request to receive a specific data structure.

The RequestADU type is a choice (ASN.1 CHOICE) among different substructures, each one specifying a different type of request. Table 6 describes the fields defined as CHOICE in a RequestADU data structure, which is formally defined in Annex A.

Table 6 — RequestADU fields

Field name	Data type/Data description	m/o
genericRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType, apduIdentifier, numberOfADUStruct (see description below Table 6)	N/A
exceptionListRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType(exceptionListRequest), period, exceptionListType (see description below Table 6)	N/A
trustObjectRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType(trustObject), requestedTrustObject (see description below Table 6)	N/A
tollDeclarationRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType(TollDeclaration), userId, startTime, endTime (see description below Table 6)	N/A

Table 6 *(continued)*

Field name	Data type/Data description	m/o
userDetailsRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType(userDetails), userId, listOfParametersRequested, userDetailsRequestReason, userInfoValidityPeriod (see description below Table 6)	N/A
cccEventRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType(reportCCCEventADU), userId, startTime, endTime (see description below Table 6)	N/A
userListRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedADUType(provideUserIdListADU), userIdRequestType, userId, userIdRequestTime (see description below Table 6)	N/A

If the **genericRequest** is selected, the following rules apply:

1. all fields are optional. If no fields are specified, the RequestADU indicates a general readiness to receive;
2. if the **requestedADUType** field is specified, an ADU of the indicated data type is expected. It is used to indicate readiness to receive ADUs of the indicated type;
3. if the **requestedADUType** field is specified, the **apduIdentifier** may be specified;
4. if the **apduIdentifier** is specified, an ADU of the given type, which was previously transmitted in a APDU identified by the **apduIdentifier** field, is requested to be re-transmitted. If no **apduIdentifier** is specified, this ADU may be transferred to all possible recipients by specifying a null value in the **informationRecipientID** field of the APCI;
5. if the apduIdentifier 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 *n*th instance.

If the **exceptionListRequest** is selected, the following rules apply:

1. **requestedADUType** field shall bear the value “exceptionListADU”;
2. **period** field can optionally request to retransmit all ExceptionList ADUs that are related to the given period of time;
3. **exceptionListType** field can optionally request to retransmit all ExceptionList ADUs that are of the specified type.

If the **trustObjectRequest** is selected, the following rules apply:

1. **requestedADUType** field shall bear the value “trustObjectADU”;
2. **requestedTrustObject** field shall either specify a trust object identifier, when an already issued trust object is requested to be resent, or indicate the type and, optionally, the purpose, of the required trust object. In case no specific trust object Type is to be requested, then a **genericRequest** shall be used, with the **requestedADUType** field bearing the value “**trustObjectADU**”.

If the **tollDeclarationRequest** is selected, the following rules apply:

1. **requestedADUType** field shall bear the value “tollDeclarationADU”;
2. **userId** field can optionally 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;
3. **startTime** field can optionally 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;

4. **endTime** field can optionally 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 ADU was generated;

5. if none of the optional fields is specified, then a **genericRequest** shall be used, with the **requestedADUType** field bearing the value "tollDeclarationADU".

If the **userDetailsRequest** is selected, the following rules apply:

1. **requestedADUType** field shall bear the value "provideUserDetailsADU";

2. **userId** field shall specify the UserId for which the additional details are requested;

3. **listOfParametersRequested** can optionally be used to specify which additional parameters, if any, are required. If this field is not specified then the response shall contain all available details;

4. **userDetailsRequestReason** can optionally be used to specify the reason for the request;

5. **userInfoValidityPeriod** can optionally be used to request user information related to the specified period in time, including cases where user information is varied during the period. If not specified, the currently available user information is meant to be requested.

If the **cccEventRequest** is selected, the following rules apply:

1. **requestedADUType** field shall bear the value "reportCCCEventADU";

2. **userId** field can optionally 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;

3. **startTime** field can optionally 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;

4. **endTime** field can optionally 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 ADU was generated.

If the **userListRequest** is selected, the Toll Charger is meant to request a Toll Service Provider to provide information on vehicles that belong to the same Service User. The following rules apply:

1. **requestedADUType** field shall bear the value "provideUserIdListADU";

2. **userIdRequestType** field shall provide the possibility to indicate the type of request. Currently, only one value is defined: allUserIdsToGivenCustomer;

3. **userId** field can optionally provide the identification of a vehicle or OBE that is known to the Toll Charger and where further vehicles or OBEs are requested that belong to the same Service User. If the field is not specified, the request is to retrieve details of all that TSP's users as filtered by userIdRequestType;

4. **userIdRequestTime** field can optionally provide the option to define a point of time. If this attribute is provided, the Toll Service Provider shall answer the request based on data valid at the provided point of time and not at the time of the request.

6.4 AckADU data structure

The ackADU 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 APDU that transported the data structure. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 7 describes the fields in an AckADU data structure, which is formally defined in Annex A.

Table 7 — AckADU fields

Field name	Data type/Data description	m/o
apduIdentifier	INTEGER that identifies an APDU	m
explicitlyAckedAdus	Record made of fields of the same type (ASN.1 SEQUENCE OF), each one of type INTEGER, identifying an ADU in the acknowledged APDU	o
apduAckCode	INTEGER representing the ApduReasonCode	m
apduAckText	Additional textual information expressed as ASN.1 UTF8String type	o
issues	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record made of the following fields: issueADUStruct, issueLocation, issueContent, issueCode, issueText (see the description below)	o

apduIdentifier shall indicate the identifier of the APDU containing the data structure(s) being acknowledged.

explicitlyAckedAdus can optionally indicate the ADUs in the APDU being explicitly acknowledged. If explicitlyAckedAdus is not present, then the whole APDU is acknowledged together with all ADUs contained in it.

apduAckCode shall contain the status of the APDU or a reason for accepting or rejecting the APDU. The apduAckCode accepts or rejects an APDU as a whole. If an APDU is accepted, there can be additional responses on individual elements of the APDU (ADUs). A single ADU could be rejected while the APDU itself is accepted. This is dealt with by using the optional **issues** field (see below). If the **apduAckCode** indicates rejection, the whole APDU is rejected, irrespective of the possible acceptance codes indicated in the **issues** field.

apduAckText can optionally add a text to further explain the reason for accepting or rejecting the APDU.

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 Ack ADU identifying the APDU that contains the BillingDetails data structures previously received, indicating for each of the rejected ADUs the reason for rejection. If all billing details are to be accepted, the acceptance of the APDU is sufficient.

issues can optionally provide detailed error information that responds to multiple instances of ADUs in the APDU being acknowledged. This allows the system to accept an APDU as a whole, but to possibly individually reject single ADUs. The only mandatory subfield in this structure is issueCode, which contains the reason code (see Table 8). The subfields issueLocation, issueContent and issueText are not further defined and can optionally be used based on bilateral negotiations.

Values of the AduReasonCode are reserved for CEN and ISO use. The values that are defined in this International Standard are listed (in decimal notation) in Table 8 and are formally defined in Annex A.

Table 8 — AduReasonCode values

Name	Meaning	Value
invalidADU	The specified ADU is invalid	0
notSupportedADU	The specified ADU is not supported	1
reserved	Codes reserved for trust object exchange errors	2..99
trustObjectExpired	The exchanged trust object is not more valid	100
trustObjectUnreadable	The received trust object is not readable	101

Table 8 *(continued)*

Name	Meaning	Value
reserved	Codes reserved for trust object exchange errors	102..199
contextDataChargeObjectIdRejected	The charge object Id in context data has been rejected	200
contextDataChargeObjectDescriptionRejected	The charge object description has been rejected	201
contextDataSystemTypeRejected	The system type in context data has been rejected	202
contextDataChargeObjectTypeRejected	The type of charge object in context data has been rejected	202
contextDataTollChargerRejected	The toll charger in context data has been rejected	204
contextDataChargeObjectValidityStartRejected	The start of validity of the charge object in context data has been rejected	205
contextDataChargeObjectValidityEndRejected	The end of validity of the charge object in context data has been rejected	206
reserved	Codes reserved for context data exchange errors	207..299
reserved	Codes reserved for contract issuer exchange errors	300..399
exceptionListsVersionRejected	The exception list version has been rejected	400
exceptionListsTypeRejected	The type of exception list has been rejected	401
exceptionListsUserIdRejected	The user ID in exception list has been rejected	402
exceptionListsBlockTypeRejected	The type of block in exception list has been rejected	403
exceptionListsReasonCodeRejected	The reason code in exception list has been rejected	404
exceptionListsDateAndTimeRejected	The date and time in exception list has been rejected	405
reserved	Codes reserved for exception lists exchange errors	406..499
abnormalObeUserIdRejected	The user id in abnormal OBE has been rejected	500
abnormalObeDateAndTimeRejected	The date and time in exception list has been rejected	501
abnormalObeReasonCodeRejected	The reason code in abnormal OBE has been rejected	502
reserved	Codes reserved for abnormal OBE exchange errors	503..599
reserved	Codes reserved for toll declarations exchange errors	603..699
billingDetailsIssuerIdRejected	The issuer in billing details has been rejected	700
billingDetailsClaimIdRejected	The claim id in billing details has been rejected	701
billingDetailsTollChargerIdRejected	The Toll Charger id in billing details has been rejected	702

Table 8 *(continued)*

Name	Meaning	Value
billingDetailsContextIdRejected	The context id in billing details has been rejected	703
billingDetailsUserIdRejected	The user id in billing details has been rejected	704
billingDetailsPeriodRejected	The period in billing details has been rejected	705
billingDetailsAmountRejected	The amount in billing details has been rejected	706
billingDetailsContextNameRejected	The context name in billing details has been rejected	707
billingDetailsAppliedUserClassRejected	The applied user class in billing details has been rejected	708
billingDetailsDeclaredVehicleClassRejected	The declared vehicle class in billing details has been rejected	709
billingDetailsAppliedTimeClassRejected	The applied time class in billing details has been rejected	710
billingDetailsEntranceTimeRejected	The entrance time in billing details has been rejected	711
billingDetailsEntranceChargeObjectRejected	The entrance charge object in billing details has been rejected	712
billingDetailsIntermediateSectionRejected	The intermediate section in billing details has been rejected	713
billingDetailsExitChargeObjectRejected	The exit charge object in billing details has been rejected	714
billingDetailsExitTimeRejected	The exit time in billing details has been rejected	715
reserved	Codes reserved for billing details exchange errors	716..799
claimRejectedByTSP	The related payment claim has been rejected by the TSP	800
claimApprovedByTSP	The related payment claim has been approved by the TSP	801
paymentClaimIdRejected	The id in payment claim has been rejected	802
paymentStartDateTimeRejected	The start date and time in payment claim has been rejected	803
paymentEndDateTimeRejected	The end date and time in payment claim has been rejected	804
paymentUserIdRejected	The user id in payment claim has been rejected	805
paymentAmountRejected	The amount in payment claim has been rejected	806
paymentStatusRejected	The status in payment claim has been rejected	807
paymentTypeOfFeeRejected	The type of fee in payment claim has been rejected	808
paymentRelatedApduIdRejected	The related APDU id in payment claim has been rejected	809

Table 8 *(continued)*

Name	Meaning	Value
reserved	Codes reserved for payment claim exchange errors	810..899
reserved	Codes reserved for payment announcement exchange errors	900..999
reserved	Codes reserved for provide user details exchange errors	1000..1099
reserved	Codes reserved for provide user lists exchange errors	1100..1199
reportQAAccepted	The quality assurance parameters have been accepted	1200
reportQANotAccepted	The quality assurance parameters have been rejected	1201
paymentGuaranteeAccepted	The payment guarantee has been accepted	1202
smCCBackEndDataCheckingThresholdExceeded	Secure monitoring reason code	1203
reserved	Codes reserved for quality assurance exchange errors	1204..1299
reserved	Codes reserved for future ISO/CEN extensions	1300..10000
reserved	Codes reserved for private extensions defined in bilateral agreements between TCs and TSPs	10000 ..

The following rules apply for handling the reception of an AckADU:

1. if the apduAckCode field indicates rejection (value of aduNotOK), the whole referred APDU is rejected. The value of the explicitlyAckedAdus parameter value is meaningless;

2. if the apduAckCode field indicates acceptance, the referred APDU is accepted (syntactically and semantically correct). If the explicitlyAckedAdus field is empty, all ADUs contained in the referred APDU are positively acknowledged;

3. If the apduAckCode indicates acceptance, but the explicitlyAckedAdus field is not empty, some of the ADUs contained in the referred APDU are not positively acknowledged. The explicitlyAckedAdus field shall list the ADUs that are not positively acknowledged by their position (order number) in the referred APDU. In this case, the issues field shall list these ADUs by their position and gives for each one a related AduReasonCode value.

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 APDUs are obsolete (e.g. to support rollback scenarios) or to indicate the inability to receive APDUs. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 9 indicates the fields of the StatusADU data structure, which is formally defined in Annex A.

Table 9 — StatusADU fields

Field name	Data type/Data description	m/o
generalStatusCode	INTEGER defined as GeneralStatusCode	m
apduStatusCode	Record (ASN.1 SEQUENCE) made of the following fields: apdu, reasonOfADUStruct	o

generalStatusCode shall indicate readiness to receive APDUs. The following values are allowed:

— 1, not ready to receive;

— 2, ready to receive.

apduStatusCode can optionally provide APDU-specific reason codes as a sequence of one or more reason codes for each by ADU in a related APDU. The **apdu** field identifies the APDU that contains the ADUs. The reasonOfADUStruct gives an AduReasonCode for each ADU within the APDU.

6.6 TrustObjectsADU data structure

The trust objects ADU can be used by either the Toll Charger or the Toll Service Provider to provide trust objects. The ADU can be either sent as a response to a Request ADU or initiated as a push ADU if new versions of trust objects are available. When transferring this ADU, the value in the informationRecipientID field of the APCI may have a null value, to indicate that the ADU is intended for all possible recipients.

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 10 indicates the fields in a TrustObjectsADU data structure, which is formally defined in Annex A.

Table 10 — TrustObjectADU fields

Field name	Data type/Data description	m/o
trustObjectID	INTEGER	m
purposesOfTrustObject	Record made of fields of the same type (ASN.1 SEQUENCE OF), each one of type INTEGER	m
startValidity	Time stamp expressed as ASN.1 GeneralizedTime	o
endValidity	Time stamp expressed as ASN.1 GeneralizedTime	o
trustObjectStatus	INTEGER defined as TrustObjectStatus	m
trustObject	TrustObjectCode	o

trustObjectID shall indicate the identifier for the trust object.

purposesOfTrustObjects shall be used to indicate the possible usage(s) the trust object is meant for. It is a list of values, to signify that each trust object may be used for multiple purposes. Values can be:

— trustObjects (0), unspecified purpose;

— dSRCCharging (1), to validate authenticators from DSRC charging applications;

— dSRCAC (2), to calculate access credentials for DSRC charging applications;

— oBEInterrogation (3), to validate authenticators received during a CCC transaction;

— oBEInterrogationAC (4), to calculate access credentials for CCC;

— sIGExceptionList (5), to authenticate received exception lists;

— sIGContextData (6), to authenticate received context data;

— sIGBillingDetails (7), to authenticate received billing details;

— sIGFiscalObjects (8), to authenticate received fiscal objects;

— sIGCommunication (9), to authenticate received ADUs by infoExchangeAuthenticator;

— eNCCommunication (10), to decrypt received ADUs;

— dSRCKeyEncryption (11), to encrypt DSRC keys;

— secChannelEstablishment (12), certificate to establish IPSec (VPN);

— certIssuing (13), certificate to issue certificates;

— sIGUserCommunication (14), user certificate to verify communication via, e.g. email;

— certRevocationListing (15), certificate revocation list;

— siGChargeReport (16), certificate or certificate chain to validate charge report.

startValidity can optionally 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.

endValidity can optionally indicate the expiry date for a trust object. If this is not completed then it shall be assumed that there is no expiry data.

trustObjectStatus shall be used to indicate the status of the specified trust object. It can take the following values:

— 0, for a valid trust object;

— 1, for an invalid trust object;

— 2, for a revoked trust object.

trustObject shall contain the definition of the trust object. Trust object types defined in this International Standard are:

1. certificates;
2. public keys;
3. DSRC keys;
4. MAC keys;
5. DSRC key references;
6. generic trust objects.

For each defined type of trust object, a different definition applies, as per the following.

The **CertificateObject** structure allows exchanging certificates between TC and TSP. The certificate structure and coding shall be according to ISO/IEC 9594-8. Table 11 explains the fields in the CertificateObject structure, which is formally defined in Annex A.

Table 11 — CertificateObject fields

Field name	Data type/Data description	m/o
certificateType	INTEGER defined as CertificateType	m
certificate	OCTET STRING containing the certificate	m

The **certificateType** field specifies the type of certificate as a one byte INTEGER. It can take the following values:

— 0, indicating a Toll Charger certificate for HTTPS usage;

— 1, indicating a Toll Charger certificate for email exchange;

— 2, indicating a Toll Charger certificate for signatures;

— 3, indicating the Toll Chargers' sub-Certification Authority certificate;

— 4, indicating the Toll Chargers' root-Certification Authority certificate;

— 5, indicating a Toll Service Provider certificate for HTTPS usage;

— 6, indicating a Toll Service Provider certificate for email exchange;

— 7, indicating a Toll Service Provider certificate for signatures;

— 8, indicating the Toll Service Provider sub-Certification Authority certificate;

— 9, indicating the Toll Service Provider root-Certification Authority certificate;

— 10, indicating the Toll Service Provider charge report verification.

The **certificate** field contains the actual public-key certificate according to ISO/IEC 9594-8.

The **PublicKeyObject** type is used to transfer public keys whose authenticity is proven by a fingerprint exchanged over another communication channel. Public keys without a certificate may be used to establish the initial trust relation between TC and TSP. Table 12 explains the fields in the PublicKeyObject structure, which is formally defined in Annex A.

Table 12 — PublicKeyObject fields

Field name	Data type/Data description	m/o
publicKeyType	INTEGER defined as PublicKeyType	m
serialNumber	INTEGER	m
issuer	Identifier of the key issuer as ASN.1 Provider type	m
algorithmIdentifier	AlgorithmIdentifier type as imported from ISO/IEC 9594-8	m
publicKey	OCTET STRING	m

The **publicKeyType** field specifies the type of key. A key can be:

— kpupTcSignature, which is a signature key of the Toll Charger;

— kpupTcEncrypt, which is an encryption key for the Toll Charger;

— kpupTspSignature, which is a signature key of the Toll Service Provider;

— kpupTspEncrypt, which is an encryption key for the Toll Service Provider.

The **serialNumber** field is an INTEGER indicating the key's serial number.

The **issuer** field specifies the issuer of the key in the format of a provider data type.

The **algorithmIdentifier** field identifies the public key algorithm, e.g. RSA, as imported from ISO/IEC 9594-8.

The **publicKey** field contains the key.

The **DSRCKeyObject** type allows the exchange of CCC and LAC DSRC master keys. In addition, the structure allows sending a DSRC master key, which is encrypted by means of a public key without a certificate. Table 13 explains the fields in the DSRCKeyObject structure, which is formally defined in Annex A.

Table 13 — DsrcKeyObject fields

Field name	Data type/Data description	m/o
encryptionKeyId	ASN.1 CHOICE type between certificate or publicKey	m
timestamp	Time stamp expressed as ASN.1 UTCTime	m
dsrcKeys	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record containing the following fields: eFCCM, key	m

The **encryptionKeyId** field identifies the encryption type used. It is a couple (serial number, issuer) identifying either a certificate or a public key.

The **timestamp** field contains the time of the encryption.

The **dsrcKeys** field identifies a series of couples (EFC Context Mark, key) to be used in the DSRC environment. Each key is a record (EncrKey data type, an ASN.1 SEQUENCE) containing the type of the key, the key reference as an integer used to identify the key, the encrypted value of the key, the key verification code and an optional description of the key.

Some data elements are possibly authenticated by a derived MAC (see ISO 17575-1). The **MacKeyObject** allows for the exchange of MAC master keys. Table 14 explains the fields in the MacKeyObject structure, which is formally defined in Annex A.

Table 14 — MacKeyObject fields

Field name	Data type/Data description	m/o
serialNumber	CertificateSerialNumber as imported from ISO/IEC 9594-8	m
issuer	Name, as imported from ISO/IEC 9594-8	m
timestamp	Time stamp, expressed as ASN.1 UTCTime type	m
algorithmIdentifier	AlgorithmIdentifier, as imported from ISO/IEC 9594-8	m
masterKeyRef	INTEGER	m
encrKey	OCTET STRING	m
kVC	OCTET STRING	m

The **serialNumber** field specifies the serial number of the certificate used.

The **issuer** field specifies the name of the certificate's receiver.

The **timestamp** field specifies the time of the encryption.

The **algorithmIdentifier** field identifies the public key algorithm, e.g. RSA.

The **masterKeyRef** field identifies the master key used for encryption.

The **encrKey** field contains the key that has encrypted using the certificate referred to by serialNumber and issuer of Key.

The **kVC** field is the key verification code according to ISO 11568-2, calculated encrypting one block size of zeros with the plain key, then truncated to leftmost three bytes.

The **DSRCKeyRef** type allows exchanging a non repudiation key reference for, e.g. CCC (ISO 12813) or LAC (ISO 13141). Table 15 explains the fields in the DSRCKeyRef structure, which is formally defined in Annex A.

Table 15 — DSRCKeyRef fields

Field name	Data type/Data description	m/o
eFCContextMark	EFCContextMark, as imported from ISO 14906	m
keyRef	INTEGER	m
referenceType	INTEGER defined as ReferenceType	m

The **eFCContextMark** field contains the EFC context mark as defined in ISO 14906.

The **keyRef** field contains the reference to the key to be used, as an INTEGER.

The **referenceType** field specifies the usage of the key as following:

0. CCC non repudiation, used for non repudiation in CCC (ISO 12813);
1. CCC authentication, used for authentication in CCC (ISO 12813);
2. LAC authentication, used for authentication in LAC (ISO 13141).

The generic trust object type allows exchanging a trust object that has not a defined structure in the current version of this International Standard. Table 16 explains the fields in the GenericTrustObject structure, which is formally defined in Annex A.

Table 16 — GenericTrustObject fields

Field name	Data type/Data description	m/o
typeOfTrustObject	INTEGER defined as ListOfTrustObjectTypes	m
purposesOfTrustObject	Record of fields of the same type (ASN.1 SEQUENCE OF), each field of type INTEGER defined as ListOfTrustObjectPurposes	m
genericTrustObject	OCTET STRING	m

The **typeOfTrustObject** field specifies the type of trust object. It can be, e.g. a certificate, a symmetric key, a certificate revocation list, or other trust objects as defined in Annex A.

The **purposesOfTrustObject** field indicates the intended usages of the trust object as a sequence of integers addressing the ListOfTrustObjectPurposes data type.

The **genericTrustObject** field contains the trust object.

6.7 EfcContextDataADU data structure

The ADU EFC context data shall be used by the Toll Charger as either a response to the request ADU sent by the Toll Service Provider or a push ADU if new versions of EFC context attributes are available. When transferring this ADU, the value in the informationRecipientID field of the APCI may have a null value, to indicate that the ADU is intended for all possible recipients.

The EfcContextDataADU contains optional Toll Charger related administrative information, defined by the EntityOverview data type, and followed by one data structure of either the gnssContext data type, dsrcContext data type or dsrcClosedContext data type.

The **EfcContextDataADU** data structure is described in Table 17 and is formally defined in Annex A.

Table 17 — EfcContextDataADU fields

Field name	Data type/Data description	m/o
entityOverview	Record (ASN.1 SEQUENCE) of fields describing the Toll Context, see Table 18	o
domainType	Choice (ASN.1 CHOICE) of one of the following: gnssContext (to specify a Toll Context based on autonomous tolling), dsrcContext (to specify a DSRC open Toll Context), dsrcClosedContext (to specify a DSRC closed context) See detailed description further on	m

The EntityOverview data type allows optionally to provide Toll Charger administrative information. It is described in Table 18 and is formally defined in Annex A.

Table 18 — EntityOverview fields

Field name	Data type/Data description	m/o
entityId	Entity identifier defined as ASN.1 type Provider, as inherited from ISO 14906	m
entityType	INTEGER defined as EntityType	m
relatedEntityID	Record made of fields of the same type (ASN.1 SEQUENCE OF), each one of type EntityId	o
entityClass	String (ASN.1 UTF8String)	o
nameLine1	String (ASN.1 UTF8String)	o
nameLine2	String (ASN.1 UTF8String)	o
addressLine1	String (ASN.1 UTF8String)	o
addressLine2	String (ASN.1 UTF8String) It may optionally give an address of the entity	o
poBox	String (ASN.1 UTF8String)	o
zip	String (ASN.1 UTF8String)	m
city	String (ASN.1 UTF8String)	m
country	String (ASN.1 UTF8String)	m
countryCode	String (ASN.1 UTF8String according to ISO 3166-1-Alpha-2 code elements)	m
description	String (ASN.1 UTF8String)	o
mainContact	Record (ASN.1 Sequence) defined as type EntityContact, with the following self-explanatory fields: .contact, telWorkingTime, telOutsideWorkingTime,email, mobile, telFax, workingTime	m
customerService-Contact	EntityContact (see above)	o
itContact	EntityContact (see above)	m
operationalCon-tact	EntityContact (see above)	m
commercialCon-tact	EntityContact (see above)	m
website	String (ASN.1 UTF8String)	o
companyRegistra-tionNumber	String (ASN.1 UTF8String)	o
established	GeneralizedTime	o
bankDetails	Record (ASN.1 SEQUENCE) made of the following self-explanatory fields: bankAccount, sortCode, bic, iban, currencyCode, vatID	m

entityId is the identifier of the entity providing the data.

entityType indicates the type of entity. It can be a Toll Charger, a service provider or a service entity that acts on behalf of one or more Toll Chargers or Service Providers.

relatedEntityId may optionally indicate a related entity (used in case of service entities).

entityClass may optionally describe the entity as text.

nameLine1 and **nameLine2** may optionally give a name of the entity.

addressLine1 and **addressLine2** may optionally give an address of the entity.

poBox may optionally give the post box address of the entity.

zip shall give the zip code of the entity.

city shall give the city of the entity.

country shall give the country of the entity as a character string

countryCode shall give the country code of the entity

description may optionally give further descriptive information of the entity.

mainContact shall give a name acting as a main contact for the entity.

customerServiceContact may optionally give a name acting as a customer service contact for the entity.

itContact shall give a name acting as a information technology contact for the entity.

operationalContact shall give a name acting as an operational contact for the entity.

commercialContact shall give a name acting as a commercial contact for the entity.

website may optionally give the url of website of the entity

companyRegistrationNumber may optionally give the registration number of the entity.

established may optionally give the registration date of the entity.

bankDetails shall give the bank coordinates of the entity.

Table 19 indicates the fields in the **gnssContext** data structure, which is formally defined in Annex A.

Table 19 — gnssContext fields

Field name	Data type/Data description	m/o
contextInterrelations	ContextInterrelations as defined in ISO 17575-3	o
regimeContextData	It is a series of records (ASN.1 SEQUENCE OF SEQUENCE), each made of the following fields: iso175753ADU, feeModifiers	m

contextInterrelations can optionally define the interrelations with other toll regimes. Related rules and configuration parameters are defined in ISO 17575-3.

regimeContextData shall be used to provide details of the GNSS context that will be used by the Toll Service Provider to determine the behaviour of the Front End. For each record in **regimeContextData**, the field **iso175753ADU**, inherited from ISO 17575-3, specifies the rules for collecting toll information, and the **feeModifiers** field may optionally list the exempted vehicle classes or the possible discounts for each class based on time periods.

Table 20 indicates the fields in the **dsrcContext** data structure, which is formally defined in Annex A.

Table 20 — dsrcContext fields

Field name	Data type/Data description	m/o
regimeContextData	It is a series of records (ASN.1 SEQUENCE OF SEQUENCE), each made of the following fields: iso175753ADU, feeModifiers	m

regimeContextData shall be used to provide details of the DSRC context that will be used by the Toll Service Provider to determine the behaviour of the OBE. It has the same format and semantics of the **regimeContextData** for the **gnssContextdata** structure.

The **feeModifiers** field may optionally list the exempted vehicle classes or the possible discounts for each class based on time periods.

Table 21 indicates the fields in the **dsrcClosedContext** data structure, which is formally defined in Annex A.

Table 21 — dsrcClosedContext fields

Field name	Data type/Data description	m/o
dsrcClosedContext	Record (ASN.1 SEQUENCE) made of the following fields: ClosedContextDefinition, FeeModifiers	m

ClosedContextDefinition is a record (ASN.1 SEQUENCE) made of the following fields: closedContextType and closedSystem.

The **closedContextType** is defined as an INTEGER on one octet that allows for distinguishing between closed contexts operated by a single operator (Toll Charger) and those made up of interconnected operators. The following values are defined for ClosedContextType:

— 1, only one operator (Toll Charger) is present in the whole closed system;

— 2, more operators (Toll Chargers) are present in the closed system.

The **closedSystem** field brings the physical description of the toll context and the related tariff schema (see below). Context information related to a closed DSRC tolling system take into consideration the various possible configurations of such tolling systems. The most complex case is an interconnected closed scheme with multiple toll chargers (no intermediate barriers), with the possibility of varying tariffs by sections, toll chargers and types of roads.

The **feeModifiers** field may optionally list the exempted vehicle classes or the possible discounts for each class based on time periods.

The **ClosedSystem** data structure is described in Table 22 and is formally defined in Annex A. The elements of the **ClosedSystem** data structure are all optional, to allow maximum flexibility in the definition of the closed context. However, at least one element shall be defined.

Table 22 — Closed System data structure

Field name	Data type/Data description	m/o
ClosedSystem	Record (ASN.1 SEQUENCE) of optional fields: tollContextOverview, tariffClassDefinition, localVehicleClassDefinition, timeClassDefinition, userClassDefinition, feeDefinition	m

The first 5 elements of the **ClosedSystem** data structure – **tollContextOverview**, **tariffClassDefinition**, **localVehicleClassDefinition**, **timeClassDefinition** and **userClassDefinition** – are imported from ISO 17575-3 and bear the same semantics.

The **FeeDefinition** data type allows defining two basic modes of tolling: one based on the recognition of tolling points encountered by the vehicle while traversing the toll context between an entry and

an exit point, and a second one based on pre-defined paths between an entry and an exit point. The FeeDefinition data type is formally defined in Annex A and described in Table 23.

Table 23 — Fee Definition data structure

Field name	Data type/Data description	m/o
FeeDefinition	Choice (ASN.1 CHOICE) between actualPath and predefinedPath	N/A

The two possible types of tolling in the closed DSRC system are specified in 6.7.1 and 6.7.2.

6.7.1 ActualPath: tolling by recognition of traversed points

The **ActualPath** data type is expressed as a record (ASN.1 SEQUENCE) made of the following fields:

— **tariffTable** which is defined by the **TariffTable** data type inherited from ISO 17575-3, and with the same semantics;

— **ClosedContextLayout**, which is the description of the toll context.

Both fields are optional. However, at least one field shall be defined.

The layout of the closed context is defined as a set of boundary points as minimum, but can further sub-structured into closed sections, as described in Table 24 and formally defined in Annex A.

Table 24 — Closed Context Layout data structure

Field name	Data type/Data description	m/o
ClosedContextLayout	Record (ASN.1 SEQUENCE) made of the following fields: tollOperators, dsrcChargeObjects, internalPoints, interconnectionPoints, closedSections, tollContextLayoutVersion, VersionAndValidity, tollContextAuthenticator.	N/A

The **tollOperators** field allows for optionally defining the interconnected toll chargers that operate inside the closed system. If not specified, the toll charger that originates the efcContextDataADU shall be meant as the only toll charger operating in the domain.

To describe a DSRC closed tolling system, a boundary (points where vehicles enter/exit the system) shall be identified. The **dsrcChargeObjects** field defines the toll domain as a record made of fields of the same type DsrcChargeObjectId (ASN.1 SEQUENCE OF), which can be used to specify this boundary. This is the minimum required description of the closed layout when the **actualPath** branch of the CHOICE is selected.

However, there might be the need to specify additional information in the toll domain when multiple toll chargers are operating within it. As a tolling system is generally a meshed network, when the toll is not just based on entry/exit stations (boundaries) it may be necessary to use additional detection points located inside the network in order to determine the actual path of a vehicle in the network. These location points are described by the ASN.1 type **InternalPoints** and can optionally be specified by the **internalPoints** field.

The **interconnectionPoints** field can optionally be used, in cases where there is a multi-toll charger closed system, to specify identified interconnection points between sections belonging to different toll chargers.

When multiple toll chargers are present in the same DSRC closed context system, **closed sections**, and **segments** (see Figure 5) may be specified. A **closed section** is defined as a record (ASN.1 SEQUENCE) of the following fields:

— **segments**, which is a record of fields of the same type (ASN.1 SEQUENCE OF) ClosedSegment (see below);

— **operatedBy**, which is an optional record (ASN.1 SEQUENCE) that contains the Toll Operator identifier as an ASN.1 provider type, and an optional description;

— **applicableTimeClasses**, which is an optional record of fields of the same type (ASN.1 SEQUENCE OF) TimeClassId, imported by ISO 17575-3 with the same semantics;

— **locationClass**, which is a field of type LocationClassId imported by ISO 17575-3 with the same semantics.

A ClosedSection is a set of consecutive segments that belong to the same Toll Charger and the same set of applicable *TimeClass* (inherited type from ISO 17575-3).

A **segment** shall be defined as a part of the tolling system that:

1. does not include tolling stations;
2. does not include connections with other closed sections;
3. does not contain bifurcations or joins;
4. belongs to a unique toll domain;
5. has the same direction as other preceding or following segments.

A segment is defined by a set of attributes:

1. **startOfSegment**, which can be alternatively the point of bifurcation or joining with other segments (indicated by a SEQUENCE of SegmentIds), or a tolling station, or the interconnection with another toll domain, or the border with another closed section of the same toll charger;
2. **endOfSegment**, defined in the same way as startOfSegment;
3. **chargeDistance**, which is the segment length used to compute the related tariff. It may be the effective length of the road or any other conventional length used to compute the toll;
4. **realDistance**, which can optionally indicate the actual length of the segment.

A general layout of a closed system is depicted in Figure 5. In the figure, different Toll Chargers are indicated by hexagons in different shades of gray and lines represent paths that can be traversed by vehicles when in the closed system.

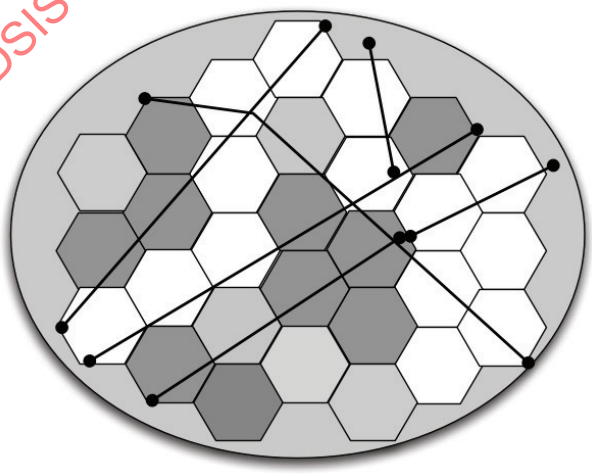

Figure 5 — DSRC closed system

The **tollDefinitionVersion** field defines the version and validity of the toll layout.

The **tollDefinitionAuthenticator** can optionally be used to authenticate the layout definition.

6.7.2 PreDefinedPath: Tolling by predefined paths

An alternate type of tolling for a closed DSRC system is by predefined paths. In this case, the tariff structure shall be based on entry point, exit point and one of a set of possible predefined paths between entry and exit.

NOTE This type of tolling is currently in use in the Italian tolling system.

The **PredefinedPath** data type is expressed as a record (ASN.1 SEQUENCE) made of a **tollDefinition** field and of a **pathDefinition** field, which, respectively, define the tariff structure and the layout of the closed DSRC system. Both elements of the SEQUENCE are optional, in order to avoid re-sending the layout structure (event with a low occurrence frequency) when only the tariffs change (event with a higher occurrence frequency). However, at least one element of the SEQUENCE shall be defined.

The **TollDefinition** data type is formally defined in Annex A and described in Table 25.

Table 25 — TollDefinition data structure

Field name	Data type/Data description	m/o
tollTable	Series of records (ASN.1 SEQUENCE OF SEQUENCE), where each record is made of the following fields: tollPath, tariffClassId, fee	m
tollDefinitionVersion	VersionAndValidity type, as imported from ISO 17575-3	m
tollDefinitionAuthenticator	Authenticator defined as ASN.1 AuthenticatorEfc	o

The **tollTable** field allows for describing the tariff schema as a multi-dimensional table, where for each occurrence of entry and exit (identified by DSRCChargeObjects) a number of paths can be uniquely identified within the scope (entry point, exit point). The tollTable is a record (ASN.1 SEQUENCE) of the following fields:

- **tollPath**, which is a record (ASN.1 SEQUENCE) made of the following fields:
  - **entryDSRCChargeObject**, which specifies the entry point in the closed system as an ASN.1 DSRCChargeObjectId type;
  - **internalPathId**, which identifies one possible path between the entry and the exit points as an InternalPathId (ASN.1 DSRCChargeObjectId type);
  - **exitDSRCChargeObject**, which specifies the exit point in the closed system as an ASN.1 DSRCChargeObjectId type;
- **tariffClassId**, which identifies the tariff to be applied for the tollPath as an ASN.1 TariffClassId, imported from ISO 17575-3 with the same format and semantics;
- **fee**, which contains the applied fee, is used elsewhere in this International Standard and is defined of type **ExtendedPaymentAmount** data type, which is a record (ASN.1 SEQUENCE) of the following fields:
  - **paymentFeeAmount**: indicates the net amount as a signed INTEGER;
  - **paymentFeeUnit**: indicates the unit of payment. The related type is imported from ISO 14906:2011/Amd1:2015;
  - **vatRate**: indicates the VAT in 0.01 % as an INTEGER ranging from 0 to 10 000.

The **tollDefinitionVersion** field shall define the version and validity of the toll table.

The **tollDefinitionAuthenticator** can optionally be used to authenticate the table definition.

The **PathDefinition** data type is formally defined in Annex A and is described in Table 26.

Table 26 — PathDefinition data structure

Field name	Data type/Data description	m/o
pathTable	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record made of the following fields: pathComposition, pathDescription	m
pathDefinitionVersion	VersionAndValidity type, as imported from ISO 17575-3	m
pathDefinitionAuthenticator	Authenticator defined as ASN.1 AuthenticatorEfc	m

The **pathTable** field lists all defined paths in the closed system. Each path shall be defined in the **pathComposition** field by an entry charge object, an exit charge object, a unique identifier within the scope (entry object, exit object) and, optionally, by a list of observation points, which can be either internal points or interconnection points (see 6.7.1).

The **pathDefinitionVersion** field shall define the version and validity of the path definition.

The **pathDefinitionAuthenticator** can optionally be used to authenticate the path definition.

6.8 ContractIssuerListADU data structure

The contract issuer list ADU is used by a Toll Service Provider to provide Toll Charger with contractual information stored on OBEs.

The contractIssuerListADU is made up of one data structure of the ContractIssuerListADU data type, which is a sequence of information related to the supported OBEs. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 27 indicates the fields of the SEQUENCE data type that constitute the ContractIssuerListADU. The ContractIssuerListADU is formally defined in Annex A.

Table 27 — ContractIssuerListADU fields

Field name	Data type/Data description	m/o
efcContextMark	EFC-ContextMark as inherited from ISO 14906	m
equipmentClass	INTEGER (0..32767)	m
manufacturerID	INTEGER(0.. 65535)	m
uniquePartOfPan	UTF8String	m
typeOfEFCApplication	UTF8String	m
securityLevel	UTF8String	m
acCrKeyReference	INTEGER (0..255)	m
authKeyReference	INTEGER	m
validFrom	GeneralizedTime	m

efcContextMark shall contain the EFC context mark as stored in the OBE.

equipmentClass shall contain the equipment class as stored in the OBE.

manufacturerID shall contain the manufacturer id as stored in the OBE and defined in ISO 14906.

uniquePartOfPan shall contain the leftmost unique digits of PAN.

typeOfEFCApplication shall specify the EFC application to use for the specific context mark (e.g. EN 15509, GNSS, PISTA, UNI, etc.).

securityLevel shall specify the security level to use for the defined EFC application.

acCrKeyReference shall specify the key in the keyset to use for access credentials.

authKeyReference shall specify the key in the keyset to use for TSP authenticator.

validFrom shall specify the starting time the information is considered valid.

The TSP shall provide one structure for each OBE it wants to be specified. This means, for example, that if a TSP is using OBEs from different manufacturers for the same PAN, the TSP has to provide a series of ContractIssuerListADUs, one for each manufacturer id.

6.9 ExceptionListADU data structure

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

The exceptionListADU is made up of a single data structure of the ExceptionListADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI may have a null value, to indicate that the ADU is intended for all recipients.

Table 28 indicates the fields in the ExceptionListADU data structure, which is formally defined in Annex A.

Table 28 — ExceptionList fields

Field name	Data type/Data description	m/o
exceptionListVersion	INTEGER	m
exceptionListType	INTEGER	m
exceptionValidityStart	Time stamp as ASN.1 GeneralizedTime	o
exceptionValidityEnd	Time stamp as ASN.1 GeneralizedTime	o
exceptionListEntries	Record (ASN.1 SEQUENCE), see Table 29	m

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

exceptionListType shall be is used to indicate the type of exception list (black list, white list, grey list, etc.). This International Standard does not define the semantics all types of lists. However, the following three values of the exceptionListType are defined:

1. black list (see 3.2);
2. white list (see 3.21);
3. user discount list: A list of users who are entitled to discounts. The means to apply these discounts and their amounts are subject to bilateral agreements and are out of the scope of this International Standard.

The above meanings of the values of exceptionListType are also documented as comments in the type formal description in Annex A. All other possible values of the ExceptionListType are just options to differentiate lists. Bilateral agreements or further specifications are needed to define the exact semantics for each list type, e.g. for EETS.

exceptionValidityStart can optionally 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.

exceptionValidityEnd can optionally specify and expiry date and time for the exception list.

exceptionListEntries is a structured field that contains a list of entries for the Toll Service Provider's exception list as described in Table 29. These fields are specified in Annex A.

Table 29 — ExceptionListEntry fields

Field name	Data type	m/o
userId	Record (ASN.1 SEQUENCE) of the following fields; pan, contractSerialNumber, licencePlateNumber, obeID	m
statusType	INTEGER on two octets	m
reasonCode	INTEGER on two octets	m
entryValidityStart	Time stamp as ASN.1 GeneralizedTime	o
entryValidityEnd	Time stamp as ASN.1 GeneralizedTime	o
vehicleParameters	Record (ASN.1 SEQUENCE), see Table 38 for a detailed description	o
actionRequested	INTEGER defined by ASN.1 ExceptionListActionType type	o
efcContextMark	EfcContextMark, as defined in ISO 14906	o
vatID	Character string as ASN.1 UTF8String type	o

userId field shall contain the identifier of the user related to the entry. It is made of the following fields:

— **pan**, which may optionally contain the PersonalAccountNumber, according to ISO 14906;

— **contractSerialNumber**, which may optionally contain the user's contract serial number, according to ISO 14906;

— **licencePlateNumber**, which may optionally contain the user's licence plate number, according to ISO 14906;

— **obeID**, which may optionally contain the OBE identifier, according to ISO 17575-1.

All fields are optional, but at least one shall be specified.

statusType shall indicate the limitations to which the user is subject to in the related toll domain. The following values are allowed:

— 0, which means that the user is blocked for all possible DSRC applications;

— 1, which means that the user is blocked for the Toll Charger receiving the information;

— 2, which means that the user is blocked for all tolling schemata requiring an odometer;

— 3, which means that the user is not blocked for any DSRC application.

reasonCode shall indicate which are the reasons for the given limitations. The following values are allowed:

— 0, reason not to be disclosed;

— 1, OBE has been deactivated;

— 2, OBE has been stolen;

— 3, temporary technical problems;

— 4, suspected technical manipulation;

— 5, user's payments are late;

— 6, no user payments;

— 7, contract holder insolvent;

— 8, user in white lists;

— 9, user optioned out of this toll domain;

— 10, user requested temporal suspension;

— 11, contract closed by Toll Service Provider and OBE not yet returned;

— 12, contract closed by user and OBE not yet returned;

— 13, OBE not valid;

— 14, lost OBE;

— 15, OBE not provided by TSP;

— 16, OBE on stock and not yet assigned by TSP;

— 17, OBE returned after end of contract;

— 18, OBE returned due to malfunction.

entryValidityStart and **entryValidityEnd** can optionally indicate the time period the information contained in the entry is valid. In case of conflicts with the similar fields in the ExceptionList structure (exceptionValidityStart and exceptionValidityEnd, respectively), the values contained in the ExceptionListEntry field shall take precedence.

vehicleParameters can optionally indicate vehicle related parameters that are allowed in cases where white list-based tolling schemata generate valid billing details entries. See Table 38 for details.

actionRequested can optionally indicate an action requested by the Toll Service Provider to the Toll Charger, e.g. to consider the OBE as valid/invalid, put it in the Toll Charger's black list or remove the OBE. The following values are allowed:

— 1, reject this OBE;

— 2, invalidate this OBE. In some contexts, this means set black list bit, if supported by TC;

— 3, accept this OBE;

— 4, remove this OBE.

efcContextMark can optionally specify the ecfContextMark related to the exception.

vatId can optionally specify the VAT identifier of the owner of a vehicle when the exception list is used as a white list.

NOTE The Toll Charger needs to report the VAT identifier of the service user to its fiscal authorities when using the mandatory reverse charge system inside the EU whenever applicable (e.g. cross-country ferry connections within or to and from the EU).

6.10 ReportAbnormalOBEADU data structure

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

The reportAbnormalOBEADU is made up of one data structure of the ReportAbnormalOBEADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 30 indicates the fields of the ReportAbnormalOBE data structure, which is formally defined in Annex A.

Table 30 — ReportAbnormalOBEADU fields

Field name	Data type/Data description	m/o
userId	Record (ASN.1 SEQUENCE) of the following fields; pan, contractSerialNumber, licencePlateNumber, obeID	m
dateandTime	Time stamp as ASN.1 GeneralizedTime	m
efcContextMark	EfcContextMark, as defined in ISO 14906	o
abnormalOBEReasonCode	INTEGER on two octets defined as AbnormalOBEReasonCode	m

userId shall contain the UserID recorded by the Toll Charger's equipment.

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

efcContextMark can optionally be used to specify the EFC context mark related to the OBU as read by the Toll Charger.

abnormalOBEReasonCode shall contain the coding of the reasons for the abnormal OBE report determined by the Toll Charger. It can be a combination of the following:

— reasonNotToBeDisclosed, when the Toll Charger does not specify the reason;

— obeIsDefect, when the Toll Charger detected a defect in the OBE;

— obeIsNotWorkingProperly, when the Toll Charger detected a malfunction in the OBE;

— userShowsFraudBehaviour, when the Toll Charger detected a fraudulent behaviour of the OBE;

— userShowsViolatingBehaviour, when the Toll Charger detected a fraudulent behaviour of the user.

6.11 Toll DeclarationADU data structure

The toll declaration ADU can be used by both the Toll Service Provider and the Toll Charger either in response to a request to receive toll declarations or as a push ADU when toll declarations are available to be transferred.

The tollDeclarationADU is made up of one data structure of the TollDeclarationADU Data Type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 31 indicates the fields of the toll declaration data structure, which is formally defined in Annex A.

Table 31 — TollDeclarationADU fields

Field name	Data type/Data description	m/o
tollDeclarationId	Identifier as an ASN.1 INTEGER type	m
gnssTollDeclaration	Record (ASN.1 SEQUENCE), whose fields are of type either ChargeReport or AuthenticatedChargeReport	m
actionCode	INTEGER on two octets defined as ActionCode	o

tollDeclarationId shall contain the identifier for the toll declaration, which is unique within the generating entity.

gnssTollDeclaration is a sequence of either ChargeReport or AuthenticatedChargeReport, which is imported from ISO 17575-1 according to the format specified in the EFC context data, which details the data required for the toll declaration.

actionCode can optionally indicate the action associated with the specific ADU, e.g. normal sending, revocation, adjustment, etc. The following values shall be used in the tollDeclarationADU:

— 0, normal sending of a toll declaration;

— 1, revocation of a toll declaration;

— 3, resend.

6.12 BillingDetailsADU data structure

The billing details ADU 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 ADU when billing details are available.

The billingDetailsADU is made up of one data structure of the BillingDetailsADU Data Type. When transferring this ADU, the value in the **informationRecipientID** field of the APCI shall identify a valid entity (a null value is not permitted).

Table 32 indicates the fields of the BillingDetailsADU data structure, which is formally defined in Annex A.

Table 32 — BillingDetailsADU fields

Field name	Data type/Data description	m/o
billingDetailsId	Record (ASN.1 SEQUENCE) of the following fields: issuerId, billingDetailsNum, and dateOfService	m
tollContext	Identifier of the tolling context as Provider type imported from ISO 14906	m
userId	Record (ASN.1 SEQUENCE) of the following fields; pan, contractSerialNumber, licencePlateNumber, obeID	o
relatedBillingDetails	INTEGER that identifies a different APDU	o
period	Time period expressed as ASN.1 Period, imported from ISO 17575-1	o
billingDetailsAmount	Record (ASN.1 SEQUENCE) of the following fields: paymentFeeAmount, paymentFeeUnit, vatRate	m
usageDetails	Record (ASN.1 SEQUENCE) of the following fields: contextName, appliedUserClass, perDeclaredVehicleClasses, perUsedTimeClasses	o
refTollDeclaration	INTEGER that identifies a different APDU	o
associatedEventData	Record (ASN.1 SEQUENCE) of the following fields cccRecord, imageRecord, anprRecord, classificationRecord, operatorRecord, dsrcData	o
actionCode	INTEGER on two octets defined as ActionCode	o
paymentReference	Character string defined as ASN.1 UTF8String type	o

billingDetailsId shall contain the unique identifier (within the Toll Charger or Toll Service Provider) for the billing detail. The optional dateOfService field allows adding time related information to the identifier, to ease classification and retrieval of BillingDetails records.

tollContext shall contain the unique identifier for the Toll Charger that issues the billing details. The semantics of this field is further explained in the description of usageDetails below.

userId can optionally contain the userId of the Service User to whom the billing detail shall be charged. If the userId is not stated, the Toll Service Provider may get this information out of the referenced toll declaration.

relatedBillingDetails can optionally contain the identifier of a previously issued billing details to which the present billing details can be associated. The meaning of this association is not defined in this International Standard.

period can optionally contain the period of time the billing details are related to.

billingDetailsAmount shall contain the sum of all fee related elements in the UsageDetails, their associated VAT rate and currency. This requires that only UsageDetails with the same VAT rate and the same currency can be contained in one billingDetailsADU. Separate billingDetailsADUs shall be used if any of VAT rate or currency differs.

usageDetails can optionally provide extended information about the real usage of a tolling object (section/area/cordon) by the user. There shall be one usage detail per user and period. The usageDetails field, if used, contains a sequence of detailed information, possibly pertaining to different tolling schemata. Subfields included in the **usageDetails** are

— **contextName**, which shall indicate the name of the toll context as a character string,

— **appliedUserClass,** which shall contain the applied tolling class to that specific user as a character string,

— **perDeclaredVehicleClasses,** which shall contain the applied tolling class to that specific vehicle as a character string, and

— **perUsedTimeClasses,** which shall specify as a SEQUENCE of fields, for each time class, expressed by the **appliedTimeClass** subfield, and for each cost centre, expressed by the **costCenter** subfield, the actual usage for which billing details are issued, expressed as a record (SEQUENCE OF) made of **UsageList** records.

usageList records can alternatively (ASN.1 type CHOICE) be specified in terms of

— travelling in sectioned roads, where information on the entry, intermediate traversed and exit sections are given. It also allows for indicating for each section, via the **modeOfOperation** field, whether the charge objects used to detect the usage are physical objects (e.g. gantries) or computed values. The **forSectionedRoads** field is a record (ASN.1 SEQUENCE) of the following fields:

  — **howManyTimes**, which is an INTEGER indicating how many times the same trip has been done. This can be used, e.g., to apply special tariffs to commuters;

  — **listOfSections**, which is a record (ASN.1 SEQUENCE) that can optionally be used to specify the traversed sections. Each Section, in itself, is a structure containing, among others, the time of tolling as an optional field named tollEventTime. If the order of passage matters, the tollEventTime must be specified in all Sections contained in the listOfSections record;

  — **invoiceAggregationNumber**, which is an identifier specified as a string of characters that can optionally be used to link usage details aggregated into one invoice;

  — **fee**, which can optionally specify the amount to be paid;

  — **feeQualifier**, which can optionally qualify the fee amount as a standard charge, a toll substitute, a belated payment or a reimbursement,

— travelling in areas, where the distance travelled in areas and optionally the time spent in areas are specified. The **forTravellingInArea** field is a record (ASN.1 SEQUENCE) of the following fields:

  — **areaDisplayName**, which is a character string indicating the name of the area;

  — **accumulatedDistance**, which is the distance travelled in the area, expressed in metres;

  — **beginOfAccumulation** and **endOfAccumulation**, which can optionally specify the starting time and the ending time of the travel in the area;

  - **fee**, and **feeQualifier,** which can optionally be used to indicate the amount to be paid and to qualify that amount,

- permanence in areas, where time of staying in given areas are specified. The **forStayingInArea** field is a record (ASN.1 SEQUENCE) of the following fields:
  - **areaDisplayName**, which is a character string indicating the name of the area;
  - **qualifier**, which can optionally indicate either the times the vehicle stayed in the given area or the entrance time;
  - **stayedDuration**, which indicates the charge units used while staying in the area;
  - **chargeUnit**, which indicates the units used for charging, as imported from and specified by ISO 17575-3;
  - **fee**, and **feeQualifier,** which can optionally be used to indicate the amount to be paid and to qualify that amount,
- cordon crossing, where indication of the entrance time, exit time, and tolling objects indicating a cordon crossing are specified. The **forCordonCrossing** field is a record (ASN.1 SEQUENCE) of the following fields:
  - **qualifier**, which indicates either the times the vehicle stayed in the given area or the entrance time;
  - **entranceChargeObject**, which indicates the charge object that detected the entrance in the cordon, as imported from and specified by ISO 17575-3;
  - **entranceChargeObjectName**, which can optionally specify the name of the entrance charge object as a character string;
  - **exitChargeObject**, which can optionally indicate the charge object that detected the exit from the cordon, as imported from and specified by ISO 17575-3;
  - **entranceChargeObjectName**, which can optionally specify the name of the exit charge object as a character string;
  - **exitTime**, which can optionally indicate the time of exit from the cordon;
  - **fee**, and **feeQualifier,** which can optionally be used to indicate the amount to be paid and to qualify that amount, and
- free text, where a free text is given, expressed as a character string (UTF8String), to describe the reasons for tolling. The freeTextDetail field is a record (ASN.1 SEQUENCE) of the following fields:
  - **textLanguage**, which indicates the language used for the text;
  - **textDetail**, which specifies as a character string the reason for tolling;
  - **fee**, and **feeQualifier,** which can optionally be used to indicate the amount to be paid and to qualify that amount.

As for each category of usage details one or more toll contexts can be specified, the BillingDetailsADU allows one Toll Charger, identified by the **tollContext** field, to issue billing details on behalf of other Toll Chargers, identified in the sequence of usage detail information contained in the **usageDetails** field.

refTollDeclaration can optionally provide a linkage to a TollDeclarationADU that is associated with the billing details.

associatedEventData can optionally be used to attach the raw event data from which the billing detail has been generated and may include

— cccRecord, when event data come from a CCC (ISO 12813) transaction,

— imageRecord, when event data are shown in an image,

— anprRecord, when event data contain results from an automatic number plate recognition,

— classificationRecord, when event data contain a classification record for the vehicle,

— operatorRecord, when event data are generated by a human intervention, and

— dsrcData, when event data come from a generic DSRC interaction.

If used, at least one of these subfields shall not be null.

actionCode can optionally indicate the action associated with the specific ADU, e.g. normal sending, revocation, adjustment, etc. The following values shall be used in the billingDetailsADU:

— 0, normal sending of a billing details;

— 2, adjust values in a previously sent billing details;

— 3, resend a billing details.

paymentReference can optionally specify reference to this billing detail to be quoted in a payment.

The structure of the AssociatedEventData type is shown in Table 33 and is formally defined in Annex A.

Table 33 — AssociatedEventData fields

Field name	Data type/Data description	m/o
cccRecord	Record (ASN.1 SEQUENCE) of the following fields: userId, timeOfEvent, locationOfEvent, cccMessages, initiatedActions, see Table 34 for details	o
imageRecord	Record (ASN.1 SEQUENCE) of the following fields: imageToBeSigned, subRecordAuthenticator, see Table 35 for details	o
anprRecord	Record (ASN.1 SEQUENCE) of the following fields: anprToBeSigned, subRecordAuthenticator, see Table 36 for details	o
classification-Record	Record (ASN.1 SEQUENCE) of the following fields: classificationToBeSigned, subRecordAuthenticator, see Table 37 for details	o
operatorRecord	Record (ASN.1 SEQUENCE) of the following fields: operatorToBeSigned, subRecordAuthenticator, see Table 39 for details	o
dsrcData	Record (ASN.1 SEQUENCE) of the following fields: dsrcRSEData, dsrcAttributesRead, dsrcAttributesWritten, dsrcAttrAuth, chargeObjectId, see Table 40 for details	o

The structure and semantics of the above fields is explained below.

Table 34 indicates the fields of the cccRecord data structure, which is formally defined in Annex A.

Table 34 — cccRecord fields

Field name	Data type/Data description	m/o
timeOfEvent	Time stamp defined by ASN.1 GeneralizedTime type	o
locationOfEvent	Record (ASN.1 SEQUENCE) of the following fields: positionOfLocation, location	o
cccMessages	Record of fields of the same type (ASN.1 SEQUENCE OF), each field containing DSRC transaction data as DsrcData type	m
initiatedActions	Record of fields of the same type (ASN.1 SEQUENCE OF), each field of type INTEGER defined as InitiatedActions	m

timeOfEvent can optionally contain the date and time of the CCCRecord.

locationOfEvent can optionally contain the Toll Charger coding of the location of where the CCCRecord was generated.

cccMessages shall contain the raw DSRC data for the CCCRecord.

initiatedActions shall contain the details of the actions that have been initiated by the Toll Charger as a result of the CCCRecord. The following values are allowed:

— 0, vehicle stopped following the CCC result;

— 1, CCC result indicated that a violation may have occurred;

— 2, CCC data has been retained as evidence of violation;

— 3, request to add UserID to the TSP exception list made;

— 4, UserId added to the TC exception list.

Table 35 indicates the fields of the imageRecord data structure, which is formally defined in Annex A.

Table 35 — imageRecord fields

Field name	Data type/Data description	m/o
imageRecordId	Identifier defined as INTEGER.	m
imageDateTime	Time stamp defined as ASN.1 GeneralizedTime type	m
imageCameraId	Generic string (UTF-8) that identifies the camera	m
imageReference	Identifier defined as INTEGER.	o
imageData	Image coded as OCTET STRING	o
subRecordAuthenticator	Authenticator defined as AuthenticatorEfc	o

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

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

imageCameraId shall contain the Toll Charger identifier of the camera which generated the ImageRecord.

imageReference can optionally contain the Toll Charger unique identifier for the image.

imageData can optionally contain the JPEG image binary encoded.

subRecordAuthenticator can optionally contain an authenticator calculated over the content of the imageRecord.

Table 36 indicates the fields of the AnprRecord data structure, which is formally defined in Annex A.

Table 36 — AnprRecord fields

Field name	Data type/Data description	m/o
anprRecordId	Identifier defined as INTEGER	m
associatedImages	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record made of the following fields: anprImage, and contextImage	m
imageDateTime	Time stamp as ASN.1 GeneralizedTime type	m
imageCameraId	Generic string (UTF-8) that identifies the camera	m
determinedVRM	Record (ASN.1 SEQUENCE) made of the following fields: anprResult, anprConfidence, secondaryAnprResult, manualResult, operatorId	m
vehicleDetails	Record (ASN.1 SEQUENCE) made of the following fields: vehicleMake, vehicleModel, vehicleColour	m
subRecordAuthenticator	Authenticator defined as AuthenticatorEfc	o

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

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

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

imageCameraId shall contain the Toll Charger identifier of the camera which generated the ImageRecord.

determinedVRM shall contain the ANPR result(s).

vehicleDetails shall contain the determined vehicle details if available.

subRecordAuthenticator can optionally contain an authenticator calculated over the content of the anprRecord.

Table 37 indicates the fields of the classificationRecord data structure, which is formally defined in Annex A.

Table 37 — classificationRecord fields

Field name	Data type/Data description	m/o
classificationRecordID	Identifier defined as INTEGER	m
derivedLocalClass	INTEGER on two octets	m
vehicleParameters	Record (ASN.1 SEQUENCE) of fields as described in Table 38	m
subRecordAuthenticator	Authenticator defined as AuthenticatorEfc	o

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

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

vehicleParameters shall contain the raw measurements of the vehicle parameters recorded by the classification system. The only mandatory field in the data structure is dateAndTime, which shall indicate the time the data have been recorded. All other fields are optional.

Table 38 indicates the fields of the VehicleParameters type, which is formally defined in Annex A.

Table 38 — VehicleParameters fields

Field name	Data type/Data description	m/o
dateAndTime	Time stamp defined as ASN.1 GeneralizedTime type	m
vehicleClass	Character string defined as ASN.1 UTF8String type	o
vehicleLength	INTEGER expressing the length of the vehicle in mm	o
vehicleWidth	INTEGER expressing the width of the vehicle in mm	o
vehicleHeight	INTEGER expressing the height of the vehicle in mm	o
heightAbove1stAxle	INTEGER expressing the height of the vehicle above the first axle in mm	o
vehicleAxles	INTEGER expressing the number of vehicle axles in the range 0 to 15	o
trailerAxles	INTEGER expressing the number of trailer axles in the range 0 to 7	o
trailerPresence	BOOLEAN expressing the presence of a trailer	o

subRecordAuthenticator can optionally contain an authenticator calculated over the content of the classificationRecord.

Table 39 indicates the fields of the operatorRecord data structure, which is formally defined in Annex A.

Table 39 — operatorRecord fields

Field name	Data type	m/o
operatorRecordId	Identifier defined as INTEGER	m
operatorData	Record (ASN.1 SEQUENCE) made of the following fields: operatorTime, operatorClass, operatorId, operatorVRM, operatorPAN, operatorOBUId, machineReadPAN, machineReadOBUId	m
subRecordAuthenticator	Authenticator defined as AuthenticatorEfc	o

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

operatorData shall contain the information collected and recorded by the human operator. It is a record made of the following fields:

— operatorTime, which contains a time stamp;

— operatorClass, which optionally specifies as a two octets INTEGER the operator class;

— operatorId, which optionally specifies the identifier of the operator as an INTEGER;

— operatorVRM, which optionally specifies the VRM, as collected by the operator as an OCTET STRING;

— operatorPAN, which optionally specifies the PAN, as collected by the operator;

— operatorOBUId, which optionally specifies the OBU identifier, as read by the operator;

— machineReadPAN, which optionally specifies the PAN, as read automatically;

— machineReadOBUId, which optionally specifies the OBU identifier, as read automatically.

subRecordAuthenticator can optionally contain an authenticator calculated over the content of the operatorRecord.

Table 40 indicates the fields of the dsrcData data structure, which is formally defined in Annex A. If used, at least one field shall be not null.

Table 40 — dsrcData fields

Field name	Data type/Data description	m/o
dsrcRSEData	DsrcRSEData, Data recorded by the RSE	o
dsrcAttributesRead	Record made of fields of the same type (ASN.1 SEQUENCE OF), that are of the Attributes data type imported from ISO 14906	o
dsrcAttributesWritten	Record made of fields of the same type (ASN.1 SEQUENCE OF), that are of the Attributes data type imported from ISO 14906	o
dsrcAttrAuth	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record of type DsrcAttrAuth, and described below Table 40	o

dsrcRSEData field can optionally contain the data recorded by the RSE during the DSRC transaction as formally defined in Annex A. It can also optionally link to an exceptionListEntry related to the user.

dsrcAttributesRead can optionally contain the details of the attributes read from the OBE during the DSRC transaction.

dsrcAttributesWritten can optionally contain the details of the attributes written to the OBE during the DSRC transaction.

dsrcAttrAuth can optionally contain the information that is required to verify the authenticators received during the DSRC transaction as formally defined in Annex A. It is a series of records, one for each DSRC transaction. Each record made of the following fields:

— **attrOrigEncoding**, which is a BIT STRING containing the encoding of the AttributeList;

— **rndRSE**, which is the random number generated by the RSE for encoding data;

— **keyRef**, which is the reference to the key used for encoding data, expressed as an INTEGER;

— **authCode**, which is an OCTET STRING containing the authentication code;

— **result**, which specifies whether the authentication has been checked and, if checked, whether the result of checking was positive or negative.

6.13 PaymentClaimADU data structure

The payment claim ADU can be used by the Toll Charger either in response to the request ADU sent by the Toll Service Provider or as a push ADU when new payment claims are available.

The payment claim APDU shall contain only one data structure of the PaymentClaimADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 41 indicates the fields of the PaymentClaimADU data structure, which is formally defined in Annex A.

Table 41 — PaymentClaimADU fields

Field name	Data type/Data description	m/o
paymentClaimId	Identifier defined as INTEGER	m
startDateTime	Time stamp defined as ASN.1 GeneralizedTime type	m
endDateTime	Time stamp defined as ASN.1 GeneralizedTime type	o
userId	Record (ASN.1 SEQUENCE) of the following fields; pan, contractSerialNumber, licencePlateNumber, obeID	o
paymentClaimAmount	Signed INTEGER defined as ExtendedPaymentAmount	m
paymentClaimStatus	Integer defined as PaymentClaimStatus	m
typeOfFee	Integer indicating the type of fee	o
referenceDetailsList	It contains a series (ASN.1 SEQUENCE OF) of records, each one being possibly of type billingDetailsList, or tollDeclarationList, or tollEventList	o
actionCode	It is used to specify the intended meaning of the payment claim	o
paymentReference	It may be used to reference this payment claim in a payment	o

paymentClaimId field shall contain the unique (within the originating entity) identifier for the payment claim.

startDateTime shall specify the start of the period for which the payment claim relates.

endDateTime can optionally 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 ADU was generated.

userId can optionally be used to provide User specific payment claims. If not specified, the amount claimed can be a result of tolls due by more users.

paymentClaimAmount shall contain the details of the net amount, currency and VAT for the payment claim.

paymentClaimStatus shall indicate the version and status of the payment claim. The following values are allowed:

— 0, which indicates the first version of the payment claim;

— 1, which indicates an amended version of the payment claim.

typeOfFee can optionally be used to indicate the type of the fee including: toll, discount, credit note, penalty, processing fee.

referenceDetailsList can optionally associate previously exchanged toll declarations, billing details or toll events to the payment claim. In particular, tolling events are specified by means of fields of TollEventId type, which is defined as a sequence of

— a CHOICE between ChargeReportCounter (imported from ISO 17575-1) and a TransactionCounter, which is used in DSRC tolling systems, and

— a usageStatementId, which shall identify the usage statement to which the tolling event is related.

actionCode can optionally indicate the action associated with the specific ADU, e.g. normal sending, revocation, adjustment, etc. The following values shall be used in the paymentClaimADU:

— 0, normal sending of Payment Claims;

— 2, adjust values in a previously sent Payment Claims ADU;

— 3, resend a Payment Claims ADU.

paymentReference can optionally specify a reference to this payment claim to be quoted in a payment.

6.14 PaymentAnnouncement ADU data structure

The payment announcement ADU can be used by the Toll Service Provider either in response to the request ADU sent by the Toll Charger or as a push ADU when new payment announcements are available.

The paymentAnnouncementADU is made up of one data structure of the PaymentAnnouncementADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 42 indicates the fields of the PaymentAnnouncementADU data structure, which is formally defined in Annex A.

Table 42 — PaymentAnnouncementADU fields

Field name	Data type/Data description	m/o
paymentAnnouncementID	Identifier defined as INTEGER	m
dueDate	Time stamp defined as ASN.1 GeneralizedTime type	m
totalamount	Amount defined as ExtendedPaymentAmount type	m
paymentStatus	INTEGER on two octets defined as PaymentStatus	m
numberOfItems	INTEGER	o
referenceDetailsList	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record being made of the following fields: referenceDetail, amount, paymentMeansType, valueDate, interestAmount	o
attachment	OCTET STRING	o
actionCode	INTEGER on two octets defined as ActionCode	o
paymentReference	Character string defines as ASN.1 UTF8String type	o

The **paymentAnnouncementId** field shall contain the unique identifier (within the originating entity) for the payment announcement.

dueDate provides date and time information on the date where the amount actually has been paid to the Toll Charger.

totalamount is the amount that is actually announced by the ADU.

paymentStatus provides detailed status information for the ADU. Currently, the following values are defined:

— paid;

— new-overdue;

— not yet due – not paid yet;

— not yet due – paid;

— due – paid;

— due – not paid yet (but interest-bearing);

— new-overdue – not paid yet (interest-bearing).

numberOfItems is an optional information on the number of referenced items in the referenceDetailsList.

referenceDetailsList provides the option to reference billing detail, toll declarations and individual usage reports. For each reference, the following fields may optionally be specified:

— **Amount**, which is the amount paid;

— **paymentMeans**, which indicates the payments means (cash, credit card, …) that was used;

— **valueDate**, which is the date when the amount of the referenceDetail was due according to contractual agreements;

— **interestAmount,** which indicates the interest that has to be paid because the referenceDetail was overdue. This attribute shall be present if paymentstatus is “new-overdue”.

attachment provides the option for attaching supporting documents, e.g. a PDF version of a payment report.

actionCode can optionally indicate the action associated with the specific ADU, e.g. normal sending, revocation, adjustment, etc. The following values shall be used in the paymentAnnouncementADU:

— 0, normal sending of payment announcement;

— 2, adjust values in a previously sent payment announcement ADU;

— 3, resend a payment announcement ADU.

paymentReference provides the possibility of specifying a reference to this payment announcement.

6.15 ProvideUserDetailsADU data structure

The provide user details ADU shall only by used by a Toll Service Provider in response to a specific request ADU from a Toll Charger.

The provideUserDetailsADU is made up of one data structure of the ProvideUserDetailsADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted). Table 43 indicates the fields of the ProvideUserDetailsADU data structure, which is formally defined in Annex A.

Table 43 — ProvideUserDetailsADU fields

Field name	Data type/Data description	m/o
originaluserIdRequest	INTEGER defined on two octets as UserIdRequestType	m
userId	Record (ASN.1 SEQUENCE) of the following fields; pan, contractSerialNumber, licencePlateNumber, obeID	m
statusFlag	INTEGER defined on two octets as UserStatus	o
listOfUserParameters	Series of records (ASN.1 SEQUENCE OF) of type UserParameterResponse (see below for details)	o

originaluserIdRequest shall contain the UserId specified in the associated request ADU.

userId shall contain the UserId contained in the corresponding response ADU.

statusFlag can optionally be used to indicate details of the current status associated with the UserId.

listOfUserParameters can optionally be used to provide the response to the requested parameters as a record (ASN.1 SEQUENCE) of the following fields:

— **requestedUserParameter** can optionally contain the code of the original requested parameter as specified in the request ADU;

— **userParameterResponse** can optionally contain the value of the requested parameter;

— **userParameterStatus** can optionally provide an explanation of why the requested user parameter has or has not been provided. The following values are allowed:

  — 0, attribute is available;

- 1, attribute is not available;
- 2, attribute cannot be sent due to privacy rule;
- 3, attribute cannot be sent by missing agreement;

— **userInfoValidityPeriod** can optionally specify the period in time when the related information is valid.

As all fields in the listOfUserParameters structure are optional, if the listOfUserParameters is used at least one field shall be specified.

6.16 ReportCCCEventADU data structure

The ReportCCCEventADU can be used in autonomous systems based toll domains by the Toll Charger either in response to the request ADU sent by the Toll Service Provider or as a push ADU when new CCC (as defined in ISO 12813) events are available.

The reportCCCEventADU is made up of a sequence of the data structure CCCEvent data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 44 indicates the fields of the CCCEvent data structure, which is formally defined in Annex A.

Table 44 — CCCEvent fields

Field name	Data type/Data description	m/o
userId	User identifier	o
timeOfEvent	Time stamp as ASN.1 GeneralizedTime type	o
locationOfEvent	Location, expressed as a record (ASN.1 SEQUENCE) of the following fields: positionOfLocation, location	o
cccMessages	Record of fields of the same type (ASN.1 SEQUENCE OF), containing DSRCData	m
initiatedActions	Record of fields of the same type (ASN.1 SEQUENCE OF), containing InitiatedAction	m

userId can optionally specify the identifier of the user.

timeOfEvent can optionally specify the time of the CCC event as determined by the Toll Charger equipment.

locationOfEvent can optionally provide details of the location of the CCC event. If specified, it is made of the following fields:

— positionOfLocation, which can optionally specify the location as a point (inherited from ISO 17575-3);

— location, which specifies the location as a couple (locationProviderId, locationID), where **locationProviderId** indicates the provider of the location (GPS, Glonass, Galileo, etc.) and **locationId** specifies as an INTEGER the identifier of the location according to the given provider.

cccMessages shall contain the record(s) of the CCC Event(s) from the Toll Charger equipment.

initiatedActions shall provide details of the actions that have been initiated by the Toll Charger as a result of the specified CCC Event. Each initiated action is defined by an INTEGER, for which the following values are allowed:

— 0, vehicle stopped following the CCC interaction result;

— 1, CCC result indicated that a violation may have occurred;

— 2, CCC data has been retained as evidence of violation;

— 3, request to add UserID to the TSP exception list;

— 4, UserId added to the TC exception list.

6.17 ProvideUserIDListADU data structure

The provide user id List ADU is used by a Toll Service Provider in response to a request ADU from a Toll Charger.

The provideUserIdListADU is made up of one data structure of the ProvideUserIDListADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 45 indicates the fields of the ProvideUserIDListADU data structure, which is formally defined in Annex A.

Table 45 — ProvideUserIDListADU fields

Field name	Data type/Data description	m/o
originalUserIdRequestType	INTEGER defined on two octets as UserIdRequestType	m
originalUserId	Record (ASN.1 SEQUENCE) of the following fields; pan, contractSerialNumber, licencePlateNumber, obeID	o
userIdList	Record of fields of the same type (ASN.1 SEQUENCE OF), each field being of type UserId	o

The **originalUserIdRequestType** and **originalUserId** attributes provide information about the original request.

The **originalUserIdRequestType** shall contain the same value as received in the related RetrieveUserIdListADU.

If the userId field was present in the request, then the **originalUserId** field shall be present in both the originalUserId field and the userIdList field.

If the userId field was not present in the request, then the userIdList shall contain details of all the TSP's users, as filtered by the originalUserIdRequestType field value.

userIdList can optionally provide the list of requested ids.

6.18 Report QA data structure

The report QA ADU can be used by either the Toll Charger or the Toll Service Provider either in response to a specific request ADU sent by counterpart or as a push ADU when new QA reports are available.

The reportQAADU is made up of one data structure of the ReportQAADU data type. When transferring this ADU, the value in the informationRecipientID field of the APCI shall identify a valid entity (a null value is not permitted).

Table 46 indicates the fields of the ReportQAADU data structure, which is formally defined in Annex A.

Table 46 — ReportQA ADU fields

Field name	Data type/Data description	m/o
qualityParameterID	Identifier defined as INTEGER	m
qualityParameterName	Character string defined as ASN.1 UTF8String	o
qualityParameterValue	REAL	o
qualityParameterStatus	INTEGER	m

qualityParameterID shall specify the type of quality parameter from a bilaterally agreed list between the Toll Service Provider and the Toll Charger.

qualityParameterName can optionally be used to provide a text description for the quality parameter.

qualityParameterValue can optionally be used to provide a value for the quality parameter.

qualityParameterStatus shall specify the status of the quality parameter. It may assume the following values:

— 0, the parameter has been measured according to a methodology defined under bilateral agreements;

— 1, the parameter has been measured as a sample;

— 2, the parameter is an average on a long period;

— 3, the parameter is a single worst value;

— 4, the parameter is a warning on a trend;

— 5, indicates that the parameter requires actions to be initiated;

— 6, indicates that the parameter is a warning for escalation;

— 7, indicates that the parameter caused escalation being activated.

7 Transfer mechanisms and supporting functions

7.1 Transfer mechanisms

The APDUs specified in Clause 6 of this International Standard shall be exchanged between the central equipment of Toll Service Providers and Toll Chargers by making use of a variety of applicable communication means. These communication means shall be used by 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) or the Canonical XML encoding rules (cXER) according to ISO/IEC 8825-4 shall be used to encode data. When a secure communication channel is to be used, the security mechanism shall be agreed between the TC and TSP for each connection, following the provisions in 7.2.

7.2 Secure communication channel

The secure channel between the TC and TSP shall be realized by installing a secure virtual private network (VPN) connection. One of the following VPN security mechanisms shall be used:

a) IPsec according to IETF RFC 6040 and related RFCs with

   1) pre-shared secrets – current state of the art method, and

   2) certificate based authentication – according to the trust model defined in ISO/TS 19299;

b) transport layer security (TLS) over a TCP/IP protocol stack according to IETF RFC 5246. In the handshake, both client and server shall be authenticated with an ISO/IEC 9594-8 (ITU-T X.509v3) certificate. The protocol shall also implement the “Transport Layer Security (TLS) Renegotiation Indication Extension” defined in IETF RFC 5746;

c) datagram transport layer security (DTLS) over a UDP/IP protocol stack according to IETF RFC 4347. In the handshake both client and server shall be authenticated with an ISO/IEC 9594-8 (X.509v3) certificate.

The exchange of certificates for IPsec, TLS and DTLS shall be according to the public key transport mechanism 3 defined in ISO/IEC 11770-3. The detailed key and certificate exchange is defined in ISO/TS 19299.

NOTE 1 IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite. Any application data traffic is secured by an IPsec connection.

NOTE 2 TLS is implemented on top of the Transport Layer protocol, encapsulating the application-specific protocols such as HTTP, FTP, SMTP, etc. This encapsulated application protocols are then referred to as HTTPS, FTPS, SMTPS, etc.

NOTE 3 The protocol extension defined in RFC 5746 protects against a possible man-in-the-middle attack using handshake renegotiation in the basic TLS protocol.

7.3 Supporting functions

7.3.1 Communication services

This International Standard does not prescribe any sequence of APDUs or timing. Within the limits of the provisions of Clause 6, the communicating parties are free to select their communications means.

7.3.2 Authenticators

Authentication of APDUs is done by using the infoExchangeAuthenticator attribute in the APDU. It shall be computed over the attributes of the header and the payload of the APDU itself. The ApduAuthenticator structure is explained in Table 47 and formally defined in Annex A.

Table 47 — ApduAuthenticator fields

Field name	Data type	m/o
authenticatorEfc	AuthenticatorEfc	m
ackAuthenticatorEfc	AuthenticatorEfc	o

The authenticatorEfc field shall ensure APDU authentication and proof of origin.

The optional ackAuthenticatorEfc field shall ensure proof of delivery. The receiver of an APDU requiring non-repudiation with proof of delivery shall send an AckAdu with an acknowledgement signature based on the whole XER encoded InfoExchange. The AckAdu shall contain this signature and the signature of the Ack ADU itself. The originator shall store this acknowledgement signature together with the originally sent InfoExchange structure for non-repudiation with proof of delivery.

A possibly negative reason code may indicate that a delivered APDU could not be used correctly.

Both fields are of the same type AuthenticatorEfc, which is described in Table 48 and formally defined in Annex A.

Table 48 — AuthenticatorEfc fields

Field name	Data type	m/o
tbsAuthenticatorEfc	TbsAuthenticatorEfc	m
signatureAlgorithm	AlgorithmIdentifier	o
signatureValue	OCTET STRING	m

The signature for the signatureValue field shall be formed over the hashed transformation of tbsAuthenticatorEfc according to the signature algorithm. The hashed transformation for the signature shall be calculated based on the XER encoded tbsAuthenticatorEfc structure.

The signatureAlgorithm is a SEQUENCE of the algorithm object identifier and all parameters required by the algorithm.

The tbsAuthenticatorEfc structure is a sequence of fields, which is listed in Table 49 and formally defined in Annex A.

Table 49 — tbsAuthenticator fields

Field name	Data type	m/o
version	INTEGER	m
signatureAlgorithm	AlgorithmIdentifier	m
issuer	Name	m
serialNumber	CertificateSerialNumber	m
apduContentDigest	ApduContentDigest	m
signatureDate	GeneralizedTime	m
certificates	SEQUENCE OF Certificate	o
signingCertificate	SigningCertificateV2	o

The values of the two data elements **signatureAlgorithm**, i.e. the optional and unprotected one in AuthenticatorEfc if present and the protected one in tbsAuthenticatorEfc, shall be identical.

The **serialNumber** and **issuer** shall indicate the originator of the signature. The respective data types are imported from ISO/IEC 9594-8.

The **apduContentDigest** field shall contain the hash calculated of the XER encoded payload of the APDU authenticator (in this case the InfoExchangeContent) including the hash algorithm identifier.

The **signatureDate** field indicates shall contain the time stamp the signature was generated.

The **certificates** field may optionally be used to attach all required certificates for signature verification. In most cases there will be only one certificate.

The **signingCertificate** field may optionally be used to avoid an attack based on reissuing a certificate by a key certification authority with the same identification but other certificate attributes, i.e. the certificates to sign the APDU and to verify it are not identical. Therefore, the signingCertificate data type shall contain the hashes of the original signing certificate (see IETF RFC 2634 and RFC 5035).

The private key used for generating the signature shall belong to the signing public key certificate of the entity or to the certificate in the field certificates, if present.

The signature and hash algorithms are defined in 7.3.3.

7.3.3 Signature and hash algorithms

Signatures shall be computed by using RSA according to ISO/IEC 14888-2:2008, Clause 6.

The minimum RSA key lengths for certificate signatures shall be 4096 Bit and for entity signatures 2048 Bit.

The padding algorithm used in this technical specification shall be padding method 2 defined in ISO/IEC 9797-1:2011, 3.3. For APDUs authenticated by a signature, the padding algorithm defined in this clause shall be used, but the padding shall not be added to the APDU for transmission.

The SHA-256 algorithm as defined in ISO/IEC 10118-3 shall be supported. In addition, SHA-1 as defined in ISO/IEC 10118-3 may be supported.

NOTE SHA-1 provides support for legacy systems. According to NIST Special Publication 800–131A[17], SHA-1 is deprecated from 2011 through 2013 and disallowed after 2013 for the use in digital signatures. According to the same NIST Special Publication, the use of SHA-256 (and SHA-224) for digital signatures is acceptable.

7.3.4 Keys encryption

Symmetric keys shall be exchanged by using a secured and authenticated channel. If provisions of this International Standard are used for key exchange, the TrustObjectADU shall be used, which will be transferred using the methods listed in 7.2. Otherwise, other secure transfers can be possible, such as, for example, a transfer using a courier, registered mail, etc., where the keys are stored on a memory device. The keys shall be additionally encrypted unless a cryptographic module with active key access authentication for the exchange is used.

The TrustObjectADU shall contain the keys and shall be signed by its originator. The RSA algorithm with the RSA encoding method (REM1) using a 2048-bits key, according to ISO/IEC 18033-2, shall be used for key encryption.

NOTE The required key encryption algorithm (REM1) according to ISO/IEC 18033-2 is equal to RSAES-OAEP in PKCS#1 v2.1 when KDF1 is applied.

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.

The actual ASN.1 module is contained in the attached file "ISO12855(2015)EfcInfoExchangev4.asn."

The following considerations shall be taken into account when using the attached ASN.1 module:

— time values are generally defined by means of the GeneralizedTime type. Unless bilateral agreements state otherwise, when a form of GeneralizedTime is used that does not specify the time zone, the expressed time is to be meant as GMT;

— licence plate numbers are defined by the LPN data type. In that data type, the value of 13 for the alphabetIndicator field shall indicate that UTF-8 encoding has been used;

— the UserID data type is defined as a sequence of optional fields. At least one of the fields shall be specified.

Annex B
(normative)

Implementation Conformance Statement (ICS)

B.1 Guidance for completing the ICS proforma

B.1.1 Purpose and structure

The purpose of this ICS 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 ICS proforma is subdivided into clauses for the following categories of information:

— guidance for completing the ICS proforma;

— identification of the implementation;

— identification of the protocol;

— global statement of conformance;

— ICS proforma tables.

B.1.2 Abbreviations and conventions

B.1.2.1 General

The ICS proforma contained in this annex comprises information in tabular form in accordance with the guidelines presented in ISO/IEC 9646-7.

B.1.2.2 Item column

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

B.1.2.3 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?".

B.1.2.4 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 that identifies a unique group of related optional items and the logic of their selection, which is defined immediately following the table;
c.i	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.

B.1.2.5 Reference column

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

B.1.2.6 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 or n/a	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.

B.1.2.7 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 1 5 .. 20

— list of values: < value1 > < value2 > , ..., < valueN >

EXAMPLE 2 2, 4, 6, 8, 9

EXAMPLE 3 ‘1101’B, ‘1011’B, ‘1111’B

EXAMPLE 4 ‘0A’H, ‘34’H, ‘2F’H

— list of named values: < name1 > (<val1 >), < name2 > (<val2 >), ..., < nameN > (<valN >)

EXAMPLE 5 reject(1), accept(2)

— length: size (<min size > .. < max size >)

EXAMPLE 6 size (1 .. 8)

B.1.2.8 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.

B.1.2.9 References to items

For each possible item answer (answer in the support column) within the ICS proforma, a unique reference exists, and is 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.

B.1.2.10 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 ICS proforma

The supplier of the implementation shall complete the ICS 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 ICS shall be named as the contact person.

Date of the statement:	
Implementation Under Test (IUT) identification	
IUT name:	
IUT version:	
System Under Test (SUT) identification	
SUT name:	
Hardware configuration:	
Operating system:	
Product supplier	
Name:	
Address:	
Telephone number:	
Facsimile number:	
Email address:	
Additional information:	
Applicant (if different from product supplier)	
Name:	
Address:	
Telephone number:	
Facsimile number:	
Email address:	
Additional information:	
ICS contact person *Appointed contact person in case of any queries concerning the content of the ICS*	
Name:	
Telephone number:	
Facsimile number:	
Email address:	
Additional information:	

B.3 Identification of the protocol

This ICS proforma applies to

ISO 12855:2015, *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 ICS, with an explanation of why the implementation is non-conforming, on pages attached to the ICS proforma.

B.5 ICS proforma tables

The ICS proformas are given in Tables B.1 to B.60.

B.5.1 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	
NOTE	o.1: it is mandatory to support at least one of these options.			

B.5.2 Functionalities

Table B.2 — Functionalities

Item	Supported types	Reference	Status	Support (Y/N)
1	Basic mandatory protocol mechanisms	5.2.3	m	
2	Status	5.2.3.5	o	
3	Exchange trust objects	5.2.4	o	
4	Exchange EFC context data and contractual information	5.2.5	o	
5	Manage exception lists	5.2.6	o	
6	Report toll declarations	5.2.7	o	
7	Report billing details	5.2.8	o	
8	Payment settlement	5.2.9	o	
9	Exchange enforcement data	5.2.10	o	
10	Exchange quality assurance parameters	5.2.11	o	

B.5.3 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	aidIdentifier	6.2.2	m	
2	apduOriginator	6.2.2	m	
3	informationSenderID	6.2.2	m	
4	informationRecipientID	6.2.2	m	
5	apduIdentifier	6.2.2	m	
6	previousApduId	6.2.2	o	
7	nextApduId	6.2.2	o	
8	inResponseToApduId	6.2.2	o	
9	apduDate	6.2.2	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	o	
4	trustObjectADU	6.1,6.6	o		6.1,6.6	o	
5	efcContextDataADU	6.1,6.7	c.1		6.1,6.7	c.2	
6	contractIssuerListADU	6.1, 6.8	c.2		6.1, 6.8	c.1	
7	exceptionListADU	6.1,6.9	c.2		6.1,6.9	c.1	
8	reportAbnormalOBEADU	6.1,6.10	c.1		6.1,6.10	c.2	
9	tollDeclarationADU	6.1,6.11	c.2		6.1,6.11	c.1	
10	billingDetailsADU	6.1,6.12	c.1		6.1,6.12	c.2	
11	paymentClaimADU	6.1,6.13	c.1		6.1,6.13	c.2	
12	paymentAnnouncementADU	6.1, 6.14	c.2		6.1, 6.14	c.1	
13	provideUserDetailsADU	6.1,6.15	c.2		6.1,6.15	c.1	
14	reportCCCEventADU	6.1,6.16	c.1		6.1,6.16	c.2	
15	provideUserListADU	6.1,6.17	c.2		6.1, 6.17	c.1	
16	reportQAADU	6.1,6.18	o		6.1,6.18	o	

Table B.6 — genericRequest fields

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

Table B.7 — exceptionListRequest fields

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

Table B.8 — trustObjectRequest fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	requestedADUType	6.3	m		6.3	m	
2	requestedTrustObject	6.3	m		6.3	m	

Table B.9 — requestedTrustObject fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	trustObjectId	6.3	o.1		6.3	m	
2	trustObjectSpec	6.3	o.1		6.3	m	
NOTE o.1: It is mandatory to support at least one of these options.							

Table B.10 — trustObjectSpec fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	typeOfTrustObject	6.3	m		6.3	m	
2	purposesOfTrustObject	6.3	o		6.3	m	

Table B.11 — tollDeclarationRequest fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	requestedADUType	6.3	m		6.3	m	
2	userId	6.3	o		6.3	m	
3	startTime	6.3	o		6.3	m	
4	endTime	6.3	o		6.3	m	

Table B.12 — userDetailsRequest fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	requestedADUType	6.3	m		6.3	m	
2	userId	6.3	m		6.3	m	
3	listOfParametersRequested	6.3	o		6.3	m	
4	userDetailsRequestReason	6.3	o		6.3	m	
5	userInfoValidityPeriod	6.3	o		6.3	m	

Table B.13 — cccEventRequest fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	requestedADUType	6.3	m		6.3	m	
2	userId	6.3	o		6.3	m	
3	startTime	6.3	o		6.3	m	
4	endTime	6.3	o		6.3	m	

Table B.14 — userListRequest fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	requestedADUType	6.3	m		6.3	m	
2	userIdRequestType	6.3	m		6.3	m	
3	userId	6.3	o		6.3	m	
4	userIdRequestTime	6.3	o		6.3	m	

Table B.15 — Ack ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	apdudentifier	6.4	m		6.4	m	
2	explicitlyAckedAdus	6.4	o		6.4	m	
3	apduAckCode	6.4	m		6.4	m	
4	apduAckText	6.4	o		6.4	m	
5	issues	6.4	o		6.4	m	

Table B.16 — issues fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	issueADUStruct	6.4	m		6.4	m	
2	issueLocation	6.4	o		6.4	m	
3	issueContent	6.4	o		6.4	m	
4	issueCode	6.4	m		6.4	m	
5	issueText	6.4	o		6.4	m	
6	issueUsers	6.4	o		6.4	m	

Table B.17 — Status ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	generalStatusCode	6.5	c.1		6.5	c.1	
2	apduStatusCode	6.5	c.2		6.5	c.1	
NOTE c.1: IF (Table B.5/3) THEN m ELSE n/a. c.2: IF (Table B.5/3) THEN o ELSE n/a.							

Table B.18 — 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	trustObject	6.6	c.1		6.6	c.1	
NOTE c.1: IF (Table B.5/4) THEN m ELSE n/a. c.2: IF (Table B.5/4) THEN o ELSE n/a.							

Table B.19 — EfcContextDataADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	entityOverview	6.7	o		6.7	m	
2	gnssContext	6.7	o.1		6.7	m	
3	dsrcContext	6.7	o.1		6.7	m	
4	dsrcClosedContext	6.7	o.1		6.7	m	
NOTE o.1: It is mandatory to support at least one of these options.							

Table B.20 — EntityOverview fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	entityId	6.7	c.1		6.7	c.1	
2	entityType	6.7	c.1		6.7	c.1	
3	relatedEntityID	6.7	c.2		6.7	c.1	
4	entityClass	6.7	c.2		6.7	c.1	
5	nameLine1	6.7	c.2		6.7	c.1	
6	nameLine2	6.7	c.2		6.7	c.1	
7	addressLine1	6.7	c.2		6.7	c.1	
8	addressLine1	6.7	c.2		6.7	c.1	
9	poBox	6.7	c.2		6.7	c.1	
10	zip	6.7	c.1		6.7	c.1	
11	city	6.7	c.1		6.7	c.1	
12	country	6.7	c.1		6.7	c.1	
13	countryCode	6.7	c.1		6.7	c.1	
14	description	6.7	c.2		6.7	c.1	
15	mainContact	6.7	c.1		6.7	c.1	
16	customerServiceContact	6.7	c.2		6.7	c.1	
17	itContact	6.7	c.1		6.7	c.1	
18	operationalContact	6.7	c.1		6.7	c.1	
19	commercialContact	6.7	c.1		6.7	c.1	
20	website	6.7	c.2		6.7	c.1	
21	companyRegistrationNumber	6.7	c.2		6.7	c.1	
22	established	6.7	c.2		6.7	c.1	
23	bankDetails	6.7	c.1		6.7	c.1	
NOTE c.1: IF (Table B.19/1) THEN m ELSE n/a. c.2: IF (Table B.19/1) THEN o ELSE n/a.							

Table B.21 — GnssContext fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	contextInterrelations	6.7	c.2		6.7	c.3	
2	regimeContextData.contextId	6.7	c.1		6.7	c.3	
3	regimeContextData.iso175753ADU	6.7	c.1		6.7	c.3	
4	regimeContextData.feeModifiers	6.7	c.2		6.7	c.3	
NOTE c.1: IF (Table B.19/2) THEN m ELSE n/a. c.2: IF (Table B.19/2) THEN o ELSE n/a. c.3: IF (Table B.5/5) THEN m ELSE n/a.							

Table B.22 — DsrcContext fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	iso175753ADU	6.7	c.1		6.7	c.3	
2	feeModifiers	6.7	c.2		6.7	c.3	
NOTE c.1: IF (Table B.19/3) THEN m ELSE n/a. c.2: IF (Table B.19/3) THEN o ELSE n/a. c.3: IF (Table B.5/5) THEN m ELSE n/a.							

Table B.23 — DsrcClosedContext fields

Item	Field	Sending			Receiving		
		Refer-ence	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	closedContextDefinition.closedContext-Type	6.7	c.1		6.7	c.3	
2	closedContextDefinition.closedSystem	6.7	c.1		6.7	c.3	
3	feeModifiers	6.7	c.2		6.7	c.3	
NOTE c.1: IF (Table B.19/4) THEN m ELSE n/a. c.2: IF (Table B.19/4) THEN o ELSE n/a. c.3: IF (Table B.5/5) THEN m ELSE n/a.							

Table B.24 — ClosedSystem fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	tollContextOverview	6.7	o.1		6.7	m	
2	tariffClassDefinition	6.7	o.1		6.7	m	
3	localVehicleClassDefinition	6.7	o.1		6.7	m	
4	timeClassDefinition	6.7	o.1		6.7	m	
5	userClassDefinition	6.7	o.1		6.7	m	
6	feeDefinition	6.7	o.1		6.7	m	
NOTE o.1: It is mandatory to support at least one of these options.							

Table B.25 — FeeDefinition fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	actualPath	6.7	o.1		6.7	m	
2	predefinedPath	6.7	o.1		6.7	m	
NOTE o.1: It is mandatory to support exactly one of these options.							

Table B.26 — ActualPath fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	tariffTable	6.7	o.1		6.7	m	
2	closedContextLayout	6.7	o.1		6.7	m	
NOTE o.1: It is mandatory to support exactly one of these options.							

Table B.27 — TariffTable fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	applicablePartitions	6.7	c.1		6.7	c.1	
2	tariffs	6.7	c.1		6.7	c.1	
3	standardCurrency	6.7	c.1		6.7	c.1	
4	typeOfFee	6.7	c.2		6.7	c.1	
5	tariffTableVersion	6.7	c.1		6.7	c.1	
NOTE 1 c.1: IF (Table B.26/1) THEN m ELSE n/a. c.2: IF (Table B.26/1) THEN o ELSE n/a. NOTE 2 Type imported from ISO 17575-3.							

Table B.28 — ClosedContextLayout fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	tollOperators	6.7	o		6.7	m	
2	dsrcChargeObjects	6.7	m		6.7	m	
3	internalPoints	6.7	o		6.7	m	
4	interconnectionPoints	6.7	o		6.7	m	
5	closedSections	6.7	o		6.7	m	
6	tollContextLayoutVersion	6.7	m		6.7	m	
7	tollContextAuthenticator	6.7	o		6.7	m	

Table B.29 — PredefinedPath fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	tollDefinition	6.7	o.1		6.7	m	
2	pathDefinition	6.7	o.1		6.7	m	
NOTE o.1: It is mandatory to support at least one of these options.							

Table B.30 — TollDefinition fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	tollTable	6.7	m		6.7	m	
2	tollDefinitionVersion	6.7	m		6.7	m	
3	tollDefinitionAuthenticator	6.7	m		6.7	m	

Table B.31 — PathDefinition fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	pathTable	6.7	m		6.7	m	
2	pathDefinitionVersion	6.7	m		6.7	m	
3	pathDefinitionAuthenticator	6.7	m		6.7	m	

Table B.32 — ContractIssuerList ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	efcContextMark	6.8	c.1		6.8	c.1	
2	equipmentClass	6.8	c.1		6.8	c.1	
3	manufacturerID	6.8	c.1		6.8	c.1	
4	uniquePartOfPAN	6.8	c.1		6.8	c.1	
5	typeOfEFCApplication	6.8	c.2		6.8	c.1	
6	securityLevel	6.8	c.1		6.8	c.1	
7	acCrKeyReference	6.8	c.1		6.8	c.1	
8	authKeyReference	6.8	c.1		6.8	c.1	
9	validFrom	6.8	c.1		6.8	c.1	
NOTE 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.33 — ExceptionList ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	exceptionListVersion	6.9	c.1		6.9	c.1	
2	exceptionListType	6.9	c.1		6.9	c.1	
3	exceptionValidityStart	6.9	c.2		6.9	c.1	
4	exceptionValidityEnd	6.9	c.2		6.9	c.1	
5	exceptionListEntries	6.9	c.1		6.9	c.1	
NOTE c.1: IF (Table B.5/7) THEN m ELSE n/a. c.2: IF (Table B.5/7) THEN o ELSE n/a.							

Table B.34 — ExceptionListEntry 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	statusType	6.9	c.1		6.9	c.1	
3	reasonCode	6.9	c.1		6.9	c.1	
4	entryValidityStart	6.9	c.2		6.9	c.1	
5	entryValidityEnd	6.9	c.2		6.9	c.1	
6	vehicleParameters	6.9	c.2		6.9	c.1	
7	actionRequested	6.9	c.2		6.9	c.1	
8	efcContextMark	6.9	c.2		6.9	c.1	
9	vatId	6.9	c.2		6.9	c.1	
NOTE c.1: IF (Table B.5/7) THEN m ELSE n/a. c.2: IF (Table B.5/7) THEN o ELSE n/a.							

Table B.35 — ReportAbnormalOBE ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	userId	6.10	c.1		6.10	c.1	
2	dateandTime	6.10	c.1		6.10	c.1	
3	efcContextMark	6.10	c.2		6.10	c.1	
4	abnormalOBEReasonCode	6.10	c.1		6.10	c.1	
NOTE 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.36 — 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.chargeReport	6.11	c.2 o.1		6.11	c.1	
3	gnssTollDeclaration.authenticated ChargeReport	6.11	c.2 o.1		6.11	c.1	
4	actionCode	6.11	c.2		6.11	c.1	
NOTE c.1: IF (Table B.5/9) THEN m ELSE n/a. c.2: IF (Table B.5/9) THEN o ELSE n/a. o.1: It is mandatory to support at least one of these options.							

Table B.37 — BillingDetails ADU fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	billingDetailsId.issuerId	6.12	c.1		6.12	c.1	
2	contextId.billingDetailsNum	6.12	c.1		6.12	c.1	
3	contextIddateOfService	6.12	c.2		6.12	c.1	
4	tollContext	6.12	c.1		6.12	c.1	
5	userId	6.12	c.2		6.12	c.1	
6	relatedBillingDetails	6.12	c.2		6.12	c.1	
7	period	6.12	c.2		6.12	c.1	
8	billingDetailAmount	6.12	c.1		6.12	c.1	
9	usageDetails	6.12	c.2		6.12	c.1	
10	refTollDeclaration	6.12	c.2		6.12	c.1	
11	associatedEventData	6.12	c.2		6.12	c.1	
12	actionCode	6.12	c.2		6.12	c.1	
13	paymentReference	6.12	c.2		6.12	c.1	

NOTE
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.38 — usageDetails fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	contextName	6.12	c.1		6.12	c.1	
2	appliedUserClass	6.12	c.1		6.12	c.1	
3	declaredVehicleClass	6.12	c.2		6.12	c.1	
4	appliedTimeClass	6.12	c.1		6.12	c.1	
5	costCenter	6.12	c.2		6.12	c.1	
6	usageList	6.12	c.1		6.12	c.1	

NOTE
c.1: IF (Table B.37/9) THEN m ELSE n/a.
c.2: IF (Table B.37/9) THEN o ELSE n/a.

Table B.39 — AssociatedEventData fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	cccRecord	6.12	o.1, c.1		6.12	c.2	
2	imageRecord	6.12	o.1, c.1		6.12	c.2	
3	anprRecord	6.12	o.1, c.1		6.12	c.2	
4	classificationRecord	6.12	o.1, c.1		6.12	c.2	
5	operatorRecord	6.12	o.1, c.1		6.12	c.2	
6	dsrcData	6.12	o.1, c.1		6.12	c.2	

NOTE

o.1: IF (Table B.37/11) THEN it is mandatory to support at least one of these options.

c.1: IF (Table B.37/11) THEN o ELSE n/a.

c.2: IF (Table B.37/11) THEN m ELSE n/a.

Table B.40 — CccRecord fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	timeOfEvent	6.12	c.1		6.12	c.2	
2	locationOfEvent	6.12	c.1		6.12	c.2	
3	cccMessages	6.12	c.2		6.12	c.2	
4	initiatedActions	6.12	c.2		6.12	c.2	

NOTE

c.1: IF (Table B.39/1) THEN o ELSE n/a.

c.2: IF (Table B.39/1) THEN m ELSE n/a.

Table B.41 — ImageRecord fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	imageRecordId	6.12	c.1		6.12	c.1	
2	imageDateTime	6.12	c.1		6.12	c.1	
3	imageCameraId	6.12	c.1		6.12	c.1	
4	imageReference	6.12	c.2		6.12	c.1	
5	imageData	6.12	c.2		6.12	c.1	
6	subRecordAuthenticator	6.12	c.2		6.12	c.1	

NOTE

c.1: IF (Table B.39/2) THEN m ELSE n/a.

c.2: IF (Table B.39/2) THEN o ELSE n/a.

Table B.42 — AnprRecord fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	anprRecordId	6.12	c.1		6.12	c.1	
2	associatedImages	6.12	c.1		6.12	c.1	
3	imageDateTime	6.12	c.1		6.12	c.1	
4	imageCameraId	6.12	c.1		6.12	c.1	
5	determinedVRM	6.12	c.1		6.12	c.1	
6	vehicleDetails	6.12	c.1		6.12	c.1	
7	vehicleDetails	6.12	c.1		6.12	c.1	
8	subRecordAuthenticator	6.12	c.2		6.12	c.1	
NOTE c.1: IF (Table B.39/3) THEN m ELSE n/a. c.2: IF (Table B.39/3) THEN o ELSE n/a.							

Table B.43 — ClassificationRecord fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	classificationRecordID	6.12	c.1		6.12	c.1	
2	derivedLocalClass	6.12	c.1		6.12	c.1	
3	vehicleParameters	6.12	c.1		6.12	c.1	
4	subRecordAuthenticator	6.12	c.2		6.12	c.1	
NOTE c.1: IF (Table B.39/4) THEN m ELSE n/a. c.2: IF (Table B.39/4) THEN o ELSE n/a.							

Table B.44 — OperatorRecord fields

Item	Field	Sending			Receiving		
		Reference	Status	Support (Y/N)	Reference	Status	Support (Y/N)
1	classificationRecordID	6.12	c.1		6.12	c.1	
2	derivedLocalClass	6.12	c.1		6.12	c.1	
3	vehicleParameters	6.12	c.1		6.12	c.1	
4	subRecordAuthenticator	6.12	c.2		6.12	c.1	
NOTE c.1: IF (Table B.39/5) THEN m ELSE n/a. c.2: IF (Table B.39/5) THEN o ELSE n/a.							