

DIN EN ISO 12855



ICS 03.220.20; 35.240.60

EntwurfEinsprüche bis 2021-02-11
Vorgesehen als Ersatz für
DIN EN ISO 12855:2016-04**Elektronische Gebührenerhebung
Informationsaustausch zwischen Dienstleistern und
Gebühreneinzugsunternehmen (ISO/DIS 12855:2020);
Englische Fassung prEN ISO 12855:2020**

Electronic fee collection –
Information exchange between service provision and toll charging (ISO/DIS 12855:2020);
English Version prEN ISO 12855:2020

Perception du télépéage –
Échange d'informations entre la prestation de service et la perception du péage
(ISO/DIS 12855:2020);
Version anglaise prEN ISO 12855:2020

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2020-12-11 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs
besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an naautomobil@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an den DIN-Normenausschuss Automobiltechnik (NAAutomobil), 10004 Berlin, Postfach 8 04 62 oder Behrenstr. 35, 10117 Berlin.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten
Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 170 Seiten

DIN-Normenausschuss Automobiltechnik (NAAutomobil)

Nationales Vorwort

Dieses Dokument (EN ISO 12855:2020) wurde vom Technischen Komitee ISO/TC 204 „Intelligent Transport Systems“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 278 „Intelligente Verkehrssysteme“, dessen Sekretariat von NEN (Niederlande) gehalten wird, erarbeitet.

Das zuständige deutsche Normungsgremium ist der Gemeinschaftsarbeitsausschuss NA 052-00-71 GA „Gemeinschaftsarbeitsausschuss NAAutomobil/DKE: Intelligente Verkehrssysteme“ des DIN-Normenausschusses Automobiltechnik (NAAutomobil).

Dieses Dokument ist derzeit zur parallelen Abstimmung vorgelegt.

Da Mautsysteme in Europa weit verbreitet sind, müssen Vorkehrungen für Anwender getroffen werden, die sich in unterschiedlichen Mautgebieten bewegen. Ziel sollte es sein, dass ein Kunde mit nur einem Mautvertrag eines beliebigen Anbieters sich in unterschiedlichen Mautgebieten bewegen kann. Dort, wo für die Fahrzeuge eine fahrzeuginterne Gebühreneinheit benötigt wird, ist es notwendig, dass diese interoperabel mit den verschiedenen Mautsystemen ist. In Europa wurde bereits die Gesetzgebung zur Interoperabilität der Gebührenerhebung mit der Direktive 2004/52 angepasst.

Diese Norm identifiziert und spezifiziert gemäß den generischen Anforderungen in ISO 17573 notwendige Botschaften für den Austausch zwischen Maut-Dienstanbieter und Mauterheber. Um diese Schnittstellen zu definieren, werden in dieser Norm entsprechende Informationen aus den Unternehmen zur Mautumgebung und deren Interaktionen verwendet.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Änderungen

Gegenüber DIN EN ISO 12855:2016-04 wurden folgende Änderungen vorgenommen:

- a) Anwendungsdateneinheiten wurden ergänzt;
- b) Anpassung der ASN.1-Datentypen basierend auf den aktuellen Fassungen von DIN EN ISO 14906 und DIN EN ISO 17575 (alle Teile);
- c) alle von DIN EN ISO 17575 importierten ASN.1 Datentypen wurden gelöscht und dafür entsprechende Festlegungen ergänzt;
- d) ASN.1 Modul wurde in zwei Module aufgeteilt;
- e) allgemeine Datentypmodule wurden in ISO 17573-3 verschoben;
- f) allgemeine redaktionelle und strukturelle Anpassungen.

Contents	Page
FOREWORD.....	4
INTRODUCTION.....	5
1 SCOPE.....	1
2 NORMATIVE REFERENCES.....	2
3 TERMS AND DEFINITIONS.....	3
4 SYMBOLS AND ABBREVIATED TERMS.....	3
5 ARCHITECTURAL CONCEPTS AND INFORMATION EXCHANGES.....	5
5.1 MAIN ROLES IN THE TOLL CHARGING ENVIRONMENT.....	5
5.2 INFORMATION EXCHANGE BETWEEN TOLL CHARGING AND PROVISION.....	5
5.2.1 General.....	5
5.2.2 Basic interaction protocol.....	7
5.2.3 Basic protocol mechanisms.....	7
5.2.4 Exchange trust objects functionality.....	8
5.2.5 Provide EFC context data.....	8
5.2.6 Provide contract issuer information functionality.....	9
5.2.7 Manage exception list functionality.....	9
5.2.8 Report toll declarations functionality.....	10
5.2.9 Report billing details functionality.....	10
5.2.10 Payment settlement functionality.....	11
5.2.11 Exchange enforcement data functionality.....	12
5.2.12 Process user complaints functionality.....	14
5.2.13 Exchange quality assurance parameters functionality.....	14
5.2.14 Provide media exception list functionality.....	15
5.2.15 Provide media settlement data functionality.....	15
6 COMPUTATIONAL SPECIFICATION.....	15
6.1 OVERVIEW.....	15
6.2 APPLICATION PROTOCOL DATA UNITS.....	19
6.2.1 General.....	19
6.2.2 Application Protocol Control Information.....	20
6.2.3 Application Data Units.....	21
6.2.4 ADU identification.....	22
6.2.5 ADU action code.....	23
6.2.6 User identification.....	23
6.3 REQUESTADU DATA STRUCTURE.....	24
6.4 ACKADU DATA STRUCTURE.....	28
6.5 STATUSADU DATA STRUCTURE.....	36
6.6 TRUSTOBJECTADU DATA STRUCTURE.....	36
6.7 EFCCONTEXTDATAADU DATA STRUCTURE.....	43
6.7.1 General.....	43
6.7.2 The gnss context.....	44
6.7.3 The dsrc context.....	44
6.7.4 The dsrc closed context.....	45
6.7.5 The EntityOverview data structure.....	54
6.7.6 The EfcContextData data structure.....	57
6.7.7 FeeModifier data type.....	95

6.8	CONTRACTISSUERLISTADU DATA STRUCTURE	97
6.9	EXCEPTIONLISTADU DATA STRUCTURE	99
6.10	REPORTABNORMALOBEADU DATA STRUCTURE	101
6.11	TOLL DECLARATIONADU DATA STRUCTURE	102
6.12	BILLINGDETAILSADU DATA STRUCTURE	110
6.12.1	<i>UsageList data type</i>	112
6.12.2	<i>AssociatedEventData data type</i>	121
6.13	PAYMENTCLAIMADU DATA STRUCTURE	127
6.14	PAYMENTANNOUNCEMENT ADU DATA STRUCTURE	129
6.15	PROVIDEUSERDETAILSADU DATA STRUCTURE	131
6.16	REPORTCCCEVENTADU DATA STRUCTURE	135
6.17	PROVIDEUSERIDLISTADU DATA STRUCTURE	136
6.18	REPORT QA DATA STRUCTURE	137
6.19	USER COMPLAINT DATA STRUCTURE	138
6.20	USER COMPLAINT RESPONSE DATA STRUCTURE	139
6.21	MEDIA EXCEPTION LIST DATA STRUCTURE	141
6.22	MEDIA SETTLEMENT DATA STRUCTURE	142
7	TRANSFER MECHANISMS AND SUPPORTING FUNCTIONS	144
7.1	TRANSFER MECHANISMS	144
7.2	SECURE COMMUNICATION CHANNEL	144
7.3	SUPPORTING FUNCTIONS	145
7.3.1	<i>Communication services</i>	145
7.3.2	<i>Authenticators</i>	145
7.3.3	<i>Signature and hash algorithms</i>	146
7.3.4	<i>Keys encryption</i>	147
	ANNEX A (NORMATIVE) DATA TYPE SPECIFICATIONS	148
	ANNEX B (INFORMATIVE) EXAMPLE ENFORCEMENT PROCESS APPLYING STANDARDIZED APDU EXCHANGES	149
	ANNEX C (INFORMATIVE) EXAMPLE OF DATA FLOWS IN A TOLL DOMAIN	154
	ANNEX D (INFORMATIVE) EXAMPLE OF ROUNDING DIFFERENCES	158
	BIBLIOGRAPHY	162

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 third edition cancels and replaces the second edition (ISO 12855:2015), which has been technically revised. The following major 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 ISO 14906, and ISO 17575 (all parts);
- removing all imported ASN.1 data types from ISO 17575 and creating corresponding definitions;
- splitting the ASN.1 module in two modules: one that contains 12855 specific definitions, and a second one that contains data type definitions that are common to other standards in the EFC domain.;
- move the common data types module into part 3 of ISO 17573;
- 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 2019/520 and related Commission delegated regulation 2020/2003 and Commission implementing regulation 2020/204 in the Bibliography). 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 17573-1 is the basis for all standards that relate to tolling systems in the toll domain. With respect to ISO 17573-1, this document

- adopts its definitions of terms and concepts and basic system functionalities and structure,
- uses its terminology, and
- specifies the interfaces therein identified.

ISO 17573-1 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.

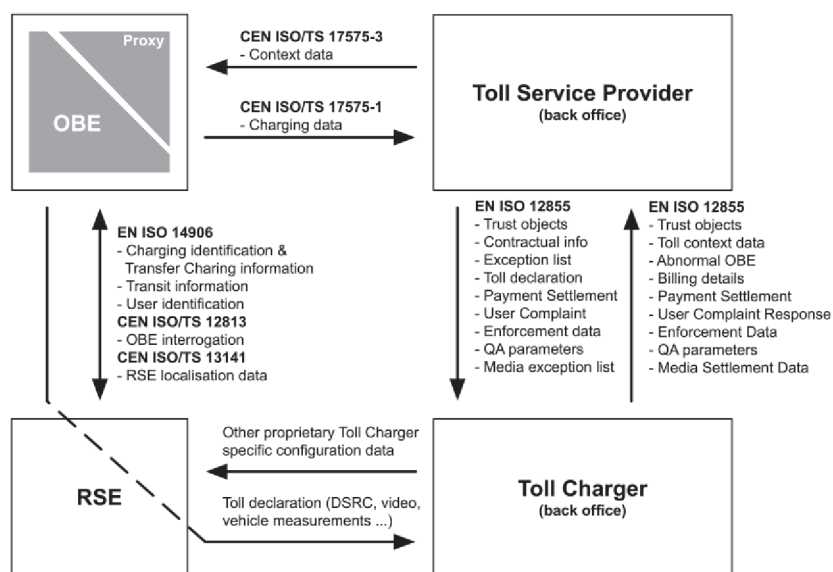


Figure 1 — Scope of EFC related standards

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 calculations are 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 document 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 document is based is defined in ISO 17573-1.

This document 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 already cited European EFC Directive and related Regulation and Implementing acts, also calls for the definition of an interoperable EFC service. It should be noted CEN/TS 16986 specifies interoperable application profiles (IAP) applicable based on this document. These profiles define a specific coherent set of transactions, triggers, conditions, data elements, transfer mechanisms and supporting functions for an interoperable exchange of data between the central equipment of TCs and TSPs. CEN/TS 16986 is consistent with and is intended to provide support for the technical specification of the EETS.

This document 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 17573-1. To specify these interfaces, this document uses the enterprise description of the toll environment, and the interactions defined between the named classes of roles, as defined in ISO 17573-1. 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 document 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 B and Annex C), and
- an example of toll rounding (see Annex D).

This document 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 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.

This document is not applicable, among others, 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.

1. ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*
2. ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*
3. ISO/IEC 8825-4, *Information technology — ASN.1 encoding rules: XML Encoding Rules (XER)*
4. ISO/IEC 9594-8:2014, *Information technology — Open Systems Interconnection — The Directory — Part 8: Public-key and attribute certificate frameworks*
5. ISO/IEC 9646-7, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 7: Implementation Conformance Statements*
6. ISO/IEC 9797-2:2011, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*
7. ISO/IEC 10118-3, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*
8. ISO/IEC 11770-3, *Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques*
9. ISO 13616-1, *Financial services - International bank account number (IBAN) — Part 1: Structure of the IBAN*
10. ISO/IEC 14888-2:2008, *Information technology — Security techniques — Digital signatures with appendix — Part 2: Integer factorization based mechanisms*
11. ISO 14906:2011/Amd1:2015, *Electronic fee collection — Application interface definition for dedicated short-range communication*
12. ISO 17573-1, *Electronic fee collection — Systems architecture for vehicle-related tolling*
13. ISO 17575-1:—¹, *Electronic fee collection — Application interface definition for autonomous systems — Part 1: Charging*
14. ISO 17575-3:—¹, *Electronic fee collection — Application interface definition for autonomous systems — Part 3: Context data*
15. ISO/IEC 18033-2:2006, *Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers*

16. ISO 19299:2020, *Electronic fee collection — Security framework*
17. ISO/IEC 9834-8:2005, *Information technology – Procedures for the operation of object identifier registration authorities: Generation of universally unique identifiers and their use in object identifiers*
18. ISO/IEC 21320-1:2015, *Information Technology - Document Container File - Part 1: Core*
19. IETF RFC 2634, *Enhanced Security Services for S/MIME*, June 1999
20. IETF RFC 4347, *Datagram Transport Layer Security*, April 2006
21. IETF RFC 5035, *Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility*, August 2007
22. IETF RFC 5246, *The Transport Layer Security (TLS) Protocol*, August 2008
23. IETF RFC 5746, *Transport Layer Security (TLS) Renegotiation Indication Extension*, February 2010
24. IETF RFC 6040, *Tunnelling of Explicit Congestion Notification*, February 2013
25. IETF RFC 2048, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, November 1996
26. ISO 4217:2015, *Codes for the representation of currencies*
27. ISO/TS 17444-1:2017, *Electronic fee collection — Charging performance — Part 1: Metrics*

3 Terms and definitions

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

In addition, the purpose of this document, the definitions in ISO/TS 17573-2 apply.

4 Symbols and abbreviated terms

ADU	Application data unit
ANPR	Automatic Number Plate Reading
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
CCC	Compliance Check Communication
CRL	Certificate revocation list
DSRC	Dedicated short-range communication
EFC	Electronic Fee Collection
GDF	Geographical Data Files

GNSS	Global Navigation Satellite System
HTTPS	Hyper-Text Transfer Protocol Secure
ICC	Integrated Circuit Card
IEC	International Electrotechnical Commission
ITU	International Telecommunication Union
LAC	Location Augmentation Communication
LPN	Licence Plate Number
OBE	On-Board Equipment
OBU	On-Board Unit
OCSP	Online Certificate Status Protocol
OSI	Open Systems Interconnection
PAN	Personal Account Number
QA	Quality Assurance
RSA	Rivest, Shamir and Adleman
RSE	Roadside Equipment
SLA	Service Level Agreement
SU	Service User
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 document is built upon ISO 17573-1.

ISO 17573-1 defines the four main roles shown in Figure 2.

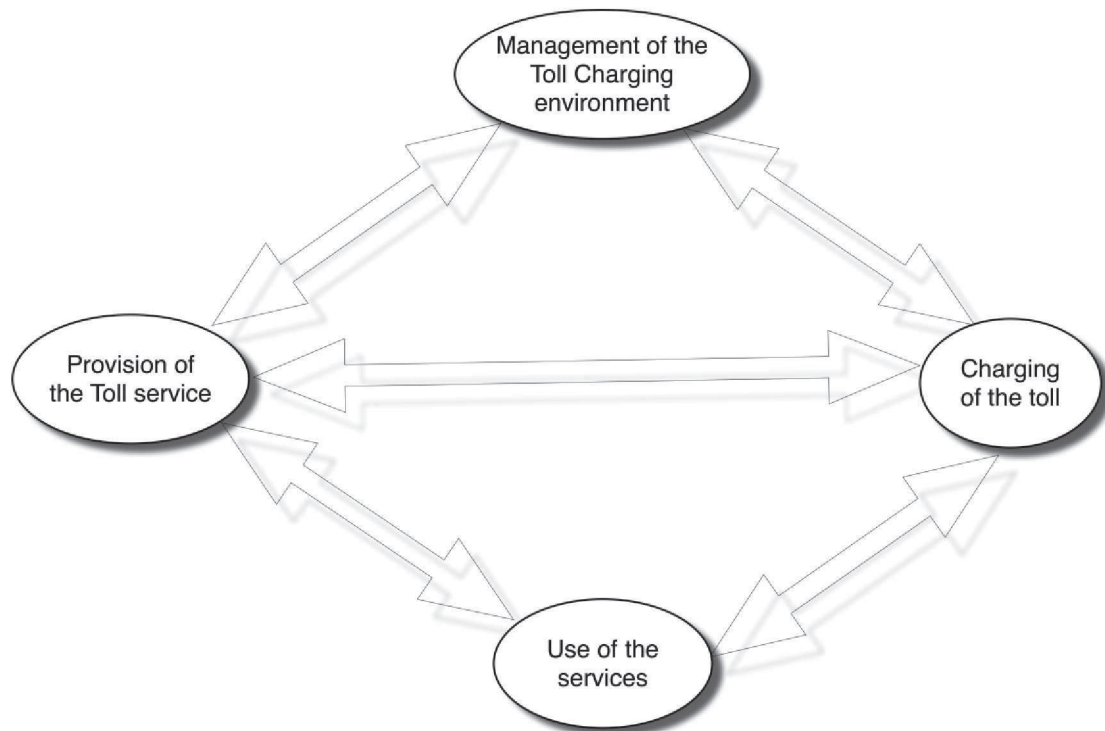


Figure 2 — Roles in the toll charging environment

Information exchanges are agreed upon between toll charger and toll 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.

5.2 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 service definitions in ISO 17573-1. The following Figure 3 gives a general picture of the provided functionalities.

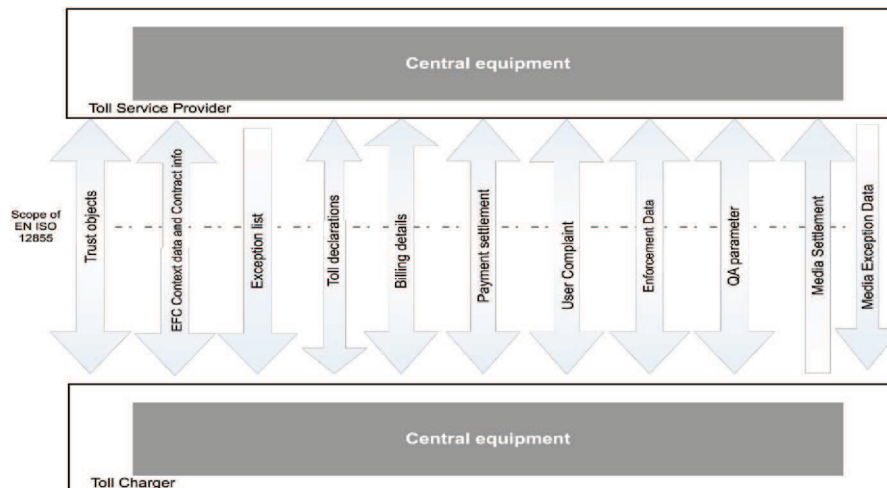


Figure 3 — Functionalities of 12855

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

- basic protocol mechanisms;
- exchange trust objects;
- provide EFC context data;
- provide contract issuer list;
- manage exception list;
- report toll declarations;
- report billing details;
- payment settlement:
 - claim payment for service usage;
 - announce payments.
- exchange enforcement data:
 - exchange of user details;
 - exchange of CCC events;
 - exchange of UserId lists;
- process user complaints:
 - provide a user complaint;
 - response to a user complaint.
- exchange quality assurance parameters.
- provide media exception list;
- provide media settlement data.

This document 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 5.2.15.

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 document defines no provisions for complex transfers (transactions), i.e. sequences of inter-related ADUs that may possibly involve several APDU exchanges. Instead, this document 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 document provides the following basic protocol mechanisms 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 document 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.

5.2.3.2 Identification schema

Each interaction is performed by means of one or more APDU exchanges. Each APDU contains 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 is 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 is used to inform the counterpart that a specific ADU has been received correctly, or errors have been detected, such as for example that the functionality requested by the received ADU is not supported. This document does not specify obligations to respond to any ADU by using the acknowledgement functionality.

5.2.3.5 Status functionality

The status functionality can be used to provide the counterpart with general status information about the interface or to inform it about a status on previously transferred information. It is 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 service “trust object exchange”. The functionality is used whenever an entity, both toll charger and toll service provider, needs to send unsolicitedly new or updated trust objects to its counterpart or when asked by its counterpart to update or resend an already existing trust object.

NOTE Examples of trust objects are: asymmetric public keys, certificates, symmetric keys and certificate revocation lists.

Requiring the counterpart to send trust objects is performed by means of a “request” ADU. The “request” ADU allows requesting already issued trust objects to be resent, as well as requesting newly issued trust objects.

The exchange trust objects functionality provides the “trust objects” ADU to transfer the requested or newly generated trust objects to the counterpart, which has the possibility of confirming or rejecting the received trust objects by using the acknowledgement functionality.

The exchanged trust objects are valid from either the time of their acknowledgment or from the validity starting date indicated in the trust objects themselves.

5.2.5 Provide EFC context data

5.2.5.1 Originating and providing EFC context data functionality

The originating and providing EFC context data functionality is derived from ISO 17573-1 as defined in the EFC system service “adding or modifying a toll regime”.

The originating and providing EFC context data functionality gives a toll charger the possibility to communicate any change of a toll domain or toll regime, including the start of a new toll domain, by issuing an “EFC context data” ADU.

A toll service provider has the possibility of requiring a toll charger to provide information about the toll context data for a toll domain under its responsibility by using a “request” ADU. The toll charger provides the information requested by this “request” ADU by using an “EFC context data” ADU.

The toll service provider has the possibility to confirm reception of an “EFC context data” ADU by using the acknowledgement functionality.

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 and data types are based on the initial work in ISO 17575 and were imported into this document and adapted to the purpose of a technology independent back office exchange. The resulting data structures and data types are therefore no longer compatible with ISO 17575.

5.2.6 Provide contract issuer 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 retrieved from the OBEs with information received by the toll service provider.

5.2.7 Manage exception list functionality

5.2.7.1 General

The manage exception list functionality originates from ISO 17573-1 and is defined in the EFC system service “exceptions handling”.

NOTE 1 This document 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, LPN) can be included in the exception list as agreed between TC and TSP.

5.2.7.2 Exception list entry requested by a toll charger

The manage exception list functionality allows a toll charger to notify a toll service provider about detected violations by a specific service user or wrong technical behaviour by a specific OBE by using a “report abnormal OBE” ADU. The “acknowledgement” functionality is used to confirm reception of a “report abnormal OBE” ADU. subsequently, the inclusion of this OBE in the exception list of the toll service provider may be done.

5.2.7.3 Exception list entry decided by the toll service provider

The manage exception list functionality allows a toll service provider to notify a toll charger about additions, modifications or deletions of items in its exception lists by using the “exception list” ADU.

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.

The toll charger has the possibility to dispute the received exception list by using the acknowledgment functionality and transmitting an error code. In this case the last correct list remains active in the systems of the toll charger, until a newly transmitted list has been positively acknowledged by using the acknowledgement functionality.

5.2.8 Report toll declarations functionality

The report toll declarations functionality originates from ISO 17573-1 and is defined in the EFC system service “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 document.

The toll service provider has the possibility to enrich the gathered charging data that are stored in its central equipment, before sending that data as a “toll declarations” ADU to the toll charger. This possibility 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” ADU provides the toll charger with all information needed to calculate the amount due for the use of a toll domain or to verify the calculation done by the toll service provider, or it may contain only summary data, according to the optional data elements that are implemented.

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 has the possibility to confirm a received “toll declaration” ADU by using by using the acknowledgement functionality.

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 has the possibility to ask the toll service provider for additional information about the provided toll declarations for a specific vehicle or service user by sending 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 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.

The procedure for handling a possible negative “ack” ADU as a response to a “toll declaration” ADU is outside the scope of this document.

5.2.9 Report billing details functionality

Depending on the characteristics of a toll system, a toll charger either acquires the toll declarations directly from the roadside equipment (RSE) it operates (e.g. DSRC-based systems) or receives it from the

toll service provider's central equipment (e.g. autonomous systems). That information is then used for the generation of billing details.

A single billing detail refers to one or more 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 the available toll declarations, the toll charger has the possibility to request it from the central equipment of the toll service provider (see 5.2.8).

The toll charger and toll service provider can agree to aggregate the generated billing details in order to reduce the number of lines to be processed in their bookkeeping systems. The toll charger and the toll service provider need to agree on common aggregation methods for billing details in order to avoid any rounding differences between VAT declarations for the local tax authorities (when applicable). Consequently, rounding rules are not defined in this document.

NOTE 2 Aggregation methods can, e.g., be:

- 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 toll charger or the toll service provider, depending on the respective operational model, have the possibility of transferring billing details by means of a “billing detail” ADU.

NOTE 3 The operational model of the toll charger in a GNSS environment could require generating 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 of a “billing details” ADU might need the possibility to check the completeness and the conformity of the provided billing details for a given service usage. Therefore, the billing details can optionally reference previously exchanged toll declarations so to provide further information.

The recipient of a “billing details” ADU has the possibility to accept or deny it by using the acknowledgement functionality.

5.2.10 Payment settlement functionality

5.2.10.1 Claim payment for service usage

The claim payment for service usage functionality is derived from ISO 17573-1 and is defined in the EFC system service “payment settlement”.

The claim payment phase starts once the billing details have been agreed between the toll charger and the toll service provider (see 5.2.9). 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).

The toll charger uses the claim payment functionality by sending a “payment claim” ADU to the toll service provider.

The “payment claim” ADU gives also the possibility to include additional claims for other services that do not directly derive from the bilaterally agreed billing details (e.g. overdue payment, reimbursement, penalties...).

The “payment claim” is meant to 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 toll service provider has the possibility of accepting or refusing a received “payment claim” ADU by using the acknowledgement functionality.

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

5.2.10.2 Payment announcement

The payment announcement functionality is derived from ISO 17573-1 and is defined in the EFC system service “payment settlement”. This functionality allows a toll service provider to inform a toll charger of a payment that has been done consequently to the use of road services. Differently from the payment claim functionality, the payment announcement functionality is initiated by the toll service provider either unsolicitedly 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 has the possibility of using this functionality by sending a “payment announcement” ADU to the toll charger informing it about an upcoming payment. The information carried in the “payment announcement” ADU may include references to billing details, toll declarations or even single elements of a charge report.

The toll charger has the possibility of accepting or refusing a received “payment announcement” ADU by using the acknowledgement functionality.

5.2.11 Exchange enforcement data functionality

5.2.11.1 General

The exchange enforcement data functionality originates from ISO 17573-1 and is defined in the EFC system service “exceptions detection”.

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 subsequent 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 may be needed to be supported on both sides.

5.2.11.2 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.

5.2.11.3 Provide user details

Provision of details of a given user is a functionality that allows a toll service provider to respond to a “request” ADU requiring user details. The functionality is performed by either sending a “provide user details” ADU that contains details about a contractual relationship with the given user, or by using the acknowledgement functionality with a specific reason code to decline any contractual relationship.

When details on a user are provided, the minimum set of data contained in a “provide user details” ADU includes:

- personal Account Number (PAN);
- contract serial number;
- licence plate number (LPN);
- on-board equipment identification (OBE ID).

Any additional details about the service user depend on what is specified in the “request” ADU and on the actual availability of the requested data.

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

The toll charger has the possibility to acknowledge any received “provide user details” ADU by using the acknowledgement functionality.

5.2.11.4 Retrieve and report CCC event

The retrieve and report CCC event functionality allows a toll charger either unsolicitedly after receiving a “request ADU” from the toll service provider to deliver the details of any CCC events for a specific service user and/or a specific period of time.

This service is realised by issuing a “report CCC event” ADU.

The toll service provider has the possibility to acknowledge any received “report CCC event” ADU by using the acknowledgement functionality.

5.2.11.5 List of users

The list of users functionality provides support to the toll charger in its enforcement activities and is useful in particular to detect potential fraud. The functionality allows a toll charger to request, by means of a specific “request” ADU, information about all user identifiers that belong to a given service user (e.g. haulier).

5.2.12 Process user complaints functionality

The process user complaints functionality is derived from the ISO 17573-1 service “Providing customer care”. In ISO 17573-1, the providing customer care service belongs in the group of services that involve the user and the toll service provider. This functionality defines a specific aspect of customer care that involves interaction between the toll charger and the toll service provider to handle user complaints, covering the case where a previously accepted billing detail, possibly included in an accepted payment claim, is requested to be revoked because of a complaint by the user.

The toll service provider, after having examined a user complaint and verified it is well founded, has the possibility to deliver the complaint to the toll charger, by means of a “user complaint” ADU.

The toll charger has the possibility to acknowledge any received “user complaint” ADU by using the acknowledgement functionality, so indicating that the complaint will be processed.

When processing of the user complaint is complete, the toll charger has the possibility of signalling the completion of the evaluation process by issuing a “user complaint response” ADU, so indicating the acceptance or the refusal of the user complaint.

5.2.13 Exchange quality assurance parameters functionality

The report quality assurance parameters functionality is derived from ISO 17573-1 and is defined in the EFC system service “monitoring operations”.

In ISO 17573-1, the report quality assurance parameters functionality is used to report and monitor the EFC activities of toll chargers and toll service providers. In this document, 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 functionality is used whenever an entity, both toll charger and toll service provider, needs to send unsolicitedly new or updated QA parameters to its counterpart or when asked by its counterpart to update or resend already sent QA parameters and is performed by issuing a “report QA” ADU.

Requiring the counterpart to send QA parameters is performed by means of a “request” ADU. The “request” ADU allows requesting already issued QA parameters to be resent, as well as requesting additional QA parameters.

The receiver of a “report QA” ADU has the possibility of acknowledging the received QA parameters by using the “acknowledgement” functionality.

5.2.14 Provide media exception list functionality

The manage media exception list functionality gives a toll service provider the possibility of delivering to a toll charger a media exception list originated by a media provider. This functionality is realised by the “media exception list” ADU and allows checking of possible media abuse in an EFC regime that uses a medium such as an integrated circuit (IC) card.

The content of the media exception list is defined by the media provider. The toll service provider is meant to deliver the provided media exception list without any modifications. The possibility of inserting in the “media exception list” ADU a media provider generated authenticator insures a tamper-proof transmission from the media provider to the toll charger.

Upon reception of a “media exception list” ADU, the toll charger has the means to formally and semantically check the received exception list. If an error is detected in the exception list, the whole exception list can be discarded by sending an ack ADU indicating the detected error.

The toll charger has the possibility to dispute the received media exception list by using the acknowledgment functionality and transmitting an error code. In this case the last correct list remains active in the systems of the toll charger, until a newly transmitted list has been positively acknowledged by using the acknowledgement functionality.

5.2.15 Provide media settlement data functionality

The provide media settlement data functionality allows a toll charger to respond to a “request” ADU from a toll service provider asking payment settlement data to be transmitted to the media provider. This functionality is performed by means of the “media settlement data” ADU. The information in the “media settlement data” ADU is generated from a medium connected with OBE in both DSRC and GNSS environment.

The settlement by using a medium is performed by OBE initiated and transparency via OBE from roadside equipment and/or central server. A medium provides a media settlement data with OBE in vehicle when passing physical charging station in DSRC based system and whenever settlement performing in GNSS based system.

NOTE 1 A media settlement data contains an evidence of payment settlement using a medium for both road charging and invoicing service user and is transmitted to the media provider via the toll charger and the toll service provider without any modification since tamper-proof code is contained.

The toll service provider has the possibility of confirming the received media settlement data by using the acknowledgment functionality.

NOTE 2 When the media provider finds something doubt and/or suspicion on received media settlement data, another recovery processing will be performed by the toll service provider and the toll charger under a bilateral agreement with the media provider.

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 document.

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 document shall declare which of the computational objects listed in Table 1 are supported. 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
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

Computational object	Interaction	Toll service provider rules	Toll charger rules
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

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
Process user complaint	User complaint	May initiate	Shall be able to receive
	Acknowledge (User complaint)	Shall be able to receive	Shall initiate
	User complaint response	Shall be able to receive	May initiate
	Acknowledge (reception of user complaint response)	Shall 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 (Report QA)	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
Provide Media Exception Data	Request	Shall be able to receive	May initiate
	Media Exception Data	May initiate	Shall be able to receive
	Acknowledge (Media Exception Data)	Shall be able to receive	Shall initiate
Provide Media Settlement Data	Request	May initiate	Shall be able to receive
	Media SettlementData	Shall be able to receive	May initiate
	Acknowledge (Media Settlement Data)	Shall 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.

To facilitate reading, the following conventions will be adopted in clause 6.2:

- 1) the names and values of the fields in APDUs will be outlined in **boldface**;
- 2) the corresponding ASN.1 types will be written using the `Courier` font.

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:

- 3) a mandatory **infoExchangeContent** field;
- 4) 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
<code>infoExchangeContent</code>	<code>InfoExchangeContent</code> . Actual content of the APDU. See Table 3.	m
<code>infoExchangeAuthenticator</code>	<code>AuthenticatorEfc</code> . Record (ASN.1 SEQUENCE) made of two fields of types <code>authenticatorEfc</code> , and <code>ackAuthenticatorEfc</code>	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 may optionally be signed by calculating an authenticator over the **infoExchangeContent** 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 7.3.2.

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
<code>apci</code>	<code>ApciFields</code> . Record (ASN.1 SEQUENCE) containing Protocol Control Information	m
<code>adus</code>	<code>Adus</code> . Contains one or more (ASN.1 SEQUENCE OF) Application Data Units of the same type, as a selection (ASN.1 CHOICE) among the types of defined ADUs.	m

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

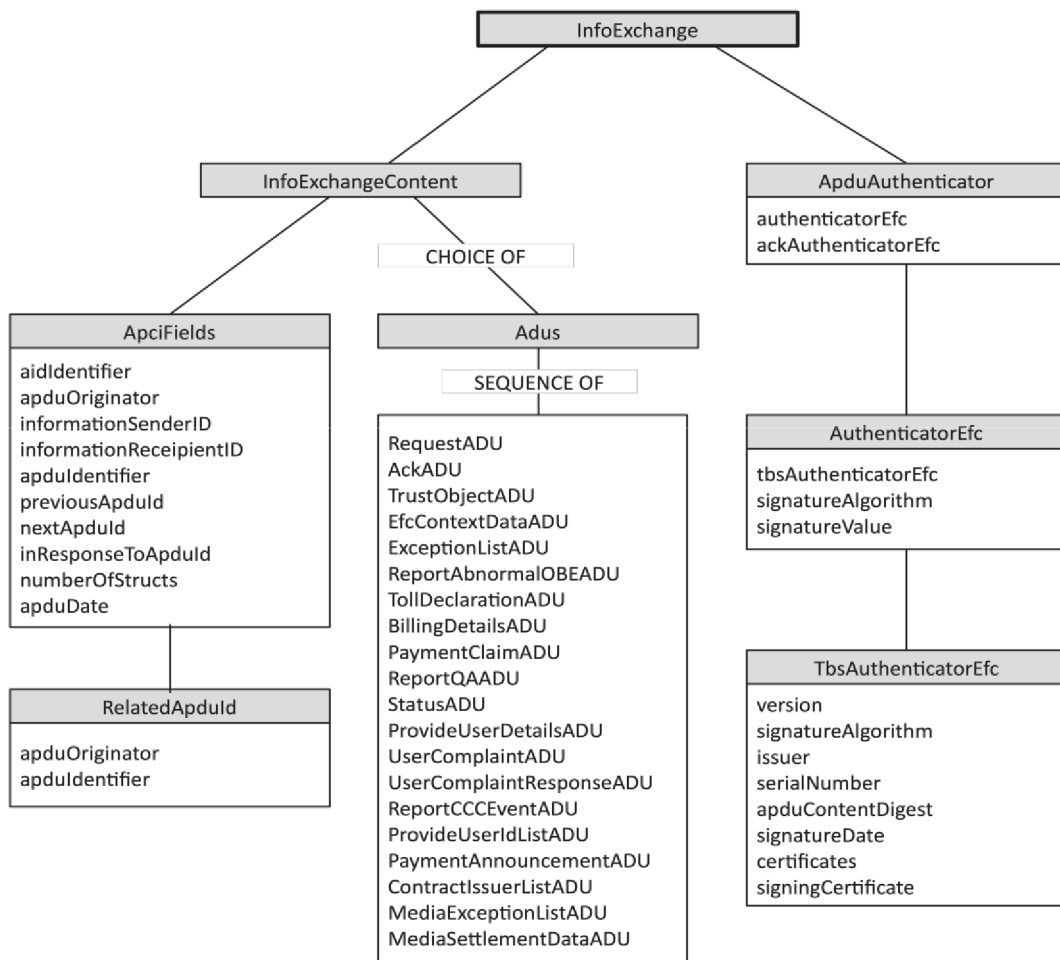


Figure 4 —InfoExchange data type structure

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 tables are inserted to better describe parameters with a complex structure;
3. additional semantics definition and usage of each parameter given as text after each table.

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	AidIdentifier. Integer ranging from 0 to 255	m
apduOriginator	Provider	m
informationSenderID	Provider	m
informationRecipientID	Provider	m
apduIdentifier	ApduIdentifier. Integer ranging from 0 to $2^{63}-1$	m
previousApduId	RelatedApduId	o
nextApduId	RelatedApduId	o
inResponseToApduId	RelatedApduId	o
apduDate	GeneralizedTime	m

aidIdentifier shall identify the Protocol Version number. This version is identified by the value 1.

apduOriginator shall be the identifier of the entity originating the contents of the APDU, which can be either a toll charger or a service provider. That may or may not be the actual sender of the information.

informationSenderID shall identify the entity actually sending the APDU. It may be used to identify a technical entity supporting the apdu originator 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 identify the 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 apdu originator that is assigned by the APDU originator themselves. It is used to either acknowledge or request retransmission of a previously issued APDU.

previousApduId may optionally be used to indicate that the APDU is related to a previously sent APDU from the apdu originator. It shall identify the APDU in terms of and apdu identifier. This field, together with the **nextApduId** field, may be used to identify chains of APDUs. The *RelatedApduId* data type is a record (ASN.1 SEQUENCE) of:

- **apduOriginator**, of type *Provider* that shall identify the APDU originator;
- **apduIdentifier**, of type *ApduIdentifier* that shall identify the APDU within **apduOriginator**.

nextApduId may optionally be used to indicate that the APDU is related to a next to be sent APDU from the APDU originator. This field, together with the **previousApduId** field, may be used to identify chains of APDUs.

inResponseToApduId may optionally be used to 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.

apduDate shall specify a time stamp for the APDU.

6.2.3 Application Data Units

The Application Data Units (ADUs field) shall contain (ASN.1 SEQUENCE OF) one or more data structures of the same type.

The ADUs listed in Table 5 are defined in this document.

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
Provide EFC context data	EfcContextDataAdu	Send EFC context data
Provide contract issuer information	ContractIssuerListAdu	Provide information on contract issuer
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
Process user complaint	UserComplaintAdu	Send a user complaint
	UserComplaintResponseAdu	Reply to a user complaint
Provide media exception lists	MediaExceptionListAdu	Send media exception list
Provide media settlement data	MediaSettlementDataAdu	Send media settlement data

These data structures are specified in 6.3 to 6.22.

6.2.4 ADU identification

Each ADU is identified by a mandatory field named **aduIdentifier** of type `AduIdentifier`, which is defined as an integer ranging from 0 to $2^{63}-1$. The field **aduIdentifier** is specified by the ADU originator with the rule that the **aduIdentifier** shall be unique within the ADU type. With that rule, considering that each ADU originator is univoquely identified by its **providerId**, each ADU may be univoquely identified by a triple of identifiers of types `ProviderId`, `AduType`, and `AduIdentifier`.

For legacy reasons, there is also another way to identify a toll declaration in the `TollDeclarationAdu` by the means of the value of its **tollDeclarationId** field (see Table 87).

For legacy reasons, there is also another way to identify a billing detail in the `BillingDetailsAdu` by the means of the value of its **billingDetailsInfo** field (see Table 94).

6.2.5 ADU action code

When sending or requesting an ADU, an action code may optionally be specified, to indicate the receiver of the ADU which action is supposedly be done with that ADU. While the exact behaviour to be followed is implementation specific, and varies from ADU to ADU, the following generally available values are specified in this version of ISO 12855:

- **send**, which specifies a normal (probably initial) sending of the ADU
- **revoke**, which specifies a revocation of a previously sent ADU
- **adjust**, which specifies an adjustment of a previously sent ADU
- **resend**, which specifies a resending of an ADU

Specific usage of the field `actionCode` for each ADU is indicated whenever needed in the following clauses.

6.2.6 User identification

A data type used in most ADUs to identify a user is the `UserId` data type. Because a user can be identified in various ways, and by different methods, the `UserId` data type is represented as a series (ASN.1 SEQUENCE) of different types, all optional. However, when the data type `UserId` is used, at least one of the fields shall be specified. The structure of the `UserId` data type is shown in Table 6 and is formally specified in Annex A.

Table 6 — `UserId` fields

Field name	Data type	m/o
<code>pan</code>	<code>PersonalAccountNumber</code>	o
<code>contractSerialNumber</code>	<code>ContractSerialNumber</code>	o
<code>licencePlateNumber</code>	<code>LPN</code>	o
<code>obeId</code>	Record (ASN.1 SEQUENCE) made of the following fields: <code>manufacturerId</code> , <code>equipmentObuId</code>	o
<code>iccId</code>	<code>IccId</code>	o

`pan` may optionally be used to identify the user by their Personal Account Number. It shall be used as defined in ISO 14906;

`contractSerialNumber` may optionally be used to identify the user by their contract serial number. It shall be used as defined in ISO 14906;

`licencePlateNumber` may optionally be used to identify the user by the licence plate number of the user's vehicle. It shall be used as defined in ISO 14906;

`obeId` may optionally be used to identify the user by the OBE identifier of the user. The **`obeId`** field is made of the following data elements:

- **`manufacturerId`** which shall be used to uniquely identify the OBE manufacturer. It shall be used as defined in ISO 14906;
- **`equipmentObuId`** which shall be used to uniquely identify the OBE within the manufacturer. It shall be used as defined in ISO 14906.

iccid may optionally be used to identify the ICC which may be associated to a user of EFC equipment for the purposes of identification of payments. It shall be used as defined in ISO 14906.

6.3 RequestAdu data structure

The **RequestAdu** data type is used to indicate a general readiness to receive, or to specifically request a data structure to be sent.

The **RequestAdu** type is a selection (ASN.1 CHOICE) among different substructures, each one specifying a different type of request. Table 7 describes the fields defined as CHOICE in the data structure, which is formally defined in Annex A.

Table 7 — RequestAdu fields

Field name	Data type/Data description	m/o
genericRequest	Record (ASN.1 SEQUENCE) made of the following fields: apduIdentifier, requestedAduType, aduIdentifier, actionCode	N/A
exceptionListRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(exceptionListRequest), aduIdentifier, period, exceptionListType, exceptionListVersion, actionCode	N/A
trustObjectRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(trustObject), aduIdentifier, TrustObjectSpec, actionCode	N/A
tollDeclarationRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(TollDeclaration), aduIdentifier, userId, startTime, endTime, actionCode	N/A
userDetailsRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(userDetails), aduIdentifier, userId, listOfParametersRequested, userDetailsRequestReason, userInfoValidityPeriod, actionCode	N/A
cccEventRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(reportCCCEventAdu), aduIdentifier, userId, startTime, endTime, actionCode	N/A
userListRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(provideUserIdListAdu), aduIdentifier, userIdRequestType, userId, userIdRequestTime, actionCode	N/A
mediaExceptionListRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(mediaExceptionListAdu), aduIdentifier, period, exceptionListType, versionId, actionCode	N/A
mediaSettlementDataRequest	Record (ASN.1 SEQUENCE) made of the following fields: requestedAduType(mediaSettlementDataAdu), aduIdentifier, period, iccid, versionId, actionCode	N/A

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

1. If none of the fields is specified, the **RequestAdu** indicates a general readiness to receive;
2. **apduIdentifier** may optionally be used to request that the APDU carrying the same identifier be (re)sent. If the **apduIdentifier** is specified, nor the **requestedAduType** neither the

aduIdentifier may be specified. If the **apduIdentifier** is not specified, the **requestedAduType** or the **aduIdentifier** may be specified;

3. if the **requestedAduType** field is specified, an ADU of the indicated data type is expected. It is used to indicate generic readiness to receive ADUs of the indicated type;
4. **aduIdentifier** is a list (ASN.1 SEQUENCE OF) of identifiers of type `AduIdentifier` that may optionally be used to ask for retransmission of the specified ADU's. If **aduIdentifier** is specified, then **requestedAduType** shall be also specified, to indicate the type of the ADU's to be re-transmitted;
5. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value **exceptionListAdu**;
2. if no optional fields are specified, this indicates generic readiness to receive ADUs of the ExceptionList type. In this case it is recommended to use a **genericRequest**, with the **requestedAduType** field bearing the value **exceptionListAdu**;
3. **aduIdentifier** may optionally be used to identify the exceptionListAdu to be re-transmitted.
4. **period** field may optionally be used to request to retransmit all ExceptionList ADUs that are related to the given period of time. If **period** is not specified, the request is not limited to a specific period of time;
5. **exceptionListType** field may optionally be used to request to re-transmit all ExceptionList ADUs that are of the specified type;
6. **exceptionListVersion** field may optionally be used to request to retransmit all ExceptionList ADUs that are of the specified version. If the field is not specified, the last available version is meant;
7. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value **trustObjectAdu**;
2. if no optional fields are specified, this indicates generic readiness to receive ADUs of the TrustObject type. In this case it is recommended to use a **genericRequest**, with the **requestedAduType** field bearing the value **trustObjectAdu**;
3. **aduIdentifier** may optionally be used to identify the requested TrustObjectAdu to be re-transmitted. If the **aduIdentifier** is specified, the **trustObjectSpec** field shall not be set;
4. **trustObjectSpec** may optionally be used to specify the type and purpose of the requested trust object. The type of the trust object requested shall be specified in the field `typeOfTrustObject`, of type `RequestedTrustObjectType`, which is a selection (ASN.1 CHOICE) of:
 - certificates;
 - public keys;
 - DSRC master keys;
 - MAC master keys;
 - DSRC key references;

— other trust objects.

Detailed description of trust objects types and purposes are in clause 6.6.

NOTE This document does not limit the use of individual combinations of trust object types and trust object purposes. In real systems, however, some of the theoretically possible combinations of these two parameters do not make sense and should therefore not be used in implementations. The correct use of these parameters remains under the responsibility of the implementor(s) and/or the entities operating the toll charger and toll service provider systems.

5. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value **tollDeclarationAdu**;
2. if no optional fields are specified, this indicates generic readiness to receive ADUs of the TollDeclaration type. In this case it is recommended to use a **genericRequest**, with the **requestedAduType** field bearing the value **tollDeclarationAdu**;
3. **aduIdentifier** may optionally be used to identify the requested TollDeclarationAdu to be re-transmitted. If the **aduIdentifier** is specified, the **userId**, **startTime**, and **endTime** fields shall not be specified;
4. **userId** field may optionally be used to identify the user for which toll declarations are requested;
5. **startTime** field may optionally be used to indicate a start of a period for which toll declarations are requested;
6. **endTime** field may optionally be used to specify the end of a period for which toll declarations are requested. If **endTime** is specified, then the **startTime** shall be specified as well;
7. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value **provideUserDetailsAdu**;
2. **aduIdentifier** may optionally be used to identify the requested UserDetailsAdu to be re-transmitted;
3. **userId** field shall specify the identifier of the user for which additional details are requested;
4. **listOfParametersRequested** may 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;
5. **userDetailsRequestReason** may optionally be used to specify the reason for the request;
6. **period** may 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;
7. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value “reportCCCEventAdu”;

2. if no optional fields are specified, this indicates generic readiness to receive ADUs of the `reportCccEvent` type. In this case it is recommended to use a **genericRequest**, with the **requestedAduType** field bearing the value **reportCCCEventAdu**;
3. **aduIdentifier** may optionally be used to identify the requested **reportCCCEventAdu** to be re-transmitted;
4. **userId** field may optionally be used to specify the identifier of the user for which the CCC events are requested;
5. **startTime** field may optionally be used to specify a start of a period for which CCC events are requested;
6. **endTime** field may optionally be used to specify the end of a period for which CCC events are requested. If **endTime** is specified, the **startTime** shall be specified as well;
7. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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. **aduIdentifier** may optionally be used to identify the requested `provideUserIdListAdu` to be re-transmitted;
3. **userIdRequestType** field shall indicate the type of request. Currently, only one value is defined: **allUserIdsToGivenCustomer**;
4. **userId** field may optionally be used to identify 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 **userId** field is not specified, the request is meant to retrieve the identifiers of all service users of that TSP as filtered by the value of **userIdRequestType**;
5. **userIdRequestTime** field may optionally be used to define a point of time. If this attribute is specified, the toll service provider shall answer to the request based on data valid at the provided point of time and not at the time of the request. If this attribute is not specified, the latest information shall be provided;
6. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value “**mediaExceptionListAdu**”;
2. if no optional fields are specified, this indicates generic readiness to receive ADUs of the `mediaExceptionList` type. In this case it is recommended to use a **genericRequest**, with the **requestedAduType** field bearing the value “**mediaExceptionListAdu**”;
3. **aduIdentifier** may optionally be used to identify of the `MediaExceptionListADU` to be re-transmitted;
4. **period** may optionally be used to request to re-transmit all `MediaExceptionListADUs` that are related to the given period of time;
5. **mediaExceptionListType** may optionally be used to request to re-transmit all `MediaExceptionListADUs` that are of the specified type;

6. **mediaExceptionListVersion** may optionally be used to request to re-transmit MediaExceptionListADUs that are of the specified version. If the field is not specified, the last available version is meant;
7. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

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

1. **requestedAduType** field shall bear the value “**mediaSettlementDataAdu**”;
2. if no optional fields are specified, this indicates generic readiness to receive ADUs of the **mediaSettlementData** type. In this case it is recommended to use a **genericRequest**, with the **requestedAduType** field bearing the value “**mediaSettlementDataAdu**”;
3. **aduIdentifier** may optionally be used to identify the requested mediaSettlementDataAdu to be re-transmitted;
4. **period** may optionally be used to request to re-transmit all MediaSettlementDataAdus that are related to the given period of time;
5. **iccid** may optionally be used to request to re-transmit all MediaSettlementDataAdus that contain the specified identifier;
6. **mediaSettlementListVersion** may optionally be used to request to re-transmit MediaSettlementDataAdus that are of the specified version. If the field is not specified, the last available version is meant;
7. **actionCode** may optionally be used to indicate specific actions to be pursued by the receiver of the requestAdu (see 6.2.5).

6.4 AckAdu data structure

The **AckAdu** data type 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 8 describes the fields in an **AckAdu** data type, which is formally defined in Annex A.

Table 8 — AckAdu fields

Field name	Data type/Data description	m/o
apduIdentifier	ApduIdentifier. Integer ranging from 0 to $2^{63}-1$	m
apduAckCode	ApduReasonCode. Integer ranging from 0 to 255. (see Table 9)	m
apduAckText	UTF8String	o
issues	List of records (ASN.1 SEQUENCE OF SEQUENCE), each record made of the following fields: issueAduStruct , issueAduIdentifier , issueLocation , issueContent , issueCode , issueText	o
actionCode	ActionCode	o

apduIdentifier shall identify the APDU containing the data structure(s) being acknowledged.

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. The values defined in this document for **apduAckCode** are listed (in decimal notation) in Table 9.

Table 9 — ApduReasonCode values

Name	Semantics	Value
reserved	Code reserved for future ISO/CEN extensions	0
obsoleteAPDU	The APDU is no longer valid or to be used.	1
apduOK	The APDU was accepted.	2
apduNotOK	The APDU was rejected.	3
sequenceError	The APDU was rejected because the APDU was sent out of the logical sequence of a transaction.	4
otherReason	The APDU was rejected. No predefined APDU Reason Code could be used, or a bilateral agreement on the APDU reason code existis.	5
protocolVersionError	The APDU was rejected because the Protocol Version used is not supported by the recipient.	6
originatorRejected	The APDU was rejected because the Apdu Originator is not known, or no valid contract exists.	7
recipientUnknown	The APDU was rejected because the Information Recipient is not known, or no valid contract exists.	8
relatedApduUnknown	The APDU was rejected because the related APDU is unknown or the transaction was finished.	9
requestSentTooOften	The APDU was rejected because the request was sent too often or to early (processing of similar request still active)	10
apduPartiallyAccepted	The APDU was partially accepted. In case this ApduAckCode is used at least one of the two optional fields apduAckText or issues shall be present to provide information about the reason for partial acceptance and/or a reference to the part of the APDU which was not accepted.	11

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

1. if the **apduAckCode** field indicates rejection (value of apduNotOK), the whole referred APDU is rejected;
2. if the **apduAckCode** field indicates acceptance, the referred APDU is accepted (syntactically and semantically correct;

apduAckText of type `UTF8String` may optionally be used to add a text to further explain the reason for accepting or rejecting the APDU.

EXAMPLE A number of ADUs of the type `BillingDetails` have been transferred in a previous APDU. The receiver accepts one specific `BillingDetail` and rejects all others. To do so, the receiver sends an `AckAdu` with the `apduAckCode` “**apduOk**” and indicates for each of the rejected ADUs the reason for rejection.

In case the **ApduAckCode** value **apduPartiallyAccepted** is used, at least one of the two optional fields **apduAckText** or **issues** shall be present to provide information about the reason for partial acceptance and/or a reference to the part of the APDU which was not accepted.

issues may optionally be used to list detailed error and/or warning information that respond 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 data structure of issues is specified in Table 10.

actionCode may optionally be used to indicate the action associated with the specific ADU (see 6.2.5).

Table 10 — issues fields

Field name	Data type/Data description	m/o
issueAduStruct	NumberOfADUStruct. Integer ranging from 0 to $2^{16}-1$	o
issueAduIdentifier	AduIdentifier	o
issueLocation	UTF8String	o
issueContent	UTF8String	o
issueCode	INTEGER	m
issueText	UTF8String	o

issueAduStruct may optionally be used to reference the ADU that contains an issue by indicating its position within the list (ASN.1 SEQUENCE OF) of ADU's in the corresponding APDU. If **issueAduStruct** is set to 0, the issue applies to all ADUs of the corresponding APDU.

issueAduIdentifier may optionally be used to specify the unique identifier of the ADU that contains an issue. It shall not be specified if the issue applies to all ADUs within the APDU.

issueLocation may optionally be used to address the location of the issue within the ADU by using XPath expression.

issueContent may optionally be used to provide further information on the content of the issue. The usage of this attribute is based on bilateral agreements between toll charger and toll service provider.

issueCode shall contain the reason code (see Table 11) of an issue.

issueText may optionally be used to provide a description of the issue. The usage of this attribute is based on bilateral agreements between toll charger and toll service provider.

In case the **ApduAckCode** value **apduPartiallyAccepted** is used, at least one of the two optional fields **apduAckText** or **issues** shall be present to provide information about the reason for partial acceptance and/or a reference to the part of the APDU which was not accepted.

Some values of the **AduReasonCode** field are reserved for CEN and ISO use. The values that are defined in this document are listed (in decimal notation) in Table 11 and are formally defined in Annex A. The value of the **AduReasonCode** provides feedback information to the sender or originator of an ADU about the acceptance or rejection of an ADU that was received and processed by the information recipient.

Table 11 — AduReasonCode values

Name	Semantics	Value
invalidAdu	The specified ADU is invalid	0

Name	Semantics	Value
notSupportedAdu	The specified ADU is not supported	1
aduPartiallyRejected	Part of the specified ADU is rejected	2
reserved	Codes reserved for general ADU related errors	3..99
trustObjectExpired	The exchanged trust object is not more valid	100
trustObjectUnreadable	The received trust object is not readable	101
reserved	Codes reserved for trust object data 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	203
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
exceptionListsStatusTypeRejected	The type of status 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

Name	Semantics	Value
abnormalObeDateAndTimeRejected	The date and time in abnormal OBE 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
tollDeclartionDuplicateTollDeclarationIdRejected	A duplicate of the tollDeclarationId has been identified	600
tollDeclarationActionCodeRejected	The actionCode in a Toll Declaration ADU is unknown or not supported by the recipient of the toll declaration	601
tollDeclarationChargeReportStructureRejected	One or more elements inside the Charge Report structure of the Toll Declaration ADU are not supported by the recipient of the Toll Declaration Note: For some data elements inside the Charge Report there are individual ADU reason codes defined below.	602
tollDeclarationActionCodeSendSentTooOftenRejected	The actionCode = send has been sent too often in Toll Declaration ADUs	603
tollDeclarationInvalidChargeObjectRejected	One or more validations failed inside the elements chargeObjectId, or timeWhenUsed, or vehicleDescription	604
tollDeclarationActionCodeAdjustSentTooOftenRejected	The actionCode = adjust has been sent too often in Toll Declaration ADUs	605
tollDeclarationTariffClassIdRejected	The TariffClassId declared in the Toll Declaration ADU was not found by the recipient of the Toll Declaration	606
tollDeclarationTariffRejected	The Tariff declared in the Toll Declaration ADU was not found by the recipient of the Toll Declaration	607
tollDeclarationDuplicateVehiclePassageRejected	A duplicate vehicle passage has been identified by the recipient of the Toll Declaration. A duplicate vehicle passage is defined by the identical combination of: [chargeReport.obeld + chargeReport.vehicleLPNr + chargeReport.paymentMeans.personalAccountNumber + DetectedChargeObject.chargeObjectId + DetectedChargeObject.timeWhenUsed]	608
tollDeclarationInvalidDetectedChargeObjectManyRejected	One or more validation steps for a toll declaration failed in many DetectedChargeObjects	609

Name	Semantics	Value
tollDeclarationTollChargerIssueRejected	Toll declaration declined by TC due to error in TC system, further organizational handling required between TC and TSP. Note: This could be used, for example, in case errors on the toll charger side prevent correct acceptance of a toll declaration.	610
tollDeclarationTollContextRejected	The tollContext declared in the Toll Declaration ADU is not known by the recipient of the Toll Declaration	611
tollDeclarationTollFeeRejected	The toll fee declared in the Toll Declaration ADU has been rejected by the recipient of the Toll Declaration. Note: depending on the use of optional data elements inside the Toll Declaration toll fee information may be provided in different data elements	612
reserved	Codes reserved for toll declarations exchange errors	613..699
billingDetailsIssuerIdRejected	The issuer id 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
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

Name	Semantics	Value
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 and accepted 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
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

Name	Semantics	Value
reserved	Codes reserved for quality assurance exchange errors	1204..1299
complaintIdMissing	Identifier missing in a UserComplaintAdu	1301
relatedBillingDetailsUnknown	The billing details id referred in the User Complaint ADU is not known by the toll charger	1302
paymentClaimUnknown	The payment claim id referred in the User Complaint ADU is not known by the toll charger	1303
dateOfComplaintIncorrect	The date of the complaint is incorrect	1304
dateOfComplaintMissing	The date of the complaint is missing	1305
complaintReasonMissing	The reason of the complaint is missing	1306
complaintReasonUnknown	The reason of the complaint is not known	1307
addComplaintReasonMissing	Additional complaint information is missing, and the reason of the complaint is “other”	1308
additionalInfoIdMissing	The id in the additional complaint information is not specified	1309
infoDateTimeMissing	The field date in additional information is missing	1310
infoDateIncorrect	The field date in additional information is incorrect	1311
incorrectAuthenticator	The authenticator is wrong	1312
relatedBillingDetailsAlreadyComplained	The referred billing details is mentioned in another User Complaint currently being processed	1313
reserved	Codes reserved for complaint related errors	1313..1399
reserved	Codes reserved for future ISO/CEN extensions	1400..2999
semanticError	The semantics of the message received was not understood by the recipient. Note: Every message (except AckAdu) can be responded with a “semantic error” code when its semantics is not understood.	3000
reserved	Codes reserved for future ISO/CEN extensions	3001...3009
actionCodeNotSupported	The sent action code is not supported. Note: This AduReasonCode shall be used only in response to ADUs which contain action codes.	3010
reserved	Codes reserved for future ISO/CEN extensions	3011...3999

Name	Semantics	Value
acceptedWithWarning	The ADU was accepted and will be processed by the recipient. Further information will be provided by means of warnings. Note: Bilateral agreement on any other issue which semantics needs to be defined for the attributes issueLocation, issueContent and issueText.	4000
reserved	Codes reserved for future ISO/CEN extensions	4001...9999
reserved	Codes reserved for private extensions defined in bilateral agreements between TCs and TSPs	10000 .. 65535

6.5 StatusAdu data structure

The `StatusAdu` data type can be used to indicate to the recipient that previously sent (and possibly acknowledged) 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 12 indicates the fields of the `StatusAdu` data type, which is formally defined in Annex A.

Table 12 — StatusAdu fields

Field name	Data type/Data description	m/o
generalStatusCode	GeneralStatusCode	m
apduStatusCode	Record (ASN.1 SEQUENCE) made of the following fields: apdu, reasonOfAduStruct	o
actionCode	ActionCode	o

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

- **notReadyToReceive**: the issuing entity is not ready to receive;
- **readyToReceive**: the issuing entity is ready to receive.

apduStatusCode may optionally be used to 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** is a list (ASN.1 SEQUENCE OF SEQUENCE) made of two fields: **aduIdentifier**, that identifies an ADU within the addressed APDU, and **reason**, that carries an `AduReasonCode` value, defined in Table 11.

actionCode may optionally be used to indicate specific actions to be pursued by the receiver of the `statusAdu` (see 6.2.5).

6.6 TrustObjectAdu data structure

The `TrustObjectAdu` data type 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.

Each **TrustObjectsAdu** data structure shall contain one trust object and additional information and parameters for that trust object.

Table 13 indicates the fields in a **TrustObjectsAdu** data type, which is formally defined in Annex A.

Table 13 — TrustObjectAdu fields

Field name	Data type/Data description	m/o
trustObjectID	AduIdentifier	m
trustObjectReplaced	AduIdentifier	o
purposesOfTrustObject	Record made of fields of the same type (ASN.1 SEQUENCE OF), each of type TrustObjectPurpose	m
startValidity	GeneralizedTime	o
endValidity	GeneralizedTime	o
trustObjectStatus	TrustObjectStatus	m
trustObject	TrustObjectCode	o
actionCode	ActionCode	o

trustObjectID shall indicate the unique identifier of the trust object. The uniqueness of the value of **trustObjectID** shall be guaranteed by the initiator of the trust object (i.e. toll charger or toll service provider) inside the realm of the same provider ID.

trustObjectReplaced shall contain the trust object identifier of the trust object which shall be replaced by the new trust object that is transferred using the present **TrustObjectAdu** structure. This optional data element shall only be present and used in case an already existing and valid trust object shall be replaced by a new trust object. If **trustObjectReplaced** is present, the new trust object shall have the **TrustObjectID** set to a new value.

purposesOfTrustObjects shall be used to indicate the possible usage(s) the trust object is meant for. It contains a list of values which allows to specify the possible usage of the same trust object for one or more than one purposes. Allowed values shall be:

- **trustObjects**, unspecified purpose;
- **dSRCCharging**, to validate authenticators from DSRC charging applications;
- **dSRCAC**, to calculate access credentials for DSRC charging applications;
- **oBEInterrogation**, to validate authenticators received during a CCC transaction;
- **oBEInterrogationAC**, to calculate access credentials for CCC;
- **sIGExceptionList**, to authenticate received exception lists;
- **sIGContextData**, to authenticate received context data;
- **sIGBillingDetails**, to authenticate received billing details;
- **sIGFiscalObjects**, to authenticate received fiscal objects;
- **sIGCommunication**, to authenticate received ADUs by infoExchangeAuthenticator;

- **eNCCommunication**, to decrypt received ADUs;
- **dSRCKeyEncryption**, to encrypt DSRC keys;
- **secChannelEstablishment**, certificate to establish IPsec (VPN);
- **certIssuing**, certificate to issue certificates;
- **siGUserCommunication**, user certificate to verify communication via, e.g. email;
- **certRevocationListing**, certificate revocation list;
- **siGChargeReport**, certificate or certificate chain to validate charge report;
- **oBEInterrogationNonRep**, Key reference for the non-repudiation authentication for CCC or LAC.

NOTE: This document does not limit the use of individual combinations of trust object types and trust object purposes. In real systems, however, some of the theoretically possible combinations of these two parameters do not make sense and may therefore not be used in implementations. The correct use of these parameters remains under the responsibility of the implementor(s) and/or the entities operating the toll charger and Toll Service Provider systems.

startValidity may optionally be used to indicate when the trust object is valid from. In case this optional element is not present, the specified trust object is valid with immediate effect.

endValidity may optionally be used to indicate the expiry date for a trust object. In case this optional element is not present, the specified trust object has no expiry date.

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

- **valid**, for a valid trust object;
- **expired**, for an expired trust object;
- **revoked**, for a revoked trust object;
- **update**, for a trust object which replaces an already existing and valid trust object.

trustObject is of data type `TrustObjectCode` and may optionally be used to specify the value of the trust object. If **trustObject** is omitted, a new value of **trustObjectStatus** is sent for an already transferred trust object.

TrustObjectCode is supported in this document for the following trust object types:

- certificates;
- public keys;
- DSRC keys;
- MAC keys;
- DSRC key references;
- generic trust objects.

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

The `CertificateObject` data type allows exchanging certificates between TC and TSP. The certificate structure and coding shall be according to ISO/IEC 9594-8. Table 14 explains the fields in the `CertificateObject` data type, which is formally defined in Annex A.

Table 14 — CertificateObject fields

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

The **certificateType** field specifies the type of certificate. The following values are specified:

- **certTcHTTPS**, indicating a toll charger certificate for HTTPS usage;
- **certTcMAIL**, indicating a toll charger certificate for email exchange;
- **certTcNSIG**, indicating a toll charger certificate for signatures;
- **certTcTA**, indicating the toll chargers' sub-Certification Authority certificate;
- **certTspRoot**, indicating the toll chargers' root-Certification Authority certificate;
- **certTspHTTPS**, indicating a toll service provider certificate for HTTPS usage;
- **certTspMAIL**, indicating a toll service provider certificate for email exchange;
- **certTspNSIG**, indicating a toll service provider certificate for signatures;
- **certTspTA**, indicating the toll service provider sub-Certification Authority certificate;
- **certTspRoot**, indicating the toll service provider root-Certification Authority certificate;
- **certTspCR**, indicating the toll service provider charge report verification.

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

The **PublicKeyObject** data type is used to transfer public keys whose authenticity is proven by a fingerprint exchanged over a separate communication channel. Public keys without a certificate may be used to establish the initial trust relation between TC and TSP. Table 15 explains the fields in the **PublicKeyObject** data type, which is formally defined in Annex A.

Table 15 — PublicKeyObject fields

Field name	Data type/Data description	m/o
publicKeyType	PublicKeyType	m
serialNumber	INTEGER ranging from 0 to $2^{63}-1$	m
issuer	Provider	m
algorithmIdentifier	AlgorithmIdentifier type as imported from ISO/IEC 9594-8	m
publicKey	OCTET STRING	m

The **publicKeyType** field shall specify 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 indicates the key's serial number.

The **issuer** field identifies the issuer of the key.

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

The **publicKey** field shall contain the key.

The `DSRCKeyObject` data 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 16 explains the fields in the `DSRCKeyObject` data type, which is formally defined in Annex A.

Table 16 — DsrcKeyObject fields

Field name	Data type/Data description	m/o
encryptionKeyId	Selection (ASN.1 CHOICE) between certificate and publicKey defined as <code>EncryptionKeyId</code> type	m
timestamp	<code>UTCTime</code>	m
dsrcKeys	Series of records (ASN.1 SEQUENCE OF SEQUENCE), each record containing the following fields: eFCCM , key	m

encryptionKeyId of type `EncryptionKeyId` is a selection (ASN.1 CHOICE) between two types of trust objects identified by the fields **certificate** and **publicKey** which shall be used to identify the encryption type used. Both fields **certificate** and **publicKey** are defined as a record (ASN.1 SEQUENCE) of two elements (**serialNumber**, **issuer**) identifying either a certificate or a public key by means of its originator and the originator's assigned serial number.

timestamp of type `UTCTime` shall contain the time of the encryption.

dsrcKeys is a list (ASN.1 SEQUENCE OF) of couples (ASN.1 SEQUENCE), each made of elements of types `EFCCM` (described in Table 17 and formally described in Annex A) and `EncrKey` (described in Table 18 and formally described in Annex A).

Table 17 — EFCCM fields

Field name	Data type/Data description	m/o
eFCCM	<code>EfcContextMark</code>	m
bitmask	OCTET STRING	m

eFCCM shall be used to specify the EFC Context Mark to be used in a DSRC environment.

bitmask shall be used to indicate which type of contract the key applies to by setting the bits corresponding to the position of the contract in the list defined by the `ContractTypes` data type (see Table 51).

Table 18 — EncrKey fields

Field name	Data type/Data description	m/o
keyType	<code>KeyType</code>	m
keyRef	INTEGER ranging from 0 to 255	m
encrKey	OCTET STRING	m

Field name	Data type/Data description	m/o
kVC	OCTET STRING	m
keyDescription	PrintableString	m

keyType is a record (ASN.1 SEQUENCE) defined as type `KeyType` that shall be used to identify the type of key. It is made of the the following fields:

- **normativeReference**: a selection (ASN.1 CHOICE) of either an `OBJECT IDENTIFIER` or a `UTF8String` that may optionally be used to identify the standard that identifies the key;
- **keyUsage**: shall be used to specify the way the key is to be used. The following values are defined for `KeyUsage`:
 - **authentication**, to compute authenticators
 - **accessCredentials**, to compute access credentials
 - **encryption**, to encrypt data
 - **otherUsage**, for other usages

keyRef shall be used to identify the key for usage in a DSRC environment

encrKey shall contain the key, encrypted using the certificate referred to by the **encryptionKeyId** field of `DsrcKeyObject`.

kVC shall be used to provide 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.

keyDescription may optionally be used to provide a textual description of the key.

Some data elements are possibly authenticated by a derived MAC. The `MacKeyObject` data type allows the exchange of MAC master keys. Table 19 explains the fields in the `MacKeyObject` data type, which is formally defined in Annex A.

Table 19 — 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	UTCTime	m
algorithmIdentifier	AlgorithmIdentifier, as imported from ISO/IEC 9594-8	m
masterKeyRef	INTEGER, ranging from 0 to $2^{32}-1$	m
encrKey	OCTET STRING	m
kVC	OCTET STRING	m

The **serialNumber** field shall specify the serial number of the certificate used.

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

The **timestamp** field shall specify the time of the encryption.

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

The **masterKeyRef** field shall identify the master key used for encryption.

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

The **kVC** field shall contain 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** data type shall be used to exchange a non-repudiation key reference for, e.g. CCC (ISO 12813) or LAC (ISO 13141). Table 20 explains the fields in the **DSRCKeYRef** data type, which is formally defined in Annex A.

Table 20 — DSRCKeYRef fields

Field name	Data type/Data description	m/o
eFCContextMark	EFCCContextMark	m
keyRef	INTEGER, ranging from 0 to 255	m
referenceType	INTEGER defined as ReferenceType	m

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

The **keyRef** field shall contain the reference to the key to be used.

The **referenceType** field shall specify the usage of the key. The following values are supported:

- **cCCNonRepKeyRef**, used for non-repudiation in CCC (ISO 12813);
- **cCCAuthenticationKeyRef**, used for authentication in CCC (ISO 12813);
- **IACAAuthenticationKeyRef**, used for authentication in LAC (ISO 13141).

The **GenericTrustObject** data type allows exchanging a trust object that does not have a pre-defined structure. Table 21 explains the fields in the **GenericTrustObject** data type, which is formally defined in Annex A.

Table 21 — GenericTrustObject fields

Field name	Data type/Data description	m/o
typeOfTrustObject	INTEGER, ranging from 0 to 255	m
genericTrustObject	OCTET STRING	m

The **typeOfTrustObject** field shall specify the type of trust object, defined as **TrustObjectType** data type. It may be a certificate, a symmetric key, a certificate revocation list, a CRL distribution point, an OCSP address, an encrypted symmetric key, a public key, a key reference or an unspecified trust object (other trust object) as defined in Annex A. The following values are supported:

- **certificate**, used for certificates;
- **symmetricKey**, used for symmetric keys;
- **crl**, used for certificate revocation lists;
- **cdp**, used for cdps;
- **ocsp**, used for ocsp;

- **encryptedSymmetricKey**, used for encrypted symmetric keys;
- **publicKey**, used for public keys;
- **otherTrustObject**, used for other types of trust objects.

The **genericTrustObject** field shall contain the trust object.

6.7 EfcContextDataAdu data structure

6.7.1 General

The **EfcContextDataAdu** data type shall be used by the toll charger either as a response to the request ADU sent by the toll service provider or as a push ADU if a new version of EFC context data attributes is 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 more than one recipient.

The **EfcContextDataAdu** data type contains toll charger related administrative information, defined by the **EntityOverview** data type, followed by one data structure of either the **GnssContext** data type, **DsrcContext** data type or **DsrcClosedContext** data type. Each of these structures contains elements with own **version** fields, so allowing independent updates of relevant parts of the **efcContextData**.

The **EfcContextDataAdu** data type is described in Table 22 and is formally defined in Annex A.

Table 22 — EfcContextDataAdu fields

Field name	Data type/Data description	m/o
aduIdentifier	AduIdentifier	m
entityOverview	List of (ASN.1 SEQUENCE OF) at least one set of records (ASN.1 SEQUENCE) of fields describing the Toll Context, see Table 34	o
domainType	Choice (ASN.1 CHOICE) among one of the following types: GnssContext (to specify a toll context for autonomous tolling), DsrcContext (to specify a DSRC open toll context), DsrcClosedContext (to specify a DSRC closed context)	m
actionCode	ActionCode	o

aduIdentifier shall contain an unambiguous identifier for the EFC context data (see 6.2.4).

entityOverview shall provide administrative information about the toll charger responsible for the EFC context data. It is described in Table 34 on clause 6.7.5, and is formally defined in Annex A.

domainType specifies the type of the toll context (GNSS, DSRC) and shall contain the actual **efc** context data and supporting data that shall be applied in the toll context. Either **GnssContext**, **DsrcContext** or **DsrcClosedContext** data types shall be used. It is described in Table 23 and is formally defined in Annex A.

actionCode may optionally be used indicate the action associated with the specific ADU (see 6.2.5).

Table 23 — domainType fields

Field name	Data type/Data description	m/o
gnssContext	Record (ASN.1 SEQUENCE) made of fields of the following types: ContextInterrelations , RegimeContextData . See clause 6.7.2 for detailed description.	n/a