TECHNICAL SPECIFICATION

ISO/TS 17573-3

First edition
2021-08

Electronic fee collection — System architecture for vehicle-related tolling —

Part 3: Data dictionary

Perception du télépéage — Architecture de systèmes pour le péage lié aux véhicules —

Partie 3: Dictionnaire de données

Reference number
ISO/TS 17573-3:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

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 of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems,* in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 278, *Intelligent transport systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

A list of all parts in the ISO 17573 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document is a part of the ISO 17573 series that defines the system architecture for vehicle-related tolling. ISO 17573-1 gives a reference model for the system architecture. ISO/TS 17573-2 provides a collection of terms and definitions within the field of electronic fee collection (EFC) and road user charging that are used in the different documents published in ISO and CEN under the general title, *Electronic fee collection*.

This document (ISO/TS 17573-3) provides a data dictionary that contains the definitions of ASN.1 (data) types and the associated semantics.

The document is intended to be used as a reference by editors of ISO and CEN documents in EFC and in related areas of standardization (such as Intelligent Transport Systems, ITS).

It is foreseen that the library of ASN.1 (data) types contained in this document will be augmented with additional definitions as these become available.

Electronic fee collection — System architecture for vehicle-related tolling —

Part 3: Data dictionary

1 Scope

This document specifies the syntax and semantics of data objects in the field of electronic fee collection (EFC). The definitions of data types and assignment of values are provided in accordance with the abstract syntax notation one (ASN.1) technique, as specified in ISO/IEC 8824-1. This document defines:

— ASN.1 (data) types within the fields of EFC;

— ASN.1 (data) types of a more general use that are used more specifically in standards related to EFC.

This document does not seek to define ASN.1 (data) types that are primarily related to other fields that operate in conjunction with EFC, such as cooperative intelligent transport systems (C-ITS), the financial sector, etc.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 612, *Road vehicles — Dimensions of motor vehicles and towed vehicles — Terms and definitions*

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

ISO 4217, *Codes for the representation of currencies*

ISO 1176, *Road vehicles — Masses — Vocabulary and codes*

ISO/IEC 7812-1, *Identification cards — Identification of issuers — Part 1: Numbering system*

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

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 8859-2, *Information technology — 8-bit single-byte coded graphic character sets — Part 2: Latin alphabet No. 2*

ISO/IEC 8859-3, *Information technology — 8-bit single-byte coded graphic character sets — Part 3: Latin alphabet No. 3*

ISO/IEC 8859-4, *Information technology — 8-bit single-byte coded graphic character sets — Part 4: Latin alphabet No. 4*

ISO/IEC 8859-5, *Information technology — 8-bit single-byte coded graphic character sets — Part 5: Latin/Cyrillic alphabet*

ISO/IEC 8859-6, *Information technology — 8-bit single-byte coded graphic character sets — Part 6: Latin/Arabic alphabet*

ISO/IEC 8859-7, *Information technology — 8-bit single-byte coded graphic character sets — Part 7: Latin/Greek alphabet*

ISO/IEC 8859-8, *Information technology — 8-bit single-byte coded graphic character sets — Part 8: Latin/Hebrew alphabet*

ISO/IEC 8859-9, *Information technology — 8-bit single-byte coded graphic character sets — Part 9: Latin alphabet No. 5*

ISO/IEC 8859-10, *Information technology — 8-bit single-byte coded graphic character sets — Part 10: Latin alphabet No. 6*

ISO 14816, *Road transport and traffic telematics — Automatic vehicle and equipment identification — Numbering and data structure*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/TS 17573-2 and the following apply.

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/

3.1
BITSTRING **type**
simple type (3.14) whose distinguished values are an ordered sequence of zero, one or more bits

[SOURCE: ISO/IEC 8824-1:2021, 3.8.7]

3.2
CHOICE **type**
type defined by referencing a list of distinct types; each value of the choice type is derived from the value of one of the *component types* (3.4)

Note 1 to entry: Each value of the choice type is derived from the value of one of the component types.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.14 — modified, Note 1 to entry added.]

3.3
complex data type
one type that has more than *three levels* (3.17)

3.4
component type
one of the types referenced when defining a *CHOICE* (3.2), *SET* (3.12), *SEQUENCE* (3.10), *SET OF* (3.13), or *SEQUENCE OF* (3.11).

[SOURCE: ISO/IEC 8824-1:2021, 3.8.15]

3.5
data type
categorization of an abstract set of possible values, characteristics, and set of operations for an attribute

[SOURCE: ISO/IEC 25012:2008, 4.7 — modified, NOTE removed.]

3.6
INTEGER **type**
simple type (3.14) with distinguished values which are the positive and negative whole numbers, including zero (as a single value)

[SOURCE: ISO/IEC 8824-1:2021, 3.8.48]

3.7
object
well-defined piece of information, definition, or specification which requires a name in order to identify its use in an instance of communication

[SOURCE: ISO/IEC 8824-1:2021, 3.8.52]

3.8
OCTET STRING **type**
simple type (3.14) whose distinguished values are an ordered sequence of zero, one or more octets, each octet being an ordered sequence of eight bits

[SOURCE: ISO/IEC 8824-1:2021, 3.8.55]

3.9
parent type
type that is being constrained when defining a *subtype* (3.16), and which governs the subtype notation

[SOURCE: ISO/IEC 8824-1:2021, 3.8.58]

3.10
SEQUENCE **type**
type defined by referencing a fixed, ordered list of types (some of which can be declared to be optional)

Note 1 to entry: Each value of the sequence type is an ordered list of values, one from each *component type* (3.4).

[SOURCE: ISO/IEC 8824-1:2021, 3.8.67 — modified, new Note 1 to entry added.]

3.11
SEQUENCE-OF **type**
type defined by referencing a single *component type* (3.4)

Note 1 to entry: Each value in the sequence-of type is an ordered list of zero, one or more values of the component type.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.68 — modified, Note 1 to entry added.]

3.12
SET **type**
type defined by referencing a fixed, unordered, list of types (some of which may be declared to be optional); each value in the set type is an unordered list of values, one from each *component type* (3.4)

Note 1 to entry: Where a component type is declared to be optional, a value of the set type need not contain a value of that component type.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.72]

3.13
SET-OF **type**
types defined by referencing a single *component type* (3.4); each value in the set-of type is an unordered list of zero, one or more values of the component type.

[SOURCE: ISO/IEC 8824-1:2021, 3.8.73]

3.14
simple type
type defined by directly specifying the set of their values

[SOURCE: ISO/IEC 8824-1:2021, 3.8.74]

3.15
single-level data type
data type (3.5) which is a *sequence* (3.10) or *sequence-of type* (3.11) defined by referencing a *simple type* (3.14) or a *subtype* (3.16) of a simple type

3.16
subtype (of a parent type)
type whose values are a subset (or the complete set) of the values of some other type (the *parent type*) (3.9)

[SOURCE: ISO/IEC 8824-1:2021, 3.8.76]

3.17
three-level data type
data type (3.5) which is a *choice* (3.2), *sequence* (3.10) or *sequence-of type* (3.11) defined by referencing a *two-level data type* (3.18)

3.18
two-level data type
data type (3.5) which is a *choice* (3.2), *sequence* (3.10) or *sequence-of type* (3.11) defined by referencing *a single-level data type* (3.15)

4 Abbreviated terms

ASN.1	abstract syntax notation one
BCD	binary coded decimal
CO	carbon monoxide
CO_2	carbon dioxide
EFC	electronic fee collection
GNSS	global navigation satellite system
HC	hydrocarbon
ICC	integrated circuit(s) card
LAC	localisation augmentation communication
NO_X	nitrogen oxides
OBU	onboard unit

5 EFC common data object definitions

5.1 General

In this clause, the structure of all EFC common data objects is described. This is formally defined in Annex A in terms of data type definitions. In addition to the structure description, each data object is also given a semantics.

Each one of the common data types defined herein is used by more than one standard in the EFC context. These standards may also define their own data types when no one of the common data types defined herein satisfies their need.

The definitions of the EFC common data types are ordered according to their data type level:

- first subtypes based on simple data types (e.g. INTEGER or OCTET STRING),
- then two-level data types,
- then three-level data types,
- then complex data types.

Data types are ordered alphabetically inside each level.

5.2 Subtypes of simple data types

5.2.1 AccountStatus

The data type `AccountStatus` shall be based on a simple type as described in Table 1.

Table 1 — AccountStatus

Subtype	Parent type	Semantics
-	`INTEGER`	`AccountStatus` provides the status of the user's account. The following values are assigned: — ok, — low, — empty, — negative.

5.2.2 ActualNumberOfPassengers

The data type `ActualNumberOfPassengers` shall be based on a subtype as described in Table 2.

Table 2 — ActualNumberOfPassengers

Subtype	Parent type	Semantics
`Int1Unsigned`		`ActualNumberOfPassengers` represents the actual number of passengers (i.e. human beings) present in the vehicle, incl. the driver. This information can affect the applicability of tolls or the value of the tariff to be applied, e.g. in High Occupancy Tolling or High Occupancy Vehicle lanes.

5.2.3 FutureCharacteristics

The data type `FutureCharacteristics` shall be based on a simple constrained type as described in Table 3.

Table 3 — FutureCharacteristics

Subtype	Parent type	Semantics
-	INTEGER	FutureCharacteristics provides information reserved for future use, stored in one octet, that will be defined in future versions of this document. The following values are assigned: — noEntry: means information is not available; — airSuspension: means the vehicle uses air suspensions;

5.2.4 Altitude

The data type Altitude shall be based on a simple type as described in Table 4.

Table 4 — Altitude

Subtype	Parent type	Semantics
Int2Signed		Altitude provides the ellipsoidal height (in 0,25 metre units) above or below the WGS84[5] ellipsoid of the geographical point. The range in metres is from -8 192,00 to +8 191,75. NOTE WGS84[5] represents a broadly adopted global geodetic reference system for the Earth for practical applications of mapping, geopositioning and navigation. Other terrestrial reference frames exist, notably the International Terrestrial Reference Frame (ITRF, the latest currently being ITRF2014, with ITRF2020 under preparation). It is possible to convert between the most commonly used terrestrial reference frames and the differences between them are typically in the order of centimetres. The international terrestrial reference frame is becoming increasingly recognized and used as the primary reference frame. All recent and up-to-date Global Navigation Satellite System (GNSS) specific terrestrial reference frames (WGS 84 for GPS, PZ-90 for GLONASS, the GTRF for Galileo, CGCS2000 for BeiDou, and the JGS for QZSS) are aligned to a primary ITRS[6] realization, according to ISO 19161-1:2020, Annex C.

5.2.5 CO2EmissionValue

The data type CO2EmissionValue shall be based on a subtype as described in Table 5.

Table 5 — CO2EmissionValue

Subtype	Parent type	Semantics
Int2		CO2EmissionValue represents the vehicle's CO_2 emission value according to vehicle registration documents, in g/km.

5.2.6 ContractAuthenticator

The data type ContractAuthenticator shall be based on a simple type as described in Table 6.

Table 6 — ContractAuthenticator

Subtype	Parent type	Semantics
-	OCTET STRING	ContractAuthenticator is an authenticator calculated by the toll service provider when issuing the contract, to prevent tampering with contract data.

5.2.7 ContractSerialNumber

The data type ContractSerialNumber shall be based on a subtype as described in Table 7.

Table 7 — ContractSerialNumber

Subtype	Parent type	Semantics
Int4Unsigned		ContractSerialNumber is an integer designating the individual contract, assigned at the discretion of the toll service provider.

5.2.8 CopValue

The data type CopValue shall be based on a simple type as described in Table 8.

Table 8 — CopValue

Subtype	Parent type	Semantics
-	INTEGER NOTE It is of type ENUMERATED in ISO 14906.	CopValue represents the vehicle's carbon dioxide pollution values as defined in Directive 2003/127/EC.[11] The following values are assigned: — noEntry, value not defined; — co2class1, for pollution values below 101 g/km; — co2class2, for pollution value from 101 to 120. g/km; — co2class3, for pollution values from 121 to 140 g/km; — co2class4, for pollution values from 141 to 160 g/km; — co2class5, for pollution values from 161 to 200 g/km; — co2class6, for pollution values from 201 to 250 g/km; — co2class7, for pollution values above 250 g/km.

5.2.9 CountryCode

The data type CountryCode shall be based on a simple type as described in Table 9.

Table 9 — CountryCode

Subtype	Parent type	Semantics
-	BITSTRING	countryCode represents a ISO 3166-1 country code. Values are encoded in accordance with the ITA-2 encoding of the ISO 3166-1 country code. EXAMPLE 1 Austria (AT) = 11000 00001. EXAMPLE 2 Belgium (BE) = 10011 10000.

5.2.10 DetectionMode

The data type DetectionMode shall be based on a simple type as described in Table 10.

Table 10 — DetectionMode

Subtype	Parent type	Semantics
-	INTEGER	DetectionMode indicates how the charge object was detected. The following values are assigned: — measured: The charge object was detected by evaluation of positioning data using regular rules defined for recognizing charge objects; — inferred: The charge object was not detected by evaluation of positioning data, but was inferred from the overall trip logic; — lac: For implementation reasons in special cases, the normal charge object detection technology could be supported by localisation augmentation communication (LAC) beacons, which directly communicate to a given charge object its location, usually using short-range communication technology.

5.2.11 DescriptiveCharacteristics

The data type DescriptiveCharacteristics shall be based on a simple type as described in Table 11.

Table 11 — DescriptiveCharacteristics

Subtype	Parent type	Semantics
-	INTEGER	DescriptiveCharacteristics provides information about the vehicle's shape. The following value is assigned: — noEntry, indicating that the descriptive characteristics are not present.

5.2.12 EmissionUnit

The data type EmissionUnit shall be based on a simple constrained type as described in Table 12.

Table 12 — EmissionUnit

Subtype	Parent type	Semantics
-	INTEGER	EmissionUnit represents the unit of emission values as an integer on one octet. The following values are assigned: — mg/km, indicating the emission values are expressed in milligrams per kilometre, — mg/Kwh, indicating the emission values are expressed in milligrams per kilowatt per hour.

5.2.13 EngineCharacteristics

The data type EngineCharacteristics shall be based on a simple type as described in Table 13.

Table 13 — EngineCharacteristics

Subtype	Parent type	Semantics
-	INTEGER	EngineCharacteristics provides information about the vehicle's engine type and fuel characteristics.

The following values have been assigned:

- `noEntry`, indicating that no information is available;
- `noEngine`, indicating that the vehicle has no engine;
- `petrolUnleaded`, indicating engine operating with unleaded petrol;
- `petrolLeaded`, indicating engine operating with leaded petrol;
- `diesel`, indicating engine operating with diesel;
- `lpg`, indicating engine operating with liquefied petroleum gas;
- `battery`, indicating vehicle powered exclusively by battery;
- `solar`, indicating engine operating with solar energy;
- `hybrid`, value kept for legacy compatibility, more differentiated values are available;
- `hydrogen`, indicating engine operating with hydrogen;
- `multiFuel`, indicating multi fuel engine;
- `bivalentPetrolLpg`, indicating bivalent operating engine with petrol or liquefied petroleum gas;
- `bivalentPetrolCng`, indicating bivalent operating engine with petrol or compressed natural gas;
- `combinedPetrolElectric`, indicating combined operation with petrol and electric engine;
- `cng`, indicating engine operating with compressed natural gas;
- `lng`, indicating engine operating with liquefied natural gas;
- `combinedDieselElectric`, indicating combined operation of diesel and electric engine;
- `combinedHydrogenElectric`, indicating combined operation of hydrogen and electric engine;
- `bivalentHydrogenPetrol`, indicating bivalent operating engine with hydrogen or petrol;
- `bivalentHydrogenPetrolElectricEngine`, indicating bivalent operating engine with hydrogen or petrol combined with electric engine;
- `fuelCellHydrogen`, indicating fuel cell with hydrogen as primary energy source and electric engine;
- `fuelCellPetrol`, indicating fuel cell with petrol as primary energy source and electric engine;
- `fuelCellMethanol`, indicating fuel cell with methanol as primary energy source and electric engine;
- `fuelCellEthanol`, indicating fuel cell with ethanol as primary energy source and electric engine;
- `fuelCellDiesel`, indicating fuel cell with diesel as primary energy source and electric engine;
- `combinedMultiFuelElectricEngine`, indicating combined operation of multi fuel and electric engine;
- `combinedCngElectricEngine`, indicating combined operation with compressed natural gas and electric engine;
- `combinedLngElectricEngine`, indicating combined operation with liquefied natural gas and electric engine;
- `petrolEthanol`, indicating fuel mix of petrol and mainly ethanol, e.g. E85;
- `combinedLpgElectricEngine`, indicating combined operation of liquefied petroleum gas and electric engine;

- `hybridPetrolExternalBattery`, indicating hybrid drive with petrol and external rechargeable battery (plug-in hybrid);
- `hybridDieselExternalBattery`, indicating hybrid drive with diesel and external rechargeable battery (plug-in hybrid);
- `hybridLpgExternalBattery`, indicating hybrid drive with liquefied petroleum gas and external rechargeable battery (plug-in hybrid);
- `hybridHydrogenExternalBattery`, indicating hybrid drive with hydrogen and external rechargeable battery (plug-in hybrid);
- `hybridMultiFuelExternalBattery`, indicating hybrid drive with multi fuel and external rechargeable battery (plug-in hybrid);
- `hybridCngExternalBattery`, indicating hybrid drive with compressed natural gas and external rechargeable battery (plug-in hybrid);
- `hybridLngExternalBattery`, indicating hybrid drive with liquefied natural gas and external rechargeable battery (plug-in hybrid);
- `hybridBivalentHydrogenPetrolExternalBattery`, indicating hybrid drive with bivalent operating hydrogen and petrol engine and external rechargeable battery (plug-in hybrid);
- `hydrogenCng`, indicating a fuel mix of hydrogen and compressed natural gas;
- `hydrogenLng`, indicating a fuel mix of hydrogen and liquefied natural gas;
- `hybridHydrogenCngExternalBattery`, indicating hybrid drive with hydrogen and compressed natural gas and external chargeable battery (plug-in hybrid);
- `hybridHydrogenLngExternalBattery`, indicating hybrid drive with hydrogen and liquefied natural gas and external chargeable battery (plug-in hybrid);
- `ethanol`, indicating ethanol or fuel mix of ethanol and other fuel (except petrol) or additives, e.g. E95;
- `hybridFuelCellHydrogen`, indicating hybrid drive with fuel cell (electric engine) and hydrogen (combustion engine);
- `hybridFuelCellHydrogenExternalBattery`, indicating hybrid drive with fuel cell (electric engine) and hydrogen (combustion engine) and external chargeable battery (plug-in hybrid);
- `dualFuelLngDiesel`, indicating dual operation with liquefied natural gas and diesel;
- `electricExternal`, indicating electric engine with external power supply;
- `biogas`, indicating mixture of different gases produced by the breakdown of organic matter;
- `bioDiesel`, indicating vegetable oil- or animal fat-based diesel fuel;
- `bioPetrol`, indicating petrol fully or partly based on vegetable sources;
- `bivalentPetrolBiogas`, indicating bivalent operating engine with petrol or biogas;
- `combinedBiogasElectricEngine`, indicating combined operation of biogas and electric engine;
- `dualFuelCngDiesel`, indicating dual operation with compressed natural gas and diesel;
- `other`.

5.2.14 EquipmentIccId

The data type `EquipmentIccId` shall be based on a simple type as described in Table 14.

Table 14 — EquipmentICC-Id

Subtype	Parent type	Semantics
-	`OCTET STRING`	`EquipmentIccId` represents the identification number of the integrated circuit(s) card (ICC).

5.2.15 EquipmentObuId

The data type `EquipmentObuId` shall be based on a simple type as described in Table 15.

Table 15 — EquipmentObuId

Subtype	Parent type	Semantics
-	`OCTET STRING`	`EquipmentObuId` represents the unique identification of the OBU within the context of the associated manufacturer. NOTE As an example of usage, EN 15509 expresses this value using four octets. According to the used encoding rules, an additional octet indicating the length may be inserted.

5.2.16 EquipmentStatus

The data type `EquipmentStatus` shall be based on a constrained simple type as described in Table 16.

Table 16 — EquipmentStatus

Subtype	Parent type	Semantics
-	`BITSTRING`	`EquipmentStatus` provides operator-specific EFC application-related information pertaining to the status of the equipment, stored in a bitstring of two octets.

5.2.17 EuroValue

The data type `EuroValue` shall be based on a simple type as described in Table 17.

Table 17 — EuroValue

Subtype	Parent type	Semantics

-	INTEGER
	NOTE It is of ENUMERATED type in ISO 14906.
		EuroValue represents the vehicle's Euro emission category as defined in the EU Directives and Regulations cited in references [13],[14],[15],[16] and[17].
		The following values are assigned:
		— noEntry, indicating that no Euro emission category value is provided;
		— euro1, indicating Euro emission category 1;
		— euro2, indicating Euro emission category 2;
		— euro3, indicating Euro emission category 3;
		— euro4, indicating Euro emission category 4;
		— euro5, indicating Euro emission category 5;
		— euro6, indicating Euro emission category 6;
		— euro7, indicating Euro emission category 7;
		— Eev, indicating Enhanced Environmentally Friendly Vehicle class.
		NOTE The value euro-7 is reserved for upcoming Euro emission category 7.

5.2.18 IssuerIdentifier

The data type IssuerIdentifier shall be based on a simple type as described in Table 18.

Table 18 — IssuerIdentifier

Subtype	Parent type	Semantics
-	INTEGER	IssuerIdentifier provides the identifier of an organization according to the corresponding registry. NOTE See Table 66 for an example of a specific registry usage.

5.2.19 Latitude

The data type Latitude shall be based on a subtype as described in Table 19.

Table 19 — Latitude

Subtype	Parent type	Semantics
Int4Signed		Latitude provides the latitude (in micro degrees) of the geographical point, with a range of 90° in the north or south hemisphere. Positive values are used for latitude north of the Equator, negative values are used for latitude south of the Equator.

5.2.20 DistanceUnit

The data type DistanceUnit shall be based on a simple type as described in Table 20.

Table 20 — DistanceUnit

Subtype	Parent type	Semantics
-	INTEGER	DistanceUnit identifies units of distance. The following values are allowed: — kilometres, — metres, — decimetres, — centimetres, — millimetres. Values identifying non-ISO units are maintained in the ASN.1 definition for backward compatibility, although deprecated.

5.2.21 LocalVehicleClassId

The data type LocalVehicleClassId shall be based on a subtype as described in Table 21.

Table 21 — LocalVehicleClassId

Subtype	Parent type	Semantics
Int2Unsigned		LocalVehicleClassId provides the unique identifier of the local vehicle class.

5.2.22 LocationClassId

The data type LocationClassId shall be based on a subtype as described in Table 22.

Table 22 — LocationClassId

Subtype	Parent type	Semantics
Int4Unsigned		LocationClassId provides the unique identifier of the location class.

5.2.23 Longitude

The data type Longitude shall be based on a subtype as described in Table 23.

Table 23 — Longitude

Subtype	Parent type	Semantics
Int4Signed		Longitude provides the longitude (in micro degrees) of the geographical point, providing a range of 180° to the east or to the west of the prime meridian (which passes through the Royal Observatory, Greenwich, England). Negative values are used for longitudes to the west, positive values are used for longitudes to the east.

5.2.24 PaymentSecurityData

The data type PaymentSecurityData shall be based on a simple type as described in Table 24.

Table 24 — PaymentSecurityData

Subtype	Parent type	Semantics
-	OCTET STRING	PaymentSecurityData provides security-related data for the authentication of the data integrity.

5.2.25 PayUnit

The data type PayUnit shall be based on a simple constrained type as described in Table 25.

Table 25 — PayUnit

Subtype	Parent type	Semantics
-	OCTET STRING	PayUnit represents the unique designation of a currency as in accordance with ISO 4217. The code can also express a company-specific token or a "charging unit code" as used in the freight unit in which the fee is expressed. PayUnit provides the possibility to express a payment means value in multiples or fractions of a currency. The following values are assigned: — Currency in main units; — Currency in minor units of 10 :1 ('dime'); — Currency in minor units of 100 :1 ('cents'); — Currency in minor units of 1 000 :1; — Currency in 'major' units / 10 (e.g. 10 Euros); — Currency in 'major' units / 100 (e.g. 100 US Dollars); — Currency in 'major' units / 1 000; — Currency in 'major' units / 10 000; — Currency in 'major' units / 100 000; — Currency in minor units of 10 000 :1; — Currency in minor units of 100 000 :1; — Currency in minor units of 1 000 000 :1; — Tokens, specifying a Purse Provider specific coding; — Charging Unit Codes denoting quantification of the service provided (e.g. man-hours).

5.2.26 PersonalAccountNumber

The data type PersonalAccountNumber shall be based on a simple type as described in Table 26.

Table 26 — PersonalAccountNumber

Subtype	Parent type	Semantics

-	`OCTET STRING`	`PersonalAccountNumber` represents the personal account number structure in accordance with ISO/IEC 7812-1 using Binary Coded Decimal (BCD) sub-structures.
		NOTE Personal account number has the following structure:
		— Issuer identifier number (“IIN”);
		— Major industry identifier (MII, 1 BCD), the following values are assigned:
		— 0: reserved for future use by ISO/TC68;
		— 1: airline sector;
		— 2: extended airline sector;
		— 3: travel and tourism sector;
		— 4: financial banking sector;
		— 5: financial banking sector;
		— 6: commerce and banking sector;
		— 7: petrol industry sector;
		— 8: telecommunication sector;
		— 9: reserved for national use.
		— Issuer identifier (5 BCD in the second edition of ISO/IEC 7812-1).
		— Account number (max 12 BCD);
		— Control digit (1 BCD);
		— Padding bits, set to 1'B, in order to accomplish a total length of 10 octets.

5.2.27 ReceiptAuthenticator

The data type `ReceiptAuthenticator` shall be based on a simple type as described in Table 27.

Table 27 — ReceiptAuthenticator

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptAuthenticator` contains an Authenticator over some Attributes of the data group Receipt, calculated by the Session-ServiceProvider.

5.2.28 ReceiptDistance

The data type `ReceiptDistance` shall be based on a subtype as described in Table 28.

Table 28 — ReceiptDistance

Subtype	Parent type	Semantics
`Int3Unsigned`		`ReceiptDistance` contains the total distance covered by the vehicle, since the beginning of its existence, in units of 100 metres expressed as an Integer.

5.2.29 ResultFin

The data type `ResultFin` shall be based on a simple constrained type as described in Table 29.

Table 29 — ResultFin

Subtype	Parent type	Semantics

-	`OCTET STRING`	`ResultFin` represents in one octet the operational result of the financial transaction. The following values are assigned:
		— Most significant 4 bits: 0 means OK:
		— '0x'H OK;
		— Most significant 4 bits > 0 means not OK:
		— '1x'H Not OK, not specified further;
		— '2x'H Not OK, Abnormal (First or Previous) Event;
		— '3x'H Not OK, Contract not accepted;
		— '4x'H Not OK, Account or Purse not accepted;
		— Least Significant 4 bits mean:
		— 'x0'H not specified further;
		— 'x1'H Balance close to zero;
		— 'x2'H Balance now negative;
		— 'x3'H Balance overflow;
		— 'x4'H Provider not accepted;
		— 'x5'H Authentication failure;
		— 'x6'H Vehicle class incorrect.

5.2.30 ReceiptIccId

The data type `ReceiptIccId` shall be based on a simple type as described in Table 30.

Table 30 — ReceiptIccId

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptIccId` provides the identification number of the smart card (ICC) used in the session.

5.2.31 ReceiptObuId

The data type `ReceiptObuId` shall be based on a simple type as described in Table 31.

Table 31 — ReceiptObuId

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptOBUId` represents the serial number of the OBU used in the session, unique within the context of the manufacturer.

Note This type is only maintained for backward compatibility reasons. Its usage is deprecated.

5.2.32 ResultOp

The data type `ResultOp` shall be based on a simple type as described in Table 32.

Table 32 — ResultOp

Subtype	Parent type	Semantics
-	INTEGER	ResultOp represents the operational result of the EFC session. The following values are assigned: — correctTransaction, the transaction has been accepted; — obeStatusNotAccepted, not acceptable OBEStatus; — equipmentStatusNotAccepted, not acceptable equipmentStatus; — contractNotInWhiteList, contract shall be in white list; — contractIdentifierInBlackList, contract blacklisted; — contractIdentifierNotCorrect, contract identifier not valid; — expiredContract, contract expired; — contractRestrictionsNotFulfilled, restrictions in the contract not fulfilled; — claimedVehicleCharacteristicsNotValid, vehicle characteristics not conforming to those detected; — vehicleClassAuthenticationFailed, failure in vehicle class authentication; — entryVehicleClassDifferentFromExitVehicleClass, vehicle class in exit not conforming to vehicle class in entry; — entryReceiptMissing, missing proof of entry; — entryReceiptNotValid, invalid proof of entry; — entryTollStationNotValid, invalid entry station; — equipmentNotCertified, equipment not certified; — timeDifference, problem with the time difference of the two last receipts; — accessCredentialsNotAccepted, invalid access credentials; — contractAuthenticatorNotAccepted, invalid contract authenticator; — receiptAuthenticatorNotAccepted, invalid receipt authenticator; — claimedVehicleCharacteristicsMissing, vehicle characteristics not detected; — paymentMeansNotAccepted, payment means not accepted; — paymentAuthenticatorNotAccepted, invalid payment authenticator; — paymentMeansInBlackList, payment means blacklisted; — paymentMeansNotCorrect, incorrect payment means; — expiredPaymentMeans., expired payment means; — paymentMeansRestrictionsNotFulfilled, restrictions in payment means not fulfilled.

5.2.33 ReceiptServiceSerialNumber

The data type `ReceiptServiceSerialNumber` shall be based on a subtype as described in Table 33.

Table 33 — ReceiptServiceSerialNumber

Subtype	Parent type	Semantics
`Int3Unsigned`		`ReceiptServiceSerialNumber` represents a specific serial number of the EFC session. NOTE The issuer of this number can be, for example, in a DSRC-based toll system, an RSE.

5.2.34 ReceiptText

The data type `ReceiptText` shall be based on a simple type as described in Table 34.

Table 34 — ReceiptText

Subtype	Parent type	Semantics
-	`OCTET STRING`	`ReceiptText` contains plain text decodable by the OBE.

5.2.35 StationType

The data type `StationType` shall be based on a simple type as described in Table 35.

Table 35 — StationType

Subtype	Parent type	Semantics
-	`INTEGER`	`StationType` represents the type of EFC Station. The following values are assigned: — unspecified, meaning that no type is specified; — closedEntryWithPayment, a payment entry station in a closed system; — closedEntryWithoutPayment, an entry station in a closed system without payment; — closedTransit, a transit station in a closed system; — closedExit, an exit station in a closed system; — closedCredit, a station where payment is by means of a rechargeable OBE; — mixed, a multi-purpose station; — passage, an exit station in an open system; — checkpoint, a checkpoint station; — reload, a station where a rechargeable OBE can be recharged.

5.2.36 TariffClassId

The data type `TariffClassId` shall be based on a subtype as described in Table 36.

Table 36 — TariffClassId

Subtype	Parent type	Semantics
Int4Unsigned		TariffClassId provides the unique identifier of the tariff class.

5.2.37 Time

The data type Time shall be based on a subtype as described in Table 37.

Table 37 — Time

Subtype	Parent type	Semantics
Int4Unsigned		Time represents the number of seconds since midnight at the start of 1st January 1970.

5.2.38 TimeClassId

The data type TimeClassId shall be based on a subtype as described in Table 38.

Table 38 — TimeClassId

Subtype	Parent type	Semantics
Int2Unsigned		TimeClassId provides the unique identifier of the time class.

5.2.39 TimeUnit

The data type TimeUnit shall be based on a simple type as described in Table 39.

Table 39 — TimeUnit

Subtype	Parent type	are
-	INTEGER	TimeUnit provides the unit of time. The following values are assigned: — seconds, — minutes, — hours, — days, — months.

5.2.40 TrailerType

The data type TrailerType shall be based on a simple type as described in Table 40.

Table 40 — TrailerType

Subtype	Parent type	Semantics
-	INTEGER	TrailerType provide information about the type of trailer. The following values are assigned: — notPresent, indicating trailer not attached; — trailer, indicating trailer (also known as pull-bar trailer) attached; — semitrailer, indicating semitrailer (also known as articulate trailer) attached.

5.2.41 TyreConfiguration

The data type TyreConfiguration shall be based on a simple type as described in Table 41.

Table 41 — TyreConfiguration

Subtype	Parent type	Semantics
-	INTEGER	TyreConfiguration provides information about the tyre configuration on the axles of the vehicle. The following values are assigned: — notSpecified, indicating no values are provided; — singleTyre, indicating single tyre on all axles; — dualTyres, indicating dual tyres on at least one axle.

5.2.42 UserClassId

The data type UserClassId shall be based on a subtype as described in Table 42.

Table 42 — UserClassId

Subtype	Parent type	Semantics
Int1Unsigned		UserClassId provides the unique identifier of the user class.

5.2.43 VehicleAuthenticator

The data type VehicleAuthenticator shall be based on a simple type as described in Table 43.

Table 43 — VehicleAuthenticator

Subtype	Parent type	Semantics
-	OCTET STRING	VehicleAuthenticator provides an authenticator calculated by the entity entering the data elements at time of entry or modification.

5.2.44 VehicleClass

The data type VehicleClass shall be based on a simple type as described in Table 44.

Table 44 — VehicleClass

Subtype	Parent type	Semantics
-	Int1Unsigned	VehicleClass provides a toll service provider specific information pertaining to the vehicle. NOTE A more specific definition and an example of usage of VehicleClass is available in EN 15509.

5.2.45 VehicleCurrentMaxTrainWeight

The data type VehicleCurrentMaxTrainWeight shall be based on a subtype as described in Table 45.

Table 45 — VehicleCurrentMaxTrainWeight

Subtype	Parent type	Semantics
Int2Unsigned		VehicleCurrentMaxTrainWeight represents the maximum permissible weight of the complete vehicle train that is currently in operation, which shall be stated in 10 kg units, in accordance with ISO 1176, rounded down to the next 10 kg step. NOTE This weight can be lower than VehicleTrainMaximumWeight as it represents the current maximum train weight and not the maximum weight by design.

5.2.46 VehicleTotalDistance

The data type VehicleTotalDistance shall be based on a subtype as described in Table 46.

Table 46 — VehicleTotalDistance

Subtype	Parent type	Semantics
Int4Unsigned		VehicleTotalDistance represents the total travelled distance as measured by the vehicle, in 10 m resolution, continuously incremented. NOTE The initial value of this data type can be either the value zero or the vehicle's kilometre reading at the time of personalization of the OBE.

5.2.47 VehicleWeightLaden

The data type VehicleWeightLaden shall be based on a subtype as described in Table 47.

Table 47 — VehicleWeightLaden

Subtype	Parent type	Semantics
Int2Unsigned		VehicleWeightLaden represents the actual weight of vehicle including load in 10 kg units, rounded down to the next 10 kg step.

5.2.48 WeekDay

The data type WeekDay shall be based on a simple type as described in Table 48.

Table 48 — Weekday

Subtype	Parent type	Semantics
-	INTEGER	Weekday represents the day of the week. The following values are assigned: — Monday, — Tuesday, — Wednesday, — Thursday, — Friday, — Saturday, — Sunday.

5.3 Single level data types

5.3.1 AbsolutePosition2d

The data type AbsolutePosition2d is an ordered list and shall consist of two components that identify a position in a 2-dimensions map as described in Table 49.

Table 49 — AbsolutePosition2d

Component name	Component type	Semantics
gnssLon	Longitude, see Table 23	gnssLon provides the longitudinal coordinate of a geodetic position.
gnssLat	Latitude, see Table 19	gnssLat provides the latitudinal coordinate of a geodetic position.

5.3.2 AbsolutePosition3d

The data type AbsolutePosition3d is an ordered list and shall consist of two mandatory components and one optional component that identify a position in a 3-dimensional map as described in Table 50.

Table 50 — AbsolutePosition3d

Component name	Component type	Semantics
longitude	Longitude, see Table 23	longitude provides the longitudinal coordinate of a point.
latitude	Latitude, see Table 19	latitude provides the latitudinal coordinate of a point.
altitude	Altitude, optional, see Table 4	altitude provides the altitude of a point.

5.3.3 AxleWeightLimit

The data type AxleWeightLimit is an ordered list and shall consist of two components that identify an axle of a vehicle and its related weight limit, as described in Table 51.

Table 51 — AxleWeightLimit

Component name	Component type	Semantics
axleNumber	Int1Unsigned	axleNumber indicates the ordinal number of the vehicle axle, counted from the front axle.
maxLadenWeightOnAxle	Int2Unsigned	maxLadenWeightOnAxle represents the technically permissible maximum laden weight on the axle of the vehicle, in 10 kg units, rounded down to the next 10 kg step.

5.3.4 AxleWeightLimits

The data type AxleWeightLimits is an ordered list and shall consist of five components that identify the weight limits on five axles, as described in Table 52.

Table 52 — AxleWeightLimits

Component name	Component type	Semantics
maxLadenWeightOnAxle1	Int2Unsigned	MaxLadenWeightOnAxle1 represents the technically permissible maximum laden weight on axle 1 of the vehicle, in 10 kg units, rounded down to the next 10 kg step, according to the registration certificate of the vehicle.
maxLadenWeightOnAxle2	Int2Unsigned	MaxLadenWeightOnAxle2 represents the technically permissible maximum laden weight on axle 2 of the vehicle, in 10 kg units, rounded down to the next 10 kg step, according to the registration certificate of the vehicle.
maxLadenWeightOnAxle3	Int2Unsigned	MaxLadenWeightOnAxle3 represents the technically permissible maximum laden weight on axle 3 of the vehicle, in 10 kg units, rounded down to the next 10 kg step, according to the registration certificate of the vehicle.
maxLadenWeightOnAxle4	Int2Unsigned	MaxLadenWeightOnAxle4 represents the technically permissible maximum laden weight on axle 4 of the vehicle, in 10 kg units, rounded down to the next 10 kg step, according to the registration certificate of the vehicle.
maxLadenWeightOnAxle5	Int2Unsigned	MaxLadenWeightOnAxle5 represents the technically permissible maximum laden weight on axle 5 of the vehicle, in 10 kg units, rounded down to the next 10 kg step, according to the registration certificate of the vehicle.

5.3.5 DateCompact

The data type DateCompact is an ordered list and shall consist of three components that identify a year, a month and a day, as described in Table 53.

Table 53 — DateCompact

Component name	Component type	Semantics
year	INTEGER	year represents the year of the date in the Gregorian calendar.
month	INTEGER	month represents the month of the date in the Gregorian calendar.
day	INTEGER	day represent the day of the date in the Gregorian calendar.

5.3.6 DieselEmissionValues

The data type DieselEmissionValues is an ordered list and shall consist of two components that identify the emission value, as described in Table 54.

Table 54 — DieselEmissionValues

Component name	Component type	Semantics
particulate	Particulate (see Table 62)	particulate represents the type and value of the particulate, as defined in Table 62.
absorptionCoeff	Int2Unsigned	absorptionCoeff represents the corrected absorption coefficient for diesel, , according to the registration certificate of the vehicle, in 10^{-3} m^{-1}.

5.3.7 DriverCharacteristics

The data type DriverCharacteristics is an ordered list of two components that identify the driver's characteristics, as described in Table 55.

Table 55 — DriverCharacteristics

Component name	Component type	Semantics
driverClass	Int1Unsigned	driverClass provides a description of the driver's characteristics as pertinent to the calculation of the tariff; contract provider specific coding.
tripPurpose	Int1Unsigned	tripPurpose provides a parameter indicating the purpose of the trip of the user as pertinent to the calculation of the tariff; contract provider specific coding.

5.3.8 Distance

The data type Distance is an ordered list of two components that identify a distance, as described in Table 56.

Table 56 — Distance

Component name	Component type	Semantics
distanceValue	Int4Unsigned	distanceValue provides the value of the distance.
distanceUnit	DistanceUnit, see Table 20	distanceUnit provides the unit of which the distance is measured. It defaults to kilometres.

5.3.9 Duration

The data type Duration is an ordered list of two components as described in Table 57.

Table 57 — Duration

Component name	Component type	Semantics
durationValue	Int4Unsigned	durationValue provides the length of time defining a duration.
durationUnit	TimeUnit	durationUnit provides the unit of time in which the duration is measured.

5.3.10 EngineDetails

The data type EngineDetails is an ordered list of two components that describe the characteristics of an engine, as described in Table 58.

Table 58 — EngineDetails

Component name	Component type	Semantics
engineCapacity	Int2Unsigned	EngineCapacity represents the capacity of the vehicle's engine in cm^3.
enginePower	Int2Unsigned	EnginePower represents the maximum net power of the vehicle's engine, in kW

5.3.11 ExhaustEmissionValues

The data type ExhaustEmissionValues is an ordered list of five components that specify emission values, as described in Table 59.

Table 59 — ExhaustEmissionValues

Component name	Component type	Semantics
unitType	EmissionUnit, see Table 12	EmissionUnit represents the unit associated to the values in the other components.
emissionCo	INTEGER	EmissionCo represents the exhaust emission of CO, according to vehicle registration documents, in 10^{-3} g/km or g/kWh.
emissionHc	Int2Unsigned	EmissionHc represents the exhaust emission of HC, according to vehicle registration documents, in 10^{-3} g/km or g/kWh.
emissionNox	Int2Unsigned	EmissionNox represents the exhaust emission of NO_X, according to vehicle registration documents, in 10^{-3} g/km or g/kWh.
emissionHcNox	Int2Unsigned	EmissionHcNox represents the exhaust emission of $HC+NO_X$, according to vehicle registration documents, in 10^{-3} g/km or g/kWh.

NOTE If the emissions are measured directly on the engine test bed, the values are declared in g/kWh.

5.3.12 NumberOfAxles

The data type NumberOfAxles shall consist of two components that specify the number of axles in the trailer and in the tractor, as described in Table 60.

Table 60 — NumberOfAxles

Component name	Component type	Semantics
trailerAxles	INTEGER	trailerAxles represents the number of axles of the trailer including drop axles. The value zero indicates no trailer present.
tractorAxles	INTEGER	tractorAxles represents the number of axles of the tractor including drop axles. The value zero indicates “not known”.

5.3.13 ObeId

The data type ObeId shall consist of two components that identify the manufacturer and the identifier of an OBU, as described in Table 61. An OBE may consist of separately manufactured parts, however it can be identified, for the purpose of this document, through the identifier of the OBU.

Table 61 — ObeId

Component name	Component type	Semantics
manufacturerId	Int2Unsigned	manufacturerId provides the unique identifier of the OBU manufacturer as defined according to ISO 14816. NOTE See www.itsstandards.eu/registries/register-of-manufacturers-cs2/ for a list of assigned values.
equipmentObuId	EquipmentObuId	Unique identifier as assigned by the manufacturer identified by component manufacturerId (see Table 15).

5.3.14 Particulate

The data type Particulate is an ordered list and shall consist of two components that identify the type and value of the particulate emitted by a diesel engine, as described in Table 62.

Table 62 — Particulate

Component name	Component type	Semantics
unitType	EmissionUnit	unitType represents the units in which the particulate is expressed, as defined in Table 12.
value	INTEGER	value represents the particulates for diesel, according to the registration certificate of the vehicle.

5.3.15 PassengerCapacity

The data type PassengerCapacity is an ordered list of two components that indicate a vehicle's capacity in terms of passengers, as described in Table 63.

Table 63 — PassengerCapacity

Component name	Component type	Semantics
numberOfSeats	Int1Unsigned	NumberOfSeats represents the number of seats of the vehicle, including the driver's seat, according to the registration certificate of the vehicle.
numberOfStandingPlaces	Int1Unsigned	NumberOfStandingPlaces represents the number of standing places of the vehicle, according to the registration certificate of the vehicle.

5.3.16 PaymentFee

The data type PaymentFee is an ordered list of two components that specify a payment, as described in Table 64.

Table 64 — PaymentFee

Component name	Component type	Semantics
paymentFeeAmount	Int1Unsigned	paymentFeeAmount represents the value of the fee being charged.
paymentFeeUnit	PayUnit	paymentFeeUnit represents the unit in which the fee is expressed. See Table 25.

5.3.17 Period

The data type Period is an ordered list of two components that specify a period of time in terms of a start and of an end time, as described in Table 65.

Table 65 — Period

Component name	Component type	Semantics
beginOfPeriod	GeneralizedTime	beginOfPeriod defines the begin of a period.
endOfPeriod	GeneralizedTime	endOfPeriod defines the end of a period.

5.3.18 Provider

The data type Provider is an ordered list of two components that identify an organization, as described in Table 66.

Table 66 — Provider

Component name	Component type	Semantics
countryCode	CountryCode, see Table 9	countryCode identifies the country code of the country of the national registration administrator for issuers according to ISO 14816.
providerIdentifier	IssuerIdentifier, see Table 18	providerIdentifier identifies the organization according to the national ISO 14816 register for issuers. NOTE See https://www.itsstandards.eu/registries/register-of-nra-i-cs1/ for a list of national registration administrators and their respective registers. EXAMPLE An organization can be a toll charger or a toll service provider according to ISO 17573-1.

5.3.19 RelativePosition3d

The data type RelativePosition3d is an ordered list of two mandatory and one optional component that identify a position in a 3-dimensional map in relation to another position, as described in Table 67.

Table 67 — RelativePosition3d

Component name	Component type	Semantics
longitude	Int2Signed (-32768..32767)	longitude provides the relative longitudinal coordinate (in microdegrees) of a point with respect to a specified reference point.
latitude	Int2Signed (-32768..32767)	latitude provides the relative latitudinal coordinate (in microdegrees) of a point with respect to a specified reference point.
altitude	Int2Signed (-32768..32767)	altitude provides the relative altitude in 0,25 metre units) of a point with respect to a specified reference point.

5.3.20 SessionClass

The data type SessionClass is an ordered list of two components that identify a class and its related tariff for a vehicle in a given session, as described in Table 68.

Table 68 — SessionClass

Component name	Component type	Semantics
sessionTariffClass	Int1Unsigned	sessionTariffClass provides a toll service provider specific tariff class applied in the session.
sessionClaimedClass	Int1Unsigned	sessionClaimedClass provides a toll service provider specific vehicle class derived from claimed characteristics.

5.3.21 SessionLocation

The data type `SessionLocation` is an ordered list of two components that identify the location of a session in terms of a direction and of a lane number, as described in Table 69.

Table 69 — SessionLocation

Component name	Component type	Semantics
`ascendingKilometrage`	`BOOLEAN`	`ascendingKilometrage` provides a travel direction indicator. The following values are assigned: — `True`, means "according to ascending kilometrage of the road". — `False`, means "according to descending kilometrage of the road".
`laneCodeNumber`	`INTEGER`	laneCodeNumber provides a lane number.

5.3.22 SignedValue

The data type `SignedValue` is an ordered list of two components that are used to identify positive or negative values, as described in Table 70.

Table 70 — SignedValue

Component name	Component type	Semantics
`positive`	`INTEGER`	`positive` represents positive integer values.
`negative`	`INTEGER`	`negative` represents negative integer values.

5.3.23 SoundLevel

The data type `SoundLevel` is an ordered list of two components that identify the levels of sound of a vehicle, as described in Table 71.

Table 71 — SoundLevel

Component name	Component type	Semantics
`soundStationary`	`Int1Unsigned`	`soundStationary` represents the stationary sound of the vehicle in dB(A), according to the registration certificate of the vehicle.
`soundDriveBy`	`Int1Unsigned`	`soundDriveBy` represents the sound of the vehicle when driving in dB(A) , according to the registration certificate of the vehicle.

5.3.24 TariffClassDescription

The data type `TariffClassDescription` is an ordered list of five optional components, the first one of which is assigned a default value, that are used to specify a tariff class, as described in Table 72.

Table 72 — TariffClassDescription

Component name	Component type	Semantics
`tariffClassId`	`TariffClassId`	See TariffClassId semantics in Table 36. If this component is not specified, its value is defaulted to 0 (zero).
`localVehicleClassId`	`LocalVehicleClassId`	See LocalVehicleClassId semantics in Table 21.
`timeClassId`	`TimeClassId`	See TimeClassId semantics in Table 36
`locationClassId`	`LocationClassId`	See LocationClassId semantics in Table 22

Table 72 *(continued)*

Component name	Component type	Semantics
userClassId	UserClassId	See UserClassId semantics in Table 42

5.3.25 TimeCompact

The data type TimeCompact is an ordered list of three components that identify a time, as described in Table 73.

Table 73 — TimeCompact

Component name	Component type	Semantics
hours	INTEGER	hours expresses the number of hours of the time of the day after midnight.
mins	INTEGER	mins expresses the number of minutes after the hour of the time of the day.
doubleSecs	INTEGER	doubleSecs expresses the number of two-seconds after the minute of the time of the day.

5.3.26 TrailerDetails

The data type TrailerDetails is an ordered list of two components that specify a trailer, as described in Table 74.

Table 74 — TrailerDetails

Component name	Component type	Semantics
trailerType	TrailerType	See the semantic definition of TrailerType in Table 40.
trailerAxles	INTEGER	trailerAxles represents the number of axles of the trailer including drop axles. The value zero indicates no trailer present.

5.4 Two-level data types

5.4.1 AxlesWeightLimits

The data type AxlesWeightLimits is a list of components of the same type that specify the weight limits for all axles of a vehicle, as described in Table 75.

Table 75 — AxlesWeightLimits

Component name	Component type	Semantics
axleWeightLimits	List of AxleWeightLimit	The component axleWeightLimits contains the list of axle weight limits, see Table 51.

5.4.2 ChargeObjectId

The data type ChargeObjectId is an ordered list of one optional component and one mandatory component that identify a charge object, as described in Table 76.

Table 76 — ChargeObjectId

Component name	Component type	Semantics
`chargeObjectOperator`	`Provider`, optional, see Table 66	`chargeObjectOperator` identifies the entity operating the EFC regime in which the charge object is contained;
`chargeObjectDesignation`	`Int4Unsigned`	`chargeObjectDesignation` identifies the charge object within the given operator.

5.4.3 ContractValidity

The data type `ContractValidity` is an ordered list of two components that specify the validity of a contract, as described in Table 77.

Table 77 — ContractValidity

Component name	Component type	Semantics
`contractRestrictions`	`OCTET STRING`	`contractRestrictions` contains a toll service provider specific coding of the validity restrictions of a contract.
`contractExpiryDate`	`DateCompact`, see Table 53	`contractExpiryDate` contains the end-date of the validity of the contract. The validity ends at 24 h of the calendar day specified in `ContractExpiryDate`.

5.4.4 DateAndTime

The data type `DateAndTime` is an ordered list of two components that specify a date and a time, as described in Table 78.

Table 78 — DateAndTime

Component name	Component type	Semantics
`timeDate`	`DateCompact`, see Table 53	`timeDate` expresses the date according to the Gregorian calendar.
`timeCompact`	`TimeCompact`, see Table 73	`timeCompact` expresses time of the day in hours, min, and sec.

5.4.5 EnvironmentalCharacteristics

The data type `EnvironmentalCharacteristics` is an ordered list of two components that specify environmental characteristics of a vehicle, as described in Table 79.

Table 79 — EnvironmentalCharacteristics

Component name	Component type	Semantics
`euroValue`	`EuroValue`, see Table 17	`euroValue` provides information about the vehicle's EURO value, according to the registration certificate of the vehicle.
`copValue`	`CopValue`, see Table 8	`copValue` provides information about the vehicle's carbon dioxide pollution values, according to the registration certificate of the vehicle.

5.4.6 Lpn

The data type `Lpn` shall consist of three components that identify a licence plate number, as described in Table 80.

Table 80 — Lpn

Component name	Component type	Semantics
`countryCode`	`CountryCode`, see Table 9	`countryCode` represents the country code associated with the licence plate number
`alphabetIndicator`	`ENUMERATED`	`alphabetIndicator` indicates the type of alphabet used to represent the licence plate number. The following values are assigned: — `latinAlphabetNo1`: Latin-1 in accordance with ISO/IEC 8859-1, — `latinAlphabetNo2`: Latin-2 in accordance with ISO/IEC 8859-2, — `latinAlphabetNo3`: latin-3 in accordance with ISO/IEC 8859-3, — `latinAlphabetNo4`: Latin-4 in accordance with ISO/IEC 8859-4, — `latinCyrillicAlphabet`: Cyrillic in accordance with ISO/IEC 8859-5, — `latinArabicAlphabet`: Arabic in accordance with ISO/IEC 8859-6, — `latinGreekAlphabet`: Greek in accordance with ISO/IEC 8859-7, — `latinHebrewAlphabet`: Hebrew in accordance with ISO/IEC 8859-8, — `latinAlphabetNo5`: Latin-5 in accordance with ISO/IEC 8859-9, — `latinAlphabetNo6`: Latin-6 in accordance with ISO/IEC 8859-10, — `twoOctetBMP`: value deprecated and not to be used, — `fourOctetCanonical`: value deprecated and not to be used. NOTE ISO 14906:2018, Annex D provides a mapping from LatinAlphabetNo1.
`licencePlateNumber`	`OCTET STRING`	`licencePlateNumber` contains the licence plate number coded in accordance to the alphabet indicated by the component `alphabetIndicator`.

5.4.7 PaymentMeans

The data type `PaymentMeans` is an ordered list of three components that identify the means used to perform a payment, as described in Table 81.

Table 81 — PaymentMeans

Component name	Component type	Semantics
`personalAccountNumber`	`PersonalAccountNumber`	`personalAccountNumber` see Table 26
`paymentMeansExpiryDate`	`DateCompact`, see Table 53	`paymentMeansExpiryDate` represents the expiring date of the payment means. Payment means expires at 24 h of `PaymentMeansExpiryDate`.

Table 81 *(continued)*

Component name	Component type	Semantics
paymentMeansUsageControl	OCTET STRING	paymentMeansUsageControl indicates issuer's specified restrictions on the geographic usage and services allowed for the applications.

5.4.8 PaymentMeansBalance

The data type PaymentMeansBalance shall be based on a parent type as described in Table 82.

Table 82 — PaymentMeansBalance

Subtype	Parent type	Semantics
-	SignedValue, see Table 70	PaymentMeansBalance represents the Balance of a payment means in units of PaymentMeansUnit.

5.4.9 Point

The data type Point provides a choice of three components, each of which can be used to identify a point in a map, as described in Table 83.

Table 83 — Point

Component name	Component type	Semantics
pointIdentifier	Int4Unsigned	pointIdentifier provides the identifier of a point defined elsewhere.
absolutePointCoordinates	AbsolutePosition3d, see Table 50	absolutePointCoordinates provides the absolute coordinates of a point.
relativePointCoordinates	RelativePosition3d, see Table 67	relativePointCoordinates provides the relative coordinates of a point.

5.4.10 PurseBalance

The data type PurseBalance is an ordered list of two components that identify a value stored in an electronic purse, as described in Table 84.

Table 84 — PurseBalance

Component name	Component type	Semantics
purseValue	SignedValue, see Table 70	purseValue represents the balance on the (electronic) purse, expressed in a currency.
purseUnit	PayUnit, see Table 25	purseUnit represents the unit of currency associated to the balance.

5.4.11 TrailerCharacteristics

The data type TrailerCharacteristics is an ordered list of three components that specify the characteristics of a trailer, as described in Table 85.

Table 85 — TrailerCharacteristics

Component name	Component type	Semantics
trailerDetails	TrailerDetails, see Table 74	trailerDetails provides information on trailer presence, type and number of axles.

Table 85 *(continued)*

Component name	Component type	Semantics
trailerMaxLadenWeight	Int2Unsigned	trailerMaxLadenWeight represents the maximum permissible total weight of the trailer including payload, which shall be stated in 10 kg units, in accordance with ISO 1176, rounded down to the next 10k g step.
trailerWeightUnladen	Int2Unsigned	trailerWeightUnladen represents the nominal unladen weight of the trailer, which shall be stated in 10 kg units, in accordance with ISO 1176, rounded down to the next 10 kg step.

5.4.12 ValidityOfContract

The data type ValidityOfContract is an ordered list of two components that specify the validity of a user contract, as described in Table 86.

Table 86 — ValidityOfContract

Component name	Component type	Semantics
issuerRestrictions	OCTET STRING	contractRestrictions contains a service provider specific coding of the validity restrictions of a contract.
contractExpiryDate	DateCompact, see Table 53	contractExpiryDate contains the end-date of the validity of the contract. The validity ends at 24 h of the calendar day specified in ContractExpiryDate.

5.4.13 VehicleAxlesNumber

The data type VehicleAxlesNumber is an ordered list of two components that specify the number and tyre characteristics of a vehicle's axles, as described in Table 87.

Table 87 — VehicleAxlesNumber

Component name	Component type	Semantics
tyreConfiguration	TyreConfiguration	tyreConfiguration contains the characteristics of the tyres of the vehicle's axles, as defined in Table 41
numberOfAxles	NumberOfAxles	numberOfAxles contains the number of axles on the vehicle, as defined in Table 60.

5.4.14 VehicleDimensions

The data type VehicleDimensions is an ordered list of three components that specify a vehicle's dimensions, as described in Table 88.

Table 88 — VehicleDimensions

Component name	Component type	Semantics
vehicleLengthOverall	Int1Unsigned	vehicleLengthOverall represents the Nominal maximum overall length of the vehicle, which shall be stated in dm, according to ISO 612, rounded to the next dm.
vehicleHeightOverall	Int1Unsigned	vehicleHeightOverall represents the nominal overall unladen height, which shall be stated in dm, according to ISO 612, rounded to the next dm.
vehicleWidthOverall	Int1Unsigned	vehicleWidthOverall represents the nominal overall width, which shall be stated in dm, according to ISO 612, rounded to the next dm.

5.4.15 VehicleWeightLimits

The data type `VehicleWeightLimits` is an ordered list of three components that specify a vehicle's weight limits, as described in Table 89.

Table 89 — VehicleWeightLimits

Component name	Component type	Semantics
`vehicleMaxLadenWeight`	`Int2Unsigned`	`vehicleMaxLadenWeight` represents the maximum permissible total weight including payload, which shall be stated in 10 kg units, in accordance with ISO 1176, rounded down to the next 10 kg step.
`vehicleTrainMaximumWeight`	`Int2Unsigned`	`vehicleTrainMaximumWeight` represents the maximum permissible weight of the complete vehicle train, which shall be stated in 10 kg units, in accordance with ISO 1176, rounded down to the next 10 kg step.
`vehicleWeightUnladen`	`Int2Unsigned`	`vehicleWeightUnladen` represents the nominal unladen weight, which shall be stated in 10 kg units, in accordance with ISO 1176, rounded down to the next 10 kg step.

5.5 Three-level data types

5.5.1 EfcContextMark

The data type `EfcContextMark` is an ordered list of three components that identify an EFC Context Mark, as described in Table 90.

Table 90 — EfcContextMark

Component name	Component type	Semantics
`contractProvider`	`Provider`, see Table 66	`contractProvider` identifies the organization that issued the service rights given in the Contract, i.e. the toll service provider, see Table 66. Numbers shall be assigned on a national basis.
`typeOfContract`	`OCTET STRING`	`typeOfContract` represents a toll service provider-specific designation of the rules that apply to the contract. NOTE It allows, for example, for the determination of the tariff or designating the type of purse associated with the contract.
`contextVersion`	`INTEGER`	`contextVersion` denotes the implementation version of the concerned contract within the context of the given toll service provider, value assigned at the discretion of the toll service provider. NOTE The `ContextVersion` can also be used as a security key reference.

5.5.2 ReceiptContract

The data type `ReceiptContract` is an ordered list of three components that identify a contract for a tolling session, as described in Table 91.

Table 91 — ReceiptContract

Component name	Component type	Semantics
`sessionContractProvider`	`Provider`, see Table 66	`sessionContractProvider` provides the identification of the organization that issued the contract applied in the session.