

# EasyGo+ DSRC transaction for tolling and enforcement

## Enclosure C to Document 202 “Roadside and on board equipment”

**This copy of the document was published on [www.easygo.com](http://www.easygo.com) and is for information purposes only. It may change without further notice.**

## Table of contents

DOCUMENT REVISION HISTORY .....	3
ABBREVIATIONS .....	4
1 GENERAL.....	6
1.1 PURPOSE OF THE DOCUMENT .....	6
1.2 EASYGO+ TOLLING AND ENFORCEMENT TRANSACTION .....	7
1.3 EASYGO+ ENFORCEMENT TRANSACTION .....	8
1.4 SECURITY FEATURES .....	8
1.5 DATA SPECIFICATION .....	8
2 EASYGO+ TOLLING AND ENFORCEMENT TRANSACTION .....	9
2.1 DECISION FLOW.....	9
2.2 TOLLING AND ENFORCEMENT TRANSACTION - BIT-LEVEL SPECIFICATION .....	10
3 EASYGO+ ENFORCEMENT TRANSACTION .....	21
3.1 ENFORCEMENT TRANSACTION - BIT-LEVEL SPECIFICATION .....	21
4 REFERENCES .....	25
4.1 STANDARDS AND EXTERNAL DOCUMENTS.....	25
4.2 EASYGO- DOCUMENTS .....	29

## Document revision history

Version	Date	Author	Main changes
V0.1	23.02.2010	HHA	Initial compilation
V0.23	29.04.2010	HHA	Bit level description for security level 1 added
V0.24	02.07.2010	HHA	Ch. 1.2 and 1.3 presentation of AC_CR added
V0.25	19.10.2010	HHA	Editorial changes, release candidate (RC)
V1.0	18.11.2010	HHA	Release by WG2
V1.01	13.04.2011	HHA	Editorial changes
V1.50		HHA	Adaptation as enclosure to EasyGo document 202
V1.51	18.11.2011	HHA	Editorial changes
V1.52	15.01.2013	HHA	Ch. 1.3: no Get.Stamped (Get.Request only)
V1.53	11.04.2013	HHA	References updated
2.0	02.05.2013		Approved by steering committee

## Abbreviations

Abbreviation	Definition	Abbreviation	Definition
ADU	Application Data Unit	HGV	Heavy Goods Vehicle
AID	DSRC-Application Entity Identifier	HMI	Human-Machine Interface
AP	Application Process	HV	Heavy Vehicle (> 3.5 tons)
APDU	Application Protocol Data Unit	IID	Invoker Identifier
ASN.1	Abstract Syntax Notation One (ISO/IEC 8824-1)	KeyRef_CI	Key reference for TSP- (CI-) authenticator key
AttrID	Attribute Identifier	KeyRef_OP	Key reference for TC- (OP-) authenticator key
AuKey_Iss	(Issuer) Authentication Key	L1	Layer 1 of DSRC (Physical Layer)
BST	Beacon Service Table	L2	Layer 2 of DSRC (Data Link Layer)
BVS	Point of sales (POS)	L7	Layer 7 of DSRC (Application Layer)
CE	Conformity Declaration	LED	Light Emitting Diode
CEN	European Committee for Standardization	LID	Link Identifier
CI	Contract Issuer = Toll Service Provider (TSP)	LLC	Logical Link Control
EASYGO+-OBU	Interoperable OBU	LPN	Licence Plate Number
DES	Data Encryption Standard, ANSI X3.92	LSB	Least Significant Bit
3DES	Triple Data Encryption Standard	MAC	Medium Access Control
DSRC	Dedicated Short Range Communication	MAC	Message Authentication Code
EAcK	ElementAccessKey	MEAcK	MasterElementAccessKey
EAuK	ElementAuthenticationKey	MEAuK	MasterElementAuthenticationKey
EFC	Electronic Fee Collection	MEDIA	International interoperability project (for reference only)
EID	Element Identifier	MLW	Maximum Laden Weight
FCS	Frame Check Sequence	MMI	Man Machine Interface
HDLC	High Data Link Control	MSB	Most Significant Bit
HF	High Frequency	OBE	On-Board Equipment

Abbreviation	Definition			
OBU	On-Board Unit			
OSI	Open Systems Interconnection			
PDU	Protocol Data Unit			
RF	Radio Frequency			
RSE	Roadside Equipment			
TC	Toll Charger			
TSP	Toll Service Provider, Contract Issuer (CI)			
UI	User Interface (= MMI)			
VST	Vehicle Service Table			

Internet  
www.easygo.com

# 1 General

## 1.1 Purpose of the document

**This specification provides the definition for the DSRC transactions used in EASYGO+ context according to EN15509.**

This specification applies for postpay transactions for communication between OBU and roadside tolling equipment and in case of multilane free flow systems for combined tolling and enforcement RSE.

**In general, the transaction is based on the “pick what you like” idea, i.e. it is up to the toll charger which attribute must be read to allow correct charging. This chapter gives an exemplary definition of the DSRC transactions, between the EASYGO+-OBU and roadside equipment (RSE).**

The table below provides an overview of the attributes involved in the data exchanges of DSRC transactions associated with the EASYGO+-OBU.

ATTRIBUTES (EID>0)	AttrID	Tolling & enforcement	Enforcement
EFC Context Mark	0	X	X
Contract Authenticator	4	*)	*)
VehicleLicencePlateNumber	16	X	X
VehicleClass	17	X	X
VehicleDimensions	18	**)	**)
VehicleAxles	19	X	X
VehicleWeightLimits	20	**)	**)
VehicleSpecificCharacteristics	22	X	X
VehicleAuthenticator	23	*)	*)
EquipmentOBUId	24	X	X
EquipmentStatus	26	X	
PaymentMeans (including PAN)	32	X	X
ReceiptData1	33	X	X
ReceiptData2	34	X	X

\*) Attributes 4 and 23 are optional (not foreseen in EN15509); use is up to the toll charger (if supported by OBU).

\*\*\*) Attributes 18 and 20 are not used in this exemplary description.

## 1.2 EASYGO+ tolling and enforcement transaction

A tolling and enforcement transaction is performed for the purpose of charging and, if applicable, enforcing the due tolling fee.

The following table shows an example for such a transaction, a dedicated implementation has to be tailored to the toll chargers needs.

Phase	Roadside Equipment		On-board unit	Remarks
<i>Initialisation</i>	<b>INITIALISATION.request (BST)</b>	→		RSE periodically sends BST.
(BST – VST)		←	<b>INITIALISATION.response (VST)</b> <ul style="list-style-type: none"> <li>EFC-ContextMark</li> <li>AC_CR-KeyReference</li> <li>RndOBE</li> </ul>	A newly arrived OBE answers with VST. According to the information in the VST, the RSE decides whether the OBU/Contract is acceptable or not. IF not the presentation phase is not started.
<i>Presentation</i>	<b>GET_STAMPED.request AC_CR</b> <ul style="list-style-type: none"> <li>PaymentMeans, including PersonalAccountNumber (RndRSE, KeyRef_CI)</li> </ul> <b>GET.request</b> <ul style="list-style-type: none"> <li>EquipmentOBUId</li> <li>ReceiptData1</li> <li>EquipmentStatus</li> <li>Vehicle data: <ul style="list-style-type: none"> <li>- VehicleClass</li> <li>- VehicleAxles</li> <li>- VehicleSpecificCharacteristics</li> </ul> </li> </ul>	→		Read PaymentMeans (including Personal Account Number) and request the OBU to calculate the CI authenticator Read data that serves as a basis for calculating the fee (i.e.vehicleClass, VehicleAxles, VehicleSpecificCharacteristics) and data for diagnostics, auditing and controlling (i.e. Equipment OBU Id, Receipt Data1, EquipmentStatus. The Attribute VehicleLicencePlateNumber is read out in the receipt phase in order to provide a better distribution of data in uplink frames between the presentation and the uplink phases .
		←	<b>GET_STAMPED.response</b> <ul style="list-style-type: none"> <li>Authenticator (Auth_CI)</li> </ul> <b>GET.response</b>	OBU responds with the requested data, plus the CI Authenticator computed over Payment Means (including Personal Account Number)
<i>Receipt</i>	<b>GET_STAMPED.request AC_CR</b> <ul style="list-style-type: none"> <li>PaymentMeans, including PersonalAccountNumber (RndRSE, KeyRef_OP)</li> </ul> <b>GET.request AC_CR</b> <ul style="list-style-type: none"> <li>Vehicle data: <ul style="list-style-type: none"> <li>- VehicleLicencePlateNumber</li> </ul> </li> </ul> <b>SET.request AC_CR</b> <ul style="list-style-type: none"> <li>ReceiptData1</li> <li>ReceiptData2</li> <li>EquipmentStatus</li> </ul> <b>SET_MMI.request</b>	→		Read Payment Means (including Personal Account Number) and request the OBU to calculate an authenticator for the toll charger. Write new receipt (or entry ticket) to ReceiptData1. Copy old receipt to ReceiptData2 Write new status information and increment transaction counter.  Signal the transaction's result via the OBU's HMI: OK, Warning or NOK
		←	<b>GET_STAMPED.response</b> <ul style="list-style-type: none"> <li>Authenticator (Auth_OP)</li> </ul> <b>GET.response</b> <b>SET.response</b> <b>SET_MMI.response</b>	
<i>Tracking</i>	<b>ECHO.request</b>	→		Track OBU by exchanging dummy information The usage of Echo is optional, at the discretion of the RSE, and may be repeated (used e.g. at enforcement equipment in multilane free flow systems).
And		←	<b>ECHO.response</b>	
<i>Closing</i>	<b>EVENT_REPORT.request (Release)</b>	→		RSE closes transaction and releases OBU

### 1.3 EASYGO+ Enforcement transaction

An enforcement transaction is performed for enforcement-only purposes, e.g. used by mobile enforcement equipment.

Phase	Roadside Equipment		On-board unit	Remarks
<i>Initialisation</i> (BST – VST)	INITIALISATION.request (BST)	→		
		←	INITIALISATION.response (VST)	
<i>Presentation 1</i>	<b>GET.request</b> <b>AC_CR</b> <ul style="list-style-type: none"> <li>• EquipmentOBUID</li> <li>• EquipmentStatus (transaction counter)</li> <li>• Vehicle data: <ul style="list-style-type: none"> <li>- VehicleLicencePlateNumber</li> <li>- VehicleClass</li> <li>- VehicleAxles</li> <li>- VehicleSpecificCharacteristics</li> </ul> </li> </ul>	→		
		←	GET.response	
<i>Presentation 2</i>	<b>GET.request</b> <b>AC_CR</b> <ul style="list-style-type: none"> <li>• ReceiptData1</li> <li>• ReceiptData2</li> </ul>	→		
		←	GET.response	
<i>Closing</i>	EVENT_REPORT.request (Release)	→		

### 1.4 Security features

The EASYGO+ DSRC transaction comprises the following security features:

- Transaction counter, increased by the RSE, allowing detection of transaction sequencing anomalies in the central system.
- Authentication to the Service Provider (Contract Issuer), i.e. challenge-response of PaymentMeans data using the GET\_STAMPED function with the CI Key
- Authentication to the Toll Charger, i.e. challenge-response of PaymentMeans data using the GET\_STAMPED function with the Operator Key
- Protected access to the OBU's data, through the implementation of Access Credentials (Security level 1 acc. to EN15509).

### 1.5 Data specification

For data definition (attribute data) refer to the document 202-B [OBU\_data].



## 2 EASYGO+ tolling and enforcement transaction

### 2.1 Decision flow

Each time the RSE receives a VST from an OBU it analyses the attribute EFC-ContextMark and if necessary, the data elements EquipmentClass+ManufacturerID, in order to decide which application to use. If those elements corresponds to one of the entries in the RSE's Contract Issuer List the RSE will proceed with the associated application to perform a suitable transaction. In case of a EASYGO+- OBU the Tolling and Enforcement Transaction (object of the present document) has to apply.

If the VST contains a list of (more than one) EFC-ContextMark, the first entry will be used that can be matched with the Contract Issuer List.

Access Credentials are used in the further transactional steps if the entry in the Contract Issuer List indicates to use security level 1.

The VST contains an ApplicationContextMark with:

For security level 0 (no AC\_CR): EFC-ContextMark (6 Bytes)

For security level 1 (AC\_CR used): EFC-ContextMark, AC\_CR\_Keyreference, RndOBU1 (in total 16 Bytes)

Access Credentials are used in the further transactional steps if the entry in the Contract Issuer List indicates to do so. Note: If AC\_CR are not to be used, the RSE shall accept both a VST with an ApplicationContextMark of 6 Bytes and of 16 Bytes. In the last case, it shall simply ignore the additional 10 Bytes.

The first GET\_STAMPED.request is performed with the suitable authenticator key for the service provider authentication as listed in the Contract Issuer List for that EFC-ContextMark.

Note: The SET\_MMI command is used with a container type compliant to the OBU, which can be controlled by an appropriate entry in the Contract Issuer List, preferable the container type acc. to the last draft version of [EFC-API] (69h) shall be used.

If the PaymentMeans-ExpiryDate is older than RSE current station's date, the OBU shall be handled like an OBU with invalid PaymentMeans (SET\_MMI.request: 4 beep code).

Value "0" of PaymentMeans-ExpiryDate means that there is no Expiry Date to be checked.

If the 15th bit of EquipmentStatus is set to 1 the OBU shall be handled as a "blacklisted OBU", if the Contract Issuer List indicates to do so (SET\_MMI.request: 4 beep code).

---

<sup>1</sup> The AC\_CR\_Keyreference and RndOBU data elements are defined as octetstrings with a container Choice = 2 and a length indicator.

The second GET\_STAMPED.request is performed with the suitable authenticator key for the toll charger authentication as listed in the Contract Issuer List for that EFC-ContextMark.

The RndRse number is the same for both GET\_STAMPED.request.

## 2.2 Tolling and enforcement transaction - bit-level specification

The specification accounts for the complete frame content (excluding the zero-bit insertions) of the data exchanged, including protocol information related to DSRC-L1, -L2 and -L7, in order to ensure unambiguity.

Note: this bit-level specification has been written with the following constraints: EquipmentOBUID with length 4+1 Byte and VehicleLicencePlateNumber with length 13 Byte (LPN with 10 characters – minimal length). The bit-level changes accordingly, when these attributes have different length (see [OBU\_data]). As the transaction acc. to EN15509 is based on the idea to read only attribute data needed for the tolling functionality in the specific tolling context, the following tables are only representing examples regarding attributes.

Lines for octets only present or valid for security level 1 (Access credentials) are marked by red colour and numbered (mostleft colum) by letters (A, B, C, ...).

### 2.2.1 Initialisation

#### 2.2.1.1 Initialisation request (BST)

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Broadcast LID	1111 1111	Link address for broadcast
3	MAC control field	1010 0000	The frame contains a command LPDU
4	LLC control field	0000 0011	UI command
5	Fragmentation header	1xxx x001	No fragmentation. PDU # shall never be set to 0000 <sub>2</sub> or 0001 <sub>2</sub> .
6	BST SEQUENCE {	1000	INITIALISATION.request
	OPTION indicator	0	NonmandApplications not present.
	BeaconId.ManufacturerId INTEGER (0..65535)	000	Manufacturer identifier:1 (=Kapsch). See [AVI No] Register at <a href="http://www.nen.nl/cen278">www.nen.nl/cen278</a> for value assignment.
7		0000 0000	
8		0000 1	
	BeaconId.IndividualIdINTEGER (0..2 <sup>27</sup> -1)	000	27 bit ID available for manufacturer. Example: Id=1052
9		0000 0000	
10		0000 0100	
11		0001 1100	
12	Time TimeReal	0100 0001	32 bit UNIX System Time, the number of seconds passed since 1st January 1970, 00:00 (UTC). Example: 1103790512 <sub>10</sub>
13		1100 1010	
14		1000 0001	
15		1011 0000	
16	Profile INTEGER (0..127,...)	0000 0000	Profile (p=0 <sub>10</sub> : 1,5 MHz sub-carrier, p=1 <sub>10</sub> : 2,0 MHz sub-carrier). Example: No extension, Profile 0
17	MandApplications SEQUENCE (0..127,...) OF {	0000 0001	No extension, Number of mandApplications= 1
18	OPTION indicator	0	EID not present
	OPTION indicator	0	Parameter not present
	AID DSRCApplicationEntityID }	00 0001	No extension. AID = 1 <sub>10</sub> =EFC
19	ProfileList SEQUENCE (0..127,...) OF Profile }	0000 0000	No extension, number of profiles in list = 0.
20	FCS	xxxx xxxx	Frame check sequence
21		xxxx xxxx	
22	FLAG	0111 1110	End Flag

### 2.2.1.2 Private window request

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	0110 0000	Private window request
7	FCS	xxxx xxxx	Frame check sequence
8		xxxx xxxx	
9	FLAG	0111 1110	End Flag

### 2.2.1.3 Private window allocation

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	0010 s000	Private window allocation
7	FCS	xxxx xxxx	Frame check sequence
8		xxxx xxxx	
9	FLAG	0111 1110	End Flag

### 2.2.1.4 Initialisation response (VST)

Note: the present VST is an example of a VST containing only one application and EFC-ContextMark. VSTs with more EFC-ContextMarks (within the L2 frame length limit) shall also be correctly interpreted by the RSE.

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1100 0000	The frame contains a command LPDU
7	LLC control field	0000 0011	UI command
8	Fragmentation header	1xxx x001	No fragmentation. PDU # shall never be set to 0000 <sub>2</sub> or 0001 <sub>2</sub> .
9	VST SEQUENCE {	1001	INITIALISATION.response
	Fill BIT STRING (SIZE(4))	0000	Set to 0
10	Profile INTEGER (0..127,...)	0000 0000	No extension, profile p. Example : 0 <sub>10</sub>
11	Applications SEQUENCE (0..127,...) OF {	0000 0001	No extension, 1 application
12	OPTION indicator	1	EID present
	OPTION indicator	1	Parameter present
	AID DSRCApplicationEntityID	00 0001	No extension, AID = 1 (EFC)
13	EID	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
14	Parameter CONTAINER {	0000 0010	Choice 2 = Octet string
15		0000 0110	No extension, octet string length = 6 <sub>10</sub>
16	EFC-ContextMark SEQUENCE {		
	ContractProvider SEQUENCE {		
	CountryCode BIT STRING (SIZE(10))	0111 0100	As Specified in [OBU_data]
17		01	
	IssuerIdentifier INTEGER (0..16383) }	00 0000	As Specified in [OBU_data]
18		0000 0001	
19	TypeOfContract OCTET STRING (SIZE(2))	0000 0011	Type of contract. As Specified in [OBU_data]
20		0000 0000	

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
21	ContextVersion INTEGER (0..127,...) }}	0000 0001	No extension, context version. As Specified in [OBU_data]
A	CONTAINER	0000 0010	Choice 2 = Octet string
B	OCTET STRING	0000 0010	No extension, field length 2 <sub>10</sub>
C	AC_CR-Reference SEQUENCE { AC-MasterKeyRef Int1 AC_CR-Diversifier Int1	0000 0001	AC_CR-Reference to, consisting of AC_CR-MasterKeyRef and AC_CR-Diversifier, used for the computation of AC_CRKey and AC_CR.
D	CONTAINER	0000 0010	Choice 2 = Octet string
E	OCTET STRING	0000 0100	No extension, field length 4 <sub>10</sub>
F	rndOBE Int4	0000 0000	Random Number (nonce) used together with AC_CRKey to calculate AC_CR. Example : 640 <sub>10</sub>
G		0000 0000	
H		0000 0010	
I		1000 0000	
22	ObeConfiguration SEQUENCE { OPTION indicator EquipmentClass INTEGER (0..32767)	1 000 0000	ObeStatus present Example : 3 <sub>10</sub>
23		0000 0011	
24	ManufacturerId INTEGER (0..65535)	0000 0000	Manufacturer identifier. See [AVI No] Register at <a href="http://www.nen.nl/cen278">www.nen.nl/cen278</a> for value assignment. Kapsch = 1 <sub>10</sub> .
25		0000 0001	
26	ObeStatus INTEGER(0..65535)	0000 0011	Example : 768 <sub>10</sub>
27	}	0000 0000	
28	FCS	xxxx xxxx	Frame check sequence
29		xxxx xxxx	
30	FLAG	0111 1110	End Flag

## 2.2.2 Presentation

### 2.2.2.1 Presentation request

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled ACn command, n bit
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.
9	GET_STAMPED.requestSEQUENCE {	0000 0101	ACTION.request (GET Stamped, AccessCredential not present, ActionParameter present, IID not present and Reply expected) or alternatively: ACTION.request (GET Stamped, AccessCredentials, ActionParameter present, IID not present and Reply expected)
		0000 1101	
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
11	ActionType INTEGER(0..127,...)	0000 0000	No extension, GET_STAMPED.request = 0
A	AccessCredential OCTET STRING {	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the Access Credential Key AC_CRKey. Example: AC_CR(0) =
C		1001 0100	
D		1111 1000	04 94 F8 97'H
E		1001 0111	
12	ActionParameter CONTAINER {	0001 0001	No extension, Choice 17 <sub>10</sub> = GetStampedRq
13	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId { PaymentMeans } }	0000 0001	No extension, number of attribute IDs = 1
14		0010 0000	AttributeId = 32 <sub>10</sub> = PaymentMeans
15	Nonce OCTET STRING {	0000 0100	No extension, octet string length = 4 <sub>10</sub>
16	RndRSE	TTTT TTTT	Random number from RSE, containing SessionTime, needed to calculate the Authenticator
17		TTTT TTTT	
18		TTTT TTTT	
19		TTTT TTTT	
20	KeyRef } }	0110 1111	KeyRef_CI (e.g. =111 <sub>10</sub> )
21	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).
22	GET.request SEQUENCE { OPTION indicator	0110 0	GET.request AccessCredential not present or alternatively: AccessCredential present
	OPTION indicator	0	IID not present

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
	OPTION indicator	1	AttributeIdList present
	Fill BIT STRING(SIZE(1))	0	Set to 0
23	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
A	AccessCredential OCTET STRING {	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97'H
E		1001 0111	
24	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId {		
		0000 0110	No extension, number of attribute Ids = 6 <sub>10</sub>
25	VehicleClass	0001 0001	AttributeId = 17 <sub>10</sub> = VehicleClass
26	VehicleAxles	0001 0011	AttributeId = 19 <sub>10</sub> = VehicleAxles
27	VehicleSpecificCharacteristics	0001 0110	AttributeId = 22 <sub>10</sub> = VehicleSpecificCharacteristics
28	EquipmentOBUID	0001 1000	AttributeId = 24 <sub>10</sub> = EquipmentOBUID
29	EquipmentStatus	0001 1010	AttributeId = 26 <sub>10</sub> = EquipmentStatus
30	ReceiptData1 } } }	0010 0001	AttributeId = 33 <sub>10</sub> = ReceiptData1
31	FCS	xxxx xxxx	Frame check sequence
32		xxxx xxxx	
33	FLAG	0111 1110	End Flag

### 2.2.2.2 Presentation response

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6		MAC control field	
7	LLC control field	n111 0111	Response available, Acn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request. .
10	GET_STAMPED.response SEQUENCE {	0001 0100	ACTION.response (Get Stamped rs)
11	EID INTEGER (0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
12	ResponseParameter CONTAINER {	0001 0010	No extension. Choice 18 <sub>10</sub> = GetStampedRs
13	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, number of attributes: 1
14	Attributes SEQUENCE { AttributeId	0010 0000	PaymentMeans = 32 <sub>10</sub>
15	AttributeValue CONTAINER {	0100 0000	Container Choice: 64 <sub>10</sub> = PaymentMeans
16	PersonalAccountNumber	xxxx xxxx	PersonalAccountNumber
17		xxxx xxxx	
18		xxxx xxxx	
19		xxxx xxxx	
20		xxxx xxxx	
21		xxxx xxxx	
22		xxxx xxxx	
23		xxxx xxxx	
24		xxxx xxxx	
25		xxxx xxxx	
26	PaymentMeansExpiryDate	0001 1110	DateCompact. Example : 2005-03-01
27		0110 0001	
28	PaymentMeansUsageControl	0000 0000	Example : Not specified 0 <sub>10</sub>
29		0000 0000	
30	Authenticator OCTET STRING {	0000 0100	No extension, octet string size = 4 <sub>10</sub>
31	Authenticator	xxxx xxxx	Authenticator over AttributeList (containing PaymentMeans) and RndRSE (containing SessionTime) calculated using AuKey
32		xxxx xxxx	
33		xxxx xxxx	
34		xxxx xxxx	
35	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
36	GET.response SEQUENCE	0111 0100	GET.response
37	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
38	AttributeList SEQUENCE (0..127,...) OF {	0000 0110	No extension, 6 attributes in list.
39	AttributeId INTEGER(0..127,...)	0001 0001	AttributeId = 17 <sub>10</sub> = VehicleClass
40	Attribute Value CONTAINER {	0011 0001	Container choice = 49 <sub>10</sub>
41	VehicleClass	1000 1001	VehicleClass Trailer present, CCC=000, class 3 (HV with 4 axles).
42	AttributeId INTEGER(0..127,...)	0001 0011	AttributeId = 19 <sub>10</sub> = VehicleAxles
43	Attribute Value CONTAINER {	0011 0011	Container choice = 51 <sub>10</sub>
44	VehicleAxles	0000 0000	VehicleAxles value. Example: vehicle first axle height = not
45		0000 0011	specified, tyre type = not specified, 3 axles (without trailer).
46	AttributeId INTEGER(0..127,...)	0001 0110	AttributeId = 22 <sub>10</sub> = VehicleSpecificCharacteristics
47	Attribute Value CONTAINER {	0011 0110	Container choice = 54 <sub>10</sub>
48	EnvironmentalCharacteristics(Euro type, Cop type)	0000	Example : no entry = 0
49	EngineCharacteristics	0000 0000	Example : no entry = 0
50	DescriptiveCharacteristics	0000 0000	Example : no entry = 0
51	FutureCharacteristics	0000 0000	Example : no entry = 0
52	AttributeId INTEGER(0..127,...)	0001 1000	AttributeId = 24 <sub>10</sub> = EquipmentOBUId
53	Attribute Value CONTAINER {	0011 1000	Container choice = 56 <sub>10</sub>
54	Equipment OBU Id	0000 0100	OCTET STRING. Example : Kapsch's format (length indicator = 4, manufacturing year = 03, manufacturing month = 05, Serial number = 640)
55		0000 0011	
56		0000 0101	
57		0000 0010	
58		1000 0000	
59	AttributeId INTEGER(0..127,...)	0001 1010	AttributeId = 26 <sub>10</sub> = EquipmentStatus
60	Attribute Value CONTAINER {	0011 1010	Container choice = 58 <sub>10</sub>
61	EquipmentStatus	0000 0000	EquipmentStatus (transaction counter). Example : Local use =
62		0011 1010	Not specified =0 ; transaction counter = 58.
63	AttributeId INTEGER(0..127,...)	0010 0001	AttributeId = 33 <sub>10</sub> = ReceiptData1
64	Attribute Value CONTAINER {	0100 0001	Container choice = 65 <sub>10</sub>
65	ReceiptData1	0001 1010	ReceiptData1.SessionTime. Example : 2003-03-01, 21:12:10
66		0110 0001	

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
67		1010 1001	
68		1000 0101	
69		xxxx xxxx	ReceiptData1.SessionServiceProvider
70		xxxx xxxx	
71		xxxx xxxx	
72		xxxx xxxx	ReceiptData1.StationLocation
73		xxxx xxxx	
74		1000 0111	ReceiptData1.SessionLocation. Example: Ascending kilometrage and passage station type.
75		xxxx xxxx	ReceiptData1.SessionType
76		xxxx xxxx	ReceiptData1.SessionResult
77		xxxx xxxx	ReceiptData1.SessionTariffClass
78		xxxx xxxx	ReceiptData1.ClaimedClass
79		xxxx xxxx	ReceiptData1.SessionFee
80		xxxx xxxx	
81		xxxx xxxx	
82		xxxx xxxx	
83		1100 0000	ReceiptData1.SessionServiceProvider (from EFC-ContextMark). Example: Austria, 1 <sub>10</sub> , ASFINAG
84		0100 0000	
85		0000 0001	ReceiptData1.SessionType of contract (from EFC-ContextMark). Example = 3 <sub>10</sub>
86		0000 0011	
87		0000 0000	ReceiptData1.SessionContext version (from EFC-ContextMark).
89		0000 0000	
90		xxxx xxxx	ReceiptData1.SessionAuthenticator
91		xxxx xxxx	
92		xxxx xxxx	
93		Xxxx xxxx	
94	FCS	Xxxx xxxx	
95		xxxx xxxx	Frame check sequence
96	FLAG	0111 1110	End Flag

## 2.2.3 Receipt

receipt request	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled ACn command, n bit
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.
9	GET_STAMPED.requestSEQUENCE {	0000 0101	ACTION.request (GET Stamped, AccessCredential not present, ActionParameter present, IID not present and Reply expected) <b>or alternatively:</b> ACTION.request (GET Stamped, AccessCredentials, ActionParameter present, IID not present and Reply expected)
		0000 1101	
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
11	ActionType INTEGER(0..127,...)	0000 0000	No extension, GET_STAMPED.request = 0
A	AccessCredential OCTET STRING	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97 H
E		1001 0111	
12	ActionParameter CONTAINER {	0001 0001	No extension, Choice 17 <sub>10</sub> = GetStampedRq
13	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId {	0000 0001	No extension, number of attribute IDs = 1
14	PaymentMeans } }	0010 0000	AttributeId = 32 <sub>10</sub> = PaymentMeans
15	Nonce OCTET STRING {	0000 0100	No extension, octet string length = 4 <sub>10</sub>
16	RndRSE	rrrr rrrr	Random number from RSE, containing SessionTime, needed to calculate the Authenticator
17		rrrr rrrr	
18		rrrr rrrr	
19		rrrr rrrr	
20	KeyRef } }	0111 0110	KeyRef_OP (e.g. =118 <sub>10</sub> )
21	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).
22	GET.request SEQUENCE {	0110	GET.request
	OPTION indicator	0	AccessCredential not present <b>or alternatively:</b> AccessCredential present
		1	
	OPTION indicator	0	IID not present
	OPTION indicator	1	AttributeIdList present
	Fill BIT STRING(SIZE(1))	0	Set to 0
23	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
A	AccessCredential OCTET STRING	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97 H
E		1001 0111	
24	AttributeIdList SEQUENCE (0..127,...) OF { INTEGER (0..127,...) AttributeId {	0000 0001	No extension, number of attribute Ids = 1 <sub>10</sub>
25	VehicleLicencePlateNumber } } }	0001 0000	AttributeId = 16 <sub>10</sub> = VehicleLicencePlateNr
26	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).
27	SET.request SEQUENCE {	0100 0001	SET.request (No AccessCredential, no IID, fill, reply expected) <b>or alternatively:</b> SET.request (AccessCredential, no IID, fill, reply expected)
		0100 1001	
28	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
A	AccessCredential OCTET STRING	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97 H
E		1001 0111	
29	AttributeList SEQUENCE ((0..127,...) OF { Attributes SEQUENCE {	0000 0011	No extension, number of attributes in list = 3 <sub>10</sub>
30	AttributeId INTEGER(0..127,...)	0001 1010	AttributeId = 26 <sub>10</sub> = EquipmentStatus
31	Attribute Value CONTAINER {	0011 1010	Container choice = 58 <sub>10</sub>
32	EquipmentStatus	0000 0000	EquipmentStatus (transaction counter). Example : Local use =
33		0011 1011	Not specified = 0 ; transaction counter = 59 (58 + 1).
34	AttributeId INTEGER(0..127,...)	0010 0001	AttributeId = 33 <sub>10</sub> = ReceiptData1
35	Attribute Value CONTAINER {	0100 0001	Container choice = 65 <sub>10</sub>
36	ReceiptData1	xxxx xxxx	ReceiptData1.SessionTime



receipt request	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
37		xxxx xxxx	
38		xxxx xxxx	
39		xxxx xxxx	
40		xxxx xxxx	ReceiptData1.SessionServiceProvider
41		xxxx xxxx	
42		xxxx xxxx	
43		xxxx xxxx	ReceiptData1.StationLocation
44		xxxx xxxx	
45		xxxx xxxx	ReceiptData1.SessionLocation.
46		xxxx xxxx	ReceiptData1.SessionType
47		xxxx xxxx	ReceiptData1.SessionResult
48		xxxx xxxx	ReceiptData1.SessionTariffClass
49		xxxx xxxx	ReceiptData1.ClaimedClass
50		xxxx xxxx	ReceiptData1.SessionFee
51		xxxx xxxx	
52		xxxx xxxx	
53		xxxx xxxx	
54		xxxx xxxx	ReceiptData1.SessionContractProvider
55		xxxx xxxx	
56		xxxx xxxx	
57		xxxx xxxx	ReceiptData1.SessionTypeOfContract
58		xxxx xxxx	
59		xxxx xxxx	ReceiptData1.SessionContextVersion
60		xxxx xxxx	ReceiptData1.Authenticator
61		xxxx xxxx	
62		xxxx xxxx	
63		xxxx xxxx	
64	AttributeId INTEGER(0..127,...)	0010 0010	AttributeId = 34 <sub>10</sub> = ReceiptData2
65	Attribute Value CONTAINER {	0100 0010	Container choice = 66 <sub>10</sub>
66	ReceiptData2	xxxx xxxx	ReceiptData2. Same format as ReceiptData1
....		.... ....	
93	}	xxxx xxxx	
94	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as before (concatenation).
95	SET_MMI.request SEQUENCE {	0000 0101	ACTION.request (no AccCred, ActionPar, no IID, confirmed, reply)
96	EID INTEGER(0..127,...)	0000 0000	No extension, EID = 0 (system element)
97	ActionType INTEGER(0..127,...)	0000 1010	No extension, SET_MMI.request = 10 <sub>10</sub>
98	ActionParameter CONTAINER	xxxx xxxx	No extension, Use Type 69 (preferred) or Type 0 depending on setting in TSP list at RSE
99	SetMMI INTEGER	0000 0000	Example : ok (0 <sub>10</sub> )
100	FCS	xxxx xxxx	Frame check sequence
101		xxxx xxxx	
102	FLAG	0111 1110	End Flag

### 2.2.3.1 Set receipt response

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1101 0000	The frame contains a response LPDU
7	LLC control field	n111 0111	Acn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request. .
10	GET_STAMPED.response SEQUENCE {	0001 0100	ACTION.response (Get Stamped rs)
11	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
12	ResponseParameter CONTAINER {	0001 0010	No extension. Choice 18 <sub>10</sub> = GetStampedRs
13	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, number of attributes: 1

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
14	Attributes SEQUENCE { AttributeId	0010 0000	PaymentMeans = 32 <sub>10</sub>
15	AttributeValue CONTAINER {	0100 0000	Container Choice: 64 <sub>10</sub> = PaymentMeans
16	PersonalAccountNumber	xxxx xxxx	PersonalAccountNumber
17		xxxx xxxx	
18		xxxx xxxx	
19		xxxx xxxx	
20		xxxx xxxx	
21		xxxx xxxx	
22		xxxx xxxx	
23		xxxx xxxx	
24		xxxx xxxx	
25		xxxx xxxx	
26	PaymentMeansExpiryDate	0001 1110	DateCompact. Example : 2005-03-01
27		0110 0001	
28	PaymentMeansUsageControl	0000 0000	Example : Not specified 0 <sub>10</sub>
29		0000 0000	
30	Authenticator OCTET STRING {	0000 0100	No extension, octet string size = 4 <sub>10</sub>
31	Authenticator	xxxx xxxx	Authenticator over AttributeList (containing PaymentMeans) and RndRSE (containing SessionTime) calculated using AuKey
32		xxxx xxxx	
33		xxxx xxxx	
34		xxxx xxxx	
35	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
36	GET.response SEQUENCE	0111 0100	GET.response
37	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
38	AttributeList SEQUENCE (0..127,...) OF {	0000 0001	No extension, 1 attributes in list.
39	AttributeId INTEGER(0..127,...)	0001 0000	AttributeId = 16 <sub>10</sub> = VehicleLicencePlateNo
40	Attribute Value CONTAINER {	0010 1111	Container choice = 47 <sub>10</sub>
41	Vehlpn {SEQUENCE countryCode,  AlphabetIndicator, LicencePlateNumber	1010 0100	VehicleLicencePlateNumber. Example : Country: SE,  alphabet indicator no 1 length indicator = 10 chars OCD560, padded with 4 'NUL'-characters
42		00	
43		00 0000	
44		0000 1010	
45		0100 1111	
46		0100 0011	
47		0100 0100	
48		0011 0101	
49		0011 0110	
50		0011 0000	
51		0000 0000	
52		0000 0000	
53		0000 0000	
54	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
55	SET.response SEQUENCE {	0101	SET.response
	OPTION indicator	0	IID not present
	OPTION indicator	0	ResponseStatus not present
	Fill BIT STRING (SIZE(2))	00	Set to 0
56	EID INTEGER (0..127,...) }	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
57	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
58	ACTION.response SEQUENCE {	0001	SET_MMI.response
	OPTION indicator	0	IID not present
	OPTION indicator	0	ResponseParameter not present
	OPTION indicator	0	ResponseStatus not present
	Fill BIT STRING (SIZE(1))	0	Set to 0
59	EID INTEGER (0..127,...) }	0000 0000	No extension, System Element EID = 0
60	FCS	xxxx xxxx	Frame check sequence
61		xxxx xxxx	
62	FLAG	0111 1110	End Flag

## 2.2.4 Tracking and closing

### 2.2.4.1 Tracking request (Echo.request)

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled ACn command n bit
8	Fragmentation header	1xxx x001	No fragmentation.
9	ECHO.request SEQUENCE {	0000	ACTION.request
	OPTION indicator	0	No Access Credentials
	OPTION indicator	1	ActionParameter present
	OPTION indicator	0	IID not present
	Mode BOOLEAN	1	Reply expected
10	EID INTEGER (0..127,...)	0000 0000	No extension, EID = 0
11	ActionType INTEGER (0..127,...)	0000 1111	No extension, ECHO.request = 15
12	ActionParameter CONTAINER	0000 0010	No extension, Choice 2 = Octet string
13	}	0000 0000	No extension. String length = 0 octets
14	FCS	xxxx xxxx	Frame check sequence
15		xxxx xxxx	
16	FLAG	0111 1110	

### 2.2.4.2 Tracking response (Echo.response)

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1101 0000	The frame contains a response LPDU
7	LLC control field	n111 0111	ACn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
10	ECHO.response SEQUENCE {	0001	ACTION.response
	OPTION indicator	0	No IID
	OPTION indicator	1	ResponseParameter present
	OPTION indicator	0	ReturnStatus not present
	FILL BIT STRING (SIZE(1))	0	Set to 0.
11	EID INTEGER (0..127,...)	0000 0000	No extension, EID = 0
12	ResponseParameter CONTAINER	0000 0010	No extension, Choice 2 = Octet string
13	}	0000 0000	No extension. String length = 0 octets
14	FCS	xxxx xxxx	Frame check sequence
15		xxxx xxxx	
16	FLAG	0111 1110	

### 2.2.4.3 Closing

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1000 0000	The frame contains a command LPDU
7	LLC control field	0000 0011	UI command
8	Fragmentation header	1xxx x001	No fragmentation.
9	RELEASE.request SEQUENCE {	0010	EVENT_REPORT.request
	OPTION indicator	0	AccessCredential not present
	OPTION indicator	0	EventParameter not present
	OPTION indicator	0	IID not present
	Mode BOOLEAN	0	No reply expected
10	EID INTEGER (0..127,...)	0000 0000	No extension, EID = 0 (system element)
11	EventType INTEGER (0..127,...) }	0000 0000	No extension, RELEASE = 0.
12	FCS	xxxx xxxx	Frame check sequence
13		xxxx xxxx	
14	FLAG	0111 1110	End Flag

### 3 EASYGO+ enforcement transaction

The EASYGO+ enforcement transaction will be used by dedicated enforcement equipment where no tolling functionality is needed, e.g. at mobile enforcement equipment.

In contrast to the combined “tolling and enforcement transaction”, only a subset of attributes and the receipt data are read.

#### 3.1 Enforcement transaction - bit-level specification

##### 3.1.1 Initialisation

The Initialisation phase is the same as for the EASYGO+ Tolling and Enforcement Transaction, see chapter 2.2.1

##### 3.1.2 Presentation 1

Similar to the Presentation Phase for the EASYGO+ Tolling and Enforcement Transaction, but only a subset of data is read-out.

##### 3.1.2.1 Presentation 1 request

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6	MAC control field	1010 s000	The frame contains a command LPDU
7	LLC control field	n111 0111	Polled ACn command, n bit
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.
9	GET.request SEQUENCE {	0110	GET.request
	OPTION indicator	0	AccessCredential not present <b>or alternatively:</b> AccessCredential present
	OPTION indicator	1	AccessCredential present
	OPTION indicator	0	IID not present
	OPTION indicator	1	AttributeIdList present
	Fill BIT STRING(SIZE(1))	0	Set to 0
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
A	AccessCredential OCTET STRING	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		1001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97'H
E		1001 0111	
	AttributeIdList SEQUENCE (0..127,...) OF		
	{ INTEGER (0..127,...) AttributeId {	0000 0110	No extension, number of attribute Ids = 6 <sub>10</sub>
11	VehicleLicencePlateNumber	0001 0000	AttributeId = 16 <sub>10</sub> = VehicleLicencePlateNr
13	VehicleClass	0001 0001	AttributeId = 17 <sub>10</sub> = VehicleClass
14	VehicleAxles	0001 0011	AttributeId = 19 <sub>10</sub> = VehicleAxles
15	VehicleSpecificCharacteristics	0001 0110	AttributeId = 22 <sub>10</sub> = VehicleSpecificCharacteristics
16	EquipmentOBUID	0001 1000	AttributeId = 24 <sub>10</sub> = EquipmentOBUID
17	EquipmentStatus } } }	0001 1010	AttributeId = 26 <sub>10</sub> = EquipmentStatus
18	FCS	xxxx xxxx	Frame check sequence
19		xxxx xxxx	
20	FLAG	0111 1110	End Flag

### 3.1.2.2 Presentation 1 response

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description		
1	FLAG	0111 1110	Start Flag		
2	Private LID	xxxx xxx0	Link address of a specific OBE		
3		xxxx xxx0			
4		xxxx xxx0			
5		xxxx xxx1			
6		MAC control field		1101 0000	The frame contains a response LPDU
7	LLC control field	n111 0111	Response available, Acn command n bit		
8	LLC status field	0000 0000	Response available and command accepted		
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.		
10	GET.response SEQUENCE	0111 0100	GET.response		
11	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>		
12	AttributeList SEQUENCE (0..127,...) OF {	0000 0111	No extension, 7 attributes in list.		
13	AttributeId INTEGER(0..127,...)	0001 0000	AttributeId = 16 <sub>10</sub> = VehicleLicencePlateNo		
14	Attribute Value CONTAINER {	0010 1111	Container choice = 47 <sub>10</sub>		
15	Vehlpn {SEQUENCE countryCode,  AlphabetIndicator, LicencePlateNumber	1010 0100	VehicleLicencePlateNumber. Example : Country: SE,  alphabet indicator no 1 length indicator = 10 chars OCD560, padded with 4 'NUL'-characters		
16		00			
17		00 0000			
18		0000 1010			
19		0100 1111			
20		0100 0011			
21		0100 0100			
22		0011 0101			
23		0011 0110			
24		0011 0000			
25		0000 0000			
26		0000 0000			
27		0000 0000			
28		AttributeId INTEGER(0..127,...)		0001 0001	AttributeId = 17 <sub>10</sub> = VehicleClass
29		Attribute Value CONTAINER {		0011 0001	Container choice = 49 <sub>10</sub>
30	VehicleClass }	1000 1001	VehicleClass Trailer present, CCC=000, class 3 (HV with 4 axles).		
31	AttributeId INTEGER(0..127,...)	0001 0011	AttributeId = 19 <sub>10</sub> = VehicleAxles		
32	Attribute Value CONTAINER {	0011 0011	Container choice = 51 <sub>10</sub>		
33	VehicleAxles	0000 0000	VehicleAxles value. Example: vehicle first axle height = not		
34		0000 0011	specified, tyre type = not specified, 3 axles (without trailer).		
35	AttributeId INTEGER(0..127,...)	0001 0110	AttributeId = 22 <sub>10</sub> = VehicleSpecificCharacteristics		
36	Attribute Value CONTAINER {	0011 0110	Container choice = 54 <sub>10</sub>		
37	EnvironmentalCharacteristics(Euro type, Cop type)	0000	Example : no entry = 0		
38	EngineCharacteristics	0000 0000	Example : no entry = 0		
39	DescriptiveCharacteristics	0000 0000	Example : no entry = 0		
40	FutureCharacteristics	0000 0000	Example : no entry = 0		
41	AttributeId INTEGER(0..127,...)	0001 1000	AttributeId = 24 <sub>10</sub> = EquipmentOBUID		
42	Attribute Value CONTAINER {	0011 1000	Container choice = 56 <sub>10</sub>		
43	Equipment OBU Id	0000 0100	OCTET STRING. Example : Kapsch's format (length indicator = 4, manufacturing year = 03, manufacturing month = 05, Serial number = 640)		
44		0000 0011			
45		0000 0101			
46		0000 0010			
47		1000 0000			
48	AttributeId INTEGER(0..127,...)	0001 1010	AttributeId = 26 <sub>10</sub> = EquipmentStatus		
49	Attribute Value CONTAINER {	0011 1010	Container choice = 58 <sub>10</sub>		
50	EquipmentStatus	0000 0000	EquipmentStatus (transaction counter). Example : Local use =		
51		0011 1010		Not specified = 0 ; transaction counter = 58.	
52	FCS	Xxxx xxxx	Frame check sequence		
53		xxxx xxxx			
54	FLAG	0111 1110	End Flag		

### 3.1.3 Presentation 2

Similar to the Presentation Phase for the EASYGO+ Tolling and Enforcement Transaction, but only receipts are read-out.

#### 3.1.3.1 Presentation 2 request

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6		MAC control field	
7	LLC control field	n111 0111	Polled ACn command, n bit
8	Fragmentation header	1xxx x001	No fragmentation. First service of chain.
9	GET.request SEQUENCE {	0110	GET.request
	OPTION indicator	0	AccessCredential not present or alternatively: AccessCredential present
	OPTION indicator	0	IID not present
	OPTION indicator	1	AttributeIdList present
	Fill BIT STRING(SIZE(1))	0	Set to 0
10	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
A	AccessCredential OCTET STRING	0000 0100	No extension, octet string length = 4 <sub>10</sub>
B	AC_CR	0000 0100	Access credential calculated by RSE using RndOBE and the
C		0001 0100	Access Credential Key AC_CRKey. Example: AC_CR(0) =
D		1111 1000	04 94 F8 97 H
E		1001 0111	
	AttributeIdList SEQUENCE (0..127,...) OF		
11	{ INTEGER (0..127,...) AttributeId {	0000 0010	No extension, number of attribute Ids = 2 <sub>10</sub>
12	ReceiptData1	0010 0001	AttributeId = 33 <sub>10</sub> = ReceiptData1
13	ReceiptData2 } } }	0010 0010	AttributeId = 34 <sub>10</sub> = ReceiptData1
14	FCS	xxxx xxxx	Frame check sequence
15		xxxx xxxx	
16	FLAG	0111 1110	End Flag

#### 3.1.3.2 Presentation 2 response

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
1	FLAG	0111 1110	Start Flag
2	Private LID	xxxx xxx0	Link address of a specific OBE
3		xxxx xxx0	
4		xxxx xxx0	
5		xxxx xxx1	
6		MAC control field	
7	LLC control field	n111 0111	Response available, Acn command n bit
8	LLC status field	0000 0000	Response available and command accepted
9	Fragmentation header	1xxx x001	No fragmentation. Same PDU # as in the corresponding request.
10	GET.response SEQUENCE	0111 0100	GET.response
11	EID INTEGER(0..127,...)	0000 0101	Uniquely associated with a context mark in the OBU. Example : 5 <sub>10</sub>
12	AttributeList SEQUENCE (0..127,...) OF {	0000 0010	No extension, 2 attributes in list.
13	AttributeId INTEGER(0..127,...)	0010 0001	AttributeId = 33 <sub>10</sub> = ReceiptData1
14	Attribute Value CONTAINER {	0100 0001	Container choice = 65 <sub>10</sub>
15	ReceiptData1	0001 1010	ReceiptData1.SessionTime. Example : 2003-03-01, 21:12:10
16		0110 0001	
17		1010 1001	
18		1000 0101	
19		xxxx xxxx	ReceiptData1.SessionServiceProvider
20		xxxx xxxx	
21		xxxx xxxx	
22		xxxx xxxx	ReceiptData1.StationLocation
23		xxxx xxxx	
24		1000 0111	ReceiptData1.SessionLocation. Example: Ascending kilometrage and passage station type.
25		xxxx xxxx	ReceiptData1.SessionType
26		xxxx xxxx	ReceiptData1.SessionResult

Octet #	Attribute / Field	Bits in Octet b <sub>7</sub> b <sub>0</sub>	Description
27		xxxx xxxx	ReceiptData1.SessionTariffClass
28		xxxx xxxx	ReceiptData1.ClaimedClass
29		xxxx xxxx	ReceiptData1.SessionFee
30		xxxx xxxx	
31		xxxx xxxx	
32		xxxx xxxx	
33		1100 0000	ReceiptData1.SessionServiceProvider. Example: Austria, 1 <sub>10</sub> (ASFINAG)
34		0100 0000	
35		0000 0001	Type of contract. Example : HVG post paid contract = 3 <sub>10</sub>
36		0000 0011	
37		0000 0000	
38		0000 0000	Security key version 0, version 1.0 of the Austrian HGV transaction
39		xxxx xxxx	ReceiptData1.Authenticator
40		xxxx xxxx	
41		xxxx xxxx	
42		Xxxx xxxx	
43	AttributeId INTEGER(0..127,...)	0010 0010	AttributeId = 34 <sub>10</sub> = ReceiptData2
44	Attribute Value CONTAINER {	0100 0010	Container choice = 66 <sub>10</sub>
45	ReceiptData2	xxxx xxxx	ReceiptData2. Same format as ReceiptData1
....		.... ....	
72	}	xxxx xxxx	
73	FCS	xxxx xxxx	Frame check sequence
74		xxxx xxxx	
75	FLAG	0111 1110	End Flag

### 3.1.4 Closing

The closing phase is the same as for the EASYGO+ Tolling and Enforcement Transaction, see chapter 2.2.4.3



## 4 References

### 4.1 Standards and external documents

For dated references, subsequent amendments to or revisions of any of these publications apply only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

Reference	Document Ref	Date / Version	Document title
[L1]	EN 12253	2004	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – Physical layer using microwave at 5.8 GHz
[L2]	EN 12795	2003	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – DSRC data link layer: Medium access and logical link control
[L7]	ISO15628 / EN12834	2007/ 2003	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – DSRC Application Layer
[Profiles]	EN 13372	2004	Road Transport and Traffic Telematics (RTTT) – Dedicated Short-Range Communication (DSRC) – Profiles for RTTT applications
[AVI No]	EN ISO 14816	2005	Road Traffic and Transport Telematics (RTTT) – Automatic Vehicle and Equipment Identification – Numbering and Data Structures
[EFC API]	ISO/DIS 14906.2	16.02.2010	Road Traffic and Transport Telematics (RTTT) – Electronic Fee Collection – Application interface definition for dedicated short range communication
[OBU DSRC tests]	CEN ISO/TS 14907-2	2006	Road Traffic and Transport Telematics (RTTT) – Electronic Fee Collection – Testprocedures for user and fixed equipment - Part 2: EFC application interface conformance test specification
[ETSI]	EN 300 674	2004	Electromagnetic Compatibility and Radio Spectrum Matters (ERM) - RTTT -DSRC-transmission equipment

Reference	Document Ref	Date / Version	Document title
[ISO CC]	EN ISO 3166-1		Codes for the representation of names of countries and their subdivisions – Part 1: Country code
[ISO 4217]	ISO 4217		Codes for the representation of currencies and funds
[ISO 7812-1]	EN ISO/IEC 7812-1	2000	Identification cards - Identification of issuers - Part 1: Numbering system
[ISO 8731-1]	ISO 8731-1	1987	Banking -- Approved algorithms for message authentication -- Part 1: DEA
[ASN.1]	ISO/IEC 8824-1	2004	Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation
[ASN.1 PER]	ISO/IEC 8825-2	2004	Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)
[CEI EN 60086-4]	IEC 60086-4		Primary batteries - Part 4: Safety standards for lithium batteries
[EN 301 489-3]	EN 301489-3	V 1.4.1:2002	Electromagnetic compatibility and Radio spectrum Matters (ERM) - ElectroMagnetic Compatibility (EMC) standard for radio equipment and services - Part 3: Specific conditions for Short-Range Devices (SRD) operating on frequencies between 9 kHz and 40 GHz; V1.4.1 :2002
[EN 60950]	EN 60950		Information technology equipment - Safety

Reference	Document Ref	Date / Version	Document title
[Env Class]	EN 60721-3		Classification of environmental conditions –Part 3: Classification of groups of environmental parameters and their severities
[ENV tests]	EN 60068-2		IEC 60068-2 Environmental testing Part 2
[IAP]	EN 15509	2007	EN 15509:2007 Road Traffic and Transport Telematics (RTTT) – Electronic Fee Collection – Interoperability application profile for DSRC
[TS102486-L2]	ETSI TS 102486-1	V1.1.1	Electromagnetic compatibility and Radio spectrum Matters (ERM); Road Transport and Traffic Telematics (RTTT); Test specifications for DSRC transmission equipment; Part 1: DSRC data link layer: medium access and logical link control; Sub parts 1 - 3
[TS102486-L7]	ETSI TS 102486-2	V1.1.1	Electromagnetic compatibility and Radio spectrum Matters (ERM); Road Transport and Traffic Telematics (RTTT); Test specifications for DSRC transmission equipment; Part 2: DSRC application layer; Sub parts 1 - 3
[MOPTT]	MOPTT	V1.0:2002	Conformance tests to the specification for interoperability in the beacon – transponder transaction (Public Works, Transport and Telecommunications Ministry, Chile, Version 1.0, 2002-01-22, CEN/TC278/WG1 N677)
[CARDME]	CARDME	2002	CARDME-4/ D4.1 – The CARDME concept (Final, 1 June 2002)
[GSS]	GSS	V3.2:2003	Global Specification for Short Range Communication (Kapsch TrafficCom AB, Kapsch Telecom GmbH, Thales e-Transactions CGA SA, version 3.2, 2003-08, <a href="http://www.etc-interop.com/pdf/gss_32.pdf">http://www.etc-interop.com/pdf/gss_32.pdf</a> )

Reference	Document Ref	Date / Version	Document title
[CESARE]	CEASARE	2002	CEASRE II – D032.1 Detailed CESARE Technical Specification – Version 3, 27.02.02
[A1]	A1	1999	A1 - Interoperable EFC Transaction using Central Account based on DSRC European Commission DG XIII - Telematics Applications Programme – TR4001 June 12, 1999 (Version ER9_1.3)
[EG2]		01.04.2005	Definition of parameters to be stored in on-board equipment designed for use with the European Electronic Toll Service - Prepared by: Expert Group 2: Vehicle Classification - Working to support the European Commission DG TREN
[EG11]		06.02.2006	Definition of the EFC Application for the EETS Based on Microwave Technologies - Prepared by Expert Group 11 - Working to support the European Commission on the work on Directive 2004/52/EC
[UNECE]			ECONOMIC COMMISSION FOR EUROPE - INLAND TRANSPORT COMMITTEE - Working Party on the Construction of Vehicles TRANS/WP.29/78/Rev.1/Amend.2 - CONSOLIDATED RESOLUTION ON THE CONSTRUCTION OF VEHICLES (R.E.3)
[Reg_doc]			Directive 1999/37/EC on Registration Documents

## 4.2 EasyGo- Documents

Reference	Document Ref	Date / Version	Document title
[PROCESS]	401		Business Process Definitions for EasyGo+
[OBU_req]	202-A		EasyGo+: Functional requirements for OBU
[OBU_data]	202-B		EasyGo+: OBU Data description (EasyGo+ OBU Personalization, Configuration and Operating Parameters )
[DSRC]	202-C		EasyGo+: DSRC Transaction for Tolling and Enforcement
	202-D		EASYGO+: RSE Functional Requirements
	202-E		EASYGO+: OBU Compatibility Tests