Copyright © 2005-2024 PayTec AG – Vogelsangstrasse 15 – 8307 Effretikon - Switzerland.
All Rights Reserved. PayTec AG license rules apply.

License
By using and/or copying this document, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:
Permission to copy, and distribute the contents of this document for the purpose of integrating electronic cash registers with terminals manufactured by PayTec AG, in any medium and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

- The pre-existing copyright notice of the original author, of the form: "Copyright © 2005-2019 PayTec AG – Vogelsangstrasse 15 – 8307 Effretikon - Switzerland. All Rights Reserved. PayTec AG document use rules apply”

When space permits, inclusion of the full text of this NOTICE should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.
No right to create modifications or derivatives of this document is granted pursuant to this license.

Disclaimers
THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.
The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

Project: **Primus**

Document: **Primus EFT ECR Interface**

Project:	Primus
Document:	Primus EFT ECR Interface
Revision n°:	3.0
Date:	2026/05/29
Author:	Markus Gagliardi, Marcel Sottnik, Pradeepan Iyakone, Christian Wäschle
Status:	confidential

Released by:

Position	Name	Date	Signature

Revision history

Rev. n°	Date	Who	Changes
1.0	2009/01/09	Mg	-First final version
1.1	2009/01/21	Mg, Mas	-Added truncated Application PAN for PCI/DSS compliance -Changed condition for plaintext PAN to m(referred) -Swapped values for transaction functions 'Combined' + 'Deposit' -Added ep2 MultiCurrencyFlag + postal banking product record -Changed AccountNumber to AccountNo -TrxTime and TrxDate introduced -AppPANStatic changed to AppPANEnc according to ep2 -Added ActSeqCnt to transaction response message -Added TrxSeqCntOri to transaction response message
1.2	2009/01/27	Mg	-Added PeSeqCnt + ActSeqCnt to activation response message -Added PeSeqCnt + ActSeqCnt to connect response message -Removed ActSeqCnt from transaction request message -Changed Tag for ActSeqCnt from '9F 84 11' to '9F 83 2A'
1.3	2009/02/02	Mg	-Added TrxRefNum to transaction response
1.4	2009/02/04	Mg	-Added AIDICC to transaction response
1.4.1	2009/02/08	Mg	-Added terminal language to connect response
1.4.2	2009/03/11	Mg	-Changed format of Amount, Available from BCD to Integer to allow negative values.
1.4.3	2009/05/05	Mg	-Changed maximum size of Amount, Available from 4 to 8 bytes
1.4.4	2009/08/10	Mg	-Corrected activation response example -Added support for external user interface
1.4.5	2009/08/18	Mg	-Added 'Locked' status -Added resource identifier to display notification message
1.4.6	2009/08/20	Mg	-Corrected display notification message example
1.4.7	2009/08/27	Mas	-Added IIN, ATR to status response and transaction response -Added DisplayName to transaction response
1.4.8	2010/01/06	Mas	-Added AID and AIDICC to status response -new status 'application selected' added to terminal status
1.4.9	2010/03/11	Mas	-Added new command for device command request/response and its parameter
1.5.0	2010-04-27	Mas	-Added new tag for e-signature in transaction response
1.6.0	2010-07-26	Mas	-Added AppCurrC tag nad WAITING FOR TRANSACTION REQUEST status in status response
1.7.0	2010-08-09	Mas	-Added WAITING_FOR_APPLICATION_SELECTION -Added TransactionInit_Request and Rsp Messages
1.7.1	2010-08-25	Mas	-Removed TransactionInit Req in favour of new DeviceCommandRequest codes
1.7.2	2010-11-05	Mas	-Added E-Signature flag to status response (Tag code to be defined) -Added CVMResults to transaction response -Added Extended transaction result to transaction response
1.7.3	2010-11-19	Mas	-Added tag code for the e-signature flag
1.7.4	2010-11-25	Mg	-Changed tage name from ESignatureActive to ESigInd (already used in PayTec specific ep2 SI(CONFIG) tag)

1.8.0	2010-12-17	Mas	-Added commands for Report and Receipt (content not yet defined)
1.8.1	2010-01-17	Mg	-Completed definitions for receipt & report exchange -Removed 'Confirm' field from Transaction Confirmation Response which does not contain this field -Removed 'ATR' and 'IIN' fields from Transaction Response which does not contain these fields -Added tip amount to transaction response
1.9.0	2011-09-21	Mg	-Added definitions for loyalty promotion transactions
1.10.0	2011-11-30	Mas	-New definitions for the new swiss post functions
1.11.0	2012-05-10	Mas	-New definitions for device handling -new device commands 2-4 -new fields in status response for Acquirers -config response changed
1.12.0	2012-05-28	Mas	-Added new stuff for new loyalty Iteration 5 - new commands AddBasket, ClearBasket - new response LoyaltyAdviceNotification -Added missing descriptions of Un-/LockEFToperations Req/Rsp
1.13.0	20120612	Mg	-Adjusted data format of status info -Moved software update info into status response
1.14.0	20121130	Mas	-added new functionality for JUMBO Card -TRANSACTION_DATA_CHANGE_REQUEST -TRANSACTION_DATA_CHANGE_RESPONSE -new tags in connect request -added new tag for abort transaction request for silent operation
1.15.0	20121204	Mg	-Adjusted tags & structure of TRANSACTION_DATA_CHANGE_REQUEST & TRANSACTION_DATA_CHANGE_RESPONSE
1.16.0	20121205	Mg	Discard additional tags concerning transaction data change reasons in connect request
1.17.0	20121206	Mg	Added Transaction Data Change Reason data for Bonus Card brands
1.18.0	20121211	Mg	Changed tag & conditions for silently aborting a transaction
1.19.0	20121212	Mg	Added Terms of Use
1.20.0	20130115	Mg	Added reporting of cards with IIN '000000' (myOne Cash)
1.21.0	20130311	Mg	Changed myOne Cash reporting from PAN to Track 2 Data
1.21.1	20130403	Mg	Corrected obsolete note about 'acknowledge' messages on page 11
1.22.0	20130409	Mg	Completed documentation of Aduno New Loyalty features
1.23.0	20130620	Mg	Added Aduno New Loyalty Field 'Campaign Sub Title' & 'Campaign Summary Text'
1.24.0	20131130	Mg	Corrected tag value of Loyalty Redemption Item
1.25.0	20140107	Mg	Added confirmation time extension request
1.26.0	20141106	Mg	- Added Cardholder Language to status response - Added Cardholder Text to confirmation time extension request - Restricted confirmation time extension to 0-30 seconds
1.27.0	20150127	Mg	Added set terminal language request message
1.28.0	20150302	Mg	Added PayModeValue to transaction request

1.29.0	20151021	Mg	Added TrxSeqCnt to transaction confirmation request
1.30.0	20161213	Mg	Added partial approval specific fields Amount, Remaining & Partial Approval Capability
1.31.0	20170126	Mg	Added serial transport layer
1.32.0	20170228	Mg	Marked Attendant Number and Attendant Name as optional
1.33.0	20170516	Mg	Corrected ranges for extended transaction result
1.34.0	20170531	Mg	- Added documentation of dialog functions - Added printing messages - Flag for silently starting a transaction
1.35.0	20170619	Mg	- Corrected documentation concerning reset of sequence number - Added tag 9F 84 17 (EP2 Static Key PAN Receipt Index)
1.36.0	20170627	Mg	Clarified presence of Transaction Sequence Counter in transaction confirmation request
1.37.0	20170703	Mg	Clarified presence of Transaction Sequence Counter in receipt request
1.38.0	20170712	Mg	Added dialog flags
1.39.0	20170712	Mg	Corrected Typo in dialog flags tag
1.40.0	20170822	Mg	Added status flags concerning online processing & receipt printer
2.0	20181101	Mg	- Added NDJSON as data format - Removed Merchant Password from connect & status responses - Removed Application Expiration Date from transaction response - Added hyperlinks for message types - Removed remark about preliminary values for postal banking transaction functions
2.1	20190225	Mg	- Added flag for silent transaction request - Added dialog resource 'Free text, card brands shown' - Added dialog parameters for pump selection & password checking - Added Surrogate PAN information to transaction request/response - Removed obsolete Features Aduno New Loyalty & Pay Mode Value
2.2	20190308	Mg	- Moved data type definitions at the end of the specification
2.3	20190403	Pi	- Added left and right custom button with configurable button text
2.4	20190617	Mg	- Added petrol dialogs for mileage & further info entry
2.5	20190709	Mg	- Added sequence diagrams describing the message flows of the most important use cases
2.6	20191002	Mg	- Added petrol dialog for function choice
2.7	20191022	Mg	- Added KeyPANRctDOLInd for ep2 7.0.0
2.8	20191112	Mg	- Added device command for setting display text when no ECR or vending machine has connected

2.9	20200108	Mg	- Added transaction functions 'cancel reservation', 'load' & 'activate card' - Added AcqID to transaction request for cancel reservation, purchase reservation (ep2 7), reservation adjustment (ep2 7) and confirm phone authorized reservation (ep2 7).
2.10	20200701	Mg	- Marked device command for setting display text when no ECR or vending machine has connected as obsolete
2.11	2020/09/07	Mg	- Added dialog flag to replace default idle text ‘Welcome’ & ‘Welcome, present card’
2.12	2020/12/03	Mg	- Added option to receive unsolicited receipts in connect request - Removed deprecated option ReportIINZeroCards from connect request
2.13	2021/02/23	Mg	- Added DCCOrigDate to transaction request - Removed obsolete field from example - Removed obsolete field PayModeValue - Removed duplicate tag
2.13.1	2021/03/29	Mg	- Clarified behavior of free text dialogs
2.14	2021/04/19	Mg	- Added extended transaction result to indicate pending task
2.15	2021/06/02	Mg	- Added option to suppress execution of scheduled tasks in connect request - Replaced field ‘NextSWUpdateDate’ in Status Response with ‘NextScheduledTask’ - Corrected length of LastReaderCleaningDate and LastAcqInitDate - Added missing Receipt Types - Added C4-specific Terminal Status ‘WWAN modem in use’ - Added device commands to start remote maintenance session & submit log data
2.16	2021/08/24	Mg	- Added Track3Data to status response - Updated condition for track 2 data presence
2.17	2021/10/25	Mg	- Added POS Entry Mode for QR-Code based payment - Clarified behaviour in case of multiple TCP connection requests
2.18	2021/11/11	Cw	- Added Device command for requesting the device to go to sleep mode - Formatting: Updated TOC to include clickable links
2.19	2022/03/10	Mg	- Added TrmID to connect response
2.20	2022/06/07	Mg	- Described flag parameters more precisely
2.21	2022/07/07	Mg	- Added description for printing inverse text and PNG images
2.22	2023/02/28	Mg	- Added device command to scan visual symbols
2.23	2023/05/30	Mg	- Added ‘UID’ field to status response - Added transaction request flag to report UID of unknown NFC tags
2.24	2024/06/04	Mg	- Added tip amount to reporting

			- Added DCC tags to JSON report example
2.25	2024/06/11	Mg	- Added Record Order Reference to transaction request
2.26	2024/08/20	Mg	- Added flag to signal offline status - Corrected some cross-references
2.27	2024/08/21	Mg	- Added Device Command to configure connectivity supervision
2.28	2024/09/02	Mg	- Added chapter about pre-authorization use cases
2.29	2024/09/19	Mg	- Added Transaction Amount to Transaction Response
2.30	2024/10/04	Mg	- Added PreAuthExpDate to transaction response
2,31	2024/11/27	Mg	- Added report type for open petrol authorizations
2.32	2025/02/02	Mg	- Added Issuer Identification Number to transaction response
2.33	2025/06/05	Mg	- Added generic password input template
2.34	2025/08/18	Mg	- Added CVM Code 3F ‘CVM not applicable’
2.35	2025/08/25	Mg	- Added transaction function for account verification - Added transaction function for purchase with cashback - Renamed account inquiry to balance inquiry - Removed references to obsolete transaction function Tip
2.36	2025/09/09	Mg	- Removed reference to obsolete list of error codes
2.37	2025/10/07	Cw	- Added device command to stop scanning for visual symbols
2.37.1	2025/11/12	Mg	- Corrected presence of OrderID from mandatory to optional
2.38	2026/03/02	Mg	- Added POSID to connect request - Corrected LastTrxSeqCnt to TrxSeqCnt in activation response - Corrected typos in TrmID and MctID in activation response - Added TextOnUnknownNFC to transaction request - Changed presence of DisplayName and CVMResults in transaction response to optional - Added BatchCaptureFlags to batch capture request - Corrected typo in transaction data change request example
3.0	2026/05/29	Mg	- Added secure WebSocket as transport layer

1 Table of content

1 Table of content 8
2 Document overview 10
2.1 Purpose and scope 10
2.2 Glossary and abbreviations 10
2.3 Notation 11
2.4 Format specifiers 11
2.5 References 12
3 Communication principles 13
3.1 Communication over TCP 13
3.2 Communication over RS232 14
3.2.1 Physical Layer 15
3.2.2 Frame Structure 15
3.2.3 Frame sequence 16
4 Messages 17
4.1 General message structure 17
4.1.1 Binary format with TLV-encoded data part 17
4.1.2 Newline Delimited JSON (NDJSON) based format 19
4.2 Message flows 20
4.2.1 Connection life cycle 20
4.2.2 Payment transaction 21
4.2.3 Terminal activation 22
4.2.4 Terminal deactivation 22
4.2.5 Final balance 23
4.2.6 Configuration 24
4.2.7 Initialization 24
4.2.8 Pre-authorization use cases 25
4.3 Message Details 29
4.3.1 Connect request 29
4.3.2 Connect response 31
4.3.3 Status request 32
4.3.4 Status response 32
4.3.5 Activation request 37
4.3.6 Activation response 37
4.3.7 Deactivation request 40
4.3.8 Deactivation response 40
4.3.9 Transaction request 41
4.3.10 Transaction response 45
4.3.11 Transaction confirmation request 50
4.3.12 Transaction confirmation response 51
4.3.13 Confirmation time extension request 52
4.3.14 Confirmation time extension response 53
4.3.15 Abort transaction request 53
4.3.16 Abort transaction response 53
4.3.17 Batch capture request 54
4.3.18 Batch capture response 54
4.3.19 Balance request 54
4.3.20 Balance response 54
4.3.21 Configuration request 56
4.3.22 Configuration response 56
4.3.23 Initialization request 56
4.3.24 Initialization response 56
4.3.25 Receipt request 57
4.3.26 Receipt response 58

4.3.27 Report request 59
4.3.28 Report response 59
4.3.29 Transaction data change request 63
4.3.30 Transaction data change response 63
4.3.31 Lock EFT operations request 64
4.3.32 Lock EFT operations response 64
4.3.33 Unlock EFT operations request 64
4.3.34 Unlock EFT operations response 64
4.3.35 Set terminal language request 65
4.3.36 Set terminal language response 65
4.3.37 Dialog request 66
4.3.38 Dialog response 67
4.3.39 Cancel UI request 67
4.3.40 Print receipt request 68
4.3.41 Print receipt response 68
4.3.42 Display notification 69
4.3.43 RFID status notification 70
4.3.44 Device command request 71
4.3.45 Device command response 71
4.3.46 Error notification 72
4.3.47 Heartbeat request 73
4.3.48 Heartbeat response 73
5 Data type definitions 74
5.1 Transaction function 74
5.2 Transaction result 74
5.3 Extended transaction result 75
5.4 Cardholder verification results 75
5.4.1 CVM Code 75
5.4.2 CVM Condition 75
5.4.3 CVM Result 76
5.5 Terminal status 76
5.6 RFID status 76
5.7 Device command code 77
5.8 Device command variable elements 77
5.9 Resource Identifier 78
5.10 Receipt Type 80
5.11 Receipt Flags 81
5.12 Report Type 81
5.13 Report Flags 81
5.14 Financial Counter Type 81
5.15 Transaction Data Change Reason Code 81
5.16 Bonus Card Brand Number 82
5.17 Transaction Request Flags 82
5.18 Abort Transaction Flags 82
5.19 Batch Capture Flags 82
5.20 Set of Transaction Data Change Reasons 82
5.21 Transaction Data Change Reason 83
5.22 Dialog Flags 84
5.23 Dialog Resource 84
5.24 Input Type 84
5.25 Dialog Button 84
5.26 Dialog Result 85
5.27 Printer Status 85
5.28 Transaction Request Flags 85

2 Document overview

2.1 Purpose and scope

This document defines a message based protocol between the EFT terminal and its client, i.e. an electronic cash register (ECR) or a vending machine.

2.2 Glossary and abbreviations

Abbreviation Term	Description
ACK	Acknowledge
ASN.1	Abstract Syntax Notation One
BCD	Binary Coded Decimal
BER	Basic Encoding Rules
ECR	Electronic Cash Register
EFT	Electronic Funds Transfer
JSON	JavaScript Object Notation, see http://json.org
LSB	Least Significant Byte
MSB	Most Significant Byte
NDJSON	Newline Delimited JSON, see http://ndjson.org
NTF	Notification
PCD	Proximity Coupling Device
PICC	Proximity Integrated Circuit Card
REQ	Request
RFU	Reserved for Future Use
RF	Radio Frequency
RSP	Response
TCP	Transmission Control Protocol
TMS	Terminal Management System
TLV	Tag Length Value

2.3 Notation

Notation	Meaning
0xnnnnnnnn	Integer represented as hexadecimal value nnnnnnnn
‘0’ – ‘9’, ‘A’ – ‘F’	Binary data, represented as hexadecimal
config	Present if configuration with TMS done
init	Present if acquirer initialization done
m	Mandatory
m (min..max)	Mandatory, from [min] to [max] occurrences
o	Optional
o (min..max)	Optional, from [min] to [max] occurrences
m(approved)	Mandatory if transaction is approved
m(ep2)	Mandatory in case of an ep2 transaction
m(referral)	Mandatory in case of a voice referral
m(auth purchase)	Mandatory if the transaction function is ‘Authorization purchase’
m(not aborted)	Mandatory in case of unaborted transaction
m(reversal)	Mandatory in case of a reversal transaction

2.4 Format specifiers

Notation	Meaning
b	Binary, encoded as octet string (TLV) resp. Base64 string (JSON)
bool	Boolean value, 0 / 1 (JSON) resp. ‘00’ / ‘01’ (TLV) means false / true
c	Constructed (TLV) resp. Object (JSON)
cn	Compressed numeric; see [**EMV**] Book 3, Chapter 4.3. JSON encoding: string containg the octets’ hexadecimal representa-tion
i	Integer, two’s complement encoding, MSB first (TLV) resp. JSON integer number
n	Binary coded decimal (TLV) resp. JSON number
s	UTF-8 encoded character string, without terminating 0 (TLV) resp. JSON string
tlv	Tag length value-encoded data, see [**EMV**], Book 3, Annex B for a summary
(min..max)	Value must be between and including [min] and [max]
(SIZE (n))	Value field has a fixed length of n bytes
(SIZE (min..max))	Value field must count at least [min] bytes and not exceed [max] bytes

2.5 References

Ref.	Document	Version
ep2	Eftpos 2000 terminal specification	4.1
BER	ITU-T Recommendation X.690 International Standard 8825-1: Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)	14 July 2002
EMV	EMVCo: EMV Terminal Specification	4.2
JSON	http://json.org	
NDJSON	http://ndjson.org	

3 Communication principles

TCP, secure WebSocket and a serial transport protocol are implemented as possible transport layers between the EFT terminal and its client.

3.1 Communication over TCP

The terminal listens for an incoming connection from the client on a defined TCP port (default: 8307) and uses the accepted client connection for exchanging messages.
Only one connection at a time is accepted; if another client from a different source IP address tries to connect on the same port, the connection will be refused. An additional connection from the same source IP address will be accepted and replaces the current TCP connection, which will be closed.

3.2 Communication over secure WebSocket

The terminal accepts secure WebSocket connections via wss:// on port 18443.
Only secure WebSocket is supported (no plain ws://).
The application protocol is identical to TCP, i.e. no separate WebSocket-specific API.
For TLS certificate validation, the terminal must be addressed by its hostname (not by IP address).
Use: wss://<terminal-hostname>:18443

Trusted Root CA certificate (to be installed on client side):


```
-----BEGIN CERTIFICATE-----
MIIF7jCCA9agAwIBAgIBATANBgkqhkiG9w0BAQsFADBYMQswCQYDVQQGEwJDSDET
MBEGA1UEBxMKRWZmcmV0aWtvbjESMBAGA1UEChMJUGF5VGVjIEFHMSAwHgYDVQQD
ExdQYXlUZWMgU2VydmljZXMgUm9vdCBDQTAgFw0yNjA0MTMxMDU2NTBaGA8yMDk5
MTIwNjEwNTY1MFowWDELMAkGA1UEBhMCQ0gxEzARBgNVBAcTCkVmZnJldGlrb24x
EjAQBgNVBAoTCVBheVRlYyBBRzEgMB4GA1UEAxMXUGF5VGVjIFNlcnZpY2VzIFJv
b3QgQ0EwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCiTpvcPlzxZ/85
x/Axky+qNnPI3xd/Hb7Jxucb35p7KCsXMLYPufC5FNLfmOZ9M6F3FLBaflN5HdK6
TBte4fOrTYqFYDPP2YvGeyB9iVd1wUasZvu2R4rRmrFldLLOgUtIv58bk2L9wZpZ
oRV/semMOB9RMUBC++K24pWzxzGb2HjHd9X5sEntDOk9cuAd+6/Ls9wk2vJU+iyZ
jfq3I+TevRDl9EIr92nxsvWspCf3X5wyDrPSduazZIpkb6Y1yd2XSx7Yu1G7Sba6
omz0+FZDF047jxMOqr5It6vIefOKGSaYBEnmdygXbvVEZaDBkzhqpOo439jrLSUE
ghLPXxd0gQiEI/WpsGTYz0BJzB7vjrbfE/dYLWW9ETABbLDNUZE3Y6/dqhtFkNHI
TgJMnsuTJOwIOX3EYolI1z42jxm3AzALZCUweZFKQH3EHPhNb05ejXw6GTfi0zHG
dI05iwzdZ16DkwK9pcoNLXClQZZO0rLFyYRhk0Mm7t1ZmMbXGyEPgZyJM3hFIsos
ryDq1UKpqWovzqYyAGqMGrxxjnvcFgfLYj2tLSlQcKIy1gY0Wqrzrwtixu A8rlyc
zI5EuXYAtIOlX8wWKLlDZQVrfJCtpkDhD3YXM2IdqIadBU8fBGd/EAX4g0MQHV3Z
TZDmd8BD000ZV/9i1rNl+0u82S4TCQIDAQABo4HAMIG9MB0GA1UdDgQWBBTgnVsQ
lgFIS6x9BP8GbAqTPyk/2TCBgAYDVR0jBHkwd4AU4J1bEJYBSEusfQT/BmwKkz8p
P9mhXKRaMFgxCzAJBgNVBAYTAkNIMRMwEQYDVQQHEwpFZmZyZXRpa29uMRIwEAYD
VQQKEwlQYXlUZWMgQUcxIDAeBgNVBAMTF1BheVRlYyBTZXJ2aWNlcyBSb290IENB
ggEBMAwGA1UdEwQFMAMBAf8wCwYDVR0PBAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IC
AQA+jqQHSYWJUI4+GqsYJO6BUaHunM+jFezGShrkPnl/9b6NOo02cONlor26cYAN
J9SQZgNgQcbA3zqjqcjvoiH56tEqu6Xj86r6k0Ted46Og94Z48zhpXfZG1wqBxX+
0mXAwDwqrmCktT3+6s+vLfkHMbim2sYMf3Xpkuajj9bv5SwX0oOVEuBIy3oW7OCS
G6yzXBxrQFZ0/mllWtQUb70lkMLwyRfa/VYqe/ae9+PaEEAX6y62o4N5meXRguTD
qCa/MBjiIv8HzAlVjZ+a1iwzv+yD8yMOxUtsZS0H9HRrX1vVEpBd/ys5dEvv5ZUf
U41cDyj2auTNxEADlVe6vBr7t30Ndhwb7cgPRIukOjLJS8NeADHfrjbxcaaCGn3l
ZdjuUMDd9r31npYCO+baDPmBSpN8WnP1IPKLKCPduscs5rMmVG2Wk614ZDQCt5ZJ
BfuBelRiDOcAIWzUx4L4AL/A3g1wIC3o0OB48AsQUM6s1OXJTHLAZGZDo2Uk0Cw9
p1Oclw2336TnEijJx8KCpCtPR1p1VdV2QoDVxNyXPS5dkf9ag19nsa7OrQ4rszza
T+CpeRTa4480buYiuDyiRcNkXCEzfBeT67FBgFkr/bqCt+r4K9zOS3O4CB8Ez7+O
xjHn3V5dywyXf3ixGpiIcf6xjFY1UcaPWW+ESpCftd36Ig==
-----END CERTIFICATE-----
```

3.3 Communication over RS232

3.3.1 Physical Layer

The protocol is packet oriented and uses a rate of 115200 baud, 1 start bit, 1 stop bit, 8 data bits, no parity and no flow control.

3.3.2 Frame Structure

The framing is achieved by using escape sequences; the escape character is 10h.
Every 10h character between the begin and the end of an information frame must be escaped, including the sequence number, frame type and CRC.

Sequence	Meaning
10h 00h	Begin of information frame
10h FFh	End of information frame
10h 10h	Character 10h within an information frame
10h 11h	ACK (acknowledged) frame
10h 12h	NACK (not acknowledged) frame

To detect transmission errors, a cyclic redundancy checksum is inserted between the information frame's content and the end of the frame. The CRC is calculated over every byte from and including the sequence number until the last content byte before the CRC; the calculation is done before escaping the frame's content.

C sample code for computing the CRC:

```
unsigned char  content[] = { 0x01, 0x11, 0x12, 0x34, 0x56, 0x78 };
unsigned char  message[(sizeof(content) + 2) * 2 + 4];
unsigned short crc = 0;
int            i, length;

length = 0;

message[length++] = 0x10;
message[length++] = 0x00;

for (i = 0; i < sizeof(content); i++) {
      if (content[i] == 0x10)
            message[length++] = 0x10;

      message[length++] = content[i];
}

for (i = 0; i < sizeof(content); i++) {
      crc = (unsigned char)(crc >> 8) | (crc << 8);
      crc ^= content[i];
      crc ^= (unsigned char)(crc & 0xff) >> 4;
      crc ^= (crc << 8) << 4;
      crc ^= ((crc & 0xff) << 4) << 1;
}

if ((crc >> 8) == 0x10) message[length++] = 0x10;
message[length++] = (unsigned char)(crc >> 8);
if ((crc & 0xFF) == 0x10) message[length++] = 0x10;
message[length++] = (unsigned char)(crc & 0xFF);

message[length++] = 0x10;
message[length++] = 0xFF;
```

Frame types:

Frame type	Name
11h	Information frame; ECR → terminal
91h	Information frame; terminal → ECR

Content of Information Frame:

Seq-No	Frame Type	Data	CRC
1 byte	1 byte	1..n bytes	2 bytes

3.3.3 Frame sequence

The data part of the information frame is defined here. Every information frame must be acklowledged with an ACK frame or rejected with a NACK frame before sending the next information frame. After a timeout of 350 ms without an ACK frame, an information frame can be considered lost and shall be resent. Any error like CRC failed or unknown frame type while receiving a frame is reported with a NACK frame. When receiveing a NACK frame, the information frame is sent again immediately.

The sequence number is incremented individually by each communication party. Repetitions are recognized by a sequence number which has the same value as the one of the last information frame received

After sending an information frame 3 times without receiving ACK, the transmission is considered unsuccessful and an Error can be returned to the upper layer.

4 Messages

4.1 General message structure

The following Formats are supported:

Format	Pros	Cons
Binary message format	more compact, simple parsing	bad human readability, web-unfriendly
NDJSON based format	human readable, web- and scripting-friendly	less compact

The maximum message size is 20kb including all header fields or line separators.

4.1.1 Binary format with TLV-encoded data part

Field name	Format	Description
Message length	i (SIZE(4))	Total message length in bytes, excluding the length bytes
Magic number	b (SIZE(4))	Constant value ‘20 08 08 26’
Sequence number	n (1..9999) (SIZE(2))	Internal message counter of each communication party. Starts at 1 and is reset with terminal restart or when the number has reached 9999. 0 is not a valid value.
Protocol version	b (SIZE(1))	Constant value ‘01’
Message type	n (SIZE(1))	‘01’: Connect request ‘02’: Connect response ‘03’: Status request ‘04’: Status response ‘05’: Activation request ‘06’: Activation response ‘07’: Deactivation request ‘08’: Deactivation response ‘09’: Transaction request ‘10’: Transaction response ‘11’: Transaction confirmation request ‘12’: Transaction confirmation response ‘13’: Abort transaction request ‘14’: Abort transaction response ‘15’: Batch capture request ‘16’: Batch capture response ‘17’: Balance request ‘18’: Balance response ‘19’: Configuration request ‘20’: Configuration response ’21’: Initialization request ‘22’: Initialization response ‘23’: Receipt request ‘24’: Receipt response

		‘25’: Report request ‘26’: Report response ‘27’: Transaction data change request ‘28’: Transaction data change response ‘29’: Confirmation time extension request ‘30’: Confirmation time extension response ‘31’: Lock eft operation request ‘32’: Lock eft operation response ‘33’: Unlock eft operation request ‘34’: Unlock eft operation response ‘35’: Set terminal language request ‘36’: Set terminal language response ‘41’: Dialog request ‘42’: Dialog response ‘43’: Cancel UI request ‘51’: Display notification ‘53’: RFID status notification ‘81’: Print receipt request ‘82’: Print receipt response ‘95’: Device command request ‘96’: Device command response ‘97’: Heartbeat request ‘98’: Heartbeat response ‘99’: Error notification
Data	tlv	Data part, optional.

4.1.2 Newline Delimited JSON (NDJSON) based format

The message is represented as a JSON object on a single line, terminated with a newline (0x0A) character.
For better readability, the JSON message examples are pretty-printed, newline within strings are presented as line breaks:

```
{
    "ReceiptResponse": {
        "ReceiptText": "      Any Shopping
     1000 Anywhere
"
    }
}
```

means:

```
{ "ReceiptResponse": { "ReceiptText": "      Any Shopping\n     1000 Anywhere\n" }}<newline>
```

- Encoding must be UTF-8
- Other white space than newlines are ignored
- Newlines in strings are escaped as “\n”

Constructed data elements are represented as JSON arrays if they contain multiple elements of the same type:

```
"Currencies": [ 756, 840, 978 ]

"SetAcqInfo": [
    {
        "AcqID": 1,
        "LastAcqInitDate": "2018-09-18 14:45:15"
    },
    {
        "AcqID": 4,
        "LastAcqInitDate": "2018-10-29 11:06:35"
    }
]
```

If they can contain members of different type, they are represented as JSON objects:

```
"ActivationRequest": {
    "AttendantName": "Alice",
    "AttendantNumber": 12
}
```

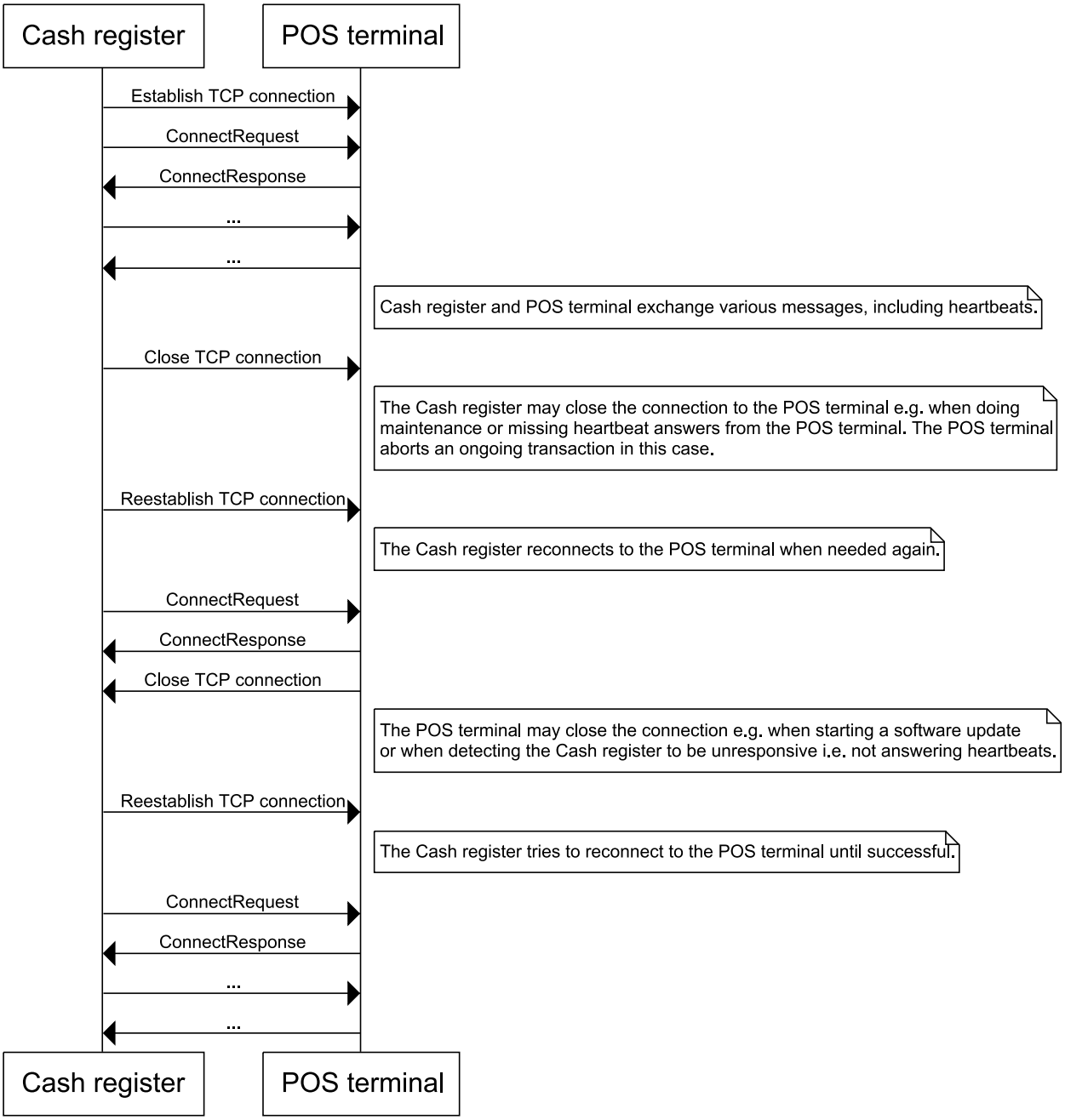
Conventions:

In case of the binary message format, the TLV-formatted part of the message is an unordered set of subelements, i.e. the subelements can appear in any order and are only recognized by their tag.

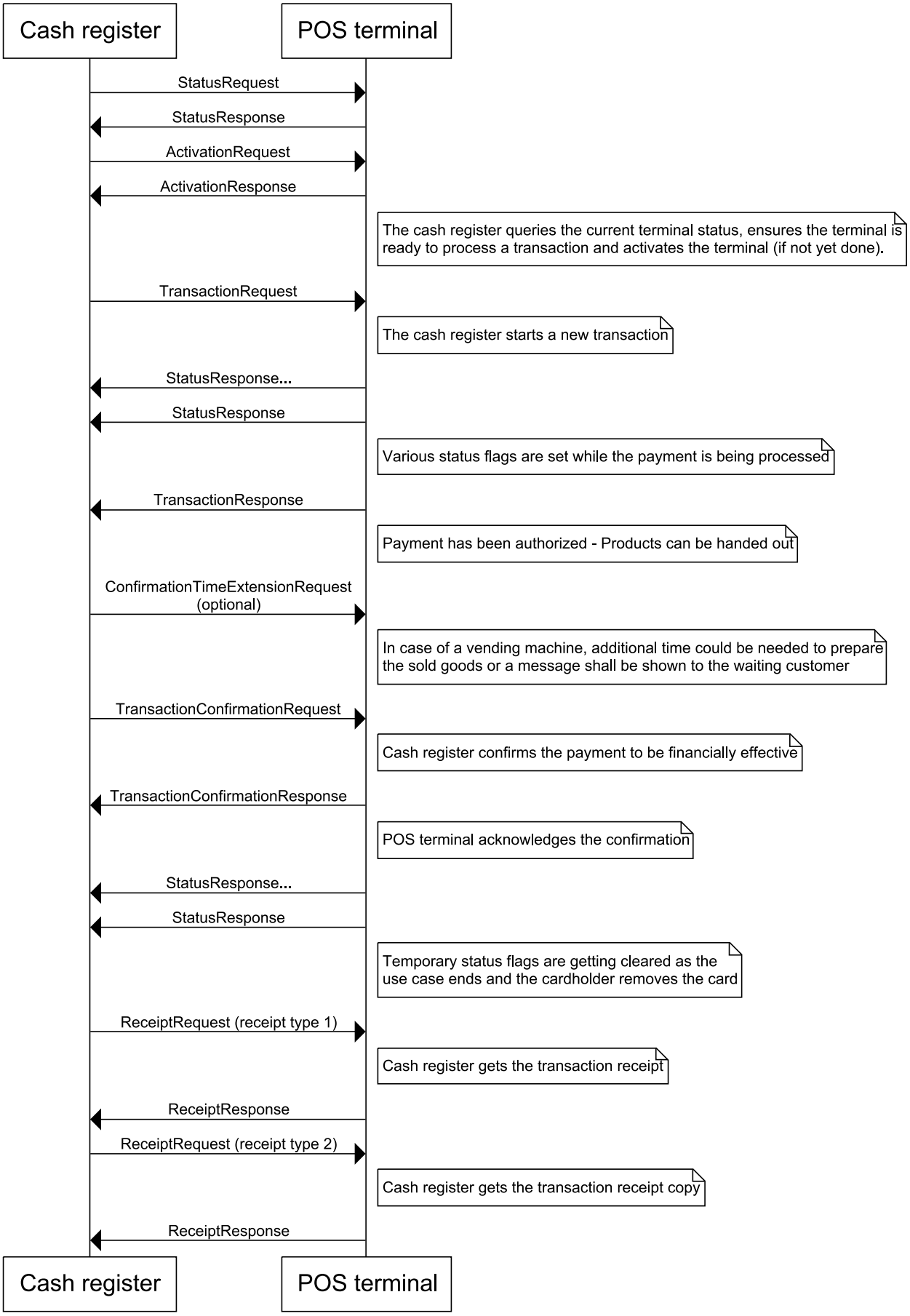
In case of NDJSON format, the root element is an object containing the message object as its only member. The message object can contain multiple members in any order.

4.2 Message flows

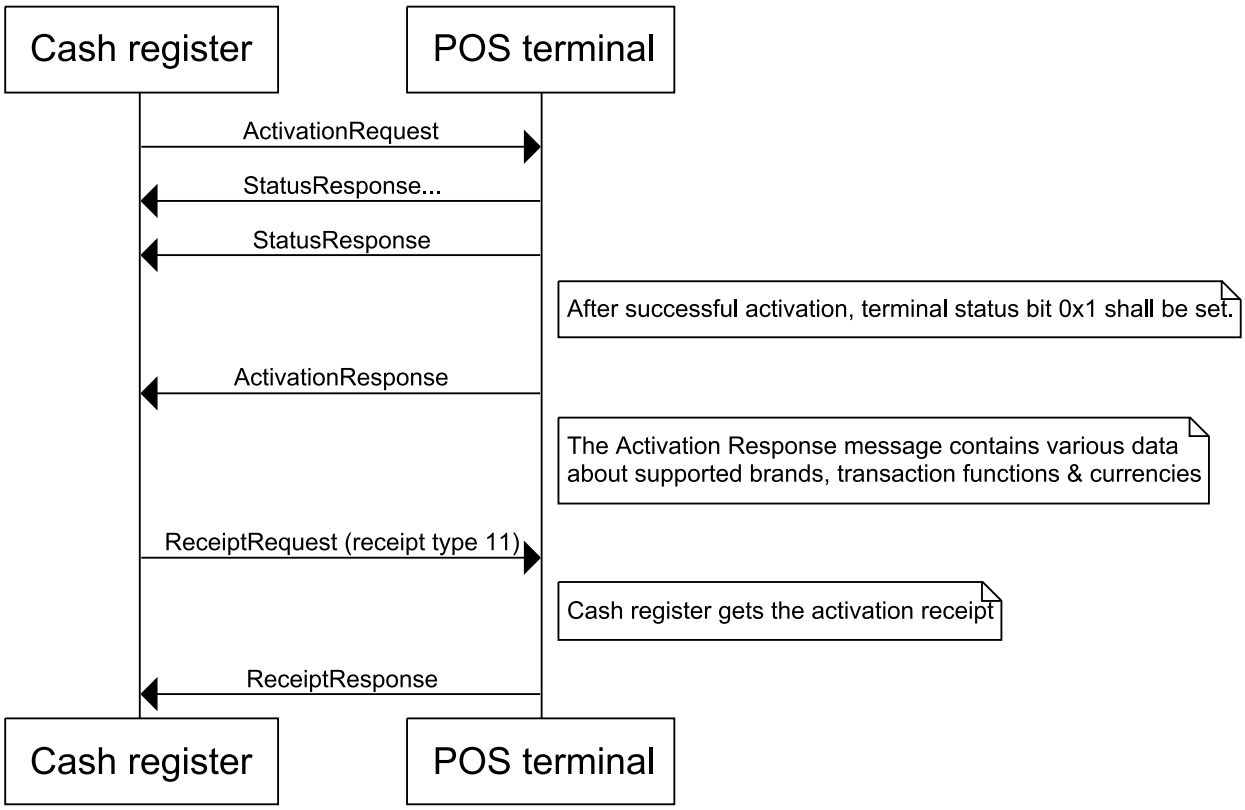
4.2.1 Connection life cycle



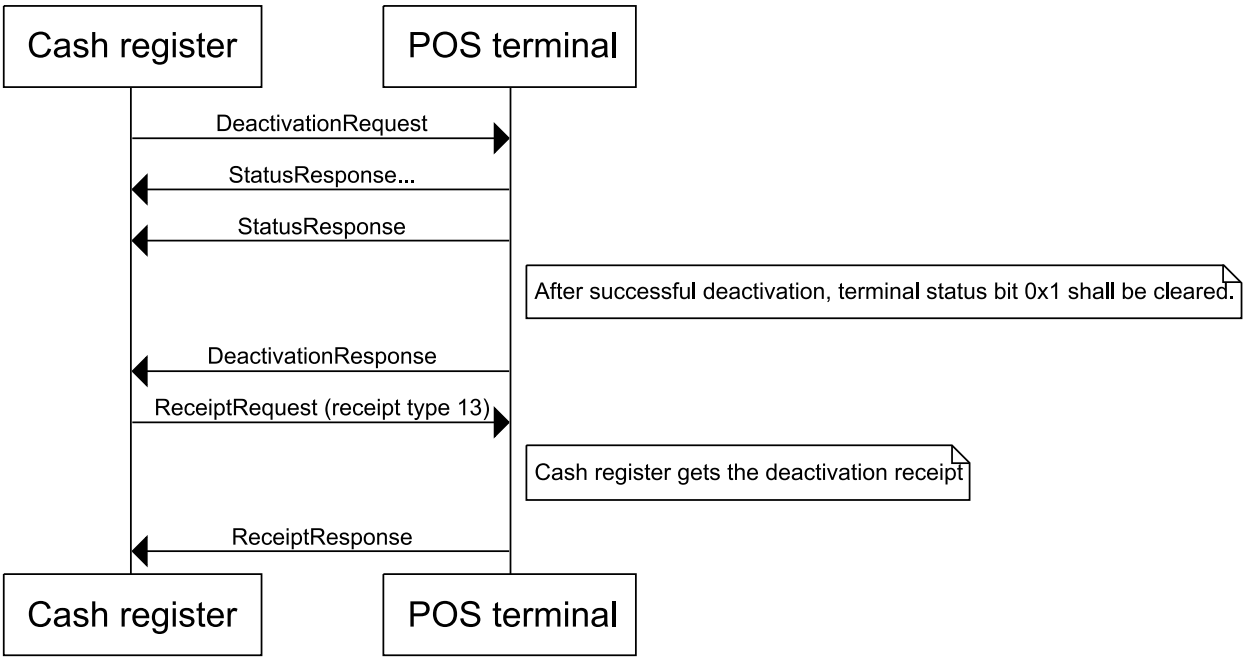
4.2.2 Payment transaction



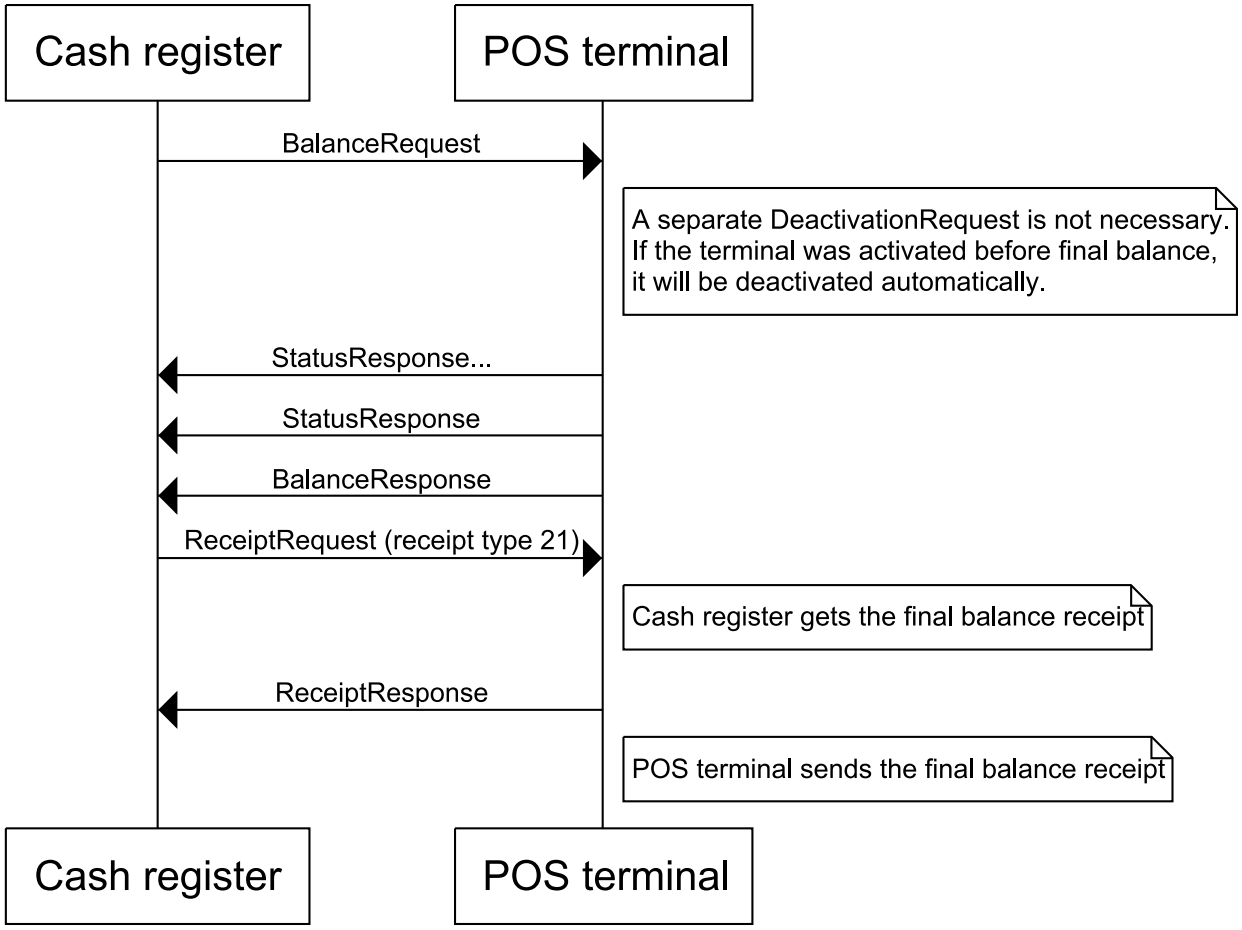
4.2.3 Terminal activation



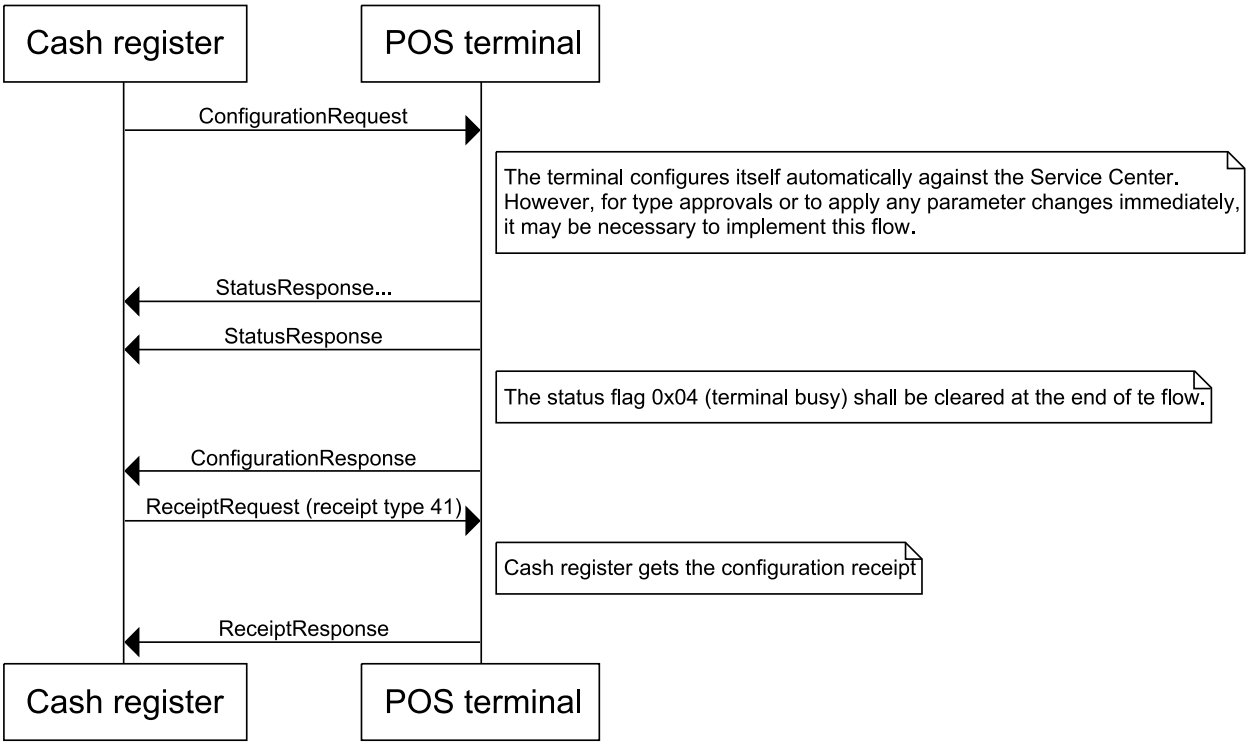
4.2.4 Terminal deactivation



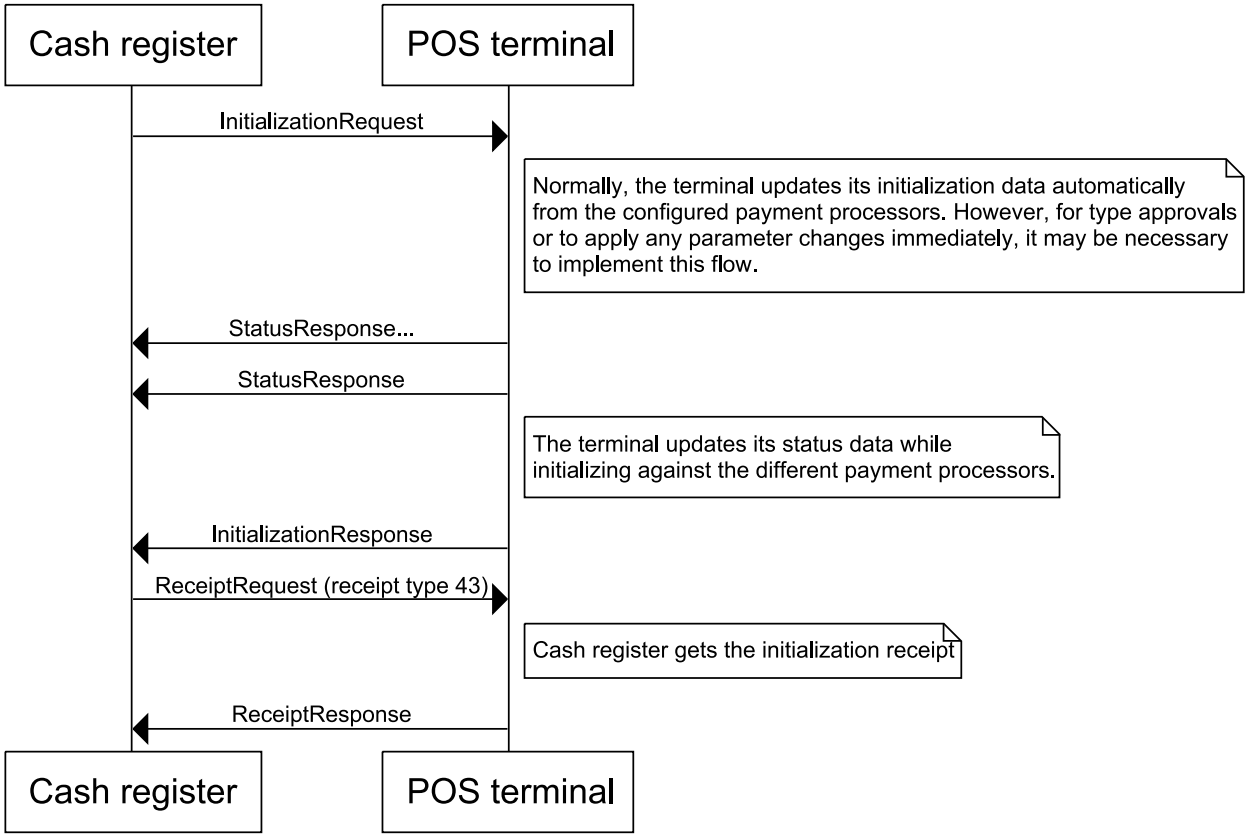
4.2.5 Final balance



4.2.6 Configuration



4.2.7 Initialization



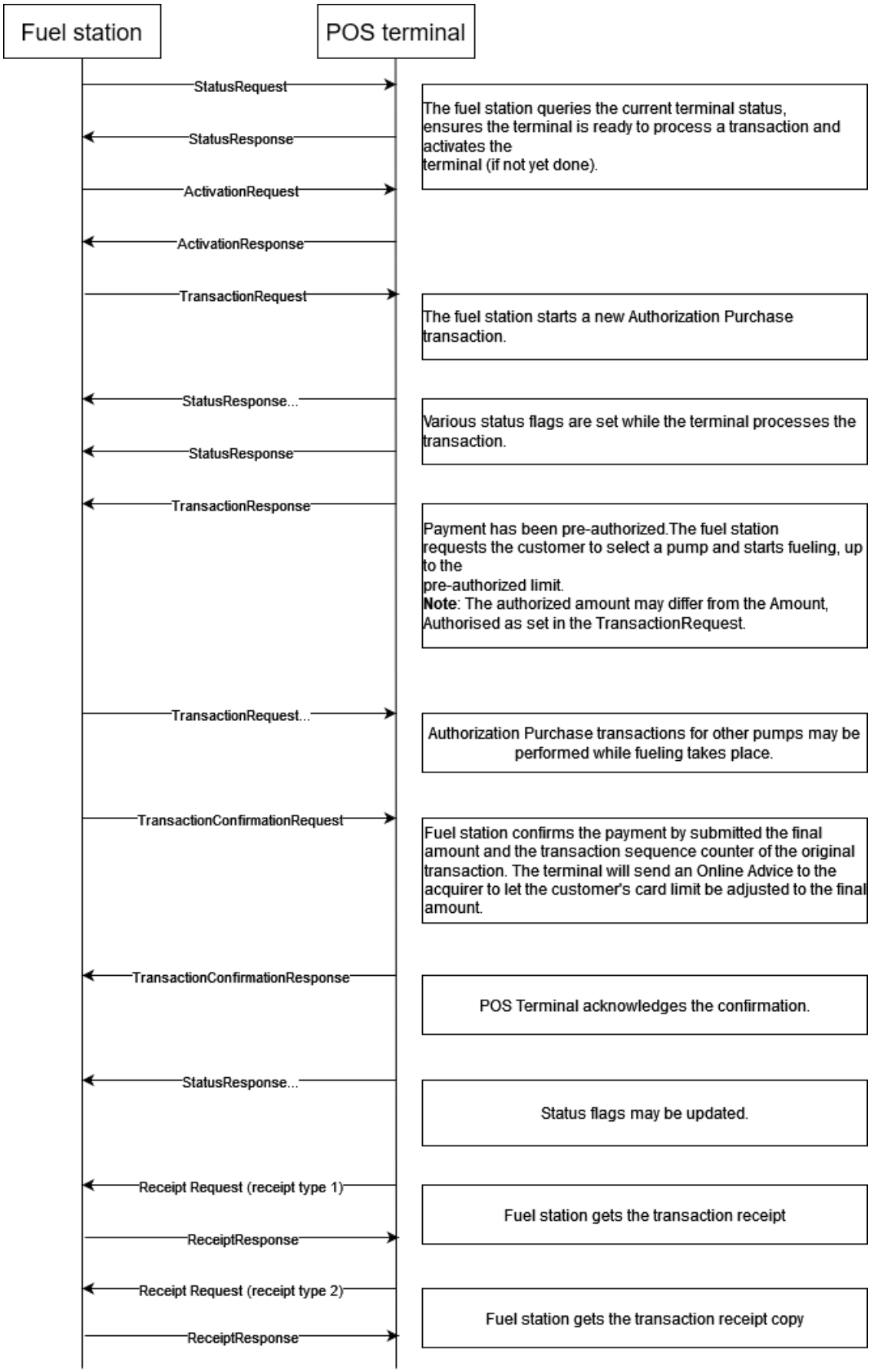
4.2.8 Pre-authorization use cases

The terminal supports two separate use cases for different pre-authorization scenarios:

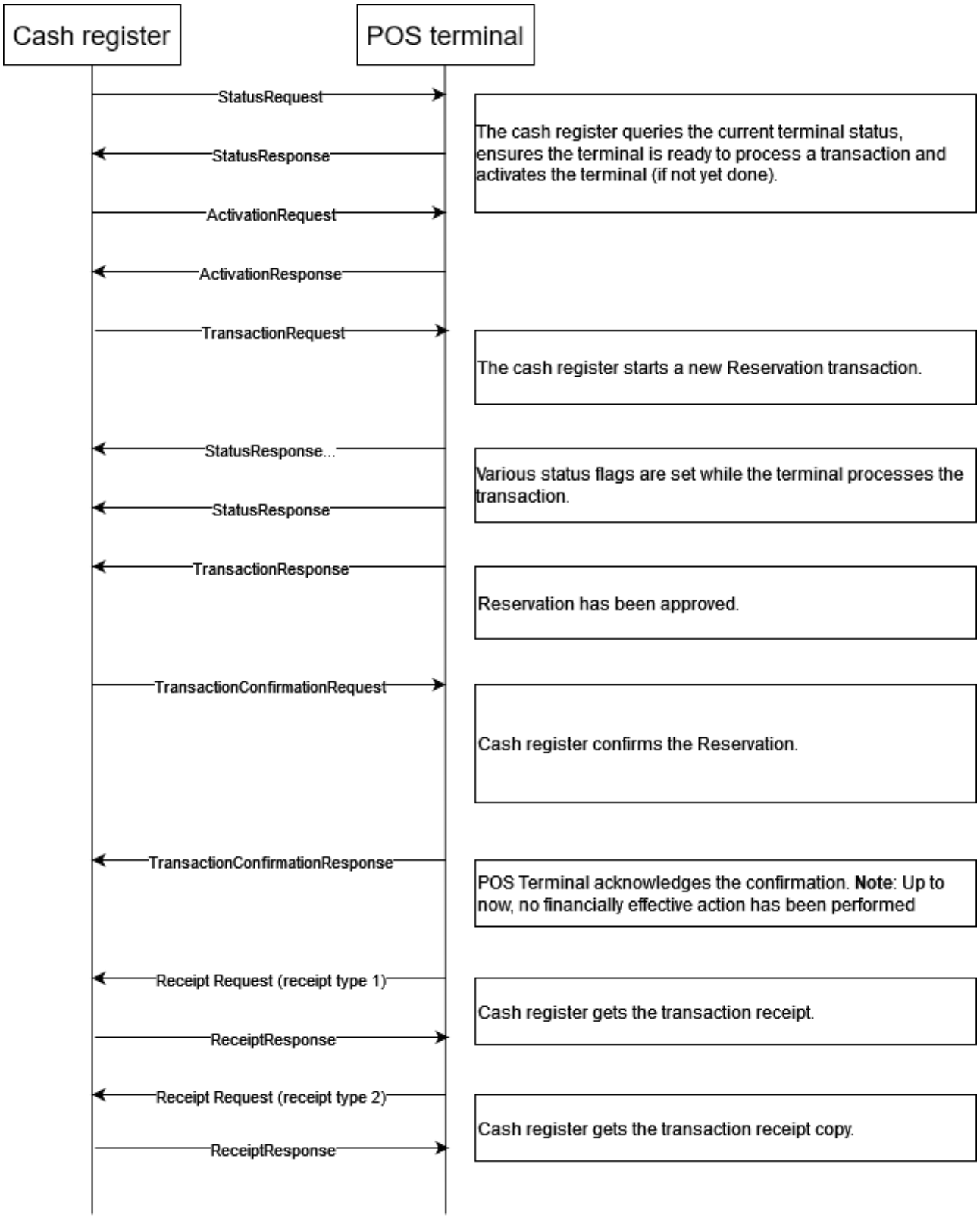
- **Authorization Purchase:** This is the transaction type provided by [ep2] for the purchase of fuel at a filling station

- **Reservation / Purchase Reservation / Reservation Adjustment / Confirmed Phone Authorized Reservation / Cancel Reservation:** These transaction types are designed to be used for pre-authorization use cases where the final amount may be provided later than a few minutes after pre-authorization. This includes the hospitality use cases and is also recommended by ep2 for electric vehicle (EV) charging.
  Since ep2 version 7.0, this use case has a reference-based approach i.e. only the initial Reservation transaction is a card-present transaction while all the secondary transactions are performed by providing the reference number and Acquirer Identifier of the original transaction. This means they may also be processed on a different terminal.

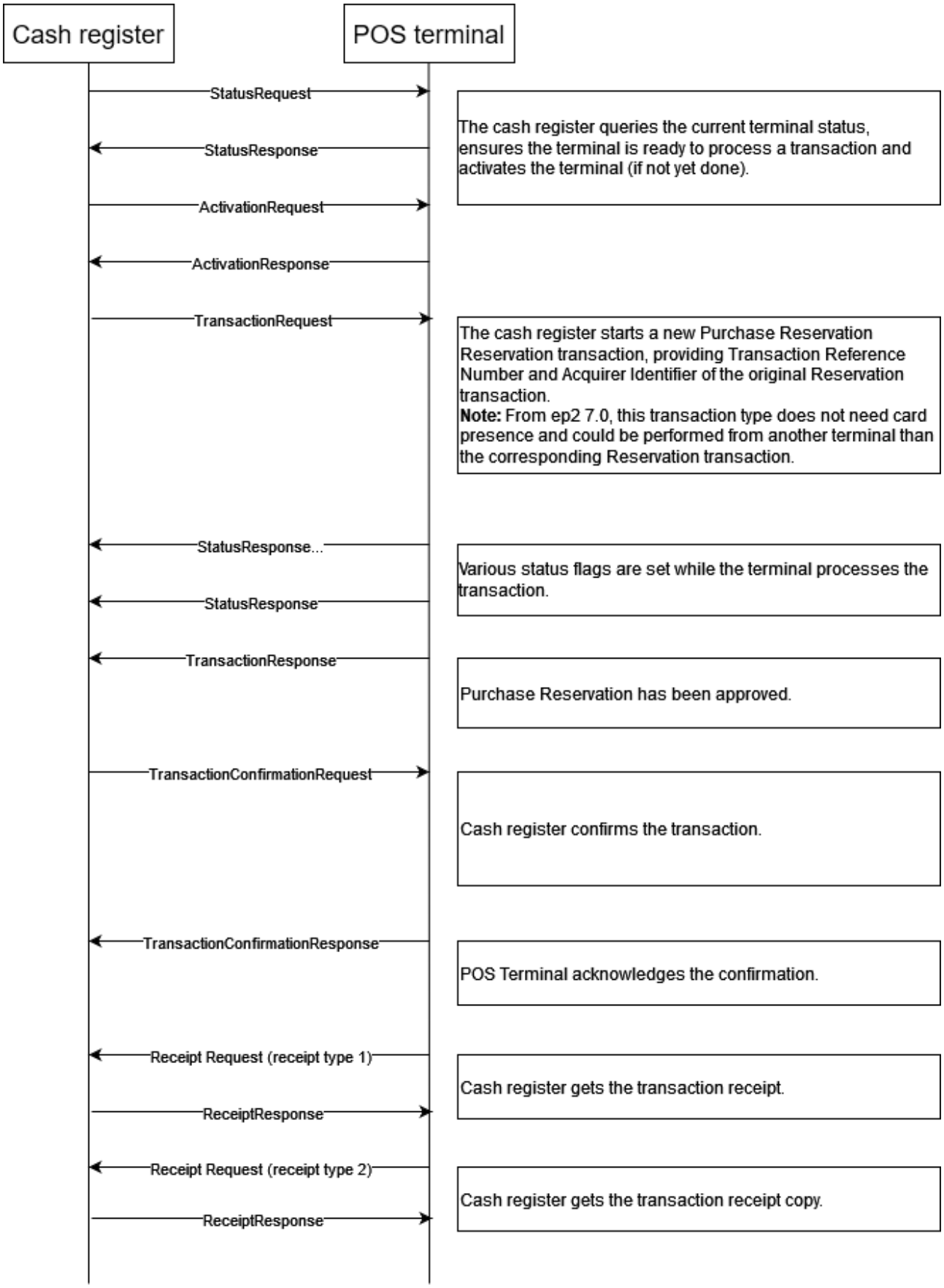
4.2.8.1 Authorization Purchase



4.2.8.2 Reservation



4.2.8.3 Purchase Reservation



4.3 Message Details

4.3.1 Connect request

This message is sent by the client after establishing the connection to exchange basic information.

Data format:

Tag	Name		Format	Description	Presence
'31'	ConnectRequest		c		m
	'9F 83 05'	TrmLng	s (SIZE(2))	Terminal language as an ISO 639-1 2 letter code, e.g. 'en', 'de'	o
	'9F 83 06'	PrinterWidth	n	Maximum numer of characters to be printed on one line	o
	'9F 83 2B'	UseExternalDisplay	bool	If this field is present and true, an external cardholder display is used (e.g. in case of a vending machine integration); all display messages and RFID status changes are forwarded to the cash register.	o
	'BF 83 49'	SetTrxDataChangeReason	c	Set of transaction data change reasons which should be activated	o
	'9F 83 58'	UnsolicitedReceipts	bool	If this field is present and true, every receipt generated by the terminal is forwarded as receipt response to the cash register, e.g. when scheduled service functions are performed.	o
	'9F 83 59'	SuppressScheduledTasks	bool	If this field is present and true, no scheduled tasks like e.g. initialisations, batch data captures etc. shall be performed. This is useful if the cash register is about to start a	o

				transaction soon and wants the terminal to not execute any scheduled task for now. This also prevents a device that supports power saving from entering standby/sleep mode. The terminal will resume processing of scheduled tasks as soon as another Connect Request without this flag set to true is received or the cash register disconnects.	
	‘9F 83 76’	POSID	s	Point of sale identifier	o

Example (binary format):

```
31 1E                       -- Connect request, length 30 bytes
   9F 83 05 02 64 65        -- languageCode ‘de’
   9F 83 06 01 30           -- 48 characters per line
   9F 83 4F 01 02           -- report myOne cash card data
   BF 83 49 0F              -- set of active change reasons
      BF 83 4A 01 0A        -- change reason
         9F 83 4B 01 01     -- myOne discount detection
```

Example (JSON):

```
{
    "ConnectRequest": {
        "TrmLng": "it",
        "PrinterWidth": 40
    }
}
```

4.3.2 Connect response

This message is sent by the EFT terminal in return of a connect request message.

Data format:

Tag	Name		Format	Description	Presence
‘31’	ConnectResponse		c		m
	‘9F 83 05’	TrmLng	s (SIZE(2))	Terminal language as an ISO 639-1 2 letter code, e.g. ‘en’, ‘de’.	m
	‘9F 83 1D’	SoftwareVersion	n (SIZE(1..3))	Software version. Consists of a mayor, minor and patch version from 0 to 99. Example: 20911 = 2.9.11	m
	‘9F 1E’	IFDSerialNum	s (SIZE(8))	Interface device serial number, as defined by EMVCo	m
	‘9F 83 29’	PeSeqCnt	n(SIZE(1..4))	Current Booking Period Sequence Counter	m
	‘9F 83 2A’	ActSeqCnt	n(SIZE(1..4))	Current Activation Sequence Counter	m
	‘9F 1C’	TrmID	s (SIZE(8))	Terminal identification	o (if terminal software version >= 20.x)

Example (binary format):

```
31 1D                                   -- Connect response, length 36 bytes
   9F 83 1D 03 02 09 11                 -- Terminal software version 2.9.11
   9F 1E 08 31 32 33 34 35 36 37 38     -- Language code ‘de’
   9F 83 29 01 05                       -- Booking period 5
   9F 83 2A 01 08                       -- Activation sequence counter 8
```

Example (JSON):

```
{
    "ConnectResponse": {
        "TrmLng": "it",
        "IFDSerialNum": "ZVD00020",
        "SoftwareVersion": 100100,
        "PeSeqCnt": 4,
        "ActSeqCnt": 10,
        "TrmID": "30143007"
    }
}
```

4.3.3 Status request

This message is sent by the client to query the status of the EFT terminal.

Data format: Empty

Example (JSON):

```
{
    "StatusRequest": {}
}
```

4.3.4 Status response

This message is sent by the EFT terminal in return of a status request message.
This message can be sent spontaneously to notify the client about status changes.

Data format:

Tag	Name		Format	Description	Presence
'31'	StatusResponse		c		m
	'9F 83 0B'	TrmStatus	i	Bitwise OR of terminal status values	m
	'9F 83 05'	TrmLng	s (SIZE(2))	Terminal language as an ISO 639-1 2 letter code, e.g. 'en', 'de'. This value may change after a downloading configuration data from the TMS	m
	'42'	IIN	n(SIZE(6))	Issuer Identification Number (first 6 digits of the Primary Account Number found in the magnetic stripe track 2 data)	o (present if TrmStatus indicates a valid card in the contact reader and magstripe is available)
	'9F 83 2F'	ATR	b(SIZE(0..32))	Answer To Reset of the ICC	o (present if TrmStatus indicates a valid card in the contact reader and an ICC has been successfully read)
	'9F 06'	AID	b (SIZE(6..17))	Application Identification Number of currently selected Application of the inserted card	o (present if TrmStatus indicates an selected application)

Tag	Name		Format	Description	Presence
	‘4F’	AIDICC	b (SIZE(6..17))	Application Identification, as read from the IC card. May differ in case of partial selection (see EMVCo specs). Sent in case of an ICC transaction that is not aborted before completion.	o
	‘9F 42’	AppCurrC	n(SIZE(3))	Application Currency Code, as read from IC card.	o(present if TrmStatus indicates waiting for transaction request and data element is present on the EMV card)
	‘9F 44’	AppCurrExp	n(SIZE(1))	Application Currency Exponent	o(present if TrmStatus indicates waiting for transaction request and data element is present on the EMV card)
	‘9F 6B’	Track2Data	b (SIZE(9..20)), as defined for Track 2 Equivalent Data in EMV/Book 3/Annex A1	Magstripe ISO track 2 data.	o (only present if beginning of track is included in the PCI whitelist)
	‘9F 83 00’	Track3Data	b (SIZE(9..54))	Magstripe ISO track 3 data; currently used for processing of petrol cards out of terminal scope. If such a card is detected, the transaction is silently aborted to hand over the processing to the vending machine,	o (only present if beginning of track is included in the PCI whitelist, the transaction is started silently and track 2 is not present)
	‘9F 83 77’	UID	b(SIZE(4..10))	NFC Unique ID	o (present if Mifare processing is enabled and terminal detects an unknown Mifare tag)
	‘9F 83 34’	ESigInd	bool	E-Signature indicator	m

Tag	Name		Format	Description	Presence
	‘9F 83 45’	LastReader-CleaningDate	s(SIZE(19))	Local date/time of the last cleaning: yyyy-MM-dd HH:mm:ss	o (present if at least one cleaning with the current motor reader was executed)
	‘BF 83 48’	SetAcqInfo	Set of AcqInfo	Set of Acquirer Information; contains all acquirers which are configured, regardless of their initialization status.	o (present if EP2 application is running)
	‘9F 83 50’	CardholderLng	s(SIZE(2))	Cardholder Language	o
	‘BF 83 62’	NextScheduled-Task	c	The task that will be performed next	o

AcqInfo:

Tag	Name		Format	Description	Presence
‘BF 83 47’	AcqInfo		c	Acquirer Information	m
	‘9F 01’	AcqID	n	Acquirer Identifier	m
	‘9F 83 44’	LastAcqInitDate	s(SIZE(19))	Local date/time of the last successful initialization: yyyy-MM-dd HH:mm:ss	o (present if acquirer has been successfully initialized)

NextScheduledTask:

Tag	Name		Format	Description	Presence
‘BF 83 62’	NextScheduledTask		c	Next scheduled task	m
	‘9F 83 63’	TaskName	s	Task name, one of: ‘CONFIG’ ‘CONFIG_FORCED’ ‘INIT’ ‘INIT_FORCED’ ‘SUBMISSION’ ‘TRANSMISSION’ ‘FINAL_BALANCE’ ‘SW_UPDATE’ ‘REBOOT’ ‘AUTO_REVERSAL’ ‘DECOMMISIONING’	m
	‘9F 83 64’	LastRun	s(SIZE(19))	Local date/time when this task was last performed: yyyy-MM-dd HH:mm:ss	o (present if task has already been performed at least once)
	‘9F 83 65’	NextRun	s(SIZE(19))	Local date/time when this task will be performed next: yyyy-MM-dd HH:mm:ss	m
	‘9F 01’	AcqID	n	Acquirer Identifier	o (present if TaskName is ‘INIT’ or ‘INIT_FORCED’)

Example (binary format):

```
31 27                          -- Status response, length 39 bytes
   9F 83 0B 01 05              -- Terminal is activated, no card present, terminal is busy
   9F 83 05 02 64 65           -- Language code ‘de’
   42 03 99 88 77              -- Issuer Identification Number 998877
   9F 83 2F 18
        3B EF 00 FF 81 31 FE 45 65 63 0D 07 63 07 64 00
        0D 90 58 45 00 06 15 84       -- Answer To Reset
   BF 83 62 38                 -- Next scheduled task:
      9F 83 63 06 43 4F 4E 46 49 47 – Task name ‘CONFIG’
      9F 83 64 13 – last run: 2021-05-10 13:53:43
         32 30 32 31 2D 30 35 2D 31 30 20 31 33 3A 35 33 3A 34 33
      9F 83 65 13 – next run: 2021-05-11 05:49:00
         32 30 32 31 2D 30 35 2D 31 31 20 30 35 3A 34 39 3A 30 30
   BF 83 48 3E – Set of acquirer info
      BF 83 47 1B – Acquirer into
         9F 01 01 29 – Acquirer ID 29
         9F 83 44 13 – last init: 2021-05-05 17:23:38
            32 30 32 31 2D 30 35 2D 30 35 20 31 37 3A 32 33 3A 33 38
      BF 83 47 1B
         9F 01 01 02 – Acquirer ID 02
         9F 83 44 13 – last init: 2021-05-05 17:23:32
            32 30 32 31 2D 30 35 2D 30 35 20 31 37 3A 32 33 3A 33 32
```

Example (JSON):


```
{
    "StatusResponse": {
        "MctPwd": "0",
        "TrmStatus": 1,
        "TrmLng": "de",
        "CardholderLng": "de",
        "ESigInd": 1,
        "NextScheduledTask": {
            "TaskName": "CONFIG",
            "LastRun": "2021-05-10 13:53:43",
            "NextRun": "2021-05-11 05:49:00"
        },
        "SetAcqInfo": [
            {
                "AcqID": 29,
                "LastAcqInitDate": "2021-05-05 17:23:38"
            },
            {
                "AcqID": 2,
                "LastAcqInitDate": "2021-05-05 17:23:32"
            }
        ]
    }
}
```

4.3.5 Activation request

This message is sent by the client to let the terminal get ready for transaction processing and open a new user shift.

Data format:

Tag	Name		Format	Description	Presence
'31'	ActivationRequest		c		m
	'9F 83 07'	AttendantNumber	n	Attendant number	o
	'9F 83 08'	AttendantName	s (SIZE(16))	Attendant name	o

Example (binary format):

```
31 0F                          -- Activation request, length 15 bytes
   9F 83 07 02 12 34           -- Attendant number 1234
   9F 83 08 05 53 6D 69 74 68  -- Attendant Smith starts working
```

Example (JSON):

```
{
    "ActivationRequest": {
        "AttendantName": "Alice",
        "AttendantNumber": 12
    }
}
```

4.3.6 Activation response

This message is sent by the EFT terminal in return of an activation request message.

Data format:

Tag	Name		Format	Description	Presence
'31'	ActivationResponse		c		m
	'9F 1C'	TrmID	s (SIZE(8))	Terminal identification	m
	'9F 35'	TrmType	n (SIZE(1))	Terminal Type as defined in [EMV] Book 4, Annex A1	m
	'9F 83 05'	TrmLng	s (SIZE(2))	Terminal language, as configured by the TMS or the client resp. the default terminal language	m
	'9F 83 29'	PeSeqCnt	n(SIZE(1..4))	Current booking period sequence counter	m
	'9F 83 2A'	ActSeqCnt	n(SIZE(1..4))	Current activation sequence counter	m
	'9F 41'	TrxSeqCnt	n (0..99999999)	Transaction sequence counter of the most recent transaction. If no transactions have been processed so far, the value 0 is used.	m

Tag	Name			Format	Description	Presence
	‘9F 1A’	TrmCntryC		n (SIZE(1..2))	ISO-3166 code of the country where the terminal is installed	config
	‘9F 16’	MctID		s (SIZE(15))	Merchant identifier	config
	‘9F 4E’	MctNameLoc		s (SIZE(1..75))	Merchant name and location	config
	‘9F 83 14’	TrxFunctions		i	Bitwise OR of the supported transaction functions	m
	‘BF 83 0C’	Currencies		c	Set of supported currencies	m
		‘5F 2A’	Trx-CurrC	n	ISO-4217 currency code	o (0..32)
	‘BF 83 0D’	Brands		c	Set of supported card brands	m
		‘9F 83 09’	Brand	s (SIZE(1..19))	Card brand	o (0..32)

Example (binary format):

```
31 81 8B                                  -- Activation response, length 137 bytes
   9F 1C 08 33 30 31 34 30 30 31 38       -- Terminal identification ‘30140018’
   9F 35 01 25                            -- Terminal type 25
   9F 83 05 02 64 65                      -- Terminal language ‘de’
   9F 41 01 33                            -- Transaction sequence counter of the last transac-
                                             tion was 33
   9F 1A 02 07 56                         -- Terminal is installed in Switzerland

   9F 83 29 01 05                         -- Booking period 5
   9F 83 2A 01 08                         -- Activation sequence counter 8
   9F 41 02 01 05                         -- Trx sequence counter 105
   9F 16 0F 53 45 4C 45 43 54 41
   30 30 30 30 30 31 32 33                -- Merchant identification

   9F 4E 1F 56 6F 67 65 6C 73 61 6E
   67 73 74 72 2E 31 35 0A 38 33 30
   37 20 45 66 66 72 65 74 69 6B 6F 6E    -- Merchant name and location

   9F 83 01 03 00 88 20                   -- Purchase, credit and reversal transactions possible

   BF 83 0C 08
      02 02 07 56
      02 02 09 78                         -- Currencies CHF + EUR available

   BF 83 0D 13
      9F 83 09 04 56 49 53 41
      9F 83 09 07 4D 61 65 73 74 72 6F    -- Brands ‘VISA’ and ‘Maestro’ available
```

Example (JSON):


```
{
    "ActivationResponse": {
        "TrmID": "30143007",
        "TrmType": 22,
        "TrmLng": "de",
        "TrxSeqCnt": 11314,
        "TrmCntryC": 756,
        "MctID": "PAYTEC000000868",
        "MctNameLoc": "Any Shopping
Any Street 10a
1000 Anywhere",
        "TrxFunctions": 34848,
        "PeSeqCnt": 4,
        "ActSeqCnt": 11,
        "Currencies": [
            756,
            978
        ],
        "Brands": [
            "DEPOSITO CARD",
            "DEPOSITOKARTE",
            "MAESTRO",
            "MAESTRO-CH",
            "MASTERCARD",
            "POSTCARD",
            "POSTFINANCE CARD",
            "POSTFINANCE PAY",
            "VISA",
            "VISA ELECTRON"
        ]
    }
}
```

4.3.7 Deactivation request

This message is sent by the client to close the current user shift and disable the transaction functionality of the EFT terminal. The terminal may perform maintenance tasks like reconfiguration with the Service Center in this state.

Data format: Empty

Example (JSON):

```
{
    "DeactivationRequest": {}
}
```

4.3.8 Deactivation response

This message is sent by the EFT terminal to acknowledge a deactivation request message.

Data format: Empty

Example (JSON):

```
{
    "DeactivationResponse": {}
}
```

4.3.9 Transaction request

This message is sent by the client to initiate a new transaction.

Data format:

Tag	Name		Format	Description	Presence
'31'	TransactionRequest		c		m
	'9F 83 01'	TrxFunction	i	Transaction function	m
	'9F 83 53'	TrxReqFlags	i	Transaction Request Flags	o
	'5F 2A'	TrxCurrC	n (SIZE(1..2))	Transaction Currency Code	o (mandatory for all transaction functions except reversal, Balance and Client ID request)
	'9F 02'	AmtAuth	n (SIZE(1..6))	Amount, Authorized	o (mandatory for all transaction functions except reversal, Balance Inquiry and Client ID request)
	'9F 03'	AmtOther	n (SIZE(1..6))	Amount, Other. Used for cashback amount.	o
	'89'	AuthC	s (SIZE(1..6))	Authorization Code. Used for referring previously authorized transactions (e.g. by phone).	o
	'9F 83 02'	TrxRefNum	s (SIZE(1..11))	Transaction Reference Number. Used by some acquirer interfaces to refer a reservation transaction.	o

Tag	Name		Format	Description	Presence
	‘9F 02’	AcqID	n (SIZE(1..6))	Acquirer Identification.	o (mandatory for cancel Reservation and reservation follow up transactions with ep2 version >= 7.0.0)
	‘9F 39’	POSEntryMode	n (SIZE(1..2))	POS Entry Mode. This field is set when the card data originates from the client: ‘01’: Manual entry ‘90’: Magnetic srtipe	o
	‘5A’	AppPAN	cn (SIZE(4..10))	Application Primary Account Number. Set when POSEntryMode is ‘01’.	o
	‘5F 24’	AppExpDate	n (SIZE(3))	Application Expiry Date. Format is YYMMDD, set when POSEntryMode is ‘01’.	o
	‘9F 6B’	Track2Data	b (SIZE(9..20)), as defined for Track 2 Equivalent Data in EMV/Book 3/Annex A1	Magstripe ISO track 2 data. Set when POSEntryMode is ‘90’.	o
	‘9F 83 00’	Track3Data	b (SIZE(1..54))	Track 3 Data. May be set when POSEntryMode is ‘90’.	o
	‘9F 83 03’	CVC2	s (SIZE(3..4))	Card Verification Code 2. May be set when POSEntryMode is ‘01’.	o
	‘9F 83 5C’	DCCOrigDate	S(SIZE(8))	DCC Original Date YYYYMMDD for credit transactions to refund original transaction with DCC	o

Tag	Name			Format	Description	Presence
	'BF 83 55'	SetActiveSurrogate-PAN		c	Set of surrogate PANs currently involved in a fuelling process (petrol). If any of these cards inserted (or tapped), the transaction is immediately aborted with result TRX_RESULT_ABORTED_MATCHED_SURROGATE_PAN to let the fuel station print a receipt	o
		'9F 83 54'	SurrogatePAN	b (SIZE(32))	A terminal specific surrogate of the Primary Account Number (PAN) of the payment card	o
	‘9F 84 18’	PartialApprovalCap		bool	Partial Approval Capability; tells the terminal whether partial approvals shall be supported for this transaction. If this field is not set, it is assumed to be false.	o
	‘9F 84 08’	MultiCurrencyFlag		bool	Swiss postal banking: Multi Currency Flag. If not set, a terminal which is set up for ep2 postal banking sends a default value of 0.	o
	‘BF 84 0D’	RecPostalBanking		c	Swiss postal banking product record; currently used for deposit transactions.	o
		‘9F 84 0E’	Line1	s (SIZE(0..35))	Message line 1 for deposit transactions	o
		‘9F 84 0F’	Line2	s (SIZE(0..35))	Message line 2 for deposit transactions	o
		‘9F 84 10’	Line3	s (SIZE(0..35))	Message line 3 for deposit transactions	o
	‘BF 84 1F’	RecOrderRef		c	Record Order Reference, containing the ep2 Order ID	o
		‘9F 83 72’	OrderID	s (SIZE(0..48))	Order Identifier, used by the merchant to reference a transaction. Will be submitted to the acquirer and POS management systems.	O

Tag	Name		Format	Description	Presence
	‘9F 88 0A’	TextOnUnknownNFC	s	Text to be shown to the cardholder if an unknown NFC was detected that could be processed by the POS	o

Example (binary format):

```
31 12                                  -- Transaction request, length 18 bytes
   9F 83 01 03 00 80 00                -- Transaction function ‘Purchase’
   5F 2A 02 07 56                      -- Transaction Currency Code 756 (CHF)
   9F 02 03 01 05 65                   -- Amount Authorized: CHF 105.65
```

Example (JSON):

```
{
    "TransactionRequest": {
        "TrxFunction": 32768,
        "TrxReqFlags": 0,
        "TrxCurrC": 756,
        "AmtAuth": 175
    }
}
```

4.3.10 Transaction response

This message is sent by the EFT terminal in return of a transaction request message.

Data format:

Tag	Name		Format	Description	Presence
'31'	TransactionResponse		c		m
	'9F 83 04'	TrxResult	n (SIZE(1))	Transaction result	m
	'9F 1C'	TrmID	s (SIZE(8))	Terminal Identification	m
	'9F 83 09'	Brand	s (SIZE(1..19))	Card brand	m(approved)
	'5F 57'	AccountType	n (SIZE(1))	Account Type	o
	'9F 01'	AcqID	n (SIZE(1..6))	Acquirer Identification	m(approved)
	'9F 06'	AID	b (SIZE(6..17))	Application Identification	m(approved)
	'4F'	AIDICC	b (SIZE(6..17))	Application Identification, as read from the IC card. May differ in case of partial selection (see EMVCo specs). Sent in case of an ICC transaction that is not aborted before completion.	o
	'42'	IIN	n (SIZE(6))	Issuer Identification Number (first 6 digits of the Primary Account Number)	if card has been read and software version >= 25.01.03
	'5A'	AppPAN	cn SIZE(4..10))	Application Primary Account Number	m(referral)
	'9F 83 25'	AppPANPrtCardholder	s (SIZE(1..19))	Application Primary Account Number, to be printed on cardholder receipt. Parts of the PAN may be concealed according to the acquirer settings.	m(approved)
	'9F 83 26'	AppPANPrtAttendant	s (SIZE(1..19))	Application Primary Account Number, to be printed on attendant receipt. Parts of the PAN may be concealed according to the acquirer settings.	m(approved)
	'9F 02'	AmtAuth	n (SIZE(1..6))	Amount, Authorized	m(approved)
	'9F 03'	AmtOther	n (SIZE(1..6))	Amount, Other. Used for cashback amount.	o
	'C1'	TrxAmt	n (SIZE(1..6))	Transaction Amount	o
	'5F 24'	AppExpDate	n (SIZE(3))	Application Expiration Date	o
	'89'	AuthC	s (SIZE(1..6))	Authorization Code	o
	'8A'	ARC	s (SIZE(2))	Authorization Response Code	m(approved)

Tag	Name		Format	Description	Presence
	‘9F 83 12’	At-tendantText	s	Display text for attendant	o
	‘9F 83 11'	Card-holder-Text	s	Displaytext for cardholder	o
	‘5F 28’	IssCnt-ryC	n (SIZE(1..2))	Issuer Country Code	o
	‘9F 39’	POSEnt-ryMode	n (SIZE(1..2))	POS Entry Mode: ‘00’: Unspecified ‘01’: Manual ‘90’: Magnetic stripe ‘05’: Integrated Circuit Card ‘91’, ‘92’, ‘93’: Fallback to magnetic stripe ‘97’: RFID chip profile ‘98’: RFID magnetic stripe profile ‘1001’: Merchant-presented QR Code	m(approved)
	‘5F 2A’	TrxCurrC	n (SIZE(1..2))	Transaction Currency Code	m(approved)
	‘9C’	TrxType	n (SIZE(1))	Transaction Type	m(approved)
	‘9F 41’	TrxSe-qCnt	n (1..99999999)	Transaction Sequence Counter	m(approved)
	‘9F 83 02’	TrxRef-Num	s (SIZE(1..11))	Transaction Reference Number. Returned by some acquirers in case of an approved online transaction	o
	‘9F 84 00’	AppPA-NEnc	s (SIZE(32))	Statically enciphered PAN. Needed for ep2 receipt.	m(approved and ep2)
	‘9F 84 17’	Stat-KeyPAN-RctInd	b (SIZE(1))	EP2 Static Key PAN Receipt Index, needed for receipt from ep2 version 6.0.0. With ep2 >= 7.0.0 this field contains <ep2:KeyPANRctInd>	m(approved and ep2)
	‘9F 84 1A’	KeyPAN-RctDO-LInd	b (SIZE(1))	EP2 Key PAN Receipt DOL Index, for transaction receipt with ep2 version >= 7.0.0	m(approved and ep2 >= 7.0)
	‘9F 84 02’	AuthResl t	n (SIZE(1..3))	ep2 field <Authorisation Result>	m(ep2)
	‘9F 83 0A’	VoicePh-one	s	Phone number of the acquirer bank. Set in case of a referral.	o
	'9F 83 54'	Surro-gatePAN	b (SIZE(32))	A terminal specific surrogate of the Primary Account Number (PAN) of the payment card	if card has been read
	‘9F 84 03’	AmtAvail	i (SIZE(1..8))	Amount, Available	o
	‘9F 84 04’	AmtA-vailCurrC	n (SIZE(1..2))	Amount, Available Currency Code	o
	‘9F 84 05’	IDFlag	bool	Swiss postal banking: ID Flag	o
	‘9F 84 06’	ClientID	s (SIZE(1..9))	Swiss postal banking: Client Identifier	o
	‘9F 84 07’	Ac-countNo	s (SIZE(1..9))	Swiss postal banking: Account Number	o

Tag	Name		Format	Description	Presence
	‘9F 21’	TrxTime	n(SIZE(3))	Transaction time	m(not aborted)
	‘9A’	TrxDate	n(SIZE(3))	Transaction date	m(not aborted)
	‘9F 84 12’	Trx-SeqCntOri	n(SIZE(1..4))	Transaction Sequence Counter of the original transaction.	if reversal
	‘9F 83 30’	Display-Name	s(SIZE(1..31))	Application Label or Application Preferred Name according to code page of terminal. Note: The length in bytes may exceed the maximum number of characters (16) because of the UTF-8 encoding.	o
	‘9F 83 31’	ESignature	b(SIZE(1..3072))	Binary data, PNG file content of an electronic signature	o
	‘9F 34’	CVMResults	b(SIZE(3))	Cardholder verification results	o
	‘9F 83 33’	TrxResultExtended	n(SIZE(1..4))	Extended transaction result	m
	‘9F 83 32’	TipAmt	n(SIZE(1..4))	Tip amount	if tip was entered
	‘9F 84 14’	CardNo	s(SIZE(1..19))	Card number of from the issuing system. Can be truncated	m(postal)
	‘9F 84 15’	CardID	s(SIZE(1..17))	Card-ID from the issuing system	m(postal)
	‘9F 84 19’	AmtRemaining	n(SIZE(1..6))	Remaining amount in case of a partial approval	o
	‘9F 84 25’	PreAuthExpDate	n (SIZE(3))	Pre-Authorization Expiry Date. Can be present in case of Reservation, Reservation Adjustment or Confirm Phone Based Reservation transactions. Format is YYMMDD.	o

Example (binary format):

```
31 81 B9                                    -- Transaction response, length 180 bytes
   9F 83 04 01 00                           -- Transaction approved
   9F 1C 08 33 30 31 34 30 30 31 38         --Terminal Identification: '30140018'
   9F 83 09 04 86 73 83 65                  -- Brand: VISA
   5A 08 40 00 12 34 56 78 90 10'           --AppPAN
   9F 01 01 02'                             --Acquirer Identification: 2
   9F 06 07 A0 00 00 00 03 10 10            --AID: VISA EMV
   9F 02 03 01 05 65                        --Amount, Authorized: 105.65
   5F 24 03 49 12 31                        --Application expires 2049/12/31
   89 04 31 32 33 34                        --Authorization Code '1234'
   8A 02 30 30                              --Authorization Response Code '00'
   9F 83 11 0B 0A 42 75 63 68 75 6e 67 20 4f 4b   --Cardholder display: 'Buchung OK'
   9F 83 12 0B 50 75 72 63 68 61 73 65 20 4f 4b   --Attendant display: 'Purchase OK'
   5F 28 02 07 56                           --Issuer Country Code: 756
   9F 39 01 05                              --ICC transaction
   5F 2A 02 07 56                           --Transaction currency: CHF
   9C 01 00                                 --Transaction Type: Goods & Services
   9F 41 02 01 05                           --Transaction Sequence Counter 105
   9F 84 11 01 03                           --Activation Sequence Counter 3
   9F 84 17 01 01                           --Key PAN Receipt Index 01
   9F 84 1A 01 01                           --Key PAN Receipt DOL Index 01
   9F 84 00 20
     31 32 33 34 35 36 37 38
     39 30 41 42 43 44 45 46
     31 32 33 34 35 36 37 38
     39 30 41 42 43 44 45 46                --Statically enciphered PAN:
                                            '1234567890ABCDEF1234567890ABCDEF'

   9F 83 30 0C
     56 49 53 41 20 43 72 C3 AB 64 69 74    --Display Name 'VISA Crëdit'
```

Example (JSON):


```
{
    "TransactionResponse": {
        "TrxResult": 0,
        "Brand": "Maestro",
        "VoicePhone": "0582345200",
        "AccountType": 0,
        "AcqID": 1,
        "AID": "oAAAAAQwYA==",
        "AIDICC": "oAAAAAQwYA==",
        "AmtAuth": 175,
        "AppExpDate": "191231",
        "AuthC": "006457",
        "ARC": "00",
        "CVMResults": "HwMC",
        "POSEntryMode": 97,
        "TrmID": "30143007",
        "TrxAmt": 175,
        "TrxCurrC": 756,
        "TrxType": 0,
        "TrxSeqCnt": 11321,
        "TrxDate": "181031",
        "TrxTime": "112233",
        "AuthReslt": 0,
        "TrxRefNum": "7338750",
        "AppPANEnc": "ADC5776DAD217443E5CAF652DEEE5693",
        "StatKeyPANRctInd": "AQ==",
        "KeyPANRctDOLInd": "AQ==",
        "DisplayName": "Maestro",
        "TrxResultExtended": 100,
        "AppPANPrtCardholder": "XXXXXXXXXXXXXXX6144",
        "AppPANPrtAttendant": "XXXXXXXXXXXXXXX6144",
        "CardholderText": "Transaction OK",
        "AttendantText": "Transaction OK",
        "TipAmt": 0,
        "AmtRemaining": 0
    }
}
```

4.3.11 Transaction confirmation request

This message is sent by the client to confirm or cancel an approved transaction response.

Data format:

Tag	Name		Format	Description	Presence
‘31’	TransactionConfirmationRequest		c		m
	‘01’	Confirm	bool	Confirm or cancel the transaction	m
	‘C1’	TrxAmt	n (SIZE(1..6))	Transaction Amount: This value may be set to a lower value than Amount, Authorized, e.g. in case of a fuel station or vending machine.	o
	‘9F 41’	TrxSeqCnt	n (SIZE(1..4))	Transaction Sequence Counter. Used to reference a former Authorisation Purchase transaction	m(auth purchase)

Example (binary format):

‘31 08’ --Transaction confirmation request, length 8 bytes
’01 01 01’ --Transaction is confirmed
‘C1 03 01 00 50’ --Transaction Amount 100.50

Example (JSON):

```
{
    "TransactionConfirmationRequest": {
        "Confirm": 1
    }
}
```

4.3.12 Transaction confirmation response

This message is sent by the EFT terminal in return of a transaction confirmation request message.

Data format:

Tag	Name		Format	Description	Presence
‘31’	TransactionConfirmationResponse		c		m
	‘9F 81 09’	MsgType	n (SIZE(1..2))	EMVCo Field 'Message Type'	o (only if rollback)
	‘9F 84 02’	AuthReslt	n (SIZE(1..3))	ep2 field <Author-isation Result>	o (only if rollback)
	‘9F 83 11'	CardholderText	s	Displaytext for cardholder	o (only if rollback)
	‘9F 83 12’	AttendantText	s	Display text for at-tendant	o (only if rollback)

Example (JSON):

```
{
    "TransactionConfirmationResponse": {
        "MsgType": 11,
        "AuthReslt": 100,
        "CardholderText": "Aborted",
        "AttendantText": "Aborted "
    }
}
```

4.3.13 Confirmation time extension request

This message is sent by the client to prolong the time the terminals waits for a transaction confirmation request.

Data format:

Tag	Name		Format	Description	Presence
‘31’	ConfirmationTimeExtensionRequest		c		m
	‘02’	Seconds	i (SIZE(1))	Number of seconds to wait for transaction confirmation request, from now. Possible Values: between 0 and 30	m
	‘9F 83 11’	CardholderText	s (SIZE(0..255)	Text to be shown to the cardholder while waiting for transaction confirmation. Ideally this should be supplied in cardholder language as received in the status response message.	o

Example (binary):

‘31 11’ --Confirmation time extension request, length 3 bytes
’02 01 0F --Wait 15 seconds on confirmation, before aborting
’9F 83 11 0A 54 68 61 6E 6B 20 79 6F 75 21’ --‘Thank you!’

Example (JSON):

```
{
    "ConfirmationTimeExtensionRequest": {
        "Seconds": 15,
        "CardholderText": "Thank you!"
    }
}
```

4.3.14 Confirmation time extension response

This message is sent by the EFT terminal in return of a confirmation time extension request message.

Data format: Empty

Example (JSON):

```
{
    "ConfirmationTimeExtensionResponse": {}
}
```

4.3.15 Abort transaction request

This message is sent to abort a transaction in progress. The message must be sent before the transaction response.

Data format:

Tag	Name		Format	Description	Presence
‘31’	AbortTransactionRequest		c		m
	‘9F 83 4E’	TrxAbortFlags	i	Abort Transaction Flags	o

Example (binary format):
‘31 05’ --Abort Transaction request, length 5 bytes
’9F 83 4E 01 01’ --Abort should be silent

Example (JSON):

```
{
    "AbortTransactionRequest": {
        "TrxAbortFlags": 1
    }
}
```

4.3.16 Abort transaction response

This message is sent by the EFT terminal in return of an abort transaction request message.

Data format: Empty

Example (JSON):

```
{
    "AbortTransactionResponse": {}
}
```

4.3.17 Batch capture request

This message is sent to initiate a submission of all the unsubmitted transaction to the acquirer resp. the PMS.

Data format:

Tag	Name		Format	Description	Presence
‘31’	BatchCaptureRequest		c		m
	‘9F 83 5A’	BatchCap-tureFlags	i	Batch Capture Flags	o

Example (binary format):
‘31 05’ --Batch capture request, length 5 bytes
’9F 83 5A 01 02’ --Don’t send the data to the acquirers

Example (JSON):

```
{
    "BatchCaptureRequest": ": {
        "BatchCaptureFlags": 2
    }
}
```

4.3.18 Batch capture response

This message is sent by the EFT terminal in return of a batch data capture request message.

Data format: Empty

Example (JSON):

```
{
    "BatchCaptureResponse": {}
}
```

4.3.19 Balance request

This message initiates a final balance or so called end of day procedure on the terminal. All pending transactions are submitted and the daily counters are reset.

Data format: Empty

Example (JSON):

```
{
    "BalanceRequest": {}
}
```

4.3.20 Balance response

This message is sent by the EFT terminal in return of a balance request message.

Data format: Empty

Example (JSON):

```
{
    "BalanceResponse": {}
}
```

4.3.21 Configuration request

This message advises the terminal to reload its configuration data from the TMS.

Data format: Empty

Example (JSON):

```
{
    "ConfigurationRequest": {}
}
```

4.3.22 Configuration response

This message is sent by the EFT terminal in return of a configuration request message.

Data format: Empty

Example (JSON):

```
{
    "ConfigurationResponse": {}
}
```

4.3.23 Initialization request

This message advises the terminal to reload its initialization data from the acquirers.

Data format: Empty

Example (JSON):

```
{
    "InitializationRequest": {}
}
```

4.3.24 Initialization response

This message is sent by the EFT terminal in return of an initialization request message.

Data format: Empty

Example (JSON):

```
{
    "InitializationResponse": {}
}
```

4.3.25 Receipt request

This message advises the terminal to prepare and send a receipt.

Data format:

Tag	Name		Format	Description	Presence
‘31’	ReceiptRequest		c		m
	‘9F 83 35’	ReceiptType	n	Receipt Type	m
	‘9F 83 36’	ReceiptIDNumeric	n	Numeric receipt identifier; in case of a transaction receipt or transaction receipt copy, the Transaction Sequence Counter.	m(auth purchase) For other receipt & transaction types: If not present, the last available receipt is returned.

Example (JSON):

```
{
    "ReceiptRequest": {
        "ReceiptType": 1,
        "ReceiptIDNumeric": 11315
    }
}
```

4.3.26 Receipt response

This message is sent in response to a receipt request. If the receipt is not available, ReceiptType is set to 0 and ReceiptText is empty.

Data format:

Tag	Name		Format	Description	Presence
'31'	ReceiptResponse		c		m
	'9F 83 35'	ReceiptType	n	Receipt Type	m
	'9F 83 38'	ReceiptFlags	i	Receipt Flags	m
	'9F 83 39'	ReceiptText	s	Receipt text, encoded as UTF-8 and formatted using <Line Feed (0x0A)> as end of line sequence. In case of an electronic signature, an XML tag <ESignature /> on a separate line is used as placeholder.	m

Example (JSON):

Note: The line breaks within the ReceiptText are escaped as "\n".

```
{
    "ReceiptResponse": {
        "ReceiptType": 1,
        "ReceiptFlags": 2,
        "ReceiptText": "              Any Shopping
             Any Street 10a
             1000 Anywhere

                Purchase
            Visa contactless

XXXXXXXXXXXX8632

31.10.2018                      11:06:02
Trm-Id:                         30143007
Act-Id:                                0
AID:                      A0000000031010
Trx. Seq-Cnt:                     11319
Trx. Ref-No:                    7338746
Auth. Code:                      011762
1FD6E92331734CB14DD96B255A9920DF      01

Total-EFT CHF:                      1.75

"
    }
}
```

4.3.27 Report request

This message advises the terminal to send defined report from the EFT database.

Data format:

Tag	Name		Format	Description	Presence
'31'	ReportRequest		c		m
	'9F 83 3A'	ReportType	n	Report Type	m

Example (JSON):

```
{
    "ReportRequest": {
        "ReportType": 1
    }
}
```

4.3.28 Report response

This message advises the terminal to send defined report from the EFT database.

Data format:

Tag	Name		Format	Description	Presence
'31'	ReportResponse		c		m
	'9F 83 3A'	ReportType	n	Report Type	m
	'9F 83 3B'	ReportFlags	n	Report Flags	m
	'BF 83 3C'	FinancialReport	Set of FinancialReportEntry	Financial report	In case of a summary report
	'BF 88 09'	TrxListReport	Set of TrxListReportEntry	Transaction list report	In case of a transaction list report

FinancialReportEntry:

Tag	Name		Format	Description	Presence
'BF 83 3D'	FinanicalReportEntry		c		m
	'9F 83 09'	Brand	s	Card brand	m
	'9F 06'	AID	b	Application Identifier	m
	'9F 83 3E'	FinancialCounterType	n	Financial Counter Type	m
	'5F 2A'	TrxCurrC	n	Transaction Currency Code	m
	'9F 83 3F'	TrxCount	n	Number of transactions counted	m
	'9F 83 40'	TotalAmount	i (SIZE(1..8))	Total amount of the transaction. For Credit and reversal debit records, this is a negative amount.	m

	‘9F 83 41’	DCCForeignCurrC	n	Cardholder currency when Dynamic Currency Conversion was made.	o
	‘9F 83 42’	DCCForeignAmount	i (SIZE(1..8))	Total amount in cardholder currency. For Credit and reversal debit records, this is a negative amount.	o
	‘9F 83 7D’	TipAmount	i (SIZE(1..8))	Tip amount in the merchant’s billing currency. Only present if not 0.	o

TrxListReportEntry:

Tag	Name		Format	Description	Presence
‘BF 88 0A’	TrxListReportEntry		c		m
	‘9F 41’	TrxSeqCnt	n (1..99999999)	Transaction Sequence Counter	m
	‘9A’	TrxDate	n(SIZE(3))	Transaction date	m
	‘9F 21’	TrxTime	n(SIZE(3))	Transaction time	m
	‘9F 83 3E’	FinancialCounterType	n	Financial Counter Type	m
	‘9F 83 01’	TrxFunction	i	Transaction function	m
	‘9F 84 02’	AuthReslt	n (SIZE(1..3))	ep2 field <Authorisation Result>	m
	‘5F 2A’	TrxCurrC	n	Transaction Currency Code	m
	‘9F 83 3F’	TrxCount	n	Number of transactions counted	m
	‘9F 02’	AmtAuth	n (SIZE(1..6))	Amount, Authorized	m
	‘9F 83 26’	AppPANPrtAttendant	s (SIZE(1..19))	Application Primary Account Number, to be printed on attendant receipt. Parts of the PAN may be concealed according to the acquirer settings.	m

Example (binary format):

```
'31 87‘                                                    -- Report response, length 135 bytes
   ‘9F 83 3A 01 01’                                        -- Report type ‘Sum of shift’
   ‘9F 83 3B 01 00’                                        -- Report flags ‘00’
   ‘BF 83 3C 9F’                                           -- Financial report
      ‘BF 83 3D 33’                                        -- Financial report entry 1
         ‘9F 83 09 0A 4D 61 73 74 65 72 43 61 72 64’       -- Brand ‘MasterCard’
         ‘9F 06 07 A0 00 00 01 57 00 20’                   -- AID MasterCard magstripe (ep2)
         ‘9F 83 3E 01 00’                                  -- Counter type ‘debit’
```

```
    ‘5F 2A 02 07 56’                          -- Transaction currency CHF
    ‘9F 83 3F 01 01’                          -- 1 transaction counted
    ‘9F 83 40 02 17 70’                       -- Total amount (CHF) 60.-
    ‘9F 83 7D 02 01 5E’                       -- Tip amount (CHF) 3.50
‘BF 83 3D 33’                                 -- Financial report entry 2
    ‘9F 83 09 04 56 49 53 41’                 -- Brand ‘VISA’
    ‘9F 06 07 A0 00 00 01 57 00 30’           -- AID VISA magstripe (ep2)
    ‘9F 83 3E 01 00’                          -- Counter type ‘debit’
    ‘5F 2A 02 07 56’                          -- Transaction currency CHF
    ‘9F 83 3F 01 01’                          -- 1 transaction counted
    ‘9F 83 40 02 1C 20’                       -- Total amount (CHF) 72.-
    ‘9F 83 41 02 08 40’                       -- Cardholder currency (DCC): USD
    ‘9F 83 42 02 1A DB’                       -- Total amount (USD): 68.75
‘BF 83 3D 33‘                                 -- Financial report entry 3
    '9F 83 09 04 56 49 53 41‘                 -- Brand ‘VISA’
    '9F 06 07 A0 00 00 01 57 00 30‘           -- AID VISA magstripe (ep2)
    '9F 83 3E 01 02‘                          -- Counter type ‘reversal debit’
    '5F 2A 02 07 56‘                          -- Transaction currency CHF
    '9F 83 3F 01 01‘                          -- 1 transaction counted
    '9F 83 40 02 E3 E0‘                       -- Total amount (CHF) -72.-
    '9F 83 41 02 08 40‘                       -- Cardholder currency (DCC): USD
    '9F 83 42 02 E5 25‘                       -- Total amount (USD): -68.75
```

Example (JSON):


```
{
    "ReportResponse": {
        "ReportType": 1,
        "ReportFlags": 0,
        "FinancialReport": [
            {
                "Brand": "Maestro",
                "AID": "oAAAAAQwYA==",
                "FinancialCounterType": 0,
                "TrxCurrC": 756,
                "TrxCount": 1,
                "TotalAmount": 175
            },
            {
                "Brand": "Maestro",
                "AID": "oAAAAAQwYA==",
                "FinancialCounterType": 2,
                "TrxCurrC": 756,
                "TrxCount": 1,
                "TotalAmount": -175
            },
            {
                "Brand": "MasterCard",
                "AID": "oAAAAAQQEA==",
                "FinancialCounterType": 0,
                "TrxCurrC": 756,
                "TrxCount": 1,
                "TotalAmount": 175
            },
            {
                "Brand": "Postcard",
                "AID": "oAAAAREBAQ==",
                "FinancialCounterType": 0,
                "TrxCurrC": 756,
                "TrxCount": 1,
                "TotalAmount": 3000,
                "TipAmount": 150
            },
            {
                "Brand": "Visa payWave",
                "AID": "oAAAAAMQEA==",
                "FinancialCounterType": 0,
                "TrxCurrC": 756,
                "TrxCount": 2,
                "TotalAmount": 350,
                "DCCForeignCurrency": 840,
                "DCCForeignAmount": 362
            }
        ]
    }
}
```

4.3.29 Transaction data change request

This message will be sent from the terminal to ECR requesting some changes on the running transaction. This can happen in case the terminal detects a card which can eventually trigger some changes in running transaction (e.g. discount on original authorization amount).

Data format:

Tag	Name	Format	Description	Presence
‘31’	Transaction data change request	c		m
	‘BF 83 49’	SetTrxDataChangeReason		m

Example (binary format):

‘31 12’ -- Trx data change request, length 12 bytes
‘BF 83 49 0E’ -- set of active change reasons
‘BF 83 4A 0A’ -- change reason
‘9F 83 4B 01 01’ -- myOne discount detection
‘9F 83 4C 01 19’ -- only for Rabattcode ‘19’

Example (JSON):

```
{
    "TransactionDataChangeRequest": {
        "SetTrxDataChangeReason": [
            {
                "TrxDataChangeReasonCode": 1,
                "MyOneDiscountCode": 19
            }
        ]
    }
}
```

4.3.30 Transaction data change response

This message will be sent from ECR back to terminal with possible newly changed data. The flags correspond with the transaction request flags. If nothing has changed this message is empty. Currently only AmtAuth is implemented for change reason 1.

Data format:

Tag	Name		Format	Description	Presence
‘31’	TransactionDataChangeResponse		c		m
	‘9F 02’	AmtAuth	n (SIZE(1..6))	Amount, Authorized	o

Example (JSON):

```
{
    "TransactionDataChangeResponse": {
        "AmtAuth": 1000
    }
}
```

4.3.31 Lock EFT operations request

This message advises the terminal to lock the EFT operations and switch to mode where it accepts the special custom dialog and screen requests.

Data format: Empty

Example (JSON):

```
{
    "LockEFTOperationsRequest": {}
}
```

4.3.32 Lock EFT operations response

This message acknowledges successful switch of the device to mode where no EFT operations are possible.

Data format: Empty

Example (JSON):

```
{
    "LockEFTOperationsResponse": {}
}
```

4.3.33 Unlock EFT operations request

This message advises the terminal to unlock the EFT operations and switch to normal operation mode where it does not accept the special custom dialog and screen requests but accepts EFT requests.

Data format: Empty

Example (JSON):

```
{
    "UnlockEFTOperationsRequest": {}
}
```

4.3.34 Unlock EFT operations response

This message acknowledges successful switch of the device to mode where EFT operations are possible again.

Data format: Empty

Example (JSON):

```
{
    "UnlockEFTOperationsResponse": {}
}
```

4.3.35 Set terminal language request

This message advises the terminal to change its default language to the new value.
The terminal language is the language used for receipts, attendant user interfaces and for the cardholder user interfaces as long as no language code from a card is in use.
The supplied value is kept until the terminal reboots or reconfigures itself with the TMS which changes back the default language to the value configured in TMS.

Data format:

Tag	Name		Format	Description	Presence
‘31’	SetTerminalLanguageRequest		c		m
	‘9F 83 05’	TrmLng	s (SIZE(2))	Terminal language as an ISO 639-1 2 letter code, e.g. ‘en’, ‘de’	m

Example (JSON):

```
{
    "SetTerminalLanguageRequest": {
        "TrmLng": "en"
    }
}
```

4.3.36 Set terminal language response

This message acknowledges a successful set terminal language request.

Data format: Empty

Example (JSON):

```
{
    "SetTerminalLanguageResponse": {}
}
```

4.3.37 Dialog request

This message requests the terminal to perform a dialog with the cardholder. Depending on the dialog resource, the message can only be used if the terminal is in locked status.

Data format:

Tag	Name		Format	Description	Presence
'31'	DialogRequest		c		m
	'9F 83 20'	DialogResource	n (SIZE(1..4))	The text resource to be displayed	m
	'9F 83 16'	Timeout	n (SIZE(1..4))	Timeout of the dialog, in milliseconds. If not given, the timeout is infinite.	o
	'9F 83 19'	InputType	n (SIZE(1..4))	One of Input Type	o
	'BF 83 21'	DialogResourceArgs	c	The variable argument list defined for the text resource as a sequence of UTF8Strings	o
	'9F 83 1C'	ReferencePassword	s (SIZE(1..31))	Value to check password against when Input Type is Password	o
	'9F 83 1F'	Buttons	i	Combination of buttons to display	o
	'9F 83 56'	DialogFlags	i	Combination of flags to influence dialog behaviour	o

Example (binary format):

'31 24' -- Dialog request, length 36 bytes
'9F 83 20 02 10 10' -- Free text
'9F 83 16 03 03 00 00' -- time out after 30 seconds
'9F 83 1F 01 03' -- 'OK' & 'Cancel' buttons
'BF 83 21 0E' -- Dialog resource arguments
'0C 0C 48 65 6C 6C 6F 20 57 6F 72 6C 64 21' -- 'Hello World!'

Example (JSON):

```
{
    "DialogRequest": {
        "DialogResource": 1010,
        "Timeout": 30000,
        "UIButtons": 3,
        "DialogResourceArgs": [
            "Hello World!"
        ]
    }
}
```

4.3.38 Dialog response

This message is sent to the ECR when a user dialog has finished.

Data format:

Tag	Name		Format	Description	Presence
‘31’	DialogResponse		c		m
	‘9F 83 24’	DialogResult	n (SIZE(1..4))	The result of the dialog	m
	'0C'	UTF8String	s	User input	o
	'01'	Boolean	bool	Result of pass-word verification	o

Example (JSON):

```
{
    "DialogResponse": {
        "DialogResult": 1,
        "Boolean": 1
    }
}
```

4.3.39 Cancel UI request

This message is sent by the ECR to prematurely abort a user dialog.

Data format: Empty

Example (JSON):

```
{
    "CancelUIRequest": {}
}
```

4.3.40 Print receipt request

Sent by the ECR to print a receipt on the terminal's thermal printer.

Data format:

Tag	Name		Format	Description	Presence
'31'	Print request		c		m
	'9F 83 38'	ReceiptFlags	i	Receipt Flags	o
	'9F 83 39'	ReceiptText	s	Receipt text, encoded as UTF-8 and formatted using <Line Feed (0x0A)> as end of line sequence.	m

Example (JSON):

```
{
    "PrintReceiptRequest": {
        "ReceiptFlags": 0,
        "ReceiptText": "LINE1\nline2"
    }
}
```

4.3.41 Print receipt response

Acknowledges the printing of a receipt, reporting the current printer status.

Data format:

Tag	Name		Format	Description	Presence
'31'	Print response		c		m
	'9F 83 5B'	PrinterStatus	i	Combination of PrinterStatus	m

Example (JSON):

```
{
    "PrintReceiptResponse": {
        "PrinterStatus": 0
    }
}
```

4.3.42 Display notification

This message is sent from the terminal to the cash register when an external cardholder display is used and the display content changes. The terminal also sends a display notification after the cash register has connected (with UseExternalDisplay = true).

Data format:

Tag	Name		Format	Description	Presence
'31'	DisplayNotification		c		m
	'9F 83 11'	CardholderText	s	Text shown on the cardholder display. Line breaks are indicated with ASCII code '0A' (newline)	m
	'9F 83 2D'	CardholderResource	n (SIZE(1..4))	Numeric resource identifier of the message shown on the cardholder display. This information can be used e.g. for acoustic feedback.	m

Example (binary format):

```
'31 15'                                          -- Display notification, length 21 bytes
  '9F 83 13 0B 50 75 72 63 61 73 65 0A 4F 4B'    -- Cardholder text 'Purchase\nOK'
  '9F 83 2D 02 20 03'                            -- Cardholder resource identifier 2003
```

Example (JSON):

```
{
    "DisplayNotification": {
        "CardholderText": "Welcome\nPresent card",
        "CardholderResource": 993
    }
}
```

4.3.43 RFID status notification

This message is sent from the terminal to the cash register when an external RFID user interface is used and the status of the interface changes. The terminal also sends an RFID status notification after the cash register has connected (with UseExternalRFIDUI = true).

Data format:

Tag	Name		Format	Description	Presence
‘31’	DisplayNotification		c		m
	‘9F 83 2C’	RFIDStatus	n (SIZE(1..4))	Status of the RFID interface.	m

Example (binary format):

‘31 05’ -- RFID status notification, length 5 bytes
’9F 83 2C 01 03’ -- Contactless transaction is in progress

Example (JSON):

```
{
    "RFIDStatusNotification": {
        "RFIDStatus": 3
    }
}
```

4.3.44 Device command request

This message is used to perform various device specific operations.

Data format:

Tag	Name		Format	Description	Presence
'31'	DeviceCommandRequest		c		m
	'9F 83 13'	DeviceCommandCode	n (SIZE(1..4))	Command code	m
	var	var	var	variable elements, depending on the command	o

Example (binary format):

'31 05' -- Device command request, length 5 bytes
'9F 83 13 01 01' -- Command 'Eject card'

Example (JSON):

```
{
    "DeviceCommandRequest": {
        "DeviceCommandCode": 1,
        "CardholderText": "Welcome"
    }
}
```

4.3.45 Device command response

This message is sent by the EFT terminal in return of an initialization request message.

Data format:

Tag	Name		Format	Description	Presence
'31'	DeviceCommandResponse		c		m
	'9F 83 13'	DeviceCommandCode	n (SIZE(1..4))	Command code	m
	var	var	var	Variable elements, depending on the command	o

Example (binary format):

'31 0A' -- Device command response, length 10 bytes
'9F 83 13 01 01' -- Command 'Eject card'

Example (JSON):

```
{
    "DeviceCommandResponse": {
        "DeviceCommandCode": 1
    }
}
```

4.3.46 Error notification

This message is sent when an unexpected condition occurs.

Data format:

Tag	Name		Format	Description	Presence
‘31’	ErrorNotification		c		m
	‘9F 83 0E’	ErrorCode	n	Error codes; can be operating system specific code	m
	‘9F 83 0F’	ErrorSource	s	Source that caused the error, e.g. a source code position	m
	‘9F 83 10’	ErrorDescription	s	Textual description of the error	m

Example (binary format):

```
31 1F                                    --Error notification, length 31 bytes
   9F 83 0E 02 01 05                     --Code 105
   9F 83 0F 09 66 6f 6f 2e 63 28 31 32 29 --Source: ‘foo.c(12)’
   9F 83 10 08 4f 76 65 72 66 6c 6f 77   --Description: ‘Overflow’
```

Example (JSON):

```
{
    "ErrorNotification": {
        "ErrorCode": 105,
        "ErrorSource": "foo.c(12)",
        "ErrorDescription": "Overflow"
    }
}
```

4.3.47 Heartbeat request

When either EFT terminal's client or the EFT terminal disconnect without a clean TCP shutdown, the peer does not take notice without exchanging data. To detect this situation, a heartbeat request message is sent in a configurable interval from the EFT terminal to its client and optionally also vice versa. If the terminal does not receive an answer less than a configurable timeout period after the last heartbeat request, it assumes that the client has disconnected and shuts down the socket.

Data format: Empty

Example (JSON):

```
{
    "HeartbeatRequest": {}
}
```

4.3.48 Heartbeat response

This message answers the heartbeat request message.

Data format: Empty

Example (JSON):

```
{
    "HeartbeatResponse": {}
}
```

5 Data type definitions

5.1 Transaction function

Note: These transaction functions originate from [ep2] where further description about them can be found.

Value	Description
0x04000000	Account verification
0x02000000	Cancel reservation
0x01000000	Load
0x00800000	Activate card
0x00400000	Authorization credit
0x00200000	Authorization deposit
0x00100000	Giro
0x00080000	Combined
0x00040000	Deposit
0x00020000	Balance inquiry
0x00010000	Client ID request
0x00008000	Purchase
0x00004000	Purchase reservation
0x00002000[1)]	Tip
0x00001000	Cash advance
0x00000800	Credit
0x00000400	Purchase phone authorized
0x00000200	Purchase forced acceptance
0x00000100	Purchase phone ordered
0x00000080	Authorization purchase
0x00000040	Purchase mail ordered
0x00000020	Reversal
0x00000010	Reservation
0x00000008	Reservation adjustment
0x00000004	Confirm phone authorized reservation
0x00000001	Purchase with cashback

[1)]: As of ep2 version 5.0.x and higher there is no separate tip transaction type. Tip handling is made in purchase transaction.

5.2 Transaction result

Value	Description
0	Transaction approved
1	Transaction declined
2	Transaction referred: Attendant must call the bank - not applicable for unattended terminals.
3	Transaction aborted

5.3 Extended transaction result

Value	Description
0-899	TRX_RESULT_APPROVED
900-999	TRX_RESULT_REFERRED
1000-9999	TRX_RESULT_DECLINED
10000-49999	TRX_RESULT_ABORTED
10014	TRX_RESULT_ABORTED_MATCHED_SURROGATE_PAN
10017	TRX_RESULT_ABORTED_SCHEDULED_TASK_PENGING Note: This is returned 1 minute after starting an authorization purchase transaction when no card is presented and a scheduled task like e.g. initialization or data submission is due. In this case, the ECR should wait for at least 20 seconds before restarting the transaction. It also should wait for the terminal to finish the task, i.e. reporting a Transaction Status where the Busy-flag is not set.

5.4 Cardholder verification results

Byte	Description
0	CVM Code
1	CVM Condition
2	CVM Result

5.4.1 CVM Code

Bit values of the byte								Description
b8	b7	b6	b5	b4	b3	b2	b1	
0								RFU
	0							Fail cardholder verification if this CVM is unsuccessful
	1							Apply succeeding CV Rule if this CVM is unsuccessful
		0	0	0	0	0	0	Fail CVM processing
		0	0	0	0	0	1	Plaintext PIN verification performed by ICC
		0	0	0	0	1	0	Enciphered PIN verified online
		0	0	0	0	1	1	Plaintext PIN verification performed by ICC and signature (paper)
		0	0	0	1	0	0	Enciphered PIN verification performed by ICC
		0	0	0	1	0	1	Enciphered PIN verification performed by ICC and signature (paper)
		0	1	1	1	1	0	Signature (paper)
		0	1	1	1	1	1	No CVM required
		1	1	1	1	1	1	CVM not applicable

5.4.2 CVM Condition

Value	Description
0	CVM_ALWAYS
1	CVM_IF_UNATTENDED_CASH
2	CVM_IF_NO_CASH
3	CVM_IF_SUPPORTED
4	CVM_IF_MANUAL_CASH
5	CVM_IF_CASHBACK

6	CVM_IF_UNDER_X
7	CVM_IF_OVER_X
8	CVM_IF_UNDER_Y
9	CVM_IF_OVER_Y

5.4.3 CVM Result

Value	Description
0	CVM_RESULT_UNKNOWN
1	CVM_RESULT_FAILED
2	CVM_RESULT_SUCCESS

5.5 Terminal status

Value	Description
0x00000001	Terminal is activated. The terminal accepts card insertion only while activated.
0x00000002	A valid card is in the contact reader.
0x00000004	Terminal is busy with another operation.
0x00000008	Reader slot occupied, e.g. if the cardholder has not yet removed its card.
0x00000010	Terminal is locked for EFT operations (transactions, initializations etc.).
0x00000020	Application selected
0x00000040	Waiting for transaction request
0x00000080	Waiting for application selection
0x00000100	Terminal is performing EP2 online processing
0x00000200	Receipt printer is unavailable
0x00000400	Receipt printer is out of paper
0x00000800	No internet connectivity. This flag is set, if the terminal detects no internet connectivity. IMPORTANT: The receiving of internet connectivity information must be configured via Device Command and consumes additional bandwidth, depending on the frequency to update this information
0x40000000	The WWAN modem is used by the device. This bit is specific to device type 'C4' which shares the WWAN modem with the attached Pay & Display Machine: The bit is set before demand-dialling starts and is cleared when PPP is hung up due to idle timeout.

5.6 RFID status

These status codes are designed to comply with both the recommended and alternative user interfaces proposed by the MasterCard PayPass Terminal Implementation Requirements and the Visa Europe Contactless Terminal Requirements and Implementation Guide.
The status codes are used to trigger the LEDs and acoustic signals; they do not reflect the text messages for the 2-line display which are transmitted within the display notification message.

Value	Description
0	Not working; contactless interface is not available.
1	Idle; contactless interface is available but not activated.
2	Ready; contactless interface is ready for transaction processing.
3	Processing; contactless card reading is in progress.
4	Complete; contactless card reading has been completed successfully.

99	Error; an error occurred during a contactless transaction.

5.7 Device command code

Value	Description
1	Eject card; only meaningful on terminals equipped with a motorized card reader
2	Force eject card; sends “turn on motor” command for period necessary to eject the card to the motorized card reader to eject the card. It does not take in account status of the application and/or sensors.
501	Reboot; reboots the terminal immediately
502	SWUpdate; terminates the payment application and starts the service application afterwards, which can initiate SW Download if necessary according to service center
503	EnterPowerSaveMode; Requests the terminal to go into power save mode.
511	Start remote maintenance session. If a session is configured on the TMS, the terminal will start it immediately instead of waiting for the usual polling interval.
521	Submit technical log data to the TMS
522	Configure internet connectivity supervision
1001	Enable language selection; will enable language selection dialog after insertion of card with different language than supported on terminal.
1002	Disable language selection; the cardholder language will be the currently selected terminal language.
2001	Show idle
2002	Show card insertion
2501	Terminal software version >= 17.x: Configure display text that is shown on terminal when no ECR or vending machine has connected. If this display text is configured, the terminal does not switch anymore to monoblock mode when ECR disconnects. To reset the display text to the default behavior, set the empty string as cardholder text. **[obsolete] from software 17.01.07, this text is configured via TMS**
2601	Terminal software >= 21.01.11: Scan visual symbol like barcode or QR code
2602	Terminal software >= 24.03.04: Stop ongoing scan for visual symbol like barcode or QR code

5.8 Device command variable elements

Command code	**Device command request**	**Device command response**
1	None	None
522	- Boolean (tag ‘01’): Enable (01h) or disable (00h) supervision, default 00h - Seconds (tag ‘02’): Interval to check connectivity.	None
1001	None	None
1002	None	None
2001	None	None

2002	None	None
2501	CardholderText (tag '9F 83 11')	None
2601	- Timeout (tag ‘9F 83 16’) in milliseconds, numeric - Title (tag ‘9F 83 6F’) as string to displayed as scanner UI title - Text (tag ‘9F 83 70’) as string to be displayed as hint text in the scanner UI	ScannedData (tag ‘9F 83 71’) as string

5.9 Resource Identifier

Note: This list is not complete; it may change with every new version of the terminal software. The values below reflect the most used messages.

Value	Message
	Messages shown on the cardholder display
501	Abort
505	Card blocked
507	Card expired
509	Card read error
510	Card reader error
512	Card unknown
513	<Currency> <Amount> Maximum amount
514	<Currency> <Amount> Minimum amount
517	Currency not supported
521	Funds too low
522	Function not supported
526	Please return terminal
527	<Currency> <Amount> Processing
529	Referral declined
531	System error
533	Tender failed
534	Too many withdrawals
541	Closed
542	Withdraw amount limit exceeded
600	Invalid amout Declined
602	System error

	Declined
701	Card not valid Please pay cash
702	Card locked Please pay cash
703	Card full Please contact card issuer
922	Tip not possible
924	Security problem
925	No answer from PIN Pad
927	Missing EP2 security application
932	Printer problem
971	Terminal not bound
972	<Currency> <Amount> Please wait
978	<Currency> <Amount> OK? Signature please
980	SW-download Please wait
981	Awaiting total amount..
982	Waiting on ECR..
985	Available fund <Currency> <Amount>
987	Unsupported ECR interface
988	Welcome Insert card
991	Abort by ECR
992	Checking printer..
	Message defined by EMVCo
2001	<Currency> <Amount>
2002	<Transaction function> <Currency> <Amount> OK?
2003	<Transaction function> OK
2004	Call your bank
2005	Cancel or enter
2006	Card error
2007	Declined
2008	Enter amount:
2009	<Currency> <Amount> PIN+OK: **********
2010	PIN incorrect
2011	Welcome

	Insert card
2012	Card not accepted
2013	PIN OK
2014	<Currency> <Amount> Please wait
2015	Processing error
2016	Remove card
2017	Use chip reader
2018	Use magstripe reader
2019	Try again
	Other messages used by the EMV application kernel
2051	Last PIN try PIN+OK **********
2053	PIN incorrect PIN+OK **********
	Messages used by the contactless EMV application kernel
2501	Present card
2502	Present one card only
2505	<Transaction function> <Application Preferred Name> OK

5.10 **Receipt Type**

Value	Description
1	Transaction receipt
2	Transaction receipt copy (cardholder receipt)
11	Activation receipt
12	Activation failed receipt (cannot be requested)
13	Deactivation receipt
14	Deactivation failed receipt (cannot be requested)
21	Final balance receipt
22	Final balance failed receipt (cannot be requested)
23	Submission receipt (cannot be requested)
24	Submission failed receipt (cannot be requested)
25	Transmission receipt (cannot be requested)
26	Transmission failed receipt (cannot be requested)
27	Period report receipt
28	Shift report receipt
41	Configuration receipt
43	Initialization receipt

5.11 Receipt Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	More data available. The complete receipt data does not fit into a single receipt response message; the receipt request message shall be repeated to receive the remaining receipt data. In case of a print receipt request message, this flag causes the terminal to wait for more data before cutting the receipt.
4	Use double height font (not used in receipt response message)
16	Print inverse text; available with terminal software version >= 20.x
32	Receipt text is the binary representation of a PNG image; available with terminal software version >= 20.x Note: To have the best printing result, use a black on white (monochrome) image, not wider than 384 pixel. Maximum image size is determined by the maximum message size.

5.12 Report Type

Value	Description	Report class
1	Sum of shift	Summary report
2	Sum of accounting period	Summary report
1001	List of open petrol authorizations	Transaction list report

5.13 Report Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	More entries available. The complete report does not fit into a single report response message; the report request message shall be repeated to receive the remaining report entries.

5.14 Financial Counter Type

Value	Description
0	Debit
1	Credit
2	Reversal debit
3	Reversal credit

5.15 Transaction Data Change Reason Code

Value	Description
1	myOne Card with discount code

2	Bonus Card brand detected

5.16 Bonus Card Brand Number

For the moment, a number is used instead of the EP2 Brand Identifier. This is due to the missing Brand ID ‘Visa Bonus Card’ within the EP2 5.3 specification.

Value	Description
1	Jelmoli Paycard
2	Jelmoli Visa Bonus Card
3	Visa Bonus Card
4	Jelmoli Geschenkkarte

5.17 Transaction Request Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	silent; do not display a message when starting a transaction. This suppresses the 'Present Card' message and allows for custom messages while waiting for the card presentation.
4	Report UID of unknown NFC tags in status reponse, e.g. Mifare cards with a data structure unknown to the terminal. If this flag is not set, the transaction will be aborted in this case. What kind of tags are supported at all depends on the terminal device type.

5.18 Abort Transaction Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	silent; do not display a message while aborting

5.19 Batch Capture Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	Manual (ep2): Perform a forced data capture including the last transaction in the log; this makes a later reversal of the last transaction impossible.
2	No submission (ep2): Don’t send the transactions to the acquirers, but only to the POS Management System.
4	No transamission (ep2): Don’t send the transaction to the POS Management System, but only to the acquriers.

5.20 Set of Transaction Data Change Reasons

This data structure can be sent by the ECR in the connect request message as well as by the terminal in the transaction data change request. The reason specifc data (e.g. MyOneDiscountCode) is only meaningful when sent in the transaction data change request.

Tag	Name	Format	Description	Presence

‘BF 83 49’	SetTrxDataChangeReason		c	Set of Transaction Data Change Reasons	m
	‘BF 83 4A’	TrxDataChangeReason	c	Reason to change transaction data; may occur multiple times	o

5.21 Transaction Data Change Reason

This constructed data type can appear multiple times within SetTrxDataChangeReason.

Tag	Name		Format	Description	Presence
‘BF 83 4A’	TrxDataChangeReason		c	Transaction Data Change Reason	m
	‘9F 83 4B’	TrxDataChangeReasonCode	n	Reason code to change the current transaction data	m
	‘9F 83 4C’	MyOneDiscountCode	n(SIZE(2))	myOne Rabattcode; applies to Reason Code 1	o
	‘9F 83 4D’	BonusCardBrandNumber	n(SIZE(2))	Bonus Card Brand Number; applies to Reason Code 2	o

5.22 Dialog Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	Leave message on screen after finishing the dialog
2	Display text along with user input
64	Replace default idle texts ‘Welcome’ & ‘Welcome, present card’

5.23 Dialog Resource

Value	Description	Terminal locking required
1010	Free text DialogResourceArgs content: - ‘0C’ – UTF-8 string	no **Note**: If terminal is not locked, no dialog response message will be sent
1060	Free text, card brands shown DialogResourceArgs content: - ‘0C’ – UTF-8 string	no **Note**: If terminal is not locked, no dialog response message will be sent
1082	Free text for password verification; asterisks shown for entered password DialogResourceArgs content: ‘0C’ – UTF-8 string	yes
10008	'Please select a pump and confirm'	yes
10009	'Please enter your mileage + OK	yes
10010	'Please enter additional info + OK'	yes
10011	'Please choose function + OK"	yes
15013	Free text, with configurable button text DialogResourceArgs content: - ‘0C’ – UTF-8 string free text - ‘0C’ – UTF-8 string left custom button text - ‘0C’ – UTF-8 string right custom button text	yes

5.24 Input Type

Value	Description
0	No input is entered; can be used to request an ok/cancel answer or simply display a text message.
51	Numeric input; result is returned as a UTF8String. This input type only works resources foreseen for clear text input as 'Please select pump'
101	Verify input against known password; result is returned as a Boolean. The value to check against must must be supplied as ReferencePassword

5.25 Dialog Button

Value	Description

1	OK
2	Cancel
16	Left custom button
32	Right custom button

5.26 Dialog Result

Value	Description
1	OK
2	User pressed the stop key
3	Dialog timed out
4	Dialog was closed with a cancel UI request
6	User pressed the left custom button
7	User pressed the right custom button

5.27 Printer Status

Value	Description
0	OK
1	No paper
2	Paper jam. This status is only reported if the hardware can detect this condition.
4	Printer overheat. Thermal printer module has got too hot and needs to cool down before further printing.
8	Printer error. Usually a missing printer module.
16	Printer buffer full.
32	Timeout. Printer hardware did not respond in a timely manner.

5.28 Transaction Request Flags

The following values encode individual flag bits and can be combined by bitwise OR.

Value	Description
1	Start transaction silently i.e. without changing display content or acoustic signals.