



CAPE

Card Payments

Message Usage Guide



Version 6.0
4 December 2017

© 2017 nexo-standards AISBL All rights reserved.

This information is protected by international intellectual property laws and its use is governed by the applicable End-User license

TABLE OF CONTENTS

1	Introduction	12
1.1	<i>Purpose and Use of this Guide.....</i>	12
1.2	<i>Intended Audience.....</i>	12
1.3	<i>Scope of the Document.....</i>	12
1.4	<i>Messages Covered in this Guide.....</i>	12
1.5	<i>How this Guide was created.....</i>	14
1.6	<i>nexo and ISO 20022.....</i>	14
1.7	<i>ISO 20022 Intellectual Property Rights Policy.....</i>	15
1.8	<i>Message Transport.....</i>	15
1.9	<i>Security.....</i>	15
1.10	<i>Coding</i>	15
1.11	<i>Related Documents and Guides.....</i>	15
1.12	<i>Conventions.....</i>	16
1.13	<i>What's new in the edition 6.....</i>	16
2	Message Exchange and Processes	17
2.1	<i>Card Payment.....</i>	17
2.1.1	<i>Authorisation.....</i>	17
2.1.2	<i>Completion</i>	17
2.1.3	<i>Financial capture</i>	18
2.1.4	<i>Batch</i>	18
2.1.5	<i>Dynamic Currency Conversion</i>	18
2.2	<i>Scope of Card Payments.....</i>	20
2.3	<i>Types of environments</i>	21
2.3.1	<i>Online only.....</i>	21
2.3.1.1	<i>Financial capture made during the authorisation exchange.....</i>	21
2.3.1.2	<i>Financial capture made during the completion exchange.....</i>	23
2.3.2	<i>Semi-online authorisations</i>	25
2.3.2.1	<i>Capture of offline and online transactions during completion</i>	25
2.3.2.2	<i>Capture during authorisation for online transactions and during completion for offline authorisations.....</i>	26
2.3.3	<i>Offline only.....</i>	27
2.3.4	<i>Capture by Batch.....</i>	28
2.3.4.1	<i>Authorisation through an authorisation exchange without completion with capture in batch</i>	28
2.3.4.2	<i>Authorisation through authorisation and completion exchanges with capture in batch.....</i>	29
2.3.4.3	<i>Offline authorisation with capture in batch.....</i>	30
2.3.4.4	<i>Capture of unsuccessful transactions in batch</i>	31
2.3.5	<i>Batch of Authorisations.....</i>	32
2.4	<i>Cancellation.....</i>	33
2.4.1	<i>Cancellation through cancellation advice or batch.....</i>	34
2.4.2	<i>Cancellation through cancellation request and advice exchanges</i>	38
2.4.3	<i>Cancellation declined by the Acquirer</i>	40
2.4.4	<i>Error in Cancellation Message Exchanges</i>	41
2.4.4.1	<i>Cancellation declined after timeout of a cancellation request</i>	41
2.4.4.2	<i>Cancellation successful after timeout of a completion advice</i>	42

2.4.4.3	Cancellation declined after timeout of a completion advice	44
2.4.4.4	Cancellation declined after timeout	46
2.4.4.5	Cancellation successful after completion errors	48
2.4.4.6	Cancellation in batch mode.....	49
2.4.4.7	Declined cancellation in batch mode.....	51
2.4.4.8	Declined cancellation after timeout in batch mode and cancellation request	53
2.4.4.9	Declined cancellation in batch mode after timeout in cancellation request	55
2.4.4.10	Cancellation in batch mode of a transaction captured but not yet cleared	57
2.5	Batch.....	59
2.5.1	Introduction.....	59
2.5.2	Data Organisation.....	60
2.5.2.1	AcceptorBatchTransfer	60
2.5.2.2	AcceptorBatchTransferResponse	60
2.5.2.3	Data Factorisation	61
2.5.2.4	Multiplicity.....	62
2.5.3	Types of Batch Transfer	63
2.5.3.1	Batch containing Completion and Cancellation Transactions	63
2.5.3.2	Batch containing Authorisation Transactions	64
2.6	Reconciliation Process	67
2.6.1	Introduction.....	67
2.6.2	Reconciliation Period Identification	68
2.6.3	Transaction Totals	71
2.6.4	Reconciliation Exchange	73
2.7	Diagnostic Messages	75
2.8	Reject Message.....	79
3	Message Functionalities.....	82
3.1	Message Organisation.....	82
3.1.1	Management of the message	82
3.1.1.1	Management information	83
3.1.1.2	Parties involved in the message	84
3.1.1.3	Traceability information.....	85
3.1.2	ApplicationData	86
3.1.2.1	Environment	87
3.1.2.2	Context.....	88
3.1.2.3	Transaction.....	88
3.1.3	SecurityTrailer	89
3.2	Traceability	90
3.2.1	Security level	90
3.2.2	Usage Condition	90
3.2.3	Traceability in an exchange	90
3.3	Message Retransmission	92
3.3.1	Acceptor Behaviour	92
3.3.1.1	Late Responses to an Advice message	93
3.3.1.2	Unordered Responses to Retransmitted Advice messages.....	94
3.3.2	Acquirer Behaviour	95
3.4	Error Handling.....	97
3.4.1	Error Cases	97
3.4.1.1	Acceptor is Unable to Send a Message	97

3.4.1.2	Acquirer Receives an Unacceptable Message.....	98
3.4.1.3	Acquirer is Unable to Process the Message.....	99
3.4.1.4	Acquirer is Unable to Send a Message.....	99
3.4.1.5	Acceptor has not Received a Response Message.....	101
3.4.1.6	Acceptor Receives an Unacceptable Message.....	101
3.4.1.7	Acceptor is Unable to Process the Response Message.....	102
3.4.1.8	Acquirer or Acceptor has received a Duplicate Message.....	103
3.4.2	Acceptor Error Handling.....	103
3.4.2.1	Reverse the Transaction.....	104
3.4.2.2	Retransmission of the Advice.....	104
3.4.2.3	Terminate the Exchange in Progress.....	105
3.4.2.4	Ignore the Error.....	105
3.4.3	Acquirer Error Handling.....	106
3.4.3.1	Message Rejection.....	106
3.4.3.2	Retransmission of the Response.....	106
3.4.3.3	Ignore the Error.....	107
4	Messages and Usage.....	108
4.1	<i>Configuration Parameters/ Condition of Presence</i>	108
4.2	<i>Authorisation Messages</i>	109
4.2.1	AcceptorAuthorisationRequest (caaa.001.001.06).....	109
4.2.1.1	Constraints.....	121
4.2.2	AcceptorAuthorisationResponse (caaa.002.001.06).....	123
4.2.2.1	Constraints.....	128
4.3	<i>Completion Messages</i>	130
4.3.1	AcceptorCompletionAdvice (caaa.003.001.06).....	130
4.3.1.1	Constraints.....	141
4.3.2	AcceptorCompletionAdviceResponse (caaa.004.001.06).....	142
4.3.2.1	Constraints.....	144
4.4	<i>Cancellation Messages</i>	145
4.4.1	AcceptorCancellationRequest (caaa.005.001.06).....	145
4.4.1.1	Constraints.....	151
4.4.2	AcceptorCancellationResponse (caaa.006.001.06).....	152
4.4.2.1	Constraints.....	154
4.4.3	AcceptorCancellationAdvice (caaa.007.001.06).....	155
4.4.3.1	Constraints.....	161
4.4.4	AcceptorCancellationAdviceResponse (caaa.008.001.06).....	162
4.5	<i>Reconciliation Messages</i>	164
4.5.1	AcceptorReconciliationRequest (caaa.009.001.06).....	164
4.5.1.1	Constraints.....	166
4.5.2	AcceptorReconciliationResponse (caaa.010.001.05).....	167
4.5.2.1	Constraints.....	169
4.6	<i>Batch</i>	170
4.6.1	AcceptorBatchTransfer (caaa.011.001.06).....	170
4.6.1.1	Constraints.....	202
4.6.2	AcceptorBatchTransferResponse (caaa.012.001.06).....	203
4.6.2.1	Constraints.....	205
4.7	<i>Diagnostic Messages</i>	206
4.7.1	AcceptorDiagnosticRequest (caaa.013.001.06).....	206

4.7.1.1	Constraints	208
4.7.2	AcceptorDiagnosticResponse (caaa.014.001.05).....	209
4.8	Reject Message	211
4.8.1	AcceptorRejection (caaa.015.001.05)	211
4.9	Dynamic Currency Conversion Messages	212
4.9.1	AcceptorCurrencyConversionRequest (caaa.016.001.04)	212
4.9.2	AcceptorCurrencyConversionResponse (caaa.017.001.04).....	217
4.9.3	AcceptorCurrencyConversionAdvice (caaa.018.001.01)	220
4.9.4	AcceptorCurrencyConversionAdviceResponse (caaa.019.001.01)	223
5	Dynamic of the Payment Exchanges	226
5.1	<i>Introduction</i>	226
5.2	<i>Determination of Payment Cases</i>	226
5.2.1	Elements Impacting the Message Flow	226
5.2.1.1	Authorisation Type.....	226
5.2.1.2	Authorisation Result	227
5.2.1.3	Incident after Authorisation	227
5.2.1.4	Merchant Forced Acceptance	228
5.2.1.5	Capture Type.....	228
5.2.1.6	Completion Exchange.....	229
5.2.2	List of Payment Cases.....	230
5.3	<i>Table of Payment Cases</i>	234
5.4	<i>Cancellation Exchange</i>	240
5.4.1	List of Cancellation Cases	241
5.4.2	Table of Cancellation Cases	243
6	Additional Payment Services	247
6.1	<i>Voice Authorisation</i>	247
6.2	<i>Deferred Payments</i>	251
6.2.1	Introduction	251
6.2.2	Constraints on the Protocol	252
6.3	<i>Cashback</i>	255
6.4	<i>Cash Advance</i>	256
6.5	<i>Gratuity</i>	257
6.6	<i>Reservation</i>	257
6.6.1	Introduction	257
6.6.2	Reservation steps	258
6.6.2.1	Initial Reservation	258
6.6.2.2	Update Reservation	258
6.6.2.3	Cancellation of the Reservation (before payment)	258
6.6.2.4	Payment for the Reservation	258
6.6.2.5	No-Show Payment for the Reservation	258
6.6.2.6	Additional Payment (after Payment for the Reservation)	258
6.6.2.7	No-show payment without performing a prior reservation service	258
6.6.3	Message flows	258
6.6.4	Description of the steps	259
6.6.4.1	Initial reservation	259
6.6.4.2	Update reservation	259
6.6.4.3	Cancellation of the reservation (before payment)	260

6.6.4.4	Payment after reservation.....	260
6.6.5	Reservation transaction data	260
6.6.5.1	Initial reservation	260
6.6.5.2	Update reservation	263
6.6.5.3	Cancellation of a Reservation before payment.....	265
6.6.5.4	Payment after reservation.....	266
6.6.5.5	Additional payment	269
6.6.5.6	No-show payment without prior reservation	272
6.6.5.7	Batch	273
6.6.5.8	Reconciliation	274
6.7	<i>Dynamic Currency Conversion (DCC)</i>	275
6.7.1	Introduction.....	275
6.7.2	Currency conversion during the authorisation	275
6.7.2.1	Transaction eligible for DCC and DCC accepted by the Cardholder.....	277
6.7.2.2	Transaction eligible for DCC and partially approved by the Issuer	279
6.7.2.3	Transaction not eligible for DCC by the DCC service provider	281
6.7.2.4	Transaction not eligible for DCC by the Agent	282
6.7.2.5	Transaction eligible for DCC but DCC refused by the Cardholder	283
6.7.2.6	Transaction eligible for DCC but interrupted due to an incident.....	285
6.7.3	Currency conversion prior to the authorisation	288
6.7.3.1	Transaction eligible for DCC and DCC accepted by the Cardholder.....	290
6.7.3.2	Transaction eligible for DCC and partially approved by the Issuer	293
6.7.3.3	Transaction not eligible for DCC by DCC service provider	295
6.7.3.4	Transaction not eligible for DCC by the Agent	297
6.7.3.5	Transaction eligible for DCC but DCC refused by the Cardholder	299
6.7.3.6	Transaction eligible for DCC but interrupted due to an incident.....	301
6.7.4	Reconciliation Totals	305
7	Messages Examples	306
8	Transport Protocols and Services	307
8.1	<i>Protocols Organisation</i>	307
8.2	<i>TCP Protocol</i>	309
8.2.1	Typical Use.....	309
8.2.2	Message Delimitation	309
8.2.3	Addressing.....	309
8.3	<i>Transport Services</i>	310
8.3.1	Message Delimitation	310
8.3.1.1	Definition	310
8.3.1.2	Specifications	310
8.3.1.3	Typical Example of Implementation	310
8.3.1.4	Notes.....	310
8.3.2	Connection and Data Transfer Management.....	312
8.3.2.1	Connection Services.....	312
8.3.2.2	Data Transfer	313
8.3.3	Single Message Pair Exchange.....	314
8.3.3.1	Definition	314
8.3.3.2	Specifications	314
8.3.3.3	Notes.....	315
8.3.4	Multiple Message Pair Exchange	316
8.3.4.1	Definition	316

8.3.4.2	Specifications	316
8.3.4.3	Notes.....	318
8.3.5	Addressing.....	319
8.3.5.1	Definition	319
8.3.5.2	Specifications	319
8.3.6	Flow Control	320
8.3.6.1	Definition	320
8.3.6.2	Specifications	320
8.3.7	Error Cases	321
8.3.7.1	Unable to Establish a Transport Connection (ERTR01)	321
8.3.7.2	Transport Connection Broken (ERTR02)	321
8.3.7.3	Unable to Send a Message (ERTR03).....	322
8.3.7.4	Message Too Big (ERTR04).....	322
8.3.7.5	Late Arrival (ERTR05)	323
8.3.7.6	Max Number of Connections (ERTR06).....	323
8.3.7.7	Incomplete Application Message (ERTR07).....	324
8.3.7.8	Other Errors.....	324
8.3.8	Transport Service Parameters	325
8.3.9	Connection and Data Management State Diagrams	327
8.3.9.1	Recipient Party Diagram	327
8.3.9.2	Initiating Party Diagram	328

Figures

Figure 1: Authorisation with capture (successful transaction).....	21
Figure 2: Authorisation with capture (failed transaction).....	22
Figure 3: Completion with capture (successful transaction).....	23
Figure 4: Completion with capture (failed transaction).....	24
Figure 5: Capture of offline and online transactions during completion	25
Figure 6: Capture during authorisation for online transactions	26
Figure 7: Capture through completion of previously offline-authorised transactions	27
Figure 8: Authorisation through an authorisation exchange without completion with capture in batch	28
Figure 9: Authorisation through authorisation and completion exchanges with capture in batch	29
Figure 10: Off-line authorisation with capture in batch	30
Figure 11: Capture of unsuccessful transactions in batch	31
Figure 12: Batch of Authorisations	32
Figure 13: Cancellation through cancellation advice or batch.....	35
Figure 14: Cancellation through cancellation advice and batch exchanges	37
Figure 15: Cancellation through cancellation request and advice exchanges	39
Figure 16: Declined cancellation of a captured transaction	40
Figure 17: Cancellation declined after timeout of a cancellation request.....	41
Figure 18: Cancellation successful after timeout of a completion advice	43
Figure 19: Cancellation declined after timeout of a completion advice	45
Figure 20: Cancellation declined after timeout	47
Figure 21: Cancellation successful after completion errors	48
Figure 22: Cancellation in batch mode	50
Figure 23: Declined cancellation in batch mode.....	52
Figure 24: Declined cancellation after timeout in batch mode and cancellation request	54
Figure 25: Declined cancellation in batch mode after timeout cancellation request	56
Figure 26: Cancellation in batch mode of a transaction captured but not yet cleared	58
Figure 27: BatchTransfer structure.....	60
Figure 28: BatchTransferResponse structure	61
Figure 29: Batch containing Financial Authorisations	65
Figure 30: Batch containing Authorisations.....	66
Figure 31: Reconciliation Exchange	67
Figure 32: Reconciliation Period Assigned by the Acceptor	68
Figure 33: Reconciliation Period Assigned by the Acquirer	69
Figure 34: Reconciliation between the Authorisation and the Completion.....	69
Figure 35: Overlapping of Reconciliation Periods	70
Figure 36: Diagnostic Exchange	75
Figure 37: Diagnostic Reject	76

Figure 38: Diagnostic Request from an Acceptor to an Agent	76
Figure 39: Diagnostic Request from an Intermediary Agent	78
Figure 40: Rejection of an Authorisation	79
Figure 41: Rejection of an Authorisation to an Agent.....	80
Figure 42: Rejection of an Completion to an Agent	81
Figure 43: Example of Traceability with an Intermediary Agent.....	91
Figure 44: Retransmission of an Advice by the Acceptor	92
Figure 45: Late Responses to an Advice message.....	93
Figure 46: Unordered Responses to Retransmitted Advice messages	94
Figure 47: Retransmission of an AdviceResponse by the Acquirer.....	95
Figure 48: Non-Receipt of an Advice by the Acquirer	95
Figure 49: Error Cases in Message Exchange.....	97
Figure 50: Acceptor Error Handling in a Message Exchange	103
Figure 51: Acquirer Error Handling in a Message Exchange	106
Figure 52: Payment Cases Tree List.....	232
Figure 53: Payment Cases Tree List (Con't)	233
Figure 54 Cancellation Cases Tree List	242
Figure 55: Successful Voice Authorisation captured with Completion	247
Figure 56: Unsuccessful Voice Authorisation captured with Completion.....	248
Figure 57: Successful Voice Authorisation Interleaved with another transaction	249
Figure 58: Successful Voice Authorisation Captured with Batch without Completion	250
Figure 59: Deferred Payment	251
Figure 60: Interleaved Deferred Payment Transactions	251
Figure 61: Approved Deferred Payment without Delivery	252
Figure 62: Declined Deferred Payment	252
Figure 63: Deferred Payment Captured by Completion	253
Figure 64: Deferred Payment Captured by Batch	253
Figure 65: Approved Deferred Payment without Delivery	254
Figure 66: Currency conversion during the authorisation	276
Figure 67: Transaction eligible for DCC and DCC accepted by the Cardholder	278
Figure 68: Transaction eligible for DCC and partially approved by the Issuer	280
Figure 69: Transaction not eligible for DCC by DCC service provider	281
Figure 70: Transaction not eligible for DCC by Agent.....	282
Figure 71: Transaction eligible for DCC but DCC refused by cardholder	284
Figure 72: : Incident at Agent level.....	285
Figure 73: Incident at DCC service provider level.....	287
Figure 74: Currency conversion before authorisation	289
Figure 75: Transaction eligible for DCC and DCC accepted by the Cardholder	292
Figure 76: Transaction eligible for DCC and partially approved by the Issuer	294
Figure 77: Transaction not eligible for DCC by DCC service	296

Figure 78: Transaction not eligible for DCC by the Agent.....	298
Figure 79: Transaction eligible for DCC but DCC refused by the Cardholder	300
Figure 80: Incident at the Agent level.....	302
Figure 81: Incident at DCC service provider level	304
Figure 82: Protocols Organisation.....	307
Figure 83: Transport Adaptation Layer.....	307
Figure 84: Peer-to-peer TCP Transport Protocol	309
Figure 85: Gateway TCP Transport Protocol	309
Figure 86: Header Length.....	310
Figure 87: Connection Services	312
Figure 88: Transport Connection Management Service Primitives	313
Figure 89: Application Protocol Message Flow	313
Figure 90: Single Message Pair Exchange Sequence Flow	314
Figure 91: Single Message Pair Exchange Transport Primitives Sequence Flow	315
Figure 92: Multiple Message Pair Exchange Sequence Flow	316
Figure 93: Multiple Message Pair Exchange Transport Primitives Sequence Flow.....	317
Figure 94: Transport Address.....	319
Figure 95: Late Arrival Error Example	323
Figure 96: Recipient Party Connection Management State Diagram	327
Figure 97: Initiating Party Connection Management State Diagram	328

Tables

Table 1: Message type and version	14
Table 2: MessageFunction Values	231
Table 3: List of Payment Cases	240
Table 4: Cancellation MessageFunction Values	242
Table 5: List of Cancellation Cases	246

1 Introduction

1.1 Purpose and Use of this Guide

This guide outlines how to use the CAPE Card Payments messages in the context of the acceptor-to-acquirer transactions (**caaa**) ISO 20022 Business Area¹. It provides a comprehensive view on how these messages fit within a card payment business process and the activities of the involved parties. Also included are detailed explanations and examples of the use of the message components to convey specific information related to these processes and activities. This guide acts as a complement to the ISO 20022 CAPE Message Definition Report and the XML Schema for those exchanges. It complies with ISO 20022 rules and specifications.

The guide provides information regarding the application of the included messages in a general context. Additional documents, published by individual user communities, may be available that discuss the application of this standard in a more specific context.

1.2 Intended Audience

The present guide is intended for both business people and message developers. It is also targeted for manufacturers, software providers, banks and card payment services providers offering these messages to their clients, technology firms seeking to embed support for these messaging standards into their applications, and standards organisations that wish to use the payment kernel as part of the messages they offer.

1.3 Scope of the Document

The present guide covers basic payments with a limited number of optional features such as cash-back, gratuity, DCC, loyalty. It includes cancellation, refund and reconciliation associated with these payments.

1.4 Messages Covered in this Guide

The present guide covers card payments in the acceptor-to-acquirer domain only and complies with ISO 20022 message specifications published by ISO 20022 in "Message Definition Report Card Payments Exchanges – Acceptor to Acquirer" on February 2016.

It addresses the following categories of exchanges of messages:

- Authorisation messages², cover the messages exchanged between an Acceptor and an Acquirer to initiate and, in some implementations, to finalise the card-based payment transactions initiated at a Point-of-Interaction.
- Completion messages³, cover the messages exchanged between an Acceptor and an Acquirer to finalise the card-based payment transactions initiated at a Point-of-Interaction.

¹ http://www.iso20022.org/documents/general/ISO20022_BusinessAreas.pdf

² AcceptorAuthorisationRequest (**caaa.001.001.06**) and AcceptorAuthorisationResponse (**caaa.002.001.06**)

³ AcceptorCompletionAdvice (**caaa.003.001.06**) and AcceptorCompletionAdviceResponse (**caaa.004.001.06**)

- Cancellation messages⁴, cover the messages exchanged between an Acceptor and an Acquirer to cancel successfully completed payment transactions or other types of transactions (e.g. reservations) which have not yet been cleared. A cancellation is sometimes called a manual reversal.
- Reconciliation messages⁵, cover the messages exchanged between an Acceptor and an Acquirer to perform checks and balances between the card acceptor and the acquirer for a reconciliation period.
- Batch transfer messages⁶ cover the messages exchanged in batch between an Acceptor and an Acquirer to finalise and/or capture card-based payment transactions initiated at a Point-of-Interaction.
- Diagnostic messages⁷ cover the messages exchanged between an *InitiatingParty* and a *RecipientParty* to prove that messages can be exchanged correctly between the parties.
- Rejection message⁸ cover the message send by a *RecipientParty* to an *InitiatingParty* to indicate that the *RecipientParty* could not process the received message.
- Dynamic Currency conversion message⁹ cover the messages exchanged between an Acceptor or an acquirer and a DCC service provider (Agent or Acquirer).

All messages exchanged according to the rules defines in this Message User Guide must specify ProtocolVersion 6.0 in the message header.

Message	Message type and version
AcceptorAuthorisationRequest	caaa.001.001.06
AcceptorAuthorisationResponse	caaa.002.001.06
AcceptorCompletionAdvice	caaa.003.001.06
AcceptorCompletionAdviceResponse	caaa.004.001.06
AcceptorCancellationRequest	caaa.005.001.06
AcceptorCancellationResponse	caaa.006.001.06
AcceptorCancellationAdvice	caaa.007.001.06
AcceptorCancellationAdviceResponse	caaa.008.001.06
AcceptorReconciliationRequest	caaa.009.001.06

⁴ AcceptorCancellationRequest (**caaa.005.001.06**), AcceptorCancellationResponse (**caaa.006.001.06**), AcceptorCancellationAdvice (**caaa.007.001.06**) and AcceptorCancellationAdviceResponse (**caaa.008.001.06**)

⁵ AcceptorReconciliationRequest (**caaa.009.001.06**) and AcceptorReconciliationResponse (**caaa.010.001.05**)

⁶ AcceptorBatchTransfer (**caaa.011.001.06**) and AcceptorBatchTransferResponse (**caaa.012.001.06**)

⁷ AcceptorDiagnosticRequest (**caaa.013.001.06**) and AcceptorDiagnosticResponse (**caaa.014.001.05**)

⁸ AcceptorRejection (**caaa.015.001.05**)

⁹ AcceptorCurrencyConversionRequest (**caaa.016.001.04**), AcceptorCurrencyConversionResponse (**caaa.017.001.04**), AcceptorCurrencyConversionAdvice (**caaa.018.001.01**) and AcceptorCurrencyConversionAdviceResponse (**caaa.019.001.01**)

Message	Message type and version
AcceptorReconciliationResponse	caaa.010.001.05
AcceptorBatchTransfer	caaa.011.001.06
AcceptorBatchTransferResponse	caaa.012.001.06
AcceptorDiagnosticRequest	caaa.013.001.06
AcceptorDiagnosticResponse	caaa.014.001.05
AcceptorRejection	caaa.015.001.05
AcceptorCurrencyConversionRequest	caaa.016.001.04
AcceptorCurrencyConversionResponse	caaa.017.001.04
AcceptorCurrencyConversionAdvice	caaa.018.001.01
AcceptorCurrencyConversionAdviceResponse	caaa.019.001.01

Table 1: Message type and version

1.5 How this Guide was created

This guide was created through the combined efforts of the nexo Acquirer Protocol Working Group involving representatives belonging to various industries (banks, card payments service providers, manufacturers, software service providers, retailers, etc.). It follows ISO 20022 rules for the design of Message Usage Guides.

1.6 nexo and ISO 20022

nexo¹⁰ is a major standardisation initiative in card-based payment transactions involving card-payment experts belonging to various industries. Nexo has been created by merging various standardisation initiative like EPAS Org, CIR and Oscar. Since its inception, EPASOrg has been following the ISO 20022 methodology in developing standards with the ultimate objective to become a full-fledged ISO 20022 standard. ISO 20022 is part of the International Organization for Standardization (ISO) under Technical Committee 68 (TC68), which is the Financial Services Technical Committee of the International Organization for Standardization. ISO 20022 message standards are submitted through the ISO 20022 Registration Management Group (RMG) via a Business Justification. When this Business Justification is approved, the RMG assigns the proposed message standards to a Standards Evaluation Group (SEG).

The Cards and Related Retail Financial Services SEG mandated by the RMG to address the card payments business domain officially endorsed the nexo CAPE (Card Payment Exchanges) series of messages in June 2010.

Complete information on the membership of the ISO 20022 SEGs, the ISO 20022 Financial Repository, and the message maintenance and registration process can be found on www.iso20022.org. For more information on ISO itself, please see www.iso.org.

For more information on nexo, the body in charge of the nexo and CAPE messages, please visit www.nexo-standards.org.

¹⁰ <http://www.nexo-standards.org>

1.7 ISO 20022 Intellectual Property Rights Policy

EPASOrg acknowledges and abides by the ISO 20022 IPR policy outlined as follows: *“Organizations that contribute information to be incorporated into the ISO 20022 Repository shall keep any Intellectual Property Rights (IPR) they have on this information. A contributing organization warrants that it has sufficient rights on the contributed information to have it published in the ISO 20022 Repository through the ISO 20022 Registration Authority in accordance with the rules set in ISO 20022. To ascertain a widespread, public and uniform use of the ISO 20022 Repository information, the contributing organization grants third parties a non-exclusive, royalty-free license to use the published information”*.

The EPAS End-User License Agreement¹¹ can be downloaded from the ISO 20022 Web site.

1.8 Message Transport

CAPE messages are designed to be transport protocol independent. The CAPE standard does, however, provide message transport conventions of its own (including header and trailer).

1.9 Security

CAPE messages embed their own security data components and structures to ensure an adequate security in the transmission of the information.

1.10 Coding

CAPE messages are proposed in ISO 20022 XML derived from ISO 20022 Message Definitions.

1.11 Related Documents and Guides

The complete catalog of official ISO 20022 CAPE messages, including the Message Definition Reports and XML Schemas, is available on the ISO 20022 official Web site (www.iso20022.org) All other CAPE or nexo related information can be found by visiting the nexo Web site (www.nexo-standards.org).

Useful information about XML is available from the following sources:

- XML recommendations of W3C can be found at:
<http://www.w3c.org/TR/2000/REC-xml-20081126>
<http://www.w3.org/TR/2006/REC-xml11-20060816/>
- XML Schema recommendations of W3C can be found at:
<http://www.w3c.org/TR/xmlschema-0/>
<http://www.w3c.org/TR/xmlschema-1/>
<http://www.w3c.org/TR/xmlschema-2/>

¹¹ http://www.iso20022.org/documents/general/EPASOrg_EPAS_End-user_License_Final.pdf

The security element of these messages are described in the nexo document named Card Payment Protocols Security v2.1.

1.12 Conventions

The words MUST, SHOULD and MAY will be used throughout this document with the meaning defined by [RFC2119].

Items represented in grey in the table are not covered by the present MUG. Within the terms of the protocol they may be used as agreed between the sender and receiver.

1.13 What's new in the edition 6

This edition brings the following improvements:

- Dynamic Currency Conversion Advice
- New fields for 3DS authentication
- New fields for customer consent and programme selection
- New fields for card product subtype
- New amounts fields in DCC for actual, minimum and maximum amount
- Message examples provided in a separate document.

2 Message Exchange and Processes

2.1 Card Payment

2.1.1 Authorisation

An authorisation is used in a card payment to request the approval of the transaction. It can be done either remotely (on-line to the Acquirer through an authorisation exchange) or locally (off-line authorisation) depending on the business context.

The possible outcomes of an authorisation are:

- a successful authorisation
- a declined authorisation
- a technical problem (eg. timeout, unable to go online, etc.)
- a voice referral, which in turn may result in a successful authorisation or a decline.

An online authorisation exchange is made up an AcceptorAuthorisationRequest message (*caaa.001.001.06*) used to request the authorisation of the related transaction and an AcceptorAuthorisationResponse message (*caaa.002.001.06*) used to provide the outcome of the authorisation.

2.1.2 Completion

Completion is the process of finalising the interaction between the cardholder and the merchant at the POI.

The outcome of a completion is the acceptance or decline of the transaction by the Acceptor.

A completion exchange may also be required by an Acquirer to finalise the transaction:

- when the Acceptor is configured to support completion,
- in the authorisation response (if required by the Acquirer),
- when an online approved authorisation did not complete successfully.

A completion exchange is made up an AcceptorCompletionAdvice message (*caaa.003.001.06*) sent by an Acceptor to notify the Acquirer of the outcome of the transaction and an AcceptorCompletionAdviceResponse message (*caaa.004.001.06*) used by the Acquirer to acknowledge this notification.

A completion exchange can be performed in realtime at the end of the transaction or in a store-and-forward mode of operation.

An AcceptorCompletionAdvice is used for three main purposes:

- to reverse an authorisation (also called a “reversal”)
- to inform the Acquirer of the outcome of a service

- to transfer the transaction data to the Acquirer for a further clearing and settlement (*financial capture*).

The Acquirer must accept an *AcceptorCompletionAdvice* message. In case of an error, the message is resent to the Acquirer (see section 3.3 *Message Retransmission*).

2.1.3 Financial capture

The ultimate financial settlement of a card payment transaction requires a prior *financial capture* of the elements of this transaction by the Acquirer or by an agent acting on his behalf. The responsibility of keeping the financial information is then transferred from the Acceptor to the Acquirer for a further clearing and settlement of the transaction.

A *financial capture* can take place in realtime, after a short delay or at a later time as part of a batch transfer.

A *financial capture* process can be done:

- within an authorisation exchange;
- within a completion exchange;
- through a batch transfer.

2.1.4 Batch

A batch is a collection of transactions sent to the Acquirer for a further processing.

A batch transfer is processed through an *AcceptorBatchTransfer* (*caaa.011.001.06*) and an *AcceptorBatchTransferResponse* (*caaa.012.001.06*) which is used to acknowledge or to reject some or all transactions contained in a batch.

2.1.5 Dynamic Currency Conversion

When the currency of the transaction is not the same as the currency of the card, it may be difficult for a cardholder to estimate the actual amount he will have to pay in his own currency at the moment of the payment.

Dynamic Currency Conversion (DCC) is a service offered by a merchant to allow the cardholder to make the payment in the currency of his card instead of the currency of the merchant when different.

The cardholder can choose whether to accept the DCC service or not. When DCC is allowed by the merchant, the currency conversion is managed through a DCC service provider proposing to the cardholder the converted amount to be ultimately debited on his card.

If the DCC service is refused by the cardholder, the transaction then proceeds as a normal card payment transaction in the currency of the merchant.

The DCC service may rely on an agent acting between the merchant and the DCC service provider. The following use cases involve agents in the process, but direct exchanges between a merchant and a DCC service provider or between a merchant and an acquirer follow the same logic.

The role of the DCC service provider can be played by any party (Agent, Acquirer, etc.).

DCC is supported by two types of exchanges:

- Authorisation exchanges where a pair of *AcceptorAuthorisationRequest* (caaa.001.001.06) and *AcceptorAuthorisationResponse* (caaa.002.001.06) messages are used
- Dynamic Currency Conversion exchanges where a pair of *AcceptorCurrencyConversionRequest* (caaa.016.001.04) and *AcceptorCurrencyConversionResponse* (caaa.017.001.04) messages are used to fulfil the same purpose.

An Advice may be sent to DCC provider for the final outcome of the DCC service with messages *AcceptorCurrencyConversionAdvice* (caaa.018.001.01) and *AcceptorCurrencyConversionAdviceResponse* (caaa.019.001.01).

Use cases outlining some typical DCC scenarios are described in Chapter 6.7 Dynamic Currency Conversion (DCC) .

2.2 Scope of Card Payments

Card payment exchanges can be domestic, cross-border and/or cross currency. They may be used by an Acceptor to request to an Acquirer to initiate and/or complete a card transaction.

An Acceptor can directly exchange with an Acquirer a series of card payment messages. The proposed messages may also go through the intermediation of one or more Intermediary Agents to request similar services. The Intermediary Agent is then acting as an agent of the Acquirer, the Acceptor or both.

In a normal card payment process, the Acquirer further forwards to the Issuer (e.g. a payment institution or a similar entity issuing the payment card to the cardholder) the related authorisation request to get an ultimate response from the entity entitled to issue the authorisation of payment for a purchase which was initiated with the card issued by them. The protocol needs to transport all relevant information to enable the Issuer to take the appropriate decision i.e. either to authorise or to decline the card payment transaction submitted by the Acquirer.

In an alternative process and based on the information submitted by an Acceptor, the Acquirer may take the responsibility to authorise or decline the transaction (below a certain amount, for instance) on behalf of the Issuer. He may also decide to forward to the Issuer the information required by the Issuer to take the ultimate authorisation decision. In this case, the information is exchanged through an Acquirer-to-Issuer protocol. This exchange is out of scope of the present specification.

In some specific environments (e.g. petrol), some purchasing information, such as a list of products, may accompany the authorisation since the final payment can occur if and only if the list of products was approved by the issuer of the payment card.

In that case, either the Acceptor submits with an authorisation request the list of products for approval to the Acquirer which further forwards this list to the Issuer for approval or, based on the information contained in the authorisation request, the Issuer submits in the authorisation response the restricted list of products (e.g. fuel dispensing products) authorised for the Cardholder. It is then the responsibility of the Acceptor to deliver a product or a service which complies with the product(s) contained in this list.

2.3 Types of environments

Depending on business constraints and the types of equipment used, typical implementations may be classified in:

- Online-only
- Offline with online capability
- Offline-only with batch capture

2.3.1 Online only

In this mode all transactions are processed in a pure on-line mode of operation. The only required exchanges are Authorisation and Completion:

Completion is required:

- In case of reversal
- For financial capture when not done during authorization.
- In case of request in the authorization response or by configuration.

2.3.1.1 Financial capture made during the authorisation exchange

Case of a successful transaction

An *AcceptorAuthorisationRequest* message is sent by an Acceptor to an Acquirer to request the approval of a payment transaction. The transaction succeeds after the approval of the authorisation. In some cases, the Acquirer may require a completion exchange for this transaction.

An *AcceptorAuthorisationResponse* message is returned by the Acquirer to inform the Acceptor of the outcome of the request. The responsibility of the financial data (*financial capture*) is transferred from the Acceptor to the Acquirer with the *AcceptorAuthorisationRequest* message. The Acquirer then stores this information for the further clearing and settlement of the transaction.

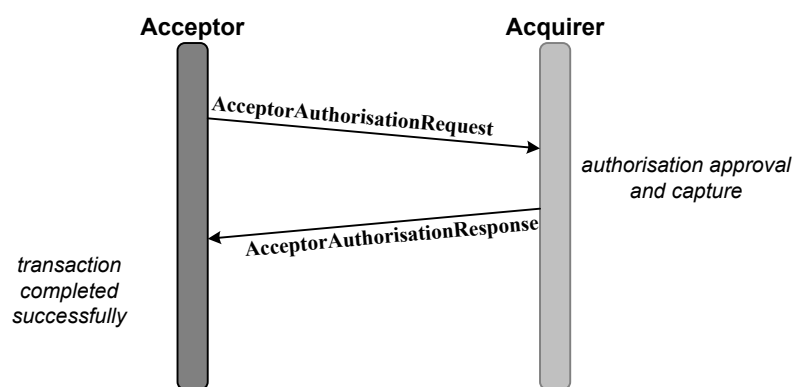


Figure 1: Authorisation with capture (successful transaction)

Case of a failed transaction

An *AcceptorAuthorisationRequest* message is sent by an Acceptor to an Acquirer to request the approval of a transaction. The transaction fails after the approval of the authorisation or if the Acceptor has not received a response to the *AcceptorAuthorisationRequest* message. A completion exchange is initiated by the Acceptor to reverse both the authorisation and the financial capture of the transaction.

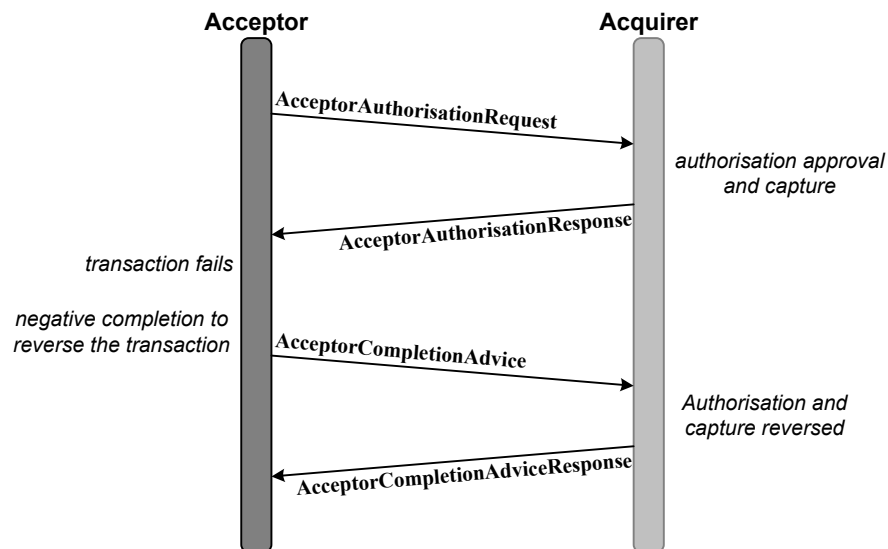


Figure 2: Authorisation with capture (failed transaction)

2.3.1.2 Financial capture made during the completion exchange

Case of a successful transaction

An *AcceptorAuthorisationRequest* message is sent by an Acceptor to an Acquirer to request the approval of the transaction. The transaction succeeds after the approval of the authorisation.

An *AcceptorCompletionAdvice* message is sent by the Acceptor to the Acquirer to capture the transaction. An *AcceptorCompletionAdviceResponse* message is sent back by the Acquirer to the Acceptor to notify the Acceptor about the successful receipt of the advice.

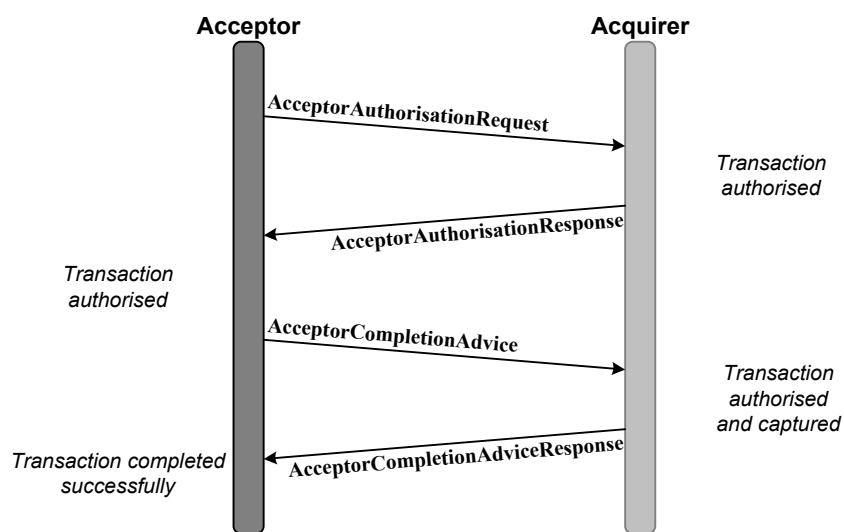


Figure 3: Completion with capture (successful transaction)

Case of a failed transaction

An *AcceptorAuthorisationRequest* message is sent by an Acceptor to an Acquirer to request the approval of the transaction. The transaction fails after the approval of the authorisation and/or the Acceptor has not received a response to the *AcceptorAuthorisationRequest* message. A completion exchange is then initiated by the Acceptor to reverse the authorisation of the transaction.

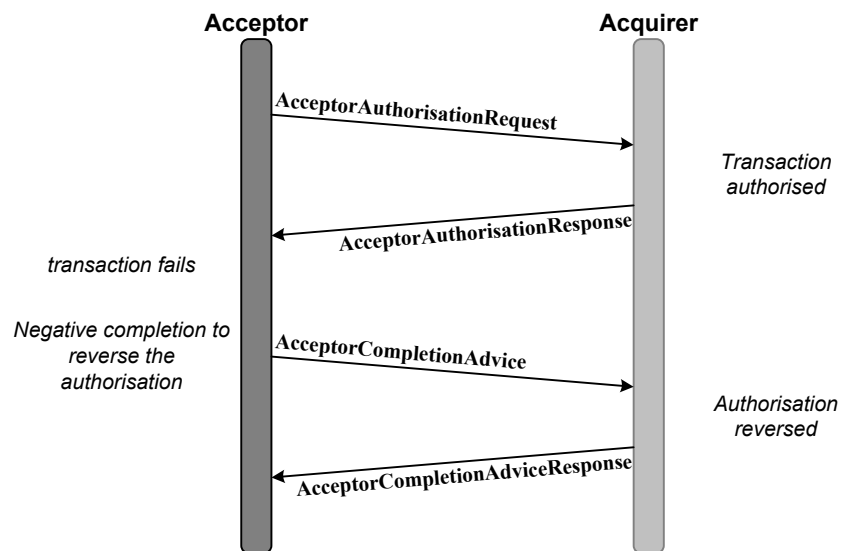


Figure 4: Completion with capture (failed transaction)

2.3.2 Semi-online authorisations

Authorisations are processed either online or offline depending on the context of the transaction. In normal cases, the financial capture of the transaction is made during the completion exchange.

2.3.2.1 Capture of offline and online transactions during completion

In this scenario, financial capture of previously online and offline authorised transactions is carried out during completion.

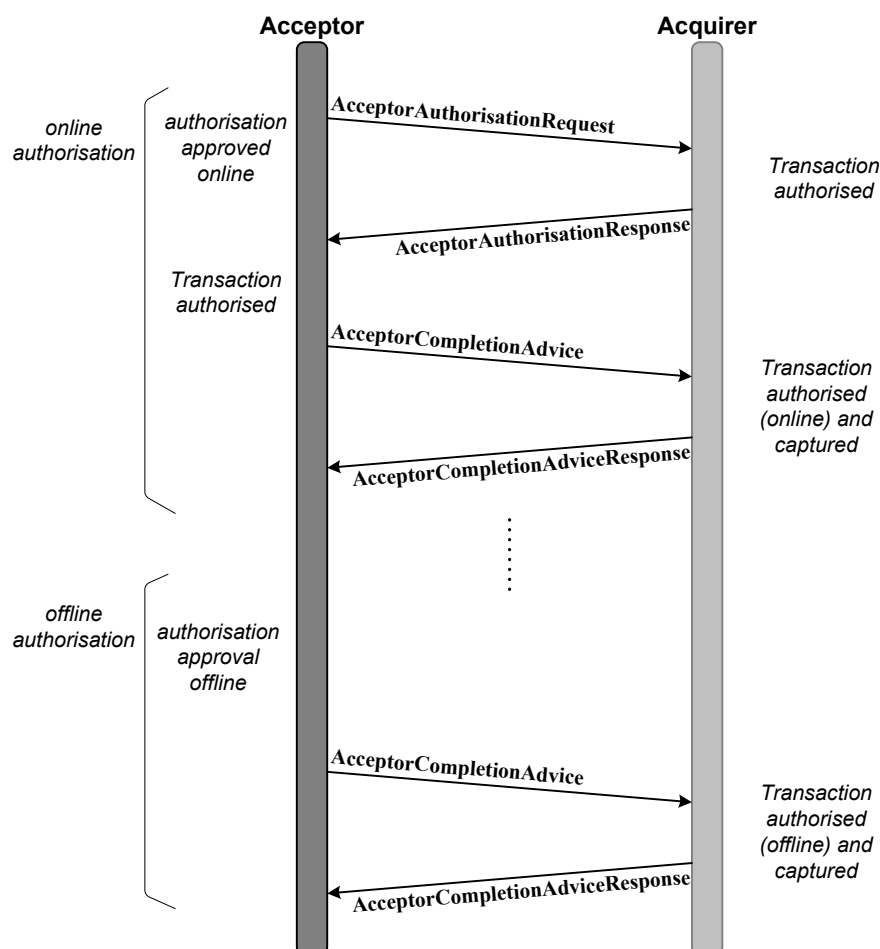


Figure 5: Capture of offline and online transactions during completion

2.3.2.2 Capture during authorisation for online transactions and during completion for offline authorisations

In order to reduce the number of exchanges, the financial capture may be carried out during authorisation exchanges for on-line authorisations and during completion exchanges for off-line authorisations.

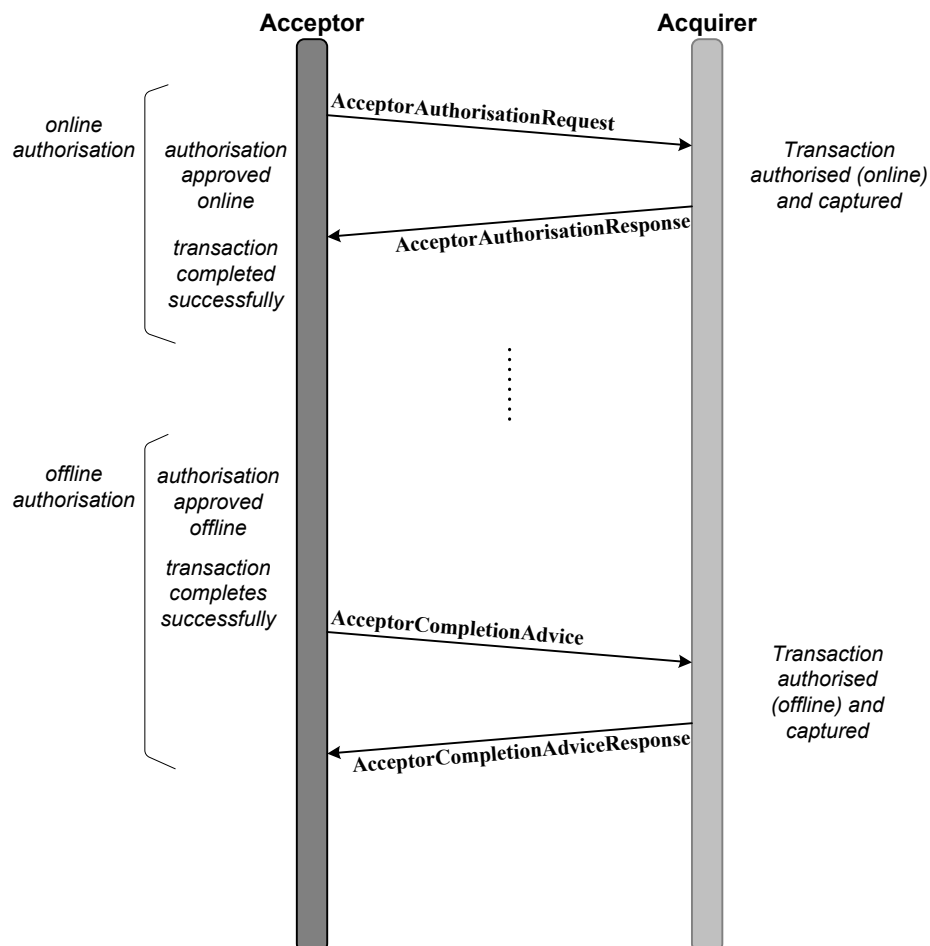


Figure 6: Capture during authorisation for online transactions and during completion for offline transactions

2.3.3 Offline only

In this scenario, an *AcceptorCompletionAdvice* is used to inform the Acquirer about the outcome of a previously offline-authorized transaction and to advise the Acquirer to capture the financial data for clearing and settlement.

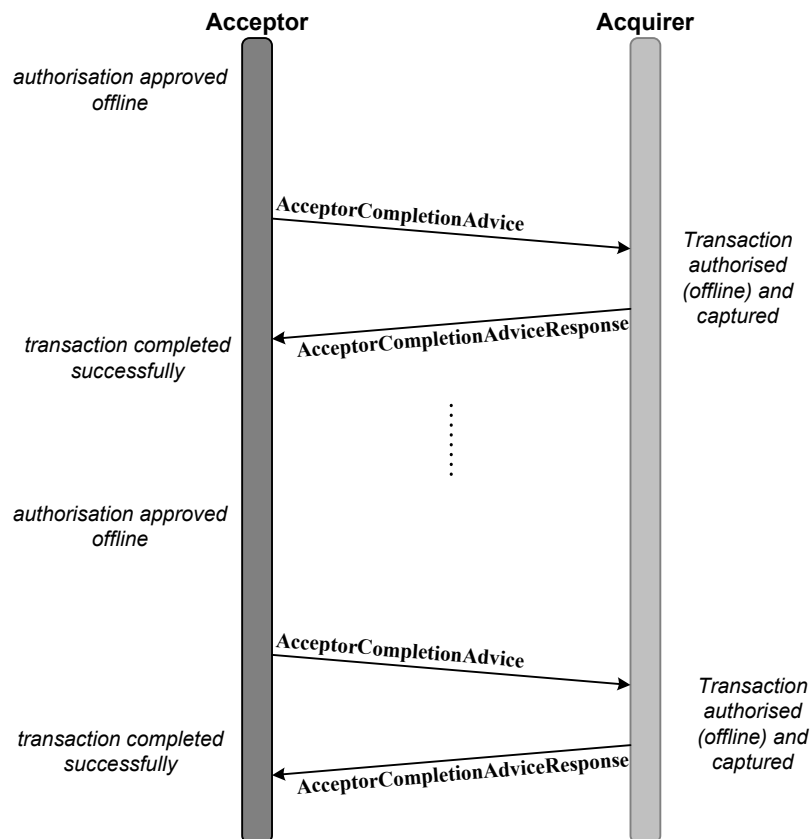


Figure 7: Capture through completion of previously offline-authorized transactions

2.3.4 Capture by Batch

The financial capture of transactions proceeds in batch.

The main relevant types of batch situations are:

- authorisation through an authorisation exchange without completion with capture in batch
- authorisation through authorisation and completion exchanges with capture in batch
- offline authorisation with capture in batch
- unsuccessful transactions communicated to the Acquirer in batch

2.3.4.1 Authorisation through an authorisation exchange without completion with capture in batch

In this scenario, an *AcceptorBatchTransfer* message is used by an Acceptor to advise an Acquirer to capture in batch mode transactions authorised through previous online or offline authorisation exchanges. The Acquirer captures the financial data of the transactions for clearing and settlement.

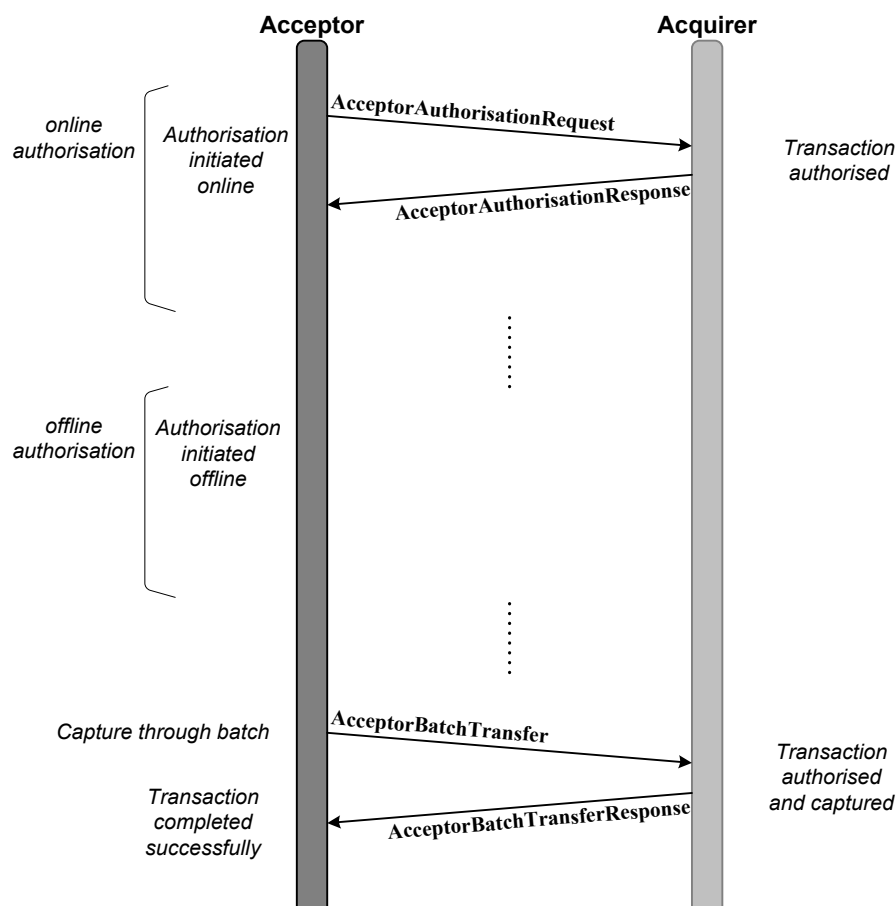


Figure 8: Authorisation through an authorisation exchange without completion with capture in batch

2.3.4.2 Authorisation through authorisation and completion exchanges with capture in batch

In this scenario, an *AcceptorBatchTransfer* message is used by the Acceptor to advise the Acquirer to capture in batch mode transactions authorised through previous authorisation and completion exchanges. The Acquirer captures the financial data of the transactions for clearing and settlement.

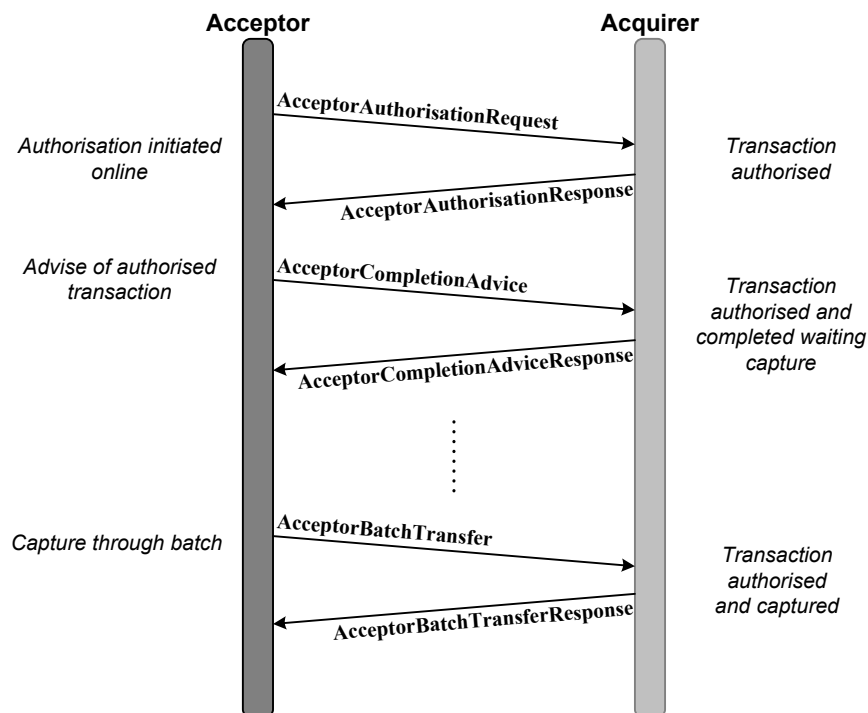


Figure 9: Authorisation through authorisation and completion exchanges with capture in batch

2.3.4.3 Offline authorisation with capture in batch

In this scenario, an *AcceptorBatchTransfer* message is used by the Acceptor to advise the Acquirer to capture in batch mode transactions that were previously authorised offline and stored. The Acquirer captures the financial data of the transactions for clearing and settlement.

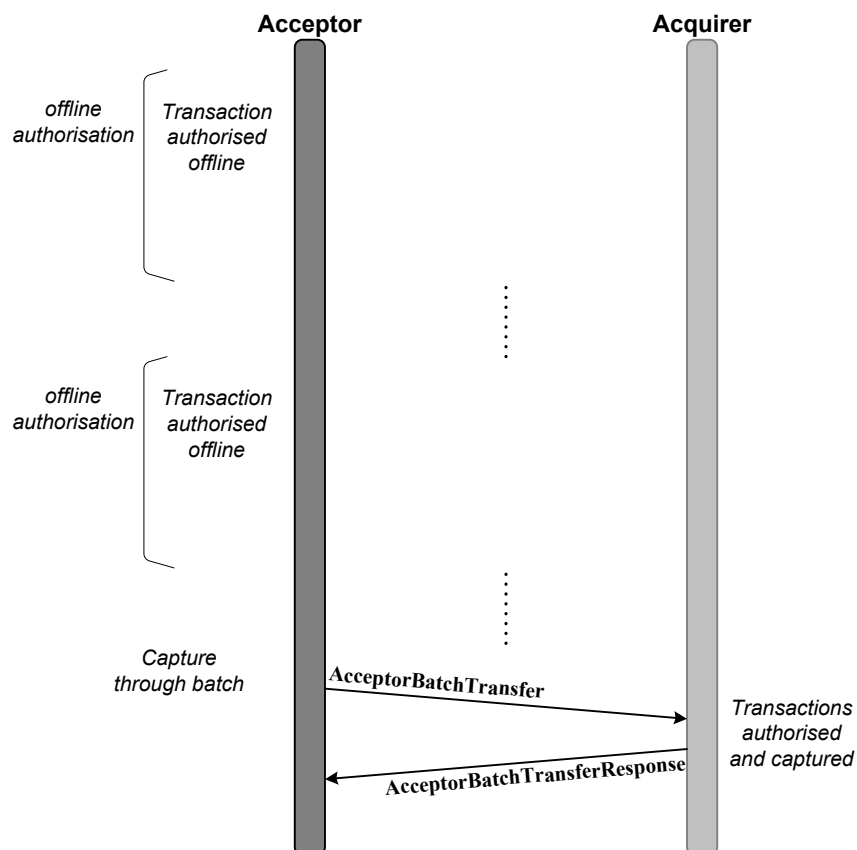


Figure 10: Off-line authorisation with capture in batch

2.3.4.4 Capture of unsuccessful transactions in batch

In this scenario, an *AcceptorBatchTransfer* message is used by the Acceptor to send to the Acquirer for capture in batch mode unsuccessfully completed and reversed transactions.

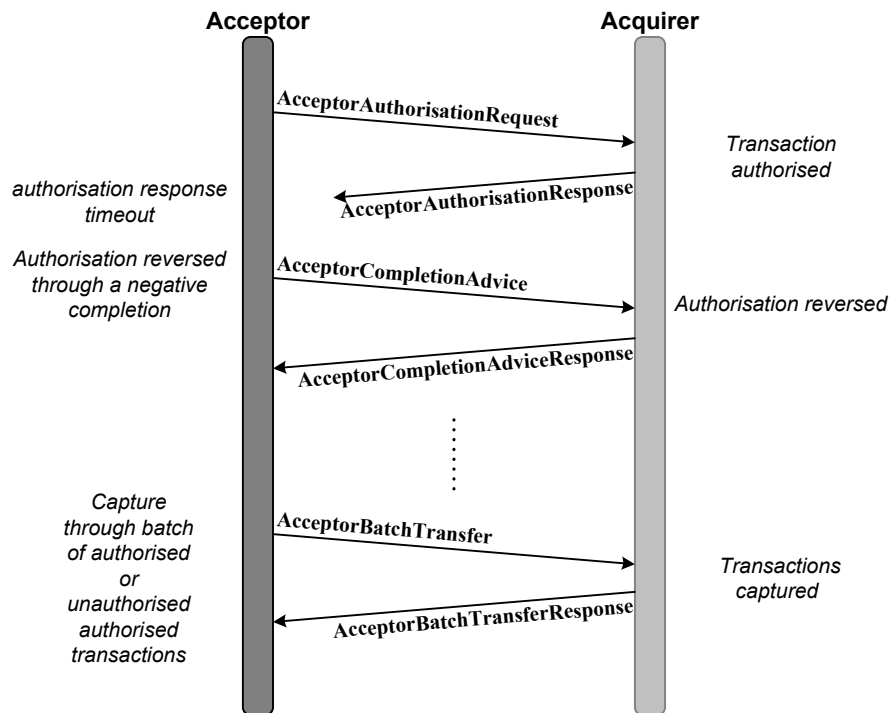


Figure 11: Capture of unsuccessful transactions in batch

2.3.5 Batch of Authorisations

An *AcceptorBatchTransfer* exchange is used by the Acceptor to send a collection of Authorisation Requests to the Acquirer to authorise transactions.

The Acquirer is sending back to the Acceptor in an *AcceptorBatchTransferResponse* the corresponding Authorisation Responses with the outcome of the authorisation process.

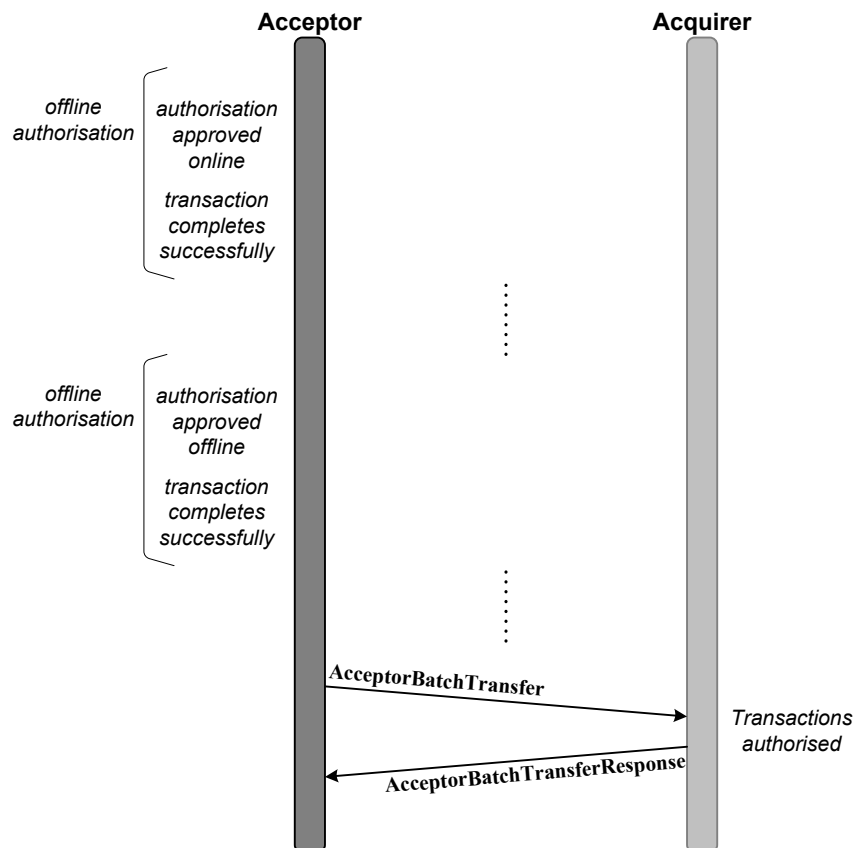


Figure 12: Batch of Authorisations

2.4 Cancellation

Cancellation is initiated by Acceptor to cancel a payment which was successfully completed. . A cancellation is a user requested reversal. A cancellation cannot be revoked.

An *AcceptorCancellationRequest* (caaa.005.001.06) is used by an Acceptor to ask the Acquirer whether a cancellation can be performed, before sending an *AcceptorCancellationAdvice*.

An *AcceptorCancellationAdvice* (caaa.007.001.06)- with or without a prior *AcceptorCancellationRequest* is used by an Acceptor to inform the Acquirer that a cancellation has been completed. It is also used to indicate that no response to an *AcceptorCancellationRequest* was received and that the cancellation was declined.

An *AcceptorCancellationAdvice* (caaa.007.001.06) can only be used without an *AcceptorCancellationRequest* (caaa.005.001.06) if the Acceptor is aware that the transaction was not yet cleared.

An Acquirer can never decline an *AcceptorCancellationAdvice* (caaa.007.001.06). If the Acceptor does not receive an *AcceptorCancellationAdviceResponse* (caaa.008.001.06), the Acceptor has to resend a *AcceptorCancellationAdvice* until the Acceptor receives the corresponding response from the Acquirer (see 3.3 *Message Retransmission* section).

Should an Acquirer decline an *AcceptorCancellationRequest*, an Acceptor always has the possibility to refund the cardholder in cash or with a refund transaction.

The cancellation process may be affected by POI configuration parameters defined outside of the present protocol.

2.4.1 Cancellation through cancellation advice or batch

A cancellation is carried out through an *AcceptorCancellationAdvice* exchange without a prior *AcceptorCancellationRequest*.

This type of cancellation is used when:

- no financial capture of the original transaction occurred (by message or by batch), or
- a financial capture of the original transaction occurred, and
 - the Acceptor is aware that the original transaction has not been cleared by the Acquirer (see note below)
 - the Acceptor has all the data of the original transaction required for building up the *AcceptorCancellationAdvice*.

Note: The Acceptor is able to know that the Acquirer did not clear the transaction because:

- the clearing of the transactions is only allowed after closing the corresponding reconciliation period by a reconciliation exchange, and
- a reconciliation message to close the reconciliation period has not been sent for the reconciliation period that includes the transaction.

A typical scenario is the following one:

- an Acceptor performs an authorisation without financial capture. The transaction is authorised offline or online by the Acquirer. The transaction is successfully completed and stored by the Acceptor.
- since no financial capture occurred, the Acceptor cancels the stored transaction and sends an *AcceptorCancellationAdvice* to advise the Acquirer to reverse the authorisation.
- the Acquirer sends back an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the advice was acknowledged by the Acquirer and the authorisation reversed.

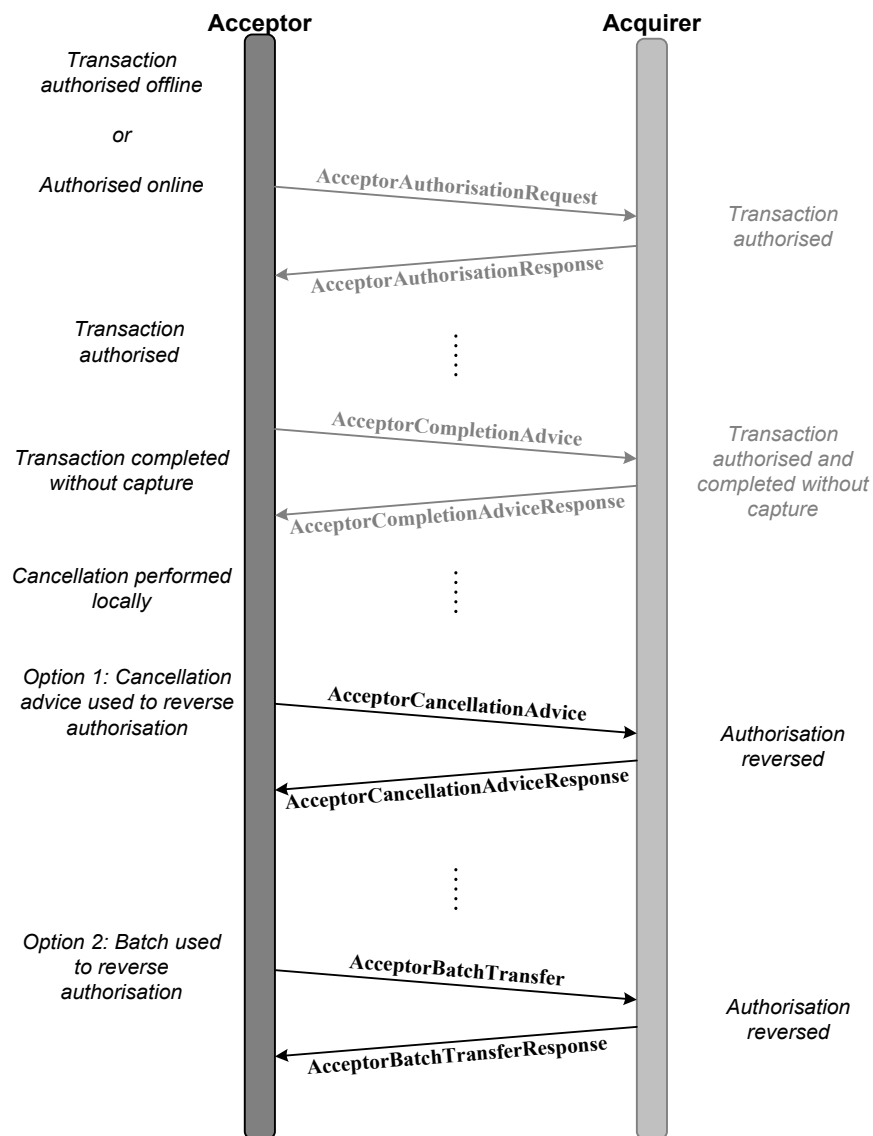


Figure 13: Cancellation through cancellation advice or batch

The message flow is the same as if the Acceptor had sent previously an *AcceptorCompletionAdvice* without financial capture for the authorised transaction before the cancellation was processed.

For offline transactions, it is:

- optional to send an *AcceptorCancellationAdvice* if no *AcceptorCompletionAdvice* for the transaction was previously sent
- mandatory to send an *AcceptorCancellationAdvice* if an *AcceptorCompletionAdvice* was previously sent.

When the Acceptor uses batch transfer for financial capture, the Acceptor sends a batch transfer depending on the configuration, with either:

- Include both the original debit (or credit) and cancellation transaction
- Remove the original debit (or credit) from the batch

An alternative scenario (with financial capture) is the following one:

- an Acceptor initiated several authorisations with financial capture (by individual messages or batch).
- the transactions are authorised and captured by the Acquirer.
- the Acceptor is aware that the Acquirer has not yet cleared the transactions and he completes the cancellation offline.
- the Acceptor sends an *AcceptorCancellationAdvice* to notify the Acquirer of the cancellation of the authorisation and financial capture on his side.
- the Acquirer sends back an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the advice was received by the Acquirer. The Acquirer reverses the authorised and captured transactions.
- should batch transfer be used by the Acceptor for financial capture, the Acceptor sends a batch transfer depending on the configuration, with either:
 - Include both the original debit (or credit) and cancellation transaction
 - Remove the original debit (or credit) from the batch
-

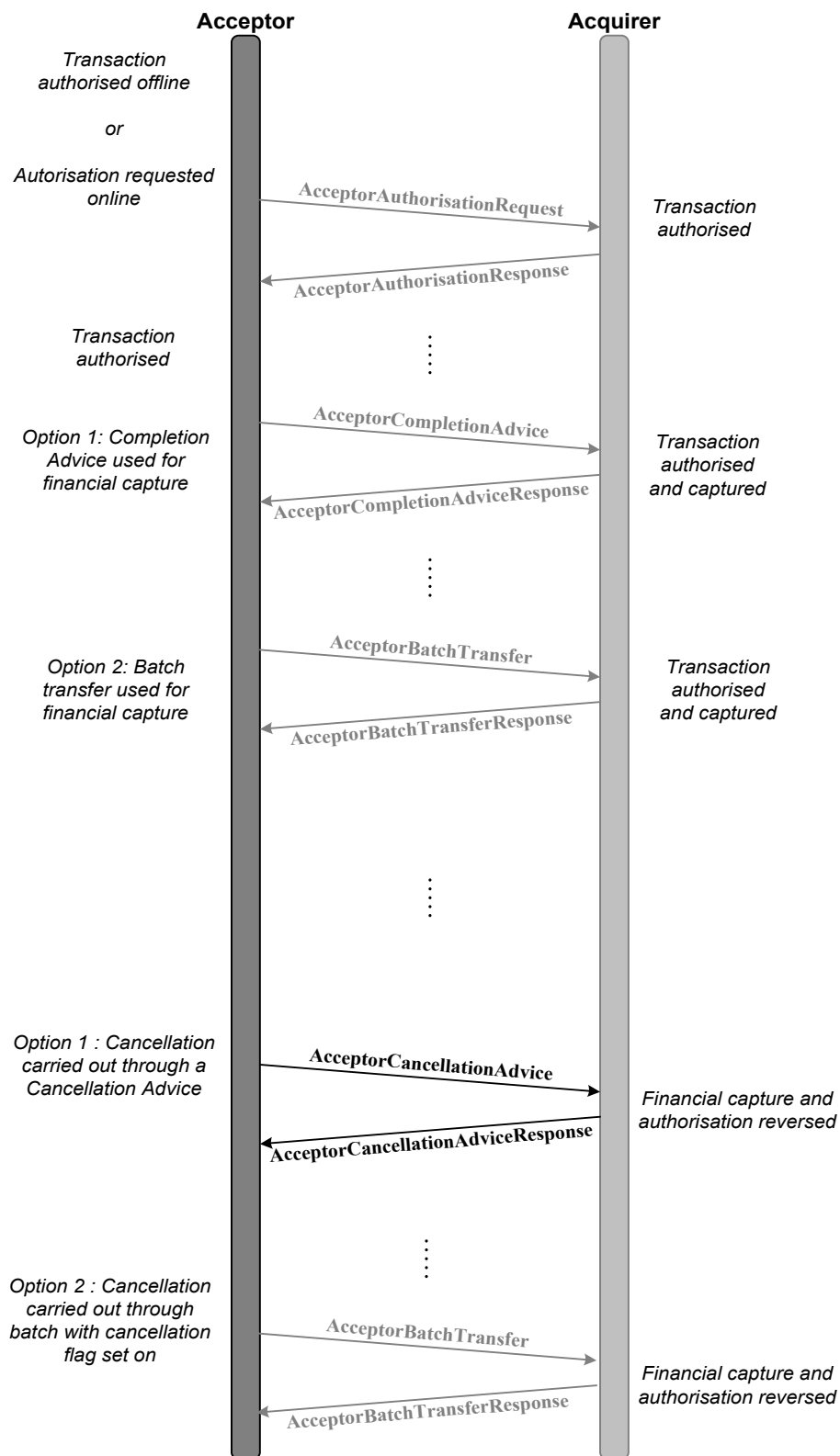


Figure 14: Cancellation through cancellation advice and batch exchanges

2.4.2 Cancellation through cancellation request and advice exchanges

A cancellation is carried out through an *AcceptorCancellationRequest* followed by an *AcceptorCancellationAdvice*.

This type of cancellation is used when all the following conditions are satisfied:

- financial capture of the transaction has already occurred (by message or by batch), and
- the Acceptor is unaware whether the Acquirer cleared the transaction or not, and
- the Acceptor has all the data of the original transaction required for building up the *AcceptorCancellationRequest* and *AcceptorCancellationAdvice* messages.

A typical scenario is the following one:

- an Acceptor performs an authorisation with financial capture. The transaction is authorised offline or online by the Acquirer. The transaction is successfully completed and captured by the Acquirer.
- the Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether the transaction can be cancelled or not.
 - should the cancellation be possible:
 - the Acquirer sends a positive *AcceptorCancellationResponse*.
 - the Acceptor sends back an *AcceptorCancellationAdvice* to complete the cancellation by the Acquirer.
 - the Acquirer responds with an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the advice was received by the Acquirer.
 - should the cancellation not be possible:
 - the Acquirer declines the request with a negative *AcceptorCancellationResponse*.
 - the Acceptor does not send an *AcceptorCancellationAdvice* except if the *AcceptorCancellationResponse* was not received.

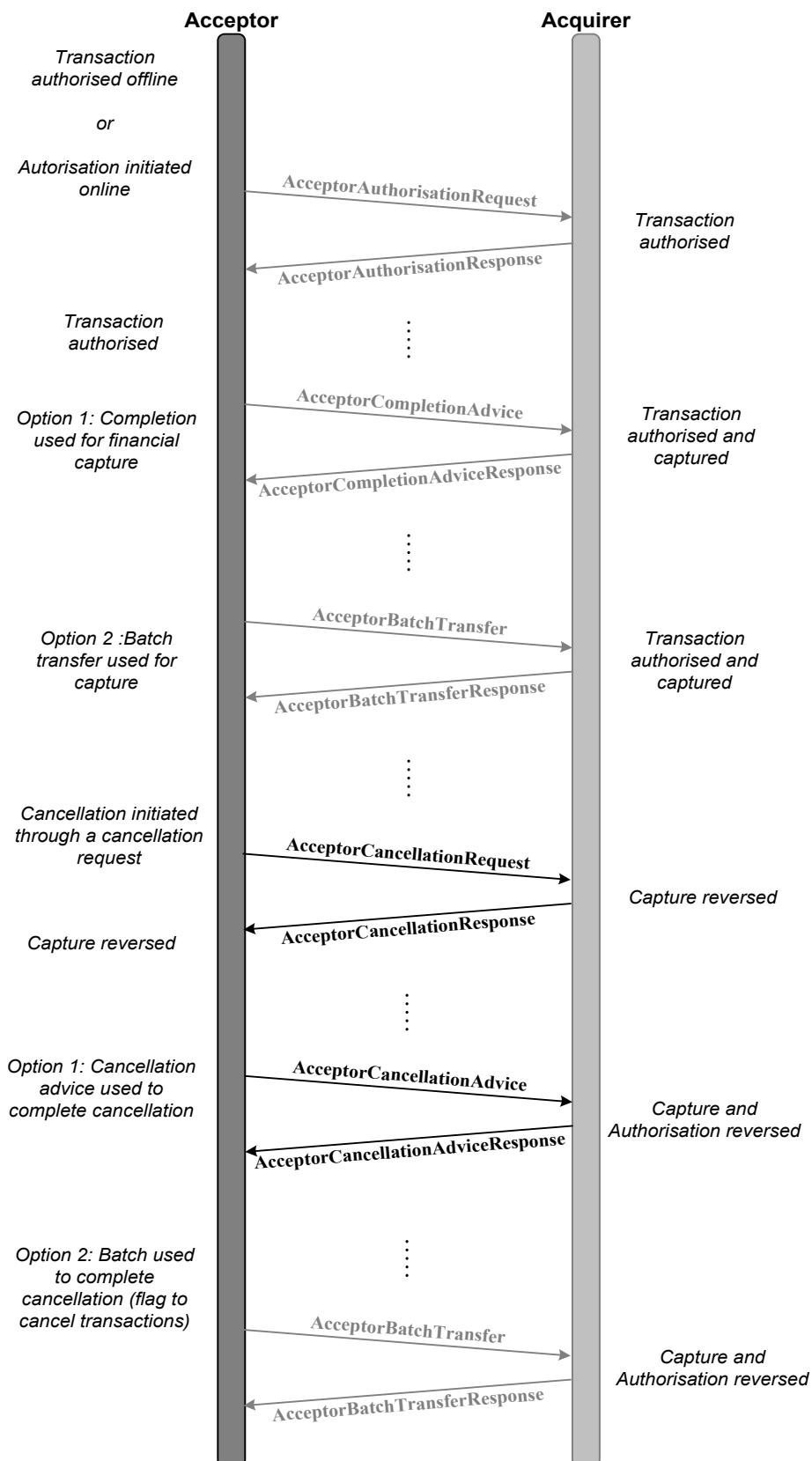


Figure 15: Cancellation through cancellation request and advice exchanges

2.4.3 Cancellation declined by the Acquirer

Should an *AcceptorCancellationRequest* be declined by the Acquirer, no further *AcceptorCancellationAdvice* is sent by the Acceptor.

When the Acceptor used batch transfer for financial capture, the Acceptor may add the cancellation in the batch according to the batch configuration.

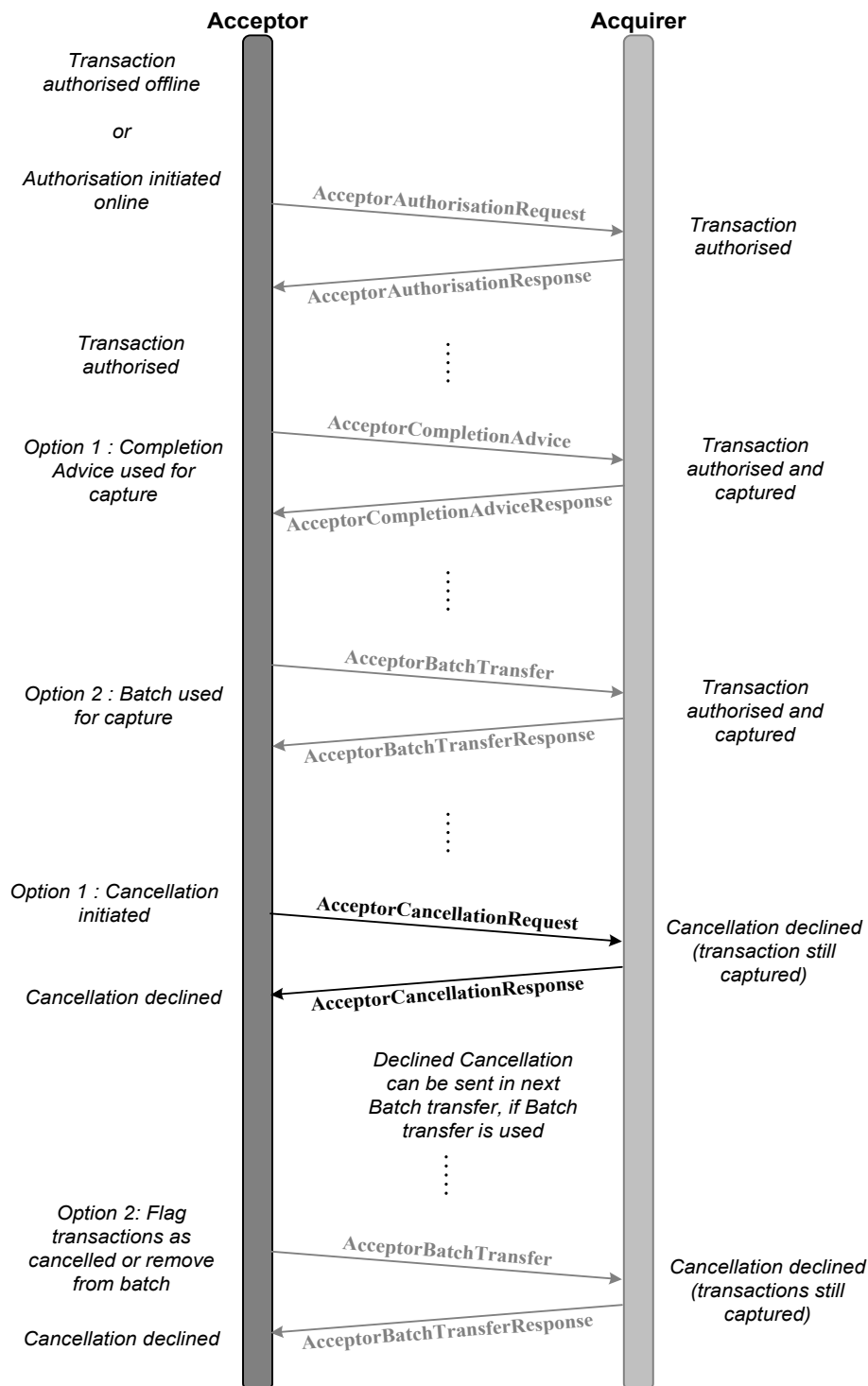


Figure 16: Declined cancellation of a captured transaction

2.4.4 Error in Cancellation Message Exchanges

In this chapter several scenarios are described where the Acceptor did not receive a response from the Acquirer to *AcceptorCancellationRequest* messages.

2.4.4.1 Cancellation declined after timeout of a cancellation request

In this scenario:

- the Acceptor received an *AcceptorCompletionAdviceResponse* from the Acquirer
- the transaction was captured by the Acquirer
- the Acceptor did not receive a response to an *AcceptorCancellationRequest*.

In this case the Acceptor must decline the cancellation to the cardholder. An *AcceptorCancellationAdvice* is sent to the Acquirer:

- to inform the Acquirer that the cancellation was declined to the cardholder

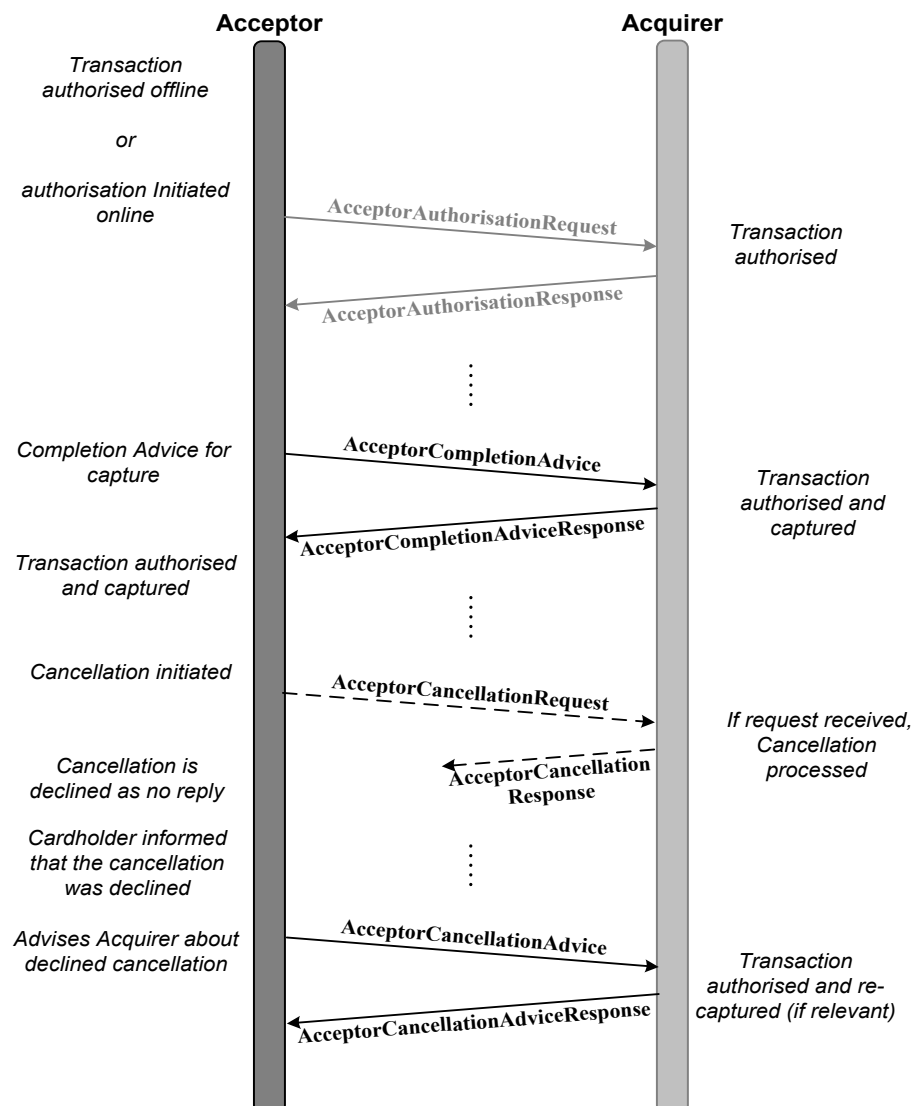


Figure 17: Cancellation declined after timeout of a cancellation request

2.4.4.2 Cancellation successful after timeout of a completion advice

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* to capture the transaction and
- the Acceptor did not receive an *AcceptorCompletionAdviceResponse* and
- the Acceptor wants to cancel the transaction

Without having received an *AcceptorCompletionAdviceResponse*, the Acceptor is unaware whether the transaction was captured or not.

Furthermore, the Acceptor is unaware whether the transaction was cleared or not (if the transaction was actually captured).

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

Should the cancellation be possible:

- the Acquirer sends a *AcceptorCancellationResponse* to inform the Acceptor that the transaction can be cancelled.
- the Acceptor informs the cardholder about the success of the cancellation and sends an *AcceptorCancellationAdvice* to inform the Acquirer about the successful completion of the cancellation.
- the Acquirer responds with an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the advice was received by the Acquirer.

Should clearing occur between the receipt of the *AcceptorCancellationRequest* and the *AcceptorCancellationAdvice*, the Acquirer will refund the cardholder through an Acquirer to Issuer exchange.

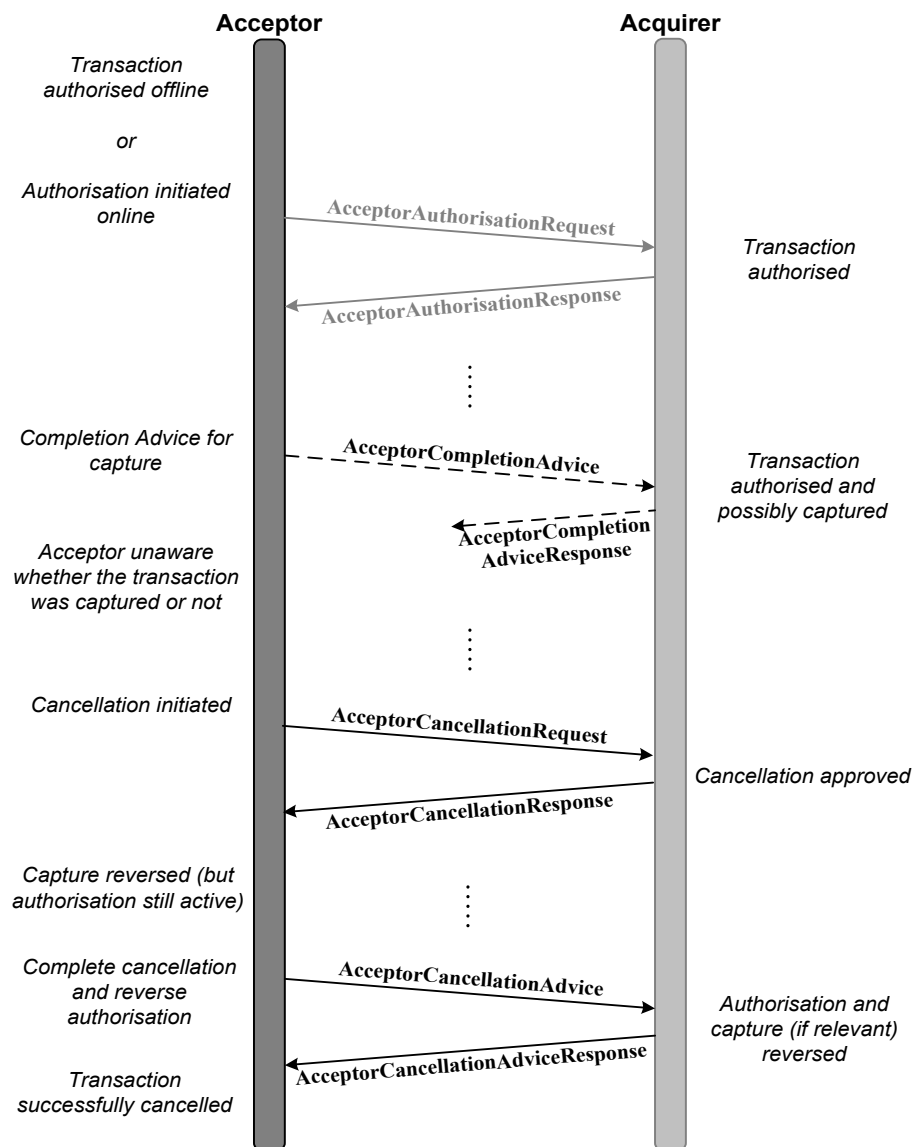


Figure 18: Cancellation successful after timeout of a completion advice

2.4.4.3 Cancellation declined after timeout of a completion advice

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* to capture the transaction and
- the Acceptor did not receive an *AcceptorCompletionAdviceResponse* and
- the Acceptor wants to cancel the transaction.

Without having received an *AcceptorCompletionAdviceResponse*, the Acceptor is unaware whether the transaction was captured or not.

Furthermore, the Acceptor is unaware whether the transaction was cleared or not (if the transaction was actually captured).

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

Option 1

The cancellation is declined by the Acquirer because the transaction has already been captured and cleared:

- the Acquirer sends an *AcceptorCancellationResponse* to inform the Acceptor that the cancellation was declined.
- the Acceptor informs the cardholder that the cancellation was declined by the Acquirer
- the Acceptor needs to send an *AcceptorCompletionAdvice* with the financial data for capture by the Acquirer because he does not know that the transaction is already captured. The Acquirer discards the message since the transaction has already been captured.

Option 2

The cancellation is declined by the Acquirer because he did not receive the *AcceptorCompletionAdvice* message:

- the Acquirer sends an *AcceptorCancellationResponse* to inform the Acceptor that the cancellation was declined.
- the Acceptor informs the cardholder that the cancellation was declined by the Acquirer
- the Acceptor needs to send an *AcceptorCompletionAdvice* with the financial data for capture by the Acquirer. The Acquirer captures the financial data for a further clearing.

The cardholder needs to get a refund for both options.

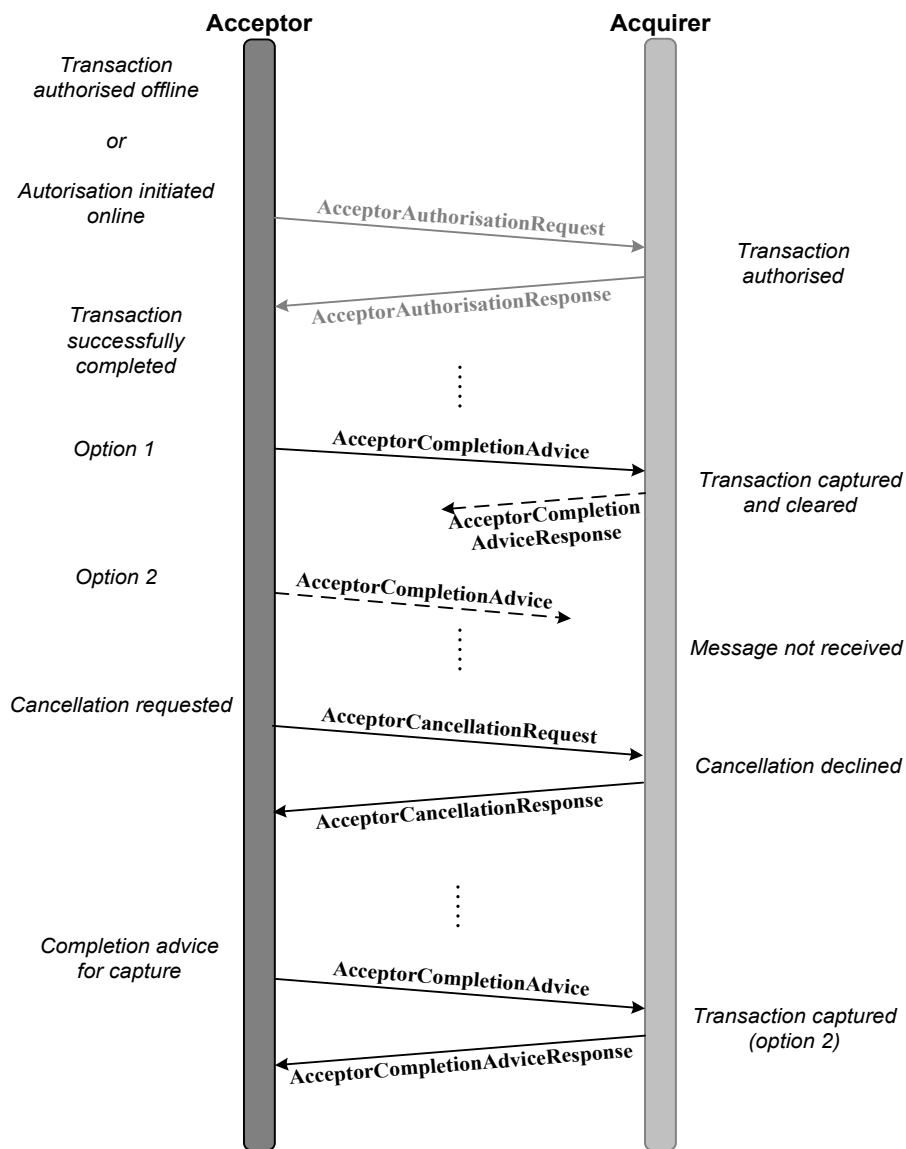


Figure 19: Cancellation declined after timeout of a completion advice

2.4.4.4 Cancellation declined after timeout

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* to capture the transaction and
- the Acceptor did not receive an *AcceptorCompletionAdviceResponse*.

The Acceptor wants to cancel the transaction but is unaware whether the Acquirer has already cleared the transaction or not.

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

The Acceptor does not receive a response for the *AcceptorCancellationRequest* (timeout):

- the Acceptor informs the cardholder that the cancellation was declined
- the Acceptor sends back an *AcceptorCompletionAdvice* with the financial data for capture by the Acquirer
- After receiving an *AcceptorCompletionAdviceResponse* from the Acquirer, the Acceptor sends an *AcceptorCancellationAdvice* to inform the Acquirer that the cancellation was declined.
- The Acquirer acknowledges this advice with an *AcceptorCancellationAdviceResponse*.

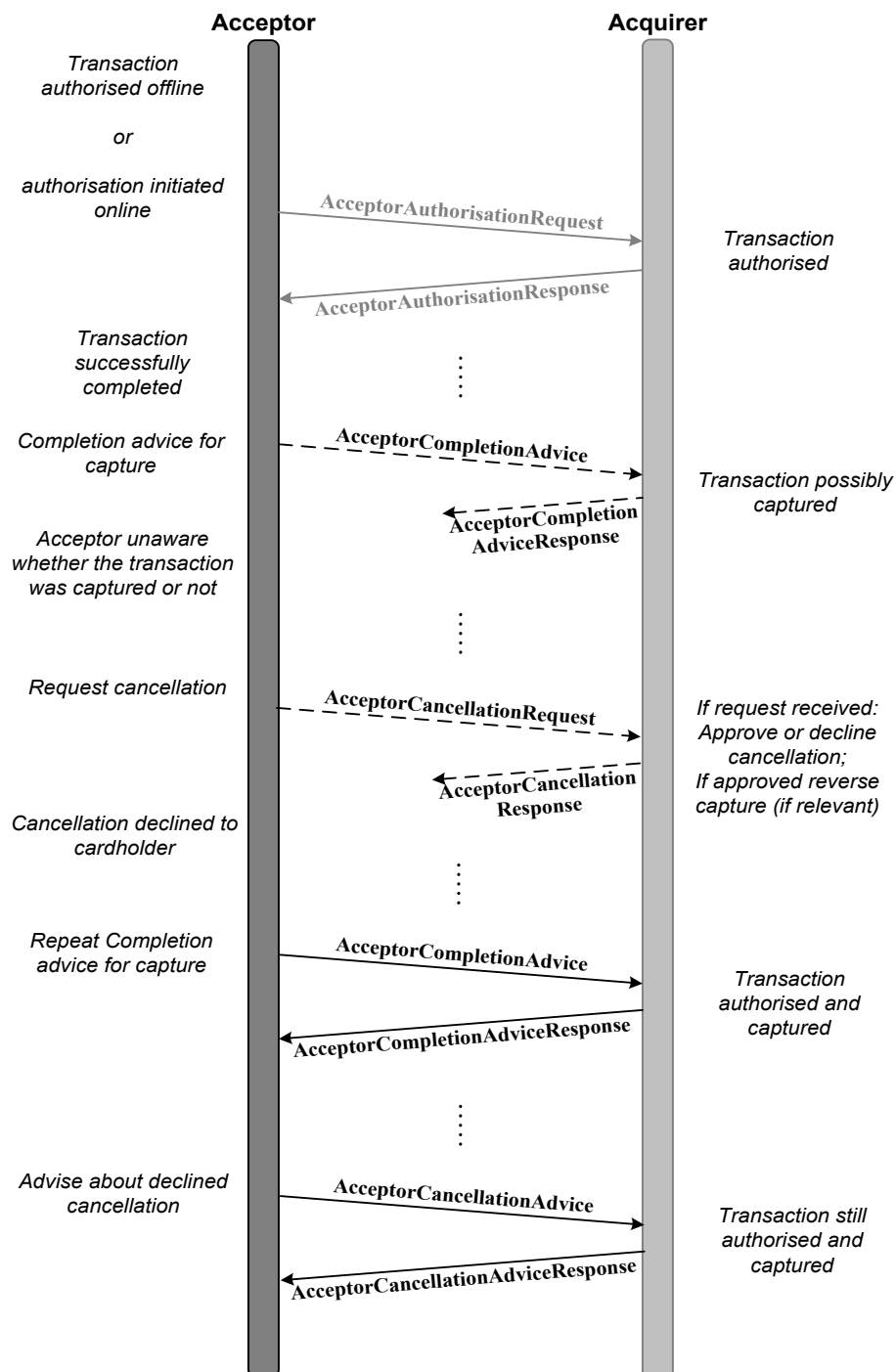


Figure 20: Cancellation declined after timeout

2.4.4.5 Cancellation successful after completion errors

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* to capture the transaction and
- the Acceptor didn't receive an *AcceptorCompletionAdviceResponse*.

The Acceptor wants to cancel the transaction and knows that the Acquirer did not clear the transaction yet.

The Acceptor sends an *AcceptorCancellationAdvice* (without sending a prior *AcceptorCancellationRequest*) to advise the Acquirer about the cancellation of the transaction.

The Acquirer accepts the cancellation:

- the Acquirer voids both the authorisation and financial capture associated to the transaction, if required
- the Acquirer sends an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the cancellation succeeded.

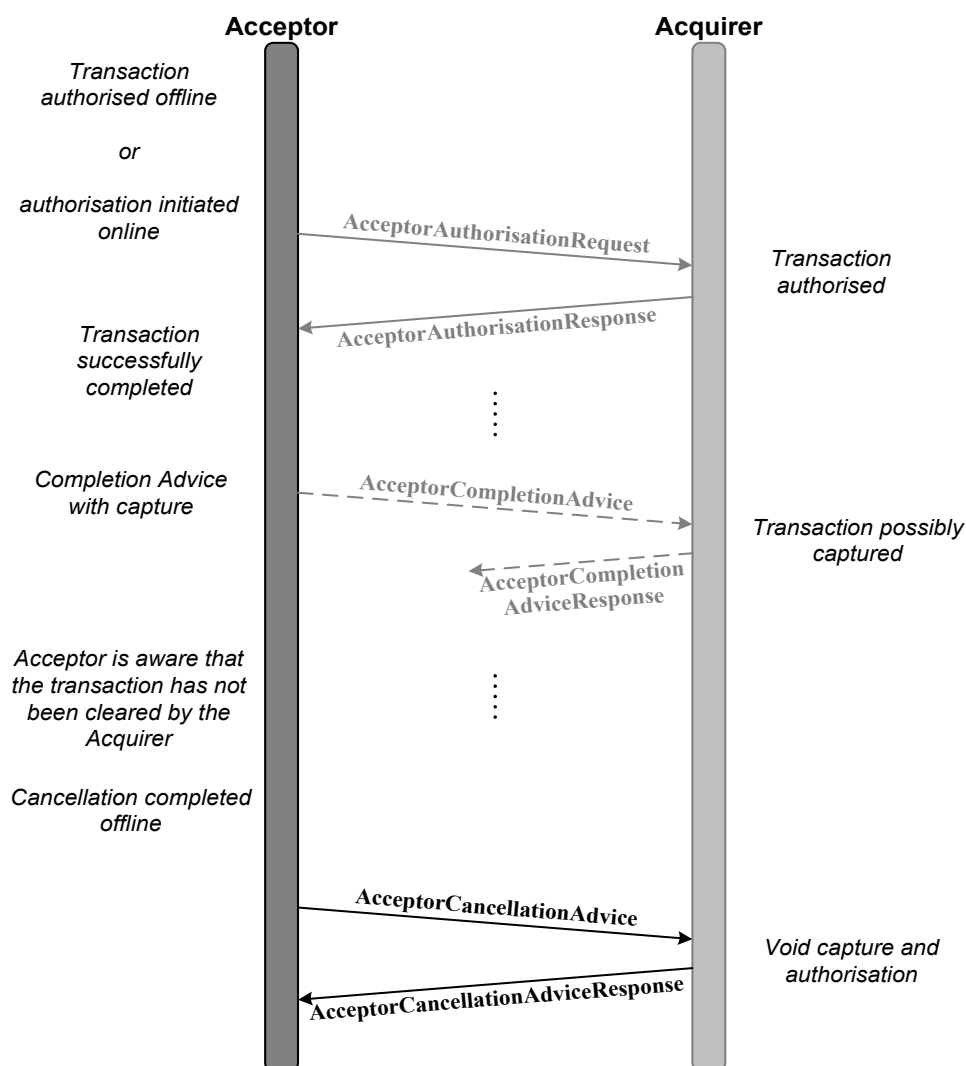


Figure 21: Cancellation successful after completion errors

2.4.4.6 Cancellation in batch mode

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* without capture and
- the Acceptor received an *AcceptorCompletionAdviceResponse*.

The Acceptor sends an *AcceptorBatchTransfer* to capture the transaction, but did not receive an *AcceptorBatchTransferResponse*.

The Acceptor wants to cancel the transaction but is unaware whether the Acquirer has already cleared the transaction or not.

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

The Acquirer accepts the cancellation:

- if needed, the Acquirer voids the financial capture associated with the transaction
- the Acquirer sends an *AcceptorCancellationResponse* to inform the Acceptor that the cancellation was possible.
- the Acceptor informs the cardholder that the cancellation was accepted
- the Acceptor sends back an *AcceptorCancellationAdvice* to the Acquirer about the successful cancellation of the transaction.
- the Acquirer voids the authorisation associated with the transaction
- the Acquirer sends an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the cancellation has been received.

The Acceptor sends a batch transfer depending on the configuration, with either:

- Include both the original debit (or credit) and cancellation transaction
- Remove the original debit (or credit) from the batch

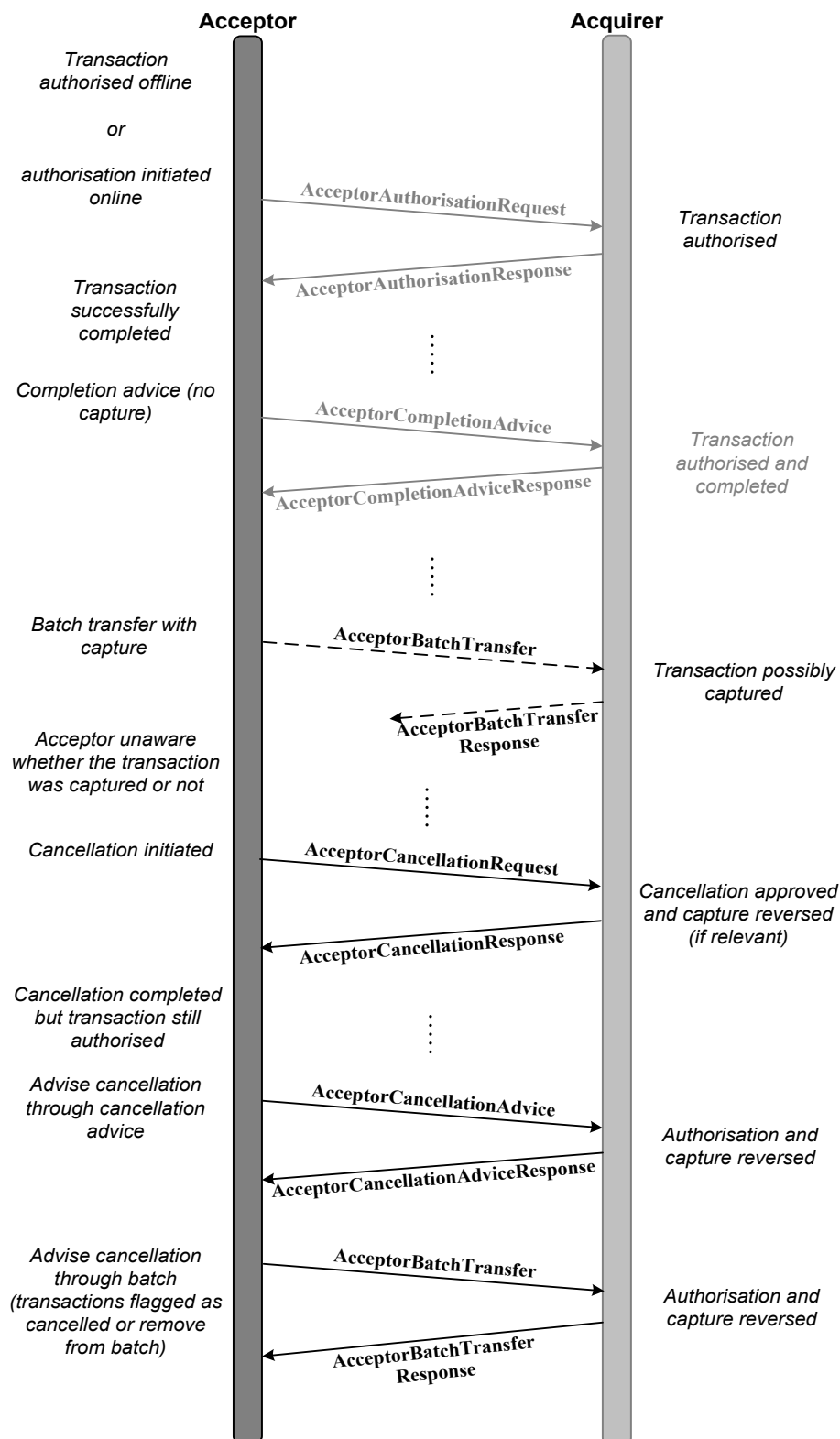


Figure 22: Cancellation in batch mode

2.4.4.7 Declined cancellation in batch mode

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* without capture and
- the Acceptor received an *AcceptorCompletionAdviceResponse*.

The Acceptor sends an *AcceptorBatchTransfer* to capture the transaction, but does not receive an *AcceptorBatchTransferResponse*.

The Acceptor wants to cancel the transaction but is unaware whether the Acquirer has already cleared the transaction or not.

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

The Acquirer declines the cancellation:

- the Acquirer sends an *AcceptorCancellationResponse* to inform the Acceptor that the cancellation was not possible.
- the Acceptor informs the cardholder that the cancellation was declined

The Acceptor has to resend the *AcceptorBatchTransfer* that failed. No *AcceptorCancellationAdvice* is sent by the Acceptor.

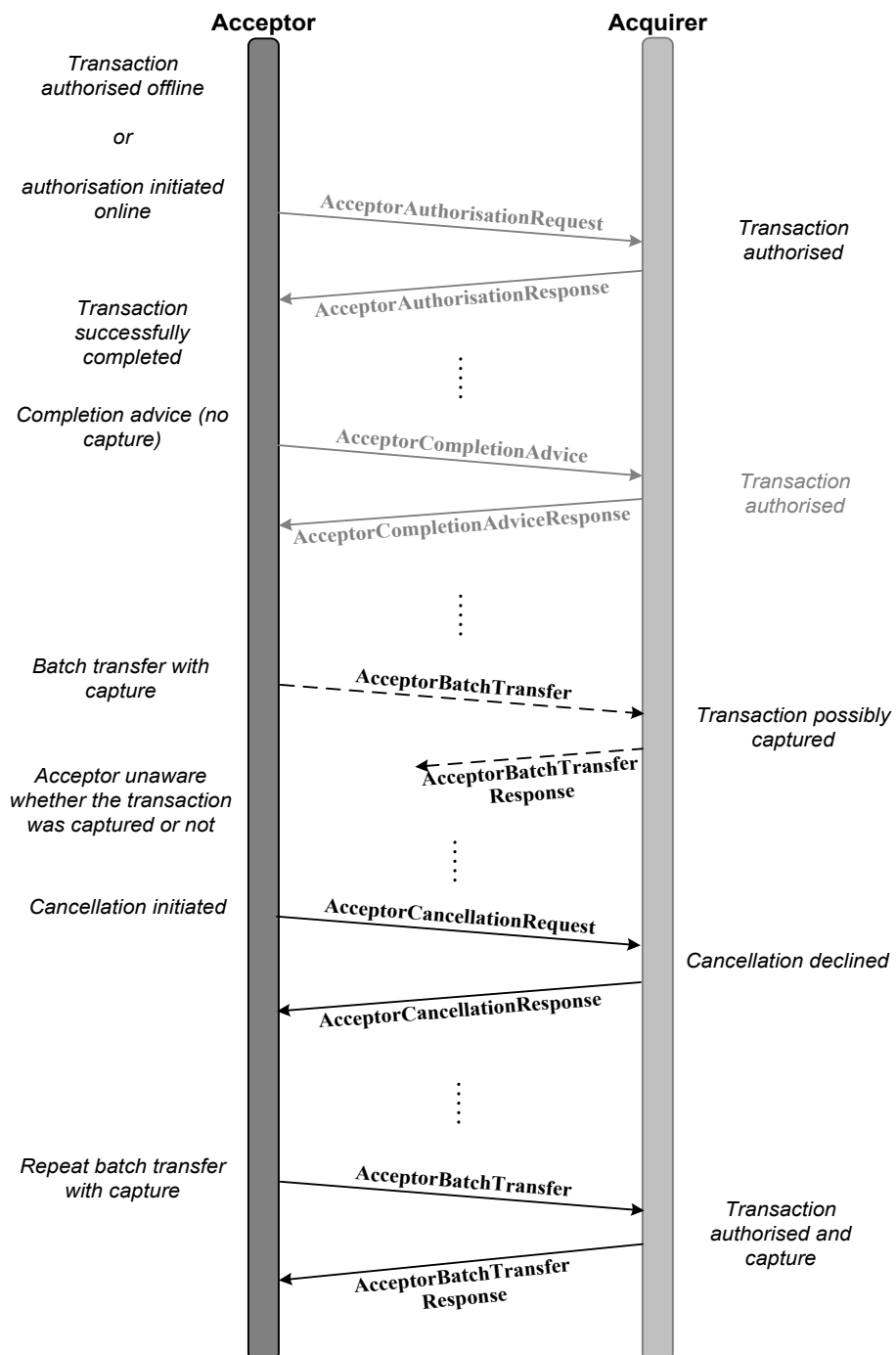


Figure 23: Declined cancellation in batch mode

2.4.4.8 Declined cancellation after timeout in batch mode and cancellation request

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* without capture and
- the Acceptor received an *AcceptorCompletionAdviceResponse*.

The Acceptor sends an *AcceptorBatchTransfer* to capture the transaction, but does not receive an *AcceptorBatchTransferResponse*.

The Acceptor wants to cancel the transaction but is unaware whether the Acquirer already has cleared the transaction or not.

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

The Acceptor does not received the response to the *AcceptorCancellationRequest* message

- the Acceptor has to decline the cancellation to the cardholder
- the Acceptor sends an *AcceptorCancellationAdvice* to inform the Acquirer about the declined cancellation of the transaction.
- the Acquirer sends an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the cancellation was declined.

Once the Acceptor has sent the *AcceptorCancellationAdvice*, it has to resend the *AcceptorBatchTransfer* for the financial capture by the Acquirer since the transaction was not cancelled.

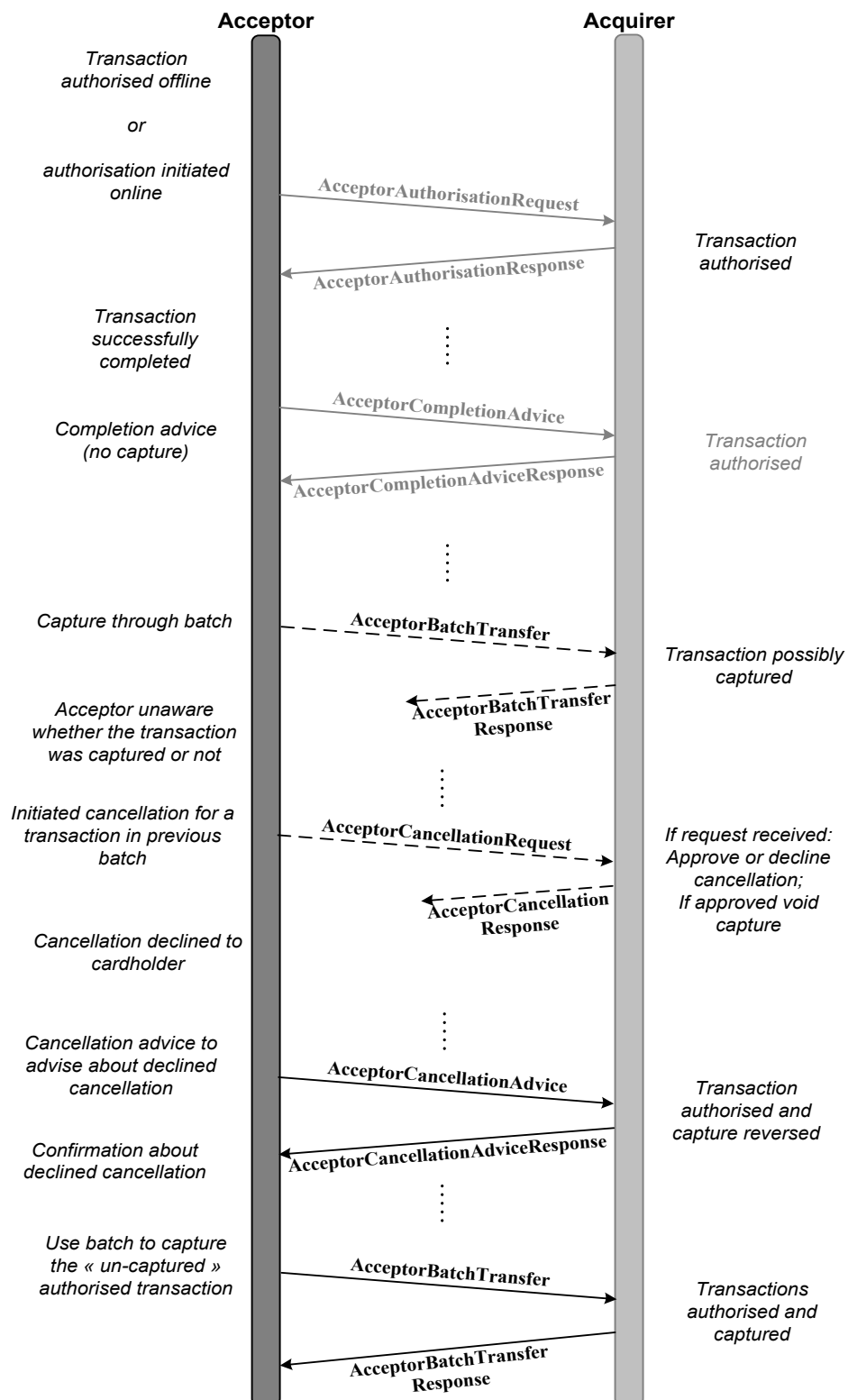


Figure 24: Declined cancellation after timeout in batch mode and cancellation request

2.4.4.9 Declined cancellation in batch mode after timeout in cancellation request

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* without capture and
- the Acceptor received an *AcceptorCompletionAdviceResponse*.

The Acceptor sends an *AcceptorBatchTransfer* to capture the transaction and he receives an *AcceptorBatchTransferResponse* to confirm the financial capture.

The Acceptor wants to cancel the transaction but is unaware whether the Acquirer has already cleared the transaction or not.

The Acceptor sends an *AcceptorCancellationRequest* to ask the Acquirer whether a cancellation is possible or not.

The Acceptor did not received a response to the *AcceptorCancellationRequest*:

- the Acceptor has to decline the cancellation to the cardholder
- the Acceptor sends an *AcceptorCancellationAdvice* to inform the Acquirer about the declined cancellation of the transaction.
- the Acquirer sends an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the cancellation was reversed.

Once the Acquirer has declined the cancellation, no *AcceptorBatchTransfer* needs to be exchanged since the Acquirer has already made the financial capture and the transaction was not cancelled.

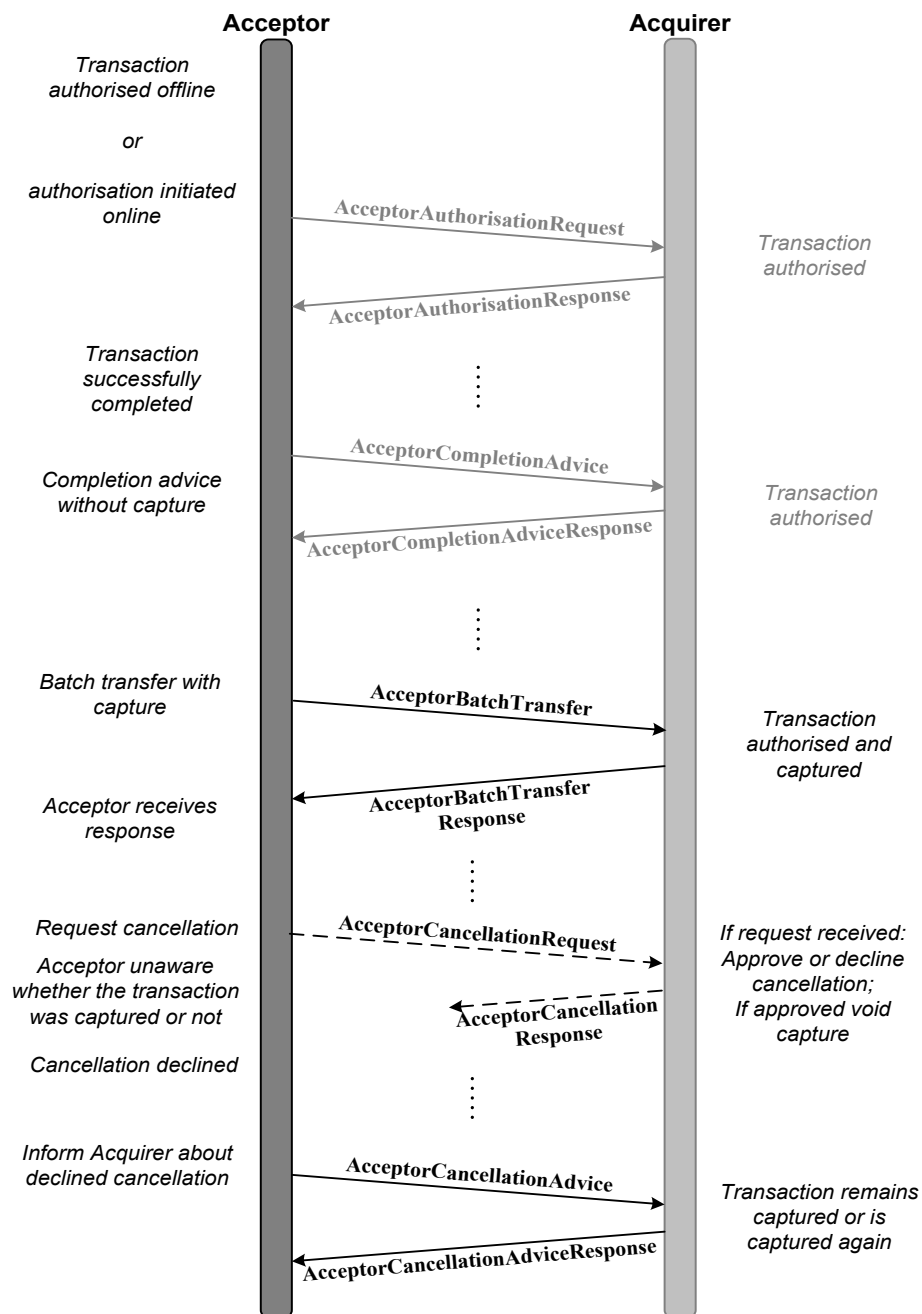


Figure 25: Declined cancellation in batch mode after timeout cancellation request

2.4.4.10 Cancellation in batch mode of a transaction captured but not yet cleared

In this scenario:

- the Acceptor sent an *AcceptorCompletionAdvice* without capture and
- the Acceptor received an *AcceptorCompletionAdviceResponse*.

The Acceptor sends an *AcceptorBatchTransfer* to capture the transaction, but does not receive an *AcceptorBatchTransferResponse*.

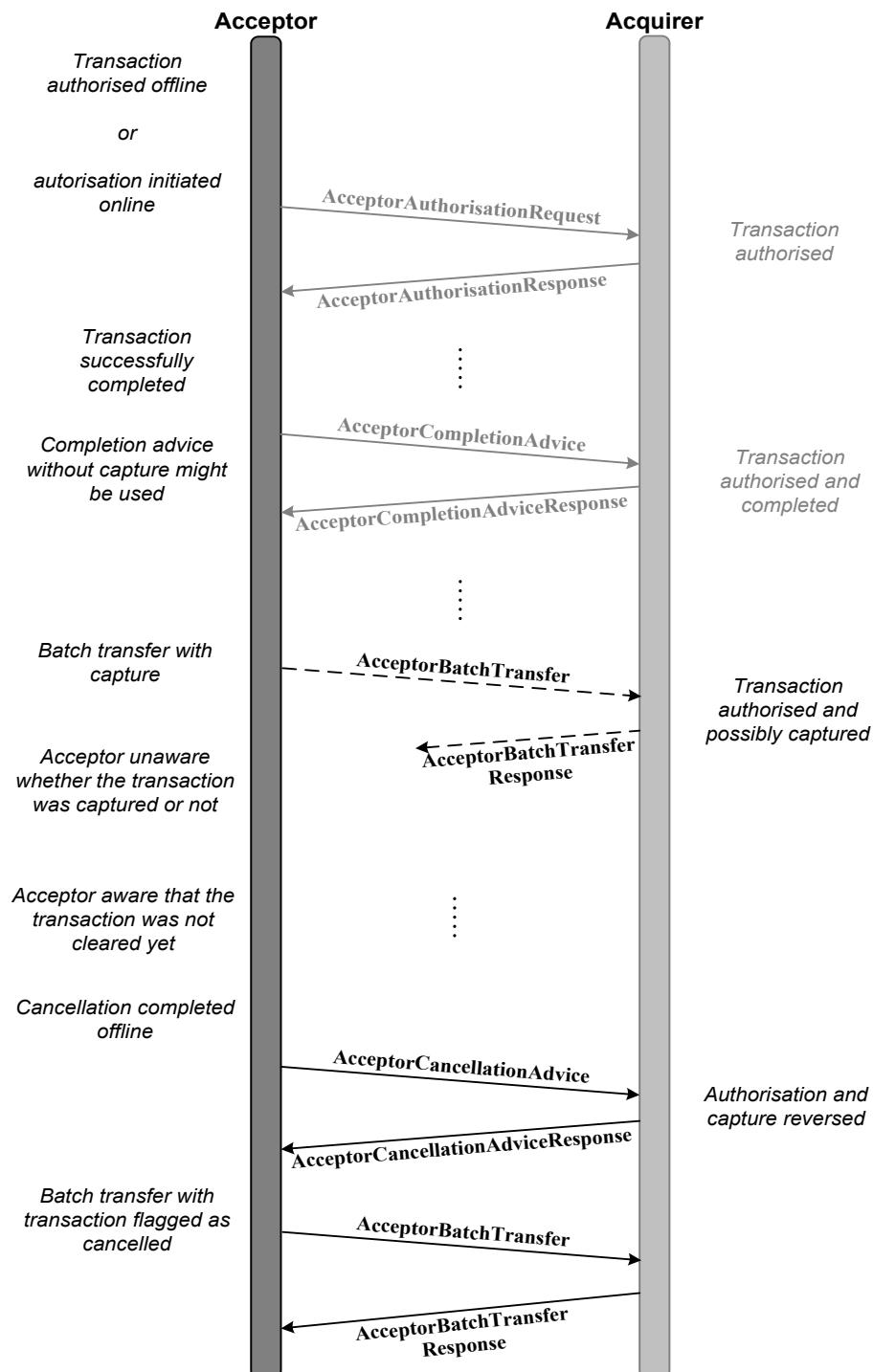
The Acceptor wants to cancel the transaction and is aware that the Acquirer has not cleared the transaction yet.

In that scenario:

- The Acceptor sends an *AcceptorCancellationAdvice* to inform the Acquirer about the cancellation of the transaction.
- The Acquirer voids the authorisation and financial capture of the transaction
- The Acquirer sends an *AcceptorCancellationAdviceResponse* to inform the Acceptor that the cancellation of the transaction succeeded.

The Acceptor sends a batch transfer depending on the configuration, with either:

- Include both the original debit (or credit) and cancellation transaction
- Remove the original debit (or credit) from the batch

Figure 26: Cancellation in batch mode of a transaction captured but not yet cleared

2.5 Batch

2.5.1 Introduction

Batch allows an Acceptor to send groups of transactions in a single message or a file to the Acquirer for financial capture.

A batch transfer exchange is composed of an *AcceptorBatchTransfer* (*caaa.011.001.06*) and an *AcceptorBatchTransferResponse* (*caaa.012.001.06*) used to acknowledge or to reject some or all transactions contained in a batch.

Towards the Acquirer

The *AcceptorBatchTransfer* to be uploaded contains locally stored offline and online transactions.

In addition to financial transactions (payments, refunds, etc.) non-financial transactions (incomplete, declined, etc.) may be added to an *AcceptorBatchTransfer*.

TransactionTotals are part of the file to enable reconciliation.

Message or File integrity is ensured through a security trailer.

Towards the Acceptor

The Acquirer informs the acceptor about the validation of *AcceptorBatchTransfer* content by sending an *AcceptorBatchTransferResponse* message.

The Acceptor retains the stored transactions which were previously sent to the Acquirer until receiving an *AcceptorBatchTransferResponse* message from the Acquirer.

The *AcceptorBatchTransferResponse* provides status information about the data received.

The Acquirer informs the Acceptor about the transfer of data.

Once the Acceptor has received this notification, from a protocol perspective, the Acceptor is freed of any technical responsibility to keep the data .

2.5.2 Data Organisation

2.5.2.1 AcceptorBatchTransfer

The upload of the data to be captured is initiated by an Acceptor using either a file transfer exchange or exchanges of messages.

Data to be sent is organised into sets. An *AcceptorBatchTransfer* contains one or more sets.

A *DataSet* contains one or more financial and/or non-financial transactions. However it is possible that a data set contains no transactions.

Data common to all transactions of a set may be factorised and sent prior to the transactions (e.g. Acquirer data, POI data...)

The content of the set and its organisation depend on the POI configuration.

Its structure is represented as follows:

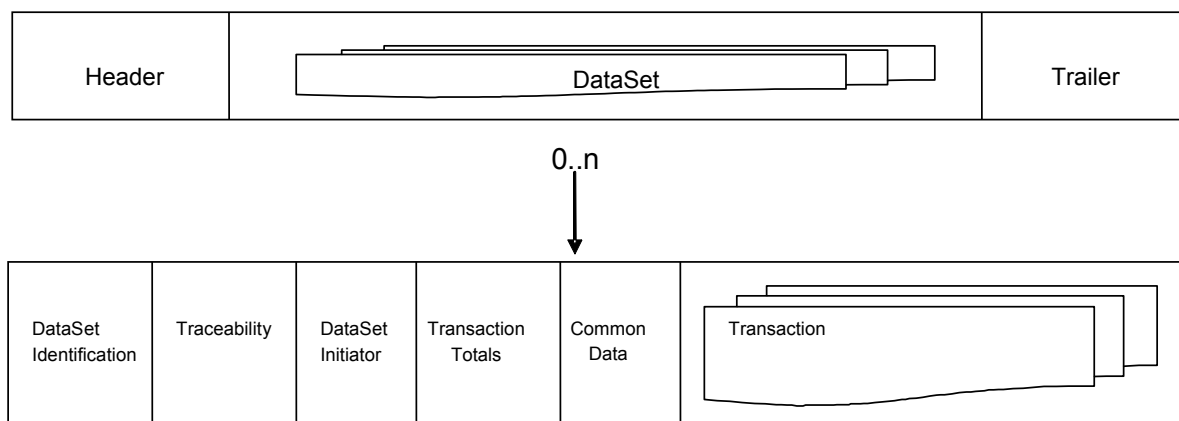


Figure 27: BatchTransfer structure

2.5.2.2 AcceptorBatchTransferResponse

The Acquirer validates an *AcceptorBatchTransfer* with an *AcceptorBatchTransferResponse* according to the following process:

- The Acquirer sends back an *AcceptorBatchTransferResponse* to the Acceptor in case of a message exchange, or
- The Acceptor downloads an *AcceptorBatchTransferResponse* file.

The *AcceptorBatchTransferResponse* contains the response from the Acquirer for each *DataSet* that has been processed. Any individual transactions that have been rejected by the Acquirer will be

contained within the appropriate DataSet response. An Acquirer can only send a DataSet in a *AcceptorBatchTransferResponse* when all transactions in that DataSet have been processed.

If a set is partially approved, the element “DataSet” is present. It contains the details of rejected transactions.

The content of the DataSet and its organisation depend on POI configuration.

File structure may be represented as follows:

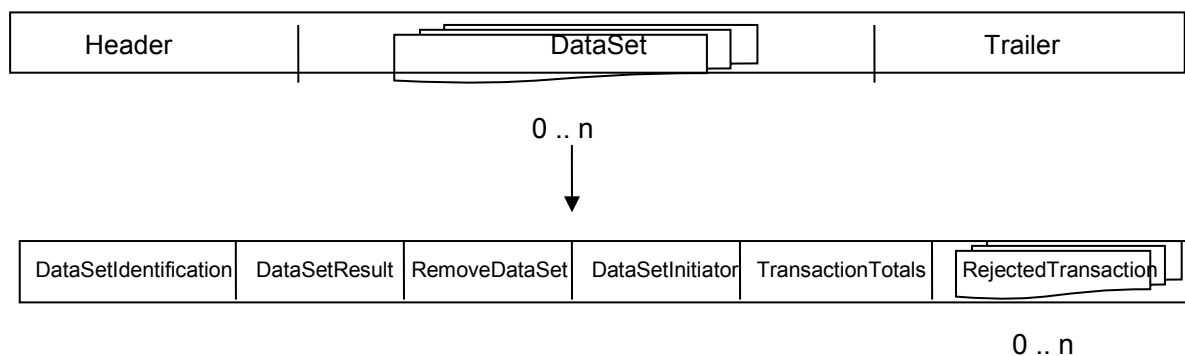


Figure 28: BatchTransferResponse structure

2.5.2.3 Data Factorisation

To prevent file size problems, some data of an *AcceptorBatchTransfer* may be factorised to reduce the occurrences of repetitive data.

The Acquirer must support factorisation. The Acceptor may choose whether to perform factorisation or not.

Factorisation is limited to the following data:

- Acquirer
- Merchant
- POI
- PaymentContext
- SaleContext
- TransactionType
- AdditionalService
- ServiceAttribute
- MerchantCategoryCode
- Currency

All transaction components within a DataSet will inherit data from the values held in CommonData unless different data is held in the transaction component itself.

The Acquirer sends an *AcceptorBatchTransferResponse* with the same DataSet factorisation as in the related *AcceptorBatchTransfer*.

2.5.2.4 Multiplicity

Data elements which are mandatory in an *AcceptorCompletionAdvice* message are de facto mandatory in an *AcceptorBatchTransfer*. Some of the elements may be factorised in *CommonData* and in that case need not appear in the *Transaction* components but they must all be present in one place or the other.

2.5.3 Types of Batch Transfer

A batch is generally used as a collection of transactions to be sent to an Acquirer for processing.

2.5.3.1 Batch containing Completion and Cancellation Transactions

An *AcceptorBatchTransfer* is sent by an Acceptor to an Acquirer. It contains a collection of completion and cancellation transactions.

Completion transactions which may be sent as:

- Successful debit transactions:
 - “*TransactionType*” with the value “*CardPayment*”; “*CashBack*” or “*DeferredPayment*” and
 - “*TransactionSuccess*” with the value “*True*”
- Successful credit transactions:
 - “*TransactionType*” with the value “*Refund*” and
 - “*TransactionSuccess*” with the value “*True*”
- Failed or declined transactions:
 - “*TransactionSuccess*” with the value “*False*”;

Cancelled transactions are sent with “*TransactionSuccess*” having the value “*True*”

Depending on a TMS configuration, the following types of transactions can be present or not :

- Debit and Credit transaction
- Failed transactions
- Declined transactions
- Cancelled transactions

An *AcceptorBatchTransferResponse* message is sent by the Acquirer to the Acceptor to confirm the proper acknowledgement of the card payment transactions by the Acquirer.

The *AcceptorBatchTransferResponse* file contains the response from the Acquirer for each DataSet that has been processed. Any individual transactions that have been rejected by the Acquirer must be contained within the appropriate DataSet response.

An Acquirer can only send a DataSet in a *BatchTransferResponse* when all transactions in that DataSet have been processed.

Responses to one *AcceptorBatchTransfer* can be sent in several *AcceptorBatchTransferResponse*.

Responses for several *AcceptorBatchTransfer* can be sent in one *AcceptorBatchTransferResponse*. (See section 4.6 Batch.)

The acquirer has to give a response in one DataSet for all the transactions contained in a DataSet received and only for these transactions.

2.5.3.2 Batch containing Authorisation Transactions

An AcceptorBatchTransfer is sent

- by an Acceptor to an Acquirer containing a collection of authorisation requests
- by an Acquirer to an Acceptor containing a collection of authorisation responses only.

The use of authorisations in BatchTransfer is not configurable but depends on a bilateral agreement between an Acceptor and Acquirers involved.

In the case of batch transferred authorisations:

- A DataSet in the batch sent from an Acceptor to an Acquirer contains only AuthorisationRequests,
- The batch sent from an Acquirer to an Acceptor contains only AuthorisationResponses.

A batch of authorisations may contain:

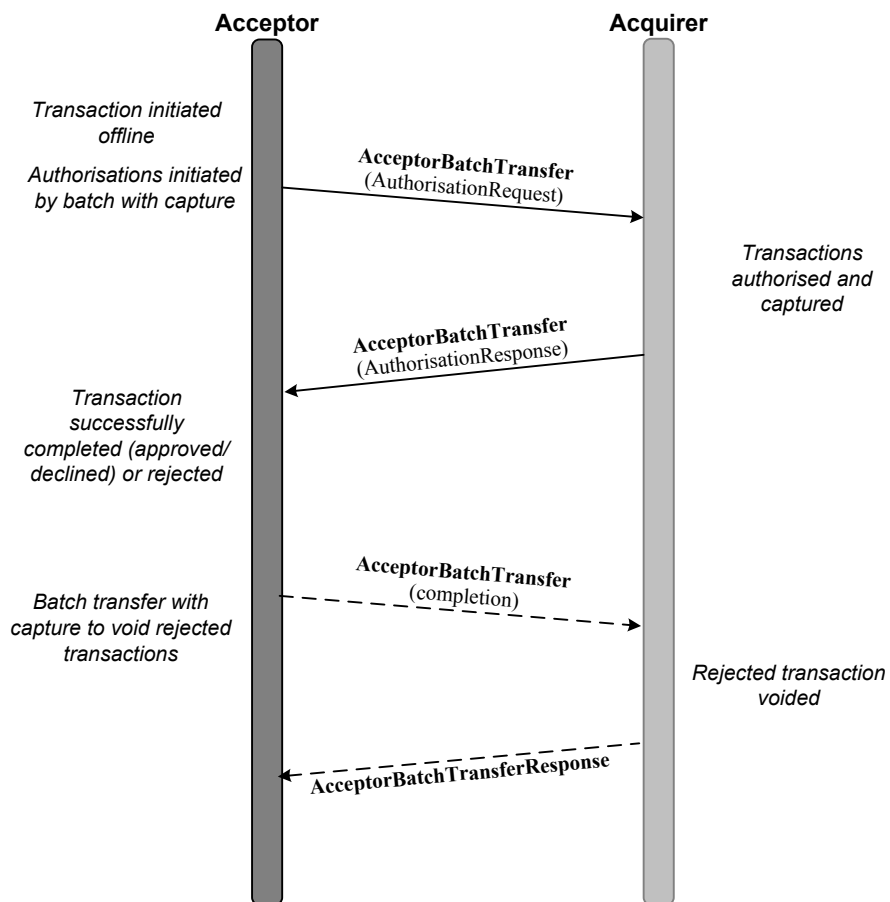
- Debit transactions : AuthorisationRequest occurrences with "transactionType" data element value equal to "CardPayment";
- Credit transactions : AuthorisationRequest occurrences with "transactionType" data element value equal to "Refund";
- Debit transactions : AuthorisationResponse occurrences;
- Credit transactions : AuthorisationResponse occurrences .

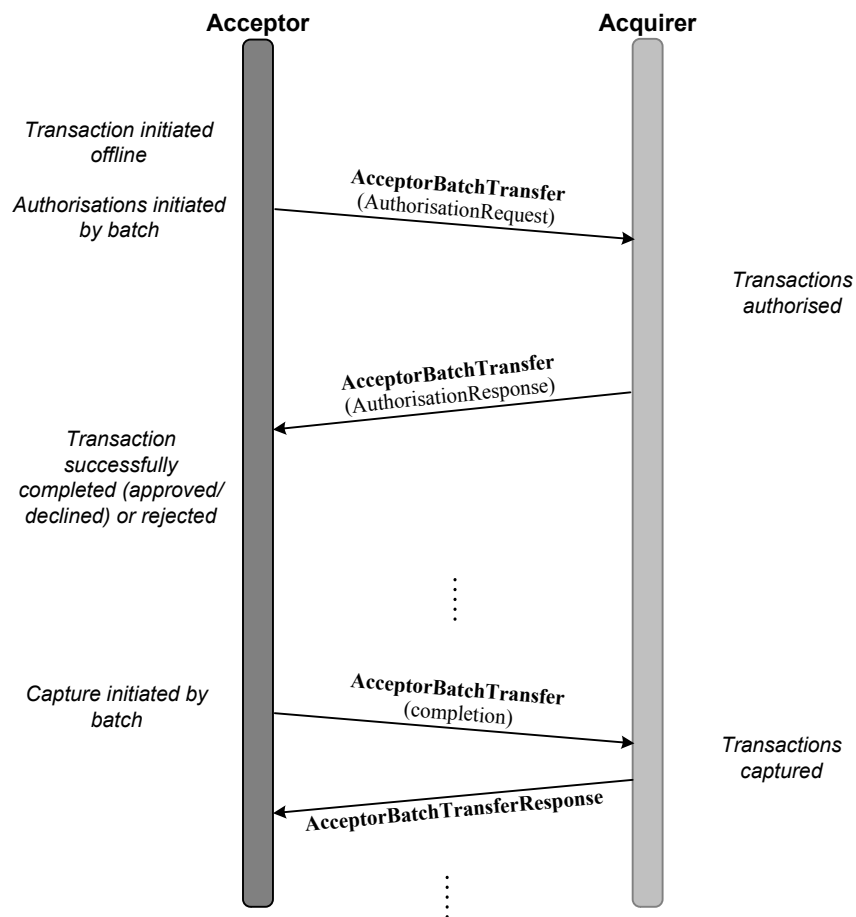
The rejected AuthorisationRequests are sent in a related AuthorisationResponse with *Response* set to "TechnicalError". The *BatchTransferResponse* is not used for acknowledging AuthorisationRequests. In this case, a completion can be sent to nullify the effect of the authorisation request.

A dataset containing AuthorisationRequest messages can be answered with more than one dataset. In this case, all the rejected transactions must be present in the first dataset containing responses. Responses contained in a dataset must refer to requests contained in only one dataset

The acquirer can require to receive completions in the batch for previously authorised online transactions and / or authorised offline transactions.

- transactions authorised online are configured to be captured in batch (e.g. the TMS parameter AcquirerProtocolParameters.OnlineTransaction.FinancialCapture is equal to *Batch*)
- transactions authorised offline are configured to be captured in batch (e.g. the TMS parameter AcquirerProtocolParameters.OfflineTransaction.FinancialCapture is equal to *Batch*)

**Figure 29: Batch containing Financial Authorisations**

**Figure 30: Batch containing Authorisations**

2.6 Reconciliation Process

2.6.1 Introduction

Reconciliation is the process of performing checks and balances of transactions previously captured. This process is carried out between an Acceptor and an Acquirer for a given reconciliation period.

An Acceptor initiates a reconciliation exchange to ensure that the debits and credits match the computed balances by the Acquirer and performed during the same reconciliation period.

An *AcceptorReconciliationRequest* message (*caaa.009.001.06*) is sent by the Acceptor to inform the Acquirer about the totals accumulated during the reconciliation period.

An *AcceptorReconciliationResponse* message (*caaa.010.001.05*) is returned by the Acquirer to inform the Acceptor about the totals accumulated during the reconciliation period.

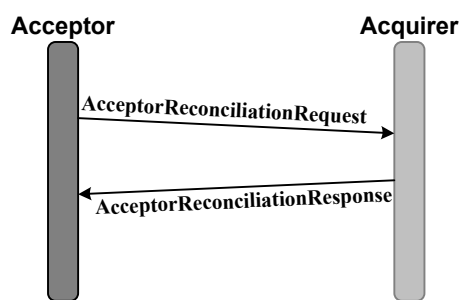


Figure 31: Reconciliation Exchange

Should the Acceptor or the Acquirer detect a difference in totals, the discrepancy must then be resolved by other means and is outside the scope of this protocol.

Reconciliation is not mandatory and reconciliation messages are never exchanged when the configuration parameter *AcquirerProtocolParameters.ReconciliationExchange.ExchangePolicy* defined in the *AcceptorConfigurationUpdate* TMS message has the value *None* or is absent.

2.6.2 Reconciliation Period Identification

If reconciliation between an Acceptor and an Acquirer is required, each transaction belongs to one and only one reconciliation period identified by the message element *Transaction.ReconciliationIdentification*.

Assignment of transactions to a reconciliation period can be made by either the Acceptor or the Acquirer depending on the TMS configuration parameter flag *ReconciliationByAcquirer*.

If the reconciliation period assignment is under the control of the Acceptor (*ReconciliationByAcquirer* parameter is *False*), the reconciliation period is identified by the message element *Transaction.ReconciliationIdentification* in the *AcceptorAuthorisationRequest*, *AcceptorCompletionAdvice*, *AcceptorCancellationRequest* and *AcceptorCancellationAdvice* messages.

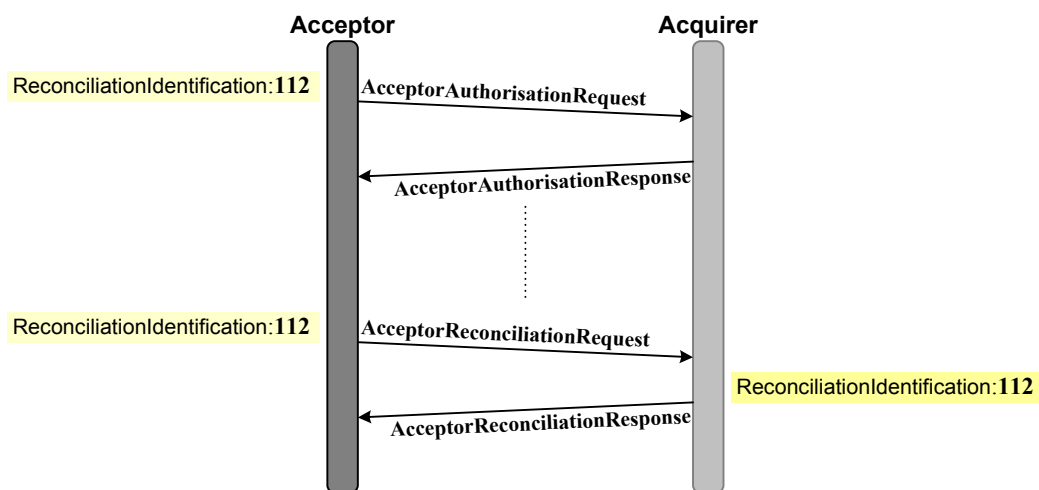


Figure 32: Reconciliation Period Assigned by the Acceptor

During the reconciliation exchange, the reconciliation period is identified in the message element *Transaction.ReconciliationIdentification* of *AcceptorReconciliationRequest* and *AcceptorReconciliationResponse*.

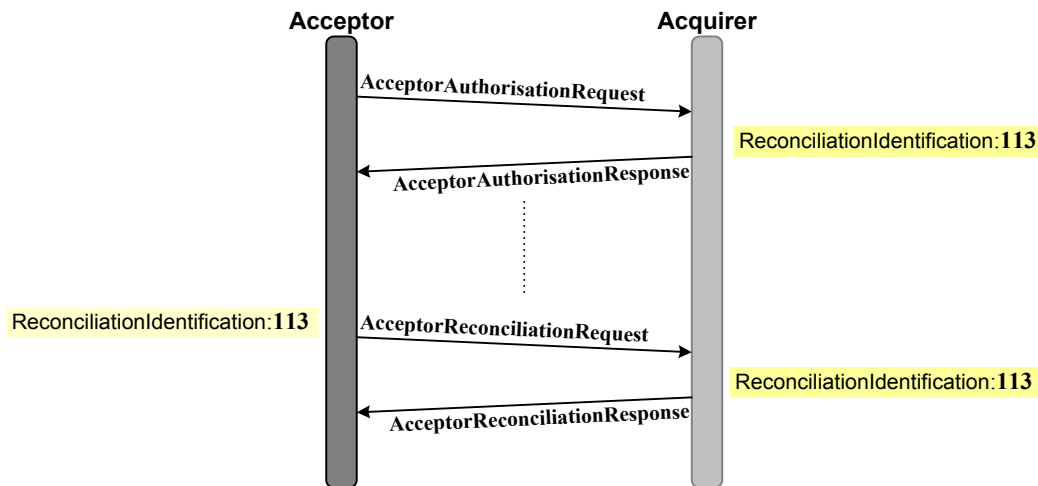


Figure 33: Reconciliation Period Assigned by the Acquirer

The reconciliation period of the transaction is identified by the *ReconciliationIdentification* at the time of the transaction capture. If the capture is performed during the completion, the *AcceptorCompletionAdvice* may have a *ReconciliationIdentification* value different from the Authorisation.

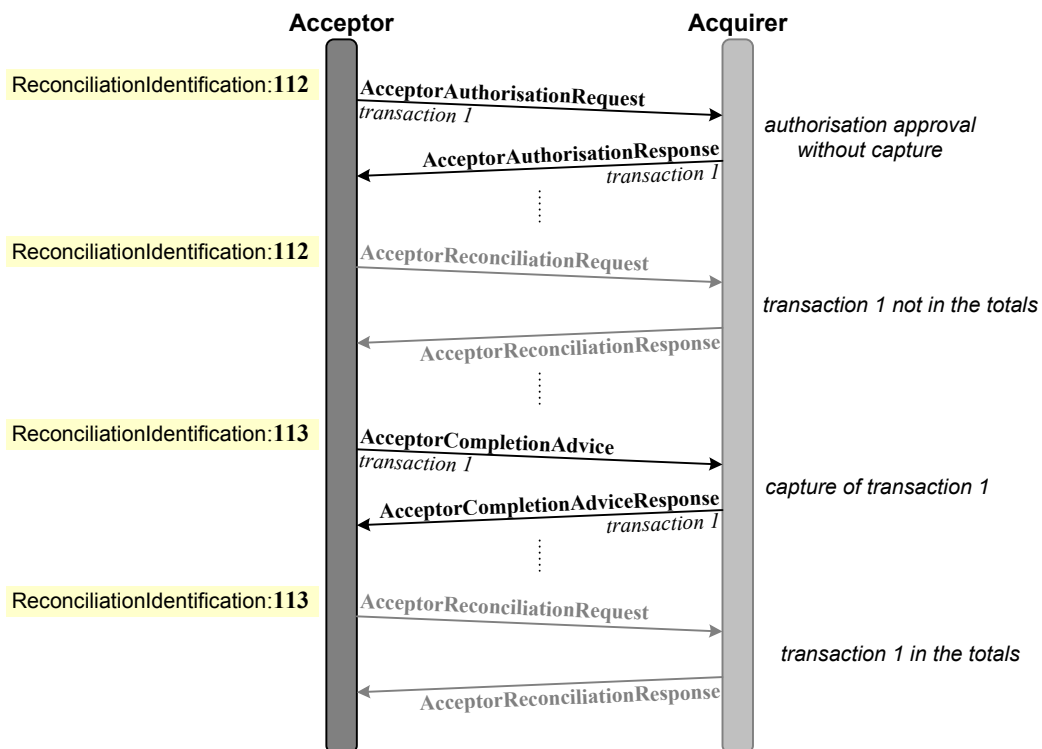


Figure 34: Reconciliation between the Authorisation and the Completion

When the flow of transaction cannot be stopped for closing the reconciliation at the acquirer, before sending the *AcceptorReconciliationRequest* message, the Acceptor must:

- Assign the new transactions to the next reconciliation period, and
- Wait for the completion of the transactions of the reconciliation period to close before sending the *AcceptorReconciliationRequest* message.

Successive reconciliation periods are then interleaved.

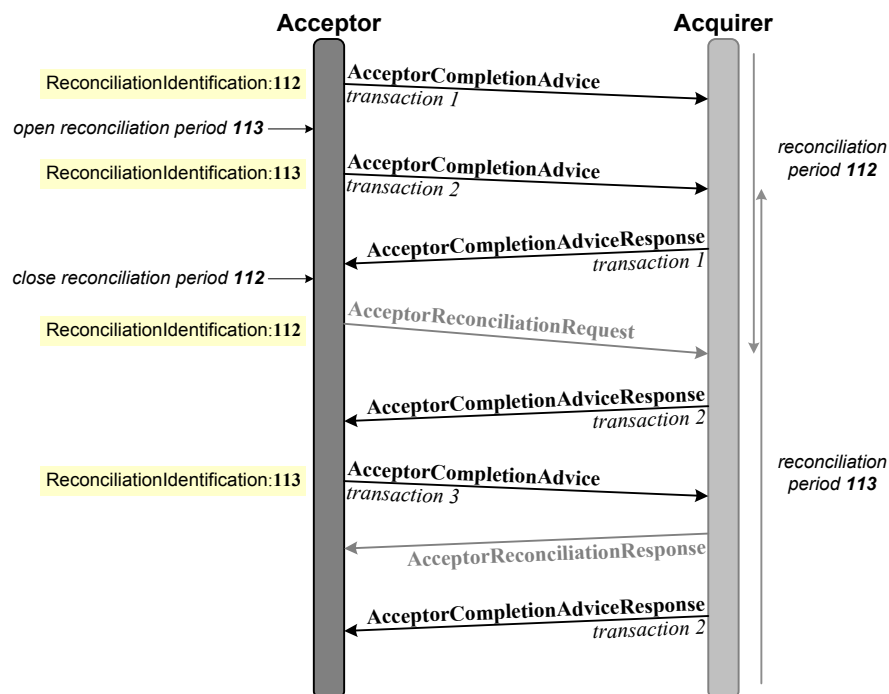


Figure 35: Overlapping of Reconciliation Periods

2.6.3 Transaction Totals

Transaction totals are calculated from transactions completed and captured during the reconciliation period.

Before initiating a reconciliation exchange, the Acceptor must:

- 1) Perform the capture of all the transactions integrated in the totals of the reconciliation period,
- 2) Inform the Acquirer about all the online cancelled transactions of the reconciliation period
- 3) Inform the Acquirer about all the offline cancelled transactions, if these are integrated in the totals of the reconciliation period, and
- 4) Inform the Acquirer about all the reversed online transactions of the reconciliation period.

Transaction totals may be split in a sequence of repeated *Transaction.TransactionTotals* message elements, according to the following criteria:

- *TransactionTotals.Type*: the type of transaction, which is mandatory,
- *TransactionTotals.POIGroupIdentification*: the grouping of transactions made by the Acceptor, when the two following conditions are fulfilled:
 - the acceptor is configured to split totals (TMS configuration parameter *AcquirerProtocolParameters.SplitTotals* is *True*), and
 - the message element *POI.GroupIdentification* is present in *AcceptorAuthorisationRequest*, *AcceptorCompletionAdvice*, *AcceptorCancellationRequest* or *AcceptorCancellationAdvice* messages.

The transactions where *POI.GroupIdentification* is absent have to be combined in instances of *TransactionTotals* without *POIGroupIdentification*.

- *TransactionTotals.CardProductProfile*: the card profile, when the two following conditions are fulfilled:
 - the acceptor is configured to split totals (TMS configuration parameter *AcquirerProtocolParameters.SplitTotals* is *True*), and
 - the message element *Card.CardProductProfile* is present in the *AcceptorAuthorisationRequest*, *AcceptorCompletionAdvice*, *AcceptorCancellationRequest* or *AcceptorCancellationAdvice* messages.

The transactions where *CardProductProfile* is absent have to be combined in instances of *TransactionTotals* without *CardProductProfile*.

- *TransactionTotals.Currency*: the currency of the transaction, when the acceptor is configured to split totals per currency (TMS configuration parameter *AcquirerProtocolParameters.TotalsPerCurrency* is *True*). When *TotalsPerCurrency* is *False*, totals are computed whatever the currency and the *TransactionTotals.Currency* message element is absent.

The Totals of declined and failed transactions are included in the *ReconciliationRequest*. A Reconciliation is never rejected because of differences with declined and failed transaction totals.

If a cancellation is allowed after the reconciliation of the original transaction, the original transaction is considered as a Debit or Credit for the first reconciliation period, and the cancelled transaction considered as a DebitReverse or a CreditReverse respectively for the second reconciliation period.

If a transaction can be cancelled on a different POI terminal from the POI terminal where the original transaction has been performed, and the totals are accumulated per *POIGroupIdentification*, the

cancellation (DebitReverse or CreditReverse) is counted in the totals of the *POIGroupIdentification* related to the original transaction, even if the Cancellation transaction belongs to another reconciliation period.

A difference in totals between the Acceptor and the Acquirer may occur after reaching the maximum number of retransmissions of an *AcceptorCompletionAdvice* (or *AcceptorCancellationAdvice*) without a positive *AcceptorCompletionAdviceResponse* (or *AcceptorCancellationAdviceResponse*). In this case, it is not possible for the Acceptor to determine the knowledge of the Acquirer about the outcome of the transaction because:

- The Acquirer may have not received or understood the repeated *AcceptorCompletionAdvice* (or *AcceptorCancellationAdvice*) messages,
- The Acquirer may have received and understood an *AcceptorCompletionAdvice* (or *AcceptorCancellationAdvice*) message, but the Acceptor has not received or has not understood the *AcceptorCompletionAdviceResponse* (or *AcceptorCancellationAdviceResponse*).

2.6.4 Reconciliation Exchange

The configuration parameter *AcquirerProtocolParameters.ReconciliationExchange.ExchangePolicy* defined in the *AcceptorConfigurationUpdate* TMS message determines whether an *AcceptorReconciliationRequest* message may be used or not.

The Policy component of this configuration parameter can contain the following values:

- *Cyclic*: an *AcceptorReconciliationRequest* message is sent periodically according to the timing conditions defined in the *AcquirerProtocolParameters.ReconciliationExchange.TimeCondition* component.
- *NumberLimit*: an *AcceptorReconciliationRequest* message is sent as soon as the sum of all *TransactionTotals.TotalNumber* of the current reconciliation period reaches the value configured in the *AcquirerProtocolParameters.ReconciliationExchange.MaximumNumber* component.
- *TotalLimit*: an *AcceptorReconciliationRequest* message is sent as soon as the sum of all *TransactionTotals.CumulativeAmount* of the current reconciliation period reaches or exceeds the value configured in the *AcquirerProtocolParameters.ReconciliationExchange.MaximumAmount* component.
- *OnDemand*: an *AcceptorReconciliationRequest* message is sent when requested by the Acceptor.
- *None*: the Acceptor never sends *AcceptorReconciliationRequest* messages.

The component *Transaction.ClosePeriod* may request the closure of a reconciliation period. This request is useful when the Acquirer assigns a reconciliation period identification to transactions.

The response to an *AcceptorReconciliationRequest* message is either an *AcceptorRejection* message, or an *AcceptorReconciliationResponse* message with *Response* = "Approved" or "Declined".

If there is no transactions in the reconciliation period, the *AcceptorReconciliationResponse* message must be sent with *Response* = "Approved".

It is possible that the verification of the totals cannot be performed in real-time before sending the response to the reconciliation. In this case, the *AcceptorReconciliationResponse* message must contain:

- *Response* = "Approved".
- *ResponseReason* = "Totals Unavailable".
- *TransactionTotals* must be absent.

If the totals are performed in real-time by the Acquirer, and some totals are different from the totals send by the Acceptor, the *AcceptorReconciliationResponse* message must contain:

- *Response* = "Declined".
- *ResponseReason* = "Difference in Totals".
- *TransactionTotals* must be present, as computed by the Acquirer.

Whatever the result of the reconciliation (*Response* = "Approved" or "Declined"), a new reconciliation period has to be started to perform new transactions. Any difference or discrepancy has to be resolved outside the protocol.

If the Acceptor did not receive an acceptable response to an *AcceptorReconciliationRequest*, it may send a new *AcceptorReconciliationRequest* message, with a copy of the body *ReconciliationRequest*, and a new value of *ExchangeIdentification* and *CreationDateTime* in the header.

It is recommended to stop exchanging transactions during a reconciliation exchange.

If it is not possible:

- the Acceptor assigns a new reconciliation period for the transactions that may follow,
- the Acceptor waits until the end of the transaction of the current reconciliation period,
- the Acceptor sends the required *AcceptorCompletionAdvice* (or *AcceptorCancellationAdvice*) messages (see section 2.6.3)
- the Acceptor starts the reconciliation exchange.

2.7 Diagnostic Messages

A diagnostic exchange is composed of an *AcceptorDiagnosticRequest* (caaa.013.001.06) message and an *AcceptorDiagnosticResponse* (caaa.014.001.05) message.

An *AcceptorDiagnosticRequest* is a message sent by an *InitiatingParty* to a *RecipientParty* to check the availability, the security or the configuration of the dialogue with the *RecipientParty*.

The Diagnostic message has been designed to perform specific administrative tasks without relying on “dummy” messages.

An *AcceptorDiagnosticResponse* message is sent back by the *RecipientParty* to the *InitiatingParty* to confirm the availability of the *RecipientParty*.

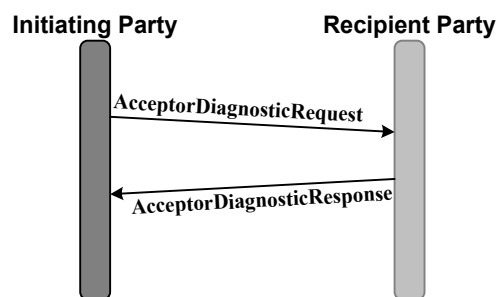


Figure 36: Diagnostic Exchange

An *AcceptorDiagnosticRequest* message is used to:

- Confirm the identification of the partners of the exchanges,
- Ensure that security on both sides of the dialogue is synchronised,
- Test the communication with a *RecipientParty* ,
- Endorse the version of the configuration parameters.

If the *AcceptorDiagnosticRequest* message is received without errors by the *RecipientParty*, an *AcceptorDiagnosticResponse* message is sent back by the *RecipientParty* to the *InitiatingParty*.

The *RecipientParty* uses an *AcceptorDiagnosticResponse* message to request the *InitiatingParty* to notify the *RecipientParty* about any maintenance operations that may be required.

If the *RecipientParty* receives an *AcceptorDiagnosticRequest* message with errors or cannot process the message, an *AcceptorRejection* message is returned to the *InitiatingParty* with the appropriate *RejectReason*:

- *UnableToProcess*: the *RecipientParty* cannot process messages during a temporary period.
- *InvalidMessage*: Invalid envelope of the message
- *ParsingError*: Problem of format, absence of element, content of an element, etc.
- *Security*: Security error such as an invalid key or MAC
- *InitiatingParty, RecipientParty*: identification of the *InitiatingParty* or the *RecipientParty* is invalid.
- *DuplicateMessage*: The message is a duplicate message for a given *InitiatingParty* and a given *RecipientParty* when the following fields have the same value:
 - Header.ExchangeIdentification,
 - CreationDateTime and
 - HeaderRetransmissionCounter if present.
- *ProtocolVersion*: The *RecipientParty* cannot support the version of the protocol contained in *ProtocolVersion*.

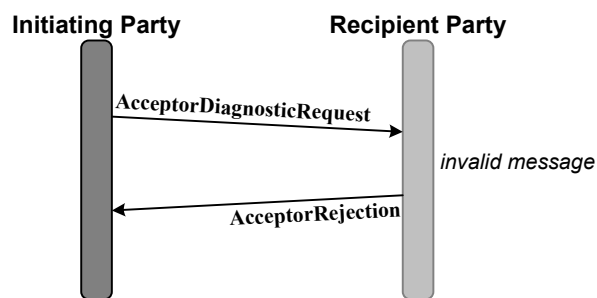


Figure 37: Diagnostic Reject

Should an *InitiatingParty* send an *AcceptorDiagnosticRequest* to an Agent, the Agent returns an *AcceptorDiagnosticResponse* to the *InitiatingParty*.

An *AcceptorDiagnosticRequest* message does not contain any information that may allow an Agent to further route the message to another Agent or *RecipientParty*.

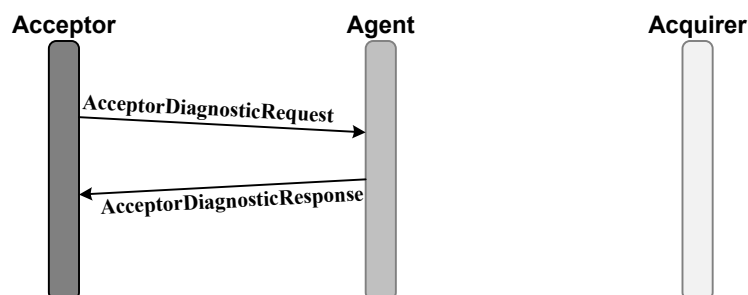


Figure 38: Diagnostic Request from an Acceptor to an Agent

An Agent must never forward an *AcceptorDiagnosticRequest* message.

An Acceptor must only use an *AcceptorDiagnosticRequest* message under exceptional situations to avoid any potential risk of message congestion (e.g. after the installation of a payment application or during an update of electronic keys).

An Intermediary Agent uses a diagnostic exchange to check the availability of the communication with an Acquirer or another Intermediary Agent.

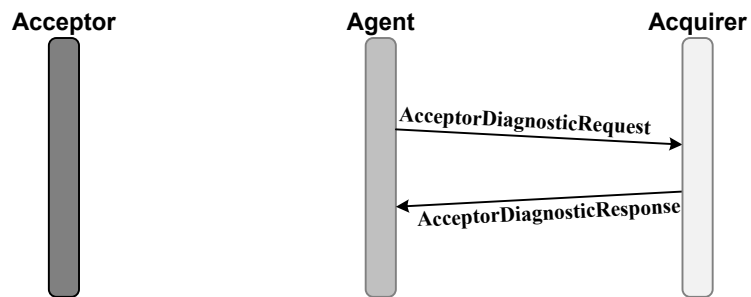


Figure 39: Diagnostic Request from an Intermediary Agent

2.8 Reject Message

A *RecipientParty* sends an *AcceptorRejection* message (caaa.015.001.05) to an *InitiatingParty* to indicate that the *RecipientParty* could not process the received message.

For instance, if an *InitiatingParty* sends an *AcceptorAuthorisationRequest* message and that message is not recognised by the *RecipientParty*, the *RecipientParty* sends back an *AcceptorRejection* in response to that message.

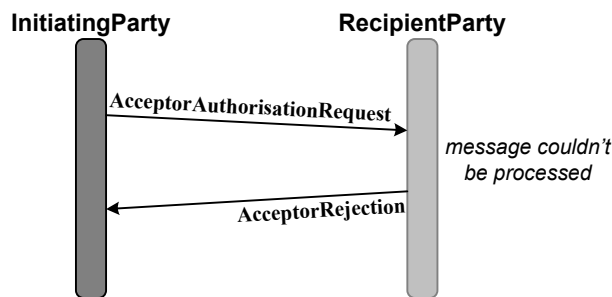


Figure 40: Rejection of an Authorisation

The *AcceptorRejection* message contains the reason of the rejection (*RejectReason*), some additional information on the rejection (*AdditionalInformation*) for further analysis as well as the rejected message itself (*MessageInError*) (to make it possible to compare with the message sent).

The *AcceptorRejection* message must be sent in the following cases:

1. The envelope of the received message is incorrect (see section 3.4.1.2 a).
RejectReason contains the value *InvalidMessage*. It is recommended to include the optional fields *AdditionalInformation* to provide the details of the error. *MessageInError* contains the received message with the error.
2. The rejected message cannot be decoded properly (see section 3.4.1.2 b).
RejectReason contains the value *ParsingError*. It is recommended to include the optional fields *AdditionalInformation* to provide the details of the coding error. *MessageInError* contains the received message with the coding error.
3. The identification of the rejected message is invalid (see section 3.4.1.2 e).
RejectReason contains the value *InitiatingParty* or *RecipientParty*. No other field is required. *AdditionalInformation* may contain the invalid identifier.
4. The verification of the security of the rejected message fails (see section 3.4.1.2 d).
RejectReason contains the value *Security*. It is recommended to include the optional fields *AdditionalInformation* to provide the details of the security error. *MessageInError* contains the received message with the security error.

5. The version of the protocol used for the message (*Header.ProtocolVersion*) is not supported by the *RecipientParty* which is not able to send a message response of this version to the *InitiatingParty* (see section 3.4.1.2 c)).
RejectReason contains the value *ProtocolVersion* and *AdditionalInformation* the invalid protocol version.
6. The rejected message has already been sent by the *InitiatingParty* to the *RecipientParty*. The message could not be processed a second time, and then a response could not be sent (see section 3.4.1.2 f)).
RejectReason contains the value *DuplicateMessage*. No other field is required.
AdditionalInformation may contain the invalid *ExchangeIdentifier* value.
7. The *RecipientParty* is not able to process the message for lack of resources. For that reason, a message response could not be built and the message is not processed (see section 3.4.1.3 b)).
RejectReason has the value *UnableToProcess*. *AdditionalInformation* contains the reason why the message could not be processed.

The reaction of the *InitiatingParty* to an *AcceptorRejection* message depends on the *RejectReason* and the type of rejected message.

As an *AcceptorRejection* message is related to a specific request or an advice message between two entities, an Agent never forwards an *AcceptorRejection* message.

In the example illustrated below, an Agent forwards an *AcceptorAuthorisationRequest* message issued by an Acceptor to the relevant Acquirer. The Acquirer cannot process the message and issues an *AcceptorRejection* message instead of an *AcceptorAuthorisationResponse*. The Agent receives the rejection message and sends to the card acceptor the appropriate *AcceptorAuthorisationResponse* message with *TransactionResponse.ResponseToAuthorisation.Response* containing the value *TechnicalError*.

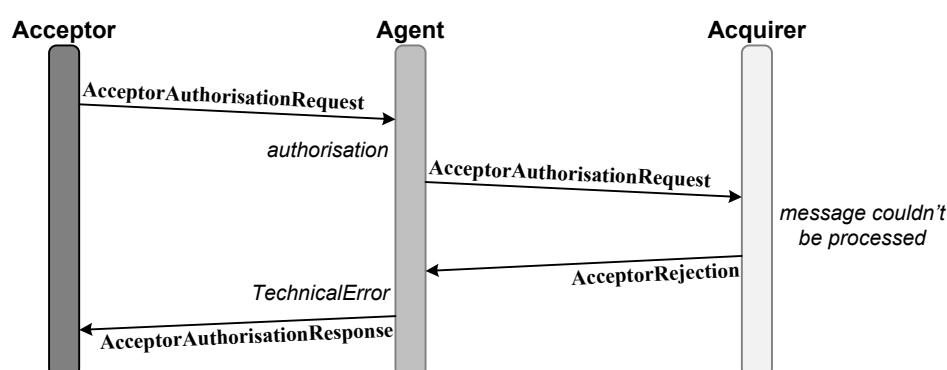


Figure 41: Rejection of an Authorisation to an Agent

However, when the acquirer returns an *AcceptorRejection* message with *RejectReason* set to *UnableToProcess*, in case of congestion of the acquirer host, the agent must return an *AcceptorRejection* message with the same *RejectReason*.

In the example below, an Agent receives an *AcceptorCompletionAdvice* message issued by an Acceptor:

- The Agent performs the completion of the transaction, including the sending of:
 - an *AcceptorCompletionAdvice* to the relevant Acquirer,
 - a positive *AcceptorCompletionAdviceResponse* to the Acceptor without waiting for the response of the Acquirer.
- The Acquirer cannot process the message and sends an *AcceptorRejection* message instead of an *AcceptorCompletionAdviceResponse* message.
- The Agent receives the rejection message and initiates the message retransmission process (see section 3.3).

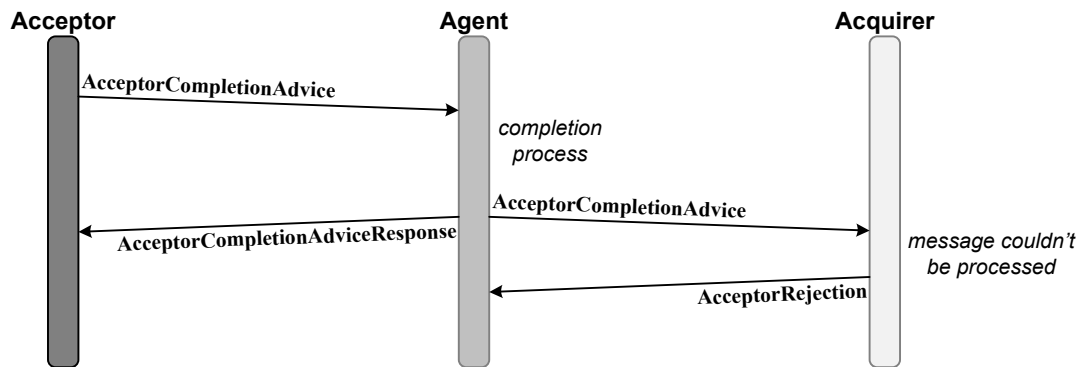


Figure 42: Rejection of an Completion to an Agent

3 Message Functionalities

3.1 Message Organisation

A CAPE message is usually composed of three major functional blocks:

- a Message Header containing information related to the management of the message (routing and processing)
- a Message Body containing information related to the application processing of the message
- a Security Trailer containing information related to the security aspects of the message (optional).

3.1.1 Management of the message

	Message item	Multiplicity
A.	Header	[1..1]
	MessageFunction	[1..1]
	ProtocolVersion	[1..1]
	ExchangeIdentification	[1..1]
	ReTransmissionCounter	[0..1]
	CreationDateTime	[1..1]
+	InitiatingParty	[1..1]
+	RecipientParty	[0..1]
+	TraceabilityInfo	[0..1]

The Header contains primary information required to either route the message to a specific destination (e.g. an intermediary agent) or to process the message in a specific way (e.g. retransmission, financial capture, etc.).

The Header is composed of three major categories of information:

- Management information
- Parties involved in the transmission of the message
- Traceability information

3.1.1.1 Management information

The Management information regroups data components associated with the routing and/or processing of the message.

Message Item	Mult.	Usage
MessageFunction	[1..1]	<p><u>Definition</u>: This component identifies the type of process related to the message.</p> <p><u>Usage</u>: Used mainly for routing purposes, namely in cases where the message relies on some kind of processing (e.g. message process with financial capture or without financial capture).</p>
ProtocolVersion	[1..1]	<p><u>Definition</u>: This component identifies the version of the protocol used in the exchange.</p> <p><u>Usage</u>: The management of the protocol version is described in sections 4.2.1 and 4.2.2 of the present specifications.</p>
ExchangeIdentification		<p><u>Definition</u>: This component provides a unique identification of the exchange of messages (request/response) for a specific service.</p> <p><u>Usage</u>: This information is structured and used according to guidelines provided by the Acquirer.</p>
RetransmissionCounter		<p><u>Definition</u>: This component identifies the number of retransmissions for this message.</p> <p><u>Usage</u>: Retransmission is used for <i>AcceptorCompletionAdvice</i> (caaa.003.001.06), <i>AcceptorCancellationAdvice</i> (caaa.007.001.06) and <i>AcceptorCurrencyConversionAdvice</i> (caaa.018.001.01) messages. Retransmission of other messages is not allowed.</p>
CreationDateTime		<p><u>Definition</u>: This component identifies the date and time of the creation of the message. This information is mandatory and needs to be present in the related message.</p>

3.1.1.2 Parties involved in the message

The parties mentioned in the Header are provided with the purpose of facilitating the routing of the message. The Header is not included when applying security to the message.

MessageItem	Mult.	Usage
InitiatingParty		<p><u>Definition</u>: Party initiating the exchange. This can either be the Acceptor or the party that initiates the exchange on behalf of the Acceptor. This information is not part of the secured section of the message.</p> <p><u>Usage</u>: Validate the origin of the exchange. The role of <i>InitiatingParty</i> stays unchanged during the whole exchange of the message (e.g. <i>AcceptorCompletion Advice</i> and <i>AcceptorCompletion AdviceResponse</i>). The <i>InitiatingParty</i> must always be present.</p>
RecipientParty		<p><u>Definition</u>: Party recipient of the exchange. This can either be the Acquirer, or the party that received the exchange on behalf of the Acquirer. This information is not part of the secured section of the message.</p> <p><u>Usage</u>: Validate the destination of the exchange. The role of <i>RecipientParty</i> stays unchanged during the whole exchange of the message (e.g. <i>AcceptorCompletionAdvice</i> and <i>AcceptorCompletionAdviceResponse</i>). The presence of <i>RecipientParty</i> is configurable.</p>

3.1.1.3 Traceability information

The Traceability information allows a RecipientParty to monitor the upstream transport and process of a message throughout its lifetime.

In order to ensure an efficient traceability process end-to-end, each intermediary entity is invited to add its own Traceability information to the incoming message before sending the resulting message to the next party in the chain.

The value and interest of this process depends on the actual use of this option by all intermediaries in the message exchange chain.

Traceability provides useful information as regards the processing and transport time for the exchange of messages. This information can be used to carry out a global analysis of the performance of the exchanges of messages end-to-end. In case of delays in processing card payment transactions, it can also be used to identify the possible bottlenecks or problems encountered in the routing or processing of the message.

This information is conditional and is not part of the secure section of the message.

MessageItem	Mult.	Usage
RelayIdentification		<p><u>Definition</u>: Party relaying a message. This can be the Acceptor, the Acquirer or any Intermediary Agent relaying a message on behalf of the Acceptor or the Acquirer.</p> <p><u>Usage</u>: Used essentially to inform parties about the routing process of the exchange (actual parties involved in the exchange chain, time spent between agents in the chain, problems encountered whilst processing or forwarding messages, etc.). Whilst the use of this message is conditional - once the relevant party has completed RelayIdentification - it must further be enriched by all following parties in the card payment chain.</p>
TraceDateTimeln		Date and time of incoming messages for relaying or processing
TraceDateTimeOut		Date and time of outgoing messages for relaying or processing

3.1.2 ApplicationData

	Message item	Multiplicity
A.	Body	[1..1]
+	Environment	[1..1]
+	Context	[1..1]
+	Transaction	[1..1]

The Body of the message contains information related to the processing of the message by an application.

The data is grouped by functional components such as:

- the environment of the card payment transaction in terms of actors involved (Acquirer, Merchant, POI, Card, Cardholder),
- the context of the transaction (payment and sale contexts)
- the information related to the card payment transaction itself.

3.1.2.1 Environment

The environment functional component contains information related to the main actors involved in or impacted by a payment transaction.

Actors need to be understood in the present context as either organisations and individuals (Acquirer, Merchant, Cardholder) or objects (POI, Card).

The *InitiatingParty* (part of the Header) and *Merchant* (part of the Body) may be distinct entities.

The *InitiatingParty* may be an entity (e.g. a Processor) acting on behalf of the *Merchant* (or the *Acquirer*) to ensure the routing and processing of the message, whilst the *Merchant* and the *Acquirer* are entities involved in a commercial relationship associated with the actual provisioning of a card payment transaction to a Cardholder.

Message Item	Mult.	Usage
Acquirer	[1..1]	<p><u>Definition</u>: A party in a contractual relation with Merchants and Card schemes accepting payments. The Acquirer makes payments to the Merchant based on data received from the Acceptor.</p> <p><u>Usage</u>: An Acquirer acquires card payment data from an Acceptor and forwards the data to the relevant Card Issuer.</p>
Merchant	[1..1]	<p><u>Definition</u>: A party providing goods and/or services at a sales location (physical or virtual). The Merchant signs an acquiring agreement with an Acquirer. The Merchant can perform the role of Acceptor or delegate it to another party.</p>
POI		<p><u>Definition</u>: A Point of Interaction defines the entry point of a card into a payment system to accept payment and loyalty cards at a sales location. It is a general term used to include all situations where payment details may be entered (e.g. petrol pump, merchant checkout, internet, mail order, telephone order, etc.).</p> <p><u>Usage</u>: The identification of the POI is usually allocated by a Merchant, an Acceptor, an Intermediary Agent or an Acquirer.</p>
Card		<p><u>Definition</u>: A physical or virtual device to enable a payment and to identify a Cardholder account.</p> <p><u>Usage</u>: The card is usually identified by its PAN and may be additionally qualified by a card sequence number.</p>
Cardholder		<p><u>Definition</u>: A person presenting a Card or card information to a Merchant for the purchase of goods and services.</p> <p><u>Usage</u>: This item contains data elements used to identify and authenticate the cardholder.</p>

3.1.2.2 Context

Context describes the two main types of contexts associated to a card payment transaction: a sale context (SaleContext) that identifies elements related to the commercial aspects of the transaction exclusively and a payment context (PaymentContext) which provides the elements of information for the payment associated with the sale transaction.

Message Item	Mult.	Usage
PaymentContext	[1..1]	<u>Definition:</u> This component identifies the elements of the sales environment for payment.
SaleContext	[0..1]	<u>Definition:</u> This component contains elements that identify the sale within the sales environment.

3.1.2.3 Transaction

This functional block contains all the information related to a card payment for which an authorisation or another type of operation is required.

Message Item	Mult.	Usage
Transaction	[1..1]	<u>Definition:</u> This component contains the transaction specific application data.
Product		<u>Definition:</u> This component identifies the class of goods and/or services.

3.1.3 SecurityTrailer

	Message item	Multiplicity
A.	SecurityTrailer	[0..1]
	ContentType	[1..1]
+	AuthenticatedData	[0..*]

The SecurityTrailer building block is conditional. It contains a message authentication code computed on the body of the message with a cryptographic key. It allows the authentication of the Initiator and protects the content of the body against any unauthorised alteration of the message.

Message Item	Mult.	Usage
ContentType	[1..1]	<u>Definition:</u> This component identifies the type of security used for the message (authentication).
AuthenticatedData	[0..*]	<u>Definition:</u> This component contains the Message Authentication Code (MAC) for the message body with the required information to check it.

3.2 Traceability

3.2.1 Security level

No security is required for traceability data. Traceability is not included in data elements taken into account for the computation of the MAC.

3.2.2 Usage Condition

The presence of Traceability is configurable. When Traceability is configured as required, traces must be present on the following messages: `AcceptorAuthorisationRequest`, `AcceptorCompletionAdvice`, `AcceptorCancellationRequest` and `AcceptorCancellationAdvice`.

3.2.3 Traceability in an exchange

The path of traceability goes from the Acceptor back to the Acceptor and includes all intermediate steps. In other terms, it means that when the Acceptor receives the response, the whole traceability of the Acceptor to Acquirer domain is in the header.

For the entity which initiates the transaction, the value of “`TraceDateTimeln`” and the value of “`TraceDateTimeOut`” are the same.

An intermediary agent and the acquirer must populate Traceability if a previous entity has populated it; The recipient must update Traceability when a message is received containing trace information by adding a new trace entry with `TraceTimeln` set to the time the recipient received the new message and `TraceDateTimeOut` set to the time the recipient sent the message to the next party.

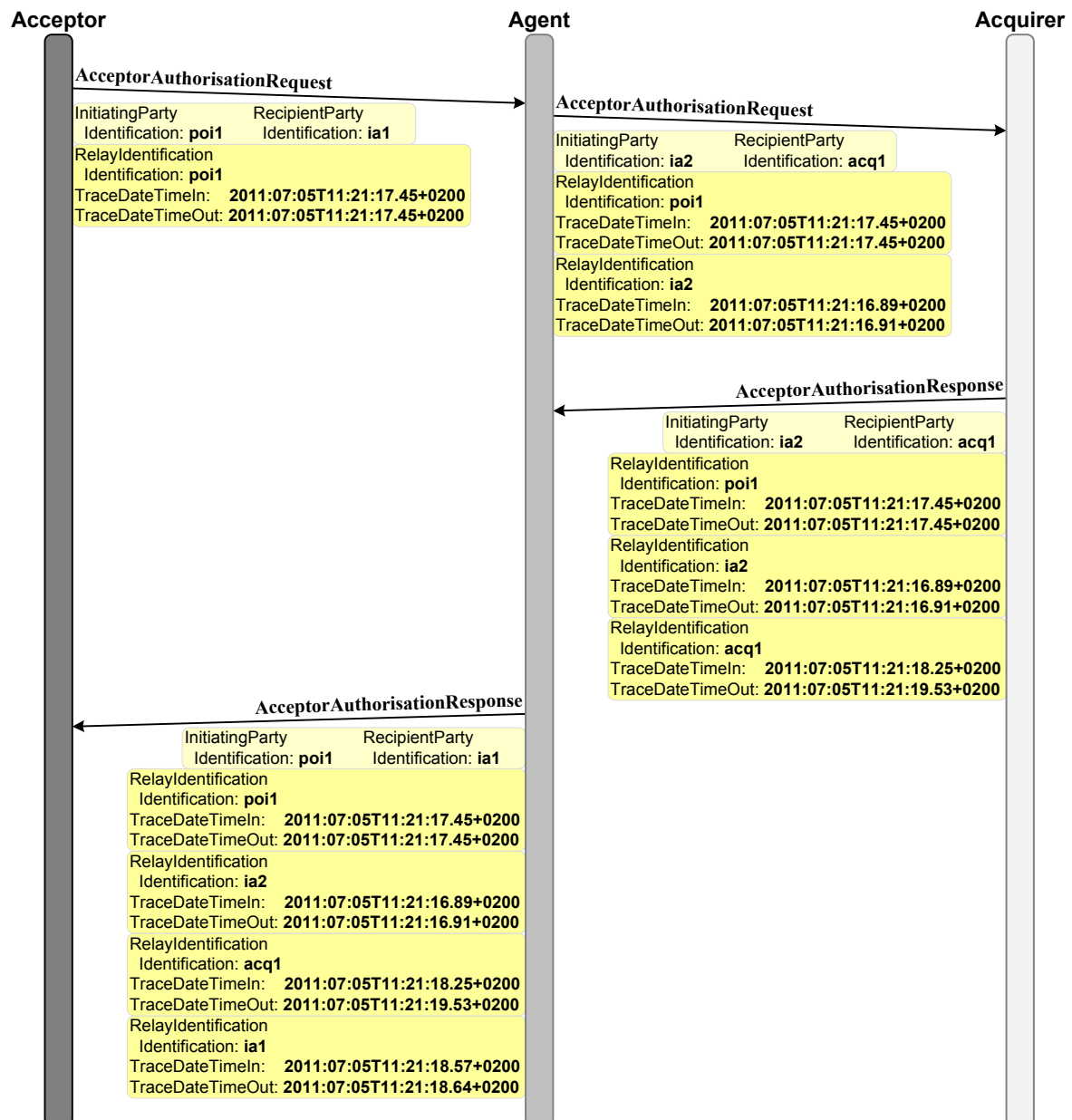


Figure 43: Example of Traceability with an Intermediary Agent

3.3 Message Retransmission

An Acceptor retransmits a message when no response was received to an advice message sent to an Acquirer.

Retransmission is used for *AcceptorCompletionAdvice* (caaa.003.001.06), *AcceptorCancellationAdvice* (caaa.007.001.06) and *AcceptorCurrencyConversionAdvice* (caaa.018.001.01) messages only. Retransmission of other messages is not allowed.

The purpose of retransmission is to ensure that an Acquirer has received and processed such messages. The Acceptor initiates a retransmission process if the response to an advice is not received.

A repeated advice message holds an unchanged copy of the message body which was originally sent. *RetransmissionCounter* and *CreationDateTime* are the only data element which are modified in the *Header* for each retransmission. In particular, the *ExchangeIdentification* has the same value for all retransmissions of the same message. For DUKPT key management, because of the key evolution, the Acceptor may have to recompute the MAC to be able to verify the MAC of the response.

3.3.1 Acceptor Behaviour

The *RetransmissionCounter* is absent from the first transmission of the advice message, or has a value of 0. This component is present in every repeated advice message, with a value of 1 for the first repetition, and incremented by 1 for each new repetition.

The number of message retransmissions must be limited, according to the configuration of the POI. The protocol does not address potential errors that may arise when the limit of retransmissions is reached.

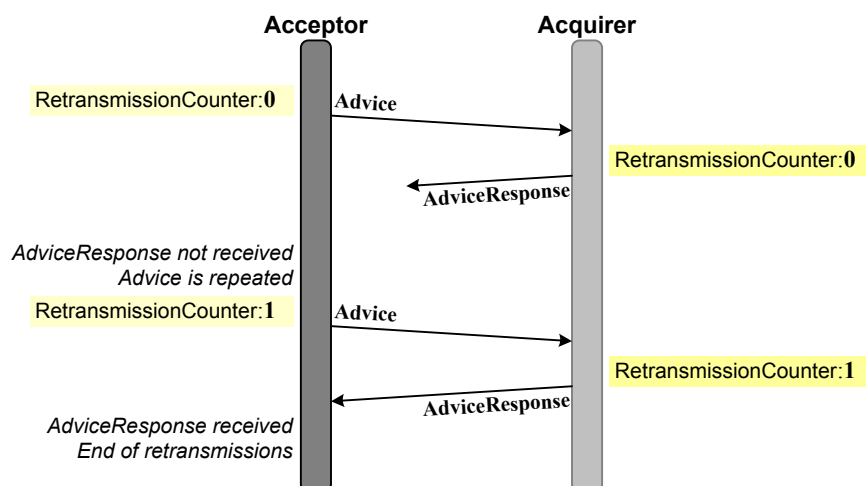


Figure 44: Retransmission of an Advice by the Acceptor

The Acceptor stops the retransmission process upon receipt of a correct *AdviceResponse* message. The value of *RetransmissionCounter* has no impact on the acceptance of the Advice Response message.

3.3.1.1 Late Responses to an Advice message

In the example below, a response to an advice message seems to be lost, since received late.

The Acceptor retransmits the advice message to the Acquirer (*RetransmissionCounter* set to 1).

The Acceptor initiates a second retransmission of the advice message (*RetransmissionCounter* value of 2) before a response to the first retransmitted advice message was received from the Acquirer.

The receipt of the first advice response (*RetransmissionCounter* value of 1) by the Acceptor stops any further retransmission of the message.

The receipt of the second advice response (*RetransmissionCounter* value of 2) by the Acceptor is simply ignored by the same party.

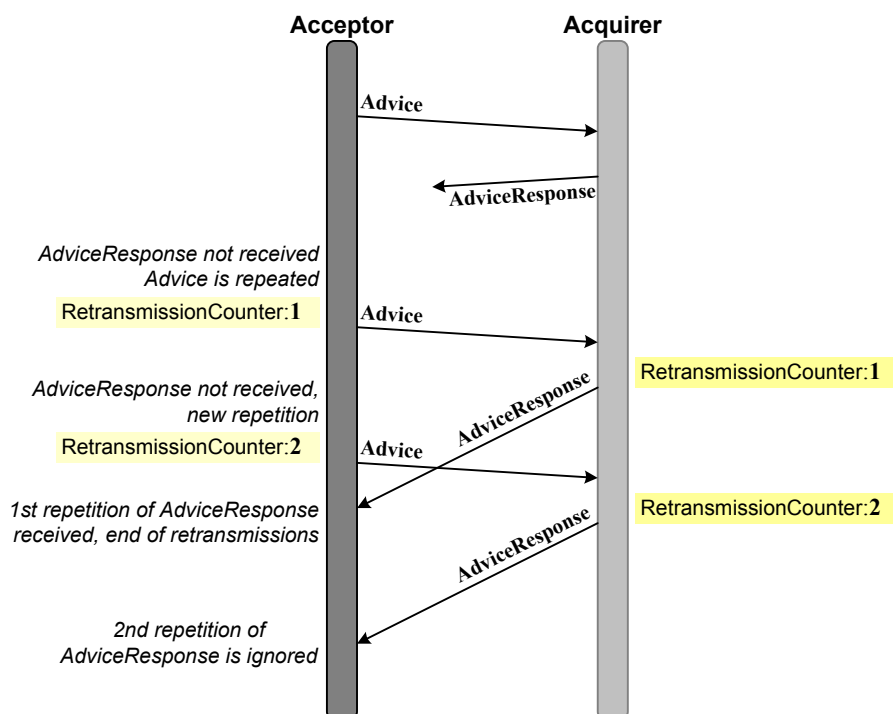


Figure 45: Late Responses to an Advice message

3.3.1.2 Unordered Responses to Retransmitted Advice messages

In some circumstances, AdviceResponse messages may be received in a different order from that in which they were sent. In the example below, the response to a second retransmission is received by the Acceptor before receiving a response to the first retransmission.

The response to the second retransmission (*RetransmissionCounter* value of 2) terminates the retransmission process and the Acceptor ignores the response to the first repetition (*RetransmissionCounter* value of 1).

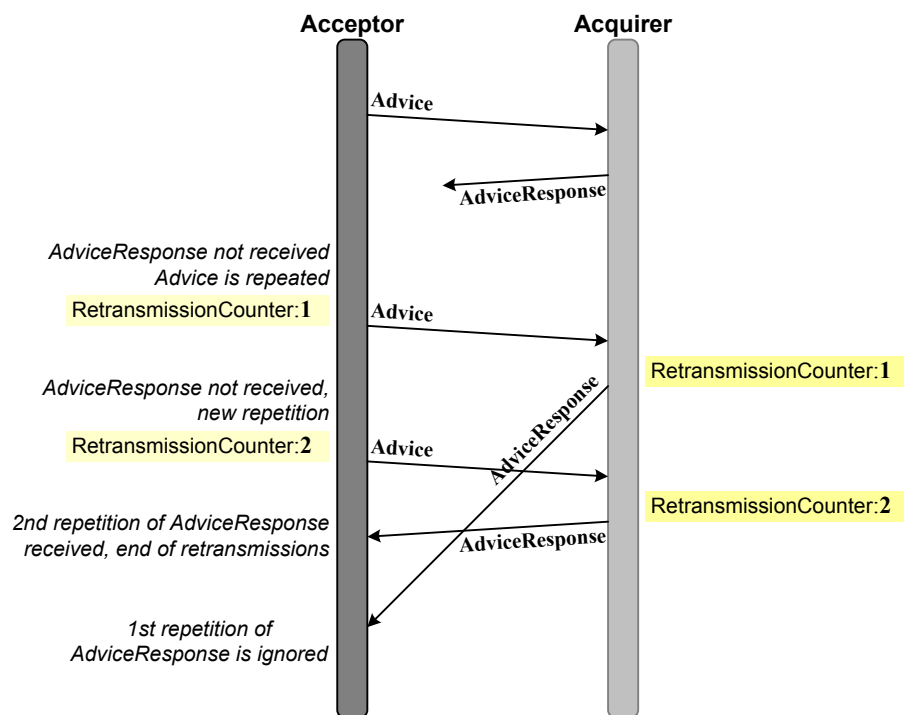


Figure 46: Unordered Responses to Retransmitted Advice messages

3.3.2 Acquirer Behaviour

An advice message with *RetransmissionCounter* set to 0 or absent means that the received message was an original one (e.g. not retransmitted).

For each advice message received, the Acquirer sends back a response message to the Acceptor containing a copy of the *RetransmissionCounter* value.

It is required for the Acquirer to send an advice response to every retransmissions of an advice.

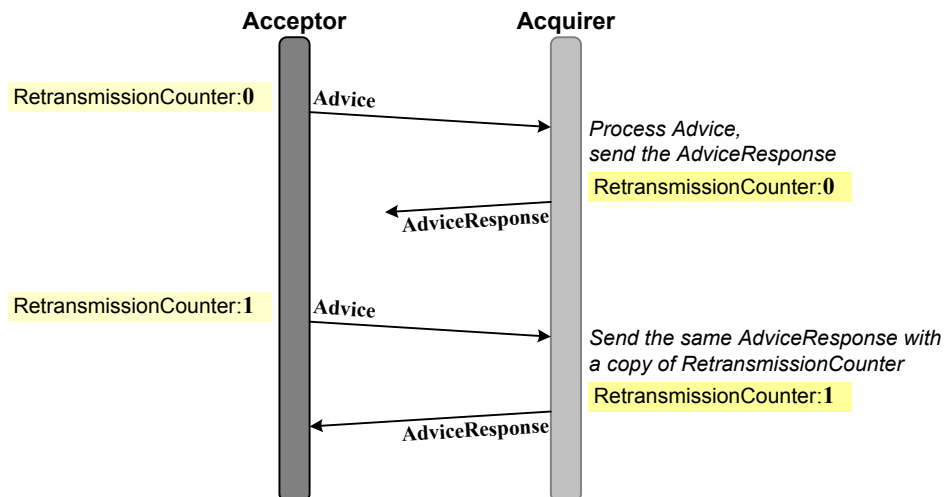


Figure 47: Retransmission of an AdviceResponse by the Acquirer

Since the original advice might never reach the Acquirer (due to a problem of communication, for instance), the retransmitted message received by the Acquirer (*RetransmissionCounter* set to 1) has to be considered as the original one and processed as such.

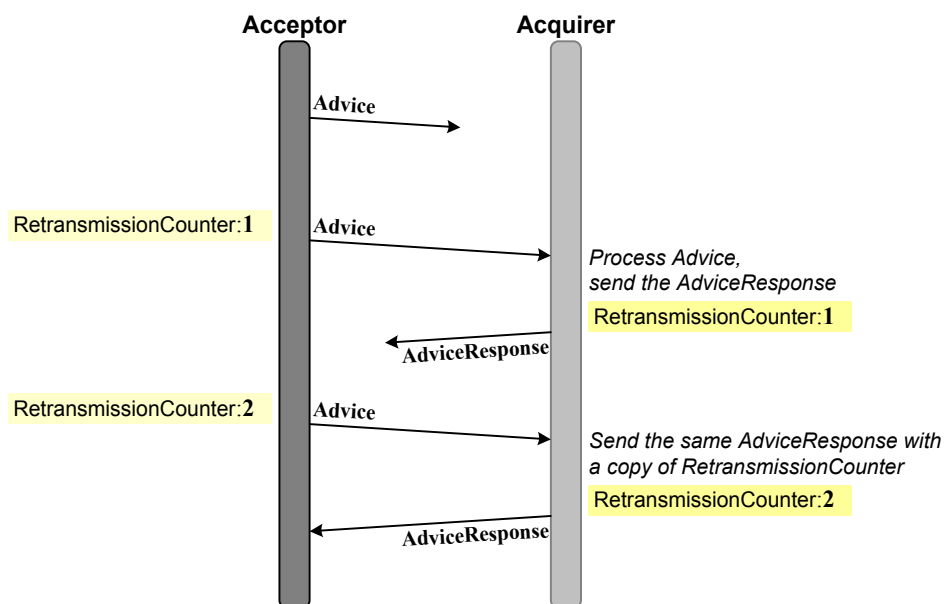


Figure 48: Non-Receipt of an Advice by the Acquirer

Similarly, advice retransmissions are not necessarily received by the Acquirer in the order adopted by the Acceptor.

3.4 Error Handling

This section presents the errors related to an exchange of messages which occurs at a protocol level as:

- The format of the message, including the security,
- The identification of the parties involved,
- The management of messages, including version management, linking a request with its response, and message duplication,
- The communication between the sender and the receiver of a message, including the absence of a response.

3.4.1 Error Cases

Error cases, summarised in the figure below, are presented in the order the messages are processed by the parties during the exchange.

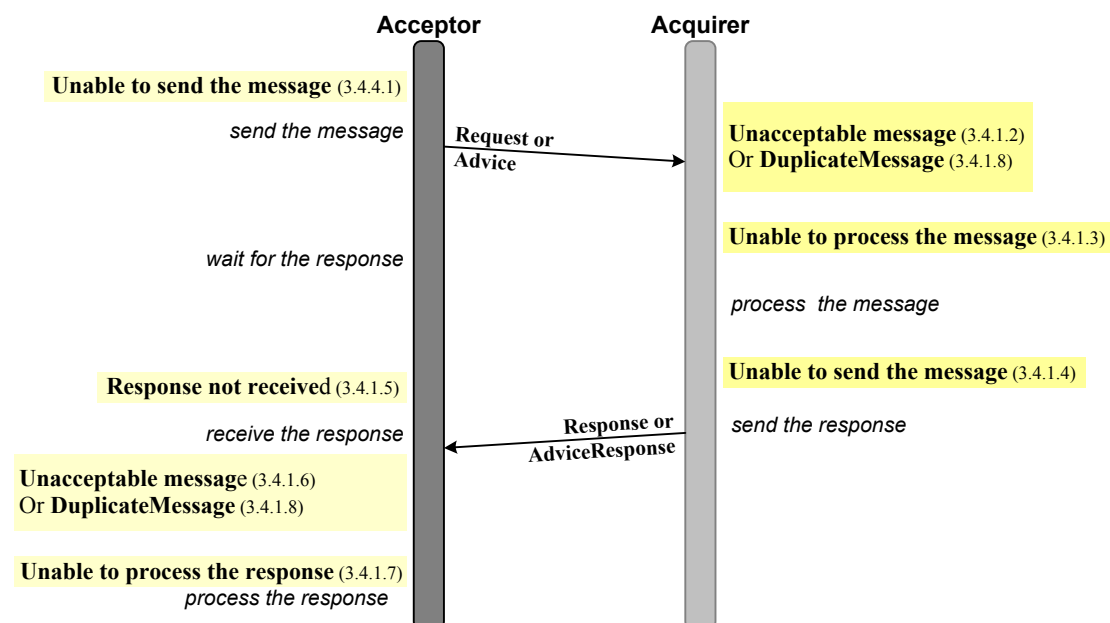


Figure 49: Error Cases in Message Exchange

3.4.1.1 Acceptor is Unable to Send a Message

An Acceptor is unable to send a request or an advice, and is aware that the Acquirer did not receive the complete message. This error occurs when:

- No open transport connection is available, and no transport connection to the Acquirer host could be opened, after reaching a maximum number of retries, or
- The sending of the complete message has failed for some reason reported by the transport layer.

3.4.1.2 Acquirer Receives an Unacceptable Message

The message received by the Acquirer is not an acceptable message for one or several of the following reasons:

a) The envelope of the received message is invalid:

- The tag of the XML root element is not `Document`,
- The root element contains more than one (child) element,
- The root element is empty and does not contain any element,
- The tag of the unique root child is not one of the following values: `AccptrAuthstnReq`, `AccptrCmpltnAdvc`, `AccptrCxlReq`, `AccptrCxlAdvc`, `AccptrRcncltnReq`, `AccptrBtchTrf`, `AccptrDgnstcReq`, `AccptrCcyConvsReq` or `AccptrCcyConvsAdvc`.

b) The message cannot be decoded properly:

- The size of the message exceeds the maximum size the Acquirer is able to handle.
- The XML parser generates a parsing error. For instance, the message is not a well formed XML document, element that contains other elements has a non-empty content, elements contain mixed content, or the message contains unexpected entities as DTD or Processing Instructions.
- The XML version of the document is neither 1.0 nor 1.1,
- The character encoding is defined either in the XML prolog or in a byte order mark (BOM), but not as UTF-8 or EF BB BF respectively (the usage of a BOM for the messages is not recommended).
- The document contains invalid UTF-8 characters.
- The root element `Document` has no default XML Namespace declaration, or the name of the Namespace is incorrect, or the name of the Namespace is inconsistent with the single child of the root element (i.e. is different from `urn:iso:std:iso:20022:tech:xsd:caaa.001.001.06`, `urn:iso:std:iso:20022:tech:xsd:caaa.003.001.06`, `urn:iso:std:iso:20022:tech:xsd:caaa.005.001.06`, `urn:iso:std:iso:20022:tech:xsd:caaa.007.001.06`, `urn:iso:std:iso:20022:tech:xsd:caaa.009.001.06`, `urn:iso:std:iso:20022:tech:xsd:caaa.013.001.06`, `urn:iso:std:iso:20022:tech:xsd:caaa.016.001.04`, and for `urn:iso:std:iso:20022:tech:xsd:caaa.018.001.01`, respectively `AccptrAuthstnReq`, `AccptrCmpltnAdvc`, `AccptrCxlReq`, `AccptrCxlAdvc`, `AccptrRcncltnReq`, `AccptrDgnstcReq`, `AccptrCcyConvsReq` and `AccptrCcyConvsAdvc`).
- The sequence of elements defined in the schema for the message is not respected. The sequence of optional elements in a data structure must respect the sequence of elements defined in the XML/Schema (e.g. In the InitiatingParty, the data element Type cannot be after Issuer).
- A mandatory data structure, a mandatory data element is absent or a data element configured as "Mandatory" is absent (e.g. Header.ExchangeIdentification is absent).
- A data element has not a valid value for the type defined in the schema. For instance, a numeric data containing non numeric characters.
- The value of a data element is not belonging to the set of enumerated values defined in the schema or is not allowed (see section 4 *Messages and Usage*).
- The size of the value exceeds the maximum size defined in the schema.
- Inconsistency between data elements (e.g. Header.MessageFunction value inconsistent with the body of the message).
- A rule or a condition of presence defined in the section 4 is not respected (e.g. `Response=Approved` and an Action with `ActionType=Referral` in the `AcceptorAuthorisationResponse` message).

If an optional data element that is not expected but present in the message it must be ignored, and the message is not rejected.

If a data element or a data structure is not known by the receiver of the message (i.e. not present in the enclosing data structure definition), this data element is ignored (the schema validation of the XML message is not required).

- c) The version of the protocol used for the message (Header.ProtocolVersion) is not supported by the RecipientParty.
- d) The verification of the security of the message fails:
 - A cryptographic key identification is invalid (including the Key Serial Number identification part of a DUKPT key),
 - The cryptographic key has expired,
 - The computed MAC does not match the MAC value sent in the message,
 - The decrypted data is inconsistent (e.g. wrong format or invalid content).
- e) The identification of the sender (Header.InitiatingParty) or the receiver (Header.RecipientParty) of the message is invalid.
- f) The message is a duplicated message for a given InitiatingParty and a given RecipientParty when the following fields have the same value:
 - Header.ExchangeIdentification
 - CreationDateTime
 - Header.RetransmissionCounter if present.

3.4.1.3 Acquirer is Unable to Process the Message

The message received by the RecipientParty cannot be performed for one of the following reasons:

- a) The type of message (Header.MessageFunction) is not supported by the RecipientParty, which is not able to process the message and send a response message to the InitiatingParty.
- b) The RecipientParty is not able to process the message for lack of resources (e.g. congestion of server, HSM unavailable). For that reason, a message response could not be built and the message is not processed.

3.4.1.4 Acquirer is Unable to Send a Message

The Acquirer is unable to send a response to a request or an advice, and is sure that the complete message has not been received by the Acceptor. The error occurs when the transport connection previously open by the Acceptor to send the request or the advice, has been broken or released before the sending of the complete response message.

3.4.1.5 Acceptor has not Received a Response Message

No response to a request or an advice message has been received by the Acceptor, resulting from one of the following reasons:

- a) The transport connection used to send the request or advice has been released or broken, and the Acquirer is then unable to send the response message.
- b) The timer monitoring the receipt of the response message has expired before the receipt of the complete response message.

3.4.1.6 Acceptor Receives an Unacceptable Message

The message received by the Acceptor is not an acceptable message for one or several of the following reasons:

- a) The envelope of the received message is incorrect:
 - The tag of the XML root element is not `Document`,
 - The root element contains more than one (child) element,
 - The root element does not contain any element,
 - The tag of the unique root child is not one of the following values: `AccptrAuthstnRspn`, `AccptrCmpltnAdvRspn`, `AccptrCxlRspn`, `AccptrCxlAdvRspn`, `AccptrCncltnRspn`, `AccptrBtchTrfRspn`, `AccptrDgnstcRspn`, `AccptrRjctn`, `AccptrCcyConvsRspn` or `AccptrCcyConvsAdvRspn`.
- b) The message cannot be decoded properly:
 - The size of the message exceeds the maximum size the Acceptor is able to handle.
 - The XML parser generates a parsing error. For instance, the message is not a well formed XML document, element that contains other elements has a non-empty content, elements contain mixed content, or the message contains unexpected entities as DTD or Processing Instructions.
 - The XML version of the document is neither `1.0`, `1.1` nor `2.0`,
 - The character encoding is defined either in the XML prolog or in a byte order mark (BOM), but not as `UTF-8` or `EF BB BF` respectively (the usage of a BOM for the messages is not recommended).
 - The document contains invalid UTF-8 characters.
 - The root element `Document` has no default XML Namespace declaration, or the name of the Namespace is incorrect, or the name of the Namespace is inconsistent with the single child of the root element (i.e. is different from `urn:iso:std:iso:2002:tech:xsd:caaa.002.001.06`, `urn:iso:std:iso:2002:tech:xsd:caaa.004.001.06`, `urn:iso:std:iso:2002:tech:xsd:caaa.006.001.06`, `urn:iso:std:iso:2002:tech:xsd:caaa.008.001.06`, `urn:iso:std:iso:2002:tech:xsd:caaa.010.001.05`, `urn:iso:std:iso:2002:tech:xsd:caaa.014.001.05`, `urn:iso:std:iso:2002:tech:xsd:caaa.015.001.05`, `urn:iso:std:iso:2002:tech:xsd:caaa.017.001.04`, and `urn:iso:std:iso:2002:tech:xsd:caaa.019.001.01`, for respectively `AccptrAuthstnRspn`, `AccptrCmpltnAdvRspn`, `AccptrCxlRspn`, `AccptrCxlAdvRspn`, `AccptrCncltnRspn`, `AccptrDgnstcRspn`, `AccptrRjctn`), `AccptrCcyConvsRspn` and `AccptrCcyConvsAdvRspn` ..

- The sequence of elements defined in the schema for the message is not respected (e.g. Header.InitiatingParty appears before Header.MessageFunction).
- A mandatory data structure or a mandatory data element is absent (e.g. Header.ExchangeIdentification is absent).
- A data element has not a valid value for the type defined in the schema. For instance, a numeric data containing non numeric characters.
- The value of a data element is not belonging to the set of enumerated values defined in the schema or is not allowed (see section 4 *Messages and Usage*).
- The size of the value exceeds the maximum size defined in the schema.
- Inconsistency between data elements (e.g. Header.MessageFunction value inconsistent with the body of the message).
- A rule or a condition of presence defined in the section 4 is not respected (e.g. ProtectedCardData and PlainCardData both present in the AcceptorAuthorisationResponse message).
- A data element in a Response message (resp. AdviceResponse message) is specified as "Copy", and has not the same value as in the related Request message (resp. Advice message).

If a data element or data structure is not known by the receiver of the message, this data element is ignored (the schema validation of the XML message is not required).

c) The verification of the security of the rejected message fails:

- A cryptographic key identification is invalid (including the Key Serial Number identification part of a DUKPT key), or is not the same than in the request,
- The cryptographic key has expired,
- The computed MAC does not match the MAC value sent in the message,
- The decrypted data is inconsistent (e.g. wrong format or invalid content).

3.4.1.7 Acceptor is Unable to Process the Response Message

The response message received by the Acceptor cannot be processed for one of the following reasons:

a) The message cannot be considered as a response to the request or the advice:

- Message identification (Header.ExchangeIdentification, Header.InitiatingParty, Header.RecipientParty) does not have the same value as the request or the advice.
- If the response is not an AcceptorRejection, the message types (Header.MessageFunction or the message body) is not related to the request or the advice (e.g. MessageFunction=*FinancialAuthorisationRequest* in the request message and MessageFunction=*AuthorisationResponse* in the response message, or the message body AuthorisationRequest is in the request message and CancellationResponse in the response message).
- A data element specified as to be copied from the request or the advice has not the same value (e.g. Transaction.TransactionIdentification.TransactionDateTime or Transaction.TransactionIdentification.TransactionReference)

b) The message received is an AcceptorRejection related to the request or the advice.

c) The response message is received too late, i.e. after the expiration of the time out, or just after the transport connection release.

3.4.1.8 Acquirer or Acceptor has received a Duplicate Message.

Regardless of the root cause, the message received by the RecipientParty has already been received. See duplicate message definition below.

In that case, the RejectReason is set to *DuplicateMessage*

3.4.2 Acceptor Error Handling

Errors handling of the Acceptor follow the process below:

- 1) Error case detection, as described in the previous section (see 3.4.1 Error Cases).
- 2) Error processing at a protocol level, depending on the type of exchange.
- 3) Continue to process the exchange or
- 4) Terminate the exchange and notify the error to the application.

Following sections describe each type of error handling of the Acceptor and the conditions under which this handling has to be performed. The diagram below summarises for each error case defined in the section 3.4.1, the error handling per message the Acquirer must perform.

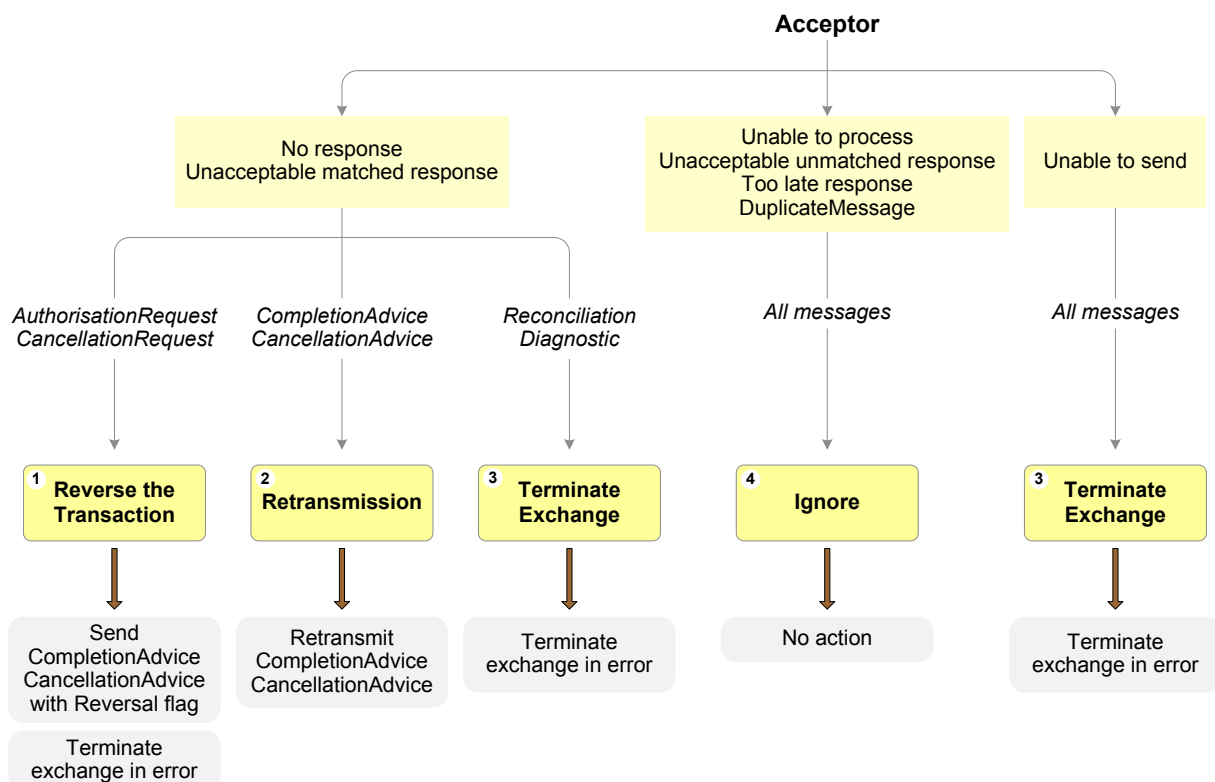


Figure 50: Acceptor Error Handling in a Message Exchange

3.4.2.1 Reverse the Transaction

3.4.2.1.1 Conditions

This process is executed by the Acceptor after sending an *AcceptorAuthorisationRequest* or an *AcceptorCancellationRequest* for the following error situations:

- The Acceptor has not received a response message (see section 3.4.1.5).
- The message is not an acceptable response (see section 3.4.1.6), and matches a request in progress.
- The Acceptor is unable to process the response message (see section 3.4.1.7 a) and b)), and matches a request in progress.

3.4.2.1.2 Action

The Acceptor sends an *AcceptorCompletionAdvice* or an *AcceptorCancellationAdvice* to reverse the respectively the *AcceptorAuthorisationRequest* and the *AcceptorCancellationRequest* (setting *Transaction.Reversal* to the value *True*), and notifies the application of the failure of the exchange.

3.4.2.2 Retransmission of the Advice

3.4.2.2.1 Conditions

This process is executed by the Acceptor after sending an *AcceptorCompletionAdvice* message or an *AcceptorCancellationAdvice* message for the following error situations:

- The Acceptor has not received a response message (see section 3.4.1.5).
- The message is not an acceptable response (see section 3.4.1.6), and matches an advice in progress.
- The Acceptor is unable to process the response message (see section 3.4.1.7 a) and b)), and matches an advice in progress.

3.4.2.2.2 Action

The Acceptor retransmits the *AcceptorCompletionAdvice* or the *AcceptorCancellationAdvice* message in accordance to the section 3.3.1 *Message Retransmission*.

3.4.2.3 Terminate the Exchange in Progress

3.4.2.3.1 Conditions

Whatever the type of exchange, this process is executed for the following error situations:

- The Acceptor is unable to send a request or an advice message (see section 3.4.1.1).

This process is also executed after sending an *AcceptorReconciliationRequest* or an *AcceptorDiagnosticRequest* message for the following error cases:

- The Acceptor has not received a response message (see section 3.4.1.5).
- The message is not an acceptable response (see section 3.4.1.6), and matches a request in progress.
- The Acceptor is unable to process the response message (see section 3.4.1.7 a) and b)), and matches an advice in progress.

3.4.2.3.2 Action

The Acceptor terminates the exchange in progress, and notifies the application of the failure of the exchange (e.g. unable to go online).

3.4.2.4 Ignore the Error

3.4.2.4.1 Conditions

Whatever the type of exchange, this process is executed for the following circumstances:

- The message is not an acceptable response (see section 3.4.1.6), and could not be linked to the request or the advice in progress (unmatched response).
- The Acceptor is unable to process the response message (see section 3.4.1.7 a) and b)), and could not be linked to the request or the advice in progress (unmatched response).
- The response message arrives too late (see section 3.4.1.7 c)).

3.4.2.4.2 Action

The Acceptor does not take any action, including any termination of exchange in progress.

3.4.3 Acquirer Error Handling

Errors handling of the Acquirer follow the process below:

- 1) Error case detection, as described in the section 3.4.1 Error Cases.
- 2) Error processing at a protocol level, depending on the type of exchange.
- 3) Complete the exchange sending the appropriate response or
- 4) Terminate the exchange and notify the error to the application.

Following sections describe each type of error handling of the Acquirer and the conditions under which this process has to be performed.

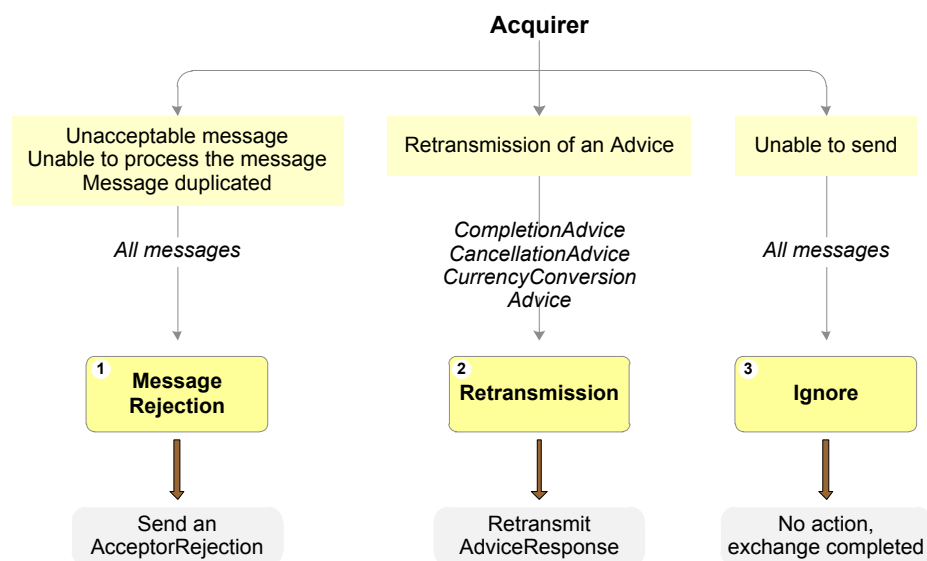


Figure 51: Acquirer Error Handling in a Message Exchange

3.4.3.1 Message Rejection

3.4.3.1.1 Conditions

Whatever the type of request or advice message, this error handling process is executed for the following error cases:

- The Acquirer receives an unacceptable message (see section 3.4.1.2),
- The Acquirer is unable to process the message (see section 3.4.1.3),
- The Acquirer receives a duplicated message (see section 3.4.1.8).

3.4.3.1.2 Action

The Acquirer sends an AcceptorRejection message with the appropriate content, including the RejectReason value related to the type of error (see section 2.8 *Reject Message*).

3.4.3.2 Retransmission of the Response

3.4.3.2.1 Conditions

This process is executed when the Acceptor retransmits an AcceptorCompletionAdvice message or an AcceptorCancellationAdvice message (see section 3.3).

3.4.3.2.2 Action

The Acquirer sends an advice response in accordance to the section 3.3.2 *Message Retransmission*.

3.4.3.3 Ignore the Error

3.4.3.3.1 Conditions

Whatever the type of request or advice message, this process is executed for the following error case:

- The Acquirer is Unable to Send a Message (see section 3.4.1.4),

3.4.3.3.2 Action

The Acquirer does not take any action.

4 Messages and Usage

This chapter explains the usage, rules and conditions of presence for all data elements involved in normal payment transactions, and complements the Message Definition Report on the usage of message elements.

4.1 Configuration Parameters/ Condition of Presence

The "Rule" column contains the condition of presence for optional data elements, or some values for mandatory data elements. Those rules have been grouped in following classes:

- *Config*: the condition of presence and possibly the value of a component in the message depend on a configuration parameter of the Initiator of the message (e.g. the merchant common name – CommonName). The value may depend on the Recipient and/or the Acquirer of the transaction.
- *Appli*: the condition of presence of a component of the protocol depends on the payment application exclusively (e.g. the PIN – CardholderOnlinePIN).
- *Copy* : the value of the mandatory component or element is copied from a related message (e.g. the message exchange identification in the message response header – ExchangeIdentification – contains a copy of that in the request header)
- *CCopy*: this condition is specific to Advice message. The condition of presence and the value of a component of the Advice message complies to the following rules in that order:
 - In the case of presence of this component in the Authorisation Response message, the value of the component is copied from the Authorisation Response message ;
 - Or in case of presence of this component in the Authorisation Request message, the value of the component is copied from the Authorisation Request message ;
 - Or when no authorisation exchange preceded the completion one, the condition associated with the component of the authorisation applies also to the Completion Advice message.

The "Usage" column may contain some information about the condition of presence of an optional element:

- *Default*: a default value is associated with the component of the message. The absence of the component produces the same result as the presence of the component with the default value (e.g. the message item Environment.Acquirer.Identification.Type has the comment "default Acquirer" at the beginning of the Rule column, meaning that an absent Acquirer.Identification is to having the value "Acquirer").
- *Message-related*: the condition of presence of a component of the protocol depends on the presence or value of another component in a related message. This kind of condition is also expressed at the beginning of the "Usage" column (e.g. the encrypted alternative of the sensitive card data – ProtectedCardData – is present if the following condition is satisfied "if Card.PlainCardData absent").

The "Constraint" (Cstr) column contains information about conditional presence or interdependency between elements of the message:

- **'*'**: if a constraint exists directly on the presence or the value of an element (mark by a star '*')
- **'C<number>'**: if a constraint exists due to interdependency of elements of the messages (e.g. data element A must be present if data element B equals X). In this latter case the constraint is labelled as 'C' plus a number. These constraints may serve for semantic analysis.

4.2 Authorisation Messages

4.2.1 AcceptorAuthorisationRequest (caaa.001.001.06)

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			It conveys information related to the protocol management on a segment of the path from the Acceptor to the Acquirer:
2	MessageFunction	[1..1]		C1 C13	The only valid codes to request an authorisation for a normal payment are: <i>AuthorisationRequest</i> : Request without financial capture (<i>TransactionCapture</i> ="False") <i>FinancialAuthorisationRequest</i> : Request with financial capture (<i>TransactionCapture</i> ="True") (if an invalid value is received, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to "ParsingError")
2	ProtocolVersion	[1..1]		*	The current version is 6.0 (reference list of specification documents).
2	ExchangeIdentification	[1..1]			Identifier per <i>InitiatingParty/RecipientParty</i> and per pair of messages used to assign a response to a request and to identify duplicate messages. It may be used in combination with <i>CreationDateTime</i> to allow the Recipient to identify retransmissions. It may be a cyclic counter that increments by one with each new message, starting at 0.
2	CreationDateTime	[1..1]			Date and time of the creation of the message Time accuracy has to be at least tenth of a second.
2	InitiatingParty	[1..1]			Information used to identify the initiator of an exchange. The content is bilaterally agreed between <i>InitiatingParty</i> and <i>RecipientParty</i> .
3	Identification	[1..1]	Config		The value of this identifier is bilaterally agreed between <i>InitiatingParty</i> and <i>RecipientParty</i> . The Recipient of the message must validate without ambiguity the Initiator of the message.
3	Type	[0..1]	Config	*	Indicates the type of <i>InitiatingParty</i> , allowed values: <i>OriginatingPOI</i> : from POI Terminal to an Intermediary Agent or an Acquirer. <i>IntermediaryAgent</i> : from an Intermediary Agent to another Intermediary Agent or an Acquirer. <i>Acceptor</i> : from a POI server performing functions of the payment application.
3	Issuer	[0..1]	Config		Indicates the assigner of the Identification value of the <i>InitiatingParty</i> .
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
2	RecipientParty	[0..1]	Config		Information used to identify the recipient of an exchange. The structure and content is bilaterally agreed between <i>InitiatingParty</i> and <i>RecipientParty</i> .
3	Identification	[1..1]	Config		
3	Type	[0..1]	Config	*	Indicates the type of <i>RecipientParty</i> , allowed values: <i>IntermediaryAgent</i> : from POI System or an Intermediary Agent to an Intermediary Agent. <i>Acquirer</i> : from POI System or an Intermediary Agent to an Acquirer.
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
3	RemoteAccess	[0..1]			

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]	Config		see section 3.2 Traceability
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]	Config		
4	Country	[0..1]			
4	ShortName	[0..1]	Config		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	AuthorisationRequest	[1..1]		C1	The <i>Header.MessageFunction</i> must be "AuthorisationRequest" or "FinancialAuthorisationRequest". (In case of an invalid value, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to "ParsingError")
2	Environment	[1..1]			
3	Acquirer	[0..1]	Config		Acquirer configuration parameter defines if this component needs to be present.
4	Identification	[0..1]	Config		
5	Identification	[1..1]	Appli		Identification of the Acquirer or the Intermediary Agent determined by the application from the card and other data used in this transaction.
5	Type	[0..1]	Appli	*	default <i>Acquirer</i> Allowed values: <i>Acquirer</i> , <i>IntermediaryAgent</i>
5	Issuer	[0..1]		*	Allowed values: <i>Acquirer</i> and <i>IntermediaryAgent</i>
5	Country	[0..1]			Country of the Acquirer (must be ISO 3166-1 alpha-2 or alpha-3)
5	ShortName	[0..1]			Name of the Acquirer or Intermediary Agent (e.g. name of the bank or the processor)
4	ParametersVersion	[1..1]			This value can be used by the Acquirer or the Intermediary Agent to: - decline the request if the configuration parameters are obsolete, - checks if a request to the POI to initiate an update of its configuration parameters (populating TMSTrigger in the response) is necessary. This element may be filled with a configuration parameter (<i>DataSet.Identification.Version</i> of the TMS data set of the AcceptorConfigurationUpdate message)
3	Merchant	[0..1]		C2	Present if it contains any data.
4	Identification	[0..1]	Config		
5	Identification	[1..1]	Appli		Identification of the Merchant determined by the application profile.
5	Type	[0..1]	Appli	*	default <i>Merchant</i> Allowed values: <i>Merchant</i> , <i>Acceptor</i> , <i>IntermediaryAgent</i> <i>Merchant</i> may be used to mention the retailer

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
					headquarters name and/or details <i>Acceptor</i> is used to identify the actual POI entity processing the transaction (e.g. a subsidiary of the Merchant)
5	Issuer	[0..1]	Config	*	The party assigning the identification. Allowed values: <i>Acquirer</i> and <i>IntermediaryAgent</i>
5	ShortName	[0..1]	Config		Name of the merchant assigned by the <i>Acquirer</i> or <i>IntermediaryAgent</i> .
4	CommonName	[0..1]	Config		Name of the merchant as appearing on the receipt.
4	LocationCategory	[0..1]	Config		Code usually derived from the merchant contract. Indicates the type of location where the transaction took place (e.g. train, in-flight, nomadic, etc.).
4	LocationAndContact	[0..1]	Config		
5	PostalAddress	[0..1]			
6	AddressLine	[0..2]			
6	StreetName	[0..1]			
6	BuildingNumber	[0..1]			
6	PostCode	[0..1]			
6	TownName	[1..1]			
6	CountrySubDivision	[0..2]			
6	Country	[1..1]			
5	Email	[0..1]			
5	URLAddress	[0..1]			
5	Phone	[0..1]			
5	CustomerService	[0..1]			
5	AdditionalContactInformation	[0..1]			
4	SchemeData	[0..1]	Config		
3	POI	[1..1]			
4	Identification	[1..1]			Identification of a POI terminal or system.
5	Identification	[1..1]	Appli		Part of the Acquirer/IntermediaryAgent or Merchant configuration.
5	Type	[0..1]		*	default <i>OriginatingPOI</i> Allowed values: <i>OriginatingPOI</i> , <i>IntermediaryAgent</i>
5	Issuer	[0..1]	Config	*	Allowed values: <i>Merchant</i> , <i>Acquirer</i> and <i>IntermediaryAgent</i>
5	ShortName	[0..1]	Config		
4	SystemName	[0..1]	Config		Allows a fast identification of the POI type by the Acquirer to monitor the transaction.
4	GroupIdentification	[0..1]	Config		This identifier can be used as a way for a merchant to group a set of POI transactions for reconciliation.
4	Capabilities	[0..1]		C3	Present if it contains any data.
5	CardReadingCapabilities	[0..*]	Config		Capabilities available to the payment application.
5	CardholderVerification-Capabilities	[0..*]	Config		Capabilities available to the payment application.
5	PINLengthCapabilities	[0..1]			
5	ApprovalCodeLength	[0..1]			
5	MaxScriptLength	[0..1]			
5	CardCaptureCapable	[0..1]			
5	OnLineCapabilities	[0..1]	Config		Indicates whether the POI authorises transactions exclusively offline (<i>Offline</i>), exclusively online (<i>Online</i>) or only authorises online if required by the payment application (<i>Semi Offline</i>).
5	MessageCapabilities	[0..*]			
6	Destination	[1..*]			

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
6	AvailableFormat	[0..*]			
6	NumberOfLines	[0..1]			
6	LineWidth	[0..1]			
6	AvailableLanguage	[0..*]			
4	TimeZone	[0..1]			Time zone name as defined by IANA (Internet Assigned Numbers Authority) in the time zone data base. America/Chicago or Europe/Paris are examples of time zone names.
4	TerminalIntegration	[0..1]			Indicates the type of integration of the POI terminal in the sale environment. INDR Indoor Indoor terminal. IPMP Inside Pump Terminal incorporated in the pump dispensing petrol. MPOI Multiple POI Terminal Multiple terminals linked to a unique sale terminal. MPMP MultiplePump Outdoor terminal serving several petrol pumps. MSLE MultipleSale Terminal Terminal serving multiple sale terminals. SSLE Single SaleTerminal Terminal linked to a unique sale terminal. VNDG VendingMachine Terminal integrated in a vending machine
4	Component	[0..*]	Config / Appli		Information used by the Acquirer for : – The traceability of components used for the transaction. – Identifying the version of the component (this information may cause the TMSTrigger to be set in the response). – The approval of POI components.
5	Type	[1..1]		*	Components to be sent in the online authorisation are configured locally or by TMS configuration.
5	Identification	[1..1]			Identification of the component.
6	ItemNumber	[0..1]			
6	ProviderIdentification	[0..1]			Identifies the provider of the component class (it replaces the data element <i>ManufacturerIdentification</i> of version 1).
6	Identification	[0..1]			Identification of the component assigned by the provider (it replaces the data element <i>Model</i> of version 1).
6	SerialNumber	[0..1]			Serial number of the component if available.
5	Status	[0..1]			Actual status of the component.
6	VersionNumber	[0..1]			Current version of component that may include the release number.
6	Status	[0..1]			
6	ExpiryDate	[0..1]			
5	StandardCompliance	[0..*]			Identification of the standard for which the component complies with.
6	Identification	[1..1]			Identification of the standard.
6	Version	[1..1]			Version of the standard.
6	Issuer	[1..1]			Entity assigning the identification
5	Characteristics	[0..1]			Only used in TMS protocol.
6	Memory	[0..*]			
7	Identification	[1..1]			
7	TotalSize	[1..1]			
7	FreeSize	[1..1]			

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
7	Unit	[1..1]			
6	Communication	[0..*]			
7	CommunicationType	[1..1]			
7	RemoteParty	[1..*]			
7	Active	[1..1]			
6	SecurityAccessModules	[0..1]			
6	SubscriberIdentityModules	[0..1]			
6	KeyCheckValue	[0..1]			
6	KeyCharacteristic	[0..1]			
7	KeyIdentification	[1..1]			
7	KeyVersion	[1..1]			
7	SequenceNumber	[0..1]			
7	DerivationIdentification	[0..1]			
7	Type	[0..1]			
7	Function	[0..1]			
6	EncryptedKey	[0..1]			
5	Assessment	[0..*]			Only used in TMS protocol.
6	Type	[1..1]			
6	Assigner	[1..*]			
6	DeliveryDate	[0..1]			
6	ExpirationDate	[0..1]			
6	Number	[1..1]			
3	Card	[1..1]	Config		The Acquirer configuration indicates if either <i>ProtectedCardData</i> or <i>PlainCardData</i> is present (e.g. TMS parameter <i>AcquirerProtocolParameters.ProtectCardData</i>).
4	ProtectedCardData	[0..1]		C4	Present if Card.PlainCardData absent. Encryption of the PlainCardData component, including the envelope, using CMS ContentType.EnvelopedData..
4	PlainCardData	[0..1]		C4	Present if ProtectedCardData absent.
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
5	ServiceCode	[0..1]	Appli		
5	Track1	[0..1]			
5	Track2	[0..1]			
5	Track3	[0..1]			
5	CardholderName	[0..1]			
4	PaymentAccountReference	[0..1]			Unique reference to the card, used by both merchants and acquirers to link tokenised and non-tokenised transactions associated to the same underlying card.
4	IssuerBIN	[0..1]			Bank identifier number of the issuer for routing purpose
4	CardCountryCode	[0..1]	Appli		Alphabetic with 2 or 3 characters, or numeric code conforms to ISO 3166 – 1. Indicates the country of the card issuer.
4	CardCurrencyCode	[0..1]			Currency code of the card issuer (ISO 4217 numeric code).
4	CardProductProfile	[0..1]	Config		Defines the acceptance processing and rules performed by the POI, after analysis of the application profile. Assigned by the Acquirer.
4	CardBrand	[0..1]	Appli		Brand name of the card or the scheme.
4	InternationalCard	[0..1]			“True” if the card may be used abroad. Else “False” for domestic cards only.

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
4	AllowedProduct	[0..*]			Product that can be purchased with the card. The list of allowed products contained in some specific cards (eg. Petrol cards)
4	ServiceOption	[0..1]			Options to the service provided by the card.
4	AdditionalCardData	[0..1]	Appli		Additional data taken from specific card products.
3	CustomerDevice	[0..1]			Device used by the customer to perform the payment transaction
4	Identification	[0..1]			Identification of the customer device used for payment
4	Type	[0..1]			Type of the device in free text
4	Provider	[0..1]			Provider of the device
3	Wallet	[0..1]			Container for tenders used by the customer to perform the payment transaction.
4	Identification	[0..1]			Identification of the wallet used for payment
4	Type	[0..1]			Type of wallet in free text
4	Provider	[0..1]			Provider of the wallet
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenRequestor	[0..1]			
5	ProviderIdentification	[1..1]			
5	RequestorIdentification	[1..1]			
3	Cardholder	[0..1]		C5	Present if it contains any data.
4	Identification	[0..1]	Appli		For verification of the Cardholder identity. A Cardholder may be identified by more than one identification method.
5	DriverLicenseNumber	[0..1]			
5	DriverLicenseLocation	[0..1]			
5	DriverLicenseName	[0..1]			
5	DriverIdentification	[0..1]			
5	CustomerNumber	[0..1]			
5	SocialSecurityNumber	[0..1]			
5	AlienRegistrationNumber	[0..1]			
5	PassportNumber	[0..1]			
5	TaxIdentificationNumber	[0..1]			
5	IdentityCardNumber	[0..1]			
5	EmployerIdentificationNumber	[0..1]			
5	EmployeeIdentification-Number	[0..1]			
5	JobNumber	[0..1]			
5	Department	[0..1]			
5	EmailAddress	[0..1]			
5	DateAndPlaceOfBirth	[0..1]			
6	BirthDate	[1..1]			
6	ProvinceOfBirth	[0..1]			
6	CityOfBirth	[1..1]			
6	CountryOfBirth	[1..1]			
5	Other	[0..*]			
6	Identification	[1..1]			
6	IdentificationType	[1..1]			
4	Name	[0..1]	Appli		For verification of the Cardholder identity.
4	Language	[0..1]	Appli		Advise the Acquirer of the cardholder language so that messages to the Cardholder can be customized to that language.
4	BillingAddress	[0..1]			

<i>Lvl</i>	<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	AddressLine	[0..2]			
5	StreetName	[0..1]			
5	BuildingNumber	[0..1]			
5	PostCode	[0..1]			
5	TownName	[1..1]			
5	CountrySubDivision	[0..2]			
5	Country	[1..1]			
4	ShippingAddress	[0..1]			
5	AddressLine	[0..2]			
5	StreetName	[0..1]			
5	BuildingNumber	[0..1]			
5	PostCode	[0..1]			
5	TownName	[1..1]			
5	CountrySubDivision	[0..2]			
5	Country	[1..1]			
4	TripNumber	[0..1]			
4	Vehicle	[0..1]			
5	VehicleNumber	[0..1]			
5	TrailerNumber	[0..1]			
5	VehicleTag	[0..1]			
5	VehicleTagEntryMode	[0..1]			
5	UnitNumber	[0..1]			
5	ReplacementCar	[0..1]			
5	Odometer	[0..1]			
5	Hubometer	[0..1]			
5	TrailerHours	[0..1]			
5	ReferHours	[0..1]			
5	MaintenancelIdentification	[0..1]			
5	DriverOrVehicleCard	[0..1]			
6	PAN	[0..1]			
6	Track1	[0..1]			
6	Track2	[0..1]			
6	Track3	[0..1]			
6	AdditionalCardData	[0..*]			
6	EntryMode	[0..1]			
5	AdditionalVehicleData	[0..*]			
6	Type	[0..1]			
6	EntryMode	[0..1]			
6	Data	[1..1]			
4	Authentication	[0..*]	Appli		Method and data used or intended to be used for this transaction to authenticate the owner of the card.
5	AuthenticationMethod	[1..1]		C6 C7	Method used to authenticate the cardholder:
5	AuthenticationValue	[0..1]		C6	Present if <i>AuthenticationMethod</i> ="SecureCertificate" or "SecureNoCertificate". (e.g. Visa Cardholder Authentication Verification Value (CAVV) or Mastercard Accountholder Authentication Value (AVV).
5	ProtectedAuthenticationValue	[0..1]			

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
5	CardholderOnlinePIN	[0..1]		C7	Present if <i>AuthenticationMethod</i> ="PINOnline" PIN verification by the Issuer during an online authorisation.
6	EncryptedPINBlock	[1..1]			Transport the PIN in an encrypted form using the <i>EnvelopedData</i> CMS data structure.
6	PINFormat	[1..1]	Appli		
6	AdditionalInput	[0..1]	Appli		Additional information required for the PIN verification, (e.g. the driver number for some fleet cards).
5	CardholderIdentification	[0..1]			
5	AddressVerification	[0..1]		C8	Present if it contains any data For verifying the cardholder's billing address.
6	AddressDigits	[0..1]	Appli		Numerics from the cardholder's address excluding the postal code (i.e. house number).
6	PostalCodeDigits	[0..1]	Appli		Numerics from the cardholder's postal code.
5	AuthenticationType	[0..1]			Type of authentication for a given method - e.g. three-domain authentication, scheme-proprietary authentication, etc.
5	AuthenticationLevel	[0..1]			Level of authentication for a given type – e.g. value assigned by scheme rules or by bilateral agreements
5	AuthenticationResult	[0..1]			Result of authentication
5	AuthenticationAdditional-Information	[0..1]			Additional information related to the result of the authentication
4	TransactionVerificationResult	[0..*]			Result of performed verifications for the transaction. Several methods may have been used for verification
5	Method	[1..1]			Method of verification that has been performed.
5	VerificationEntity	[0..1]			Entity or device that has performed the verification
5	Result	[0..1]			Result of the verification.
5	AdditionalResult	[0..1]			Additional result of the verification
4	PersonalData	[0..1]	Appli		Not to be used for cardholder identification or authentication.
3	ProtectedCardholderData	[0..1]			
2	Context	[1..1]			
3	PaymentContext	[1..1]			
4	CardPresent	[0..1]		C9	default <i>True</i> Indicates whether the transaction has been initiated by a card physically present or not: Not present if <i>CardDataEntryMode</i> ="AccountData", Present or not if <i>CardDataEntryMode</i> ="Physical", Present for other values of <i>CardDataEntryMode</i> .
4	CardholderPresent	[0..1]		C10	default <i>True</i> Indicates whether the transaction has been initiated in presence of the cardholder or not (e.g. False for the 2 nd and subsequent payments on a recurring transaction).
4	AttendanceContext	[0..1]	Config	C10	If <i>CardholderPresent</i> is "True": <i>Attended</i> : an attendant is present and performs the financial transaction (face to face). <i>SemiAttended</i> : one attendant monitors several POIs, to offer assistance if needed. <i>Unattended</i> : an attendant is not present Otherwise the element is absent
4	TransactionEnvironment	[0..1]			default <i>Merchant</i> Information required by some Acquirers or card schemes: <i>Merchant</i> : POI is located at the premises of the merchant. <i>Private</i> : remote payment, not at the premises of the merchant <i>Public</i> : POI is located in a public area, not at the premises of the merchant.

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
4	TransactionChannel	[0..1]	Config		Information required by some Acquirers or card schemes for specific environments. <i>MailOrder</i> : services or good purchased by mail (written). <i>TelephoneOrder</i> : services or good purchased by phone (voice) <i>ElectronicCommerce</i> : services or good purchased by Internet (electronic) <i>TelevisionPayment</i> : services or good purchased by TV
4	AttendantMessageCapable	[0..1]		C11	default <i>True</i> Indicates to the Acquirer whether a message in the authorisation response can be displayed to the Cashier or not.
4	AttendantLanguage	[0..1]		C11	Present If <i>AttendantMessageCapable</i> is <i>True</i> Indicates to the Acquirer in the Authorisation Response message the language of the message to be displayed to the Cashier.
4	CardDataEntryMode	[1..1]		C9	The entry mode used to get the card data, and not the list of entry modes in case of fall-back. .
4	FallbackIndicator	[0..1]	Appli		default <i>False</i> Card data entry mode fallback..
4	SupportedOption	[0..*]			Payment options the card acceptor can support.
3	SaleContext	[0..1]		C12	Present if it contains any data. Information provided by the sale system (e.g. EPAS Retailer protocol).
4	SaleIdentification	[0..1]	Appli		
4	SaleReferenceNumber	[0..1]	Appli		
4	SaleReconciliationIdentification	[0..1]	Appli		
4	CashierIdentification	[0..1]	Appli		
4	ShiftNumber	[0..1]	Appli		
4	PurchaseOrderNumber	[0..1]			
4	InvoiceNumber	[0..1]			
4	DeliveryNoteNumber	[0..1]			
4	SponsoredMerchant	[0..*]			
5	CommonName	[1..1]			
5	Address	[0..1]			
5	CountryCode	[1..1]			
5	MerchantCategoryCode	[1..1]			
5	RegisteredIdentifier	[1..1]			
4	SplitPayment	[0..1]			
4	RemainingAmount	[0..1]			
4	AdditionalSaleData	[0..1]	Appli		
2	Transaction	[1..1]			
3	TransactionCapture	[1..1]		C13	If "True", this financial transaction must be captured for clearing by the Acquirer. <i>MessageFunction</i> value must be "AuthorisationRequest" if <i>TransactionCapture</i> = "False", "FinancialAuthorisationRequest" if <i>TransactionCapture</i> = "True".
3	TransactionType	[1..1]		C14	
3	AdditionalService	[0..*]			
3	ServiceAttribute	[0..1]			
3	MerchantCategoryCode	[1..1]			Code assigned by the Acquirer, containing the ISO18245-4 MCC code associated with the category of services or goods purchased in this transaction (the merchant can have several MCCs associated to its

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
					business).
3	CustomerConsent	[0..*]			This enables retailers, if they so wish, to clearly indicate whether the consent of the customer was explicitly obtained for a given service instead of being implicitly derived.
3	CardProgrammeProposed	[0..*]			The card program proposed by a retailer to a cardholder among a series of payment programmes offered by the retailer.
3	CardProgrammeApplied	[0..*]		*	The card program actually selected by the cardholder among the ones supported by the retailer and/or the one actually proposed to him. At most one <i>CardProgrammeApplied</i> should be provided.
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			Identification of the transaction assigned by the POI.
4	TransactionDateTime	[1..1]	Appli		UTC date and time with offset or local date time.
4	TransactionReference	[1..1]	Appli		Identification of the transaction that has to be unique in combination with <i>TransactionDateTime</i> for the merchant and the POI.
3	OriginalTransaction	[0..1]	Appli		Not used if <i>TransactionType</i> ="CardPayment", "DeferredPayment" or "CashBack", optional if <i>TransactionType</i> ="Refund"
4	SaleReferenceIdentification	[0..1]			
4	TransactionIdentification	[1..1]			
5	TransactionDateTime	[1..1]			
5	TransactionReference	[1..1]			
4	POIIdentification	[0..1]	Appli		
5	Identification	[1..1]			
5	Type	[0..1]			default <i>OriginatingPOI</i>
5	Issuer	[0..1]	Config		
5	ShortName	[0..1]	Config		
4	InitiatorTransactionIdentification	[0..1]			If present in the original transaction
4	RecipientTransactionIdentification	[0..1]			If present in the original transaction
4	TransactionType	[1..1]			
4	AdditionalService	[0..*]			
4	ServiceAttribute	[0..1]			
4	CardDataEntryMode	[0..1]			
4	TransactionResult	[0..1]	Appli		
5	AuthorisationEntity	[0..1]			
6	Identification	[0..1]			
6	Type	[1..1]			
6	Issuer	[0..1]	Copy		
6	Country	[0..1]			
6	ShortName	[0..1]	Copy		
5	ResponseToAuthorisation	[1..1]			
6	Response	[1..1]			
6	ResponseReason	[0..1]	Appli		
6	AdditionalResponse-Information	[0..1]			
5	AuthorisationCode	[0..1]	Appli		
3	InitiatorTransactionIdentification	[0..1]	Appli		A value provided by the card acceptor that is meaningful in the card acceptor's system and that the acquirer can use in referencing back to this transaction.
3	ReconciliationIdentification	[0..1]	Appli		Identification of the reconciliation period assigned by the POI to the transaction. Conforming to the configuration parameters (by the EPAS TMS configuration parameters

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
					ReconciliationByAcquirer and ReconciliationExchange), absent if: The acquirer assigns the reconciliation period, and provides it in the response, or There is no reconciliation exchange (e.g. capture in batch)
3	TransactionDetails	[1..1]			
4	Currency	[1..1]	Appli		Currency of TotalAmount and DetailedAmount if present
4	TotalAmount	[1..1]			Requested amount to be authorised, including DetailedAmount occurrences.
4	AmountQualifier	[0..1]		C14	default <i>Actual</i> Allowed values depends on the TransactionType value: "CardPayment": <i>Actual</i> "DeferredPayment": <i>Estimated</i> or <i>Maximum</i> "Reservation": <i>Estimated</i> , <i>Incremental</i> or <i>Decremental</i>
4	DetailedAmount	[0..1]	Appli		
5	AmountOfGoodAndServices	[0..1]			
5	CashBack	[0..1]			
5	Gratuity	[0..1]			
5	Fees	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Rebate	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	ValueAddedTax	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Surcharge	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
4	ValidityDate	[0..1]	Appli		
4	OnlineReason	[0..1]	Appli		Indicates to the Acquirer the primary reason why the transaction has been sent online by the Card Acceptor.
4	UnattendedLevelCategory	[0..1]	Appli		Some card schemes may require for unattended POI terminals that the application determines the category level for the transaction. The value of this data is set by the application in compliance with the rules of the card scheme.
4	AccountType	[0..1]	Appli		default <i>Default</i> Allows a cardholder to select the type of account used for the transaction.
4	CurrencyConversionResult	[0..1]			
5	AcceptedByCardholder	[0..1]			
5	Conversion	[0..1]			
6	CurrencyConversion-Identification	[0..1]			
6	TargetCurrency	[1..1]			
7	AlphaCode	[1..1]			
7	NumericCode	[1..1]			
7	Decimal	[1..1]			
7	Name	[0..1]			
6	ResultingAmount	[1..1]			
6	ExchangeRate	[1..1]			

Lvl	AcceptorAuthorisationRequest	Mult.	Rule	Cstr	Usage
6	InvertedExchangeRate	[0..1]			
6	QuotationDate	[0..1]			
6	ValidUntil	[0..1]			
6	SourceCurrency	[1..1]			
7	AlphaCode	[1..1]			
7	NumericCode	[1..1]			
7	Decimal	[1..1]			
7	Name	[0..1]			
6	OriginalAmount	[1..1]			
7	ActualAmount	[0..1]			Actual amount to be converted
7	MinimumAmount	[0..1]			Minimum amount for conversion (in case of range of amounts)
7	MaximumAmount	[0..1]			Maximum amount for conversion (in case of range of amounts)
6	CommissionDetails	[0..*]			
7	Amount	[1..1]			
7	AdditionalInformation	[0..1]			
6	MarkUpDetails	[0..*]			
7	Rate	[1..1]			
7	AdditionalInformation	[0..1]			
6	DeclarationDetails	[0..1]			
7	Format	[0..1]			
7	MessageContent	[1..1]			
4	Instalment	[0..1]			
5	InstalmentPlan	[0..*]			
5	PlanIdentification	[0..1]			
5	SequenceNumber	[0..1]			
5	PeriodUnit	[0..1]			
5	InstalmentPeriod	[0..1]			
5	TotalNumberOfPayments	[0..1]			
5	FirstPaymentDate	[0..1]			
5	TotalAmount	[0..1]			
5	FirstAmount	[0..1]			
5	Charges	[0..1]			
4	AggregationTransaction	[0..1]			
5	FirstPaymentDateTime	[0..1]			
5	LastPaymentDateTime	[0..1]			
5	NumberOfPayments	[0..1]			
5	IndividualPayment	[0..*]			
6	Amount	[1..1]			
6	DateTime	[1..1]			
6	CardDataEntryMode	[0..1]			
6	ICCRelatedData	[0..1]			
6	Label	[0..1]			
4	ProductCodeSetIdentification	[0..1]			
4	SaleItem	[0..*]			
5	ItemIdentification	[0..1]			
5	ProductCode	[1..1]			
5	AdditionalProductCode	[0..1]			

<i>Lvl</i>	<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	UnitOfMeasure	[0..1]			
5	ProductQuantity	[0..1]			
5	UnitPrice	[0..1]			
5	UnitPriceSign	[0..1]			
5	ProductAmount	[1..1]			
5	ProductAmountSign	[0..1]			
5	ValueAddedTax	[0..1]			
5	TaxType	[0..1]			
5	ProductDescription	[0..1]			
5	DeliveryLocation	[0..1]			
5	DeliveryService	[0..1]			
4	DeliveryLocation	[0..1]			
4	CardPaymentInvoice	[0..1]			
5	InvoiceHeader	[1..1]			
6	Identification	[1..1]			
6	TypeCode	[1..1]			
6	Name	[0..*]			
6	IssueDateTime	[1..1]			
6	Issuer	[0..1]			
7	PartyIdentification	[1..1]			See MDR for sub elements
7	LegalOrganisation	[0..1]			
8	Identification	[0..1]			
8	Name	[0..1]			
7	TaxParty	[0..*]			See MDR for sub elements
6	LanguageCode	[0..1]			
6	CopyIndicator	[0..1]			
6	DocumentPurpose	[0..1]			
6	IncludedNote	[0..*]			
7	InformationType	[1..1]			
7	InformationValue	[1..1]			
5	TradeAgreement	[1..1]			See MDR for sub elements
5	TradeDelivery	[1..1]			See MDR for sub elements
5	LineItem	[0..*]			See MDR for sub elements
4	ICCRelatedData	[0..1]	Appli		A sequence of one or more TLV data elements in accordance with ISO 7816-6, not in a specific order.
3	MerchantReferenceData	[0..1]			
3	AdditionalTransactionData	[0..*]	Appli		
2	SupplementaryData	[0..*]	Appli		
3	PlaceAndName	[0..1]			
3	Envelope	[1..1]			
1	SecurityTrailer	[0..1]			The MAC of the message body AuthorisationRequest (including the body envelope) held in the AuthenticatedData element of the CMS data structure.

4.2.1.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
--------------------------	-------------------	--------------------------

C1	When <i>AuthorisationRequest</i> is present the <i>Header.MessageFunction</i> must be <i>AuthorisationRequest</i> or <i>FinancialAuthorisationRequest</i>	<ul style="list-style-type: none"> Header.MessageFunction AuthorisationRequest
C2	If <i>AuthorisationRequest.Environment.Merchant</i> is present it must have at least one child	<ul style="list-style-type: none"> AuthorisationRequest.Environment.Merchant
C3	If <i>AuthorisationRequest.Environment.POI.Capabilities</i> is present it must have at least one child	<ul style="list-style-type: none"> AuthorisationRequest.Environment.POI.Capabilities
C4	Either <i>AuthorisationRequest.Environment.Card.ProtectedCardData</i> or <i>AuthorisationRequest.Environment.Card.PlainCardData</i> must be present.	<ul style="list-style-type: none"> AuthorisationRequest.Environment.Card.ProtectedCardData AuthorisationRequest.Environment.Card.PlainCardData
C5	If <i>AuthorisationRequest.Environment.Cardholder</i> is present it must have at least one child	<ul style="list-style-type: none"> AuthorisationRequest.Environment.Cardholder
C6	<i>AuthenticationValue</i> must be present if <i>AuthenticationMethod</i> is "SecureCertificate" or "SecureNoCertificate".	<ul style="list-style-type: none"> AuthorisationRequest.Environment.Cardholder.Authentication.AuthenticationMethod AuthorisationRequest.Environment.Cardholder.Authentication.AuthenticationValue
C7	<i>CardholderOnlinePIN</i> must be present if <i>AuthenticationMethod</i> is "PINOnline".	<ul style="list-style-type: none"> AuthorisationRequest.Environment.Cardholder.Authentication.AuthenticationMethod AuthorisationRequest.Environment.Cardholder.Authentication.CardholderOnlinePIN
C8	If <i>AddressVerification</i> is present it must have at least one child	<ul style="list-style-type: none"> AuthorisationRequest.Environment.Cardholder.Authentication.AddressVerification
C9	<i>CardPresent</i> must be absent if <i>CardDataEntryMode</i> ="AccountData", <i>CardPresent</i> must be present if <i>CardDataEntryMode</i> has any other value than "AccountData" and "Physical"	<ul style="list-style-type: none"> AuthorisationRequest.Context.PaymentContext.CardPresent AuthorisationRequest.Context.PaymentContext.CardDataEntryMode
C10	<i>AttendanceContext</i> must present only if <i>CardholderPresent</i> is "True" and has value "Attended", "SemiAttended" or "Unattended"	<ul style="list-style-type: none"> AuthorisationRequest.Context.AttendanceContext AuthorisationRequest.Context.CardholderPresent
C11	<i>AttendantLanguage</i> must be present If <i>AttendantMessageCapable</i> is "True"	<ul style="list-style-type: none"> AuthorisationRequest.Context.AttendantLanguage AuthorisationRequest.Context.AttendantMessageCapable
C12	If <i>SaleContext</i> is present it must have at least one child	<ul style="list-style-type: none"> AuthorisationRequest.Context.SaleContext
C13	<i>MessageFunction</i> value must be "AuthorisationRequest" if <i>TransactionCapture</i> = "False", "FinancialAuthorisationRequest" if <i>TransactionCapture</i> = "True".	<ul style="list-style-type: none"> Header.MessageFunction AuthorisationRequest.Transaction.TransactionCapture
C14	If <i>TransactionType</i> value is "CardPayment" or "Refund" then <i>AmountQualifier</i> value must be "Actual" or absent. If <i>TransactionType</i> value is "DeferredPayment" then <i>AmountQualifier</i> value must be either "Estimated" or "Maximum". If <i>TransactionType</i> value is "Reservation" then <i>AmountQualifier</i> value must be either "Estimated", "Incremental" or "Decremental"	<ul style="list-style-type: none"> AuthorisationRequest.Transaction.TransactionType AuthorisationRequest.Transaction.TransactionDetails.AmountQualifier

4.2.2 AcceptorAuthorisationResponse (caaa.002.001.06)

Lvl	AcceptorAuthorisationResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	MessageFunction	[1..1]		C1	The only valid codes in a normal payment response are: <i>AuthorisationResponse</i> : Response for <i>AuthorisationRequest</i> <i>FinancialAuthorisationResponse</i> : Response for <i>FinancialAuthorisationRequest</i>
2	ProtocolVersion	[1..1]	Copy		The Recipient Party has to adapt the <i>AcceptorAuthorisationResponse</i> format to the version of the Initiator sent in the <i>AuthroisationRequest</i> . If this version is not supported the Recipient must reject the request with <i>RejectReason</i> = <i>ProtocolVersion</i> .
2	ExchangeIdentification	[1..1]	Copy		
2	CreationDateTime	[1..1]			Date and time of the creation of the message response. Time accuracy has to be at least tenth of a second.
2	InitiatingParty	[1..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			Present and completed if present in the request. see section 3.2 Traceability
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]	Copy		
4	Country	[0..1]			
4	ShortName	[0..1]	Copy		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	AuthorisationResponse	[1..1]		C1	The Header.MessageFunction must be <i>AuthorisationResponse</i> or

Lvl	AcceptorAuthorisationResponse	Mult.	Rule	Cstr	Usage
					<i>FinancialAuthorisationResponse</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]	Appli		If the identification is present in the request: It must be present in the response, but it may be different. If the identification is absent in the request: It may be absent in the response (if the POI does not need it for capture, reconciliation, reporting, etc) It may be present in the response (if the choice is not made by the POI, and the POI needs it)
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Acquirer</i>
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			If the identification is present in the request: It must be present in the response, but it may be different from the request. If the identification is absent in the request: It may be absent in the response (if the POI don't need it for capture, reconciliation, report, etc)
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Merchant</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[1..1]	Appli		May be different from the request.
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	Card				
4	ProtectedCardData	[0..1]	Appli	-	- Encryption of the PlainCardData component, including the envelope, using CMS ContentType.EnvelopedData. Present if: -- the protection of sensitive card data is required (TMS parameter ProtectedCardData equal to True) and - the validation of card data by the acceptor is required. These card data could be a subset of those in the request
4	PlainCardData	[0..1]	Appli		Present if: -sensitive card data does not require a protection (TMS parameter ProtectedCardData equal to False) and -the validation of card data by the acceptor is required.
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			

Lvl	AcceptorAuthorisationResponse	Mult.	Rule	Cstr	Usage
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
4	PaymentToken	[0..1]			
3	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]	Copy		For verification.
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]	Appli		Used by the InitiatingParty when it refers back to the transaction of the RecipientParty (e.g. reconciliation mismatch). If supplied, it is mandatory in the completion and the batch.
3	ReconciliationIdentification	[0..1]	Copy or Appli		Conforming to the configuration parameters ReconciliationByAcquirer and ReconciliationExchange. Present if: The acquirer copies the value of the request assigned by the acceptor. The acquirer assigns the reconciliation period, and provides it in the response.
3	InterchangeData	[0..1]	Appli		Data received from a card scheme message and required by the card scheme to be sent in the financial capture.
3	TransactionDetails	[1..1]			
4	Currency	[1..1]	Copy		
4	TotalAmount	[1..1]			If the Response is <i>PartialApproved</i> , the TotalAmount (ie the amount authorised) must be different from the amount requested; otherwise it must be the same as in the request message.
4	DetailedAmount	[0..1]	Appli		
5	AmountOfGoodAndServices	[0..1]			
5	CashBack	[0..1]			
5	Gratuity	[0..1]			
5	Fees	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Rebate	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	ValueAddedTax	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Surcharge	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
4	InvoiceAmount	[0..1]			
5	Amount	[1..1]			
5	Label	[0..1]			
4	ValidityDate	[0..1]	Appli		
4	AccountType	[0..1]	Appli		If present in the request this field must contain a copy of the request value. If not present in the request, it may inform the cardholder about the account used for the transaction.

Lvl	AcceptorAuthorisationResponse	Mult.	Rule	Cstr	Usage
4	ICCRelatedData	[0..1]			A sequence of one or more TLV data elements in accordance with ISO 7816-6, not in a specific order.
3	MerchantReferenceData	[0..1]			
2	TransactionResponse	[1..1]			
3	AuthorisationResult	[1..1]			
4	AuthorisationEntity	[0..1]			Identifies the entity which initially sets the <i>Response</i> value (approves, declines, or generates a "technical error").
5	Identification	[0..1]			
5	Type	[1..1]			<i>Intermediary Agent, Acquirer</i> : conforming to the rules of the card scheme, the Acquirer authorises the transaction (e.g. because a stand-in process, or declined because of technical problem). <i>CardIssuer</i> : the standard case. <i>DelegateIssuer</i> : for this transaction, the Issuer delegated the authorising right to another entity, for some reason.
5	Issuer	[0..1]	Appli		
5	Country	[0..1]			
5	ShortName	[0..1]	Appli		
4	ResponseToAuthorisation	[1..1]			
5	Response	[1..1]		C6	<i>Declined</i> : authorisation is declined or the requested capture is not performed. <i>Approved</i> : authorisation is approved for the full amount requested, including capture if requested <i>PartialApproved</i> : same as Approved, with TotalAmount lower in the response than in the request. <i>TechnicalError</i> : The message could not be processed for technical reasons (e.g. timeout to the Issuer, HSM problem)
5	ResponseReason	[0..1]	Appli		Additional information related to the response.
5	AdditionalResponse-Information	[0..1]			
4	AuthorisationCode	[0..1]	Appli		This code proves the approval delivered by the authorising entity (e.g. ISO 8583 - Elem. 38 - approval code).
4	CompletionRequired	[0..1]			If the flag has the value "True", an AcceptorCompletionAdvice has to be sent after the end of the transaction (see Table 2 : List of Payment Cases for the details)
4	TMSTrigger	[0..1]	Appli		Present when the POI needs maintenance. The POI may require maintenance prior to the execution of a specific service.
5	TMSContactLevel	[1..1]		* C3	Allowed values: <i>Critical</i> : TMS to be contacted before the next transaction <i>ASAP</i> : TMS to be contacted as soon as possible (e.g. after reconciliation) <i>Date Time</i> : TMS to be contacted at the date and time provided in TMSContactDateTime
5	TMSIdentification	[0..1]	Appli	*	Must be present if TMSTrigger is not empty
5	TMSContactDateTime	[0..1]		C3	Present if TMSContactLevel = <i>Date Time</i>
3	TransactionVerificationResult	[0..*]			Present if data structure is not empty
4	Method	[1..1]			
4	VerificationEntity	[0..1]			
4	Result	[0..1]			
4	AdditionalResult	[0..1]			
3	AllowedProductCode	[0..*]			
4	ProductCode	[1..1]			
4	AdditionalProductCode	[0..1]			

Lvl	AcceptorAuthorisationResponse	Mult.	Rule	Cstr	Usage
3	NotAllowedProductCode	[0..*]			
4	ProductCode	[1..1]			
4	AdditionalProductCode	[0..1]			
3	AdditionalAvailableProduct	[0..*]			
4	ProductCode	[1..1]			
4	AdditionalProductCode	[0..1]			
4	AmountLimit	[0..1]			
4	QuantityLimit	[0..1]			
4	UnitOfMeasure	[0..1]			
3	Balance	[0..1]	Appli		
4	Amount	[1..1]			Currency may be provided as an XML Attribute. ¹²
4	Sign	[0..1]			
3	ProtectedBalance	[0..1]			
3	Action	[0..*]	Appli		Several combinations of actions may be sent in the response (e.g. CaptureCard plus a display message for the merchant, or Referral plus separate display and print messages for the cardholder and merchant)
4	ActionType	[1..1]		C5 C6	<p>Additional actions to complete the transaction:</p> <p><i>DisplayMessage</i>: Display a message.</p> <p><i>PrintMessage</i>: Print a message.</p> <p>One of the following actions may be sent, if the Response is "Declined":</p> <p><i>Busy</i>: Server busy. Try later.</p> <p><i>CaptureCard</i>: Capture card.</p> <p><i>ForbidOverride</i>: Payment application cannot offer to the merchant the possibility to override the transaction.</p> <p><i>IDRequired</i>: Additional identification required (passport, ID card, etc.).</p> <p><i>PINRetry</i>: PIN verification retry.</p> <p><i>PINLastTry</i>: Last PIN try.</p> <p><i>Referral</i>: Referral with a voice authorisation has to be performed.</p> <p><i>RequestData</i>: Request additional data through a displayed text and request confirmation by attendant.</p>
4	MessageToPresent	[0..1]		C5	if ActionType is <i>DisplayMessage</i> or <i>PrintMessage</i>
5	MessageDestination	[1..1]			
5	Format	[0..1]			
5	MessageContent	[1..1]			
5	MessageContentSignature	[0..1]	Appli		
3	CurrencyConversionEligibility	[0..1]			
4	CurrencyConversion-Identification	[0..1]			
4	TargetCurrency	[1..1]			
5	AlphaCode	[1..1]			
5	NumericCode	[1..1]			
5	Decimal	[1..1]			
5	Name	[0..1]			
4	ResultingAmount	[1..1]			
4	ExchangeRate	[1..1]			
4	InvertedExchangeRate	[0..1]			

¹² Currency will be converted to a XML element in version 7.0

<i>Lvl</i>	<i>AcceptorAuthorisationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	QuotationDate	[0..1]			
4	ValidUntil	[0..1]			
4	SourceCurrency	[1..1]			
5	AlphaCode	[1..1]			
5	NumericCode	[1..1]			
5	Decimal	[1..1]			
5	Name	[0..1]			
4	OriginalAmount	[1..1*]			
5	ActualAmount	[0..1]			Actual amount to be converted
5	MinimumAmount	[0..1]			Minimum amount for conversion (in case of range of amounts)
5	MaximumAmount	[0..1]			Maximum amount for conversion (in case of range of amounts)
4	CommissionDetails	[0..*]			
5	Amount	[1..1]			
5	AdditionalInformation	[0..1]			
4	MarkUpDetails	[0..*]			
5	Rate	[1..1]			
5	AdditionalInformation	[0..1]			
4	DeclarationDetails	[0..1]			
5	Format	[1..1]			
5	MessageContent	[1..1]			
2	SupplementaryData	[0..*]			
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body AuthorisationResponse including the body envelope (see chapter).

4.2.2.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>AuthorisationResponse</i> is present the <i>Header.MessageFunction</i> must be "AuthorisationResponse" or "FinancialAuthorisationResponse".	<ul style="list-style-type: none"> Header.MessageFunction AuthorisationResponse
C3	<i>TMSContactDateTime</i> must be present if <i>TMSContactLevel</i> = "DateTime"	<ul style="list-style-type: none"> AuthorisationResponse.TransactionResponse.TMSTrigger.TMSContactDateTime AuthorisationResponse.TransactionResponse.TMSTrigger.TMSContactDateTime
C5	<i>MessageToPresent</i> must be present if <i>ActionType</i> is "DisplayMessage" or "PrintMessage"	<ul style="list-style-type: none"> AuthorisationResponse.TransactionResponse.Action.ActionType AuthorisationResponse.TransactionResponse.Action.MessageToPresent
C6	One of the following actions may be sent, if the Response is "Declined": <i>Busy</i> : Server busy. Try later. <i>CaptureCard</i> : Capture card. <i>ForbidOverride</i> : Payment application cannot offer to the merchant the possibility to override the transaction. <i>IDRequired</i> : Additional identification required (passport, ID card, etc.).	<ul style="list-style-type: none"> AuthorisationResponse.TransactionResponse.ResponseToAuthorisation.Response AuthorisationResponse.TransactionResponse.Action.ActionType

	<p><i>PINRetry</i>: PIN verification retry.</p> <p><i>PINLastTry</i>: Last PIN try.</p> <p><i>Referral</i>: Referral with a voice authorisation has to be performed.</p> <p><i>RequestData</i>: Request additional data through a displayed text and request confirmation by attendant.</p>	
--	---	--

4.3 Completion Messages

4.3.1 AcceptorCompletionAdvice (caaa.003.001.06)

Lvl	AcceptorCompletionAdvice	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]		C1 C5	<p>The only valid codes to advise a normal payment are:</p> <p><i>CompletionAdvice</i>: Completion without financial capture (TransactionCapture=False, Reversal=False) or (TransactionCapture=False, Reversal=True, TransactionSuccess=True)</p> <p><i>FinancialCompletionAdvice</i>: Completion with financial capture (TransactionCapture=True, Reversal=False) or (TransactionCapture=True, Reversal=True, TransactionSuccess=True)</p> <p><i>ReversalAdvice</i>: Reversal of an authorisation without financial capture (TransactionCapture=False, Reversal=True, TransactionSuccess=False)</p> <p><i>FinancialReversalAdvice</i>: Reversal of a <i>FinancialAuthorisation</i> (TransactionCapture=True, Reversal=True, TransactionSuccess=False)</p> <p>(in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)</p>
2	ProtocolVersion	[1..1]		*	see AcceptorAuthorisationRequest
2	ExchangeIdentification	[1..1]			see AcceptorAuthorisationRequest
2	RetransmissionCounter	[0..1]			default 0 see 3.3 Message Retransmission
2	CreationDateTime	[1..1]			see AcceptorAuthorisationRequest
2	InitiatingParty	[1..1]			see AcceptorAuthorisationRequest
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
2	RecipientParty	[0..1]	Config		see AcceptorAuthorisationRequest
3	Identification	[0..1]			
3	Type	[1..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]	Config		
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	Type	[1..1]			
4	Issuer	[0..1]	Config		
4	Country	[0..1]			
4	ShortName	[0..1]	Config		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CompletionAdvice	[1..1]		C1	The Header.MessageFunction must be <i>CompletionAdvice</i> , <i>FinancialCompletionAdvice</i> , <i>ReversalAdvice</i> or <i>FinancialReversalAdvice</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
2	Environment	[1..1]			
3	Acquirer	[0..1]	Config		If Acquirer identification has been sent in the Authorisation response, this value has to be used in CompletionAdvice.
4	Identification	[0..1]			
5	Identification	[1..1]			
5	Type	[0..1]		*	default <i>Acquirer</i> see <i>AcceptorAuthorisationRequest</i>
5	Issuer	[0..1]			
5	Country	[0..1]			
5	ShortName	[0..1]			
4	ParametersVersion	[1..1]	CCopy		For online authorisation, the value has to be the same as that in the Authorisation.
3	Merchant	[0..1]	CCopy		Conditional copy for online transactions. Configurable for offline transactions.
4	Identification	[0..1]			
5	Identification	[1..1]			
5	Type	[0..1]			default <i>Merchant</i>
5	Issuer	[0..1]			
5	ShortName	[0..1]			
4	CommonName	[0..1]			
4	LocationCategory	[0..1]			
4	LocationAndContact	[0..1]	Config		
5	PostalAddress	[0..1]			
6	AddressLine	[0..2]			
6	StreetName	[0..1]			
6	BuildingNumber	[0..1]			
6	PostCode	[0..1]			
6	TownName	[1..1]			
6	CountrySubDivision	[0..2]			
6	Country	[1..1]			
5	Email	[0..1]			
5	URLAddress	[0..1]			
5	Phone	[0..1]			
5	CustomerService	[0..1]			
5	AdditionalContactInformation	[0..1]			
5	SchemeData	[0..1]			
4	POI	[1..1]			Copy from AcceptorAuthorisationRequest.
5	Identification	[1..1]			Copy from AcceptorAuthorisationResponse if any.

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	Identification	[1..1]			Identification of a POI terminal or system.
5	Type	[0..1]			default <i>OriginatingPOI</i>
5	Issuer	[0..1]			
4	ShortName	[0..1]			
4	SystemName	[0..1]			
4	GroupIdentification	[0..1]			
5	Capabilities	[0..1]			
5	CardReadingCapabilities	[0..1]			
5	CardholderVerification-Capabilities	[0..1]			
5	PINLengthCapabilities	[0..1]			
5	ApprovalCodeLength	[0..1]			
5	MaxScriptLength	[0..1]			
5	CardCaptureCapable	[0..1]			
5	OnLineCapabilities	[0..1]			
6	MessageCapabilities	[0..*]			
6	Destination	[1..*]			
6	AvailableFormat	[0..*]			
6	NumberOfLines	[0..1]			
6	LineWidth	[0..1]			
4	AvailableLanguage	[0..*]			
4	TimeZone	[0..1]			
4	TerminalIntegration	[0..1]			
4	Component	[0..*]			see <i>AcceptorAuthorisationRequest</i>
5	Type	[1..1]			
5	Identification	[1..1]			
6	ItemNumber	[0..1]			
6	ProviderIdentification	[0..1]			
6	Identification	[0..1]			
6	SerialNumber	[0..1]			
5	Status	[0..1]			
6	VersionNumber	[0..1]			
6	Status	[0..1]			
6	ExpiryDate	[0..1]			
5	StandardCompliance	[0..*]			
6	Identification	[1..1]			
6	Version	[1..1]			
6	Issuer	[1..1]			
5	Characteristics	[0..1]			
6	Memory	[0..*]			
7	Identification	[1..1]			
7	TotalSize	[1..1]			
7	FreeSize	[1..1]			
7	Unit	[1..1]			
6	Communication	[0..*]			
7	CommunicationType	[1..1]			
7	RemoteParty	[1..*]			
7	Active	[1..1]			
6	SecurityAccessModules	[0..1]			
6	SubscriberIdentityModules	[0..1]			

Lvl	AcceptorCompletionAdvice	Mult.	Rule	Cstr	Usage
6	KeyCheckValue	[0..1]			
6	KeyCharacteristic	[0..1]			
7	KeyIdentification	[1..1]			
7	KeyVersion	[1..1]			
7	SequenceNumber	[0..1]			
7	DerivationIdentification	[0..1]			
7	Type	[0..1]			
7	Function	[0..1]			
6	EncryptedKey	[0..1]			
5	Assessment	[0..*]			
6	Type	[1..1]			
6	Assigner	[1..*]			
6	DeliveryDate	[0..1]			
6	ExpirationDate	[0..1]			
6	Number	[1..1]			
3	Card	[1..1]			
4	ProtectedCardData	[0..1]			Only present if requested by the Acquirer configuration (e.g. TMS parameter <i>CardDataVerification</i> is "True" and <i>ProtectCardData</i> is "True") or <i>AcceptorAuthorisationResponse</i> was not received.
4	PlainCardData	[0..1]			Only present if requested by the Acquirer configuration (e.g. TMS parameter <i>CardDataVerification</i> is "True" and <i>ProtectCardData</i> is "False") or <i>AcceptorAuthorisationResponse</i> was not received.
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
5	ServiceCode	[0..1]	Appli		
5	Track1	[0..1]			
5	Track2	[0..1]			
5	Track3	[0..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	IssuerBIN	[0..1]			
4	CardCountryCode	[0..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	CardCurrencyCode	[0..1]			
4	CardProductProfile	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	CardBrand	[0..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	CardProductType	[0..1]			
4	AdditionalCardData	[0..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	CardProductSubType	[0..1]			
3	CustomerDevice	[0..1]			
4	Identification	[0..1]			
4	Type	[0..1]			
4	Provider	[0..1]			
3	Wallet	[0..1]			
4	Identification	[0..1]			
4	Type	[0..1]			
4	Provider	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	TokenRequestor	[0..1]			
5	ProviderIdentification	[1..1]			
5	RequestorIdentification	[1..1]			
4	TokenAssuranceLevel				
3	Cardholder	[0..1]		C2	Present if it contains any data.
4	Identification	[0..1]	Appli		If required for capture of the transaction.
5	DriverLicenseNumber	[0..1]			
5	DriverLicenseLocation	[0..1]			
5	DriverLicenseName	[0..1]			
5	DriverIdentification	[0..1]			
5	CustomerNumber	[0..1]			
5	SocialSecurityNumber	[0..1]			
5	AlienRegistrationNumber	[0..1]			
5	PassportNumber	[0..1]			
5	TaxIdentificationNumber	[0..1]			
5	IdentityCardNumber	[0..1]			
5	EmployerIdentificationNumber	[0..1]			
5	EmployeeIdentification-Number	[0..1]			
5	JobNumber	[0..1]			
5	Department	[0..1]			
5	EmailAddress	[0..1]			
5	DateAndPlaceOfBirth	[0..1]			
6	BirthDate	[1..1]			
6	ProvinceOfBirth	[0..1]			
6	CityOfBirth	[1..1]			
6	CountryOfBirth	[1..1]			
5	Other	[0..*]			
6	Identification	[1..1]			
6	IdentificationType	[1..1]			
4	Name	[0..1]			
4	Language	[0..1]			
4	BillingAddress	[0..1]			
5	AddressLine	[0..2]			
5	StreetName	[0..1]			
5	BuildingNumber	[0..1]			
5	PostCode	[0..1]			
5	TownName	[1..1]			
5	CountrySubDivision	[0..2]			
5	Country	[1..1]			
4	ShippingAddress	[0..1]			
5	AddressLine	[0..2]			
5	StreetName	[0..1]			
5	BuildingNumber	[0..1]			
5	PostCode	[0..1]			
5	TownName	[1..1]			
5	CountrySubDivision	[0..2]			
5	Country	[1..1]			
4	TripNumber	[0..1]			
4	Vehicle	[0..1]			
5	VehicleNumber	[0..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	TrailerNumber	[0..1]			
5	VehicleTag	[0..1]			
5	VehicleTagEntryMode	[0..1]			
5	UnitNumber	[0..1]			
5	ReplacementCar	[0..1]			
5	Odometer	[0..1]			
5	Hubometer	[0..1]			
5	TrailerHours	[0..1]			
5	ReferHours	[0..1]			
5	Maintenanceldentification	[0..1]			
5	DriverOrVehicleCard	[0..1]			
6	PAN	[0..1]			
6	Track1	[0..1]			
6	Track2	[0..1]			
6	Track3	[0..1]			
6	AdditionalCardData	[0..*]			
6	EntryMode	[0..1]			
5	AdditionalVehicleData	[0..*]			
6	Type	[0..1]			
6	EntryMode	[0..1]			
6	Data	[1..1]			
4	PersonalData	[0..1]			
3	ProtectedCardholderData	[0..1]			
2	Context	[0..1]			
3	PaymentContext	[1..1]			
4	CardPresent	[0..1]			default <i>True</i> see <i>AcceptorAuthorisationRequest</i>
4	CardholderPresent	[0..1]			default <i>True</i> see <i>AcceptorAuthorisationRequest</i>
4	OnlineContext	[0..1]			default <i>True</i> The flag is set to "False" if the authorisation was offline without the need of sending an <i>AcceptorAuthorisationRequest</i> during the transaction.
4	AttendanceContext	[0..1]			default <i>Attended</i> see <i>AcceptorAuthorisationRequest</i>
4	TransactionEnvironment	[0..1]			default <i>Merchant</i> see <i>AcceptorAuthorisationRequest</i>
4	TransactionChannel	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	CardDataEntryMode	[1..1]			see <i>AcceptorAuthorisationRequest</i>
4	FallbackIndicator	[0..1]	Appli		default <i>False</i> see <i>AcceptorAuthorisationRequest</i>
4	SupportedOption	[0..*]			
3	SaleContext	[0..1]		C4	Present if it contains any data. see <i>AcceptorAuthorisationRequest</i>
4	SaleIdentification	[0..1]	Appli		
4	SaleReferenceNumber	[0..1]	Appli		
4	SaleReconciliationIdentification	[0..1]	Appli		
4	CashierIdentification	[0..1]	Appli		
4	ShiftNumber	[0..1]	Appli		
4	PurchaseOrderNumber	[0..1]			
4	InvoiceNumber	[0..1]			
4	DeliveryNoteNumber	[0..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	SponsoredMerchant	[0..*]			
5	CommonName	[1..1]			
5	Address	[0..1]			
5	CountryCode	[1..1]			
5	MerchantCategoryCode	[1..1]			
5	RegisteredIdentifier	[1..1]			
4	SplitPayment	[0..1]			
4	RemainingAmount	[0..1]			
4	AdditionalSaleData	[0..1]	Appli		
2	Transaction	[1..1]	CCopy		
3	TransactionCapture	[0..1]	Config	C5	default <i>False</i> If True, this financial transaction must be captured by the Acquirer. TransactionCapture is True if MessageFunction value is <i>FinancialCompletionAdvice</i> or <i>FinancialReversalAdvice</i> and False otherwise.
3	TransactionType	[1..1]			
3	AdditionalService	[0..*]			<i>VoiceAuthorisation</i> if transaction approved by phone after <i>Referral</i> as ActionType in the declined authorisation
3	ServiceAttribute	[0..1]			
3	MerchantCategoryCode	[1..1]			see <i>AcceptorAuthorisationRequest</i>
3	CustomerConsent	[0..*]			This enables retailers, if they so wish, to clearly indicate whether the consent of the customer was explicitly obtained for a given service instead of being implicitly derived.
3	CardProgrammeProposed	[0..*]			The card program proposed by a retailer to a cardholder among a series of payment programmes offered by the retailer.
3	CardProgrammeApplied	[0..*]		*	The card program actually selected by the cardholder among the ones supported by the retailer and/or the one actually proposed to him. At most one <i>CardProgrammeApplied</i> should be provided.
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]	CCopy		Identification of the transaction assigned by the POI Same value as the AuthorisationRequest if any.
3	TransactionDateTime	[1..1]			
3	TransactionReference	[1..1]			
3	OriginalTransaction	[0..1]	Appli		
4	SaleReferenceIdentification	[0..1]			
4	TransactionIdentification	[1..1]			
5	TransactionDateTime	[1..1]			
5	TransactionReference	[1..1]			
4	POI	[0..1]			
5	Identification	[1..1]			
5	Type	[0..1]			
5	Issuer	[0..1]			
5	ShortName	[0..1]			
4	<i>InitiatorTransactionIdentification</i>	[0..1]			
4	<i>RecipientTransactionIdentification</i>	[0..1]			
4	TransactionType	[1..1]			
4	<i>AdditionalService</i>	[0..*]			
4	<i>ServiceAttribute</i>	[0..1]			
4	CardDataEntryMode	[0..1]			
4	<i>TransactionResult</i>	[0..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	AuthorisationEntity	[1..1]			
6	Identification	[0..1]			
6	Type	[1..1]			
6	Issuer	[0..1]			
6	Country	[0..1]			
6	ShortName	[0..1]			
5	ResponseToAuthorisation	[1..1]			
6	Response	[1..1]			
6	ResponseReason	[0..1]			
6	AdditionalResponse- Information	[0..1]			
5	AuthorisationCode	[0..1]	Appli		
3	TransactionSuccess	[1..1]		C6	<i>True</i> if the outcome of the transaction is successful (see section 5)
3	Reversal	[0..1]			default <i>False</i> If <i>True</i> , a reversal of the online authorisation is requested (see section 5).
3	MerchantOverride	[0..1]	Appli		default <i>False</i> <i>True</i> if the acceptor has forced the transaction to be successful (see section 5).
3	FailureReason	[0..*]	Appli.	C6	Mandatory if TransactionSuccess is <i>False</i> , or Reversal is <i>True</i> or MerchantOverride is <i>True</i> (see section 5). <i>CardDeclined</i> : Integrated circuit card declines the transaction after the authorisation. <i>CustomerCancel</i> : Customer cancellation, for example removing the chip card after sending the AcceptorAuthorisationRequest, but before the end of the transaction. <i>Malfunction</i> : Suspected malfunction (e.g. card reader defect, printer out of order). <i>OfflineDeclined</i> : Offline authorisation declined the transaction. <i>OnLineDeclined</i> : Online authorisation declined the transaction. <i>SuspectedFraud</i> : Card payment transaction failed because the merchant suspected a fraud. <i>Timeout</i> : Timeout waiting for response from the acquirer, or no response was received (for example connection release before receiving the response). <i>TooLateResponse</i> : Response to the authorisation received too late. <i>UnableToComplete</i> : Card acceptor device unable to complete transaction after the authorisation response (e.g. written signature invalid). <i>UnableToSend</i> : Unable to deliver the AcceptorAuthorisationRequest to the acquirer.
3	InitiatorTransactionIdentification	[0..1]	Appli		Copy from AuthorisationRequest if any.
3	RecipientTransactionIdentification	[0..1]	Appli		Copy from AuthorisationResponse if any.
3	ReconciliationIdentification	[0..1]	Appli		For online transactions: <ul style="list-style-type: none"> ▪ If the transaction is captured during the authorisation or by batch, copy from AuthorisationRequest. ▪ If the transaction is captured by the Completion, identification of the reconciliation period. The value may be different from the AcceptorAuthorisationRequest. <i>see AcceptorAuthorisationRequest</i> for offline transactions.
3	InterchangeData	[0..1]	Appli		Copy from AuthorisationResponse if any.
3	TransactionDetails	[1..1]			
4	Currency	[1..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	TotalAmount	[1..1]			Final amount of the payment transaction when <i>TransactionSuccess</i> is "True". If <i>TransactionSuccess</i> is "False": the amount of <i>AuthorisationResponse</i> for an online approval, the amount of <i>AuthorisationRequest</i> if unable to go online or timeout, the purchase amount for a declined offline authorisation
4	AmountQualifier	[0..1]	Appli		
4	DetailedAmount	[0..1]	Appli		
5	AmountOfGoodAndServices	[0..1]			
5	CashBack	[0..1]			
5	Gratuity	[0..1]			
5	Fees	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Rebate	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	ValueAddedTax	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Surcharge	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
4	RequestAmount	[0..1]			
4	AuthorisedAmount	[0..1]			
4	InvoiceAmount	[0..1]			
4	ValidityDate	[0..1]			
4	UnattendedLevelCategory	[0..1]	Appli		Present if <i>AttendanceContext</i> is <i>Unattended</i> see <i>AcceptorAuthorisationRequest</i>
4	AccountType	[0..1]	Appli		default <i>Default</i> see <i>AcceptorAuthorisationRequest</i>
4	CurrencyConversionResult	[0..1]			
5	AcceptedByCardholder	[0..1]			
5	Conversion	[0..1]			
6	CurrencyConversion-Identification	[0..1]			
6	TargetCurrency	[1..1]			
7	AlphaCode	[1..1]			
7	NumericCode	[1..1]			
7	Decimal	[1..1]			
7	Name	[0..1]			
6	ResultingAmount	[1..1]			
6	ExchangeRate	[1..1]			
6	InvertedExchangeRate	[0..1]			
6	QuotationDate	[0..1]			
6	ValidUntil	[0..1]			
6	SourceCurrency	[1..1]			
7	AlphaCode	[1..1]			
7	NumericCode	[1..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
7	Decimal	[1..1]			
7	Name	[0..1]			
6	OriginalAmount	[1..1]			
7	ActualAmount	[0..1]			
7	MinimumAmount	[0..1]			
7	MaximumAmount	[0..1]			
6	CommissionDetails	[0..*]			
7	Amount	[1..1]			
7	AdditionalInformation	[0..1]			
6	MarkUpDetails	[0..*]			
7	Rate	[1..1]			
7	AdditionalInformation	[0..1]			
6	DeclarationDetails	[0..1]			
7	Format	[0..1]			
7	MessageContent	[1..1]			
4	Instalment	[0..1]			
5	InstalmentPlan	[0..*]			
5	PlanIdentification	[0..1]			
5	SequenceNumber	[0..1]			
5	PeriodUnit	[0..1]			
5	InstalmentPeriod	[0..1]			
5	TotalNumberOfPayments	[0..1]			
5	FirstPaymentDate	[0..1]			
5	TotalAmount	[0..1]			
5	FirstAmount	[0..1]			
5	Charges	[0..1]			
4	AggregationTransaction	[0..1]			
5	FirstPaymentDateTime	[0..1]			
5	LastPaymentDateTime	[0..1]			
5	NumberOfPayments	[0..1]			
5	IndividualPayment	[0..*]			
6	Amount	[1..1]			
6	DateTime	[1..1]			
6	CardDataEntryMode	[0..1]			
6	ICCRelatedData	[0..1]			
6	Label	[0..1]			
4	ProductCodeSetIdentification	[0..1]			
4	SaleItem	[0..*]			
5	ItemIdentification	[0..1]			
5	ProductCode	[1..1]			
5	AdditionalProductCode	[0..1]			
5	UnitOfMeasure	[0..1]			
5	ProductQuantity	[0..1]			
5	UnitPrice	[0..1]			
5	UnitPriceSign	[0..1]			
5	ProductAmount	[1..1]			
5	ProductAmountSign	[0..1]			
5	ValueAddedTax	[0..1]			
5	TaxType	[0..1]			
5	ProductDescription	[0..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	DeliveryLocation	[0..1]			
5	DeliveryService	[0..1]			
4	DeliveryLocation	[0..1]			
4	CardPaymentInvoice	[0..1]			
5	InvoiceHeader	[1..1]			
6	Identification	[1..1]			
6	TypeCode	[1..1]			
6	Name	[0..*]			
6	IssueDateTime	[1..1]			
6	Issuer	[0..1]			
7	PartyIdentification	[1..1]			See MDR for sub elements
7	LegalOrganisation	[0..1]			
8	Identification	[0..1]			
8	Name	[0..1]			
7	TaxParty	[0..*]			See MDR for sub elements
6	LanguageCode	[0..1]			
6	CopyIndicator	[0..1]			
6	DocumentPurpose	[0..1]			
6	IncludedNote	[0..*]			
7	InformationType	[1..1]			<i>see AcceptorAuthorisationRequest</i>
7	InformationValue	[1..1]			
5	TradeAgreement	[1..1]			See MDR for sub elements
5	TradeDelivery	[1..1]			See MDR for sub elements
5	LineItem	[0..*]			See MDR for sub elements
4	ICCRelatedData	[0..1]	Appli		A sequence of one or more TLV data elements in accordance with ISO 7816-6
3	AuthorisationResult	[0..1]	Appli		Present for online authorisation if an AcceptorAuthorisationResponse has been received.
4	AuthorisationEntity	[0..1]	CCopy		<i>see AcceptorAuthorisationResponse</i> if received, otherwise absent.
5	Identification	[0..1]			
5	Type	[1..1]			
5	Issuer	[0..1]			
5	Country	[0..1]			
5	ShortName	[0..1]			
4	ResponseToAuthorisation	[1..1]			<i>see AcceptorAuthorisationResponse</i>
5	Response	[1..1]			
5	ResponseReason	[0..1]	Appli		
5	AdditionalResponse-Information	[0..1]			
4	AuthorisationCode	[0..1]	Appli		Obtained from the corresponding AcceptorAuthorisationResponse if any. For Voice Authorisation the AuthorisationCode is obtained from the Acquirer (see section 6.1).
3	TransactionVerificationResult	[0..1]	Appli		
4	Method	[1..1]			
4	VerificationEntity	[0..1]			
4	Result	[0..1]			
4	AdditionalResult	[0..1]			
3	AdditionalTransactionData	[0..*]	Appli		
2	SupplementaryData	[0..*]	Appli		
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the

<i>Lvl</i>	<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
					AuthenticatedData alternative, containing the MAC of the message body CompletionAdvice including the body envelope.

4.3.1.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>CompletionAdvice</i> is present the <i>Header.MessageFunction</i> must be "CompletionAdvice", "FinancialCompletionAdvice", "ReversalAdvice" or "FinancialReversalAdvice".	<ul style="list-style-type: none"> Header.MessageFunction CompletionAdvice
C2	If <i>Cardholder</i> is present it must have at least one child	<ul style="list-style-type: none"> CompletionAdvice.Environment.CardHolder
C3		<ul style="list-style-type: none">
C4	If <i>SaleContext</i> is present it must have at least one child	<ul style="list-style-type: none"> CompletionAdvice.Context.SaleContext
C5	If <i>MessageFunction</i> value is "FinancialCompletionAdvice" or "FinancialReversalAdvice" then <i>TransactionCapture</i> must be "True". Otherwise it must be "False".	<ul style="list-style-type: none"> Header.MessageFunction CompletionAdvice.Transaction.TransactionCapture
C6	<i>FailureReason</i> must be present if <i>TransactionSuccess</i> is "False".	<ul style="list-style-type: none"> CompletionAdvice.Transaction.TransactionSuccess CompletionAdvice.Transaction.FailureReason
C7	If <i>TransactionSuccess</i> is "True" then <i>TotalAmount</i> must be Final amount of the payment transaction. Otherwise, <i>TotalAmount</i> is: the amount of <i>AuthorisationResponse</i> for an online approval, the amount of <i>AuthorisationRequest</i> if unable to go online or timeout, the purchase amount for an offline authorisation	<ul style="list-style-type: none"> CompletionAdvice.Transaction.TransactionDetails.TotalAmount CompletionAdvice.Transaction.TransactionSuccess

4.3.2 AcceptorCompletionAdviceResponse (caaa.004.001.06)

Lvl	AcceptorCompletionAdviceResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]		C1	The only valid codes are: <i>CompletionAdviceResponse</i> : response for <i>CompletionAdvice</i> <i>FinancialCompletionAdviceResponse</i> : response for <i>FinancialCompletionAdvice</i> <i>ReversalAdviceResponse</i> : response for <i>ReversalAdvice</i> <i>FinancialReversalAdviceResponse</i> : response for <i>FinancialReversalAdvice</i> (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
2	ProtocolVersion	[1..1]		*	The Recipient Party has to adapt the response format to the version of the Initiator sent in the advice. If this version is not supported the Recipient must reject the advice with RejectReason = ProtocolVersion.
2	ExchangeIdentification	[1..1]	Copy		
2	RetransmissionCounter	[0..1]	CCopy		
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]	Copy		
3	Identification	[1..1]			
3	Type	[1..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[1..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			Present if present in the completion advice. see section 3.2 Traceability
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]	Copy		
4	Country	[0..1]	Config		
4	ShortName	[0..1]	Copy		
3	ProtocolName	[0..1]			

<i>Lvl</i>	<i>AcceptorCompletionAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CompletionAdviceResponse	[1..1]		C1	
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]			Copy from AcceptorCompletionAdvice if present
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Acquirer</i>
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			Copy from AcceptorCompletionAdvice if present
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Merchant</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[1..1]			Copy from AcceptorCompletionAdvice
4	Identification	[1..1]			
4	Type	[0..1]			default <i>OriginatingPOI</i>
4	Issuer	[0..1]	Config		
4	ShortName	[0..1]	Config		
3	Card				
4	ProtectedCardData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
4	PlainCardData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]			
5	EffectiveDate	[0..1]			
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			Copy from <i>AcceptorCompletionAdvice</i> if present
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]	Appli		Used by the InitiatingParty when it refers back to the transaction of the RecipientParty (e.g. reconciliation mismatch). If supplied, it is mandatory in the completion and the batch.
3	ReconciliationIdentification	[0..1]			see <i>AcceptorAuthorisationResponse</i>
3	Response	[1..1]			<i>Approved</i> : the CompletionAdvice is accepted. For other responses, an error resolution process has to be performed which is out of scope of the protocol.
2	TMSTrigger	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>

<i>Lvl</i>	<i>AcceptorCompletionAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
3	TMSContactLevel	[1..1]		C2	
3	TMSIdentification	[0..1]	Appli		
3	TMSContactDateTime	[0..1]		C2	Present if TMSContactLevel = <i>DateTime</i> see <i>AcceptorAuthorisationResponse</i>
2	SupplementaryData	[0..*]			
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body CompletionAdviceResponse including the body envelope, using the MAC key of the related AcceptorCompletionAdvice message.

4.3.2.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>CompletionAdviceResponse</i> is present the <i>Header.MessageFunction</i> must be "CompletionAdviceResponse", "FinancialCompletionAdviceResponse", "ReversalAdviceResponse", "ReversalAdvice", "FinancialReversalAdviceResponse"	<ul style="list-style-type: none"> Header.MessageFunction CompletionAdviceResponse
C2	<i>TMSContactDateTime</i> must be present if <i>TMSContactLevel</i> = "DateTime"	<ul style="list-style-type: none"> CompletionAdviceResponse.TMSTrigger.TMSContactDateTime CompletionAdviceResponse.TMSTrigger.TMSContactDateTime

4.4 Cancellation Messages

4.4.1 AcceptorCancellationRequest (caaa.005.001.06)

Lvl	AcceptorCancellationRequest	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]		C1	The only valid code is <i>CancellationRequest</i> (CCAQ) to request a cancellation for a previously completed transaction, where the original transaction could not retrieved at the POI. (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
2	ProtocolVersion	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	ExchangeIdentification	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	CreationDateTime	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	InitiatingParty	[1..1]			see <i>AcceptorAuthorisationRequest</i>
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]			
3	ShortName	[0..1]	Config		
2	RecipientParty	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
3	Identification	[1..1]	Config		
3	Type	[0..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]			
3	ShortName	[0..1]	Config		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]	Config		see section 3.2 Traceability
3	RelayIdentification	[1..1]	Config		
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]	Config		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]	Config		
3	TraceDateTimeOut	[1..1]			
1	CancellationRequest	[1..1]		C1	The Header.MessageFunction must be <i>CancellationRequest</i> .

<i>Lvl</i>	<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
					(if not the case, a Reject message is sent by the Recipient with RejectReason equal to "ParsingError")
2	Environment	[1..1]			<i>see AcceptorAuthorisationRequest</i>
3	Acquirer	[0..1]	Config		
4	Identification	[0..1]	Config		
5	Identification	[1..1]	Appli		
5	Type	[0..1]			default <i>Acquirer</i>
5	Issuer	[0..1]	Config		
5	Country	[0..1]			
5	ShortName	[0..1]	Config		
4	ParametersVersion	[1..1]			<i>see AcceptorAuthorisationRequest</i>
3	Merchant	[0..1]		C2	Present if it contains any data. <i>see AcceptorAuthorisationRequest</i>
4	Identification	[0..1]	Config		<i>see AcceptorAuthorisationRequest</i>
5	Identification	[1..1]	Appli		
5	Type	[0..1]			default <i>Merchant</i>
5	Issuer	[0..1]	Config		
5	ShortName	[0..1]	Config		
4	CommonName	[0..1]			
4	LocationCategory	[0..1]			
4	LocationAndContact	[0..1]			
5	PostalAddress	[0..1]			
6	AddressLine	[0..2]			
6	StreetName	[0..1]			
6	BuildingNumber	[0..1]			
6	PostCode	[0..1]			
6	TownName	[1..1]			
6	CountrySubDivision	[0..2]			
6	Country	[1..1]			
5	Email	[0..1]			
5	URLAddress	[0..1]			
5	Phone	[0..1]			
5	CustomerService	[0..1]			
5	AdditionalContactInformation	[0..1]			
4	SchemeData	[0..1]	Config		
3	POI	[1..1]			<i>see AcceptorAuthorisationRequest</i>
4	Identification	[1..1]			
5	Identification	[1..1]	Config		
5	Type	[0..1]			default <i>OriginatingPOI</i>
5	Issuer	[0..1]	Config		
5	ShortName	[0..1]	Config		
4	SystemName	[0..1]	Config		<i>see AcceptorAuthorisationRequest</i>
4	GroupIdentification	[0..1]	Config		<i>see AcceptorAuthorisationRequest</i>
4	Capabilities	[0..1]			Present if it contains any data <i>see AcceptorAuthorisationRequest</i>
5	CardReadingCapabilities	[0..*]	Config		
5	CardholderVerification-Capabilities	[0..*]	Config		
5	PINLengthCapabilities	[0..1]			
5	ApprovalCodeLength	[0..1]			
5	MaxScriptLength	[0..1]			

<i>Lvl</i>	<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	CardCaptureCapable	[0..1]			
5	OnLineCapabilities	[0..1]	Config		
5	MessageCapabilities	[0..*]			
6	Destination	[1..*]			
6	AvailableFormat	[0..*]			
6	NumberOfLines	[0..1]			
6	LineWidth	[0..1]			
6	AvailableLanguage	[0..*]			
4	TimeZone	[0..1]			
4	TerminalIntegration	[0..1]			
4	Component	[0..*]	Config		<i>see AcceptorAuthorisationRequest</i>
5	Type	[1..1]			
5	Identification	[1..1]			
6	ItemNumber	[0..1]			
6	ProviderIdentification	[0..1]			
6	Identification	[0..1]			
6	SerialNumber	[0..1]			
5	Status	[0..1]			
6	VersionNumber	[0..1]			
6	Status	[0..1]			
6	ExpiryDate	[0..1]			
5	StandardCompliance	[0..*]			
6	Identification	[1..1]			
6	Version	[1..1]			
6	Issuer	[1..1]			
5	Characteristics	[0..1]			
6	Memory	[0..*]			
7	Identification	[1..1]			
7	TotalSize	[1..1]			
7	FreeSize	[1..1]			
7	Unit	[1..1]			
6	Communication	[0..*]			
7	CommunicationType	[1..1]			
7	RemoteParty	[1..*]			
7	Active	[1..1]			
6	SecurityAccessModules	[0..1]			
6	SubscriberIdentify	[0..1]			
6	KeyCheckValue	[0..1]			
6	KeyCharacteristic	[0..1]			
7	KeyIdentification	[1..1]			
7	KeyVersion	[1..1]			
7	SequenceNumber	[0..1]			
7	Derivation- Identification	[0..1]			
7	Type	[0..1]			
7	Function	[0..1]			
6	EncryptedKey	[0..1]			
5	Assessment	[0..*]			
6	Type	[1..1]			
6	Assigner	[1..*]			

<i>Lvl</i>	<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
6	DeliveryDate	[0..1]			
6	ExpirationDate	[0..1]			
6	Number	[1..1]			
3	Card	[1..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	ProtectedCardData	[0..1]			see <i>AcceptorAuthorisationRequest</i>
4	PlainCardData	[0..1]			see <i>AcceptorAuthorisationRequest</i>
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
5	ServiceCode	[0..1]	Appli		
5	Track1	[0..1]			
5	Track2	[0..1]			
5	Track3	[0..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	IssuerBIN	[0..1]			
4	CardCountryCode	[0..1]	Appli		
4	CardCurrencyCode	[0..1]			
4	CardProductProfile	[0..1]	Config		
4	CardBrand	[0..1]	Appli		
4	CardProductType	[0..1]			
4	AdditionalCardData	[0..1]	Appli		
4	CardProductSubType	[0..1]			
3	CustomerDevice	[0..1]			
4	Identification	[0..1]			
4	Type	[0..1]			
4	Provider	[0..1]			
3	Wallet	[0..1]			
4	Identification	[0..1]			
4	Type	[0..1]			
4	Provider	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenRequestor	[0..1]			
5	ProviderIdentification	[1..1]			
5	RequestorIdentification	[1..1]			
4	TokenAssuranceLevel	[0..1]			
2	Context	[1..1]			
3	PaymentContext	[1..1]			Context of the cancellation transaction
4	CardPresent	[0..1]			default <i>True</i> see <i>AcceptorAuthorisationRequest</i>
4	CardholderPresent	[0..1]			default <i>True</i> see <i>AcceptorAuthorisationRequest</i>
4	AttendanceContext	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	TransactionEnvironment	[0..1]			default <i>Merchant</i> see <i>AcceptorAuthorisationRequest</i>
4	TransactionChannel	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	AttendantMessageCapable	[0..1]		C4	default <i>True</i> Indicates to the acquirer whether a message in the cancellation response can be displayed to the attendant.

<i>Lvl</i>	<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	AttendantLanguage	[0..1]		C4	Present If AttendantMessageCapable is <i>True</i> Indicates the language of message in the cancellation response to be displayed to the attendant
4	CardDataEntryMode	[1..1]			The entry mode used to get the card data for the cancellation, the same values as in the AcceptorAuthorisationRequest are allowed.
4	FallbackIndicator	[0..1]	Appli		default <i>False</i> Alternative for the card data entry mode of the cancellation transaction.
3	SaleContext	[0..1]	Appli	C5	Present if it contains any data. <i>see AcceptorAuthorisationRequest</i>
4	SaleIdentification	[0..1]	Appli		
4	SaleReferenceNumber	[0..1]	Appli		
4	SaleReconciliationIdentification	[0..1]	Appli		
4	CashierIdentification	[0..1]	Appli		
4	ShiftNumber	[0..1]	Appli		
4	PurchaseOrderNumber	[0..1]			
4	InvoiceNumber	[0..1]			
4	DeliveryNoteNumber	[0..1]			
4	SponsoredMerchant	[0..*]			
5	CommonName	[1..1]			
5	Address	[0..1]			
5	CountryCode	[1..1]			
5	MerchantCategoryCode	[1..1]			
5	RegisteredIdentifier	[1..1]			
4	SplitPayment	[0..1]			
4	RemainingAmount	[0..1]			
4	AdditionalSaleData	[0..1]	Appli		
2	Transaction	[1..1]			
3	TransactionCapture	[0..1]			Not used in this version
3	MerchantCategoryCode	[1..1]			<i>see AcceptorAuthorisationRequest</i>
3	CustomerConsent	[0..*]			This enables retailers, if they so wish, to clearly indicate whether the consent of the customer was explicitly obtained for a given service instead of being implicitly derived.
3	CardProgrammeProposed	[0..*]			The card program proposed by a retailer to a cardholder among a series of payment programmes offered by the retailer.
3	CardProgrammeApplied	[0..*]		*	The card program actually selected by the cardholder among the ones supported by the retailer and/or the one actually proposed to him. At most one <i>CardProgrammeApplied</i> should be provided.
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			Identification of the cancellation transaction assigned by the POI. This identification must be different from the original transaction that is being cancelled.
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	OriginalTransaction	[1..1]	Appli		Information about the transaction to be cancelled. The OriginalTransaction components must have the value of the related components of the transaction to cancel.
4	SaleReferenceIdentification	[0..1]			
4	TransactionIdentification	[1..1]	Appli		Identification of the transaction to cancel assigned by the POI.

<i>Lvl</i>	<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	TransactionDateTime	[1..1]			
5	TransactionReference	[1..1]			
4	POIIdentification	[0..1]	Appli		POI where the transaction to cancel was performed.
5	Identification	[1..1]			
5	Type	[0..1]			
5	Issuer	[0..1]			
5	ShortName	[0..1]			
4	InitiatorTransactionIdentification	[0..1]	Appli		Present if present in the transaction to cancel, with the same value.
4	RecipientTransactionIdentification	[0..1]	Appli		Present if present in the transaction to cancel, with the same value.
4	TransactionType	[1..1]	Copy		TransactionType of the original transaction.
4	AdditionalService	[0..*]	Copy		Cancellation of a transaction must cancel all associated additional services; partial cancellation is not permitted.
4	ServiceAttribute	[0..1]	Copy		Cancellation of the original transaction cancels the transaction with its service attributes.
4	CardDataEntryMode	[0..1]			
4	TransactionResult	[0..1]	Appli		TransactionResult of the transaction to cancel.
5	AuthorisationEntity	[0..1]	Appli		AuthorisationEntity which has approved the transaction to be cancelled. <i>see AcceptorAuthorisationResponse</i>
6	Identification	[0..1]			
6	Type	[1..1]			
6	Issuer	[0..1]			
6	Country	[0..1]			
6	ShortName	[0..1]			
5	ResponseToAuthorisation	[1..1]			ResponseToAuthorisation of the transaction to be cancelled. <i>see AcceptorAuthorisationResponse</i>
6	Response	[1..1]			
6	ResponseReason	[0..1]	Appli		
6	AdditionalResponse-Information	[0..1]			
5	AuthorisationCode	[0..1]	Appli		AuthorisationCode of the transaction to be cancelled. <i>see AcceptorAuthorisationResponse</i>
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]			
3	ReconciliationIdentification	[0..1]	Appli		Identification of the reconciliation period assigned by the POI to the cancellation transaction. <i>see AcceptorAuthorisationRequest</i>
3	TransactionDetails	[1..1]			
4	Currency	[0..1]	Copy		Currency of TotalAmount of the transaction to cancel
4	TotalAmount	[1..1]	Copy		Amount to cancel which is always the TotalAmount of the approved transaction to cancel. Only the full amount of the transaction can be cancelled.
4	ValidityDate	[0..1]	Appli		
4	ICCRRelatedData	[0..1]	Appli		A sequence of one or more TLV data elements in accordance with ISO 7816-6.
3	AdditionalTransactionData	[0..*]	Appli		
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body CancellationRequest including the body envelope.

4.4.1.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>CancellationRequest</i> is present the <i>Header.MessageFunction</i> must be "CancellationRequest "	<ul style="list-style-type: none"> Header.MessageFunction CancellationRequest
C2	If <i>Merchant</i> is present it must have at least one child	<ul style="list-style-type: none"> CancellationRequest.Environment.Merchant
C3	If <i>Capabilities</i> is present it must have at least one child	<ul style="list-style-type: none"> CancellationRequest.Environment.POI.Capabilities
C4	If <i>AttendantMessageCapable</i> is "True" then <i>AttendantLanguage</i> must be present.	<ul style="list-style-type: none"> CancellationRequest..Context.PaymentContextAttendantMessageCapable CancellationRequest..Context.PaymentContextAttendantLanguage
C5	If <i>Salecontext</i> is present it must have at least one child	<ul style="list-style-type: none"> CancellationRequest..Context.SaleContext

4.4.2 AcceptorCancellationResponse (caaa.006.001.06)

Lvl	AcceptorCancellationResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]			The only valid codes in a response to a cancellation request is : <i>CancellationResponse</i>
2	ProtocolVersion	[1..1]	Copy	C1	The Recipient Party has to adapt the AcceptorCancellationResponse format to the version of the Initiator sent in the AcceptorCancellationRequest. If this version is not supported the Recipient must reject the request.
2	ExchangeIdentification	[1..1]	Copy		
2	CreationDateTime	[1..1]			see <i>AcceptorAuthorisationResponse</i>
2	InitiatingParty	[1..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]			
3	ShortName	[0..1]	Copy		
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]			
3	ShortName	[0..1]	Copy		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			Present if present in the request. see section 3.2 Traceability
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]	Copy		
4	Country	[0..1]			
4	ShortName	[0..1]	Copy		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CancellationResponse	[1..1]		C1	
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>

<i>Lvl</i>	<i>AcceptorCancellationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Acquirer</i>
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			see <i>AcceptorAuthorisationResponse</i>
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Merchant</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIdentification	[1..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	Card	[0..1]			
4	ProtectedCardData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
4	PlainCardData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]	Copy		see <i>AcceptorAuthorisationResponse</i>
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
3	ReconciliationIdentification	[0..1]	Copy or Appli		see <i>AcceptorAuthorisationResponse</i>
3	InterchangeData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
3	TransactionDetails	[1..1]			
4	Currency	[1..1]	Copy		
4	TotalAmount	[1..1]			Amount to cancel which is always the TotalAmount of the transaction to cancel. Only the full amount of the transaction can be cancelled.
4	ICCRRelatedData	[0..1]			
2	TransactionResponse	[1..1]			
3	AuthorisationResult	[1..1]			Outcome of the CancellationRequest
4	AuthorisationEntity	[0..1]			Qualify and allows recognition the entity which has authorises the cancellation offline or line: see <i>AcceptorAuthorisationResponse</i>

<i>Lvl</i>	<i>AcceptorCancellationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	Identification	[0..1]			
5	Type	[1..1]			
5	Issuer	[0..1]	Appli		
5	Country	[0..1]			
5	ShortName	[0..1]	Appli		
4	ResponseToAuthorisation	[1..1]			
5	Response	[1..1]		*	Valid values are: <i>Approved</i> : cancellation is approved, including capture if requested. <i>Declined</i> : cancellation is declined.
5	ResponseReason	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
5	AdditionalResponse-Information	[0..1]			
4	AuthorisationCode	[0..1]	Appli		
4	TMSTrigger	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
5	TMSContactLevel	[1..1]		C2	
5	TMSIdentification	[0..1]	Appli		
5	TMSContactDateTime	[0..1]		C2	Present if TMSContactLevel = <i>DateTime</i>
3	Action	[0..*]	Appli		see <i>AcceptorAuthorisationResponse</i>
4	ActionType	[1..1]		* C3	Allowed values: <i>Busy</i> , <i>CaptureCard</i> , <i>DisplayMessage</i> or <i>PrintMessage</i>
4	MessageToPresent	[0..1]		C3	if ActionType is <i>DisplayMessage</i> or <i>PrintMessage</i>
5	MessageDestination	[1..1]			
5	Format	[0..1]			
5	MessageContent	[1..1]			
5	MessageContentSignature	[0..1]	Appli		
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body CancellationResponse including the body envelope, using the MAC key of the related AcceptorCancellationRequest message.

4.4.2.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>CancellationResponse</i> is present the <i>Header.MessageFunction</i> must be "CancellationResponse"	<ul style="list-style-type: none"> Header.MessageFunction CancellationResponse
C2	<i>TMSContactDateTime</i> must be present if <i>TMSContactLevel</i> = "DateTime"	<ul style="list-style-type: none"> CancellationResponse.TMSTrigger.TMSContactDateTime CancellationResponse.TMSTrigger.TMSContactDateTime
C3	<i>MessageToPresent</i> must be present if <i>ActionType</i> is "DisplayMessage" or "PrintMessage"	<ul style="list-style-type: none"> CancellationResponse.TransactionResponse.Action.ActionType CancellationResponse.TransactionResponse.Action.MessageToPresent

4.4.3 AcceptorCancellationAdvice (caaa.007.001.06)

Lvl	AcceptorCancellationAdvice	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]		C1	The only valid code to advice a cancellation for a previously completed transaction is : <i>CancellationAdvice</i> (CCAV)
2	ProtocolVersion	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	ExchangeIdentification	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	RetransmissionCounter	[0..1]			default 0 see 3.3 Message Retransmission
2	CreationDateTime	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	InitiatingParty	[1..1]			see <i>AcceptorAuthorisationRequest</i>
3	Identification	[1..1]			
3	Type	[0..1]	Config		
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
2	RecipientParty	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
3	Identification	[1..1]			
3	Type	[0..1]	Config		
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]	Config		see section 3.2 Traceability
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]	Config		
4	Country	[0..1]	Config		
4	ShortName	[0..1]	Config		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CancellationAdvice	[1..1]		C1	The Header.MessageFunction must be <i>CancellationAdvice</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
2	Environment	[1..1]			
3	Acquirer	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>

<i>Lvl</i>	<i>AcceptorCancellationAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	Identification	[0..1]	Config		
5	Identification	[1..1]	Appli		
5	Type	[0..1]			default <i>Acquirer</i>
5	Issuer	[0..1]	Config		
5	Country	[0..1]			
5	ShortName	[0..1]	Config		
4	ParametersVersion	[1..1]			see <i>AcceptorAuthorisationRequest</i> and <i>AcceptorCompletionAdvice</i>
3	Merchant	[0..1]			see <i>AcceptorAuthorisationRequest</i>
4	Identification	[0..1]	Config		
5	Identification	[1..1]	Appli		
5	Type	[0..1]			default <i>Merchant</i>
5	Issuer	[0..1]	Config		
5	ShortName	[0..1]	Config		
4	CommonName	[0..1]	Config		
4	LocationCategory	[0..1]	Config		
4	LocationAndContact	[0..1]	Config		
5	PostalAddress	[0..1]			
6	AddressLine	[0..2]			
6	StreetName	[0..1]			
6	BuildingNumber	[0..1]			
6	PostCode	[0..1]			
6	TownName	[1..1]			
6	CountrySubDivision	[0..2]			
6	Country	[1..1]			
5	Email	[0..1]			
5	URLAddress	[0..1]			
5	Phone	[0..1]			
5	CustomerService	[0..1]			
5	AdditionalContactInformation	[0..1]			
4	SchemeData	[0..1]	Config		
3	POI	[1..1]			
4	Identification	[1..1]			see <i>AcceptorAuthorisationRequest</i>
5	Identification	[1..1]	Config		
5	Type	[0..1]			default <i>OriginatingPOI</i>
5	Issuer	[0..1]	Config		
5	ShortName	[0..1]	Config		
4	SystemName	[0..1]	Config		
4	GroupIdentification	[0..1]	Config		
4	Capabilities	[0..1]		C2	Present if it contains any data see <i>AcceptorAuthorisationRequest</i>
5	CardReadingCapabilities	[0..*]	Config		
5	CardholderVerification-Capabilities	[0..*]	Config		
5	PINLengthCapabilities	[0..1]			
5	ApprovalCodeLength	[0..1]			
5	MaxScriptLength	[0..1]			
5	CardCaptureCapable	[0..1]			
5	OnLineCapabilities	[0..1]	Config		
5	MessageCapabilities	[0..*]			

<i>Lvl</i>	<i>AcceptorCancellationAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
6	Destination	[1..*]			
6	AvailableFormat	[0..*]			
6	NumberOfLines	[0..1]			
6	LineWidth	[0..1]			
6	AvailableLanguage	[0..*]			
4	TimeZone	[0..1]			
4	TerminalIntegration	[0..1]			
4	Component	[0..*]	Config		<i>see AcceptorAuthorisationRequest</i>
5	Type	[1..1]			
5	Identification	[1..1]			
6	ItemNumber	[0..1]			
6	ProviderIdentification	[0..1]			
6	Identification	[0..1]			
6	SerialNumber	[0..1]			
5	Status	[0..1]			
6	VersionNumber	[0..1]			
6	Status	[0..1]			
6	ExpiryDate	[0..1]			
5	StandardCompliance	[0..*]			
6	Identification	[1..1]			
6	Version	[1..1]			
6	Issuer	[1..1]			
5	Characteristics	[0..1]			
6	Memory	[0..*]			
7	Identification	[1..1]			
7	TotalSize	[1..1]			
7	FreeSize	[1..1]			
7	Unit	[1..1]			
6	Communication	[0..*]			
7	CommunicationType	[1..1]			
7	RemoteParty	[1..*]			
7	Active	[1..1]			
6	SecurityAccessModules	[0..1]			
6	SubscriberIdentityModules	[0..1]			
6	KeyCheckValue	[0..1]			
6	KeyCharacteristic	[0..1]			
7	KeyIdentification	[1..1]			
7	KeyVersion	[1..1]			
7	SequenceNumber	[0..1]			
7	Derivation- Identification	[0..1]			
7	Type	[0..1]			
7	Function	[0..1]			
6	EncryptedKey	[0..1]			
5	Assessment	[0..*]			
6	Type	[1..1]			
6	Assigner	[1..*]			
6	DeliveryDate	[0..1]			
6	ExpirationDate	[0..1]			
6	Number	[1..1]			

<i>Lvl</i>	<i>AcceptorCancellationAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
3	Card	[1..1]	Appli		Card components retrieved from the transaction to cancel (e.g. log or receipt), or read from the card.
4	ProtectedCardData	[0..1]			see <i>AcceptorAuthorisationRequest</i>
4	PlainCardData	[0..1]			see <i>AcceptorAuthorisationRequest</i>
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
5	ServiceCode	[0..1]	Appli		
5	Track1	[0..1]			
5	Track2	[0..1]			
5	Track3	[0..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	IssuerBIN	[0..1]			
4	CardCountryCode	[0..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	CardCurrencyCode	[0..1]			
4	CardProductProfile	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	CardBrand	[0..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	CardProductType	[0..1]			
4	AdditionalCardData	[0..1]	Appli		see <i>AcceptorAuthorisationRequest</i>
4	CardProductSubType	[0..1]			
3	CustomerDevice	[0..1]			
4	Identification	[0..1]			
4	Type	[0..1]			
4	Provider	[0..1]			
3	Wallet	[0..1]			
4	Identification	[0..1]			
4	Type	[0..1]			
4	Provider	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenRequestor	[0..1]			
5	ProviderIdentification	[1..1]			
5	RequestorIdentification	[1..1]			
4	TokenAssuranceLevel	[0..1]			
2	Context	[0..1]		C3	Present if it contains any data
3	PaymentContext	[1..1]			Context of the cancellation transaction
4	CardPresent	[0..1]			default <i>True</i> see <i>AcceptorAuthorisationRequest</i>
4	CardholderPresent	[0..1]			default <i>True</i> see <i>AcceptorAuthorisationRequest</i>
4	OnlineContext	[0..1]			default <i>True</i> <i>True</i> if a <i>CancellationRequest</i> has been sent previously.
4	AttendanceContext	[0..1]	Config		default <i>Attended</i> see <i>AcceptorAuthorisationRequest</i>
4	TransactionEnvironment	[0..1]			default <i>Merchant</i> see <i>AcceptorAuthorisationRequest</i>
4	TransactionChannel	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	CardDataEntryMode	[1..1]			see <i>AcceptorAuthorisationRequest</i>
4	FallbackIndicator	[0..1]			default <i>False</i>

Lvl	AcceptorCancellationAdvice	Mult.	Rule	Cstr	Usage
					see AcceptorAuthorisationRequest
3	SaleContext	[0..1]		C4	Present if it contains any data. Context of the cancellation transaction see AcceptorAuthorisationRequest
4	SaleIdentification	[0..1]	Appli		
4	SaleReferenceNumber	[0..1]	Appli		
4	SaleReconciliationIdentification	[0..1]	Appli		
4	CashierIdentification	[0..1]	Appli		
4	ShiftNumber	[0..1]	Appli		
4	PurchaseOrderNumber	[0..1]			
4	InvoiceNumber	[0..1]			
4	DeliveryNoteNumber	[0..1]			
4	SponsoredMerchant	[0..*]			
5	CommonName	[1..1]			
5	Address	[0..1]			
5	CountryCode	[1..1]			
5	MerchantCategoryCode	[1..1]			
5	RegisteredIdentifier	[1..1]			
4	SplitPayment	[0..1]			
4	RemainingAmount	[0..1]			
4	AdditionalSaleData	[0..1]	Appli		
2	Transaction	[1..1]			
3	MerchantCategoryCode	[1..1]			see AcceptorAuthorisationRequest
3	CustomerConsent	[0..*]			
3	CardProgrammeProposed	[0..*]			
3	CardProgrammeApplied	[0..*]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			Identification of the cancellation transaction assigned by the POI. This identification must be different from the original transaction to cancel, but the same as in the AcceptorCancellationRequest if any.
4	TransactionDateTime	[1..1]			see AcceptorAuthorisationRequest
4	TransactionReference	[1..1]			see AcceptorAuthorisationRequest
3	OriginalTransaction	[0..1]			see AcceptorCancellationRequest
4	SaleReferenceIdentification	[0..1]			
4	TransactionIdentification	[1..1]	Appli		see AcceptorCancellationRequest
5	TransactionDateTime	[1..1]			
5	TransactionReference	[1..1]			
4	POIIdentification	[0..1]	Appli		see AcceptorCancellationRequest
5	Identification	[1..1]			
5	Type	[0..1]			
5	Issuer	[0..1]			
5	ShortName	[0..1]			
4	InitiatorTransactionIdentification	[0..1]	Appli		see AcceptorCancellationRequest
4	RecipientTransactionIdentification	[0..1]	Appli		see AcceptorCancellationRequest
4	TransactionType	[1..1]	Copy		see AcceptorCancellationRequest
4	AdditionalService	[0..*]			see AcceptorCancellationRequest
4	ServiceAttribute	[0..1]			see AcceptorCancellationRequest
4	CardDataEntryMode	[0..1]			
4	TransactionResult	[0..1]	Appli		see AcceptorCancellationRequest
5	AuthorisationEntity	[0..1]	Appli		AuthorisationEntity which has approved the transaction

Lvl	AcceptorCancellationAdvice	Mult.	Rule	Cstr	Usage
					to cancel. <i>see AcceptorAuthorisationResponse</i>
6	Identification	[0..1]			
6	Type	[1..1]			
6	Issuer	[0..1]			
6	Country	[0..1]			
6	ShortName	[0..1]			
5	ResponseToAuthorisation	[1..1]			ResponseToAuthorisation of the transaction to cancel. <i>see AcceptorAuthorisationResponse</i>
6	Response	[1..1]			
6	ResponseReason	[0..1]	Appli		
6	AdditionalResponse- Information	[0..1]			
5	AuthorisationCode	[0..1]	Appli		AuthorisationCode of the transaction to cancel. <i>see AcceptorAuthorisationResponse</i>
3	TransactionSuccess	[1..1]			
3	Reversal	[0..1]			Default: <i>False</i> <i>True</i> : no acceptable CancellationResponse message has been received, or cancellation couldn't be completed successfully after an approved cancellation request.
3	FailureReason	[0..*]		*	No acceptable CancellationResponse message has been received. Allowed values: <i>Malfunction</i> or <i>Timeout</i> (see <i>AcceptorCompletionAdvice</i>).
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]	Appli		<i>see AcceptorCancellationRequest</i>
3	ReconciliationIdentification	[0..1]	Appli		Identification of the reconciliation period assigned by the POI to the cancellation transaction. <i>see AcceptorCompletionAdvice</i>
3	InterchangeData	[0..1]	Appli		<i>see AcceptorCompletionAdvice</i>
3	TransactionDetails	[1..1]			
4	Currency	[1..1]	copy		Currency of TotalAmount of the transaction to cancel
4	TotalAmount	[1..1]	copy		TotalAmount of the transaction to cancel
4	ValidityDate	[0..1]	Appli		
4	ICCRRelatedData	[0..1]	Appli		A sequence of one or more TLV data elements in accordance with ISO 7816-6.
3	AuthorisationResult	[0..1]			Result of the outcome of the AcceptorCancellationResponse if any.
4	AuthorisationEntity	[0..1]			<i>see AcceptorCancellationResponse</i>
5	Identification	[0..1]			
5	Type	[1..1]			
5	Issuer	[0..1]			
5	Country	[0..1]			
5	ShortName	[0..1]			
4	ResponseToAuthorisation	[1..1]			<i>see AcceptorCancellationResponse</i>
5	Response	[1..1]			
5	ResponseReason	[0..1]			
5	AdditionalResponse- Information	[0..1]			
4	AuthorisationCode	[0..1]			
4	TMSTrigger	[0..1]			Not used in this message.
5	TMSContactLevel	[1..1]			
5	TMSIdentification	[0..1]			
5	TMSContactDateTime	[0..1]			

<i>Lvl</i>	<i>AcceptorCancellationAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
3	AdditionalTransactionData	[0..*]	Appli		
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body CancellationAdvice including the body envelope.

4.4.3.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>CancellationAdvice</i> is present the <i>Header.MessageFunction</i> must be "CancellationAdvice"	<ul style="list-style-type: none"> Header.MessageFunction CancellationAdvice
C2	If <i>Capabilities</i> is present it must have at least one child	<ul style="list-style-type: none"> CancellationAdvice.Environment.POI.Capabilities
C3	If <i>Context</i> is present it must have at least one child	<ul style="list-style-type: none"> CancellationAdvice.Context
C4	If <i>SaleContext</i> is present it must have at least one child	<ul style="list-style-type: none"> CancellationAdvice.Context.SaleContext

4.4.4 AcceptorCancellationAdviceResponse (caaa.008.001.06)

Lvl	AcceptorCancellationAdviceResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]			The only valid code in a response to a cancellation advice is: <i>CancellationAdviceResponse</i>
2	ProtocolVersion	[1..1]	Copy		
2	ExchangeIdentification	[1..1]	Copy		
2	ReTransmissionCounter	[0..1]	CCopy		
2	CreationDateTime	[1..1]			see <i>AcceptorCompletionAdviceResponse</i>
2	InitiatingParty	[1..1]	Copy		
3	Identification	[1..1]	Copy		
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			Present if present in the advice. see section 3.2 Traceability
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]	Copy		
4	Country	[0..1]	Config		
4	ShortName	[0..1]	Copy		
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	AcceptorCancellationAdviceResponse	[1..1]			
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]	Appli		Copy from AcceptorCancellationAdvice if present
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Acquirer</i>

<i>Lvl</i>	<i>AcceptorCancellationAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			see <i>AcceptorCompletionAdviceResponse</i>
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Merchant</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[1..1]	Copy		
4	Identification	[1..1]			
4	Type	[0..1]			default <i>OriginatingPOI</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	Card	[0..1]			
4	ProtectedCardData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
4	PlainCardData	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]	Appli		
5	EffectiveDate	[0..1]	Appli		
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]	Copy		For verification.
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]	Appli		Used by the InitiatingParty when it refers back to the transaction of the RecipientParty (e.g. reconciliation mismatch). If supplied, it is mandatory in the completion and the batch.
3	ReconciliationIdentification	[0..1]			see <i>AcceptorAuthorisationResponse</i>
3	Response	[1..1]			see <i>AcceptorCompletionAdvice</i> .
2	TMSTrigger	[0..1]	Appli		see <i>AcceptorAuthorisationResponse</i>
3	TMSContactLevel	[1..1]			
3	TMSIdentification	[0..1]	Appli		
3	TMSContactDateTime	[0..1]			Present if TMSContactLevel = <i>DateTime</i> see <i>AcceptorAuthorisationResponse</i>
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body CancellationAdviceResponse including the body envelope, using the MAC key of the related AcceptorCancellationAdvice message.

4.5 Reconciliation Messages

4.5.1 AcceptorReconciliationRequest (caaa.009.001.06)

Lvl	AcceptorReconciliationRequest	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			Unless stated otherwise, see <i>AcceptorAuthorisationRequest</i> message header usage.
2	MessageFunction	[1..1]		C1	The only valid code to request the reconciliation is <i>ReconciliationRequest</i> .
2	ProtocolVersion	[1..1]			
2	ExchangeIdentification	[1..1]			
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
2	RecipientParty	[0..1]	Config		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	ReconciliationRequest	[1..1]		C1	The Header.MessageFunction must be <i>ReconciliationRequest</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)

<i>Lvl</i>	<i>AcceptorReconciliationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
2	Environment	[1..1]			
3	Acquirer	[1..1]			
4	Identification	[0..1]	Appli		
5	Identification	[1..1]			
5	Type	[1..1]			
5	Issuer	[0..1]			default <i>Acquirer</i>
5	Country				
5	ShortName	[0..1]			
4	ParametersVersion	[1..1]			
3	MerchantIdentification	[0..1]	Appli		Present if reconciliation groups transactions performed by the same Merchant. <i>see AcceptorAuthorisationRequest</i>
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			default <i>Merchant</i>
4	Country	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[0..1]	Appli		Absent if the reconciliation groups transactions from several POI terminals. This identification can also group all the POI terminals of a POI system. <i>see AcceptorAuthorisationRequest</i>
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			default <i>OriginatingPOI</i>
4	Country				
4	ShortName	[0..1]			
3	Component	[0..*]			Not used in this version of the protocol.
2	Transaction	[1..1]			
3	ClosePeriod	[0..1]			default <i>True</i> <i>True</i> : notifies that the reconciliation period must be closed after the reconciliation exchange. <i>False</i> : the current reconciliation period must stay open after this exchange.
3	ReconciliationTransaction-Identification	[1..1]			Identification of the reconciliation transaction.
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	ReconciliationIdentification	[1..1]			Identification of the reconciliation period. If <i>ClosePeriod</i> is <i>True</i> , <i>ReconciliationIdentification</i> identifies the current reconciliation period. If <i>ClosePeriod</i> is <i>False</i> , <i>ReconciliationIdentification</i> identifies the current or a closed reconciliation period.
3	TransactionTotals	[0..*]			<i>TransactionTotals</i> of the reconciliation period. <i>TransactionTotals</i> is absent if the reconciliation period contains no transactions.
4	POIGroupIdentification	[0..1]	Appli		Collection of transactions containing this value sent in the <i>POI.GroupIdentification</i> component of the message request and advice.
4	CardProductProfile	[0..1]	Appli		Collection of transactions containing this value sent in the <i>Card.CardProductProfile</i> component of the message request and advice.
4	Currency	[0..1]	Appli		Currency is present when <i>TransactionTotals</i> are computed per currency (TMS configuration parameter <i>AcquirerProtocolParameters.TotalsPerCurrency</i> is <i>True</i>).
4	Type	[1..1]		*	All the values are allowed: <i>Debit</i> : Debit transactions (<i>TransactionType</i> is

<i>Lvl</i>	<i>AcceptorReconciliationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
					<i>CardPayment</i> , <i>CashBack</i> , <i>CashAdvance</i> , <i>DeferredPayment</i>) during the reconciliation period. <i>DebitReverse</i> : Cancelled debit transactions. <i>Credit</i> : Credit transactions (TransactionType is <i>Refund</i>) during the reconciliation period. <i>CreditReverse</i> : Cancelled credit transactions. <i>Declined</i> : transactions declined online or offline. <i>Failed</i> : failed transactions.
4	TotalNumber	[1..1]		*	Total number of transactions for the related Type, Currency, CardProductProfile and POIGroupIdentification, performed by the InitiatingParty during the reconciliation period ReconciliationIdentification.
4	CumulativeAmount	[1..1]		*	Total amount of transactions for the related Type, Currency, CardProductProfile and POIGroupIdentification, performed by the InitiatingParty during the reconciliation period ReconciliationIdentification.
3	AdditionalTransactionData	[0..1]	Appli		
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body ReconciliationRequest including the body envelope.

4.5.1.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>ReconciliationRequest</i> is present the <i>Header.MessageFunction</i> must be "ReconciliationRequest"	<ul style="list-style-type: none"> Header.MessageFunction ReconciliationRequest

4.5.2 AcceptorReconciliationResponse (caaa.010.001.05)

Lvl	AcceptorReconciliationResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			Unless stated otherwise, see Authorisation Response message header usage.
2	MessageFunction	[1..1]		C1	The only valid code to request the reconciliation is ReconciliationResponse.
2	ProtocolVersion	[1..1]			
2	ExchangeIdentification	[1..1]	Copy		
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeout	[1..1]			
1	ReconciliationResponse	[1..1]		C1	The Header.MessageFunction must be ReconciliationResponse. (if not the case, a Reject message is sent by the RecipientParty with RejectReason equal to ParsingError)
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]	Copy		
4	Identification	[1..1]			
4	Type	[1..1]			default Acquirer

<i>Lvl</i>	<i>AcceptorReconciliationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]	CCopy		see <i>AcceptorCompletionAdviceResponse</i>
4	Identification	[1..1]			
4	Type	[1..1]			default <i>Merchant</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[0..1]	Copy		
4	Identification	[1..1]			
4	Type	[1..1]			default <i>OriginatingPOI</i>
4	Issuer	[0..1]			
4	ShortName	[0..1]			
2	TransactionResponse	[1..1]			
3	Response	[1..1]		*	The following codes are allowed: <i>Approved</i> : The reconciliation is accepted, totals of transactions performed by the Recipient are provided in the response, or not computed in real-time. <i>Declined</i> : Totals are different.
3	ResponseReason	[0..1]			One of the reasons defined in section 2.6.4 p. 73, if Response is not <i>Approved</i> .
3	AdditionalResponseInformation	[0..1]			
2	Transaction	[1..1]			
3	ClosePeriod	[0..1]	Copy		
3	ReconciliationTransaction-Identification	[1..1]	Copy		
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	ReconciliationIdentification	[1..1]			Copy if the value of the request is not 0, otherwise put the last used value of the ReconciliationIdentification.
3	TransactionTotals	[0..*]			<i>TransactionTotals</i> is absent if the reconciliation period contains no transactions, or the totals computation is not performed in real-time by the acquirer. The sequence of TransactionTotals components, if present, must be in the same order as the request, according to the value of Type, Currency, CardProductProfile and POIGroupIdentification. In case of discrepancy, additional TransactionTotals must be located at the end of the sequence.
4	POIGroupIdentification	[0..1]	Appli		see <i>AcceptorReconciliationRequest</i>
4	CardProductProfile	[0..1]	Appli		see <i>AcceptorReconciliationRequest</i>
4	Currency	[0..1]	Appli		see <i>AcceptorReconciliationRequest</i>
4	Type	[1..1]			
4	TotalNumber	[1..1]		*	Total number of transactions for the related Type, Currency, CardProductProfile and POIGroupIdentification, performed by the RecipientParty during the reconciliation period ReconciliationIdentification. The value can be zero.
4	CumulativeAmount	[1..1]		*	Sum of TotalAmount of transactions for the related Type, Currency, CardProductProfile and POIGroupIdentification, performed by the RecipientParty during the reconciliation period ReconciliationIdentification. The value can be zero.
3	AdditionalTransactionData	[0..1]	Appli		
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body ReconciliationResponse including the

<i>Lvl</i>	<i>AcceptorReconciliationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
					body envelope, using the MAC key of the related AcceptorReconciliationRequest message.

4.5.2.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>ReconciliationResponse</i> is present the <i>Header.MessageFunction</i> must be "ReconciliationResponse"	<ul style="list-style-type: none"> Header.MessageFunction ReconciliationResponse

4.6 Batch

4.6.1 AcceptorBatchTransfer (caaa.011.001.06)

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
1		Header	[1..1]			
2		DownloadTransfer	[1..1]			<i>True</i> if BatchTransfer/Transaction occurrences contain AuthorisationResponse. <i>False</i> otherwise
2		FormatVersion	[1..1]			Same value as ProtocolVersion <i>see AcceptorAuthorisationRequest</i>
2		ExchangeIdentification	[1..1]			Unique identifier for the InitiatingParty to detect duplication of the transfer for a period of time. It is a cyclic counter that increments by one with each new transfer between the InitiatingParty and the RecipientParty.
2		CreationDateTime	[1..1]			Date and time of the file creation. Time accuracy has to be at least tenth of a second.
2		InitiatingParty	[1..1]			<i>see AcceptorAuthorisationRequest</i>
3		Identification	[1..1]			The value of this identifier is bilaterally agreed between InitiatingParty and RecipientParty. The Recipient of the message must identify without ambiguity the Initiator of the message.
3		Type	[0..1]	Appli		Indicates the type of InitiatingParty: <i>Acceptor, Merchant, OriginatingPOI, IntermediaryAgent.</i>
3		Issuer	[0..1]	Config		Indicates the assigner for identifying the InitiatingParty.
3		Country	[0..1]	Config		
3		ShortName	[0..1]	Config		
2		RecipientParty	[0..1]	Config		Information used to identify the recipient of an exchange. The structure and content is bilaterally agreed between InitiatingParty and RecipientParty.
3		Identification	[1..1]			
3		Type	[0..1]			Indicates the type of RecipientParty: <i>Acceptor, Merchant, IntermediaryAgent, Acquirer, DelegatIssuer.</i>
3		Issuer	[0..1]	Config		
3		Country	[0..1]	Config		
3		ShortName	[0..1]	Config		
3		RemoteAccess	[0..1]			
4		Address	[1..*]			
5		NetworkType	[1..1]			
5		AddressValue	[1..1]			
4		UserName	[0..1]			
4		AccessCode	[0..1]			
4		ServerCertificate	[0..*]			
4		ServerCertificateIdentifier	[0..*]			
4		ClientCertificate	[0..*]			
4		SecurityProfile	[0..1]			
1		BatchTransfer	[1..1]			
2		TransactionTotals	[0..*]	Appli		<i>TransactionTotals</i> of the whole <i>BatchTransfer</i> . <i>TransactionTotals</i> is absent if the batch contains no transactions.
3		POIGroupIdentification	[0..1]	Appli		<i>see AcceptorReconciliationRequest</i>

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
3		CardProductProfile	[0..1]	Appli		see <i>AcceptorReconciliationRequest</i>
3		Currency	[0..1]			see <i>AcceptorReconciliationRequest</i>
3		Type	[1..1]			see <i>AcceptorReconciliationRequest</i>
3		TotalNumber	[1..1]			see <i>AcceptorReconciliationRequest</i>
3		CumulativeAmount	[1..1]			see <i>AcceptorReconciliationRequest</i>
2		DataSet	[0..*]			A data set may include both financial (completed transactions) and non financial transactions (uncompleted transactions).
3		DataSetIdentification	[1..1]			Unique identifier for each of the DataSet for the InitiatingParty
4		Name	[1..1]			Identifier of the DataSet for the InitiatingParty
4		Type	[1..1]			<i>BatchCapture</i>
4		Version	[0..1]			Not used in batch capture context
4		CreationDateTime	[1..1]			Date and time of the creation of the data set. Time accuracy has to be at least tenth of a second
3		Traceability	[0..*]	Config		An intermediary Agent must include its own traceability info, if traceability was present in the data set received and if the data set is forwarded with the same transactions.
4		RelayIdentification	[1..1]			Information used to identify the recipient of an exchange. The structure and content is bilaterally agreed between InitiatingParty and RecipientParty.
5		Identification	[1..1]			
5		Type	[1..1]			Indicates the type of RecipientParty: <i>Acceptor, Merchant, IntermediaryAgent, Acquirer, DelegateIssuer.</i>
5		Issuer	[0..1]	Config		
5		Country	[0..1]			
5		ShortName	[0..1]	Config		
4		ProtocolName	[0..1]			
4		ProtocolVersion	[0..1]			
4		TraceDateTimeln	[1..1]			
4		TraceDateTimeOut	[1..1]			
3		DataSetInitiator	[0..1]	Config		The initiating party and the dataset initiator can be the same entity. If not, it is possible for each data set to identify the entity that creates it by populating DataSetInitiator.
4		Identification	[1..1]			Information used to identify the initiator of a dataset. The structure and content is bilaterally agreed between InitiatingParty and RecipientParty.
4		Type	[0..1]			Indicates the type of DataSetInitiator <i>Acceptor, Merchant, OriginatingPOI, IntermediaryAgent.</i> default <i>OriginatingPOI</i>
4		Issuer	[0..1]			Indicates the assigner for identifying the DataSetInitiator. <i>Acceptor, Merchant, , IntermediaryAgent, acquirer.</i>
4		Country	[0..1]			
4		ShortName	[0..1]			
3		TransactionTotals	[1..*]			<i>TransactionTotals</i> of the <i>DataSet</i> .
4		POIGroupIdentification	[0..1]	Appli		see <i>AcceptorReconciliationRequest</i>
4		CardProductProfile	[0..1]	Config		see <i>AcceptorReconciliationRequest</i>
4		Currency	[0..1]	Appli		see <i>AcceptorReconciliationRequest</i>

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
4		Type	[1..1]			see <i>AcceptorReconciliationRequest</i>
4		TotalNumber	[1..1]			see <i>AcceptorReconciliationRequest</i>
4		CumulativeAmount	[1..1]			see <i>AcceptorReconciliationRequest</i>
3		CommonData	[0..1]			Data common to transactions of a DataSet may be factorised in CommonData to reduce the message length. All transactions of the DataSet inherit the data element value present in CommonData except if this data element is present in the occurrence of TransactionToCapture.
4		Environment	[0..1]			
5		Acquirer	[0..1]			
6		Identification	[1..1]			
7		Identification	[1..1]			
7		Type	[0..1]			
7		Issuer	[0..1]			
7		Country	[0..1]			
7		ShortName	[0..1]			
6		ParametersVersion	[0..1]			
5		Merchant	[0..1]			
6		Identification	[1..1]			
7		Identification	[1..1]			
7		Type	[0..1]			
7		Issuer	[0..1]			
7		ShortName	[0..1]			
6		CommonName	[0..1]			
6		LocationCategory	[0..1]			
6		Address	[0..1]			
6		<i>CountryCode</i>	[0..1]			
6		SchemeData	[0..1]			
5		POI	[0..1]		<i>C1</i>	
6		Identification	[1..1]			
7		Identification	[1..1]			
7		Type	[0..1]			
7		Issuer	[0..1]			
7		ShortName	[0..1]			
6		SystemName	[0..1]			
6		GroupIdentification	[0..1]			
6		Capabilities	[0..1]			
7		CardReadingCapabilities	[0..*]			
7		CardholderVerification-Capabilities	[0..*]			
7		PINLengthCapabilities	[0..1]			
7		ApprovalCodeLength	[0..1]			
7		MaxScriptLength	[0..1]			
7		CardCaptureCapable	[0..1]			
7		OnLineCapabilities	[0..1]			
7		MessageCapabilities	[0..*]			
8		Destination	[1..*]			
8		AvailableFormat	[0..*]			
8		NumberOfLines	[0..1]			
8		LineWidth	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		AvailableLanguage	[0..*]			
6		TimeZone	[0..1]			
6		TerminalIntegration	[0..1]			
6		Component	[0..*]			
7		Type	[1..1]			
7		Identification	[1..1]			
8		ItemNumber	[0..1]			
8		ProviderIdentification	[0..1]			
8		Identification	[0..1]			
8		SerialNumber	[0..1]			
7		Status	[0..1]			
8		VersionNumber	[0..1]			
8		Status	[0..1]			
8		ExpiryDate	[0..1]			
7		StandardCompliance	[0..*]			
8		Identification	[1..1]			
8		Version	[1..1]			
8		Issuer	[1..1]			
7		Characteristics	[0..1]			
8		Memory	[0..*]			
9		Identification	[1..1]			
9		TotalSize	[1..1]			
9		FreeSize	[1..1]			
9		Unit	[1..1]			
8		Communication	[0..*]			
9		CommunicationType	[1..1]			
9		RemoteParty	[1..*]			
9		Active	[1..1]			
8		SecurityAccessModules	[0..1]			
8		SubscriberIdentityModules	[0..1]			
8		KeyCheckValue	[0..1]			
8		KeyCharacteristic	[0..1]			
8		EncryptedKey	[0..1]			
7		Assessment	[0..*]			
8		Type	[1..1]			
8		Assigner	[1..*]			
8		DeliveryDate	[0..1]			
8		ExpirationDate	[0..1]			
8		Number	[1..1]			
4		Context	[0..1]			
5		PaymentContext	[0..1]			
6		CardPresent	[0..1]			
6		CardholderPresent	[0..1]			
6		OnlineContext	[0..1]			
6		AttendanceContext	[0..1]			
6		TransactionEnvironment	[0..1]			

Lvl	Or	AcceptorBatchTransfer	Mult.	Rule	Cstr	Type / Definition / Code List
6		TransactionChannel	[0..1]			
6		AttendantMessageCapable	[0..1]			
6		AttendantLanguage	[0..1]			
6		CardDataEntryMode	[0..1]			
6		FallbackIndicator	[0..1]			
6		SupportedOption	[0..*]			
5		SaleContext	[0..1]			
6		SaleIdentification	[0..1]			
6		SaleReferenceNumber	[0..1]			
6		SaleReconciliationIdentification	[0..1]			
6		CashierIdentification	[0..1]			
6		ShiftNumber	[0..1]			
6		PurchaseOrderNumber	[0..1]			
6		InvoiceNumber	[0..1]			
6		DeliveryNoteNumber	[0..1]			
6		SponsoredMerchant	[0..*]			
7		CommonName	[1..1]			
7		Address	[0..1]			
7		CountryCode	[1..1]			
7		MerchantCategoryCode	[1..1]			
7		RegisteredIdentifier	[1..1]			
6		SplitPayment	[0..1]			
6		RemainingAmount	[0..1]			
6		AdditionalSaleData	[0..1]			
4		TransactionType	[0..1]			see <i>AcceptorAuthorisationRequest</i> .
4		AdditionalService	[0..*]			
4		ServiceAttribute	[0..1]			
4		MerchantCategoryCode	[0..1]		C3	
4		ReconciliationIdentification	[0..1]			
4		Currency	[0..1]			
3		Transaction	[1..*]			Transaction of the data set. It must be a completion, a cancellation advice, an authorisation request or an authorisation response.
4	{Or	Completion	[1..1]			
5		TransactionSequenceCounter	[1..1]			Sequential number assigned for each transaction of the DataSet to identify the rejected transactions in the related <i>AcceptorBatchTransferResponse</i> . The <i>TransactionSequenceCounter</i> must be increased for each transaction in the data set.
5		Traceability	[0..*]	Config		An intermediary Agent must include its own traceability info, if present in the previous reorganisation of the batch file
6		RelayIdentification	[1..1]			
7		Identification	[1..1]			
7		Type	[1..1]			
7		Issuer	[0..1]	Config		
7		Country	[0..1]			
7		ShortName	[0..1]	Config		
6		ProtocolName	[0..1]			
6		ProtocolVersion	[0..1]			
6		TraceDateTimeln	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
6		TraceDateTimeOut	[1..1]			
5		Environment	[1..1]			
6		Acquirer	[0..1]	Config		see <i>AcceptorCompletionAdvice</i>
7		Identification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			default <i>Acquirer</i> see <i>AcceptorAuthorisationRequest</i>
8		Issuer	[0..1]			
8		Country	[0..1]			
8		ShortName	[0..1]			
7		ParametersVersion	[1..1]			To be optional in future version. see <i>AcceptorCompletionAdvice</i>
6		Merchant	[0..1]	CCopy		see <i>AcceptorCompletionAdvice</i>
7		Identification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			default <i>Merchant</i>
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		CommonName	[0..1]			
7		LocationCategory	[0..1]			
7		LocationAndContact	[0..1]	Config		
8		PostalAddress	[0..1]			
9		AddressLine	[0..2]			
9		StreetName	[0..1]			
9		BuildingNumber	[0..1]			
9		PostCode	[0..1]			
9		TownName	[1..1]			
9		CountrySubDivision	[0..2]			
9		Country	[1..1]			
8		Email	[0..1]			
8		URLAddress	[0..1]			
8		Phone	[0..1]			
8		CustomerService	[0..1]			
8		AdditionalContact- Information	[0..1]			
7		SchemeData	[0..1]			
6		POI	[0..1]		C1	Mandatory if absent from CommonData. see <i>AcceptorCompletionAdvice</i>
7		Identification	[1..1]			
8		Identification	[1..1]			
8		Type	[0..1]			default <i>OriginatingPOI</i>
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		SystemName	[0..1]			
7		GroupIdentifier	[0..1]			
7		Capabilities	[0..1]			
8		CardReadingCapabilities	[0..*]			
8		CardholderVerification- Capabilities	[0..*]			
8		PINLengthCapabilities	[0..1]			
8		ApprovalCodeLength	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		MaxScriptLength	[0..1]			
8		CardCaptureCapable	[0..1]			
8		OnLineCapabilities	[0..1]			
8		MessageCapabilities	[0..*]			
9		Destination	[1..*]			
9		AvailableFormat	[0..*]			
9		NumberOfLines	[0..1]			
9		LineWidth	[0..1]			
9		AvailableLanguage	[0..*]			
7		TimeZone	[0..1]			
7		TerminalIntegration	[0..1]			
7		Component	[0..*]			<i>see AcceptorAuthorisationRequest</i>
8		Type	[1..1]			
8		Identification	[1..1]			
9		ItemNumber	[0..1]			
9		ProviderIdentification	[0..1]			
9		Identification	[0..1]			
9		SerialNumber	[0..1]			
8		Status	[0..1]			
9		VersionNumber	[0..1]			
9		Status	[0..1]			
9		ExpiryDate	[0..1]			
8		StandardCompliance	[0..*]			
9		Identification	[1..1]			
9		Version	[1..1]			
9		Issuer	[1..1]			
8		Characteristics	[0..1]			
9		Memory	[0..*]			
10		Identification	[1..1]			
10		TotalSize	[1..1]			
10		FreeSize	[1..1]			
10		Unit	[1..1]			
9		Communication	[0..*]			
10		CommunicationType	[1..1]			
10		RemoteParty	[1..*]			
10		Active	[1..1]			
9		SecurityAccessModules	[0..1]			
9		SubscriberIdentityModules	[0..1]			
9		KeyCheckValue	[0..1]			
9		KeyCharacteristic	[0..1]			
9		EncryptedKey	[0..1]			
8		Assessment	[0..*]			
9		Type	[1..1]			
9		Assigner	[1..*]			
9		DeliveryDate	[0..1]			
9		ExpirationDate	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		Number	[1..1]			
6		Card	[1..1]	Appli		
7		ProtectedCardData	[0..1]			see AcceptorAuthorisationRequest
8		ContentType	[1..1]			
8		EnvelopedData	[1..1]			
7		PlainCardData	[0..1]			
8		PAN	[1..1]			
8		CardSequenceNumber	[0..1]	Appli		
8		EffectiveDate	[0..1]	Appli		
8		ExpiryDate	[1..1]			
8		ServiceCode	[0..1]	Appli		
8		Track1	[0..1]			
8		Track2	[0..1]			
8		Track3	[0..1]			
7		PaymentAccountReference	[0..1]			
7		MaskedPAN	[0..1]			
7		IssuerBIN				
7		CardCountryCode	[0..1]	Appli		see AcceptorAuthorisationRequest
7		CardCurrencyCode				
7		CardProductProfile	[0..1]	Config		see AcceptorAuthorisationRequest
7		CardBrand	[0..1]	Appli		see AcceptorAuthorisationRequest
7		CardProductType	[0..1]			
7		AdditionalCardData	[0..1]	Appli		see AcceptorAuthorisationRequest
7		CardProductSubType	[0..1]			
6		CustomerDevice	[0..1]			
7		Identification	[0..1]			
7		Type	[0..1]			
7		Provider	[0..1]			
6		Wallet	[0..1]			
7		Identification	[0..1]			
7		Type	[0..1]			
7		Provider	[0..1]			
6		PaymentToken	[0..1]			
7		TokenCharacteristic	[0..*]			
7		TokenRequestor	[0..1]			
8		ProviderIdentification	[1..1]			
8		RequestorIdentification	[1..1]			
7		TokenAssuranceLevel	[0..1]			
6		Cardholder	[0..1]			see AcceptorCompletionAdvice
7		Identification	[0..*]			
8		DriverLicenseNumber	[0..1]			
8		DriverLicenseLocation	[0..1]			
8		DriverLicenseName	[0..1]			
8		DriverIdentification	[0..1]			
8		CustomerNumber	[0..1]			
8		SocialSecurityNumber	[0..1]			
8		AlienRegistrationNumber	[0..1]			
8		PassportNumber	[0..1]			
8		TaxIdentificationNumber	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		IdentityCardNumber	[0..1]			
8		EmployerIdentification- Number	[0..1]			
8		EmployeeIdentification- Number	[0..1]			
8		JobNumber	[0..1]			
8		Department	[0..1]			
8		EmailAddress	[0..1]			
8		DateAndPlaceOfBirth	[0..1]			
9		BirthDate	[1..1]			
9		ProvinceOfBirth	[0..1]			
9		CityOfBirth	[1..1]			
9		CountryOfBirth	[1..1]			
8		Other	[0..*]			
9		Identification	[1..1]			
9		IdentificationType	[1..1]			
7		Name	[0..1]			
7		Language	[0..1]			
7		BillingAddress	[0..1]			
8		AddressLine	[0..2]			
8		StreetName	[0..1]			
8		BuildingNumber	[0..1]			
8		PostCode	[0..1]			
8		TownName	[1..1]			
8		CountrySubDivision	[0..2]			
8		Country	[1..1]			
7		ShippingAddress	[0..1]			
8		AddressLine	[0..2]			
8		StreetName	[0..1]			
8		BuildingNumber	[0..1]			
8		PostCode	[0..1]			
8		TownName	[1..1]			
8		CountrySubDivision	[0..2]			
8		Country	[1..1]			
7		TripNumber	[0..1]			
7		Vehicle	[0..1]			
8		VehicleNumber	[0..1]			
8		TrailerNumber	[0..1]			
8		VehicleTag	[0..1]			
8		VehicleTagEntryMode	[0..1]			
8		UnitNumber	[0..1]			
8		ReplacementCar	[0..1]			
8		Odometer	[0..1]			
8		Hubometer	[0..1]			
8		TrailerHours	[0..1]			
8		ReferHours	[0..1]			
8		MaintenanceIdentification	[0..1]			
8		DriverOrVehicleCard	[0..1]			
9		PAN	[0..1]			
9		Track1	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		Track2	[0..1]			
9		Track3	[0..1]			
9		AdditionalCardData	[0..1]			
9		EntryMode	[0..1]			
8		AdditionalVehicleData	[0..*]			
9		Type	[0..1]			
9		EntryMode	[0..1]			
9		Data	[1..1]			
7		PersonalData	[0..1]			
6		ProtectedCarholderData	[0..1]			
5		Context	[0..1]		C2	Present if it contains any data
6		PaymentContext	[0..1]			see <i>AcceptorCompletionAdvice</i>
7		CardPresent	[0..1]			
7		CardholderPresent	[0..1]			
7		OnlineContext	[0..1]			see <i>AcceptorCompletionAdvice</i> .
7		AttendanceContext	[0..1]			
7		TransactionEnvironment	[0..1]			
7		TransactionChannel	[0..1]			
7		AttendantMessageCapable	[0..1]			
7		AttendantLanguage	[0..1]			
7		CardDataEntryMode	[0..1]			
7		FallbackIndicator	[0..1]			
7		SupportedOption	[0..*]			
6		SaleContext	[0..1]			see <i>AcceptorCompletionAdvice</i>
7		SaleIdentification	[0..1]			
7		SaleReferenceNumber	[0..1]			
7		SaleReconciliation- Identification	[0..1]			
7		CashierIdentification	[0..1]			
7		ShiftNumber	[0..1]			
7		PurchaseOrderNumber	[0..1]			
7		InvoiceNumber	[0..1]			
7		DeliveryNoteNumber	[0..1]			
7		SponsoredMerchant	[0..*]			
8		CommonName	[1..1]			
8		Address	[0..1]			
8		CountryCode	[1..1]			
8		MerchantCategoryCode	[1..1]			
8		RegisteredIdentifier	[1..1]			
7		SplitPayment	[0..1]			
7		RemainingAmount	[0..1]			
7		AdditionalSaleData	[0..1]			
5		Transaction	[1..1]			
6		TransactionType	[0..1]			<i>CardPayment</i> or <i>Refund</i>
6		AdditionalService	[0..*]			see <i>AcceptorCompletionAdvice</i>
6		ServiceAttribute	[0..1]			Not used for card payment
6		MerchantCategoryCode	[0..1]		C3	Mandatory If absent from <i>CommonData</i> see <i>AcceptorCompletionAdvice</i>
6		CustomerConsent	[0..*]			
6		CardProgrammeProposed	[0..*]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
6		CardProgrammeApplied	[0..*]			
6		SaleReferenceIdentification	[0..1]			
6		TransactionIdentification	[1..1]			see <i>AcceptorCompletionAdvice</i>
7		TransactionDateTime	[1..1]			
7		TransactionReference	[1..1]			
6		OriginalTransaction	[0..1]	<i>Appli</i>		see <i>AcceptorCompletionAdvice</i>
7		SaleReferenceIdentification	[0..1]			
7		TransactionIdentification	[1..1]			
8		TransactionDateTime	[1..1]			
8		TransactionReference	[1..1]			
7		POIIdentification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		InitiatorTransaction- Identification	[0..1]			
7		RecipientTransaction- Identification	[0..1]			
7		TransactionType	[1..1]			
7		AdditionalService	[0..*]			
7		ServiceAttribute	[0..1]			
7		CardDataEntryMode	[0..1]			
7		TransactionResult	[0..1]			
8		AuthorisationEntity	[0..1]			
9		Identification	[0..1]			
9		Type	[1..1]			
9		Issuer	[0..1]			
9		Country	[0..1]			
9		ShortName	[0..1]			
8		ResponseToAuthorisation	[1..1]			
9		Response	[1..1]			
9		ResponseReason	[0..1]			
9		AdditionalResponse- Information	[0..1]			
8		AuthorisationCode	[0..1]			
6		TransactionSuccess	[1..1]			see <i>AcceptorCompletionAdvice</i>
6		Reversal	[0..1]			default <i>False</i> see <i>AcceptorCompletionAdvice</i>
6		MerchantOverride	[0..1]	<i>Appli</i>		default <i>False</i> see <i>AcceptorCompletionAdvice</i>
6		FailureReason	[0..*]	<i>Appli.</i>		see <i>AcceptorCompletionAdvice</i>
6		InitiatorTransactionIdentification	[0..1]	<i>Appli</i>		
6		RecipientTransaction- Identification	[0..1]	<i>Appli</i>		see <i>AcceptorCompletionAdvice</i>
6		ReconciliationIdentification	[0..1]	<i>Appli</i>		Not used for batch
6		InterchangeData	[0..1]	<i>Appli</i>		see <i>AcceptorCompletionAdvice</i>
6		TransactionDetails	[1..1]			
7		Currency	[0..1]			Mandatory if absent from <i>CommonData</i>
7		TotalAmount	[1..1]			
7		AmountQualifier	[0..1]	<i>Appli</i>		

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
7		DetailedAmount	[0..1]	Appli		
8		AmountOfGood- AndServices	[0..1]			
8		CashBack	[0..1]			
8		Gratuity	[0..1]			
8		Fees	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		Rebate	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		ValueAddedTax	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		Surcharge	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
7		RequestAmount	[0..1]			
7		AuthorisedAmount	[0..1]			
7		InvoiceAmount	[0..1]			
7		ValidityDate	[0..1]			
7		UnattendedLevelCategory	[0..1]	Appli		see <i>AcceptorCompletionAdvice</i>
7		AccountType	[0..1]	Appli		default <i>Default</i> see <i>AcceptorAuthorisationRequest</i>
7		CurrencyConversionResult	[0..1]			
8		AcceptedByCardholder	[0..1]			
8		Conversion	[0..1]			
9		CurrencyConversion- Identification	[0..1]			
9		TargetCurrency	[1..1]			
10		AlphaCode	[1..1]			
10		NumericCode	[1..1]			
10		Decimal	[1..1]			
10		Name	[0..1]			
9		ResultingAmount	[1..1]			
9		ExchangeRate	[1..1]			
9		InvertedExchangeRate	[0..1]			
9		QuotationDate	[0..1]			
9		ValidUntil	[0..1]			
9		SourceCurrency	[1..1]			
10		AlphaCode	[1..1]			
10		NumericCode	[1..1]			
10		Decimal	[1..1]			
10		Name	[0..1]			
9		OriginalAmount	[1..1]			
10		ActualAmount	[0..1]			
10		MinimumAmount	[0..1]			
10		MaximumAmount	[0..1]			
9		CommissionDetails	[0..*]			
10		Amount	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
10		Additional-Information	[0..1]			
9		MarkUpDetails	[0..*]			
10		Rate	[1..1]			
10		Additional-Information	[0..1]			
9		DeclarationDetails	[0..1]			
10		Format	[0..1]			
10		MessageContent	[1..1]			
7		Instalment	[0..1]			
8		InstalmentPlan	[0..*]			
8		PlanIdentification	[0..1]			
8		SequenceNumber	[0..1]			
8		PeriodUnit	[0..1]			
8		InstalmentPeriod	[0..1]			
8		TotalNumberOfPayments	[0..1]			
8		FirstPaymentDate	[0..1]			
8		TotalAmount	[0..1]			
8		FirstAmount	[0..1]			
8		Charges	[0..1]			
7		AggregationTransaction	[0..1]			
8		FirstPaymentDateTime	[0..1]			
8		LastPaymentDateTime	[0..1]			
8		NumberOfPayments	[0..1]			
8		IndividualPayment	[0..*]			
9		Amount	[1..1]			
9		DateTime	[1..1]			
9		CardDataEntryMode	[0..1]			
9		ICCRelatedData	[0..1]			
9		Label	[0..1]			
7		ProductCodeSetIdentification	[0..1]			
7		SaleItem	[0..*]			
8		ItemIdentification	[0..1]			
8		ProductCode	[1..1]			
8		AdditionalProductCode	[0..1]			
8		UnitOfMeasure	[0..1]			
8		ProductQuantity	[0..1]			
8		UnitPrice	[0..1]			
8		UnitPriceSign	[0..1]			
8		ProductAmount	[1..1]			
8		ProductAmountSign	[0..1]			
8		ValueAddedTax	[0..1]			
8		TaxType	[0..1]			
8		ProductDescription	[0..1]			
8		DeliveryLocation	[0..1]			
8		DeliveryService	[0..1]			
7		DeliveryLocation	[0..1]			
7		CardPaymentInvoice	[0..1]			
8		InvoiceHeader	[1..1]			
9		Identification	[1..1]			See MDR for sub elements
9		TypeCode	[1..1]			
9		Name	[0..*]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		IssueDateTime	[1..1]			
9		Issuer	[0..1]			
10		PartyIdentification	[1..1]			See MDR for sub elements
10		LegalOrganisation	[0..1]			
11		Identification	[0..1]			
11		Name	[0..1]			
10		TaxParty	[0..*]			See MDR for sub elements
9		LanguageCode	[0..1]			
9		CopyIndicator	[0..1]			
9		DocumentPurpose	[0..1]			
9		IncludedNote	[0..*]			
10		InformationType	[1..1]			
10		InformationValue	[1..1]			
8		TradeAgreement	[1..1]			See MDR for sub elements
8		TradeDelivery	[1..1]			See MDR for sub elements
8		LineItem	[0..*]			See MDR for sub elements
7		ICCRelatedData	[0..1]	Appli		A sequence of one or more TLV data elements in accordance with ISO 7816-6
6		AuthorisationResult	[0..1]	Appli		
7		AuthorisationEntity	[0..1]			see AcceptorAuthorisationResponse
8		Identification	[0..1]			
8		Type	[1..1]			
8		Issuer	[0..1]			
8		Country	[0..1]			
8		ShortName	[0..1]			
7		ResponseToAuthorisation	[1..1]			see AcceptorAuthorisationResponse
8		Response	[1..1]			
8		ResponseReason	[0..1]	Appli		
8		AdditionalResponse-Information	[0..1]			
7		AuthorisationCode	[0..1]	Appli		see AcceptorAuthorisationResponse
7		CompletionRequired	[0..1]			Copy from AuthorisationResponse
7		TMSTrigger	[0..1]	Appli		
8		TMSContactLevel	[1..1]			
8		TMSIdentification	[0..1]	Appli		
8		TMSContactDateTime	[0..1]			
6		TransactionVerificationResult	[0..1]	Appli		
7		Method	[1..1]			
7		VerificationEntity	[0..1]			
7		Result	[0..1]			
7		AdditionalResult	[0..1]			
6		AdditionalTransactionData	[0..*]	Appli		
4	Or	Cancellation	[0..1]			Cancelled card payment transaction.
5		TransactionSequenceCounter	[1..1]			see Completion/TransactionSequenceCounter
5		Traceability	[0..*]			
6		RelayIdentification	[1..1]			
7		Identification	[1..1]			
7		Type	[1..1]			
7		Issuer	[0..1]			
7		Country	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
7		ShortName	[0..1]			
6		ProtocolName	[0..1]			
6		ProtocolVersion	[0..1]			
6		TraceDateTimeln	[1..1]			
6		TraceDateTimeOut	[1..1]			
5		Environment	[1..1]			<i>see AcceptorCancellationAdvice</i>
6		Acquirer	[0..1]			
7		Identification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		Country	[0..1]			
8		ShortName	[0..1]			
7		ParametersVersion	[1..1]			
6		Merchant	[0..1]			
7		Identification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		CommonName	[0..1]			
7		LocationCategory	[0..1]			
7		LocationAndContact	[0..1]	Config		
8		PostalAddress	[0..1]			
9		AddressLine	[0..2]			
9		StreetName	[0..1]			
9		BuildingNumber	[0..1]			
9		PostCode	[0..1]			
9		TownName	[1..1]			
9		CountrySubDivision	[0..2]			
9		Country	[1..1]			
8		Email	[0..1]			
8		URLAddress	[0..1]			
8		Phone	[0..1]			
8		CustomerService	[0..1]			
8		AdditionalContact- Information	[0..1]			
7		SchemeData	[0..1]			
6		POI	[0..1]			
7		Identification	[1..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		SystemName	[0..1]			
7		GroupIdentifier	[0..1]			
7		Capabilities	[0..1]			
8		CardReadingCapabilities	[0..*]			
8		CardholderVerification- Capabilities	[0..*]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		PINLengthCapabilities	[0..1]			
8		ApprovalCodeLength	[0..1]			
8		MaxScriptLength	[0..1]			
8		CardCaptureCapable	[0..1]			
8		OnLineCapabilities	[0..1]			
8		MessageCapabilities	[0..*]			
9		Destination	[1..*]			
9		AvailableFormat	[0..*]			
9		NumberOfLines	[0..1]			
9		LineWidth	[0..1]			
9		AvailableLanguage	[0..*]			
7		TimeZone	[0..1]			
7		TerminalIntegration	[0..1]			
7		Component	[0..*]			
8		Type	[1..1]			
8		Identification	[1..1]			
9		ItemNumber	[0..1]			
9		ProviderIdentification	[0..1]			
9		Identification	[0..1]			
9		SerialNumber	[0..1]			
8		Status	[0..1]			
9		VersionNumber	[0..1]			
9		Status	[0..1]			
9		ExpiryDate	[0..1]			
8		StandardCompliance	[0..*]			
9		Identification	[1..1]			
9		Version	[1..1]			
9		Issuer	[1..1]			
8		Characteristics	[0..1]			
9		Memory	[0..*]			
10		Identification	[1..1]			
10		TotalSize	[1..1]			
10		FreeSize	[1..1]			
10		Unit	[1..1]			
9		Communication	[0..*]			
10		CommunicationType	[1..1]			
10		RemoteParty	[1..*]			
10		Active	[1..1]			
9		SecurityAccessModules	[0..1]			
9		SubscriberIdentityModules	[0..1]			
9		KeyCheckValue	[0..1]			
9		KeyCharacteristic	[0..1]			
9		EncryptedKey	[0..1]			
8		Assessment	[0..*]			
9		Type	[1..1]			
9		Assigner	[1..*]			
9		DeliveryDate	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		ExpirationDate	[0..1]			
9		Number	[1..1]			
6		Card	[1..1]			
7		ProtectedCardData	[0..1]			
8		ContentType	[1..1]			
8		EnvelopedData	[1..1]			
7		PlainCardData	[0..1]			
8		PAN	[1..1]			
8		CardSequenceNumber	[0..1]			
8		EffectiveDate	[0..1]			
8		ExpiryDate	[1..1]			
8		ServiceCode	[0..1]			
8		Track1	[0..1]			
8		Track2	[0..1]			
8		Track3	[0..1]			
7		PaymentAccountReference	[0..1]			
7		MaskedPAN	[0..1]			
7		IssuerBIN				
7		CardCountryCode	[0..1]			
7		CardCurrencyCode				
7		CardProductProfile	[0..1]			
7		CardBrand	[0..1]			
7		CardProductType	[0..1]			
7		AdditionalCardData	[0..1]			
7		CardProductSubType	[0..1]			
6		CustomerDevice	[0..1]			
7		Identification	[0..1]			
7		Type	[0..1]			
7		Provider	[0..1]			
6		Wallet	[0..1]			
7		Identification	[0..1]			
7		Type	[0..1]			
7		Provider	[0..1]			
6		PaymentToken	[0..1]			
7		TokenCharacteristic	[0..*]			
7		TokenRequestor	[0..1]			
8		ProviderIdentification	[1..1]			
8		RequestorIdentification	[1..1]			
7		TokenAssuranceLevel	[0..1]			
6		Cardholder	[0..1]			
7		Identification	[0..*]			
8		DriverLicenseNumber	[0..1]			
8		DriverLicenseLocation	[0..1]			
8		DriverLicenseName	[0..1]			
8		DriverIdentification	[0..1]			
8		CustomerNumber	[0..1]			
8		SocialSecurityNumber	[0..1]			
8		AlienRegistrationNumber	[0..1]			
8		PassportNumber	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		TaxIdentificationNumber	[0..1]			
8		IdentityCardNumber	[0..1]			
8		EmployerIdentification- Number	[0..1]			
8		EmployeeIdentification- Number	[0..1]			
8		JobNumber	[0..1]			
8		Department	[0..1]			
8		EmailAddress	[0..1]			
8		DateAndPlaceOfBirth	[0..1]			
9		BirthDate	[1..1]			
9		ProvinceOfBirth	[0..1]			
9		CityOfBirth	[1..1]			
9		CountryOfBirth	[1..1]			
8		Other	[0..*]			
9		Identification	[1..1]			
9		IdentificationType	[1..1]			
7		Name	[0..1]			
7		Language	[0..1]			
7		BillingAddress	[0..1]			
8		AddressLine	[0..2]			
8		StreetName	[0..1]			
8		BuildingNumber	[0..1]			
8		PostCode	[0..1]			
8		TownName	[1..1]			
8		CountrySubDivision	[0..2]			
8		Country	[1..1]			
7		ShippingAddress	[0..1]			
8		AddressLine	[0..2]			
8		StreetName	[0..1]			
8		BuildingNumber	[0..1]			
8		PostCode	[0..1]			
8		TownName	[1..1]			
8		CountrySubDivision	[0..2]			
8		Country	[1..1]			
7		TripNumber	[0..1]			
7		Vehicle	[0..1]			
8		VehicleNumber	[0..1]			
8		TrailerNumber	[0..1]			
8		VehicleTag	[0..1]			
8		VehicleTagEntryMode	[0..1]			
8		UnitNumber	[0..1]			
8		ReplacementCar	[0..1]			
8		Odometer	[0..1]			
8		Hubometer	[0..1]			
8		TrailerHours	[0..1]			
8		ReferHours	[0..1]			
8		MaintenancIdentification	[0..1]			
8		DriverOrVehicleCard	[0..1]			
9		PAN	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		Track1	[0..1]			
9		Track2	[0..1]			
9		Track3	[0..1]			
9		AdditionalCardData	[0..1]			
9		EntryMode	[0..1]			
8		AdditionalVehicleData	[0..*]			
9		Type	[0..1]			
9		EntryMode	[0..1]			
9		Data	[1..1]			
7		PersonalData	[0..1]			
6		ProtectedCardholderData	[0..1]			
5		Context	[0..1]			see <i>AcceptorCancellationAdvice</i>
6		PaymentContext	[0..1]			
7		CardPresent	[0..1]			
7		CardholderPresent	[0..1]			
7		OnlineContext	[0..1]			
7		AttendanceContext	[0..1]			
7		TransactionEnvironment	[0..1]			
7		TransactionChannel	[0..1]			
7		AttendantMessageCapable	[0..1]			
7		AttendantLanguage	[0..1]			
7		CardDataEntryMode	[1..1]			
7		FallbackIndicator	[0..1]			
6		SaleContext	[0..1]			
7		SaleIdentification	[0..1]			
7		SaleReferenceNumber	[0..1]			
7		SaleReconciliation- Identification	[0..1]			
7		CashierIdentification	[0..1]			
7		ShiftNumber	[0..1]			
7		PurchaseOrderNumber	[0..1]			
7		InvoiceNumber	[0..1]			
7		DeliveryNoteNumber	[0..1]			
7		SponsoredMerchant	[0..*]			
8		CommonName	[1..1]			
8		Address	[0..1]			
8		CountryCode	[1..1]			
8		MerchantCategoryCode	[1..1]			
8		RegisteredIdentifier	[1..1]			
7		SplitPayment	[0..1]			
7		RemainingAmount	[0..1]			
7		AdditionalSaleData	[0..1]			
5		Transaction	[1..1]			see <i>AcceptorCancellationAdvice</i>
6		MerchantCategoryCode	[0..1]		C3	
6		CustomerConsent	[0..*]			
6		CardProgrammeProposed	[0..*]			
6		CardProgrammeApplied	[0..*]			
6		SaleReferenceIdentification	[0..1]			
6		TransactionIdentification	[1..1]			
7		TransactionDateTime	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
7		TransactionReference	[1..1]			
6		OriginalTransaction	[0..1]			
7		SaleReferenceIdentification	[0..1]			
7		TransactionIdentification	[1..1]			
8		TransactionDateTime	[1..1]			
8		TransactionReference	[1..1]			
7		POIIdentification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		InitiatorTransaction- Identification	[0..1]			
7		RecipientTransaction- Identification	[0..1]			
7		TransactionType	[1..1]			
7		AdditionalService	[0..*]			
7		ServiceAttribute	[0..1]			
7		CardDataEntryMode	[0..1]			
7		TransactionResult	[0..1]			
8		AuthorisationEntity	[0..1]			
9		Identification	[0..1]			
9		Type	[1..1]			
9		Issuer	[0..1]			
9		Country	[0..1]			
9		ShortName	[0..1]			
8		ResponseToAuthorisation	[1..1]			
9		Response	[1..1]			
9		ResponseReason	[0..1]			
9		AdditionalResponse- Information	[0..1]			
8		AuthorisationCode	[0..1]			
6		TransactionSuccess	[1..1]			
6		Reversal	[0..1]			
6		FailureReason	[0..*]			
6		InitiatorTransactionIdentification	[0..1]			
6		RecipientTransaction- Identification	[0..1]			
6		ReconciliationIdentification	[0..1]			
6		InterchangeData	[0..1]			
6		TransactionDetails	[1..1]			
7		Currency	[0..1]			
7		TotalAmount	[1..1]			
7		ValidityDate	[0..1]			
7		ICCRelatedData	[0..1]			
6		AuthorisationResult	[0..1]			
7		AuthorisationEntity	[0..1]			
8		Identification	[0..1]			
8		Type	[1..1]			
8		Issuer	[0..1]			
8		Country	[0..1]			

Lvl	Or	AcceptorBatchTransfer	Mult.	Rule	Cstr	Type / Definition / Code List
8		ShortName	[0..1]			
7		ResponseToAuthorisation	[1..1]			
8		Response	[1..1]			
8		ResponseReason	[0..1]			
8		AdditionalResponse- Information	[0..1]			
7		AuthorisationCode	[0..1]			
7		TMSTrigger	[0..1]			
8		TMSContactLevel	[1..1]			
8		TMSIdentification	[0..1]			
8		TMSContactDateTime	[0..1]			
6		AdditionalTransactionData	[0..*]			
4	Or	AuthorisationRequest	[0..1]			Authorisation request of a card payment transaction.
5		TransactionSequenceCounter	[1..1]			see Completion/TransactionSequenceCounter
5		Traceability	[0..*]			
6		RelayIdentification	[1..1]			
7		Identification	[1..1]			
7		Type	[1..1]			
7		Issuer	[0..1]			
7		Country	[0..1]			
7		ShortName	[0..1]			
6		ProtocolName	[0..1]			
6		ProtocolVersion	[0..1]			
6		TraceDateTimeln	[1..1]			
6		TraceDateTimeOut	[1..1]			
5		Environment	[1..1]			see AcceptorAuthorisationRequest
6		Acquirer	[0..1]			
7		Identification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		Country	[0..1]			
8		ShortName	[0..1]			
7		ParametersVersion	[1..1]			
6		Merchant	[0..1]			
7		Identification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		CommonName	[0..1]			
7		LocationCategory	[0..1]			
7		LocationAndContact	[0..1]	Config		
8		PostalAddress	[0..1]			
9		AddressLine	[0..2]			
9		StreetName	[0..1]			
9		BuildingNumber	[0..1]			
9		PostCode	[0..1]			
9		TownName	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		CountrySubDivision	[0..2]			
9		Country	[1..1]			
8		Email	[0..1]			
8		URLAddress	[0..1]			
8		Phone	[0..1]			
8		CustomerService	[0..1]			
8		AdditionalContact- Information	[0..1]			
7		SchemeData	[0..1]			
6		POI	[0..1]			
7		Identification	[1..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		SystemName	[0..1]			
7		GroupIdentifier	[0..1]			
7		Capabilities	[0..1]			
8		CardReadingCapabilities	[0..*]			
8		CardholderVerification- Capabilities	[0..*]			
8		PINLengthCapabilities	[0..1]			
8		ApprovalCodeLength	[0..1]			
8		MaxScriptLength	[0..1]			
8		CardCaptureCapable	[0..1]			
8		OnLineCapabilities	[0..1]			
8		MessageCapabilities	[0..*]			
9		Destination	[1..*]			
9		AvailableFormat	[0..*]			
9		NumberOfLines	[0..1]			
9		LineWidth	[0..1]			
9		AvailableLanguage	[0..*]			
7		TimeZone	[0..1]			
7		TerminalIntegration	[0..1]			
7		Component	[0..*]			
8		Type	[1..1]			
8		Identification	[1..1]			
9		ItemNumber	[0..1]			
9		ProviderIdentification	[0..1]			
9		Identification	[0..1]			
9		SerialNumber	[0..1]			
8		Status	[0..1]			
9		VersionNumber	[0..1]			
9		Status	[0..1]			
9		ExpiryDate	[0..1]			
8		StandardCompliance	[0..*]			
9		Identification	[1..1]			
9		Version	[1..1]			
9		Issuer	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		Characteristics	[0..1]			
9		Memory	[0..*]			
10		Identification	[1..1]			
10		TotalSize	[1..1]			
10		FreeSize	[1..1]			
10		Unit	[1..1]			
9		Communication	[0..*]			
10		CommunicationType	[1..1]			
10		RemoteParty	[1..*]			
10		Active	[1..1]			
9		SecurityAccessModules	[0..1]			
9		SubscriberIdentity- Modules	[0..1]			
9		KeyCheckValue	[0..1]			
9		KeyCharacteristic	[0..1]			
9		EncryptedKey	[0..1]			
8		Assessment	[0..*]			
9		Type	[1..1]			
9		Assigner	[1..*]			
9		DeliveryDate	[0..1]			
9		ExpirationDate	[0..1]			
9		Number	[1..1]			
6		Card	[1..1]			
7		ProtectedCardData	[0..1]			
8		ContentType	[1..1]			
8		EnvelopedData	[1..1]			
7		PlainCardData	[0..1]			
8		PAN	[1..1]			
8		CardSequenceNumber	[0..1]			
8		EffectiveDate	[0..1]			
8		ExpiryDate	[1..1]			
8		ServiceCode	[0..1]			
8		Track1	[0..1]			
8		Track2	[0..1]			
8		Track3	[0..1]			
8		CardholderName	[0..1]			
7		PaymentAccountReference	[0..1]			
7		IssuerBIN				
7		CardCountryCode	[0..1]			
7		CardCurrencyCode				
7		CardProductProfile	[0..1]			
7		CardBrand	[0..1]			
7		InternationalCard	[0..1]			
7		AllowedProduct	[0..*]			
7		ServiceOption	[0..1]			
7		AdditionalCardData	[0..1]			
6		CustomerDevice	[0..1]			
7		Identification	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
7		Type	[0..1]			
7		Provider	[0..1]			
6		Wallet	[0..1]			
7		Identification	[0..1]			
7		Type	[0..1]			
7		Provider	[0..1]			
6		PaymentToken	[0..1]			
7		TokenCharacteristic	[0..*]			
7		TokenRequestor	[0..1]			
8		ProviderIdentification	[1..1]			
8		RequestorIdentification	[1..1]			
6		Cardholder	[0..1]			
7		Identification	[0..*]			
8		DriverLicenseNumber	[0..1]			
8		DriverLicenseLocation	[0..1]			
8		DriverLicenseName	[0..1]			
8		DriverIdentification	[0..1]			
8		CustomerNumber	[0..1]			
8		SocialSecurityNumber	[0..1]			
8		AlienRegistrationNumber	[0..1]			
8		PassportNumber	[0..1]			
8		TaxIdentificationNumber	[0..1]			
8		IdentityCardNumber	[0..1]			
8		EmployerIdentification- Number	[0..1]			
8		EmployeeIdentification- Number	[0..1]			
8		JobNumber	[0..1]			
8		Department	[0..1]			
8		EmailAddress	[0..1]			
8		DateAndPlaceOfBirth	[0..1]			
9		BirthDate	[1..1]			
9		ProvinceOfBirth	[0..1]			
9		CityOfBirth	[1..1]			
9		CountryOfBirth	[1..1]			
8		Other	[0..*]			
9		Identification	[1..1]			
9		IdentificationType	[1..1]			
7		Name	[0..1]			
7		Language	[0..1]			
7		BillingAddress	[0..1]			
8		AddressLine	[0..2]			
8		StreetName	[0..1]			
8		BuildingNumber	[0..1]			
8		PostCode	[0..1]			
8		TownName	[1..1]			
8		CountrySubDivision	[0..2]			
8		Country	[1..1]			
7		ShippingAddress	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		AddressLine	[0..2]			
8		StreetName	[0..1]			
8		BuildingNumber	[0..1]			
8		PostCode	[0..1]			
8		TownName	[1..1]			
8		CountrySubDivision	[0..2]			
8		Country	[1..1]			
7		TripNumber	[0..1]			
7		Vehicle	[0..1]			
8		VehicleNumber	[0..1]			
8		TrailerNumber	[0..1]			
8		VehicleTag	[0..1]			
8		VehicleTagEntryMode	[0..1]			
8		UnitNumber	[0..1]			
8		ReplacementCar	[0..1]			
8		Odometer	[0..1]			
8		Hubometer	[0..1]			
8		TrailerHours	[0..1]			
8		ReferHours	[0..1]			
8		MaintenancelIdentification	[0..1]			
8		DriverOrVehicleCard	[0..1]			
9		PAN	[0..1]			
9		Track1	[0..1]			
9		Track2	[0..1]			
9		Track3	[0..1]			
9		AdditionalCardData	[0..1]			
9		EntryMode	[0..1]			
8		AdditionalVehicleData	[0..*]			
9		Type	[0..1]			
9		EntryMode	[0..1]			
9		Data	[1..1]			
7		Authentication	[0..*]			see AcceptorAuthorisationRequest
8		AuthenticationMethod	[1..1]			
8		AuthenticationValue	[0..1]			
8		ProtectedAuthentication-Value	[0..1]			
8		CardholderOnlinePIN	[0..1]			
9		EncryptedPINBlock	[1..1]			
9		PINFormat	[1..1]			
9		AdditionalInput	[0..1]			
8		CardholderIdentification	[0..1]			See MDR for sub elements
8		AddressVerification	[0..1]			
9		AddressDigits	[0..1]			
9		PostalCodeDigits	[0..1]			
8		AuthenticationType	[0..1]			
8		AuthenticationLevel	[0..1]			
8		AuthenticationResult	[0..1]			
8		AuthenticationAdditional-Information	[0..1]			
7		TransactionVerificationResult	[0..*]			see AcceptorAuthorisationRequest

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		Method	[1..1]			
8		VerificationEntity	[0..1]			
8		Result	[0..1]			
8		AdditionalResult	[0..1]			
7		PersonalData	[0..1]			
6		ProtectedCardholderData	[0..1]			
5		Context	[0..1]			see AcceptorAuthorisationRequest
6		PaymentContext	[0..1]			
7		CardPresent	[0..1]			
7		CardholderPresent	[0..1]			
7		OnlineContext	[0..1]			
7		AttendanceContext	[0..1]			
7		TransactionEnvironment	[0..1]			
7		TransactionChannel	[0..1]			
7		AttendantMessageCapable	[0..1]			
7		AttendantLanguage	[0..1]			
7		CardDataEntryMode	[0..1]			
7		FallbackIndicator	[0..1]			
7		SupportedOption	[0..*]			
6		SaleContext	[0..1]			
7		SaleIdentification	[0..1]			
7		SaleReferenceNumber	[0..1]			
7		SaleReconciliation- Identification	[0..1]			
7		CashierIdentification	[0..1]			
7		ShiftNumber	[0..1]			
7		PurchaseOrderNumber	[0..1]			
7		InvoiceNumber	[0..1]			
7		DeliveryNoteNumber	[0..1]			
7		SponsoredMerchant	[0..*]			
8		CommonName	[1..1]			
8		Address	[0..1]			
8		CountryCode	[1..1]			
8		MerchantCategoryCode	[1..1]			
8		RegisteredIdentifier	[1..1]			
7		SplitPayment	[0..1]			
7		RemainingAmount	[0..1]			
7		AdditionalSaleData	[0..1]			
5		Transaction	[1..1]			see AcceptorAuthorisationRequest
6		TransactionCapture	[1..1]			
6		TransactionType	[0..1]			
6		AdditionalService	[0..*]			
6		ServiceAttribute	[0..1]			
6		MerchantCategoryCode	[0..1]		C3	
6		CustomerConsent	[0..*]			
6		CardProgrammeProposed	[0..*]			
6		CardProgrammeApplied	[0..*]			
6		SaleReferenceIdentification	[0..1]			
6		TransactionIdentification	[1..1]			
7		TransactionDateTime	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
7		TransactionReference	[1..1]			
6		OriginalTransaction	[0..1]			
7		SaleReferenceIdentification	[0..1]			
7		TransactionIdentification	[1..1]			
8		TransactionDateTime	[1..1]			
8		TransactionReference	[1..1]			
7		POIIdentification	[0..1]			
8		Identification	[1..1]			
8		Type	[0..1]			
8		Issuer	[0..1]			
8		ShortName	[0..1]			
7		InitiatorTransaction- Identification	[0..1]			
7		RecipientTransaction- Identification	[0..1]			
7		TransactionType	[1..1]			
7		AdditionalService	[0..*]			
7		ServiceAttribute	[0..1]			
7		CardDataEntryMode	[0..1]			
7		TransactionResult	[0..1]			
8		AuthorisationEntity	[0..1]			
9		Identification	[0..1]			
9		Type	[1..1]			
9		Issuer	[0..1]			
9		Country	[0..1]			
9		ShortName	[0..1]			
8		ResponseToAuthorisation	[1..1]			
9		Response	[1..1]			
9		ResponseReason	[0..1]			
9		AdditionalResponse- Information	[0..1]			
8		AuthorisationCode	[0..1]			
6		InitiatorTransactionIdentification	[0..1]			
6		ReconciliationIdentification	[0..1]			
6		TransactionDetails	[1..1]			
7		Currency	[0..1]			
7		TotalAmount	[1..1]			
7		AmountQualifier	[0..1]			
7		DetailedAmount	[0..1]			
8		AmountOfGood- AndServices	[0..1]			
8		CashBack	[0..1]			
8		Gratuity	[0..1]			
8		Fees	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		Rebate	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		ValueAddedTax	[0..*]			
9		Amount	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
9		Label	[0..1]			
8		Surcharge	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
7		ValidityDate	[0..1]			
7		OnlineReason	[0..1]			
7		UnattendedLevelCategory	[0..1]			
7		AccountType	[0..1]			
7		CurrencyConversionResult	[0..1]			
8		AcceptedByCardholder	[0..1]			
8		Conversion	[0..1]			
9		CurrencyConversion- Identification	[0..1]			
9		TargetCurrency	[1..1]			
10		AlphaCode	[1..1]			
10		NumericCode	[1..1]			
10		Decimal	[1..1]			
10		Name	[0..1]			
9		ResultingAmount	[1..1]			
9		ExchangeRate	[1..1]			
9		InvertedExchangeRate	[0..1]			
9		QuotationDate	[0..1]			
9		ValidUntil	[0..1]			
9		SourceCurrency	[1..1]			
10		AlphaCode	[1..1]			
10		NumericCode	[1..1]			
10		Decimal	[1..1]			
10		Name	[0..1]			
9		OriginalAmount	[1..1]			
10		ActualAmount	[0..1]			
10		MinimumAmount	[0..1]			
10		MaximumAmount	[0..1]			
9		CommissionDetails	[0..*]			
10		Amount	[1..1]			
10		AdditionalInformation	[0..1]			
9		MarkUpDetails	[0..*]			
10		Rate	[1..1]			
10		AdditionalInformation	[0..1]			
9		DeclarationDetails	[0..1]			
10		Format	[0..1]			
10		MessageContent	[1..1]			
7		Instalment	[0..1]			
8		InstalmentPlan	[0..*]			
8		PlanIdentification	[0..1]			
8		SequenceNumber	[0..1]			
8		PeriodUnit	[0..1]			
8		InstalmentPeriod	[0..1]			
8		TotalNumberOfPayments	[0..1]			
8		FirstPaymentDate	[0..1]			
8		TotalAmount	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		FirstAmount	[0..1]			
8		Charges	[0..1]			
7		AggregationTransaction	[0..1]			
8		FirstPaymentDateTime	[0..1]			
8		LastPaymentDateTime	[0..1]			
8		NumberOfPayments	[0..1]			
8		IndividualPayment	[0..*]			
9		Amount	[1..1]			
9		DateTime	[1..1]			
9		CardDataEntryMode	[0..1]			
9		ICCRelatedData	[0..1]			
9		Label	[0..1]			
7		ProductCodeSetIdentification	[0..1]			
7		SaleItem	[0..*]			
8		ItemIdentification	[0..1]			
8		ProductCode	[1..1]			
8		AdditionalProductCode	[0..1]			
8		UnitOfMeasure	[0..1]			
8		ProductQuantity	[0..1]			
8		UnitPrice	[0..1]			
8		UnitPriceSign	[0..1]			
8		ProductAmount	[1..1]			
8		ProductAmountSign	[0..1]			
8		ValueAddedTax	[0..1]			
8		TaxType	[0..1]			
8		ProductDescription	[0..1]			
8		DeliveryLocation	[0..1]			
8		DeliveryService	[0..1]			
7		DeliveryLocation	[0..1]			
7		CardPaymentInvoice	[0..1]			
8		InvoiceHeader	[1..1]			
9		Identification	[1..1]			
9		TypeCode	[1..1]			
9		Name	[0..*]			
9		IssueDateTime	[1..1]			
9		Issuer	[0..1]			
10		PartyIdentification	[1..1]			See MDR for sub elements
10		LegalOrganisation	[0..1]			
11		Identification	[0..1]			
11		Name	[0..1]			
10		TaxParty	[0..*]			See MDR for sub elements
9		LanguageCode	[0..1]			
9		CopyIndicator	[0..1]			
9		DocumentPurpose	[0..1]			
9		IncludedNote	[0..*]			
10		InformationType	[1..1]			
10		InformationValue	[1..1]			
8		TradeAgreement	[1..1]			See MDR for sub elements
8		TradeDelivery	[1..1]			See MDR for sub elements
8		LineItem	[0..*]			See MDR for sub elements

<i>Lvl</i>	<i>Or</i>	<i>AcceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
7		ICCRelatedData	[0..n]			
6		AdditionalTransactionData	[0..n]			
4	Or}	AuthorisationResponse	[0..1]			Authorisation response of a card payment transaction.
5		TransactionSequenceCounter	[1..1]			see Completion/TransactionSequenceCounter
5		Traceability	[0..*]			
6		RelayIdentification	[1..1]			
7		Identification	[1..1]			
7		Type	[1..1]			
7		Issuer	[0..1]			
7		Country	[0..1]			
7		ShortName	[0..1]			
6		ProtocolName	[0..1]			
6		ProtocolVersion	[0..1]			
6		TraceDateTimeln	[1..1]			
6		TraceDateTimeOut	[1..1]			
5		Environment	[1..1]			see AcceptorAuthorisationResponse
6		AcquirerIdentification	[0..1]			
7		Identification	[1..1]			
7		Type	[0..1]			
7		Issuer	[0..1]			
7		Country	[0..1]			
7		ShortName	[0..1]			
6		MerchantIdentification	[0..1]			
7		Identification	[1..1]			
7		Type	[0..1]			
7		Issuer	[0..1]			
7		ShortName	[0..1]			
6		POIIdentification	[0..1]			
7		Identification	[1..1]			
7		Type	[0..1]			
7		Issuer	[0..1]			
7		ShortName	[0..1]			
6		Card	[0..1]			
7		ProtectedCardData	[0..1]			
8		ContentType	[1..1]			
8		EnvelopedData	[1..1]			
7		PlainCardData	[0..1]			
8		PAN	[1..1]			
8		CardSequenceNumber	[0..1]			
8		EffectiveDate	[0..1]			
8		ExpiryDate	[1..1]			
7		PaymentAccountReference	[0..1]			
7		MaskedPAN	[0..1]			
7		CardBrand	[0..1]	Appli		see AcceptorAuthorisationRequest
7		CardProductType	[0..1]	Config		see AcceptorAuthorisationRequest
7		CardProductSubType	[0..1]			
6		PaymentToken	[0..1]			
7		TokenCharacteristic	[0..*]			
7		TokenAssuranceLevel	[0..1]			

Lvl	Or	AcceptorBatchTransfer	Mult.	Rule	Cstr	Type / Definition / Code List
5		Transaction	[1..1]			see AcceptorAuthorisationResponse
6		SaleReferenceIdentification	[0..1]			
6		TransactionIdentification	[1..1]			
7		TransactionDateTime	[1..1]			
7		TransactionReference	[1..1]			
6		InitiatorTransactionIdentification	[0..1]			
6		RecipientTransaction- Identification	[0..1]			
6		ReconciliationIdentification	[0..1]			
6		InterchangeData	[0..1]			
6		TransactionDetails	[1..1]			
7		Currency	[0..1]			
7		TotalAmount	[1..1]			
7		DetailedAmount	[0..1]			
8		AmountOfGood- AndServices	[0..1]			
8		CashBack	[0..1]			
8		Gratuity	[0..1]			
8		Fees	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		Rebate	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		ValueAddedTax	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
8		Surcharge	[0..*]			
9		Amount	[1..1]			
9		Label	[0..1]			
7		InvoiceAmount	[0..1]			
7		ValidityDate	[0..1]			
7		AccountType	[0..1]			
7		ICCRelatedData	[0..1]			
5		TransactionResponse	[1..1]			
6		AuthorisationResult	[1..1]			
7		AuthorisationEntity	[1..1]			
8		Identification	[0..1]			
8		Type	[1..1]			
8		Issuer	[0..1]			
8		Country	[0..1]			
8		ShortName	[0..1]			
7		ResponseToAuthorisation	[1..1]			
8		Response	[1..1]			
8		ResponseReason	[0..1]			
8		AdditionalResponse- Information	[0..1]			
7		AuthorisationCode	[0..1]			
7		CompletionRequired	[0..1]			
7		TMSTrigger	[0..1]			
8		TMSContactLevel	[1..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		TMSIdentification	[0..1]			
8		TMSContactDateTime	[0..1]			
6		TransactionVerificationResult	[0..1]			
7		Method	[1..1]			
7		VerificationEntity	[0..1]			
7		Result	[0..1]			
7		AdditionalResult	[0..1]			
6		AllowedProductCode	[0..*]			
7		ProductCode	[1..1]			
7		AdditionalProductCode	[0..1]			
6		NotAllowedProductCode	[0..*]			
7		ProductCode	[1..1]			
7		AdditionalProductCode	[0..1]			
6		AdditionalAvailableProduct	[0..*]			
7		ProductCode	[1..1]			
7		AdditionalProductCode	[0..1]			
7		AmountLimit	[0..1]			
7		QuantityLimit	[0..1]			
7		UnitOfMeasure	[0..1]			
6		Balance	[0..1]			
7		Amount	[1..1]			Currency can be provided as an XML Attribute.
7		Sign	[0..1]			
6		ProtectedBalance				
6		Action	[0..*]			
7		ActionType	[1..1]			
7		MessageToPresent	[0..1]			
8		MessageDestination	[1..1]			
8		Format				
8		MessageContent	[1..1]			
8		MessageContentSignature	[0..1]			
6		CurrencyConversionEligibility	[0..1]			
7		CurrencyConversion- Identification	[0..1]			
7		TargetCurrency	[1..1]			
8		AlphaCode	[1..1]			
8		NumericCode	[1..1]			
8		Decimal	[1..1]			
8		Name	[0..1]			
7		ResultingAmount	[1..1]			
7		ExchangeRate	[1..1]			
7		InvertedExchangeRate	[0..1]			
7		QuotationDate	[0..1]			
7		ValidUntil	[0..1]			
7		SourceCurrency	[1..1]			
8		AlphaCode	[1..1]			
8		NumericCode	[1..1]			
8		Decimal	[1..1]			
8		Name	[0..1]			
7		OriginalAmount	[1..1*]			
8		ActualAmount	[0..1]			

<i>Lvl</i>	<i>Or</i>	<i>AceptorBatchTransfer</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Type / Definition / Code List</i>
8		MinimumAmount	[0..1]			
8		MaximumAmount	[0..1]			
7		CommissionDetails	[0..*]			
8		Amount	[1..1]			
8		AdditionalInformation	[0..1]			
7		MarkUpDetails	[0..*]			
8		Rate	[1..1]			
8		AdditionalInformation	[0..1]			
7		DeclarationDetails	[0..1]			
8		Format	[1..1]			
8		MessageContent	[1..1]			
1		SecurityTrailer	[0..1]			
2		ContentType	[1..1]			
2		EnvelopedData	[0..1]			
2		AuthenticatedData	[0..1]			
2		SignedData	[0..1]			

4.6.1.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	POI must either be present in <i>CommonData</i> or in <i>Transaction</i>	<ul style="list-style-type: none"> BatchTransfer.DataSet.CommonData.Environment.POI BatchTransfer.DataSet.Transaction.Completion.Environment.POI BatchTransfer.DataSet.Transaction.Cancellation.Environment.POI BatchTransfer.DataSet.Transaction.AuthorisationRequest.Environment.POI BatchTransfer.DataSet.Transaction.AuthorisationResponse.Environment.POI
C2	If <i>Context</i> is present it must have at least one child	<ul style="list-style-type: none"> BatchTransfer.DataSet.Transaction.Completion.Context BatchTransfer.DataSet.Transaction.Cancellation.Context BatchTransfer.DataSet.Transaction.AuthorisationRequest.Context BatchTransfer.DataSet.Transaction.AuthorisationResponse.Context
C3	<i>MerchantCategoryCode</i> must be present either in <i>CommonData</i> or in <i>Transaction</i>	<ul style="list-style-type: none"> BatchTransfer.DataSet.CommonData.MerchantCategoryCode BatchTransfer.DataSet.Transaction.Completion.Transaction.MerchantCategoryCode BatchTransfer.DataSet.Transaction.Cancellation.Transaction.MerchantCategoryCode BatchTransfer.DataSet.Transaction.AuthorisationRequest.Transaction.MerchantCategoryCode BatchTransfer.DataSet.Transaction.AuthorisationResponse.Transaction.MerchantCategoryCode

4.6.2 AcceptorBatchTransferResponse (caaa.012.001.06)

Lvl	AcceptorBatchTransferResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	DownloadTransfer	[1..1]			<i>True</i>
2	FormatVersion	[1..1]			The Recipient Party has to adapt the batch transfer response format to the version of the Initiator sent in the batch transfer.
2	ExchangeIdentification	[1..1]	Copy		Several AcceptorBatchTransferResponse related to the same AcceptorBatchTransfer share the same ExchangeIdentification value. Duplication of a AcceptorBatchTransferResponse is detected with a common value of CreationDateTime. Mixing DataSet from different AcceptorBatchTransfer in the same AcceptorBatchTransferResponse is forbidden.
2	CreationDateTime	[1..1]			<i>see AcceptorBatchTransfer</i>
2	InitiatingParty	[1..1]	Copy		Initiator of the AcceptorBatchTransfer <i>see AcceptorBatchTransfer</i>
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[1..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
1	BatchTransferResponse	[1..1]			
2	TransactionTotals	[0..*]	Appli		
3	POIGroupIdentification	[0..1]	Appli		
3	CardProductProfile	[0..1]	Appli		
3	Currency	[0..1]			
3	Type	[1..1]			
3	TotalNumber	[1..1]			
3	CumulativeAmount	[1..1]			
2	DataSet	[0..*]			DataSet result concerns all the transactions present in the original DataSet. Each occurrence of DataSet present in the AcceptorBatchTransferResponse must be present in the original AcceptorBatchTransfer.

<i>Lvl</i>	<i>AcceptorBatchTransferResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
					Part of the DataSet could be missing and transferred in another AcceptorBatchTransferResponse.
3	DataSetIdentification	[1..1]	Copy		
4	Name	[1..1]			
4	Type	[0..1]			<i>CaptureResponse</i>
4	Version	[0..1]			Not used in Batch Capture context
4	CreationDateTime	[1..1]			
3	DataSetResult	[1..1]			Global result of the DataSet.
4	Response	[1..1]		C1	<i>Approved</i> : all transactions in DataSet are accepted. <i>Declined</i> : all transactions in DataSet are rejected. <i>PartialApproved</i> : only a subset of transactions is accepted and the others are rejected.
4	ResponseReason	[0..1]			
4	AdditionalResponseInformation	[0..1]			
3	RemoveDataSet	[1..1]			May be used to inform the Initiator to remove the transactions from the memory.
3	DataSetInitiator	[0..1]	Copy		
4	Identification	[1..1]	Copy		
4	Type	[0..1]	Copy		
4	Issuer	[0..1]	Copy		
4	Country	[0..1]	Config		
4	ShortName	[0..1]	Copy		
3	TransactionTotals	[1..*]			
4	POIGroupIdentification	[0..1]	Copy		
4	CardProductProfile	[0..1]	Copy		
4	Currency	[0..1]	Copy		
4	Type	[1..1]	Copy		See AcceptorReconciliation Request
4	TotalNumber	[1..1]			This data element contains the total number of transactions calculated by the recipient. In case of a difference with the total calculated by the InitiatingParty, the InitiatingParty has to take care of the error which has to be resolved by other mean.
4	CumulativeAmount	[1..1]			This data element contains the cumulative amount of transactions calculated by the recipient. In case of a difference with the CumulativeAmount calculated by the InitiatingParty, the InitiatingParty has to take care of the error which has to be resolved by other mean.
3	RejectedTransaction	[0..*]		C1	Present if DataSetResult.Response is <i>PartialApproved</i> . All transactions with TransactionResponse equal to <i>Declined</i> or <i>TechnicalError</i> must be present.
4	TransactionSequenceCounter	[1..1]	Copy		
4	TransactionResponse	[1..1]			
5	Response	[1..1]		*	Allowed values : <i>Declined</i> or <i>TechnicalError</i> .
5	ResponseReason	[0..1]			
4	Environment	[1..1]			
5	AcquirerIdentification	[0..1]	Copy		
6	Identification	[1..1]			
6	Type	[0..1]	Copy		
6	Issuer	[0..1]			
6	Country	[0..1]	Config		
6	ShortName	[0..1]			
5	MerchantIdentification	[0..1]	Copy		
6	Identification	[1..1]			

<i>Lvl</i>	<i>AcceptorBatchTransferResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
6	Type	[0..1]			
6	Issuer	[0..1]			
6	ShortName	[0..1]			
5	POIdentification	[1..1]	Copy		
6	Identification	[1..1]			
6	Type	[0..1]			
6	Issuer	[0..1]			
6	ShortName	[0..1]			
5	Card				
6	ProtectedCardData	[0..1]	Copy		see <i>AcceptorBatchTransfer</i>
6	PlainCardData	[0..1]	Copy		see <i>AcceptorBatchTransfer</i>
7	PAN	[1..1]			
7	CardSequenceNumber	[0..1]			
7	EffectiveDate	[0..1]			
7	ExpiryDate	[1..1]			
6	PaymentAccountReference	[0..1]			
6	MaskedPAN	[0..1]			
6	CardBrand	[0..1]			
6	CardProductType	[0..1]			
6	CardProductSubType	[0..1]			
5	PaymentToken	[0..1]			
6	TokenCharacteristic	[0..*]			
6	TokenAssuranceLevel	[0..1]			
4	Transaction	[1..1]			
5	SaleReferenceIdentification	[0..1]			
5	TransactionIdentification	[1..1]	Copy		
6	TransactionDateTime	[1..1]			
6	TransactionReference	[1..1]			
5	Response	[1..1]			Same value as RejectedTransaction.TransactionResponse.Response (not to be verified by the RecipientParty)
1	SecurityTrailer	[0..1]			

4.6.2.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	If <i>DataSetResult.Response</i> is "PartialApproved" then <i>RejectedTransaction</i> must be present.	<ul style="list-style-type: none"> BatchTransferResponse.DataSet.DataSetResult.Response BatchTransferResponse.DataSet.RejectedTransaction

4.7 Diagnostic Messages

4.7.1 AcceptorDiagnosticRequest (caaa.013.001.06)

Lvl	AcceptorDiagnosticRequest	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]		C1	The only valid code is <i>DiagnosticRequest</i> to request a diagnostic. (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
2	ProtocolVersion	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	ExchangeIdentification	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	CreationDateTime	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	InitiatingParty	[1..1]			see <i>AcceptorAuthorisationRequest</i>
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
2	RecipientParty	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
3	Identification	[1..1]	Config		
3	Type	[0..1]			
3	Issuer	[0..1]	Config		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Config		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	DiagnosticRequest	[1..1]		C1	The Header.MessageFunction must be <i>DiagnosticRequest</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to "ParsingError")

<i>Lvl</i>	<i>AcceptorDiagnosticRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
2	Environment	[1..1]			see <i>AcceptorAuthorisationRequest</i>
3	Acquirer	[1..1]			
4	Identification	[1..1]	Appli		
5	Identification	[1..1]	Appli		
5	Type	[0..1]			default <i>OriginatingPOI</i>
5	Issuer	[0..1]	Config		
5	Country				
5	ShortName	[0..1]	Config		
4	ParametersVersion	[1..1]			
3	AcquirerAvailabilityRequested	[0..1]			
3	MerchantIdentification	[0..1]	Config		see <i>AcceptorAuthorisationRequest</i>
4	Identification	[1..1]	Appli		
4	Type	[0..1]			default <i>Merchant</i>
4	Issuer	[0..1]	Config		
4	Country				
4	ShortName	[0..1]	Config		
3	POIIdentification	[1..1]			
4	Identification	[1..1]	Config		
4	Type	[0..1]			default <i>OriginatingPOI</i>
4	Issuer	[0..1]	Config		
4	ShortName	[0..1]	Config		
3	POIComponent	[0..*]			
4	Type	[1..1]			Components to be sent in the online authorisation are configured locally or by TMS configuration.:
4	Identification	[1..1]			Identification of the component.
5	ItemNumber	[0..1]			
5	ProviderIdentification	[0..1]			Identifies the provider of the component class (it replaces the data element <i>ManufacturerIdentification</i> of version 1).
5	Identification	[0..1]			Identification of the component assigned by the provider (it replaces the data element <i>Model</i> of version 1).
5	SerialNumber	[0..1]			Serial number of the component if available.
4	Status	[0..1]			Actual status of the component.
5	VersionNumber	[0..1]			Current version of component that may include the release number.
5	Status	[0..1]			
5	ExpiryDate	[0..1]			
4	StandardCompliance	[0..*]			Identification of the standard for which the component complies with.
5	Identification	[1..1]			Identification of the standard.
5	Version	[1..1]			Version of the standard.
5	Issuer	[1..1]			Entity assigning the identification
4	Characteristics	[0..1]			Only used in TMS protocol.
5	Memory	[0..*]			
6	Identification	[1..1]			
6	TotalSize	[1..1]			
6	FreeSize	[1..1]			
6	Unit	[1..1]			
5	Communication	[0..*]			
6	CommunicationType	[1..1]			
6	RemoteParty	[1..*]			

<i>Lvl</i>	<i>AcceptorDiagnosticRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
6	Active	[1..1]			
5	SecurityAccessModules	[0..1]			
5	SubscriberIdentityModules	[0..1]			
5	KeyCheckValue	[0..1]			
5	KeyCharacteristic	[0..1]			
6	KeyIdentification	[1..1]			
6	KeyVersion	[1..1]			
6	SequenceNumber	[0..1]			
6	DerivationIdentification	[0..1]			
6	Type	[0..1]			
6	Function	[0..1]			
5	EncryptedKey	[0..1]			
4	Assessment	[0..*]			Only used in TMS protocol.
5	Type	[1..1]			
5	Assigner	[1..*]			
5	DeliveryDate	[0..1]			
5	ExpirationDate	[0..1]			
5	Number	[1..1]			
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body DiagnosticRequest including the body envelope.

4.7.1.1 Constraints

This section lists all business rules implying at least 2 different elements inside the message.

<i>Constraint Number</i>	<i>Definition</i>	<i>Involved elements</i>
C1	When <i>DiagnosticRequest</i> is present the <i>Header.MessageFunction</i> must be <i>DiagnosticRequest</i>	<ul style="list-style-type: none"> Header.MessageFunction DiagnosticRequest

4.7.2 AcceptorDiagnosticResponse (caaa.014.001.05)

Lvl	AcceptorDiagnosticResponse	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]			The only valid code in a response to a diagnostic is : <i>DiagnosticResponse</i>
2	ProtocolVersion	[1..1]	Copy		see <i>AcceptorAuthorisationResponse</i>
2	ExchangeIdentification	[1..1]	Copy		
2	CreationDateTime	[1..1]			see <i>AcceptorAuthorisationResponse</i>
2	InitiatingParty	[1..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
2	RecipientParty	[0..1]	Copy		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]	Copy		
3	Country	[0..1]	Config		
3	ShortName	[0..1]	Copy		
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	Country	[0..1]	Config		
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeln	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	DiagnosticResponse	[1..1]			
2	Environment	[1..1]			
3	Acquirer				
4	Identification	[1..1]			
5	Identification	[1..1]			
5	Type	[0..1]			default <i>Merchant</i>
5	Issuer	[0..1]			
5	Country	[0..1]	Config		

<i>Lvl</i>	<i>AcceptorDiagnosticResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	ShortName	[0..1]			
4	ParametersVersion	[1..1]	Copy		
3	AcquirerAvailable	[0..1]			
3	MerchantIdentification	[0..1]			<i>see AcceptorAuthorisationResponse</i>
4	Identification	[1..1]			
4	Type	[0..1]			default <i>Merchant</i>
4	Issuer	[0..1]			
4	Country	[0..1]	Config		
4	ShortName	[0..1]			
3	POIdentification	[1..1]	Copy		
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
2	TMSTrigger	[0..1]	Appli		<i>see AcceptorAuthorisationResponse</i>
3	TMSContactLevel	[1..1]			
3	TMSIdentification	[0..1]	Appli		
3	TMSContactDateTime	[0..1]			Present if TMSContactLevel = <i>DateTime</i>
1	SecurityTrailer	[0..1]			CMS data structure ContentInfoType with the AuthenticatedData alternative, containing the MAC of the message body DiagnosticResponse including the body envelope, using the MAC key of the related AcceptorDiagnosticRequest message.

4.8 Reject Message

4.8.1 AcceptorRejection (caaa.015.001.05)

Lvl	AcceptorRejection	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			see <i>AcceptorAuthorisationRequest</i>
2	MessageFunction	[1..1]		*	The only valid codes to be used in the case a reject message is sent is: <i>Rejection</i> : The receiving party could not handle the request or the advice.
2	ProtocolVersion	[1..1]			see <i>AcceptorAuthorisationResponse</i>
2	ExchangeIdentification	[0..1]			Copy from the request if successfully extracted.
2	CreationDateTime	[1..1]			see <i>AcceptorAuthorisationResponse</i>
2	InitiatingParty	[0..1]			Copy from the request if successfully extracted .
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]			
3	ShortName	[0..1]			
2	RecipientParty	[0..1]			Copy from the request if successfully extracted, otherwise absent.
3	Identification	[0..1]			
3	Type	[1..1]			
3	Issuer	[0..1]			
3	Country	[0..1]			
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	Reject	[1..1]			
2	RejectReason	[1..1]			High level information allowing the sender of a request or an advice to know the types of error, and handle them accordingly (see section 3.4 p. 97). <i>UnableToProcess</i> : Not possible to process the

<i>Lvl</i>	<i>AcceptorRejection</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
					<p>message (e.g. Host unavailable, Hardware Security Module unavailable, problem of resource)</p> <p><i>InvalidMessage</i>: Invalid message envelope:</p> <p><i>ParsingError</i>: Invalid message, at least one of the data element or data structure is not present, or the format or the content of one data element or one data structure is not correct.</p> <p>This addresses the formatting of the field and also high level accuracy of its content (e.g. MessageFunction).</p> <p><i>Security</i>: any security related error (e.g. invalid key, incorrect MAC result...)</p> <p><i>InitiatingParty</i>: Invalid identification data for the sender InitiatingParty.</p> <p><i>RecipientParty</i>: Invalid identification data for the the receiver RecipientParty.</p> <p><i>DuplicateMessage</i>: Duplicate message, i.e. same ExchangeIdentification and CreationDateTime as for a previous message from the initiator.</p> <p><i>ProtocolVersion</i>: The ProtocolVersion is not supported by the recipient.</p>
2	AdditionalInformation	[0..1]	Appli		<p>Additional information related to the sending of a reject message in response to a request or an advice.</p> <p>For logging purpose, in order to allow further analysis, statistics and deferred processing on the success or the failure of the request processing.</p>
2	MessageInError	[0..1]	Appli		<p>Message received by the recipient which has been rejected and produced this AcceptorRejection message.</p>

4.9 Dynamic Currency Conversion Messages

4.9.1 AcceptorCurrencyConversionRequest (caaa.016.001.04)

<i>Lvl</i>	<i>AcceptorCurrencyConversionRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
1	Header	[1..1]			
2	MessageFunction	[1..1]			
2	ProtocolVersion	[1..1]			
2	ExchangeIdentification	[1..1]			
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
2	RecipientParty	[0..1]	Config		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CurrencyConversionRequest	[1..1]			
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	POI	[1..1]			
4	Identification	[1..1]			
5	Identification	[1..1]			
5	Type	[0..1]			
5	Issuer	[0..1]			
5	ShortName	[0..1]			
4	SystemName	[0..1]			
4	GroupIdentification	[0..1]			
4	Capabilities	[0..1]			
5	CardReadingCapabilities	[0..*]			
5	CardholderVerification-Capabilities	[0..*]			
5	PINLengthCapabilities	[0..1]			
5	ApprovalCodeLength	[0..1]			
5	MaxScriptLength	[0..1]			
5	CardCaptureCapable	[0..1]			
5	OnLineCapabilities	[0..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	MessageCapabilities	[0..*]			
6	Destination	[1..*]			
6	AvailableFormat	[0..*]			
6	NumberOfLines	[0..1]			
6	LineWidth	[0..1]			
6	AvailableLanguage	[0..*]			
4	TimeZone	[0..1]			
4	TerminalIntegration	[0..1]			
4	Component	[0..*]			
5	Type	[1..1]			
5	Identification	[1..1]			
6	ItemNumber	[0..1]			
6	ProviderIdentification	[0..1]			
6	Identification	[0..1]			
6	SerialNumber	[0..1]			
5	Status	[0..1]			
6	VersionNumber	[0..1]			
6	Status	[0..1]			
6	ExpiryDate	[0..1]			
5	StandardCompliance	[0..*]			
6	Identification	[1..1]			
6	Version	[1..1]			
6	Issuer	[1..1]			
5	Characteristics	[0..1]			
6	Memory	[0..*]			
7	Identification	[1..1]			
7	TotalSize	[1..1]			
7	FreeSize	[1..1]			
7	Unit	[1..1]			
6	Communication	[0..*]			
7	CommunicationType	[1..1]			
7	RemoteParty	[1..*]			
7	Active	[1..1]			
6	SecurityAccessModules	[0..1]			
6	SubscriberIdentityModules	[0..1]			
6	KeyCheckValue	[0..1]			
6	KeyCharacteristic	[0..1]			
7	KeyIdentification	[1..1]			
7	KeyVersion	[1..1]			
7	SequenceNumber	[0..1]			
7	DerivationIdentification	[0..1]			
7	Type	[0..1]			
7	Function	[0..1]			
6	EncryptedKey	[0..1]			
5	Assessment	[0..*]			
6	Type	[1..1]			
6	Assigner	[1..*]			
6	DeliveryDate	[0..1]			
6	ExpirationDate	[0..1]			
6	Number	[1..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
3	Card	[1..1]			
4	ProtectedCardData	[0..1]			
4	PlainCardData	[0..1]			
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]			
5	EffectiveDate	[0..1]			
5	ExpiryDate	[1..1]			
5	ServiceCode	[0..1]			
5	Track1	[0..1]			
5	Track2	[0..1]			
5	Track3	[0..1]			
5	CardholderName	[0..1]			
4	PaymentAccountReference	[0..1]			
4	IssuerBIN	[0..1]			
4	CardCountryCode	[0..1]			
4	CardCurrencyCode	[0..1]			
4	CardProductProfile	[0..1]			
4	CardBrand	[0..1]			
4	InternationalCard	[0..1]			
4	AllowedProduct	[0..*]			
4	ServiceOption	[0..1]			
4	AdditionalCardData	[0..1]			
3	CardholderLanguage	[0..1]			
2	Transaction	[1..1]			
3	TransactionCapture	[0..1]			
3	TransactionType	[1..1]			
3	AdditionalService	[0..*]			
3	ServiceAttribute	[0..1]			
3	MerchantCategoryCode	[1..1]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	OriginalTransaction	[0..1]			
4	SaleReferenceIdentification	[0..1]			
4	TransactionIdentification	[1..1]			
5	TransactionDateTime	[1..1]			
5	TransactionReference	[1..1]			
4	POIdentification	[1..1]			
5	Identification	[1..1]			
5	Type	[0..1]			
5	Issuer	[0..1]			
5	ShortName	[0..1]			
4	CurrencyConversion	[1..1]			
5	Result	[1..1]			
5	ResultReason	[0..1]			
5	ConversionDetails	[0..1]			
6	CurrencyConversion- Identification	[0..1]			
6	TargetCurrency	[1..1]			

Lvl	AcceptorCurrencyConversionRequest	Mult.	Rule	Cstr	Usage
7	AlphaCode	[1..1]			
7	NumericCode	[1..1]			
7	Decimal	[1..1]			
7	Name	[0..1]			
6	ResultingAmount	[1..1]			
6	ExchangeRate	[1..1]			
6	InvertedExchangeRate	[0..1]			
6	QuotationDate	[0..1]			
6	ValidUntil	[0..1]			
6	SourceCurrency	[1..1]			
7	AlphaCode	[1..1]			
7	NumericCode	[1..1]			
7	Decimal	[1..1]			
7	Name	[0..1]			
6	OriginalAmount	[1..1]			
7	ActualAmount	[0..1]			
7	MinimumAmount	[0..1]			
7	MaximumAmount	[0..1]			
6	CommissionDetails	[0..*]			
7	Amount	[1..1]			
7	AdditionalInformation	[0..1]			
6	MarkUpDetails	[0..*]			
7	Rate	[1..1]			
7	AdditionalInformation	[0..1]			
6	DeclarationDetails	[0..1]			
7	Format	[0..1]			
7	MessageContent	[1..1]			
3	TransactionDetails	[1..1]			
4	Currency	[1..1]			
4	TotalAmount	[1..1]			
4	AmountQualifier	[0..1]			
4	DetailedAmount	[0..1]			
5	AmountOfGoodAndServices	[0..1]			
5	CashBack	[0..1]			
5	Gratuity	[0..1]			
5	Fees	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Rebate	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	ValueAddedTax	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Surcharge	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
4	ICCRelatedData	[0..1]			
3	AdditionalTransactionData	[0..*]			
1	SecurityTrailer	[0..1]			

4.9.2 AcceptorCurrencyConversionResponse (caaa.017.001.04)

Lvl	AcceptorCurrencyConversionResponse	Mult.	Rule	Cstr	
1	Header	[1..1]			
2	MessageFunction	[1..1]			
2	ProtocolVersion	[1..1]			
2	ExchangeIdentification	[1..1]			
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]			
3	ShortName	[0..1]			
2	RecipientParty	[0..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]			
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CurrencyConversionResponse	[1..1]			
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	Card				
4	ProtectedCardData	[0..1]			
4	PlainCardData	[0..1]			
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]			
5	EffectiveDate	[0..1]			
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]			
3	ReconciliationIdentification	[0..1]			
3	InterchangeData	[0..1]			
3	TransactionDetails	[1..1]			
4	Currency	[1..1]			
4	TotalAmount	[1..1]			
4	DetailedAmount	[0..1]			
5	AmountOfGoodAndServices	[0..1]			
5	CashBack	[0..1]			
5	Gratuity	[0..1]			
5	Fees	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Rebate	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	ValueAddedTax	[0..*]			
6	Amount	[1..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	
6	Label	[0..1]			
5	Surcharge	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
4	InvoiceAmount	[0..1]			
5	Amount	[1..1]			
5	Label	[0..1]			
4	ValidityDate	[0..1]			
4	AccountType	[0..1]			
4	ICCRelatedData	[0..1]			
3	MerchantReferenceData	[0..1]			
2	CurrencyConversionResult	[1..1]			
3	Result	[1..1]			
3	ResultReason	[0..1]			
3	ConversionDetails	[0..1]			
4	CurrencyConversion- Identification	[0..1]			
4	TargetCurrency	[1..1]			
5	AlphaCode	[1..1]			
5	NumericCode	[1..1]			
5	Decimal	[1..1]			
5	Name	[0..1]			
4	ResultingAmount	[1..1]			
4	ExchangeRate	[1..1]			
4	InvertedExchangeRate	[0..1]			
4	QuotationDate	[0..1]			
4	ValidUntil	[0..1]			
4	SourceCurrency	[1..1]			
5	AlphaCode	[1..1]			
5	NumericCode	[1..1]			
5	Decimal	[1..1]			
5	Name	[0..1]			
4	OriginalAmount	[1..1]			
5	ActualAmount	[0..1]			
5	MinimumAmount	[0..1]			
5	MaximumAmount	[0..1]			
4	CommissionDetails	[0..*]			
5	Amount	[1..1]			
5	AdditionalInformation	[0..1]			
4	MarkUpDetails	[0..*]			
5	Rate	[1..1]			
5	AdditionalInformation	[0..1]			
4	DeclarationDetails	[0..1]			
5	Format	[0..1]			
5	MessageContent	[1..1]			
1	SecurityTrailer	[0..1]			

4.9.3 AcceptorCurrencyConversionAdvice (caaa.018.001.01)

Lvl	AcceptorCurrencyConversionAdvice	Mult.	Rule	Cstr	Usage
1	Header	[1..1]			
2	MessageFunction	[1..1]			
2	ProtocolVersion	[1..1]			
2	ExchangeIdentification	[1..1]			
2	RetransmissionCounter	[0..1]			default 0 see 3.3 Message Retransmission
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
2	RecipientParty	[0..1]	Config		
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]	Config		
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	AcceptorCurrencyConversionAdvice	[1..1]			
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[1..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POI	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	Card	[1..1]			
4	ProtectedCardData	[0..1]			
4	PlainCardData	[0..1]			
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]			
5	EffectiveDate	[0..1]			
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
3	SaleReferenceIdentification	[0..1]			
3	TransactionIdentification	[1..1]			
4	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
3	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]			
3	ReconciliationIdentification	[0..1]			
3	InterchangeData	[0..1]			
3	TransactionDetails	[1..1]			
4	Currency	[1..1]			
4	TotalAmount	[1..1]			
4	DetailedAmount	[0..1]			
5	AmountOfGoodAndServices	[0..1]			
5	CashBack	[0..1]			
5	Gratuity	[0..1]			
5	Fees	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Rebate	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	ValueAddedTax	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
5	Surcharge	[0..*]			
6	Amount	[1..1]			
6	Label	[0..1]			
4	InvoiceAmount	[0..1]			
5	Amount	[1..1]			
5	Label	[0..1]			
4	ValidityDate	[0..1]			
4	AccountType	[0..1]			
4	ICCRelatedData	[0..1]			
3	MerchantReferenceData	[0..1]			
2	CurrencyConversionResult	[1..1]			
3	AcceptedByCardholder	[0..1]			
3	Conversion	[0..1]			
4	CurrencyConversion- Identification	[0..1]			
4	TargetCurrency	[1..1]			
5	AlphaCode	[1..1]			
5	NumericCode	[1..1]			
5	Decimal	[1..1]			
5	Name	[0..1]			
4	ResultingAmount	[1..1]			
4	ExchangeRate	[1..1]			
4	InvertedExchangeRate	[0..1]			
4	QuotationDate	[0..1]			
4	ValidUntil	[0..1]			
4	SourceCurrency	[1..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	<i>Usage</i>
5	AlphaCode	[1..1]			
5	NumericCode	[1..1]			
5	Decimal	[1..1]			
5	Name	[0..1]			
4	OriginalAmount	[1..1]			
5	ActualAmount	[0..1]			
5	MinimumAmount	[0..1]			
5	MaximumAmount	[0..1]			
4	CommissionDetails	[0..*]			
5	Amount	[1..1]			
5	AdditionalInformation	[0..1]			
4	MarkUpDetails	[0..*]			
5	Rate	[1..1]			
5	AdditionalInformation	[0..1]			
4	DeclarationDetails	[0..1]			
5	Format	[0..1]			
5	MessageContent	[1..1]			
1	SecurityTrailer	[0..1]			

4.9.4 AcceptorCurrencyConversionAdviceResponse (caaa.019.001.01)

<i>Lvl</i>	<i>AcceptorCurrencyConversionAdvice-Response</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	
1	Header	[1..1]			
2	MessageFunction	[1..1]			
2	ProtocolVersion	[1..1]			
2	ExchangeIdentification	[1..1]			
2	RetransmissionCounter	[0..1]	CCopy		
2	CreationDateTime	[1..1]			
2	InitiatingParty	[1..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]			
3	ShortName	[0..1]			
2	RecipientParty	[0..1]			
3	Identification	[1..1]			
3	Type	[0..1]			
3	Issuer	[0..1]			
3	Country	[0..1]			
3	ShortName	[0..1]			
3	RemoteAccess	[0..1]			
4	Address	[1..*]			
5	NetworkType	[1..1]			
5	AddressValue	[1..1]			
4	UserName	[0..1]			
4	AccessCode	[0..1]			
4	ServerCertificate	[0..*]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionAdvice-Response</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	
4	ServerCertificateIdentifier	[0..*]			
4	ClientCertificate	[0..*]			
4	SecurityProfile	[0..1]			
2	Traceability	[0..*]			
3	RelayIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	ProtocolName	[0..1]			
3	ProtocolVersion	[0..1]			
3	TraceDateTimeIn	[1..1]			
3	TraceDateTimeOut	[1..1]			
1	CurrencyConversionAdviceResponse	[1..1]			
2	Environment	[1..1]			
3	AcquirerIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	Country	[0..1]			
4	ShortName	[0..1]			
3	MerchantIdentification	[0..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	POIIdentification	[1..1]			
4	Identification	[1..1]			
4	Type	[0..1]			
4	Issuer	[0..1]			
4	ShortName	[0..1]			
3	Card				
4	ProtectedCardData	[0..1]			
4	PlainCardData	[0..1]			
5	PAN	[1..1]			
5	CardSequenceNumber	[0..1]			
5	EffectiveDate	[0..1]			
5	ExpiryDate	[1..1]			
4	PaymentAccountReference	[0..1]			
4	MaskedPAN	[0..1]			
4	CardBrand	[0..1]			
4	CardProductType	[0..1]			
4	CardProductSubType	[0..1]			
3	PaymentToken	[0..1]			
4	TokenCharacteristic	[0..*]			
4	TokenAssuranceLevel	[0..1]			
2	Transaction	[1..1]			
2	SaleReferenceIdentification	[0..1]			

<i>Lvl</i>	<i>AcceptorCurrencyConversionAdvice-Response</i>	<i>Mult.</i>	<i>Rule</i>	<i>Cstr</i>	
3	TransactionIdentification	[1..1]			
3	TransactionDateTime	[1..1]			
4	TransactionReference	[1..1]			
4	InitiatorTransactionIdentification	[0..1]			
3	RecipientTransactionIdentification	[0..1]			
3	ReconciliationIdentification	[0..1]			
3	Response	[1..1]			
2	TMSTrigger	[0..1]			
3	TMSContactLevel	[1..1]			
3	TMSIdentification	[0..1]			
3	TMSContactDateTime	[0..1]			
1	SecurityTrailer	[0..1]			

5 Dynamic of the Payment Exchanges

5.1 Introduction

This chapter presents an exhaustive catalogue of payment transactions cases by:

- Listing exhaustively the cases from the various process flows of the payment transaction,
- presenting the associated sequences of Authorisation, Completion and Batch exchanges between an Acceptor and an Acquirer,
- determining the values of the message data components and configuration parameters relevant for these exchanges of messages, the financial capture and the outcome of the transaction.

5.2 Determination of Payment Cases

5.2.1 Elements Impacting the Message Flow

The following sections address the message flows of the transactions by focusing on conditions influencing the sequence of messages such as:

- authorisation type,
- authorisation result,
- incidents after the authorisation,
- transactions forced by the merchant,
- capture type, and
- use of completion messages,
- acquirer configuration parameters.

5.2.1.1 Authorisation Type

An authorisation can be made:

- online (exchange of authorisation messages between the Acceptor and the Acquirer)
- offline (no exchange of messages between the Acceptor and the Acquirer). In such a case, the payment approval follows the rules of the POI payment application.

AcceptorCompletionAdvice or BatchTransfer messages contains the information indicating whether the authorisation has been made online or offline in the data element *Context.PaymentContext.OnLineContext*.

5.2.1.2 Authorisation Result

The result of the authorisation can be:

- *Approved* online by the Acquirer or offline at the POI.
- *Declined* online by the Acquirer or offline at the POI.
- *No Acceptable Response*:
 - 1/ When the Acceptor has tried to send an *AcceptorAuthorisationRequest* message to the Acquirer without receiving a response in time (online authorisation only):
 - *Transaction.FailureReason* contains one of the values *UnableToSend*, *TimeOut* or *TooLateResponse*,
 - *Transaction.Reversal* flag is set to *True*.
 - 2/ When the *AcceptorAuthorisationResponse* contains *ResponseToAuthorisation/Response* with the value "TechnicalError":

5.2.1.3 Incident after Authorisation

After an online or offline authorisation, the transaction can terminate with incident.

Transaction.FailureReason has then one of the following values:

- *CustomerCancel*, transaction cancelled by the Cardholder
- *CardDeclined*, payment transaction finally declined by the card,
- *Malfunction*, malfunction of the card or of the card reader
- *UnableToComplete*, POI or Sale unable to complete transaction after the authorisation (e.g. written signature invalid, risk too high for the Acceptor).

5.2.1.4 Merchant Forced Acceptance

The Merchant (Acceptor) may force the acceptance of a transaction (eg. in case of a trusted or loyal customer).

A transaction can only be forced if it is allowed by the Acquirer and in accordance with the rules of the card scheme.

The option to force a transaction may be presented to the Acceptor when a transaction does not complete as approved, since:

- No response was received
- The transaction was declined
- Some incident occurred after authorisation at the POI.

If the Merchant forces the payment transaction, in the AcceptorCompletionAdvice message the following values are used:

- Transaction.MerchantOverride flag is set to *True*,
- Transaction.FailureReason is present with one of the values: *CardDeclined*, *Malfunction*, *OfflineDeclined*, *OnlineDeclined*, *TimeOut*, *TooLateResponse*, *UnableToSend* or *UnableToComplete* value.

5.2.1.5 Capture Type

The capture of the transaction can occur in:

- the authorisation exchange, when the configuration parameter OnlineTransaction.FinancialCapture has the value *Authorisation*.
- the completion exchange, when the configuration parameter OnlineTransaction.FinancialCapture (if the authorisation was online) or OfflineTransaction.FinancialCapture (if the authorisation was offline) has the value *Completion*.
- a batch transfer, when the configuration parameter OnlineTransaction.FinancialCapture (if the authorisation was online) or OfflineTransaction.FinancialCapture (if the authorisation was offline) has the value *Batch*.

This configuration information is neither sent in the AcceptorAuthorisationRequest, nor in the AcceptorCompletionAdvice messages.

5.2.1.6 Completion Exchange

A completion exchange for successful approved transactions is required depending on the value of the TMS parameter `CompletionExchange.ExchangePolicy`:

- *None*: a Completion advice is never sent to the Acquirer (for offline authorisation only),
- Any other value defined in `ExchangePolicy` except *OnDemand*: a Completion advice is always sent to the Acquirer. In addition, the value of the parameter determines the conditions of the exchange (e.g. *TotalLimit*: exchange performed after reaching a cumulative amount of transactions). The `ExchangePolicy` has the priority over a `CompletionRequired` flag set to *False*.

On demand `CompletionAdvice` is sent after the Authorisation exchange if the `CompletionRequired` flag of the `AcceptorAuthorisationResponse` message is set to *True*. The *OnDemand* value of the `CompletionExchange.ExchangePolicy` has no effect on the completion behavior.

Regardless of the above values, a completion exchange is also required when:

- The transaction completes successfully and the capture is realised during the completion exchange.
- An online authorisation needs to be reversed (e.g. no acceptable response to an `AcceptorAuthorisationRequest` or incident after an approved authorisation).
- The merchant successfully forces the transaction (see 5.2.1.4) and the capture must be made on-line (i.e. with an authorisation exchange or a completion exchange).
- a successful voice authorisation was performed and the capture must be made with the completion exchange.
- The authorisation is declined or the payment transactions failed and the TMS parameter `BatchTransferContent` is set to send declined or failed transactions.

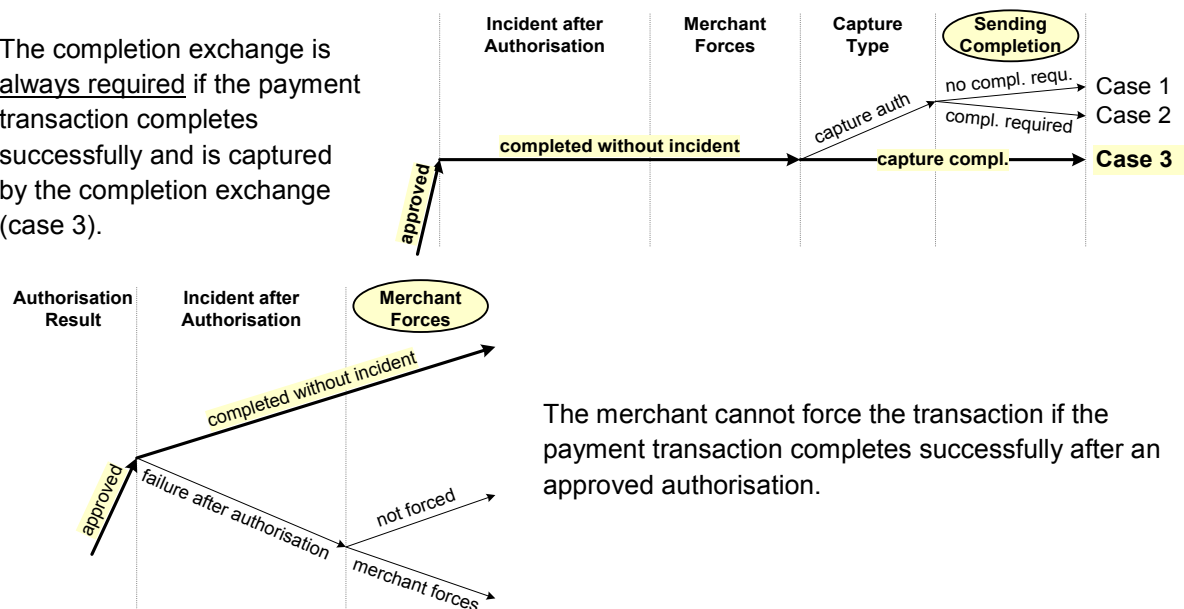
In case of reversal (`Transaction.Reversal` has the value *True*), the `AcceptorCompletionAdvice` message must be sent immediately after the transaction is completed, whatever the value of `CompletionExchange.ExchangePolicy`.

5.2.2 List of Payment Cases

In the following figures, *Figure 52: Payment Cases Tree List* and *Figure 53: Payment Cases Tree List (Con't)*, payment cases are identified through the combination of the elements described in the previous section.

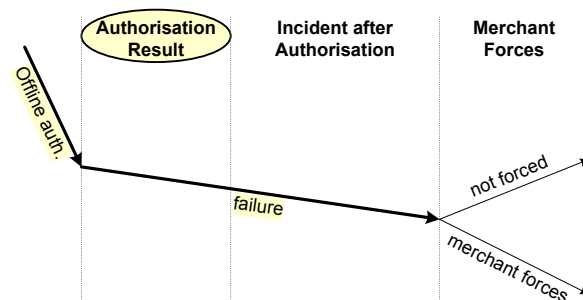
Some combinations may be irrelevant and so do not produce cases, for instance:

The completion exchange is always required if the payment transaction completes successfully and is captured by the completion exchange (case 3).



The merchant cannot force the transaction if the payment transaction completes successfully after an approved authorisation.

If an incident occurs for an offline authorisation, the result of the offline authorisation (declined or approved) is not taken into consideration.



On the right side of the tree, after the case number, the figure provides the sequence of exchange related to the case:

- **auth**: Authorisation exchange (AcceptorAuthorisationRequest/Response),
- **compl**: Completion exchange (AcceptorCompletionAdvice/Response),
- **reversal**: Completion exchange with reversal (AcceptorCompletionAdvice/Response),
- **batch**: Batch exchange (AcceptorBatchTransfer/Response)

The exchange is in bold and underlined if the capture is realised during that exchange, in square bracket if the exchange is optional, and define as reversal for a completion which is a reversal :

<u>auth</u> <u>compl</u>	Capture is done with the Authorisation exchange, the Completion is required by the Acquirer (without capture).
auth <u>compl</u>	Capture is done with the Completion exchange.
auth <u>batch</u>	Capture is done by Batch.
<u>auth</u> <u>reversal</u>	Capture is done with the Authorisation exchange, no response is received by the POI, a "financial reversal" is performed with the Completion.
auth [batch]	The payment transaction fails, the transaction must be included in the batch if the

Acceptor is configured to do so.

The form of "auth" and "compl" gives the value of the *Header.MessageFunction* component of the Authorisation and Completion messages, as summarized in the table below:

	<i>MessageFunction</i>	
auth	"AuthorisationRequest"	"AuthorisationResponse"
<u>auth</u>	"FinancialAuthorisationRequest"	"FinancialAuthorisationResponse"
compl	"CompletionAdvice"	"CompletionAdviceResponse"
<u>compl</u>	"FinancialCompletionAdvice"	"FinancialCompletionAdviceResponse"
reversal	"ReversalAdvice"	"ReversalAdviceResponse"
<u>reversal</u>	"FinancialReversalAdvice"	"FinancialReversalAdviceResponse"

Table 2: MessageFunction Values

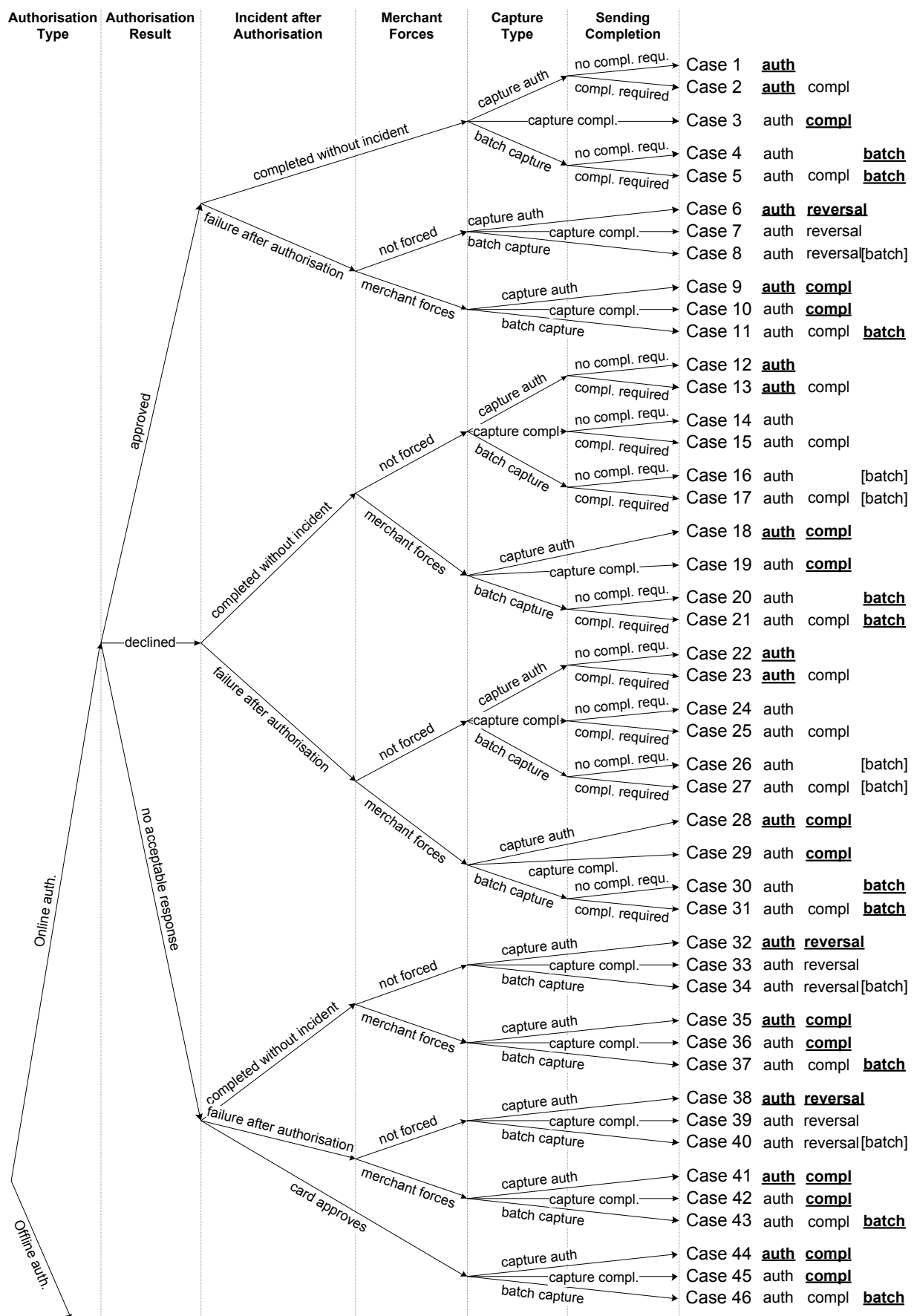


Figure 52: Payment Cases Tree List

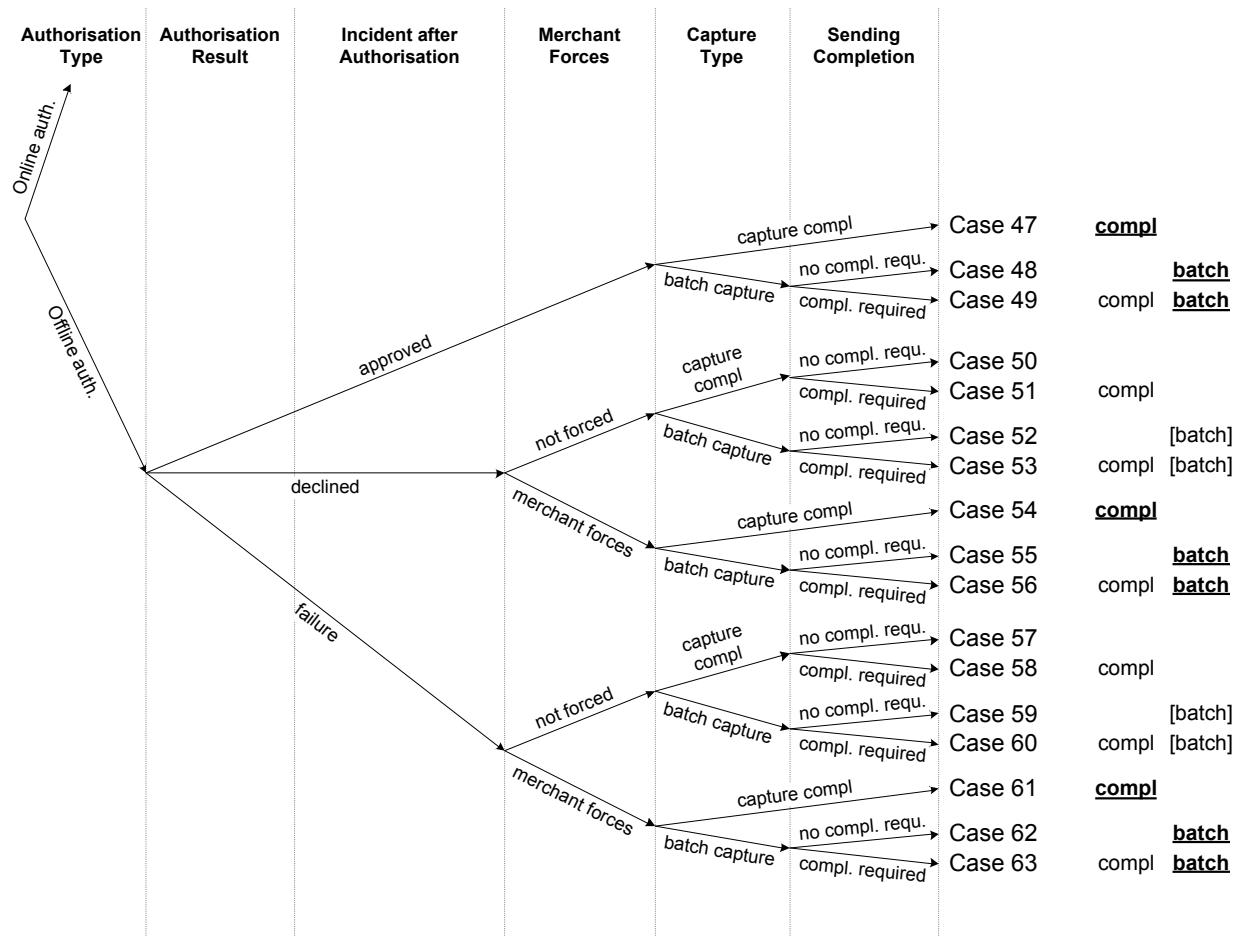


Figure 53: Payment Cases Tree List (Con't)

5.3 Table of Payment Cases

The table below provides for each case the value of the key data components inside the messages or configuration parameters.

It reads as follows:

Authorisation	Refers to both <i>AcceptorAuthorisationRequest</i> and <i>AcceptorAuthorisationResponse</i> messages.
CompletionAdvice	Refers to an <i>AcceptorCompletionAdvice</i> message.
Batch	Refers to a <i>BatchTransfer</i> message.

Authorisation	
<i>Txn Capt.</i>	Refers to <i>TransactionCapture</i> flag to be used in an <i>AcceptorAuthorisationRequest</i> message.
<i>Response</i>	Refers to <i>Response</i> data component to be used in an <i>AcceptorAuthorisationResponse</i> messages to give the outcome of the on-line authorisation: – Appr.: for the values <i>Approved</i> or <i>PartialApproved</i> . – Decl.: for the values <i>Declined</i> .
<i>Compl.Requ.</i>	An <i>AcceptorCompletionAdvice</i> message has to be sent in this instance. For declined and failed transaction, this flag has the priority over the configuration parameters <i>ExchangeDeclined</i> and <i>ExchangeFailed</i> .

CompletionAdvice	
<i>condition</i>	Express the condition for Completion exchange: – mandatory: the Completion exchange always occurs, – ExchPol = None (or ≠ None): the configuration parameter <i>OfflineTransaction.ExchangePolicy</i> or <i>OfflineTransaction.ExchangePolicy</i> has (or has not) the value <i>None</i> . – ExchDecl = True or False: the configuration parameter <i>ExchangeDeclined</i> has the value <i>True</i> or <i>False</i> . – ExchFail = True or False: the configuration parameter <i>ExchangeFailed</i> has the value <i>True</i> or <i>False</i> . – required : Completion is sent according to the <i>ExchangePolicy</i> .
<i>Txn. Succ.</i>	Refers to <i>TransactionSuccess</i> flag to be used in an <i>AcceptorCompletionAdvice</i> message to provide the success or the failure of the transaction. In addition to the value of this flag, it contains the value of <i>MessageFunction</i> in the Completion messages: (c): "CompletionAdvice" (fc): "FinancialCompletionAdvice" (r): "ReversalAdvice" (fr): "FinancialReversalAdvice"

<i>Txn. Capt.</i>	Refers to <i>TransactionCapture</i> flag to be used in an <i>AcceptorCompletionAdvice</i> message to indicate whether the transaction needs to be captured or not.
<i>Reversal</i>	Refers to <i>Reversal</i> flag to be used in an <i>AcceptorCompletionAdvice</i> message to indicate whether the message is to be considered as a reversal or not.
<i>Failure Reason</i>	Refers to <i>FailureReason</i> data component to be used in an <i>AcceptorCompletionAdvice</i> message which contains the reasons of the failure causing the reversals, merchant override or failure of the transaction. A "+" mean that an acceptor may send more than one failure reason.
<i>Merch. Overr.</i>	Refers to <i>MerchantOverride</i> flag to be used in an <i>AcceptorCompletionAdvice</i> message to indicate whether the merchant forced the acceptance of the transaction or not.

Batch	
<i>condition</i>	<p>Express the condition to include the payment transaction in a Batch transfer:</p> <ul style="list-style-type: none"> – Empty cell: the transaction is never included in a Batch transfer, – BatchCont=DebitCredit: the transaction is included in the Batch transfer if the configuration parameter BatchTransferContent has the value <i>DebitCredit</i>, – BatchCont=Declined: the transaction is included in the Batch transfer if the configuration parameter BatchTransferContent has the value <i>Declined</i>, – BatchCont= Failed: the transaction is included in the Batch transfer if the configuration parameter BatchTransferContent has the value <i>Failed</i>. <p>The number below the condition refers to the line number containing the value of the flags <i>TransactionSuccess</i>, <i>Reversal</i>, <i>MerchantOverride</i>, and the data elements <i>FailureReason</i> and <i>Response</i> of the related transaction in the <i>BatchTransfer</i>.</p>

An empty cell means that the component is absent.

		Authorisation			CompletionAdvice						Batch
		<i>Txn Capt.</i>	<i>Resp.</i>	<i>Compl Requ</i>	<i>condition</i>	<i>Txn Succ.</i>	<i>Txn Capt.</i>	<i>Rev-ersal</i>	<i>Failure Reason</i>	<i>Merch. Overr.</i>	<i>Condition</i>
1	Online approved Capture Auth. No Completion	True	Appr.	False							
2	Online approved Capture Auth. Completion	True	Appr.	True absent	required	True (c)	False				
3	Online approved Capture Compl.	False	Appr.	absent	mandatory	True (fc)	True				
4	Online approved Batch Capture Compl. not requ.	False	Appr.	False	ExchPol=None						BatchCont=DebitCredit 3
5	Online approved Batch Capture Compl. required	False	Appr.	True absent	ExchPol=None required	True (c)	False				BatchCont=DebitCredit 3

		Authorisation			CompletionAdvice						Batch
		Txn Capt.	Resp.	Compl Requ	condition	Txn Succ.	Txn Capt.	Rev-ersal	Failure Reason	Merch. Overr.	Condition
6	Online approved Incident after Auth. Capture Auth.	True	Appr.		mandatory	False (fr)	True	True	CustCancel Malfunction CardDecline Unab2Com		
7	Online approved Incident after Auth. Capture Compl.	False	Appr.	absent	mandatory	False (r)	False	True	CustCancel Malfunction CardDecline Unab2Com		
8	Online approved Incident after Auth. Batch Capture	False	Appr.		mandatory	False (r)	False	True	CustCancel Malfunction CardDecline Unab2Com		BatchCont= Failed 8
9	Online approved Incident after Auth. Merchant forces Capture Auth.	True	Appr.		mandatory	True (fc)	True	True	Malfunction CardDecline Unab2Com	True	
10	Online approved Incident after Auth. Merchant forces Capture Compl.	False	Appr.	absent	mandatory	True (fc)	True	True	Malfunction CardDecline Unab2Com	True	
11	Online approved Incident after Auth. Merchant forces Batch Capture	False	Appr.	True absent	 required	True (c)	False	True	Malfunction CardDecline Unab2Com	True	BatchCont= DebitCredit 10
12	Online declined Capture Auth. Compl. not requ.	True	Decl.	False absent	 ExchDecl=False						
13	Online declined Capture Auth. Compl. required	True	Decl.	True absent	 ExchDecl=True	False (c)	False		OnlineDecl		
14	Online declined Capture Compl. Compl. not requ.	False	Decl.	False absent	 ExchDecl=False						
15	Online declined Capture Auth. Compl. required	False	Decl.	True absent	 ExchDecl=True	False (c)	False		OnlineDecl		
16	Online declined Batch Capture Compl. not requ.	False	Decl.	False absent	 ExchDecl=False						BatchCont= Declined 17
17	Online declined Batch Capture Compl. required	False	Decl.	True absent	 ExchDecl=True	False (c)	False		OnlineDecl		BatchCont= Declined 17
18	Online declined Merchant forces Capture Auth.	True	Decl.	absent or present	mandatory	True (fc)	True		OnlineDecl	True	
19	Online declined Merchant forces Capture Compl.	False	Decl.	absent or present	mandatory	True (fc)	True		OnlineDecl	True	

		Authorisation			CompletionAdvice						Batch
		Txn Capt.	Resp.	Compl Requ	condition	Txn Succ.	Txn Capt.	Rev-ersal	Failure Reason	Merch. Overr.	Condition
20	Online declined Merchant forces Batch Capture Compl. not requ.	False	Decl.	False							BatchCont= DebitCredit 19
21	Online declined Merchant forces Batch Capture Compl. required	False	Decl.	True absent	required	True (c)	False		OnlineDecl	True	BatchCont= DebitCredit 19
22	Online declined Incident after Auth. Capture Auth. Compl. not requ.	True	Decl.	False absent	ExchFail=False						
23	Online declined Incident after Auth. Capture Auth. Compl. required	True	Decl.	True absent	ExchFail=True	False (c)	False		OnlineDecl + CustCancel Malfunction CardDecline Unab2Com		
24	Online declined Incident after Auth. Capture Compl. Compl. not requ.	False	Decl.	False absent	ExchFail=False						
25	Online declined Incident after Auth. Capture Compl. Compl. required	False	Decl.	True absent	ExchFail=True	False (c)	False		OnlineDecl + CustCancel Malfunction CardDecline Unab2Com		
26	Online declined Incident after Auth. Batch Capture Compl. not requ.	False	Decl.	False absent	ExchFail=False						BatchCont= Failed 27
27	Online declined Incident after Auth. Batch Capture Compl. required	False	Decl.	True absent	ExchFail=True	False (c)	False		OnlineDecl + CustCancel Malfunction CardDecline Unab2Com		BatchCont= Failed 27
28	Online declined Incident after Auth. Merchant forces Capture Auth.	True	Decl.	absent	mandatory	True (fc)	True		OnlineDecl + Malfunction CardDecline Unab2Com	True	
29	Online declined Incident after Auth. Merchant forces Capture Compl.	False	Decl.	absent	mandatory	True (fc)	True		OnlineDecl + Malfunction CardDecline Unab2Com	True	
30	Online declined Incident after Auth. Merchant forces Batch Capture Compl. not requ.	False	Decl.	False	ExchPol=OnDem						BatchCont= DebitCredit 29
3	Online declined	False	Decl.	True		True	False		OnlineDecl	True	BatchCont=

		Authorisation			CompletionAdvice						Batch
		<i>Txn Capt.</i>	<i>Resp.</i>	<i>Compl Requ</i>	<i>condition</i>	<i>Txn Succ.</i>	<i>Txn Capt.</i>	<i>Rev-ersal</i>	<i>Failure Reason</i>	<i>Merch. Overr.</i>	<i>Condition</i>
1	Incident after Auth. Merchant forces Batch Capture Compl. required			absent	required	(c)			+ Malfunction CardDecline Unab2Com		DebitCredit 29
3 2	Online no resp. Capture Auth.	True	Appr. or Decl.		mandatory	False (fr)	True	True	TimeOut Unable2Send		
3 3	Online no resp. Capture Compl.	False	Appr. or Decl.	absent	required	False (r)	False	True	TimeOut Unable2Send		
3 4	Online no resp. Batch Capture	False	Appr. or Decl.		mandatory	False (r)	False	True	TimeOut Unable2Send		BatchCont= Failed 34
3 5	Online no resp. Merchant forces Capture Auth.	True	Appr. or Decl.		mandatory	True (fc)	True	True	TimeOut Unable2Send	True	
3 6	Online no resp. Merchant forces Capture Compl.	False	Appr. or Decl.		required	True (fc)	True	True	TimeOut Unable2Send	True	
3 7	Online no resp. Merchant forces Batch Capture	False	Appr. or Decl.		required	True (c)	False	True	TimeOut Unable2Send	True	BatchCont= DebitCredit 36
3 8	Online no resp. Incident after Auth. Capture Auth.	True	Appr. or Decl.		mandatory	False (fr)	True	True	TimeOut Unable2Send + CustCancel Malfunction CardDecline Unab2Com		
3 9	Online no resp. Incident after Auth. Capture Compl.	False	Appr. or Decl.		required	False (r)	False	True	TimeOut Unable2Send + CustCancel Malfunction CardDecline Unab2Com		
4 0	Online no resp. Incident after Auth. Batch Capture	False	Appr. or Decl.		required	False (r)	False	True	TimeOut Unable2Send + CustCancel Malfunction CardDecline Unab2Com		BatchCont= Failed 40

		Authorisation			CompletionAdvice						Batch
		Txn Capt.	Resp.	Compl Requ	condition	Txn Succ.	Txn Capt.	Rev-ersal	Failure Reason	Merch. Overr.	Condition
4 1	Online no resp. Incident after Auth. Merchant forces Capture Auth.	True	Appr. or Decl.		mandatory	True (fc)	True	True	TimeOut Unable2Send + Malfunction CardDecline Unab2Com	True	
4 2	Online no resp. Incident after Auth. Merchant forces Capture Compl.	False	Appr. or Decl.	absent	mandatory	True (fc)	True	True	TimeOut Unable2Send + Malfunction CardDecline Unab2Com	True	
4 3	Online no resp. Incident after Auth. Merchant forces Batch Capture	False	Appr. or Decl.		required	True (c)	False	True	TimeOut Unable2Send + Malfunction CardDecline Unab2Com	True	BatchCont= DebitCredit 42
4 4	Online no resp. Card approves Capture Auth.	True	Appr. or Decl.		mandatory	True (fc)	True	True	TimeOut Unable2Send		
4 5	Online no resp. Card approves Capture Compl.	False	Appr. or Decl.	absent	mandatory	True (fc)	True	True	TimeOut Unable2Send		
4 6	Online no resp. Card approves Batch Capture	False	Appr. or Decl.		required	True (c)	False	True	TimeOut Unable2Send		BatchCont= DebitCredit 45
4 7	Offline approved Capture Compl.				mandatory	True (fc)	True				
4 8	Offline approved Batch Capture Compl. not requ.				ExchPol=None						BatchCont= DebitCredit 47
4 9	Offline approved Batch Capture Compl. required				required	True (c)	False				BatchCont= DebitCredit 47
5 0	Offline declined Capture Compl. Compl. not requ.				ExchDecl=False						
5 1	Offline declined Capture Compl. Compl. required				ExchDecl=True	False (c)	False		OfflineDecl		
5 2	Offline declined Batch Capture Compl. not requ.				ExchDecl=False						BatchCont= Declined 53
5 3	Offline declined Batch Capture Compl. required				ExchDecl=True	False (c)	False		OfflineDecl		BatchCont= Declined 53
5 4	Offline declined Merchant forces Capture Compl.				mandatory	True (fc)	True		OfflineDecl	True	

		Authorisation			CompletionAdvice						Batch
		<i>Txn Capt.</i>	<i>Resp.</i>	<i>Compl Requ</i>	<i>condition</i>	<i>Txn Succ.</i>	<i>Txn Capt.</i>	<i>Rev-ersal</i>	<i>Failure Reason</i>	<i>Merch. Overr.</i>	<i>Condition</i>
5	Offline declined Merchant forces Batch Capture Compl. not requ.				ExchPol=None						BatchCont= DebitCredit 54
5 6	Offline declined Merchant forces Batch Capture Compl. required				required	True (c)	False		OfflineDecl	True	BatchCont= DebitCredit 54
5 7	Offline failure Capture Compl. Compl. not requ.				ExchFail=False						
5 8	Offline failure Capture Compl. Compl. required				ExchFail=True	False (c)	False		CustCancel Malfunction CardDecline Unab2Com		
5 9	Offline failure Batch Capture Compl. not requ.				ExchFail=False						BatchCont= Failed 60
6 0	Offline failure Batch Capture Compl. required				ExchFail=True	False (c)	False		CustCancel Malfunction CardDecline Unab2Com		BatchCont= Failed 60
6 1	Offline failure Merchant forces Capture Compl.				mandatory	True (fc)	True		Malfunction CardDecline Unab2Com	True	
6 2	Offline failure Merchant forces Batch Capture Compl. not requ.				ExchPol=None						BatchCont= DebitCredit 61
6 3	Offline failure Merchant forces Batch Capture Compl. required				required	True (c)	False		Malfunction CardDecline Unab2Com	True	BatchCont= DebitCredit 61

Table 3: List of Payment Cases

5.4 Cancellation Exchange

A cancellation exchange should rely on a configuration with 3 possible options:

- The usage of both *AcceptorCancellationRequest* and *AcceptorCancellationAdvice* messages to fulfil a cancellation
- The usage of solely *AcceptorCancellationAdvice* message to fulfil a cancellation
- Cancellation is not supported by the acquirer.

5.4.1 List of Cancellation Cases

In the Figure 54 Cancellation Cases Tree List cancellation cases are described, based on the type of capture and the combination of Cancellation Request and Cancellation Advice requested. For batch file, the tree also shows the case where cancelled transactions must (or must not) be included in the batch transfer.

Case 5 to 14 are the same, the only difference being the authorisation type. This has no impact on cancellation processing.

The tree also covers the case where the transaction is not available in the log file because of at least one of the following:

- the cancellation is done in a new batch
- the cancellation and the original transaction are not in the same reconciliation period
- the original transaction originates from another POI.

In this case cancellation request is mandatory and a cancellation advice must be sent if the request is approved.

In the right hand side of the tree, after the case number, the figure provides the sequence of exchanges related to the case:

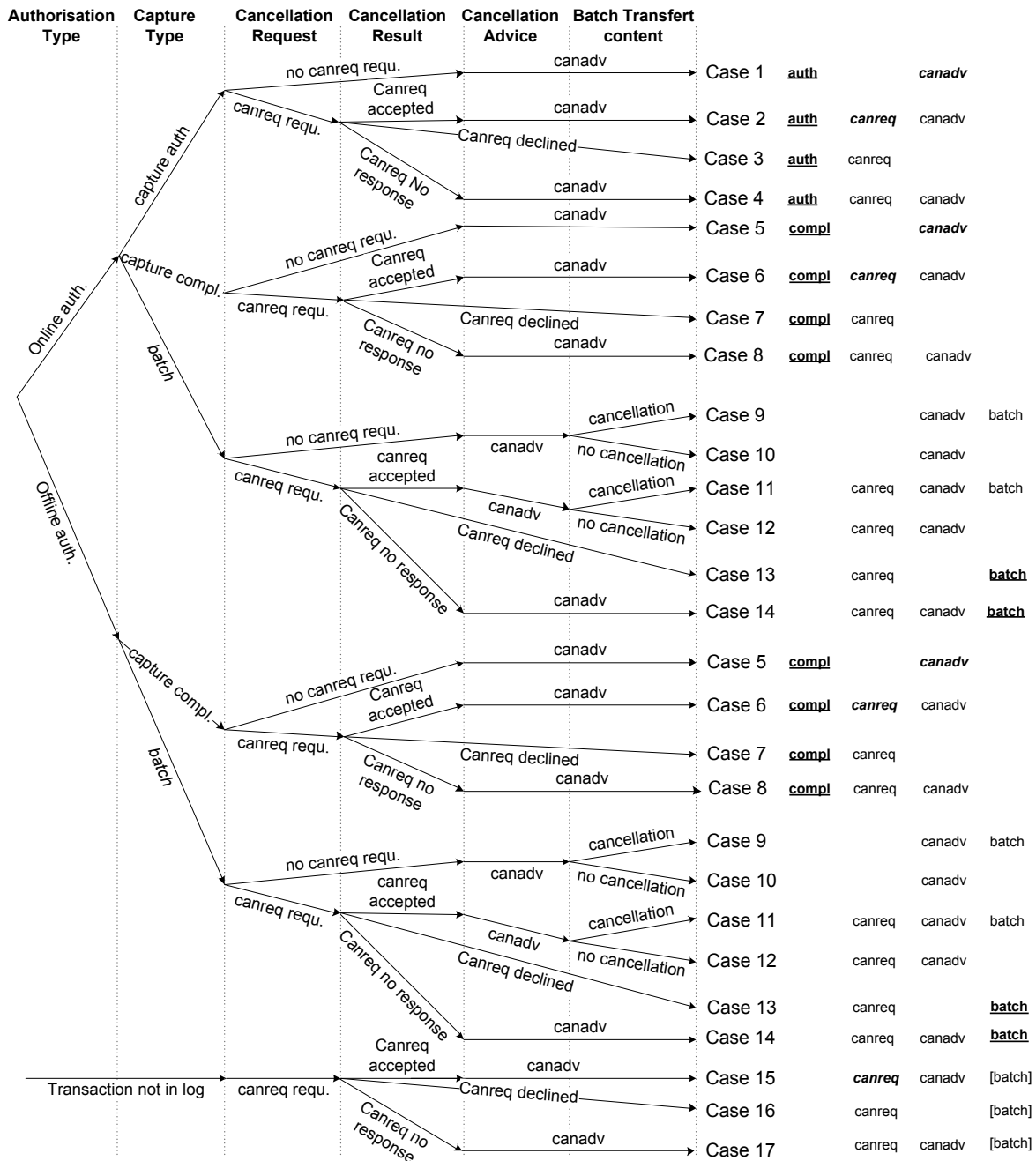
- auth: Authorisation exchange (AcceptorAuthorisationRequest/Response),
- compl: Completion exchange (AcceptorCompletionAdvice/Response),
- batch: Batch exchange (AcceptorBatchTransfer/Response)
- canreq: Cancellation request exchange (AcceptorCancellationRequest/Response)
- canadv: Cancellation advice exchange (AcceptorCancellationAdvice/Response)

The exchange is **in bold and underlined** if the capture was done in the payment processing. As for the payment case, the message is **in bold and underlined** if the financial capture was done. Cancellation request or advice are ***in bold and italic*** if the capture is undone.

<u>auth</u>	<u>canadv</u>	Capture was done on authorisation, no cancellation request and capture is undone with a cancellation advice.
<u>compl</u>	<u>canreq</u> <u>canadv</u>	Capture was done with a completion, a cancellation request is accepted and undo the capture, followed by a cancellation advice.
canreq	canadv batch	The capture hasn't occurred yet, transaction is cancelled by request and advice and both original transaction and cancellation are included in batch.

For cancellation, "canreq" and "canadv" gives the value of the *Header.MessageFunction* component of the Cancellation Request and CancellationAdvice messages, as summarized in the table below:

	<i>MessageFunction</i>	
canreq	"CancellationRequest"	"CancellationResponse"
canadv	"CancellationAdvice"	"CancellationAdviceResponse"

Table 4: Cancellation MessageFunction Values**Figure 54 Cancellation Cases Tree List**

5.4.2 Table of Cancellation Cases

The table below provides for each case the value of the key data components inside the messages or configuration parameters.

It reads as follows:

Payment Case	Refers to the possible payment that can be cancelled according to the cancellation case.
Cancellation Request	Refers to both <i>AcceptorCancellationRequest</i> and <i>AcceptorCancellationResponse</i> messages.
Cancellation Advice	Refers to an <i>AcceptorCancellationAdvice</i> message.
Batch	Refers to a <i>BatchTransfer</i> message.

Cancellation Request	
<i>condition</i>	Express the condition for Cancellation exchange: <ul style="list-style-type: none"> – CancExch = Advice: the configuration parameter OnlineTransaction. <i>CancellationExchange</i> or OfflineTransaction. <i>CancellationExchange</i> has the value <i>Advice</i>. – CancExch = Request: the configuration parameter OnlineTransaction. <i>CancellationExchange</i> or OfflineTransaction. <i>CancellationExchange</i> has the value <i>Request</i>.
<i>Response</i>	Refers to <i>Response</i> data component to be used in an <i>AcceptorCancellationResponse</i> messages to give the outcome of the requested cancellation: <ul style="list-style-type: none"> – Appr.: for the values <i>Approved</i>. – Decl.: for the values <i>Declined</i>.
<i>Capture undone</i>	Give the final outcome of the cancellation.

Cancellation Advice	
<i>Advice Sent</i>	Express the requirement to send or not an Advice.: – True: A cancellation advice MUST be sent. – False: A cancellation advice MUST NOT be sent.
<i>Capture undone</i>	Give the final outcome of the cancellation.
<i>Reverse Flag</i>	Refers to the <i>Reversal</i> component to be used in an <i>AcceptorCancellationAdvice</i> to inform the acquirer that no response was received for a <i>CancellationRequest</i> : – False: Default value. – True: no acceptable <i>CancellationResponse</i> message has been received, or cancellation couldn't be completed successfully after an approved cancellation request.

Batch	
<i>Cancellation in batch</i>	Express the condition to include the cancellation transaction in a Batch transfer: – Empty cell: the cancellation is never included in a Batch transfer, – BatchCont=Cancellation: the cancellation transaction is included in the Batch transfer because the configuration parameter BatchTransferContent has the value <i>Cancellation.</i> , – BatchCont≠Cancellation: the cancellation transaction is not included in the Batch transfer because the configuration parameter BatchTransferContent has not the value <i>Cancellation.</i> ,

		<i>Payment Case</i>	<i>Cancellation Request</i>			<i>Cancellation Advice</i>			<i>Batch</i>
			<i>Condition</i>	<i>Resp</i>	<i>Capture Undone</i>	<i>Advice sent</i>	<i>Capture undone</i>	<i>Reverse flag</i>	<i>Cancellation in batch</i>
1	Capture Auth. No cancellation requ. Cancellation advice	1,2	CancExch=Advice			True	True	False	
2	Capture Auth. Cancellation requ. Canc. approved Cancellation advice	1,2	CancExch=Request	Appr.	True	True	False	False	
3	Capture Auth. Cancellation requ. Canc. declined	1,2	CancExch=Request	Decl.	False	False			
4	Capture Auth. Cancellation requ. Canc. no response Cancellation advice	1,2	CancExch=Request	No resp.	False	True	False	True	

		Payment Case	Cancellation Request			Cancellation Advice			Batch
			Condition	Resp	Capture Undone	Advice sent	Capture undone	Reverse flag	Cancellation in batch
5	Capture Completion. No cancellation requ. Cancellation advice	3,9,10,18,19,28, 29,35,36,41,42, 44,45,47,54,61	CancExch= Advice			True	True	False	
6	Capture Completion Cancellation requ. Canc. approved Cancellation advice	3,9,10,18,19,28, 29,35,36,41,42, 44,45,47,54,61	CancExch= Request	Appr.	True	True	False	False	
7	Capture Completion Cancellation requ. Canc. declined	3,9,10,18,19,28, 29,35,36,41,42, 44,45,47,54,61	CancExch= Request	Decl.	False	False			
8	Capture Completion Cancellation requ. Canc. no response Cancellation advice	3,9,10,18,19,28, 29,35,36,41,42, 44,45,47,54,61	CancExch= Request	No resp.	False	True	False	True	
9	Batch Capture . No cancellation requ. Cancellation advice Cancellation in batch	4,5,11,20,21,30, 31,37,43,46,48, 49,55,56,62,63	CancExch= Advice			True	False	False	BatchCont= Cancellation
10	Batch Capture No cancellation requ. Cancellation advice Cancellation not in batch	4,5,11,20,21,30, 31,37,43,46,48, 49,55,56,62,63	CancExch= Advice			True	False	False	BatchCont≠ Cancellation
11	Batch Capture Cancellation requ. Canc. approved Cancellation advice Cancellation in batch	4,5,11,20,21,30, 31,37,43,46,48, 49,55,56,62,63	CancExch= Request	Appr.	True	True	False	False	BatchCont= Cancellation
12	Batch Capture Cancellation requ. Canc. approved Cancellation advice Cancellation not in batch	4,5,11,20,21,30, 31,37,43,46,48, 49,55,56,62,63	CancExch= Request	Appr.	True	True	False	False	BatchCont≠ Cancellation
13	Batch Capture Cancellation requ. Canc. declined	4,5,11,20,21,30, 31,37,43,46,48, 49,55,56,62,63	CancExch= Request	Decl.	False	False			
14	Batch Capture Cancellation requ. Canc. no response Cancellation advice	4,5,11,20,21,30, 31,37,43,46,48, 49,55,56,62,63	CancExch= Request	No resp.	False	True	False	True	
15	Transaction not in log Cancellation requ. Canc. approved Cancellation advice	1,2,3,4,5,9,10,1 1,18,19,20,21,2 8,29,30,31,35,3 6,37,41,42,,43,4 4,45,,46,47,48,4 9,54,55,56,61,6 2,63	N/A	Appr.	True	True	False	False	BatchCont≠ Cancellation

		<i>Payment Case</i>	<i>Cancellation Request</i>			<i>Cancellation Advice</i>			<i>Batch</i>
			<i>Condition</i>	<i>Resp</i>	<i>Capture Undone</i>	<i>Advice sent</i>	<i>Capture undone</i>	<i>Reverse flag</i>	<i>Cancellation in batch</i>
1 6	Transaction not in log Cancellation requ. Canc. declined	1,2,3,4,5,9,10,1 1,18,19,20,,21,2 8,29,30,31,35,3 6,37,41,42,43,4 4,45,,46,47,48,4 9,54,55,56,61,6 2,63	N/A	Decl.	False	False			BatchCont# Cancellation
1 7	Transaction not in log Cancellation requ. Canc. no response Cancellation advice	1,2,3,4,5,9,10,1 1,18,19,20,,21,2 8,29,30,31,35,3 6,37,41,42,43,4 4,45,,46,47,48,4 9,54,55,56,61,6 2,63	N/A	No resp.	False	True	False	True	BatchCont# Cancellation

Table 5: List of Cancellation Cases

6 Additional Payment Services

Unless stated otherwise, specifications presented in the previous sections refer to the standard payment (i.e. *Transaction.TransactionType* = "CardPayment" with *AdditionalService* and *ServiceAttribute* absents). This section contains specifications related to other values of *TransactionType*, or *AdditionalService* and *ServiceAttribute* when these are presents.

6.1 Voice Authorisation

Voice Authorisation is an additional feature to approve a transaction over the phone when requested in the *AcceptorAuthorisationResponse*.

1. The Acceptor performs an online authorisation, sending an *AcceptorAuthorisationRequest* for a main service populated in the *TransactionType* component.
2. The Acquirer declined the authorisation (*Response*= "Declined"), and requests a voice authorisation setting the action *Referral* in the message element *Action.ActionType* of the *AcceptorAuthorisationResponse* message. The Acceptor is expected to provide information included in the authorisation exchanged to the Voice Authorisation service verbally.
3. If the transaction is approved through the Voice Authorisation,
 - a. The Voice Authorisation service provides an authorisation code by phone to the Acceptor and the transaction can be successfully completed.
 - b. The transaction must be captured by completion or batch with the following information:
 - One occurrence of *AdditionalService* added and set to "VoiceAuthorisation",
 - *TransactionSuccess* equal "True",
 - *AuthorisationResult.AuthorisationCode* containing the authorisation code provided with the Acquirer,
 - *ResponseToAuthorisation.Response* set to "Declined",
 - c. In the Completion exchange *TransactionCapture* is present with the value "True" if the Acceptor is configured to make the financial capture with the Authorisation or the Completion.

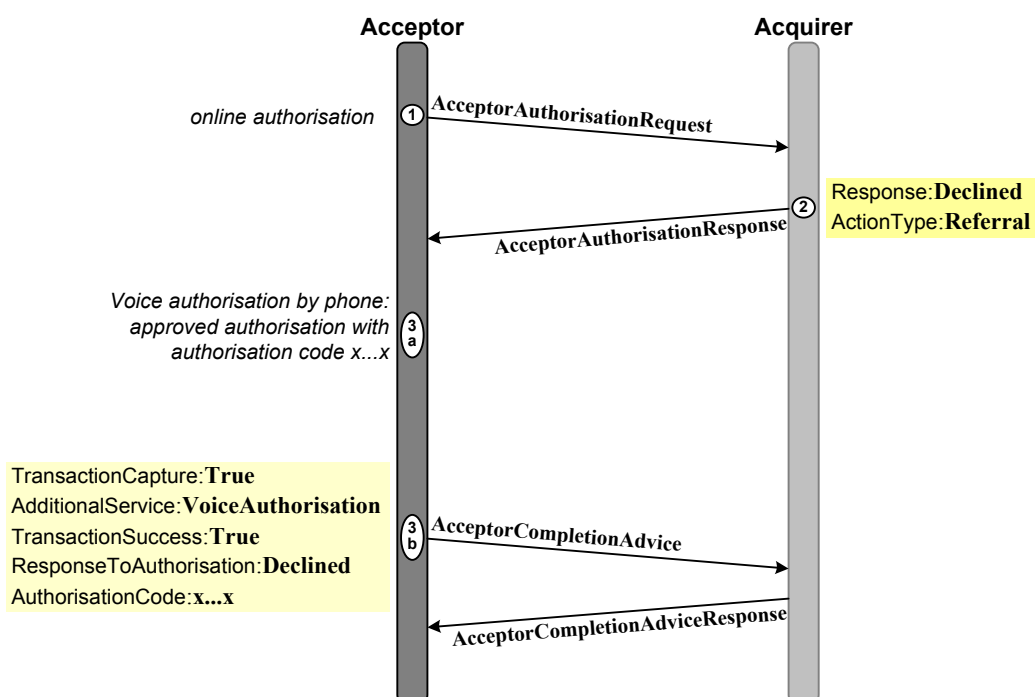


Figure 55: Successful Voice Authorisation captured with Completion

4. If the voice authorisation is declined, the transaction fails. The Acceptor must not send a capture, and has to follow the the standard process flow of the declined transactions for the Completion exchange (see section 5 *Dynamic of the Payment Exchanges*). If the transaction is sent in a completion or a batch, it contains the following informations:
- *TransactionCapture* present with the value "False".
 - An occurrence of *AdditionalService* must be added and set to "VoiceAuthorisation",
 - *TransactionSuccess* equal "False".
 - *AuthorisationResult.AuthorisationCode* is absent.
 - *ResponseToAuthorisation.Response* set to "Declined"

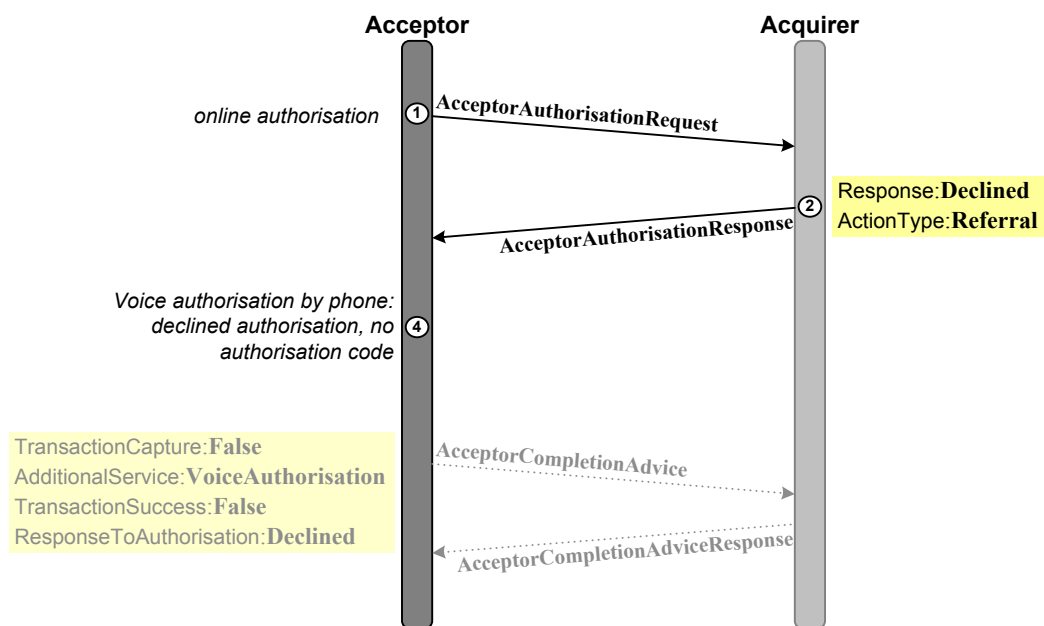


Figure 56: Unsuccessful Voice Authorisation captured with Completion.

The Acceptor is not necessary blocked between the declined/referral authorisation and the result of the Voice Authorisation. Other transactions may be interleaved between the declined/referral authorisation and the Completion containing the outcome of the Voice Authorisation. In the figure below:

- An AcceptorAuthorisationRequest is declined with the *ActionType* "Referral" (transaction 1);
- Another payment transaction is performed on the terminal with an online authorisation and a completion exchange (transaction 2);
- The Acceptor contact the voice authorisation center to get the authorisation code for transaction 1;
- The Acceptor initiates a Completion after entering the authorisation code.

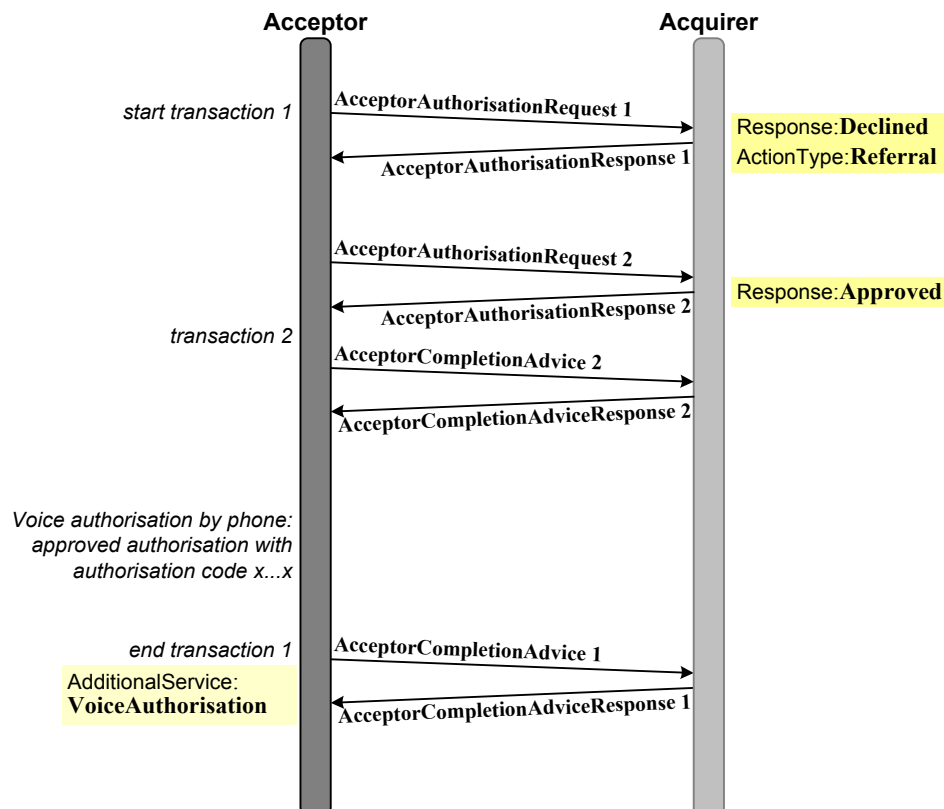


Figure 57: Successful Voice Authorisation Interleaved with another transaction

An example of successful Voice Authorisation with Batch capture and no Completion is presented in the figure below. The batch contains one occurrence of *TransactionToCapture* with:

- *AdditionalService* = "VoiceAuthorisation",
- *TransactionSuccess* = "True",
- *ResponseToAuthorisation.Response* = "Declined",
- *AuthorisationCode* containing the code provided by phone.

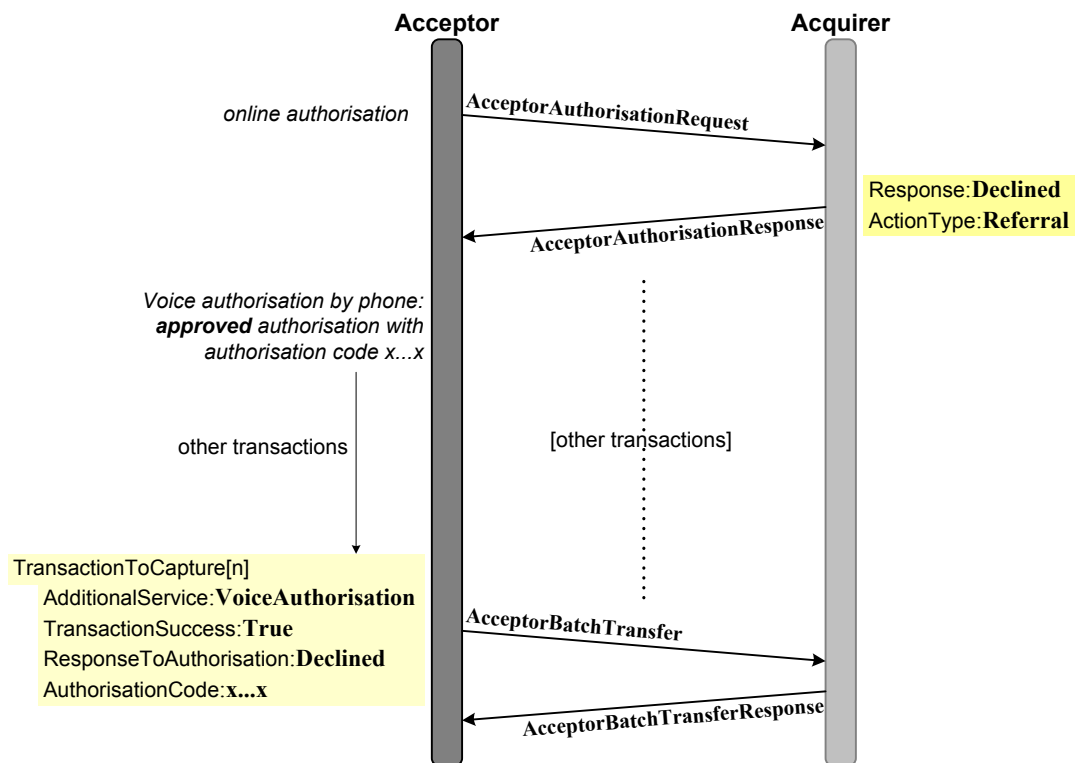


Figure 58: Successful Voice Authorisation Captured with Batch without Completion

6.2 Deferred Payments

6.2.1 Introduction

Deferred payment is a service which enable the Acceptor to:

- Request an authorisation to get a maximum amount to be able to pay,
- Complete the delivery of goods or use of service to be paid up to the maximum amount,
- Inform the Acquirer of the payment of these goods or services with the final amount.

The Authorisation and Completion exchanges are required to perform a deferred payment transaction.

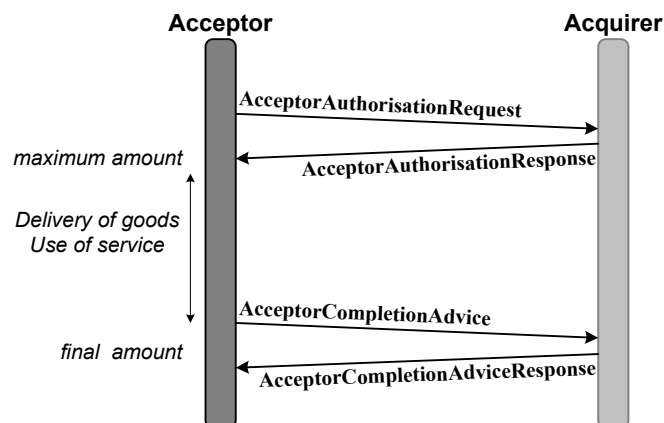


Figure 59: Deferred Payment

During delivery of goods, other payment transactions can be performed, and the two steps of a deferred payment transaction (Authorisation exchange and Completion exchange) may be interleaved.

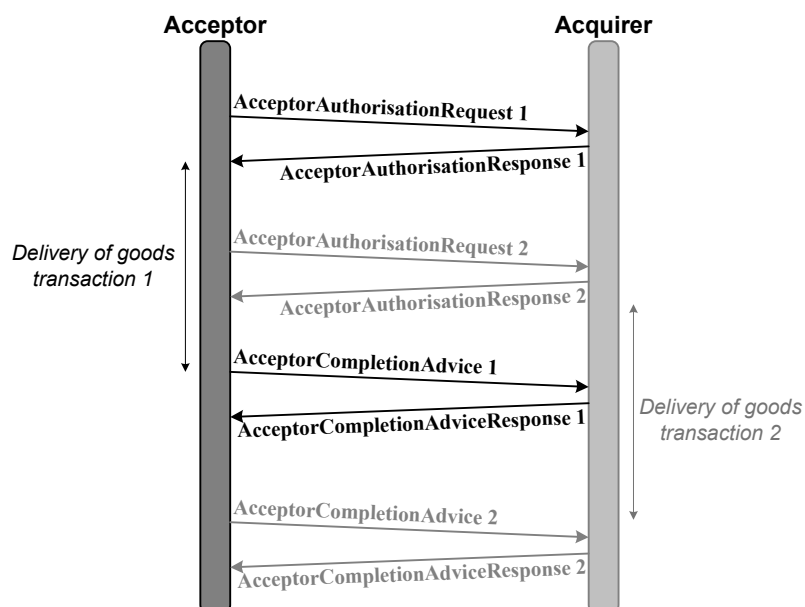


Figure 60: Interleaved Deferred Payment Transactions

A deferred payment transaction is characterised in the exchanged messages by the value "DeferredPayment" of the data element *TransactionType*.

6.2.2 Constraints on the Protocol

The deferred payment transaction respects the following rules, particular to this service:

1. The Authorisation is performed online.
2. If an approved `AcceptorAuthorisationResponse` is received by the Acceptor, and goods or services have been delivered, an `AcceptorCompletionAdvice` message must be sent with:
 - `MessageFunction` = "CompletionAdvice" or "FinancialCompletionAdvice",
 - `TotalAmount` with the final amount,
 - `Reversal` = "False" or absent,
 - `TransactionSuccess` = "True".

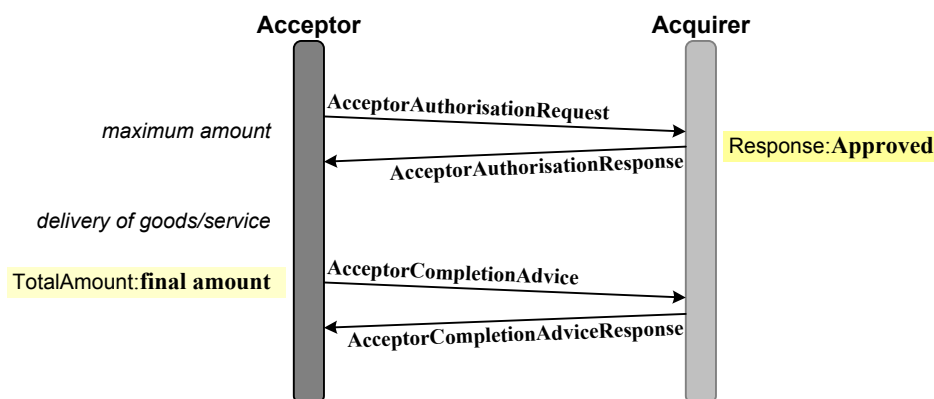


Figure 61: Approved Deferred Payment without Delivery

3. The Completion exchange is mandatory, except when the two following conditions are fulfilled:
 - the `AcceptorAuthorisationResponse` has been received by the Acceptor, the authorisation is declined and the payment transaction is not overridden by the Merchant, and
 - the Acceptor is not configured to send a completion for declined online authorisations (see section 5, *Dynamic of the Payment Exchanges*),

The `AcceptorAuthorisationResponse` of a declined deferred payment must have the final amount `TotalAmount` with a copy of the value sent in the `AcceptorAuthorisationRequest`.

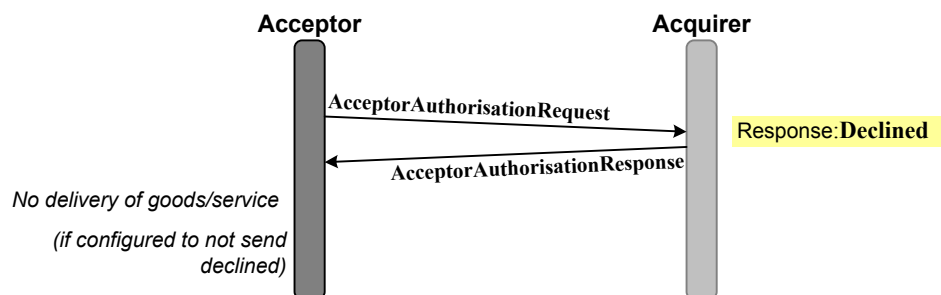


Figure 62: Declined Deferred Payment

4. If the `AuthorisationRequest` contains `AmountQualifier` value:
 - "Estimated", the `AuthorisationResponse` may contain `TotalAmount` higher, equal or lower than in the request,
 - "Maximum", the `AuthorisationResponse` may contain `TotalAmount` equal or lower than in the request,

If the authorisation is approved, `ResponseToAuthorisation.Response` must be :

- "Approved" when *TotalAmount* of the *AuthorisationResponse* is higher than or equal to the requested amount,
 - "PartialApproved" when *TotalAmount* of the *AuthorisationResponse* is lower than in the request.
5. If the authorisation is approved, the Completion exchange must be performed immediately after the finalisation of the delivery of goods or service with:
- A value of *TotalAmount* lower or equal the value of *TotalAmount* sent in the *AcceptorAuthorisationResponse*,
 - *AmountQualifier* absent or equal to "Actual".

The configuration parameters of the Completion (TMS parameters *OnlineTransaction.CompletionExchange* or *OfflineTransaction.CompletionExchange*) as well as the *CompletionRequired* of the *AuthorisationResponse* are ignored.

6. The financial capture must be performed either within the Completion exchange, or through a batch transfer. It cannot be performed as part of the Authorisation exchange.

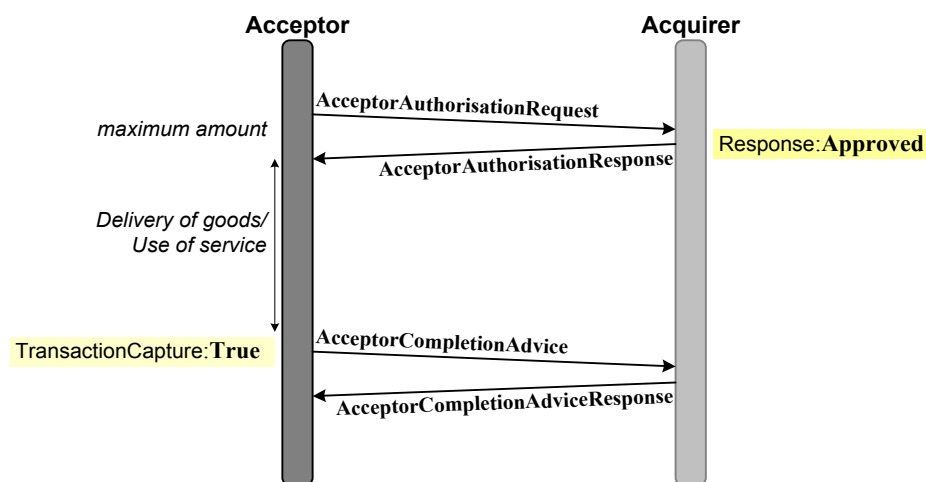


Figure 63: Deferred Payment Captured by Completion

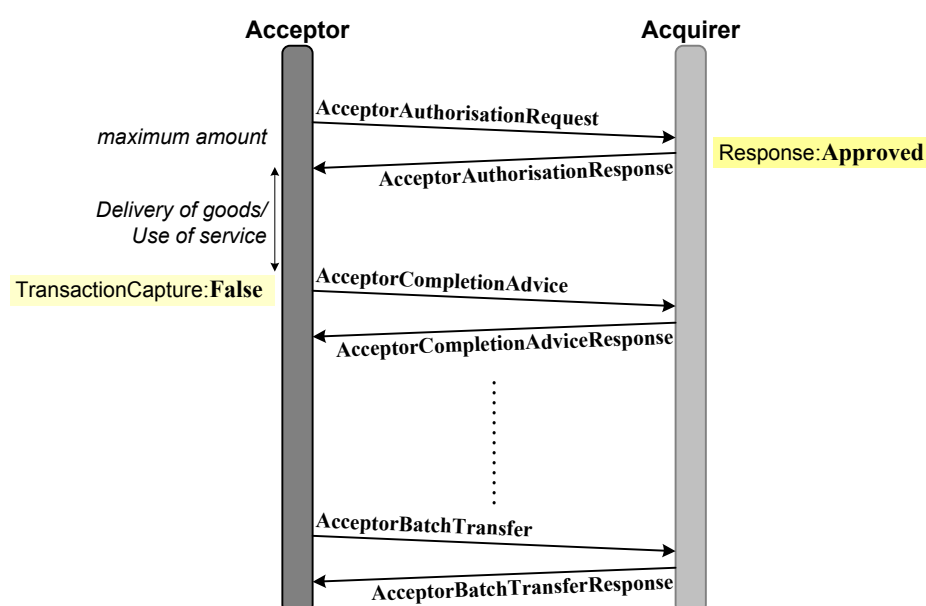


Figure 64: Deferred Payment Captured by Batch

7. After confirmation to the sale system that the goods can be delivered (e.g. after the payment response to the sale system of the Retailer protocol), if the goods or services have not been delivered because the customer did not continue or the delivery failed, an `AcceptorCompletionAdvice` message must be sent with:

- `MessageFunction` = "CompletionAdvice" or "FinancialCompletionAdvice",
- `TotalAmount` with the value "0",
- `Reversal` = "False",
- `TransactionSuccess` = "True",
- `FailureReason` = "CustomerCancel", "Malfunction".

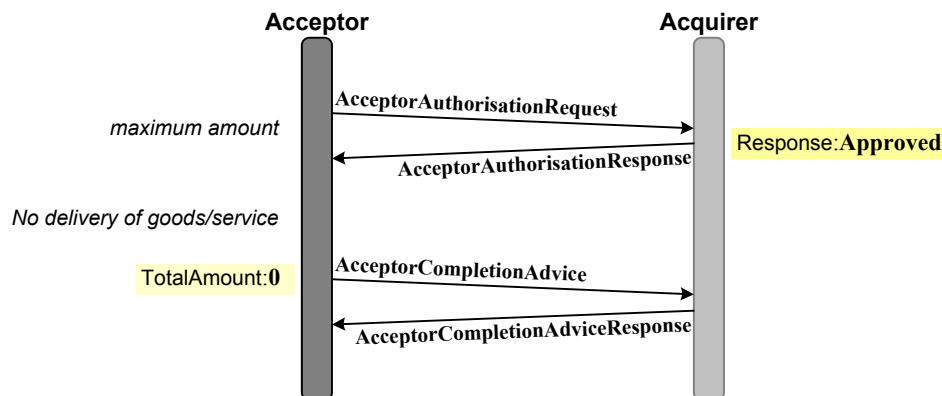


Figure 65: Approved Deferred Payment without Delivery

8. If there is an incident after receiving an approved `AcceptorAuthorisationResponse`, but before confirming to the sale system that the goods can be delivered, an `AcceptorCompletionAdvice` message must be sent with:
- `MessageFunction` = "ReversalAdvice",
 - `TotalAmount` with the maximum authorised amount,
 - `Reversal` = "True",
 - `TransactionSuccess` = "False",
 - `FailureReason` = "CardDeclined", "CustomerCancel", "Malfunction" or "UnableToComplete".
9. The following scenarios from the **Table 2 - List of Payment Cases** page **Erreur ! Signet non défini.** do not apply to the `DeferredPayment` type of transaction:
- All the offline authorisation cases are irrelevant (cases 47 to 63)
 - All the cases where financial capture is performed during the authorisation (cases 1, 2, 6, 9, 12, 13, 18, 22, 23, 28, 32, 35, 38, 41 and 44),
 - The successful transaction cases where the completion is not required (cases 4, 20, and 30).

6.3 Cashback

This feature allows the cardholder to obtain cash from the Card Acceptor in conjunction with a payment. The cardholder receives the extra cash amount in notes or coins along with the goods/services.

It is an additional feature on a payment and does not change message exchanges.

A cashback transaction must only permit the dispensing of cash on successful authorisation or successful referral.

A cashback transaction must obey the following rules :

1. The cardholder must be present at the POI (*PaymentContext.CardholderPresent* = "True")
2. Cashback is requested in the *AcceptorAuthorisationRequest* (*Transaction.TransactionType* = "CashBack")
3. An instance of *TransactionDetails.DetailedAmount* must be included in *AuthorisationRequest* and *AuthorisationResponse* messages:
 - *Type*="CashBack"
 - *Value*=(cashback amount)
4. If *Response* = "Approved" or "PartialApproved", the authorised cashback amount (which may be 0) is returned in the *AcceptorAuthorisationResponse* message.
5. Any cashback given to the cardholder must be included in the *AcceptorCompletionAdvice* and *AcceptorBatchTransfer* messages. If no cashback is delivered, a value of 0 is used.

6.4 Cash Advance

This is a service which allows the card acceptor to give cash to the cardholder without any other goods or services.

It is a special case of a payment and does not change message exchanges.

A cash advance transaction must only permit the dispensing of cash on successful authorisation or successful referral.

Normally a cash advance would be a face-to-face transaction between the card acceptor and cardholder but with the rise of internet-based currency exchanges, this protocol does not limit the cash advance service to this environment.

A cash advance transaction must obey the following rules :

1. Cash advance is requested in the *AcceptorAuthorisationRequest* with *TransactionType*= "CashAdvance".
2. The amount of cash advance transaction (*TransactionDetails.TotalAmount*) is the amount of cash being distributed.
3. The Authorisation is always online (*PaymentContext.OnLineContext*="True").

6.5 Gratuity

This feature allows adding a supplementary amount to the payment of goods/services.

It is an additional service on a payment and does not change message exchanges.

This protocol does not address any business rules regarding when in the payment process a gratuity may be added to the transaction nor when a gratuity may need additional authorisations.

This protocol shows two different methods for implementing gratuity :

1. Gratuity is added before authorisation
2. Gratuity is added after authorisation

The cardholder must be present at the POI, *PaymentContext.CardholderPresent*="True".

Whenever a gratuity amount is included in a message exchange it must be clearly identified in an instance of *TransactionDetails.DetailedAmount* as follows :

- *Type*="Gratuity"
- *Value*=(gratuity amount)

Gratuity added before authorisation

1. A card payment is requested in the *AcceptorAuthorisationRequest*
 - *Transaction.AdditionalService*="Gratuity"
2. The total amount, *TransactionDetails.TotalAmount*, of a gratuity transaction (*TransactionDetails.TotalAmount*) in the *AcceptorAuthorisationRequest* is the sum amount of purchase amount and the gratuity amount.
3. If *Response* = "Approved" or "PartialApproved", the authorised gratuity amount (which may be 0) is returned in the *AcceptorAuthorisationResponse* message.

Gratuity added after authorisation

Gratuity must only be present in the *AcceptorCompletionAdvice* or *AcceptorBatchTransfer*.

Any gratuity must be included in the *AcceptorCompletionAdvice* and *AcceptorBatchTransfer* messages related to the transaction and *TotalAmount* must contain the sum amount of purchase amount and the gratuity amount.

6.6 Reservation

6.6.1 Introduction

A reservation service is mostly used in hotels and for rental business (car, video ...). The reservation service is implemented through one mandatory and four optional steps:

Mandatory step for standard case :

- Initial Reservation

Optional steps

- Update Reservation

- Cancellation of the Reservation (before payment)
- Payment after Reservation, it may be a No-show Payment
- Additional Payment after Reservation

A No-Show payment can be performed without prior reservation.

6.6.2 Reservation steps

6.6.2.1 Initial Reservation

The “Initial Reservation” allows the card acceptor to reserve an amount for a period of time in order to secure that sufficient fund is available to complete a subsequent payment.

6.6.2.2 Update Reservation

The “Update Reservation” allows the card acceptor to change the amount and/or the specified period of a Reservation.

6.6.2.3 Cancellation of the Reservation (before payment)

If the Cancellation of the reservation occurs before the payment is performed; then it allows the card acceptor to cancel a previously approved reservation.

6.6.2.4 Payment for the Reservation

The Payment for the Reservation allows the card acceptor to make a payment using a previously approved initial or updated reservation.

The amount may be zero.

6.6.2.5 No-Show Payment for the Reservation

The “No-Show Payment for the Reservation” allows the card acceptor to charge the cardholder for an agreed amount because the cardholder did not show up as expected.

6.6.2.6 Additional Payment (after Payment for the Reservation)

The “Additional Payment” allows the card acceptor to charge the cardholder for additional expenses after the “Payment for the Reservation” has been done.

If the cardholder is absent during this phase, stored card data will be used which implies that the card security code is not present. This transaction has to be identified as Additional Payment linked to the reservation.

6.6.2.7 No-show payment without performing a prior reservation service

A no-show transaction can be performed although no prior reservation transaction was carried out.

6.6.3 Message flows

Message flows regarding the Authorisation Request, Authorisation Response, Completion Advice and Completion Response are the same as for any other service and are therefore not specifically documented here.

6.6.4 Description of the steps

6.6.4.1 Initial reservation

The initial reservation transaction always processes an online Authorisation to reserve an amount for a specific period for a later payment of goods or services. Financial Capture with the authorisation is not applicable.

If the authorisation request is approved by the card acceptor:

- Depending on the protocol configuration a successful completion can be sent
- The possible subsequent steps are:
 - The reservation can be cancelled
 - The reservation can be updated
 - A payment after reservation can be done
 - Or a No-Show payment after reservation can be done

If the authorisation request is declined:

- The reservation process is ended and, based on configuration, a completion may be sent.

A completion for an unsuccessful transaction must be sent under the following conditions:

- An approved authorisation is not correctly completed at card acceptor side
- There is no response to the authorization request
- The response of the authorization request is received too late:

Then the reservation process is ended

6.6.4.2 Update reservation

The update reservation transaction always processes an online Authorisation to change the previous reservation amount and/or the specified period. Financial Capture with the authorisation is not applicable

If the authorisation request is approved and correctly processed by the card acceptor:

- Depending on the configuration a successful completion can be sent
- The possible subsequent steps are:
 - The reservation can be cancelled
 - A new update reservation can be done
 - A payment after reservation or a No-Show payment after reservation can be done

If the authorisation request is declined:

- The previous reservation remains unchanged.

If an approved update authorisation transaction is not correctly completed or if there is no response or if the response is received too late :

- An unsuccessful completion must be sent.
- The previous reservation remains unchanged.

6.6.4.3 Cancellation of the reservation (before payment)

A reservation can be cancelled before the payment takes place. A reversal (completion) is sent on line to reverse the reservation. The reservation process is ended.

6.6.4.4 Payment after reservation

A payment after reservation is carried out through a completion advice or batch transfer.

A Payment After Reservation can be a No-Show payment

The reservation process is ended, nevertheless, the following subsequent steps are possible:

- An additional payment can be done.
- The payment for the reservation can be cancelled.

6.6.5 Reservation transaction data

Message structures regarding the Authorisation Request, Authorisation response, Completion Advice and Completion Response are the same as for a normal payment and only the components with specific reservation tags or values are represented in these tables.

6.6.5.1 Initial reservation

6.6.5.1.1 AcceptorAuthorisationRequest

<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		It conveys information related to the protocol management on a segment of the path from the Acceptor to the Acquirer:
MessageFunction	[1..1]		The only valid codes to request an authorisation for an initial reservation is: <i>AuthorisationRequest</i> : Request without financial capture (<i>TransactionCapture</i> ="False") (if an invalid value is received, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to "ParsingError")
...			
AuthorisationRequest	[1..1]		The <i>Header.MessageFunction</i> must be "AuthorisationRequest". (In case of an invalid value, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to "ParsingError")
...			
Transaction	[1..1]		
TransactionCapture	[1..1]		<i>TransactionCapture</i> = "False"
TransactionType	[1..1]		" Reservation "
...			
ServiceAttribute	[0..1]		<i>Mandatory. Equal to "InitialReservation"</i>
...			
SaleReferenceIdentification	[0..1]		Mandatory for reservation
...			

<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
OriginalTransaction	[0..1]	Appli	Not used if <i>TransactionType</i> ="Reservation" and <i>ServiceAttribute</i> ="InitialReservation"
...			
TransactionDetails	[1..1]		
...			
AmountQualifier	[0..1]		"Estimated"
...			
ValidityDate	[0..1]	Config	The date after which the reservation expires
...			

6.6.5.1.2 *AcceptorAuthorisationResponse*

<i>AcceptorAuthorisationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		see <i>AcceptorAuthorisationRequest</i>
MessageFunction	[1..1]		The only valid codes to request an authorisation for an initial reservation is: <i>AuthorisationResponse</i> :
...			
AuthorisationResponse	[1..1]		The Header.MessageFunction must be <i>AuthorisationResponse</i>
...			
Transaction	[1..1]		
...			
TransactionDetails	[1..1]		
...			
ValidityDate	[0..1]	Config	Depending of the scheme rules, the value of validity date in the authorisation response may differ from the value contained in the authorisation request
...			

6.6.5.1.3 *AcceptorCompletionAdvice*

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes to advice an initial reservation are : <i>CompletionAdvice</i> : Completion without financial capture (TransactionCapture=False, Reversal=False) or (TransactionCapture=False, Reversal=True, TransactionSuccess=True) <i>ReversalAdvice</i> : Reversal of an authorisation without financial capture (TransactionCapture=False, Reversal=True, TransactionSuccess=False) in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>
...			
CompletionAdvice	[1..1]		The Header.MessageFunction must be <i>CompletionAdvice</i> , or <i>ReversalAdvice</i> (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			
Transaction	[1..1]	CCopy	
TransactionCapture	[0..1]	Config	<i>False or is absent</i>
TransactionType	[1..1]		<i>"Reservation"</i>
ServiceAttribute	[0..1]		<i>"InitialReservation"</i>
...			
SaleReferenceIdentification	[0..1]		Mandatory for reservation
...			
OriginalTransaction	[0..1]	Appli	Not used if <i>TransactionType</i> ="Reservation and <i>ServiceAttribute</i> ="InitialReservation"
...			
MerchantOverride	[0..1]	Appli	<i>False or is absent</i>
FailureReason	[0..*]	Appli	<i>CardDeclined, CustomerCancel, UnableToComplete, TimeOut</i>
...			
TransactionDetails	[1..1]		
...			
AmountQualifier	[0..1]	Appli	<i>"Estimated"</i>

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
...			
ValidityDate	[0..1]	Config	
...			

6.6.5.1.4 AcceptorCompletionAdviceResponse

<i>AcceptorCompletionAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes are: <i>CompletionAdviceResponse</i> : response for <i>CompletionAdvice</i> <i>ReversalAdviceResponse</i> : response for <i>ReversalAdvice</i> (in case of an invalid value, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to <i>ParsingError</i>)
...			

6.6.5.2 Update reservation

6.6.5.2.1 AcceptorAuthorisationRequest

<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		It conveys information related to the protocol management on a segment of the path from the Acceptor to the Acquirer:
MessageFunction	[1..1]		The only valid codes to request an authorisation for an update reservation is: <i>AuthorisationRequest</i> : Request without financial capture (<i>TransactionCapture</i> ="False") (if an invalid value is received, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to "ParsingError")
...			
AuthorisationRequest	[1..1]		The <i>Header.MessageFunction</i> must be "AuthorisationRequest". (In case of an invalid value, a Reject message is sent by the Recipient with <i>RejectReason</i> equal to "ParsingError")
...			
Transaction	[1..1]		
TransactionCapture	[1..1]		<i>TransactionCapture</i> = "False"
TransactionType	[1..1]		" Reservation "
...			
ServiceAttribute	[0..1]		"UpdateReservation"
...			
SaleReferenceIdentification	[0..1]		Must be the same value as the initial value
...			
OriginalTransaction	[0..1]	Appli	Mandatory if available. Identify the previous transaction within the reservation process
...			
TransactionType	[1..1]		"Reservation"
...			
ServiceAttribute	[0..1]		"InitialReservation" or "UpdateReservation"
...			
TransactionDetails	[1..1]		
...			
AmountQualifier	[0..1]		"Estimated" – The estimated amount could be above or below

<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
			"Incremental" - Incremental amount for reservation "Decremental" – Decremental amount for reservation
			...
ValidityDate	[0..1]	Config	
			...

6.6.5.2.2 AcceptorAuthorisationResponse

<i>AcceptorAuthorisationResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		see <i>AcceptorAuthorisationRequest</i>
MessageFunction	[1..1]		The only valid codes to request an authorisation for an update reservation is: <i>AuthorisationResponse</i> :
			...
AuthorisationResponse	[1..1]		The Header.MessageFunction must be <i>AuthorisationResponse</i>
			...
Transaction	[1..1]		
			...
TransactionDetails	[1..1]		
			...
ValidityDate	[0..1]	Config	Depending of the scheme rules, the value of validity date in the authorisation response may differ from the value contained in the authorisation request
			...

6.6.5.2.3 AcceptorCompletionAdvice

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes to advice an update reservation are : <i>CompletionAdvice</i> : Completion without financial capture (TransactionCapture=False, Reversal=False) or (TransactionCapture=False, Reversal=True, TransactionSuccess=True) <i>ReversalAdvice</i> : Reversal of an authorisation without financial capture (TransactionCapture=False, Reversal=True, TransactionSuccess=False) in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>
			...
CompletionAdvice	[1..1]		The Header.MessageFunction must be <i>CompletionAdvice</i> , or <i>ReversalAdvice</i> (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
			...
Transaction	[1..1]	CCopy	
TransactionCapture	[0..1]	Config	False or is absent
TransactionType	[1..1]		"Reservation"
ServiceAttribute	[0..1]		"UpdateReservation"
			...
SaleReferenceIdentification	[0..1]		Mandatory for reservation
			...

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
OriginalTransaction	[0..1]	Appli	Mandatory, Identify the previous transaction within the reservation process
...			
TransactionType	[1..1]		"Reservation"
...			
ServiceAttribute	[0..1]		"InitialReservation" or "updateReservation"
...			
MerchantOverride	[0..1]	Appli	False or is absent
FailureReason	[0..*]	Appli	"CardDeclined, " CustomerCancel, UnableToComplete, TimeOut
...			
TransactionDetails	[1..1]		
...			
AmountQualifier	[0..1]	Appli	"Estimated"
...			
ValidityDate	[0..1]	Config	
...			

6.6.5.2.4 AcceptorCompletionAdviceResponse

<i>AcceptorCompletionAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes are: <i>CompletionAdviceResponse</i> : response for <i>CompletionAdvice</i> <i>ReversalAdviceResponse</i> : response for <i>ReversalAdvice</i> (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			

6.6.5.3 Cancellation of a Reservation before payment

6.6.5.3.1 AcceptorCompletionAdvice

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes to advice an initial reservation is : <i>ReversalAdvice</i> : Reversal of an authorisation without financial capture (TransactionCapture=False, Reversal=True, in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>
...			
CompletionAdvice	[1..1]		The Header.MessageFunction must be <i>CompletionAdvice</i> , or <i>ReversalAdvice</i> (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			
Transaction	[1..1]	CCopy	
TransactionCapture	[0..1]	Config	False or is absent
TransactionType	[1..1]		"Reservation"
ServiceAttribute	[0..1]		"InitialReservation" or "UpdateReservation"
...			

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
SaleReferenceIdentification	[0..1]		Must be the same value as the initial reservation
...			
OriginalTransaction	[0..1]	Appli	Must be present if available
...			
TransactionSuccess	[1..1]		<i>True</i>
Reversal	[0..1]	Appli	<i>True</i>
MerchantOverride	[0..1]	Appli	<i>False or is absent</i>
FailureReason	[0..*]	Appli	<i>CustomerCancel, Malfunction</i>
...			
TransactionDetails	[1..1]		
...			
TotalAmount	[1..1]	Appli	Must be equal to zero as specified for normal payment: "equal to Final amount of the payment transaction when TransactionSuccess is "True"".
AmountQualifier	[0..1]	Appli	<i>Actual or is absent</i>
...			
AuthorisedAmount	[0..1]	Appli	Must be present as specified for normal payment. "Amount authorised for the payment transaction, , mandatory if the authorized amount (TotalAmount of the AcceptorAuthorisationResponse) is different from the requested amount (TotalAmount of the AcceptorAuthorisationRequest)"
...			
ValidityDate	[0..1]		<i>Mandatory for reservation</i>
...			

6.6.5.3.2 AcceptorCompletionAdviceResponse

<i>AcceptorCompletionAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes is: <i>ReversalAdviceResponse</i> : response for <i>ReversalAdvice</i> (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			

6.6.5.4 Payment after reservation

6.6.5.4.1 AcceptorCompletionAdvice

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes to advice a payment after reservation is: <i>FinancialCompletionAdvice</i> : Completion with financial capture (TransactionCapture= <i>True</i> , Reversal= <i>False</i>) (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			
CompletionAdvice	[1..1]		The Header.MessageFunction must be <i>FinancialCompletionAdvice</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			
Transaction	[1..1]	CCopy	
TransactionCapture	[0..1]	Config	<i>True</i>

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
TransactionType	[1..1]		"Reservation"
AdditionalService	[0..*]		"NoShow" is possible for a payment after reservation
ServiceAttribute	[0..1]		"PaymentReservation"
			...
SaleReferenceIdentification	[0..1]		Must be the same value as the initial reservation
			...
Reversal	[0..1]		False or is absent"
			...
OriginalTransaction	[0..1]	Appli	Mandatory, Identify the previous transaction within the reservation process
			...
TransactionType	[1..1]		"Reservation"
ServiceAttribute	[0..1]		"InitialReservation" or "updateReservation"
			...
TransactionDetails	[1..1]		
			...
AmountQualifier	[0..1]	Appli	"Actual" or is absent
			...
ValidityDate	[0..1]	Config	
			...

6.6.5.4.2 AcceptorCompletionAdviceResponse

<i>AcceptorCompletionAdviceResponse</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes are: <i>FinancialCompletionAdviceResponse</i> : response for <i>CompletionAdvice</i> (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
			...

6.6.5.4.3 AcceptorCancellationRequest

<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
			...
CancellationRequest	[1..1]		The Header.MessageFunction must be <i>CancellationRequest</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to "ParsingError")
			...
Transaction	[1..1]		
			...
SaleReferenceIdentification	[0..1]		Mandatory for reservation
			...
TransactionDetails	[1..1]		
			...
ValidityDate	[0..1]	Config	
			...

6.6.5.4.4 AcceptorCancellationResponse

Refer to Normal Payment

6.6.5.4.5 AcceptorCancellationAdvice

<i>AcceptorCancellationAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
CancellationAdvice	[1..1]		The Header.MessageFunction must be <i>CancellationAdvice</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
Transaction	[1..1]		
SaleReferenceIdentification	[0..1]		Mandatory for reservation
TransactionDetails	[1..1]		
ValidityDate	[0..1]	Config	
...			

6.6.5.4.6 AcceptorCancellationAdviceResponse

Refer to Normal Payment

6.6.5.5 Additional payment

6.6.5.5.1 AcceptorAuthorisationRequest

<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		It conveys information related to the protocol management on a segment of the path from the Acceptor to the Acquirer:
MessageFunction	[1..1]		The only valid codes to request an authorisation for payment after reservation are: <i>AuthorisationRequest</i> : Request without financial capture (TransactionCapture="False") <i>FinancialAuthorisationRequest</i> : Request with financial capture TransactionCapture="True") (if an invalid value is received, a Reject message is sent by the Recipient with RejectReason equal to "ParsingError")
...			
AuthorisationRequest	[1..1]		The <i>Header.MessageFunction</i> must be "AuthorisationRequest" or "FinancialAuthorisationRequest". (In case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to "ParsingError")
...			
Transaction	[1..1]		
...			
TransactionType	[1..1]		"Reservation"
...			
ServiceAttribute	[0..1]		"AdditionalPayment"
...			
SaleReferenceIdentification	[0..1]		Must be the same value as the initial reservation
...			
OriginalTransaction	[0..1]	Appli	Not used if <i>TransactionType</i> ="Reservation and <i>ServiceAttribute</i> ="AdditionalPayment"

<i>AcceptorAuthorisationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
			...
TransactionDetails	[1..1]		
			...
AmountQualifier	[0..1]		"Actual"
			...
ValidityDate	[0..1]	Config	
			...

6.6.5.5.2 AcceptorauthorisationResponse

Refer to Normal Payment

6.6.5.5.3 AcceptorCompletionAdvice

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		<p>The only valid codes to advise a payment after reservation are:</p> <p><i>CompletionAdvice</i>: Completion without financial capture (TransactionCapture=<i>False</i>, Reversal=<i>False</i>) or (TransactionCapture=<i>False</i>, Reversal=<i>True</i>, TransactionSuccess=<i>True</i>)</p> <p><i>FinancialCompletionAdvice</i>: Completion with financial capture (TransactionCapture=<i>True</i>, Reversal=<i>False</i>) or (TransactionCapture=<i>True</i>, Reversal=<i>True</i>, TransactionSuccess=<i>True</i>)</p> <p><i>ReversalAdvice</i>: Reversal of an authorisation without financial capture (TransactionCapture=<i>False</i>, Reversal=<i>True</i>, TransactionSuccess=<i>False</i>)</p> <p><i>FinancialReversalAdvice</i>: Reversal of a <i>FinancialAuthorisation</i> (TransactionCapture=<i>True</i>, Reversal=<i>True</i>, TransactionSuccess=<i>False</i>)</p> <p>(in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)</p>
			...
CompletionAdvice	[1..1]		<p>The Header.MessageFunction must be <i>CompletionAdvice</i>, <i>FinancialCompletionAdvice</i>, <i>ReversalAdvice</i> or <i>FinancialReversalAdvice</i>. (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)</p>
			...
Transaction	[1..1]	CCopy	
			...
TransactionType	[1..1]		"Reservation"
			...
ServiceAttribute	[0..1]		"AdditionalPayment"
			...
SaleReferenceIdentification	[0..1]		Must be the same value as the initial reservation
			...
OriginalTransaction	[0..1]	Appli	Not used if TransactionType="Reservation" and ServiceAttribute="AdditionalPayment"
			...
TransactionDetails	[1..1]		
			...

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
ValidityDate	[0..1]	Config	
...			

6.6.5.5.4 *AcceptorCompletionAdviceResponse*

Refer to Normal Payment

6.6.5.5.5 *AcceptorCancellationRequest*

<i>AcceptorCancellationRequest</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
...			
CancellationRequest	[1..1]		The Header.MessageFunction must be <i>CancellationRequest</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to "ParsingError")
...			
Transaction	[1..1]		
...			
SaleReferenceIdentification	[0..1]		Mandatory for reservation
...			
TransactionDetails	[1..1]		
...			
ValidityDate	[0..1]	Config	
...			

6.6.5.5.6 *AcceptorCancellationResponse*

Refer to Normal Payment

6.6.5.5.7 *AcceptorCancellationAdvice*

<i>AcceptorCancellationAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
CancellationAdvice	[1..1]		The Header.MessageFunction must be <i>CancellationAdvice</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
Transaction	[1..1]		
SaleReferenceIdentification	[0..1]		Mandatory for reservation
TransactionDetails	[1..1]		
ValidityDate	[0..1]	Config	
...			

6.6.5.5.8 *AcceptorCancellationAdviceResponse*

Refer to Normal Payment

6.6.5.6 No-show payment without prior reservation

6.6.5.6.1 *AcceptorCompletionAdvice*

<i>AcceptorCompletionAdvice</i>	<i>Mult.</i>	<i>Rule</i>	<i>Usage</i>
Header	[1..1]		
MessageFunction	[1..1]		The only valid codes to advice No-Show payment is :: <i>FinancialCompletionAdvice</i> : Completion with financial capture (TransactionCapture= <i>True</i> , Reversal= <i>False</i>) (in case of an invalid value, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			
CompletionAdvice	[1..1]		The Header.MessageFunction must be <i>FinancialCompletionAdvice</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
...			
Transaction	[1..1]	CCopy	
...			
TransactionType	[1..1]		"CardPayment"
AdditionalService	[1..1]		"NoShow"
...			

6.6.5.6.2 *AcceptorCompletionAdviceResponse*

Refer to Normal Payment

6.6.5.6.3 *AcceptorCancellationRequest*

Refer to Normal Payment

6.6.5.6.4 *AcceptorCancellationResponse*

Refer to Normal Payment

6.6.5.6.5 *AcceptorCancellationAdvice*

Refer to Normal Payment

6.6.5.6.6 *AcceptorCancellationAdviceResponse*

Refer to Normal Payment

6.6.5.7 Batch

6.6.5.7.1 AcceptorBatchTransfer

Or	AcceptorBatchTransfer	Mult.	Rule	Type / Definition / Code List
	Header	[1..1]		
	...			
	BatchTransfer	[1..1]		
	...			
	DataSet	[0..*]		A data set may include both financial (completed transactions) and non financial transactions (uncompleted transactions).
	TransactionTotals	[1..*]		<i>TransactionTotals</i> of the <i>DataSet</i> .
	...			
	Type	[1..1]		see <i>AcceptorReconciliationRequest</i> for <i>Reservation</i>
	...			
	CommonData	[0..1]		Data common to transactions of a DataSet may be factorised in CommonData to reduce the message length. All transactions of the DataSet inherit the data element value present in CommonData except if this data element is present in the occurrence of Transaction.
	...			
	TransactionType	[0..1]		"Reservation"
	AdditionalService	[0..*]		"NoShow" is possible
	ServiceAttribute	[0..1]		"PaymentReservation"
	...			
	Transaction	[1..*]		Transaction of the data set. It must be a completion, a cancellation advice, an authorisation request or an authorisation response.
{Or	Completion	[1..1]		
	...			
	Transaction	[1..1]		
	TransactionType	[0..1]		"Reservation"
	AdditionalService	[0..*]		"NoShow" is possible
	ServiceAttribute	[0..1]		"PaymentReservation"
	...			
	SaleReferenceIdentification	[0..1]	Appli	Mandatory for Reservation
	...			
	TransactionIdentification	[1..1]		see <i>AcceptorCompletionAdvice</i> for <i>Reservation</i> after payment
	...			
	OriginalTransaction	[0..1]	Appli	see <i>AcceptorCompletionAdvice</i> for <i>Reservation</i> after payment
	...			
	TransactionDetails	[1..1]		see <i>AcceptorCompletionAdvice</i> for <i>Reservation</i> after payment
	...			
Or	Cancellation	[1..1]		Cancelled card payment transaction.
	...			
	Transaction	[1..1]		see <i>AcceptorCancellationAdvice</i> for reservation after payment
	...			
Or	AuthorisationRequest	[1..1]		Authorisation request of a card payment transaction.
	...			
	Transaction	[1..1]		see <i>AcceptorAuthorisationRequest</i> for Initial and update reservation
	...			
Or}	AuthorisationResponse	[1..1]		Authorisation response of a card payment transaction.

Or	<i>AcceptorBatchTransfer</i>	Mult.	Rule	Type / Definition / Code List
				...
	Transaction	[1..1]		see <i>AcceptorAuthorisationResponse</i> for Initial and update reservation
				...

6.6.5.8 Reconciliation

6.6.5.8.1 AcceptorReconciliationRequest

<i>AcceptorReconciliationRequest</i>	Mult.	Rule	Usage
			...
ReconciliationRequest	[1..1]		The Header.MessageFunction must be <i>ReconciliationRequest</i> . (if not the case, a Reject message is sent by the Recipient with RejectReason equal to <i>ParsingError</i>)
			...
Transaction	[1..1]		
			...
TransactionTotals	[0..*]		<i>TransactionTotals</i> of the reconciliation period. <i>TransactionTotals</i> is absent if the reconciliation period contains no transactions.
			...
Type	[1..1]		All the values are allowed: <i>Debit</i> : Debit transactions (TransactionType is <i>CardPayment</i> , <i>CashBack</i> , <i>CashAdvance</i> , <i>DeferredPayment</i> , <i>Reservation</i>) during the reconciliation period. <i>DebitReverse</i> : Cancelled debit transactions. <i>Credit</i> : Credit transactions (TransactionType is <i>Refund</i> during the reconciliation period. <i>CreditReverse</i> : Cancelled credit transactions. <i>Declined</i> : transactions declined online or offline. <i>Failed</i> : failed transactions.
			...

6.6.5.8.2 AcceptorReconciliationResponse

Refer to Normal Payment

6.7 Dynamic Currency Conversion (DCC)

6.7.1 Introduction

Some transactions may be handled in foreign currencies which makes difficult for a cardholder to estimate the actual amount he will have to pay in his own currency at the moment of the payment.

A Dynamic Currency Conversion is a service offered by a merchant to allow the cardholder to make the payment in the currency of his card instead of the currency of the merchant when different.

The cardholder can accept or not the DCC service. When DCC is allowed by the merchant, the currency conversion is managed through a DCC service provider proposing to the cardholder the converted amount to be ultimately debited on his card.

If the DCC service is refused by the cardholder, the transaction then proceeds as a normal card payment transaction in the currency of the merchant.

The DCC service may be relying on an agent acting between the merchant and the DCC service provider. The following use cases involve agents in the process, but direct exchanges between a merchant and a DCC service provider or between a merchant and an acquirer do follow the same logic.

The role of the DCC service provider can be played by any party (Agent, Acquirer, etc.).

6.7.2 Currency conversion during the authorisation

In this process, the Dynamic Currency Conversion service is implemented during the authorisation through an *AcceptorAuthorisationRequest* message ("*implicit rate request*").

The Acceptor or Agent asks the DCC service provider whether the card is eligible for a Dynamic Currency Conversion service or not.

The actual authorisation is carried out either in the currency of the card or in the currency of the merchant depending on the outcome of the DCC process and the decision of the cardholder.

The conversion service is carried out along the following steps:

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to an Agent or an Acquirer.
2. The Agent or Acquirer detects whether the transaction is eligible for DCC and sends an *AcceptorCurrencyConversionRequest* message to the DCC service provider to ensure the eligibility of the card and to get the conversion rate for the transaction in case of eligibility.
3. The DCC Provider answers with an *AcceptorCurrencyConversionResponse* message with the requested information.
4. The Agent or Acquirer sends back an *AcceptorAuthorisationResponse* message to the Acceptor with the proposed currency conversion data when eligibility has been confirmed.
5. The Cardholder accepts the proposed currency conversion.
6. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent or Acquirer with the accepted converted amount and all relevant information related to the conversion process.
7. The Agent or Acquirer forwards the *AcceptorAuthorisationRequest* for authorisation.
8. The next steps follow a normal transaction flow (i.e. Completion capture, batch capture, etc.) with some additional information related to the conversion process.
9. If the DCC service was accepted by the cardholder, the Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* that the DCC transaction was completed.
10. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* about the completion of the transaction on his side.

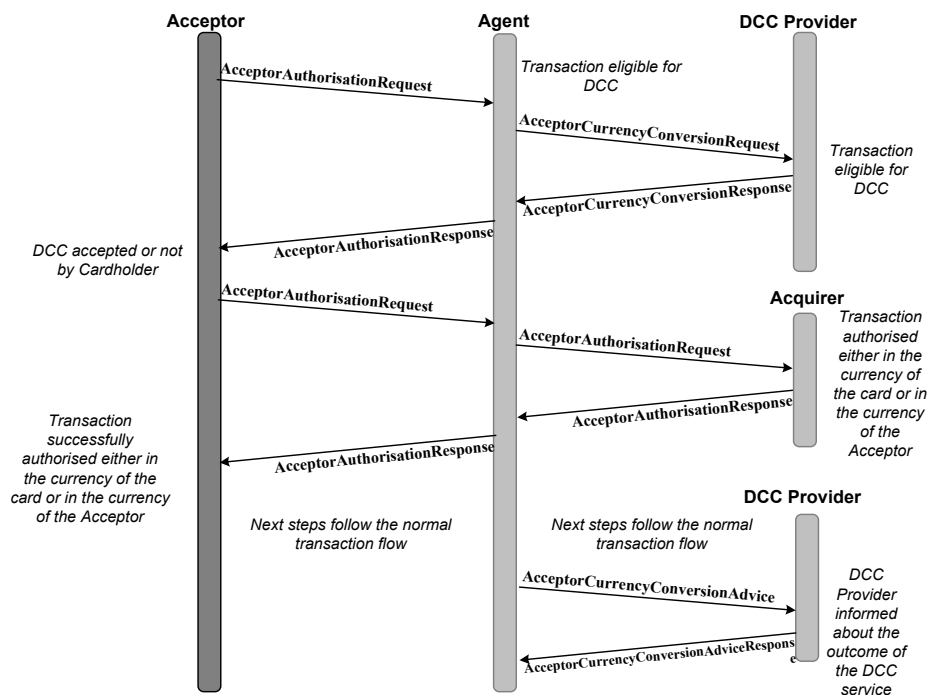


Figure 66: Currency conversion during the authorisation

6.7.2.1 Transaction eligible for DCC and DCC accepted by the Cardholder

In this scenario, the transaction is considered eligible for DCC by the Agent and the DCC service provider. The Cardholder accepts the DCC service for the amount in the currency of his card presented to him. The transaction proceeds as a normal card payment transaction in the currency of the card.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to an Agent. The message contains the following information :
 - CurrencyConversion: absent.
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment.
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The Agent sends an *AcceptorCurrencyConversionRequest* to the DCC service provider with the following information :
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture: not relevant.
 - TransactionType : copy from *AcceptorAuthorisationRequest*.
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message with the following information:
 - Result of currency conversion: Allowed.
 - Conversion details.
4. The Agent sends back an *AcceptorAuthorisationResponse* to the Acceptor with the proposed currency conversion data and the response code "*Declined*" with the action type "*AcceptCurrencyConversion*".
5. The Cardholder accepts the proposed currency conversion.
6. The Acceptor sends an *AcceptorAuthorisationRequest* with the following data:
 - TransactionIdentification: copy of the original *AcceptorAuthorisationRequest* message
 - TransactionType: copy of the original *AcceptorAuthorisationRequest* message.
 - TransactionDetail.Currency: copy of Conversion.TargetCurrency of the original *AcceptorAuthorisationResponse* message.
 - TransactionDetail.TotalAmount: copy of the Conversion.ResultingAmount of original *AcceptorAuthorisationResponse* message.
 - AdditionalService: copy of original *AcceptorAuthorisationRequest* plus DCC.
 - AcceptedByCardholder: True.
 - CurrencyConversion: copy of *AcceptorCurrencyConversionResponse* message.
7. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer.
8. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.) with some additional information related to the conversion process.
9. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* that the DCC transaction was completed.
10. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* about the completion of the transaction on his side.

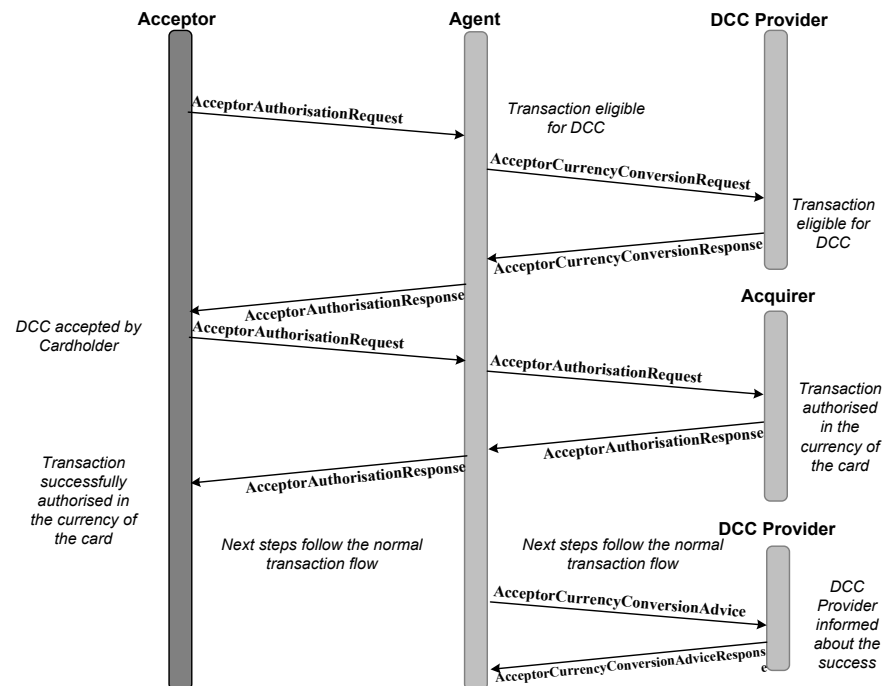


Figure 67: Transaction eligible for DCC and DCC accepted by the Cardholder

6.7.2.2 Transaction eligible for DCC and partially approved by the Issuer

In this scenario, the transaction is considered eligible for DCC by the Agent and the DCC service provider. It is also assumed that the DCC service provider accept to deliver a catalogue of a range of amounts for which DCC is offered to the cardholder which would permit partial authorisation by the Issuer. The Cardholder accepts the DCC service for the amount in the currency of his card presented to him. The transaction is authorised by the Issuer but for a reduced amount in the currency of his card (*PartialAuthorisation*). The Cardholder accepts the transaction for this reduced amount. The transaction proceeds as a normal card payment transaction in the currency of the card and for the reduced authorised amount.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to an Agent. The message contains the following information :
 - CurrencyConversion: absent.
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment.
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The Agent sends an *AcceptorCurrencyConversionRequest* to the DCC service provider with the following information :
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture: not relevant.
 - TransactionType : copy from *AcceptorAuthorisationRequest*.
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message with the following information:
 - CurrencyConversionResult.Result: CATG (Conversion accepted for a range of amounts - rate and fees specific per range of amounts)
 - Conversion details per range of amounts provided.
4. The Agent sends back an *AcceptorAuthorisationResponse* to the Acceptor with the proposed currency conversion data ranges and the response code "*Declined*" with the action type "*AcceptCurrencyConversion*".
5. The Cardholder accepts the proposed currency conversion.
6. The Acceptor sends an *AcceptorAuthorisationRequest* with the following data:
 - TransactionIdentification: copy of the original *AcceptorAuthorisationRequest* message
 - TransactionType: copy of the original *AcceptorAuthorisationRequest* message.
 - TransactionDetail.Currency: copy of Conversion.TargetCurrency of the original *AcceptorAuthorisationResponse* message.
 - TransactionDetail.TotalAmount: copy of the Conversion.ResultingAmount of original *AcceptorAuthorisationResponse* message.
 - AdditionalService: copy of original *AcceptorAuthorisationRequest* plus DCC.
 - AcceptedByCardholder: True.
 - CurrencyConversion: copy of *AcceptorCurrencyConversionResponse* message
7. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer.
8. The Issuer authorised the transaction but for a reduced amount. The Acquirer sends back an *AcceptorAuthorisationResponse* message to the Agent. The Agents computes the actual amount to be presented to the cardholder (after computation of fees and taking into account the reduced authorised amount by the Issuer) and forwards the amended message to the Acceptor. The Acceptor presents to the cardholder the reduced authorised amount after the withdrawal of fees and in the currency of his card.

9. The Cardholder accepts the DCC transaction for the proposed reduced authorised amount in the currency of his card.
10. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent for the reduced amount which is then forwards to the Acquirer.
11. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.) with some additional information related to the conversion process.
12. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* that the DCC transaction was completed but for a reduced amount. The details of the reduced amount accepted by the cardholder are provided to the DCC service provider.
13. The DCC service provider responds with an *AcceptorCurrencyConversionAdviceResponse* about the completion of the transaction on his side.

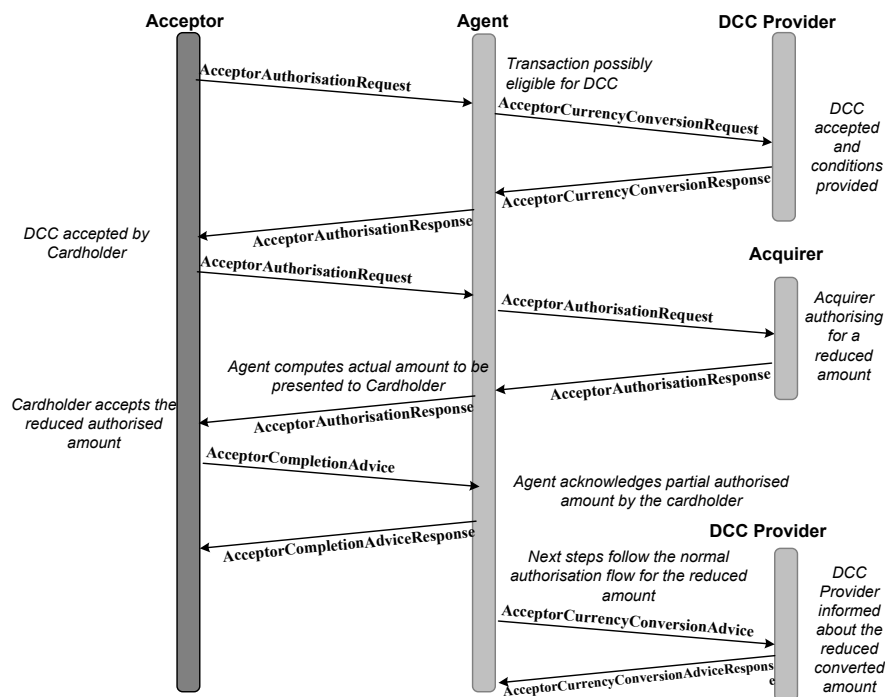


Figure 68: Transaction eligible for DCC and partially approved by the Issuer

6.7.2.3 Transaction not eligible for DCC by the DCC service provider

In this scenario, the transaction is considered eligible for DCC by the Agent but not by the DCC service provider. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent with the following information.
 - CurrencyConversion : absent.
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment.
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The Agent sends an *AcceptorCurrencyConversionRequest* message to the DCC Provider with the following information.
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture: not relevant.
 - TransactionType : copy from AcceptorAuthorisationRequest.
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message with the following information:
 - Result: InvalidCard, NoRate, InvalidMerchant, NotAvailable or InvalidProduct.
 - Conversiondetails: absent.
4. The Agent sends an *AcceptorAuthorisationRequest* message to the Acquirer in the currency of the Acceptor.
5. The Acquirer sends back an *AcceptorAuthorisationResponse* to the Agent.
6. The Agent forwards the *AcceptorAuthorisationResponse* to the Acceptor.
7. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).

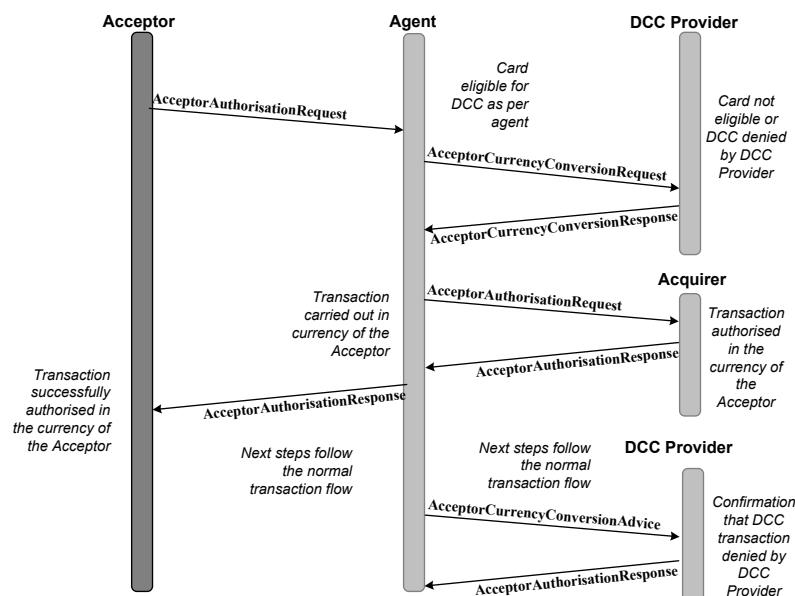


Figure 69: Transaction not eligible for DCC by DCC service provider

6.7.2.4 Transaction not eligible for DCC by the Agent

In this scenario, the transaction is considered as not eligible for DCC by the Agent. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent. The message should have the following information.
 - CurrencyConversion : absent.
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment.
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency
2. The Agent detects that the card is not eligible for DCC (the Acceptor doesn't have a contract with a DCC service provider or the line to the DCC service provider is not operational).
3. The Agent sends an *AcceptorAuthorisationRequest* message to the Acquirer in the currency of the Acceptor.
4. The Acquirer sends back an *AcceptorAuthorisationResponse* message to the Agent.
5. The Agent forwards the *AcceptorAuthorisationResponse* message to the Acceptor.
6. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).

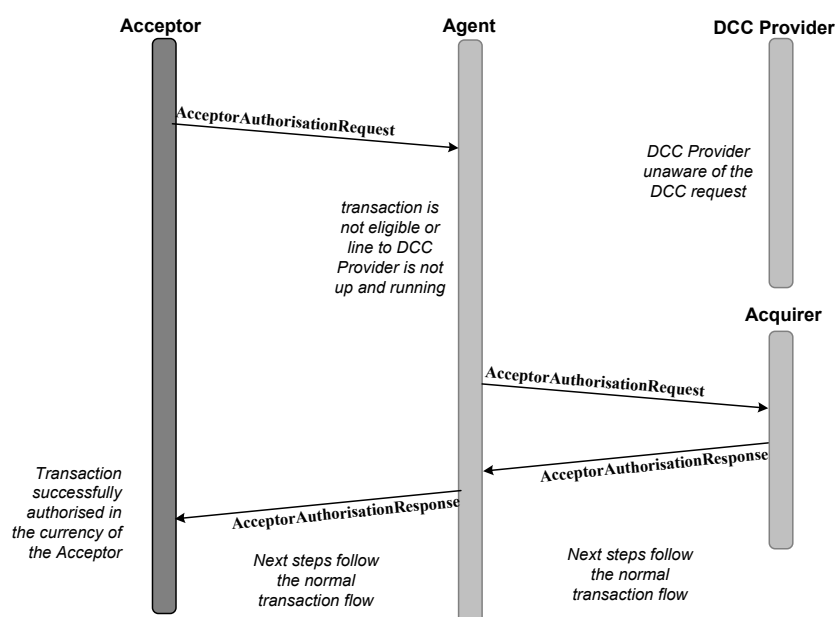


Figure 70: Transaction not eligible for DCC by Agent

6.7.2.5 Transaction eligible for DCC but DCC refused by the Cardholder

In this scenario, the transaction is considered eligible for DCC by the Agent and the DCC service provider. The cardholder refused the DCC service for the amount presented to him in the currency of his card. The transaction proceeds as a normal card payment transaction in the currency of the merchant.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent with the DCC relevant data.
2. The Agent sends an *AcceptorCurrencyConversionRequest* message to the DCC service provider with the DCC relevant data.
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message containing the following information:
 - Result of the currency conversion: Allowed.
 - Conversiondetails: Present
4. The Agent sends back an *AcceptorAuthorisationResponse* message to the Acceptor with the proposed currency conversion data and the response “Declined” with the action type “AcceptCurrencyConversion”.
5. The Cardholder refuses the DCC service.
6. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent in the currency of the Acceptor.
 - TransactionIdentification: copy of original *AcceptorAuthorisationRequest* message
 - TransactionType : copy of original *AcceptorAuthorisationRequest* message.
 - TransactionDetail.Currency: copy of original *AcceptorAuthorisationRequest* message.
 - TransactionDetail.TotalAmount: copy of original *AcceptorAuthorisationRequest* message.
 - AdditionalService: copy of original *AcceptorAuthorisationRequest* plus DCC information.
 - AcceptedByCardholder: False.
 - CurrencyConversion: copy of *AcceptorCurrencyConversionResponse* message.
7. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the Acceptor’s currency.
8. The Acquirer sends back an *AcceptorAuthorisationResponse* message to the Agent.
9. The Agent forwards the *AcceptorAuthorisationResponse* message to the Acceptor.
10. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).
11. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* message that the DCC transaction was refused by the Cardholder.
12. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* message about the completion of the transaction on his side.

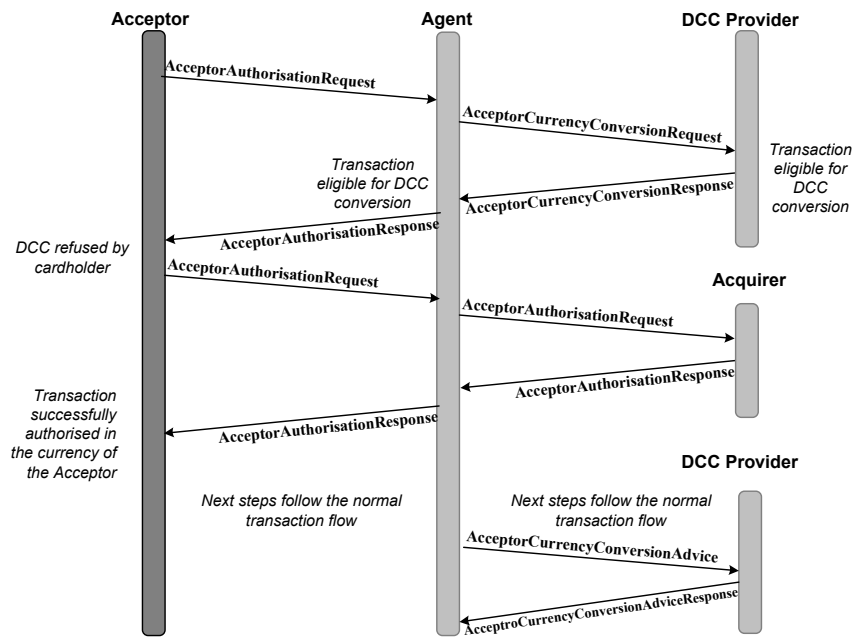


Figure 71: Transaction eligible for DCC but DCC refused by cardholder

6.7.2.6 Transaction eligible for DCC but interrupted due to an incident

6.7.2.6.1 Incident at the Agent level

In this scenario, the transaction is considered eligible for DCC by the Agent but the response didn't reach the Acceptor or an error was encountered in the response. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent with the following information.
 - CurrencyConversion : absent.
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment.
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The AcceptorCurrencyConversionResponse is not received by the Acceptor due to an incident.
3. The Cardholder is not offered the possibility to proceed with DCC.
4. The Agent sends an *AcceptorAuthorisationRequest* message to the Acquirer in the currency of the Acceptor.
5. The Acquirer sends back an *AcceptorAuthorisationResponse* to the Agent.
6. The Agent forwards the *AcceptorAuthorisationResponse* to the Acceptor.
7. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).

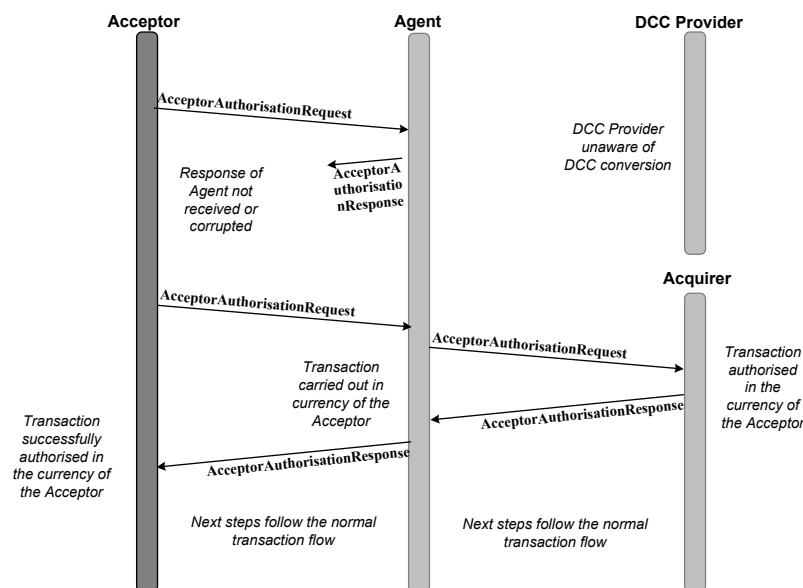


Figure 72: : Incident at Agent level

6.7.2.6.2 Incident at DCC service provider level

In this scenario, the transaction is considered eligible for DCC by the Agent and the DCC service provider but the response sent by the DCC provider didn't reach the Agent or an error was encountered in the response. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant. A negative advice is forwarded to the DCC service provider at the end of the transaction.

1. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent with the following information.
 - CurrencyConversion : absent.
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment.
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The Agent sends an *AcceptorCurrencyConversionRequest* message to the DCC service provider with the following information.
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture: not relevant.
 - TransactionType : copy from AcceptorAuthorisationRequest.
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message to the Agent with all the relevant information.
4. The *AcceptorCurrencyConversionResponse* message is not received by the Agent due to an incident.
5. The Cardholder is not offered the possibility to proceed with DCC.
6. The Agent sends an *AcceptorAuthorisationRequest* message to the Acquirer in the currency of the Acceptor.
7. The Acquirer sends back an *AcceptorAuthorisationResponse* message to the Agent.
8. The Agent forwards the *AcceptorAuthorisationResponse* message to the Acceptor.
9. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).
10. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* message that the DCC transaction did not succeed.
11. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* message about the completion of the transaction on his side.

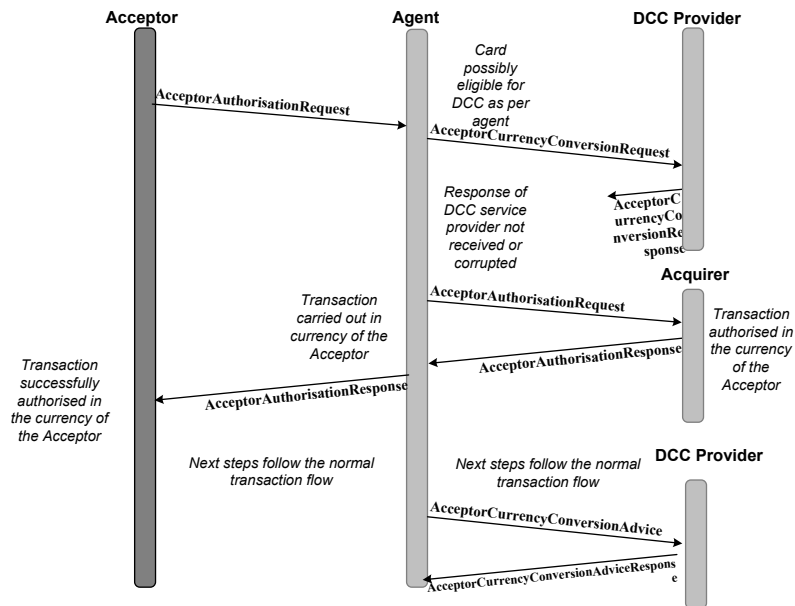


Figure 73: Incident at DCC service provider level

6.7.3 Currency conversion prior to the authorisation

In this process, the Dynamic Currency Conversion service is implemented prior to the authorisation actually taking place and through an *AcceptorCurrencyConversion* exchange (“explicit rate request”)

The Acceptor or Agent asks the DCC service provider whether the card is eligible for a Dynamic Currency Conversion service or not.

The actual authorisation is carried out either in the currency of the card or in the currency of the merchant depending on the outcome of the DCC process and the decision of the cardholder.

The conversion service is carried out along the following steps:

1. The Acceptor sends an *AcceptorCurrencyConversionRequest* message to an Agent or an Acquirer to ask whether the card is eligible for a currency conversion.
2. The Agent or Acquirer detects whether the transaction is eligible for DCC and, if eligible, forwards the *AcceptorCurrencyConversionRequest* message to the DCC service provider to ensure eligibility of the card and to get the conversion rate for the transaction.
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message with the requested information.
4. The Agent or Acquirer forwards the *AcceptorCurrencyConversionResponse* message to the Acceptor with the proposed currency conversion data.
5. The Cardholder accepts the proposed currency conversion.
6. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent or Acquirer with the accepted converted amount and all relevant information related to the conversion process.
7. The Agent or Acquirer forwards the *AcceptorAuthorisationRequest* message for authorisation.
8. The next steps follow a normal transaction flow (i.e. Completion capture, batch capture, etc.) with some additional information related to the currency conversion process.
9. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* message that the DCC transaction was completed.
10. The DCC service provider responds with an *AcceptorCurrencyConversionAdviceResponse* message about the completion of the transaction on his side.

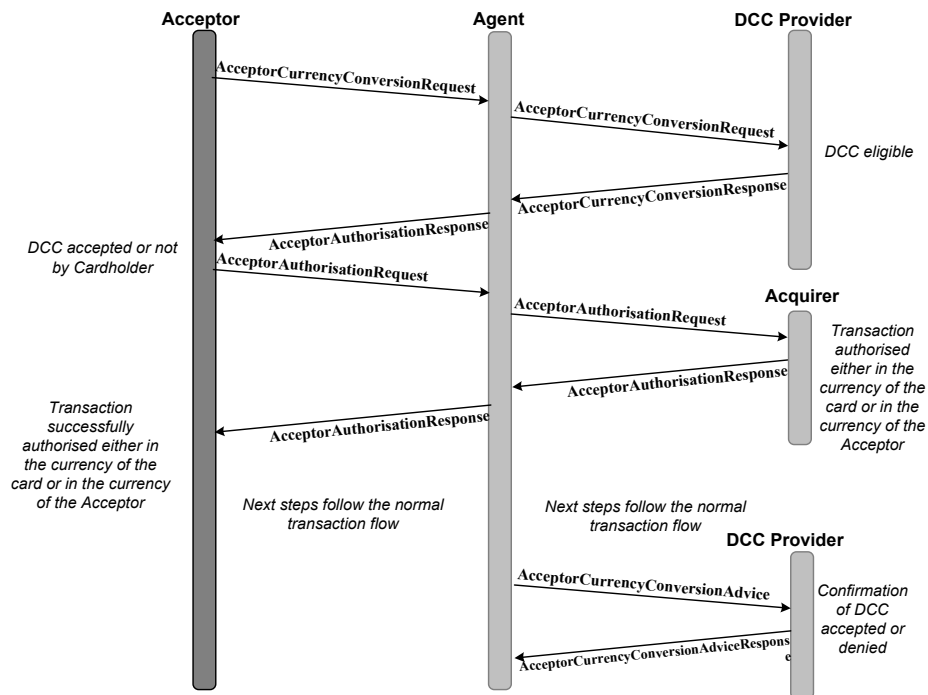


Figure 74: Currency conversion before authorisation

6.7.3.1 Transaction eligible for DCC and DCC accepted by the Cardholder

In this scenario, the transaction is considered eligible for DCC by both the Agent and the DCC service provider. The cardholder accepts the DCC service for the amount presented to him in the currency of his card. The transaction proceeds as a normal card payment transaction in the currency of the card.

1. The acceptor sends an *AcceptorCurrencyConversionRequest* message to the Agent with the following information :
 - *CurrencyConversionResult*: absent
 - *TransactionType* : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - *AdditionalService* : Cashback, Gratuity (if *TransactionType* = CardPayment)
 - *TransactionDetails.TotalAmount*: Total Amount in the Acceptor's currency.
2. The Agent forwards the *AcceptorCurrencyConversionRequest* message with following information to the DCC service provider:
 - *Card.CardCurrencyCode*: Target currency (currency of the card).
 - *TransactionDetails.Currency*: Source currency (currency of the Acceptor).
 - *TransactionCapture*: not relevant.
 - *TransactionType*: copy from the *AcceptorCurrencyConversionRequest* message

3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message containing the following information:
 - Result : Allowed
 - ConversionDetails: Present
4. The Agent forwards the *AcceptorCurrencyConversionResponse* message to the Acceptor with the currency conversion data.
5. The cardholder accepts the proposed currency conversion.
6. The Acceptor sends an *AcceptorAuthorisationRequest* message with the agreed currency conversion data:
 - TransactionType : copy of the *AcceptorCurrencyConversionRequest* message
 - TransactionDetail.Currency: copy of ConversionDetails.TargetCurrency of the *AcceptorCurrencyConversionResponse* message.
 - TransactionDetail.TotalAmount : copy of ConversionDetails.ResultingAmount of the *AcceptorCurrencyConversionResponse* message
 - AdditionalService : copy of the *AcceptorCurrencyConversionRequest* message plus DCC information
 - CurrencyConversionResult.AcceptedByCardholder : True
 - CurrencyConversionResult.ConversionDetails: copy of *AcceptorCurrencyConversionResponse*
7. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the currency of the card.
8. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.) with some additional information related to the conversion process.
9. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* message that the DCC transaction was completed.
10. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* message about the completion of the transaction on his side.

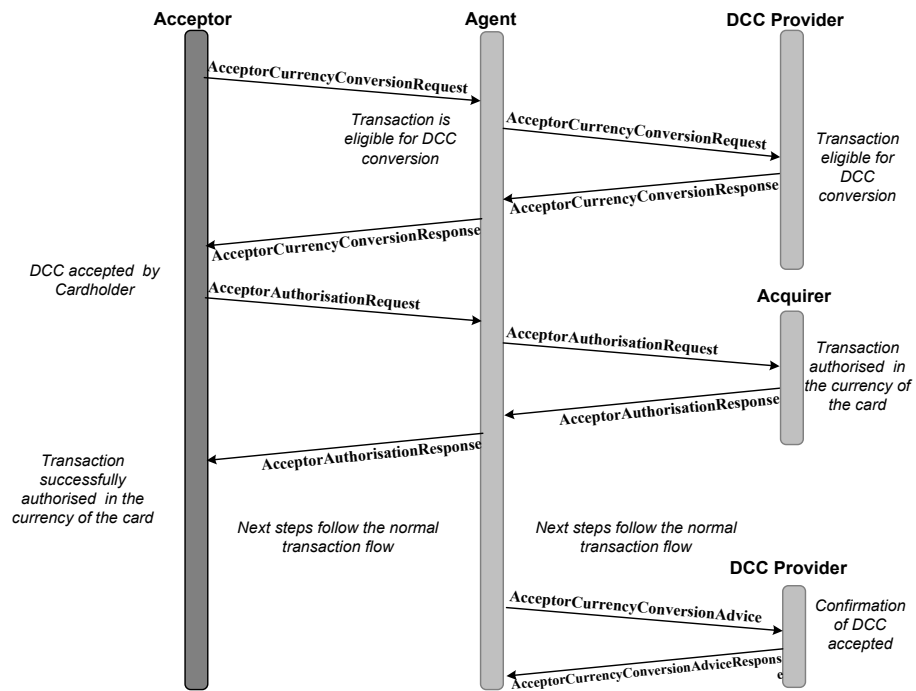


Figure 75: Transaction eligible for DCC and DCC accepted by the Cardholder

6.7.3.2 Transaction eligible for DCC and partially approved by the Issuer

In this scenario, the transaction is considered eligible for DCC by the Agent and the DCC service provider. It is also assumed that the DCC service provider accept to deliver a catalogue of a range of amounts for which DCC is offered to the cardholder which would permit partial authorisation by the Issuer. The Cardholder accepts the DCC service for the amount in the currency of his card presented to him. The transaction is authorised by the Issuer but for a reduced amount in the currency of his card (*PartialAuthorisation*). The Cardholder accepts the transaction for this reduced amount. The transaction proceeds as a normal card payment transaction in the currency of the card and for the reduced authorised amount.

1. The Acceptor sends an *AcceptorCurrencyConversionRequest* message to the Agent with the following information :
 - *CurrencyConversionResult*: absent
 - *TransactionType* : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - *AdditionalService* : Cashback, Gratuity (if *TransactionType* = CardPayment)
 - *TransactionDetails.TotalAmount*: Total Amount in the Acceptor's currency.
2. The Agent forwards the *AcceptorCurrencyConversionRequest* message with following information to the DCC service provider:
 - *Card.CardCurrencyCode*: Target currency (currency of the card).
 - *TransactionDetails.Currency*: Source currency (currency of the Acceptor).
 - *TransactionCapture*: not relevant.
 - *TransactionType*: copy from the *AcceptorCurrencyConversionRequest* message
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message containing the following information:
 - *Result* : Allowed
 - *CurrencyConversionResult.Result*: CATG (Conversion accepted for a range of amounts – rate and fees specified per range of amounts)
 - Conversion details per range of amounts provided
4. The Agent forwards the *AcceptorCurrencyConversionResponse* message to the Acceptor with the currency conversion data.
5. The cardholder accepts the proposed currency conversion.
6. The Acceptor sends an *AcceptorAuthorisationRequest* message with the agreed currency conversion data:
 - *TransactionType* : copy of the *AcceptorCurrencyConversionRequest* message
 - *TransactionDetail.Currency*: copy of *ConversionDetails.TargetCurrency* of the *AcceptorCurrencyConversionResponse* message.
 - *TransactionDetail.TotalAmount* : copy of *ConversionDetails.ResultingAmount* of the *AcceptorCurrencyConversionResponse* message
 - *AdditionalService* : copy of the *AcceptorCurrencyConversionRequest* message plus DCC information
 - *CurrencyConversionResult.AcceptedByCardholder* : True
 - *CurrencyConversionResult.ConversionDetails*: copy of *AcceptorCurrencyConversionResponse*
7. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the currency of the card.

8. The Issuer authorises the transaction but for a reduced amount. The Acquirer sends back an *AcceptorAuthorisationResponse* message to the Agent. The Agent computes the actual amount to be presented to the Cardholder (after computation of the fees and taking into account the reduced authorised amount by the Issuer) and forwards the message to the Acceptor. The Acceptor presents to the Cardholder the reduced authorised amount after the withdrawal of fees and in the currency of the card.
9. The Cardholder accepts the DCC transaction for the proposed reduced authorised amount in the currency of his card.
10. The Acceptor sends an *AcceptorAuthorisationRequest* message (confirmation of acceptance of the Cardholder) to the Agent for the reduced amount presented to the Cardholder which is then further forwarded to the Acquirer.
11. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.) with some additional information related to the conversion process.
12. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* message that the DCC transaction was completed.
13. The DCC service provider responds with an *AcceptorCurrencyConversionAdviceResponse* message about the completion of the transaction on his side.

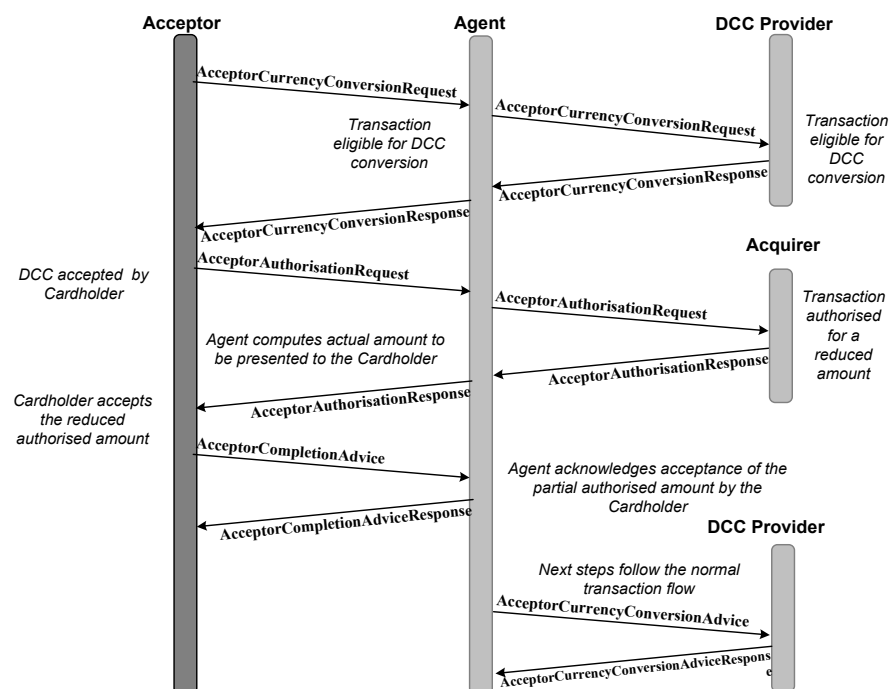
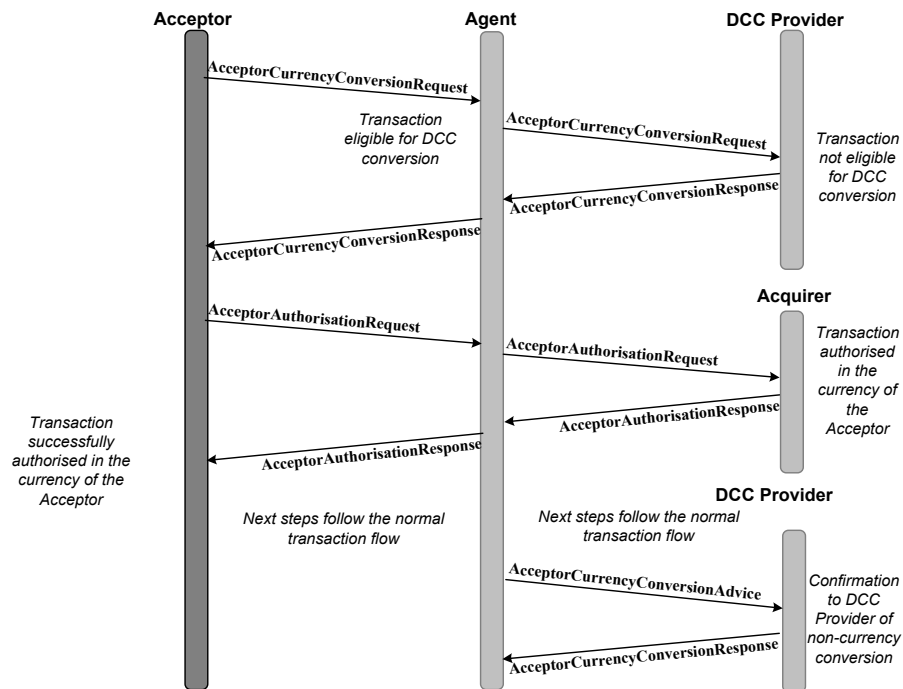


Figure 76: Transaction eligible for DCC and partially approved by the Issuer

6.7.3.3 Transaction not eligible for DCC by DCC service provider

In this scenario, the transaction is considered eligible for DCC by the Agent but not by the DCC service provider. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant.

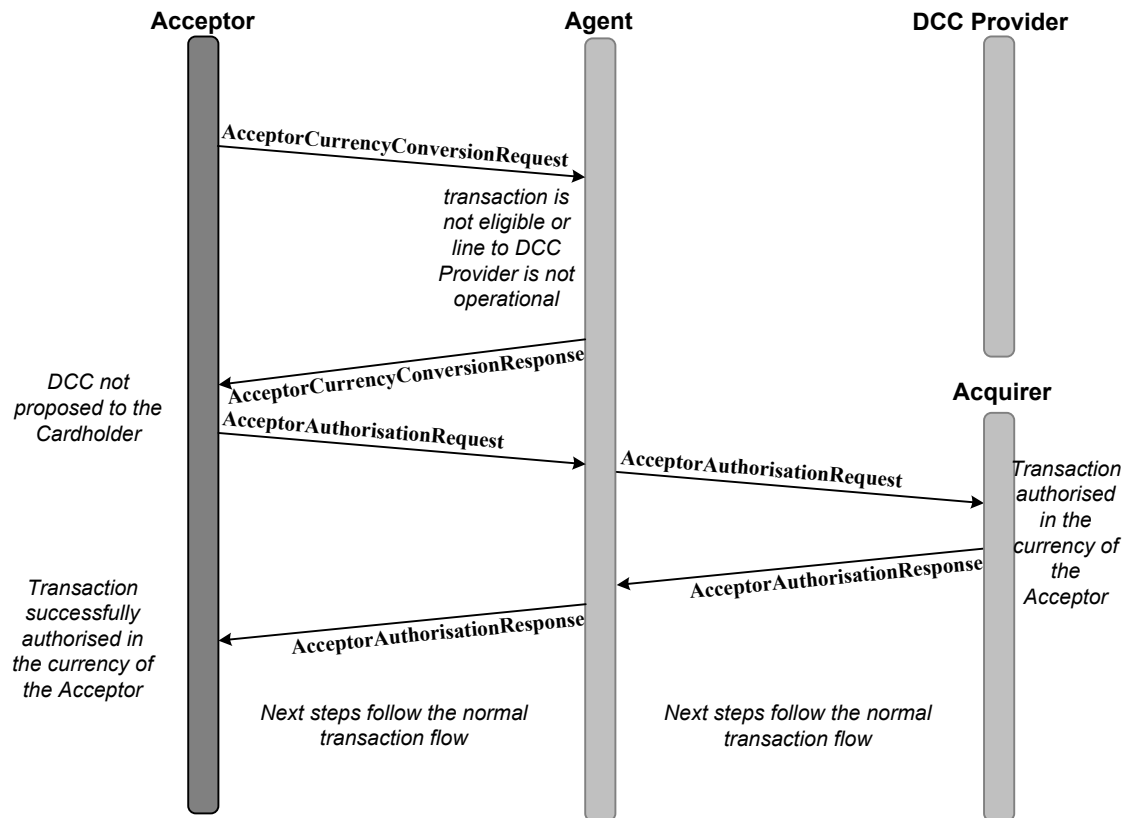
1. The Acceptor sends an *AcceptorCurrencyConversionRequest* message to the Agent with the following information:
 - CurrencyConversionResult: absent
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The Agent detects that the transaction is eligible for DCC and forwards the *AcceptorCurrencyConversionRequest* message to the DCC service provider with the following information:
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture: not relevant.
 - TransactionType : copy from *AcceptorCurrencyConversionRequest* message
8. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message with the following information:
 - Result : InvalidCard, NoRate, InvalidMerchant, NotAvailablen or InvalidProduct
 - ConversionDetails: absent
3. The Agent forwards the *AcceptorCurrencyConversionResponse* message to the Acceptor to inform him that DCC was not possible with the following information:
 - Result: InvalidCard, NoRate, InvalidMerchant, NotAvailable or InvalidProduct
 - Conversiondetails : absent
4. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Acquirer with the following information:
 - TransactionType : copy of the original *AcceptorCurrencyConversionRequest* message
 - TransactionDetails.Currency : copy of Conversion.TargetCurrency of the *AcceptorCurrencyConversionResponse* message
 - TransactionDetail.TotalAmount : copy of the Conversion.ResultingAmount of the *AcceptorCurrencyConversionResponse* message
 - AdditionalService : copy of the original *AcceptorCurrencyConversionRequest* message plus DCC
 - CurrencyConversionResult: absent
 - OriginalTransaction: Link to the original *AcceptorCurrencyConversionRequest* message
5. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the Acceptor's currency
6. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.)

**Figure 77: Transaction not eligible for DCC by DCC service**

6.7.3.4 Transaction not eligible for DCC by the Agent

In this scenario, the transaction is not considered eligible for DCC by the Agent. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant.

1. The acceptor sends the *AcceptorCurrencyConversionRequest* message to the Agent with the following information:
 - CurrencyConversionResult: absent
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency
2. The Agent detects that the card is not eligible for DCC (the Acceptor doesn't have a contract with a DCC service provider or the line to the DCC Provider is not operational).
3. The Agent send an *AcceptorCurrencyConversionResponse* message to the Acceptor to inform that DCC is not possible with the following information:
 - Result: InvalidCard, NoRate, InvalidMerchant, NotAvailable or InvalidProduct
 - ConversionDetails: absent
4. The Acceptor sends an *AcceptorAuthorisationRequest* message in the currency of the Acceptor with the following information:
 - TransactionType : copy of the original *AcceptorCurrencyConversionRequest* message
 - TransactionDetails.Currency : copy of the original *AcceptorCurrencyConversionRequest* message
 - TransactionDetail.TotalAmount : copy of the original *AcceptorCurrencyConversionRequest* message
 - AdditionalService : copy of original *AcceptorCurrencyConversionRequest* message plus DCC information
 - CurrencyConversionResult : absent
 - OriginalTransaction: Link to *AcceptorCurrencyConversionRequest* message.
5. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the Acceptor's currency
6. The Acquirer sends back an *AcceptorAuthorisationResponse* message to the Acceptor.
7. Next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).

**Figure 78: Transaction not eligible for DCC by the Agent**

6.7.3.5 Transaction eligible for DCC but DCC refused by the Cardholder

In this scenario, the transaction is considered eligible for DCC by both the Agent and the DCC service provider. The cardholder refused the DCC service for the amount presented to him in the currency of his card. The transaction proceeds as a normal card payment transaction in the currency of the merchant.

1. The Acceptor sends an *AcceptorCurrencyConversionRequest* message to the Agent with the following information:
 - CurrencyConversionResult: absent
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in Acceptor's currency
2. The Agent forwards the *AcceptorCurrencyConversionRequest* message to the DCC service provider with the following information:
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture : not relevant
 - TransactionType : copy from *AcceptorCurrencyConversionRequest* message
3. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message with the following information:
 - Result of the currency conversion : Allowed
 - ConversionDetails: Present
4. The Agent sends back an *AcceptorCurrencyConversionResponse* message to the Acceptor with the proposed currency conversion data.
5. The cardholder refuses the DCC service.
6. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Agent in the currency of the Acceptor and with the following information:
 - TransactionType : copy of original *AcceptorCurrencyConversionRequest* message
 - TransactionDetails.Currency : copy of the original *AcceptorCurrencyConversionRequest* message
 - TransactionDetail.TotalAmount : copy of the original *AcceptorCurrencyConversionRequest* message
 - AdditionalService : copy original *AcceptorCurrencyConversionRequest* message plus DCC information.
 - CurrencyConversionResult.AcceptedByCardholder : False
 - CurrencyConversionResult.ConversionDetails: copy of original *AcceptorCurrencyConversionResponse* message.
7. The Agent sends an *AcceptorAuthorisationRequest* message to the Acquirer in the Acceptor's currency.
8. The Acquirer send back an *AcceptorAuthorisationResponse* message to the Acceptor.
9. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).
10. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* that the DCC transaction was refused by the Cardholder.
11. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* about the completion of the transaction on his side

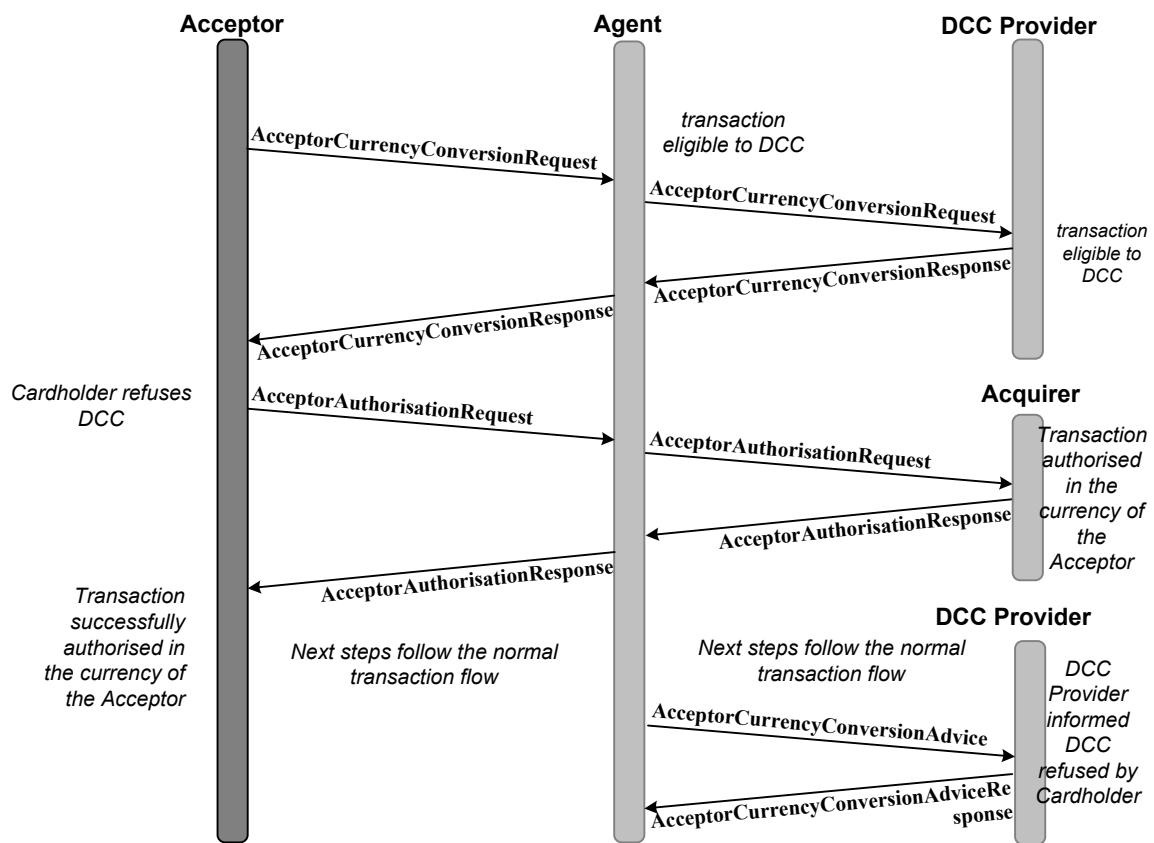


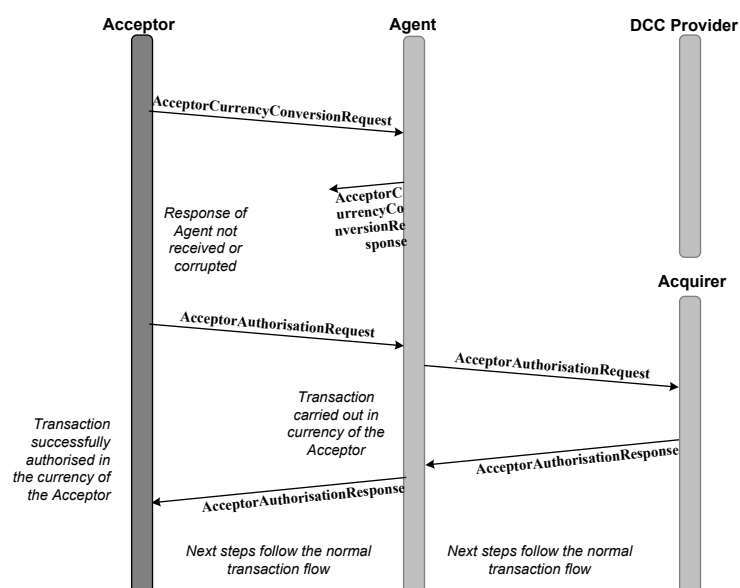
Figure 79: Transaction eligible for DCC but DCC refused by the Cardholder

6.7.3.6 Transaction eligible for DCC but interrupted due to an incident

6.7.3.6.1 Incident at the Agent level

In this scenario, the transaction is considered eligible for DCC by the Agent the response of the Agent didn't reach the Acceptor or an error was encountered in the response. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant. A negative advice is forwarded to the DCC service provider at the end of the transaction.

1. The Acceptor sends an *AcceptorCurrencyConversionRequest* message to the Agent with the following information:
 - CurrencyConversionResult: absent
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
2. The *AcceptorCurrencyConversionResponse* is not received by the Acceptor due to an incident.
3. The Cardholder is not offered the possibility to proceed with DCC.
4. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Acquirer with the following information:
 - TransactionType : copy of the original *AcceptorCurrencyConversionRequest* message
 - TransactionDetails.Currency : copy of Conversion.TargetCurrency of the *AcceptorCurrencyConversionResponse* message
 - TransactionDetail.TotalAmount : copy of the Conversion.ResultingAmount of the *AcceptorCurrencyConversionResponse* message
 - AdditionalService : copy of the original *AcceptorCurrencyConversionRequest* message plus DCC
 - CurrencyConversionResult: absent
 - OriginalTransaction: Link to the original *AcceptorCurrencyConversionRequest* message
5. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the Acceptor's currency
6. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).

**Figure 80: Incident at the Agent level**

6.7.3.6.2 Incident at DCC service provider level

In this scenario, the transaction is considered eligible for DCC by the Agent and the DCC service provider but the response sent by the DCC provider didn't reach the Agent or an error was encountered in the response. The cardholder is not offered the DCC service and the transaction proceeds as a normal card payment transaction in the currency of the merchant. A negative advice is forwarded to the DCC service provider at the end of the transaction.

7. The Acceptor sends an *AcceptorCurrencyConversionRequest* message to the Agent with the following information:
 - CurrencyConversionResult: absent
 - TransactionType : CardPayment, CashAdvance, Refund, Reservation, QuasiCash or DeferredPayment
 - AdditionalService : Cashback, Gratuity (if TransactionType = CardPayment)
 - TransactionDetails.TotalAmount: Total Amount in the Acceptor's currency.
8. The Agent detects that the transaction is eligible for DCC and forwards the *AcceptorCurrencyConversionRequest* message to the DCC service provider with the following information:
 - Card.CardCurrencyCode: Target currency (currency of the card).
 - TransactionDetails.Currency: Source currency (currency of the Acceptor).
 - TransactionCapture: not relevant.
 - TransactionType : copy from *AcceptorCurrencyConversionRequest* message
9. The DCC service provider answers with an *AcceptorCurrencyConversionResponse* message.
10. The *AcceptorCurrencyConversionResponse* is not received by the Agent due to an incident.
11. The Agent send an *AcceptorCurrencyConversionResponse* message to the Acceptor to inform him that DCC was not possible with the following information:
 - Result: InvalidCard, NoRate, InvalidMerchant, NotAvailable or InvalidProduct
 - Conversiondetails : absent
12. The Cardholder is not offered the possibility to proceed with DCC.
13. The Acceptor sends an *AcceptorAuthorisationRequest* message to the Acquirer with the following information:
 - TransactionType : copy of the original *AcceptorCurrencyConversionRequest* message
 - TransactionDetails.Currency : copy of Conversion.TargetCurrency of the *AcceptorCurrencyConversionResponse* message
 - TransactionDetail.TotalAmount : copy of the Conversion.ResultingAmount of the *AcceptorCurrencyConversionResponse* message
 - AdditionalService : copy of the original *AcceptorCurrencyConversionRequest* message plus DCC
 - CurrencyConversionResult: absent
 - OriginalTransaction: Link to the original *AcceptorCurrencyConversionRequest* message
14. The Agent forwards the *AcceptorAuthorisationRequest* message to the Acquirer in the Acceptor's currency
15. The next steps follow the normal transaction flow (i.e. Completion capture, batch capture, etc.).
16. The Agent or Acquirer advises the DCC service provider with an *AcceptorCurrencyConversionAdvice* message that the DCC transaction did not succeed.
17. The DCC service provider responses with an *AcceptorCurrencyConversionAdviceResponse* message about the completion of the transaction on his side.

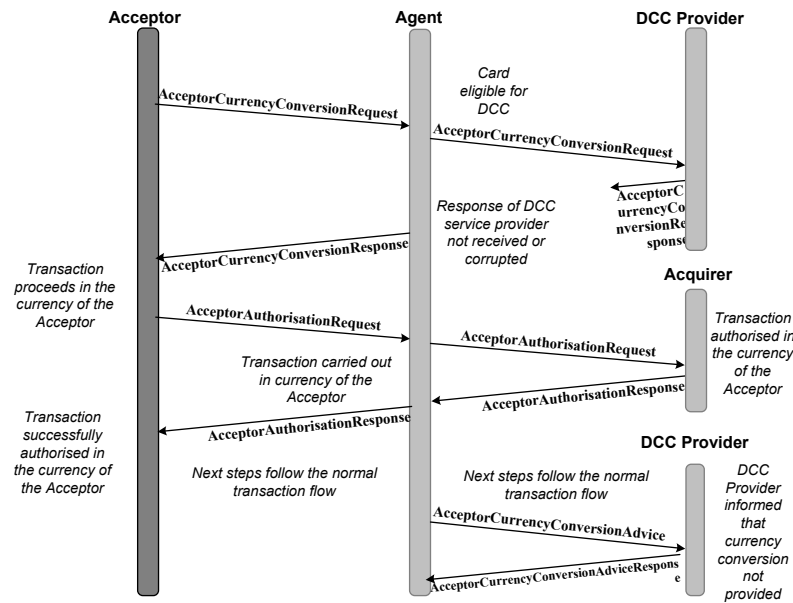


Figure 81: Incident at DCC service provider level

6.7.4 Reconciliation Totals

The following elements shall be used to compute reconciliation totals:

- a. TransactionDetails.TotalAmount
- b. TransactionDetails.Currency

The Totals should be computed by Currency if the parameter TotalsPerCurrency in the AcceptorConfigurationUpdate is set to True

7 Messages Examples

The examples are now provided in new document. See “Card Payment Message Example version 6.0”.

8 Transport Protocols and Services

8.1 Protocols Organisation

Protocol implementation shall follow the general principle of robustness: be strict in what you do, be tolerant in what you accept from others.

The Acquirer Protocols follows the traditional organisation in five layers of the model defined for the TCP/IP protocol suite.

The two layers we are interested in are the Application Layer where the Acquirer protocol is located, and the Transport Layer that the Application protocols interface with the Transport Services.

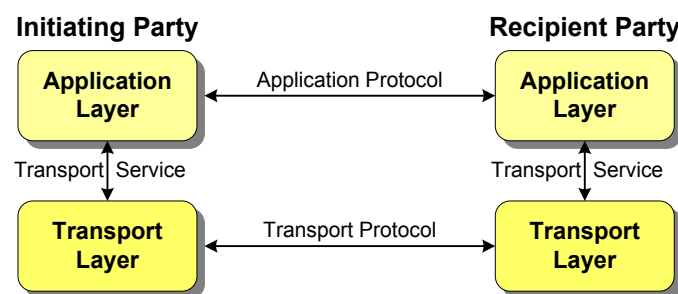


Figure 82: Protocols Organisation

The Application Protocol is independent of the Transport Protocol, and can be used without modification with various Transport Protocols, except where the interface to the Transport Services needs specific information such as the transport addresses.

The Transport Protocol and the Application Protocol are also independent of the communication infrastructure used below these layers, with the exception of the quality of service of the communication which may involve tuning of some values of the configuration parameters (e.g. value of timeout).

The Transport Protocol is a standard protocol allowed by the Application Protocol. Application Protocol requires the Transport Services which can be used by the Application Protocol, and specifies the way to use them.

If a standard Transport Protocol used by the Application Protocol does not offer a particular Transport Service required by the Application Protocol, the Application Protocol specifies this Service through a Transport Adaptation Layer. This is for instance the case for the application message delimitation, which is a service not provided by the TCP Transport Protocol, but required for the decoding of a XML message before delivering it to the Application Layer. It could be also be used to provide a particular flow control.

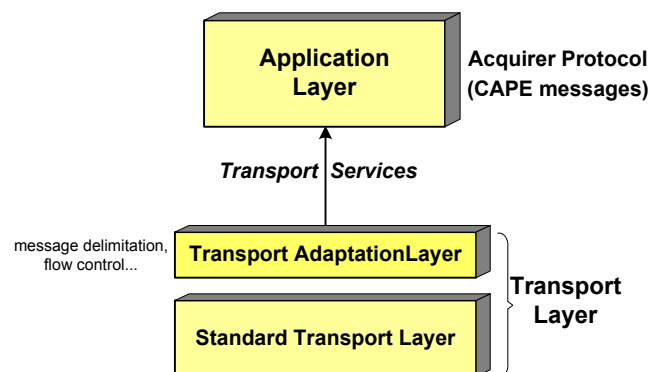


Figure 83: Transport Adaptation Layer

As far as transport services and their usage are defined, the definition in future of a new Transport Protocol compliant to the defined Transport Service, will not impact the specifications of the Application Protocol.

For the current specification the only selected transport protocol is TCP (Transmission Control Protocol, specified in the RFC 793).

8.2 TCP Protocol

8.2.1 Typical Use

Because of its widespread availability, the TCP transport protocol remains the favourite transport protocol. It can be used as:

1. An end-to-end transport protocol between two entities at each extremity of the Application Protocol.

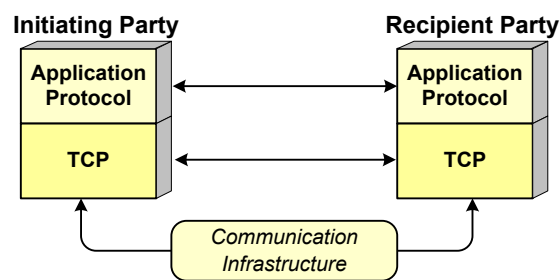


Figure 84: Peer-to-peer TCP Transport Protocol

2. An interface to a gateway or a driver to make the conversion of transport protocol with the other side of the Application Protocol. This case is only a transitory solution allowing the adaptation a legacy system to the Application Protocol, avoiding the specification of all the existing communication protocols.

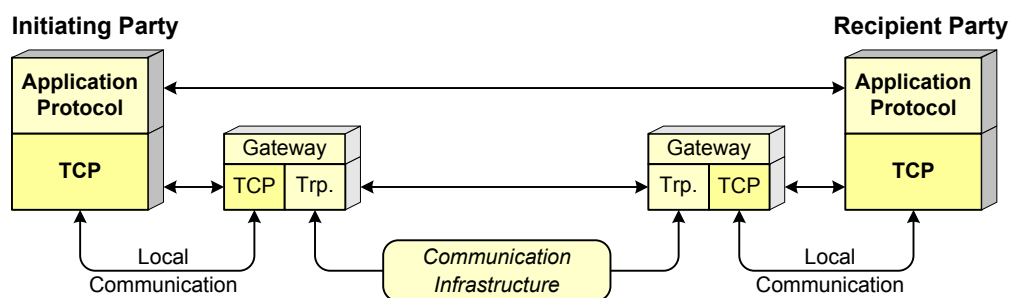


Figure 85: Gateway TCP Transport Protocol

8.2.2 Message Delimitation

The TCP protocol is a stream protocol which does not offer message delimitation or data-unit delimiting. It uses the general mechanism of message delimitation provided for all the transport protocols.

8.2.3 Addressing

A transport address for the TCP protocol is composed of:

1. The *IP Address* or the *DNS Name* to resolve of the host on which the application protocol lives.
2. The *TCP Port*, to dispatch the connections to the application.

8.3 Transport Services

8.3.1 Message Delimitation

8.3.1.1 Definition

Message delimitation provides the recognition of the application data (i.e. application message).

8.3.1.2 Specifications

Application messages are prefixed by four bytes containing the length, in network order¹³, of the application message. This length value does not include the length of this prefix.



Figure 86: Header Length

If the four bytes are equal to zero, it is considered as a zero length application message, and these four bytes are ignored.

8.3.1.3 Typical Example of Implementation

For sending a message:

- The Application Protocol layer requests to send an application message to the interface of the transport protocol,
- This interface, or transport adaptation layer, adds the length prefix described above and passes the new message to the transport protocol.

For the receipt of a message, the interface to the transport protocol:

- Waits for the receipt of four bytes to get the length L of the message to receive, and arms the timeout TC4 to monitor the receipt of the complete application message.
- Waits for the receipt of L bytes to provide a message to the Application Protocol layer,
- On the receipt of the last data unit of this message, stop the TC4 timer, and deliver to the application layer, the L bytes, content of message.

Timer management is specified in section *Connection and Data Management* State Diagrams, which contains message sending and message receiving management.

8.3.1.4 Notes

Use of message delimitation at the transport level:

- Restricts message delimitation to the transport layer interface avoids partial progressive decoding of the message mixed with receipt of message fragments,
- Separates decoding of the message from its receipt processing.
- Independence of this service from the data coding used by the application, if the data coding can provide this functionality (e.g. ASN.1/BER data coding).

Use of the message length to delimit a message has the advantage of being able to send any value in the application message, but prohibits resynchronisation of message after a data loss.

¹³ Most significant byte first (i.e.; big endian).

Use of a fixed number of bytes to contain the length of the message facilitates implementation of message receipt.

8.3.2 Connection and Data Transfer Management

8.3.2.1 Connection Services

Application protocols use the transport connection services which are organised in three phases:

- The *Connection Establishment*, to establish an association between the Initiating Party and the Recipient Party,
- The *Transport of Data Units* or *Data Transfer*, to exchange application messages within an already established connection, and
- The *Connection Release* service to release a connection established beforehand.

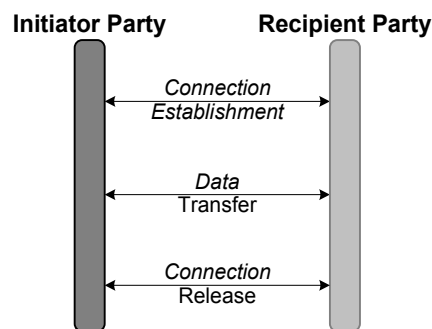


Figure 87: Connection Services

These services are usually done by transport primitives' services through various implementations:

- *Connection Request*: to send an outgoing request in order to establish a transport connection (done by the Initiating Party).
- *Connection Indication*: for the receipt an incoming request from the Initiating Party to establish a transport connection (done by the Recipient Party).
- *Connection Response*: to send the positive or negative response to the incoming request for the transport connection establishment (done by the Recipient Party).
- *Connection Confirmation*: for the receipt an incoming response from the Recipient Party to the previous outgoing request for the transport connection establishment (done by the Initiating Party).
- *Data Request*: to send an outgoing request containing data to deliver to the other peer of a transport connection (done by the Initiating Party or the Recipient Party).
- *Data Acknowledgment*: for the receipt an incoming request informing the sender of a *Data Request* that the transport layer has sent the related data (done by the Initiating Party or the Recipient Party).
- *Data Indication*: for the receipt an incoming request containing data delivered by the other peer of a transport connection (done by the Initiating Party or the Recipient Party).
- *Disconnection Request*: to send an outgoing request in order to release a transport connection (done by the Initiating Party or the Recipient Party).
- *Disconnection Indication*: for the receipt an incoming request to release a transport connection (done by the Initiating Party or the Recipient Party).

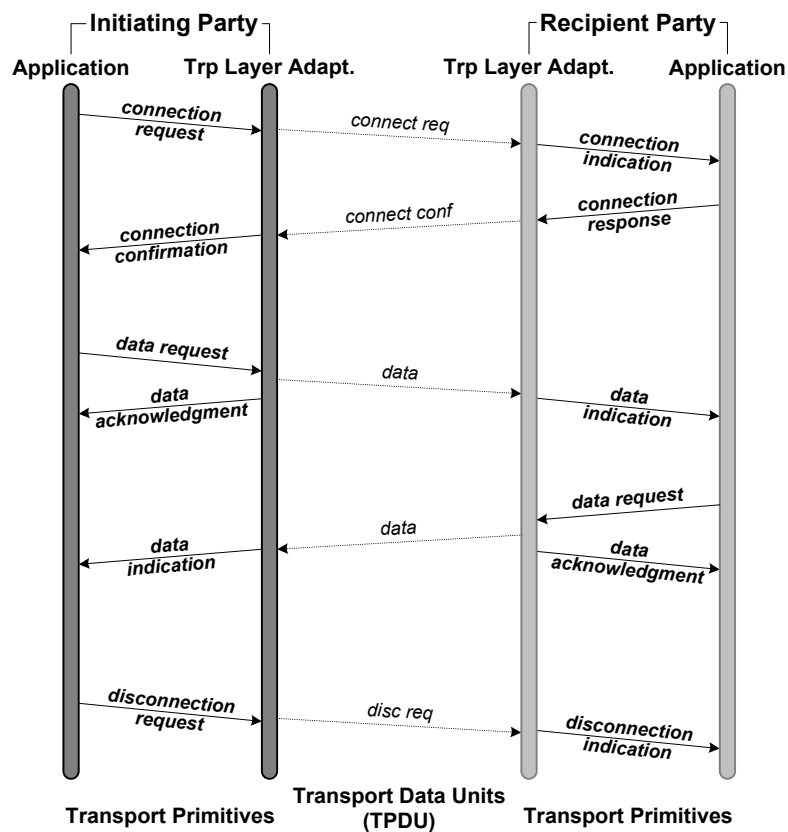


Figure 88: Transport Connection Management Service Primitives

8.3.2.2 Data Transfer

The real time or interactive exchanges of the Application Protocol are exclusively dialogues of the type question/answer, involving the related transport primitives:

- The question is a request message sent by the *Initiating Party* of the message pair to the *Recipient Party* of the message pair.
- The answer is a response message sent by the *Recipient Party* of the message pair to the *Initiating Party* of the message pair.

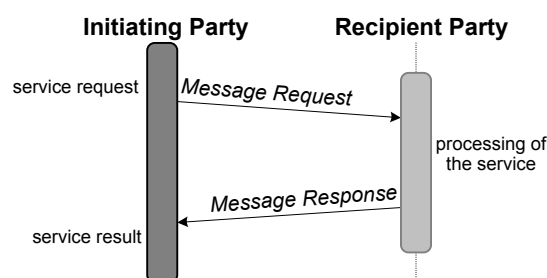


Figure 89: Application Protocol Message Flow

This message sequence is used to request a service to the Recipient Party, the response message is sent to the Initiating Party with the result of the processing. In some case, the request message notifies a particular event to the Recipient Party, the response message is then an acknowledgement sent to the Initiating Party.

A pair of request and response messages is identified unambiguously. If a request message is repeated with the same identifier, (e.g. in case of error, or if the response message have not been received), depending on the application protocol specification, it may be refused by the Recipient Party or generate the same response message.

8.3.3 Single Message Pair Exchange

8.3.3.1 Definition

Each pair of application messages is enclosed in a unique transport connection that is established for that purpose by the Initiating Party.

8.3.3.2 Specifications

The sequence flow of a message pair exchange inside a transport connection contains the steps presented below.

1. The Initiating Party sends a transport connection request to the Recipient Party. The Initiating Party arms the timeout TC1 to wait for the transport connection response from the Recipient Party.
2. The Recipient Party receives the transport connection request, accepts it and sends a transport connection confirmation. At this time the transport connection is established for the Recipient Party. The Recipient Party arms the timeout TC2 to wait for the application message request from the Initiating Party.
3. After receipt of the connection confirmation, the transport connection is established for the Initiating Party, it sends the application message request.
4. The Recipient Party processes the requested service and sends the result within the application message response to the Initiating Party. The Recipient Party arms the timeout TC2 to wait for the receipt of a disconnection request from the Initiating Party after it has received the application message response.
5. The Initiating Party receives the application message response and sends a transport disconnection request.
6. The Recipient Party receives the transport disconnection request and sends a transport disconnection request to the Initiating Party. The transport connection is then considered as released for the Initiating Party which has not to wait for the response of the Recipient Party.

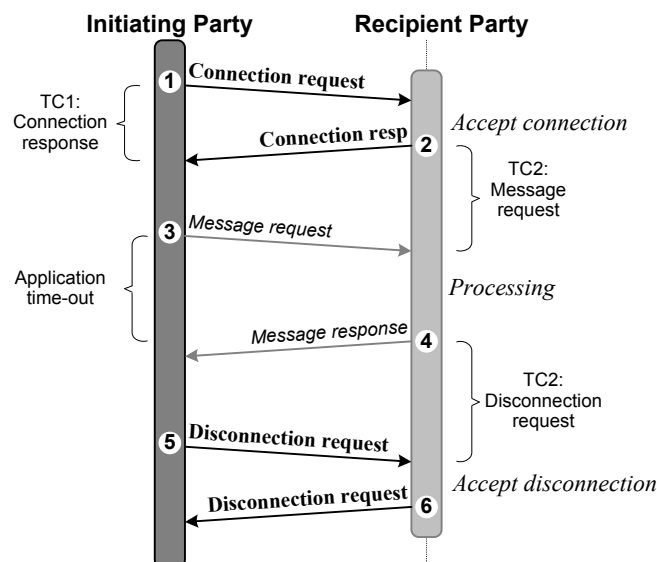


Figure 90: Single Message Pair Exchange Sequence Flow

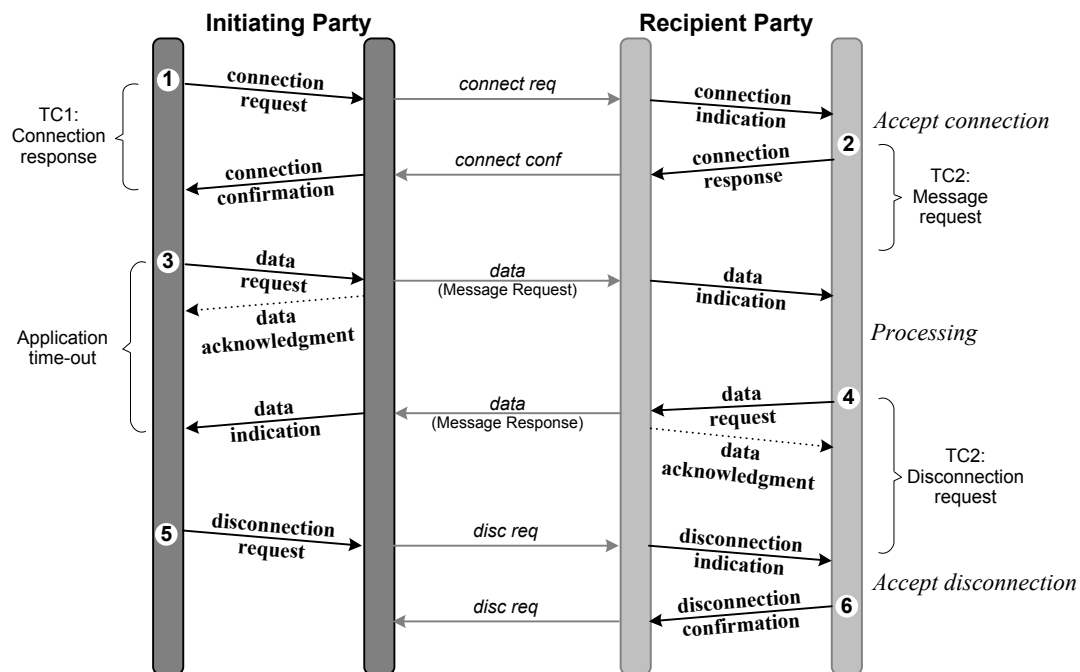


Figure 91: Single Message Pair Exchange Transport Primitives Sequence Flow

8.3.3.3 Notes

Each peer of the transport connection has to activate timeout to avoid loss of resources in case of event not received by one peer. These timeout control the management of the transport connection, establishment, release, sending and receipt of data.

The application has to manage its own timeout to supervise the processing of the service by the Recipient Party. This is not the role of the transport service to specify the behaviour of the application layer, but on the other hand the application layer might send disconnection request for this reason or others to release the transport connection at any time.

A best practice for the connection release is that the receiver of the last message has the responsibility to release the connection.

The value of the timeout TC2 has to be large enough to take into account the management of connections by the Initiating Party. Its purpose is rather to avoid a forgotten transport connection that remains open forever.

8.3.4 Multiple Message Pair Exchange

8.3.4.1 Definition

Multiple Message Pair Exchange in the same connection is the option to keep a transport connection open (or alive) in order to send the next application message pair on the same transport connection.

It allows the exchange of a sequence of message pairs on the same connection, avoiding the overhead of the connection establishment.

8.3.4.2 Specifications

Sequence flow of a message pair exchange inside a transport connection contains the steps below.

1. The Initiating Party and the Recipient Party establish the transport connection. The Initiating Party sends the application message request number 1.
2. When the Initiating Party receives the application message response number 1, it arms the timeout TC3 to wait for another message request to send from the application layer.
3. The Initiating Party sends the application message request number 2 before the expiration of the timeout TC3.
4. The Initiating Party receives the application message response number 2 and arms again the timeout TC3 to wait for the message response from the Recipient Party.
5. In the absence of new application message to send, the TC3 timer timed out. The Requester sends then the transport disconnection request to release the connection because of inactivity.

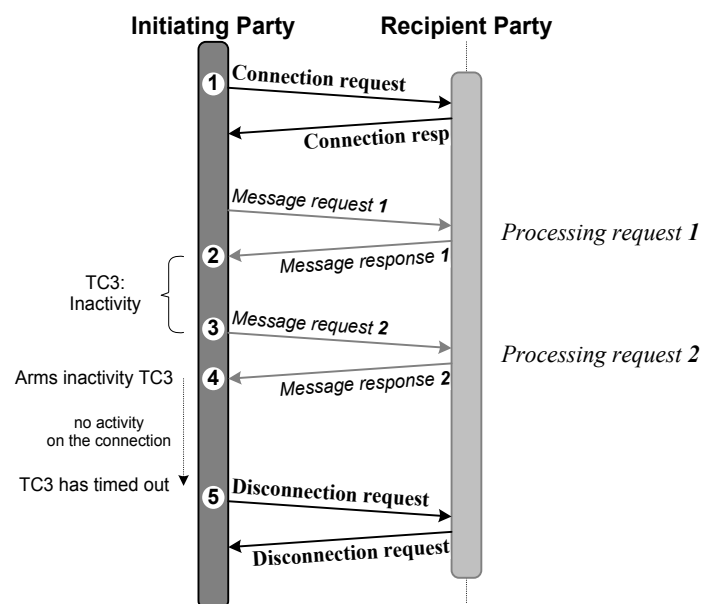


Figure 92: Multiple Message Pair Exchange Sequence Flow

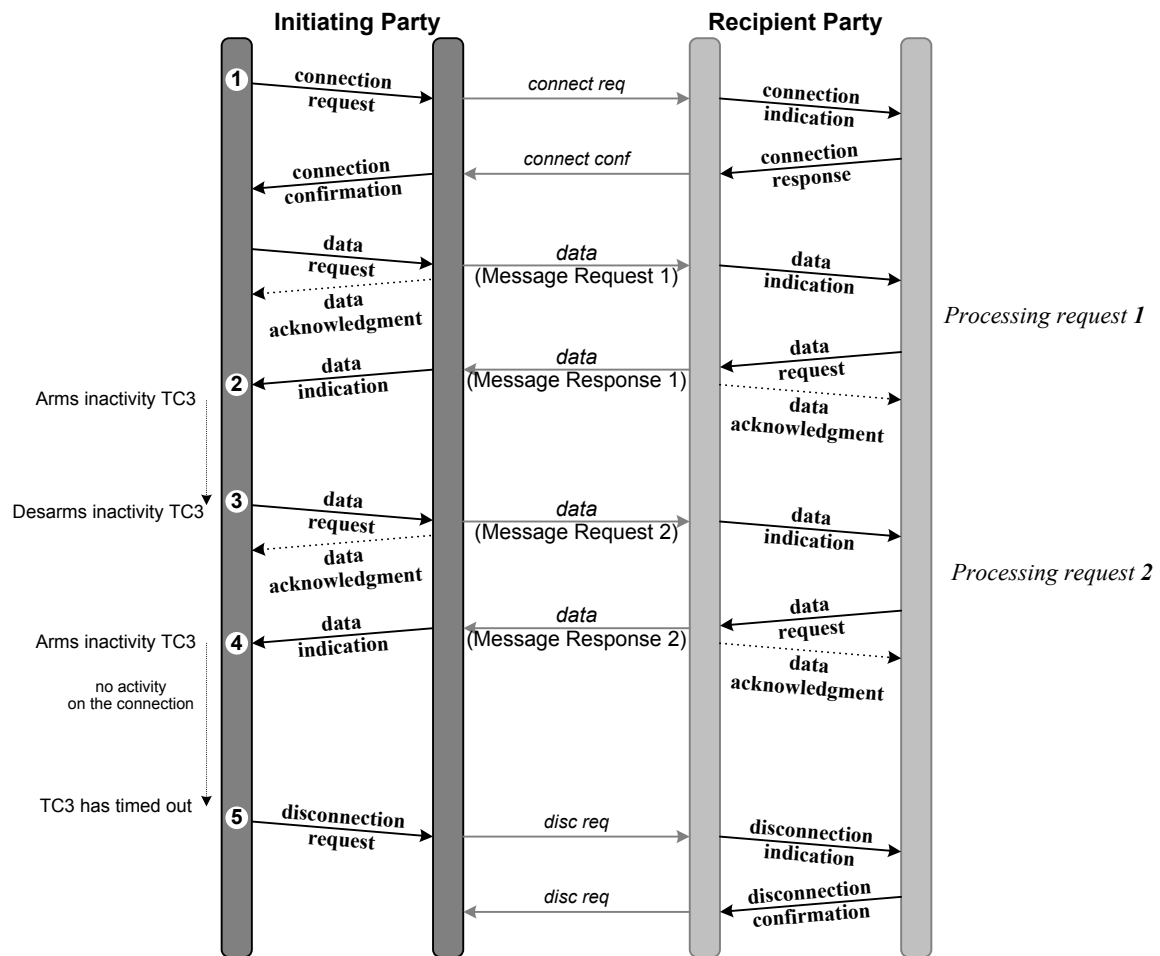


Figure 93: Multiple Message Pair Exchange Transport Primitives Sequence Flow

8.3.4.3 Notes

The transport is based upon a connectionless network protocol like IP that does not reserve any resource on the network except at the of extremities of the transport connection. In this case the overhead of a connection establishment is the exchange of two packets, and the resource consumption of an established connection context entry in the protocol stack.

This mechanism has several advantages to manage significant volume of messages:

- To avoid the use of a header carrying application-only information, and provide a better separation between transport and application layers.
- The simplicity of the message management by the transport service layer and the application layer, for the Initiating Party and the Recipient Party.
- To be able to verify of the response on both transport layer (the transport connection) and application layer (application data in the header of the message), with a clear separation between these two layers.
- The adaptability of the communication resources to the traffic of messages.
- The ability to isolate the processing and resolution of problems in the various message pairs.

8.3.5 Addressing

8.3.5.1 Definition

A peer (Recipient and Initiating Parties) is identified by its transport address; it is this address which is used when connecting to a peer.

The form of the transport address depends on the type of transport protocol being used (e.g. an IP address or a DNS name of the node together with a TCP port would form a TCP transport address).

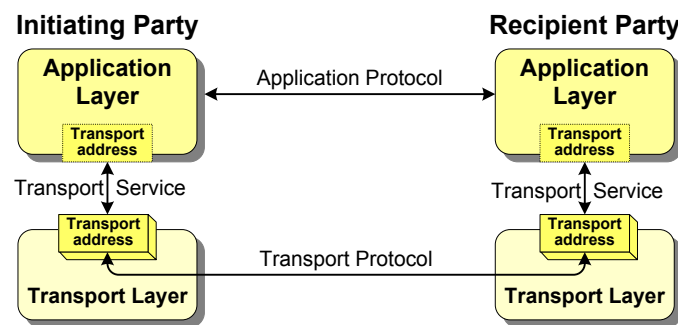


Figure 94: Transport Address

8.3.5.2 Specifications

The transport address of the Recipient Party is a parameter used at the application level by the Initiating Party as a destination address to establish a connection.

Application protocol may define assignment of the Recipient Party's transport address at any level: globally for the whole application protocol, per class of message, or per service.

8.3.6 Flow Control

8.3.6.1 Definition

Flow control is the service which allows the control of the flow of data between the application layer and the transport layer, or between the Initiating Party and the Recipient Party.

8.3.6.2 Specifications

Flow control of the transport protocol may be not available in the transport service. The Application and Transport layers have to put in place a data acknowledgment mechanism to avoid resource saturation of a layer in the event that a large volume of data is to be sent to the other peer.

Application protocols have to design a flow control between the Initiating Party and the Recipient Party. This flow control is completely transparent to the transport service layer.

The transport service used is the Data Acknowledgment primitive, sent to the application when the transport service layer is able to take into account new data.

8.3.7 Error Cases

This section defines errors detected by the transport protocol and transmitted to the application layer. The responsibility and the choice of error resolution are left to the application layer, unless the transport service layer needs to perform some action to continue to work.

The Application Layer Protocols must define the actions to be taken in the event of these errors as the actions will vary according to the message exchange (eg actions in the event of Authorisation failure are different to those to be taken when performing a Completion exchange).

The errors are generic; the transport layer is not always able to report a specific reason for the errors.

8.3.7.1 Unable to Establish a Transport Connection (ERTR01)

Definition

The transport layer is unable to establish the transport connection that the application layer has requested to open, because:

- The lower protocol layer has notified an “out of service” communication state as permanent or in response to the attempt to open the connection.
- The TC1 timer has expired, and the connection in progress is considered as broken (the connection establishment as a failure).
- The Recipient Party has denied the connection request.
- The transport layer has reached the maximum number of concurrent connections it can manage.

Transport Service Behaviour

The transport service layer considers the establishment of the transport connection as a failure, and notifies the result to the application layer.

8.3.7.2 Transport Connection Broken (ERTR02)

Definition

The transport layer realises that an open connection is broken. Several reasons may cause this error:

- The transport layer receives an unexpected disconnection request.
- An error occurred at a lower level while the transport layer waits for the end of a message after the receipt of the length prefix and incomplete data units.
- The lower level notifies the transport layer that a defect occurs (e.g. on the physical interface), and the transport connection is broken.
- The remote peer has not respected the state diagram of the connection management. The Initiating Party has sent data after sending the message request but before the Recipient Party has finished sending the message response. The Recipient Party sends data before the Initiating Party has finished sending the message request.
- The remote peer has released the transport connection.
- A connection deemed to be inactive when TC2 expires. The connection seems to be forgotten by the Initiating Party, and the Recipient Party closes the transport connection at the expiration of the TC2.

Transport Service Behaviour

The transport service layer notifies the closing of the transport connection to the application layer. Partial messages are deleted and not sent to the application layer.

8.3.7.3 Unable to Send a Message (ERTR03)

Definition

The transport layer cannot send an application message. Several reasons may cause this error:

- An error occurred at a lower level when the transport layer sends an application message at the request of the application layer.
- The connection was broken just before the sending of the message, and the application layer did not receive the disconnection notification before the request to the transport layer to send a message.
- The remote peer is not ready to receive data. The Recipient Party has not yet received the message request. The Initiating Party has not received the connection confirmation or has not yet received the message response.

Transport Service Behaviour

The transport service layer notifies the error to the application layer.

Note that the transport service layer is not always able to report error to the application layer. In particular when a connection is broken, the transport service has removed the connection context, and might have some difficulty to identify the context of the error.

8.3.7.4 Message Too Big (ERTR04)

Definition

The transport layer cannot receive an application message, because the prefix contains a message length above the limit of the transport layer memory.

Transport Service Behaviour

The transport service layer has an internal parameter containing the size limit of a message, or a global limit for the set of message buffers. When the limit is reached, the transport layer:

- Releases the connection on which the message has to be received.
- Notifies the error to the application layer.

As the transport layer cannot receive the message, it has to release the transport connection. Most of the time error of this class happens when there is a desynchronisation of message delimitation between the two peers.

8.3.7.5 Late Arrival (ERTR05)

Definition

The transport service layer receives a data unit for the application after a timeout has expired and before the request for a disconnection.

The application layer must also handle this error.

The drawing below shows such error example, when the application timeout on the message response expires.

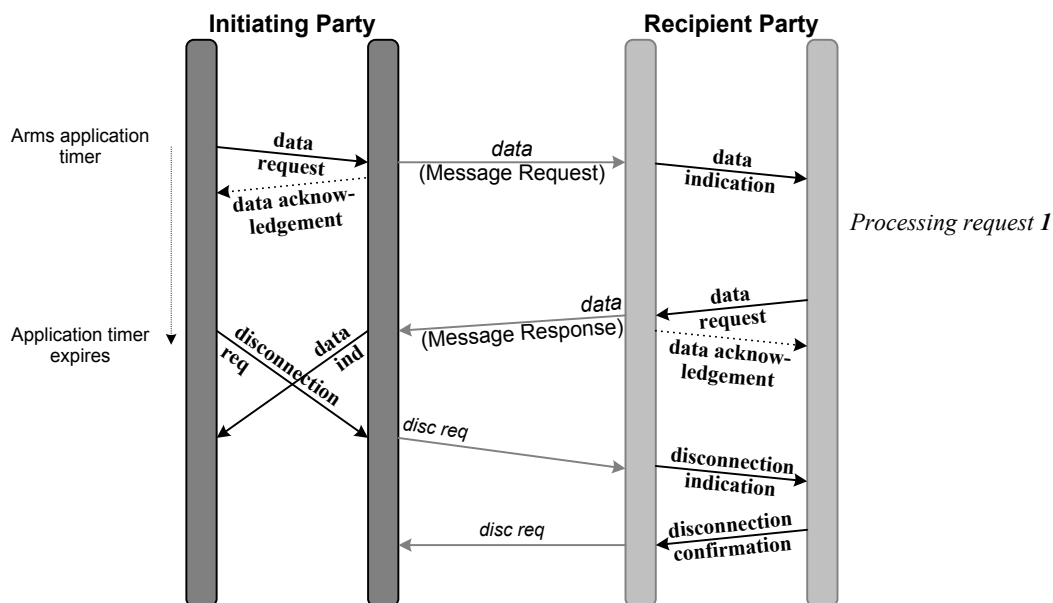


Figure 95: Late Arrival Error Example

Transport Service Behaviour

The transport service layer ignores the message or data units .

The transport layer may have removed the connection context, and may not be able to identify the context of the error. In addition, the application layer will have already started the error handling and cannot process the late response.

The implementation of the Transport Protocol has to ensure that any data unit is not assigned to the wrong connection.

8.3.7.6 Max Number of Connections (ERTR06)

Definition

If, on the arrival of an incoming connection request, the transport service layer cannot open a new transport connection this error will be returned to the Application Layer. This could occur, for example, when the Application or Transport Layer has reached the maximum number of concurrent transport connections it can handle.

Transport Service Behaviour

The behaviour is implementation dependant. For instance, the transport service layer may have an internal parameter containing the limit number of open connections. When the limit is reached, the transport layer must :

- decline the incoming connection request,
- notify the error to the application layer.

8.3.7.7 Incomplete Application Message (ERTR07)

Definition

After the receipt of an application message header, the transport layer arms the timeout TC4 to monitor the receipt of the complete application message.

This error will be triggered if the TC4 timer expires before the end of the application message is received, and might happen:

- When a serious error occurs on a lower communication level or on the communication infrastructure between the 2 peers.
- The remote transport or application layer encounters a serious error, or is out of order.

Transport Service Behaviour

The transport service layer must carry out the following actions:

1. discard the incomplete application message,
2. close the transport connection to avoid further problems of data synchronisation (the next data unit may be the start of the next application message, or part of the previous application message),
3. notify the error to the application layer.

8.3.7.8 Other Errors

All other errors are detected or resolved at the application layer, as:

- Non-understandable message: all decoding of message are made by the application layer, so this error is detected and resolved at the application layer.
- Busy situation: when the application layer has not enough resources to process all the messages received, it can answer that it is busy at the application level to the other peer.

8.3.8 Transport Service Parameters

This section summarises the set of configuration parameters required by the transport service management. These parameters are configurable and may be downloaded using the EPAS TMS protocol or another proprietary means. These values depends first upon the communication infrastructure.

<i>Name</i>	TC1
<i>Definition</i>	Transport connection establishment timer
<i>Owner</i>	Initiating Party of a transport connection establishment
<i>Usage</i>	After the sending of a transport connection request, the Initiating Party arms the TC1 timer to wait for the response from the Recipient Party. TC1 is reset on the transport connection confirmation being received..
<i>Specification</i>	8.3.3 Single Message Pair Exchange

<i>Name</i>	TC2
<i>Definition</i>	Transport connection idle timer
<i>Owner</i>	Recipient Party of a transport connection
<i>Usage</i>	Keeping open transport connections which are not being used by the Initiating Party. After the receipt of a message request or sending of a message response, the Recipient Party arms the TC2 timer to wait for the use of this transport connection. TC2 must be disarmed when a new message is sent across this connection. This timer has to be long enough to not interfere with the TC3 timer
<i>Specification</i>	8.3.3 Single Message Pair Exchange

<i>Name</i>	TC3
<i>Definition</i>	Transport connection activity timer
<i>Owner</i>	Initiating Party on a transport connection
<i>Usage</i>	This timer is armed by the Initiating Party before the release of a transport connection, to wait for a new message request to send. It is armed when a response message is received and disarmed when a new request is sent. Expiry of the timer triggers a request to close the idle connection. It avoids the management of permanent transport connections between remote peers, the multiplexing of message exchanges, and the re-connection for every message exchange.
<i>Specification</i>	0 Multiple Message Pair Exchange

<i>Name</i>	TC4
<i>Definition</i>	Application message receipt timer
<i>Owner</i>	Initiating Party or Recipient Party on a transport connection
<i>Usage</i>	This timer is armed by the Initiating Party or Recipient Party at the receipt of the application message prefix to supervise the receipt of the complete application message. It is armed at the receipt of a message header and disarmed when the message is complete. It allows the detection of an incomplete application message.
<i>Specification</i>	8.3.7.7 Incomplete Application Message

<i>Name</i>	Recipient Party Address
<i>Definition</i>	Transport address of the Recipient Party
<i>Owner</i>	Initiating Party of a transport connection
<i>Usage</i>	This (or these) transport address is used by the Initiating Party to establish a transport connection with the Recipient Party.
<i>Specification</i>	8.3.5 Addressing

8.3.9 Connection and Data Management State Diagrams

This section presents the state diagrams of the transport connection management on both Initiating Party and Recipient Party side.

These state diagrams are valid for both Single and Multiple messages pair exchanges.

The transitions between states have the syntax: *Event [Condition to Activate]/Action*

8.3.9.1 Recipient Party Diagram

States

The Recipient Party may have the following states:

- **Await Request:** The Recipient Party has received a connection request, and sent the connection confirmation. The connection is established. No message pair exchange is in progress. The Recipient Party waits for the beginning of a message request, or a disconnect request.
- **Receiving:** The Recipient Party has received the beginning of a message request, which is not complete yet.
- **Responding:** The Recipient Party has received a complete message request. The transport adaptation layer waits for application layer to send the message response.

Events

The Recipient Party may receive the following events:

- **CON_IND:** The Recipient Party receives a connection indication from the network.
- **DATA_IND:** The Recipient Party receives data.
- **DATA_REQ:** The Recipient Party application layer wants to send data.
- **TC2, TC4:** The timer TC2 (respectively TC4) has expired.
- **Error:** There is an error situation, such as receiving a disconnection request from the network or the application layer, a message that is too big, or other errors.

Diagram

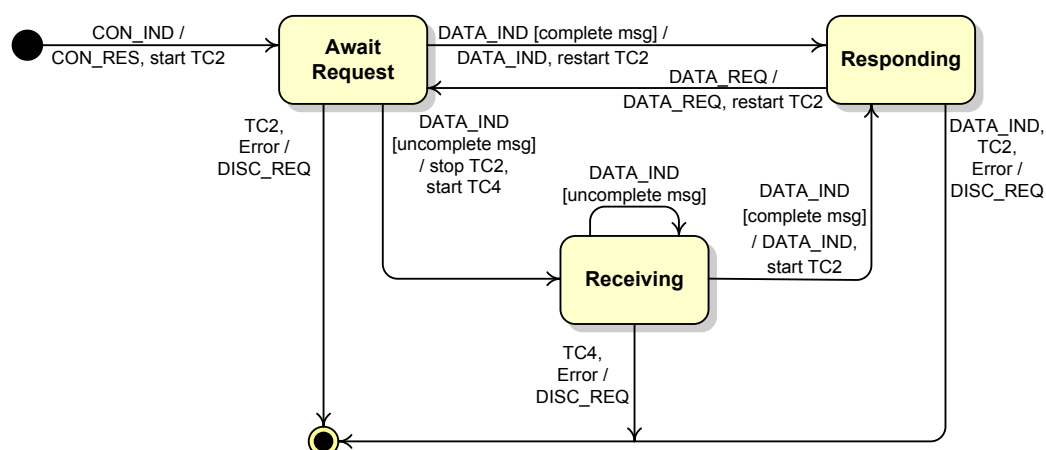


Figure 96: Recipient Party Connection Management State Diagram

8.3.9.2 Initiating Party Diagram

States

The Initiating Party may have the following states:

- **Connecting:** The Initiating Party has sent a connection request, and waits the positive or negative answer to that request.
- **Connected:** The Initiating Party has received a connection confirmation, positive response, and the transport connection is established. No message pair exchange is in progress. The transport adaptation layer waits for application layer to send a message request.
- **Await Response:** The Initiating Party has sent the message request, and has not received any part of the response message.
- **Receiving:** The Initiating Party has sent the message request, and has received the beginning of the response message which is not complete yet.

Events

The Initiating Party may receive the following events:

- **CON_REQ:** The Initiating Party application layer wants to send a connection request.
- **CON_CNF:** The Initiating Party receives a connection confirmation.
- **DATA_REQ:** The Initiating Party application layer wants to send data.
- **DATA_IND:** The Initiating Party receives data.
- **TC1, TC4:** The timer TC1 (resp. TC4) has expired.
- **Error:** There is an error situation, such as receiving a disconnection request from the network or the application layer, a message that is too big, or other errors.

Diagram

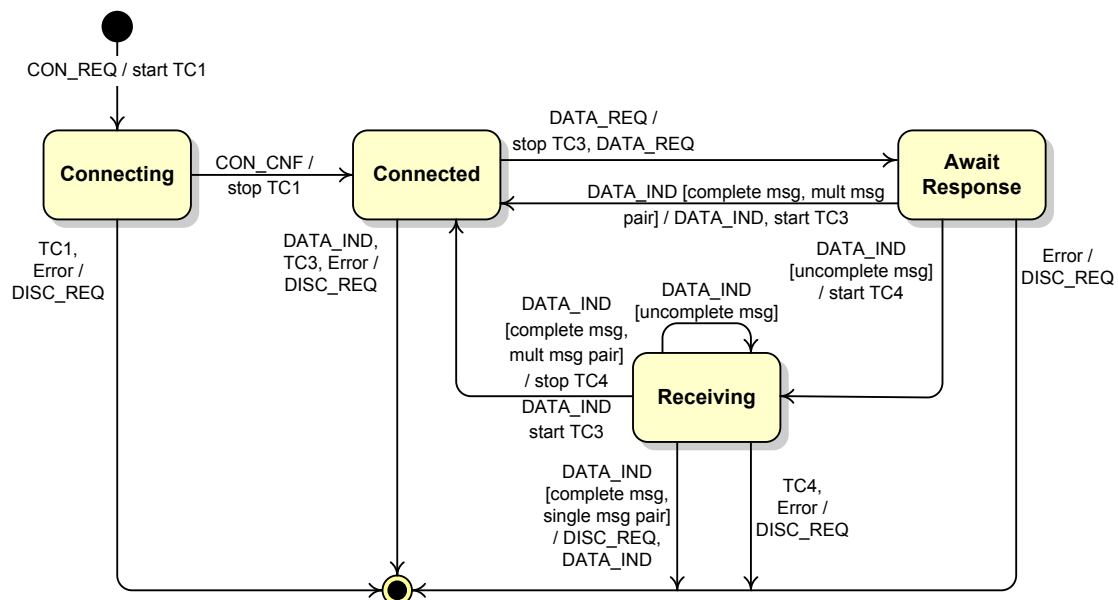


Figure 97: Initiating Party Connection Management State Diagram