x

ISO20022 Acquirer to Issuer Card Messages (ATICA) User Guide

ISO/TC 068/SC 09/TG 01 "Card Standards"

Change log

Date	Who	Version	Main changes / remarks
18/2/2022	MUG TEAM	1.0	Initial version for ATICA v2, for handing over to TG1
25/4/2022	MUG TEAM	1.1	Typo correction
24/6/2022	TG1	1.0	Revision for ATICA v3
22/07/2022	MUG TEAM	1.1	UKPT 3DES and AES encrypted PIN Block examples
13/09/2022	TG1	1.1	Revision for ATICA v3
23/09/2022	MUG TEAM	1.2	Final revision

Content

1 Introduction

1.1 Purpose and Use of this Message Usage Guide (MUG)

This MUG refers to ATICA version 3.

The present MUG outlines the way to use the ATICA Card Payments messages in the global context of Acquirer-to-Issuer transactions within the related ISO 20022 Business Area. It provides a comprehensive view on how these messages fit within a card payment business process and the activities of the involved parties. It contains detailed explanations and examples on the use of the message components to convey specific information related to these processes and activities. This guide acts as a complement to the Message Definition Report and the XML Schema for those exchanges.

1.2 Intended audience

The present guide is intended for implementers of ATICA messages.

1.3 Identification of Institution and routing

Acquirer to card issuer

Initiation			
Institution	**Acquirer to A**	**A to B**	**B to card issuer**
Initiating	Acquirer Endpoint ID	A Endpoint ID	B Endpoint ID
Recipient	A Endpoint ID	B Endpoint ID	Card issuer Endpoint ID
Acquirer	Remains the same throughout the transaction		
Card issuer	Remains the same throughout the transaction		
Forwarding	[a]	A	B
Receiving	A	B	[b]
Response			
Institution	**Card Issuer to B**	**B to A**	**A to Acquirer**
Initiating	Card issuer Endpoint ID	B Endpoint ID	A Endpoint ID
Recipient	B Endpoint ID	A Endpoint ID	Acquirer Endpoint ID
Acquirer	Remains the same throughout the transaction		
Card issuer	Remains the same throughout the transaction		
Forwarding	[b]	B	A
Receiving	B	A	[a]

[a] Not used when the forwarding or receiving institution is the acquirer.

[b] Not used when the receiving or forwarding institution is the card issuer.

Card issuer to Acquirer

Initiation			
Institution	**Card issuer to A**	**A to B**	**B to Acquirer**
Initiating	Acquirer Endpoint ID	A Endpoint ID	B Endpoint ID
Recipient	A Endpoint ID	B Endpoint ID	Card issuer Endpoint ID
Acquirer	Remains the same throughout the transaction		
Card issuer	Remains the same throughout the transaction		
Forwarding	[a]	A	B
Receiving	A	B	[b]
Response			
Institution	**Acquirer to B**	**B to A**	**A to Card issuer**
Initiating	Card issuer Endpoint ID	B Endpoint ID	A Endpoint ID
Recipient	B Endpoint ID	A Endpoint ID	Acquirer Endpoint ID
Acquirer	Remains the same throughout the transaction		
Card issuer	Remains the same throughout the transaction		
Forwarding	[b]	B	A
Receiving	B	A	[a]

[a] Not used when the forwarding or receiving institution is the card issuer.
[b] Not used when the receiving or forwarding institution is the acquirer.

Transaction origination to transaction destination

Initiation			
Institution	**Originator to A**	**A to B**	**B to Destination**
Initiating	Acquirer Endpoint ID	A Endpoint ID	B Endpoint ID
Recipient	A Endpoint ID	B Endpoint ID	Card issuer Endpoint ID
Acquirer	Remains the same throughout the transaction		
Forwarding	[a]	A	B
Receiving	A	B	[b]
Originator	Remains the same throughout the transaction		
Destination	Remains the same throughout the transaction		
Response			
Institution	**Destination to B**	**B to A**	**A to Originator**
Initiating	Card issuer Endpoint ID	B Endpoint ID	A Endpoint ID
Recipient	B Endpoint ID	A Endpoint ID	Acquirer Endpoint ID
Acquirer	Remains the same throughout the transaction		

Forwarding	[b]	B	A
Receiving	B	A	[a]
Originator	Remains the same throughout the transaction		
Destination	Remains the same throughout the transaction		

[a] Not used when the forwarding or receiving institution is the acquirer.

[b] Not used when the receiving or forwarding institution is the card issuer.

2 Message Organisation

In ISO 20022, the message definition is the message format as defined in MDR Part 2 of ATICA as for all other ISO 20022 specifications. Referring to a message actually means one instance of the message definition actually transmitted.

2.1 Business Areas

Business Areas are used to classify ISO 20022 Message Definitions for different types of businesses covered by ISO 20022. They are identified by a four-character code used in the Message Identifier of the related ISO 20022 Message Definitions.

The below Business Areas have been retained for the definition of messages related to ATICA.

https://www.iso20022.org/sites/default/files/documents/D7/ISO20022_BusinessAreas.pdf

Business Area Name	Business area code	Description
Acquirer to Issuer Card Transactions	cain	Messages that support any card payment related transactions and services between a card transaction acquirer and a card issuer. It includes the authorisation, reversal and financial presentment of card transactions.
Card Administration	caad	Messages that support any card related administrative services between financial institutions and their agents.
Fee Collection	cafc	Messages that support the reporting and advising of card payment transactions, including the collection of fees and processing of charge-backs.
Network Management	canm	Messages that support network management services in a card payment environment between financial institutions and/or their agents.
File Management	cafm	Messages that support file management services in a card payment environment between financial institutions and/or their agents

2.2 Message Structure

A common message structure is used for all ATICA messages, and is composed of three major 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 from a business viewpoint
  - Environment
  - Context Transaction
  - Addendum Data
  - Processing Result
  - ICC related data
  - Protected Data
  - Supplementary Data
- a **Security Trailer** containing information related to the security aspects of the message (optional).

2.3 Message tracking information

In the message header, there are message tracking information allows an entity to monitor the upstream transport and process of a message.

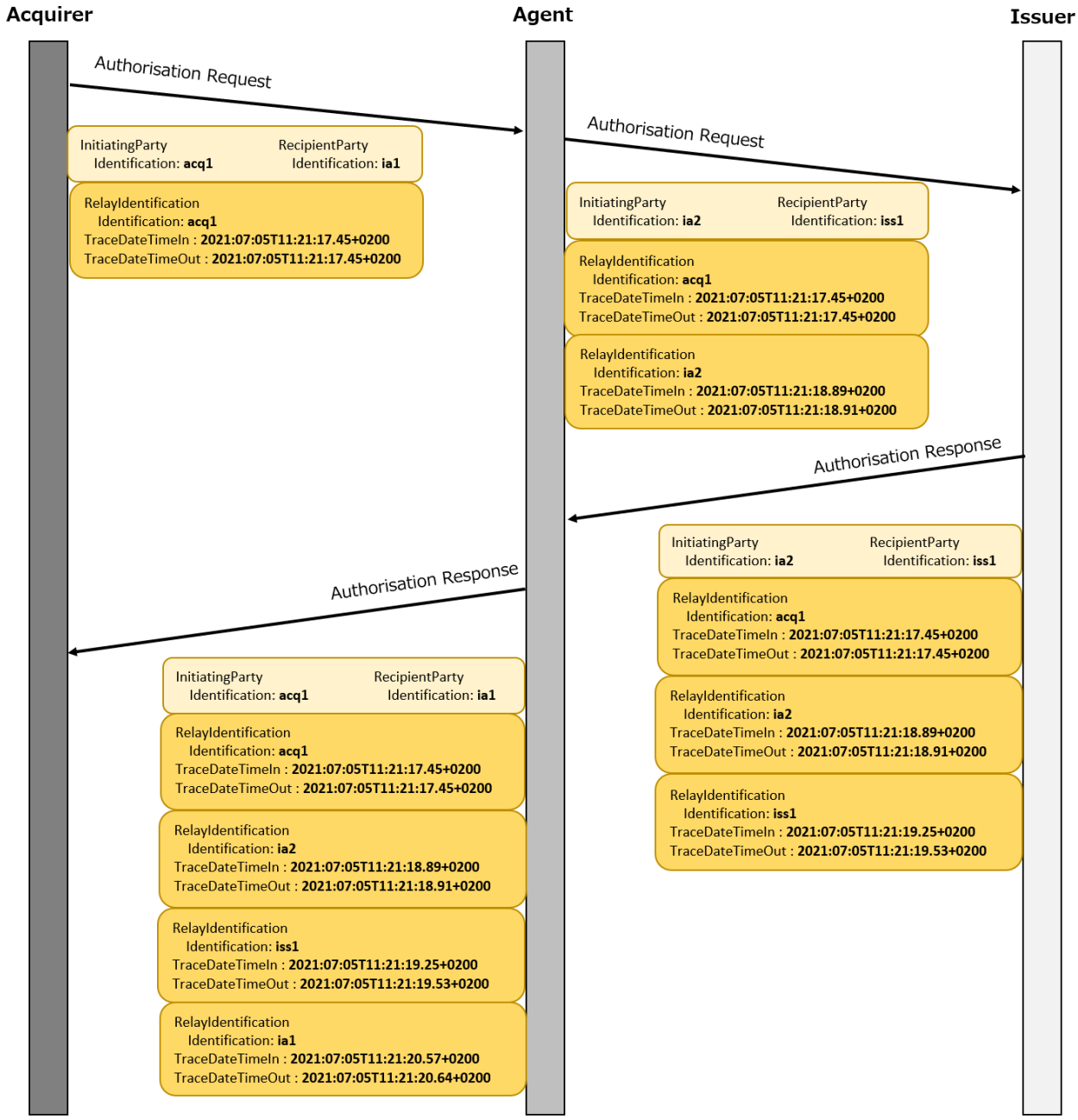
The **TraceData** element is a component allowing an entity receiving the message to monitor the progression of it along its whole routing and processing path.

The **Traceability** component provides details to trace the path of the message through its successive processing steps. 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 forwarding the message to the next party in the chain.

The value and interest of using such type of information relies essentially on the discipline of each intermediary in the message exchange chain to provide the relevant information when passing the message to the next recipient of the 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.



* 「Authorisation Request」 means the AuthorisationInitiation with the message function “Request”.
「Authorisation Response」 means the AuthorisationResponse with the message function “Request”.

Traceability may start with the original initiator of the message (e.g. Acceptor, Acquirer) and ultimately end up with the same party including all intermediaries steps in the path of the message. When receiving back this information, the initiator of the message should be in a position to gather the whole history of the transaction.

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

As a general rule, each party receiving a message where the Traceability component was provided upstream should add its own traceability information to avoid interrupting the whole traceability chain. This would also enable each party in the chain to continue monitoring all messages pertaining to the whole lifecycle of the transaction.

The recipient will update traceability when a message is received containing trace information by adding a new trace entry with TraceTimeIn 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.

2.4 Supplementary data

The supplementary data structure is meant for extension of the messages by specific communities.

The general description of the supplementary data is on the ISO20022 website:
https://www.iso20022.org/catalogue-messages/additional-content-messages/supplementary-data

Currently there is an SCA extension registered : https://www.iso20022.org/sites/default/files/2020-11/ISO20022_StrongCustomerAuthentication_SupplementaryData_v1.zip, for use in Europe following the EBA (European Banking Authority) requirements.

3 Common functionality and building blocks

This chapter introduces the functionality which is common across the ATICA messages. Specifics of the particular messages are covered in the relevant chapters.

3.1 Optional data elements

Following ISO 20022 rules, ATICA implements only mandatory and optional data elements. There is no concept of conditional data elements, however this document suggests cases when particular component should be populated depending on the use case, e.g. echoing elements from an initiation message to the response message.

3.2 National, private data elements

ATICA does not implement data element or code ranges reserved for national or private use. This purpose is served by either the Additional Data component or the Supplementary Data structure .

3.3 Code Sets

3.3.1 Internal codes

Pre-defined (e.g. Message function, Transaction attribute)

Alternate (e.g. Other Private, Other Type)

3.3.2 External codes

ATICA external code list : ExternalAuthenticationMethod1Code

3.3.3 Other standards codes

Message reasons, Country codes, Currency codes, ExternalEncryptedElementIdentification1Code etc.

3.4 Message Function

List of values

The type of the message is identified by the value of the element **MessageFunction**, inside the **Header** component.

- Request
- Advice
- Notification
- Capture Advice: refer to MDR part1, Acquirer capture
- Capture Notification: refer to MDR part1, Acquirer capture
- Status Request
- Status Advice: refer to MDR part1, ChargeBack message

- Status Notification
- Error Notification: refer to MDR part1, Error message
- Reject Notification: refer to MDR part1, Error message
- Reversal Advice
- Reversal Notification

In the Initiation-Response message pair, the message function remains the same.

3.5 LifecycleTraceIdentifier's usage

> Transaction life cycle identification data is a unique identifier used to match transactions across message classes, e.g. authorization to financial presentment or financial presentment to chargeback. It shall contain the same value in all message classes throughout a transaction's life cycle.

Acquirers and Issuers must be able to receive this identifier in all messages in which it is present.

Case A:

> N Authorisations for 1 Presentment (for example aggregation of transactions): the **AuthorisationSequenceNumber** identifies an authorisation in a sequence of different authorisations leading to a single financial presentment; its value is incremented for each authorisation in that sequence. For matching purposes, the **AuthorisationSequenceNumber** of the financial presentment may be populated with the value of the last authorisation sequence number.

Case B: 1 Authorisation for N Presentments

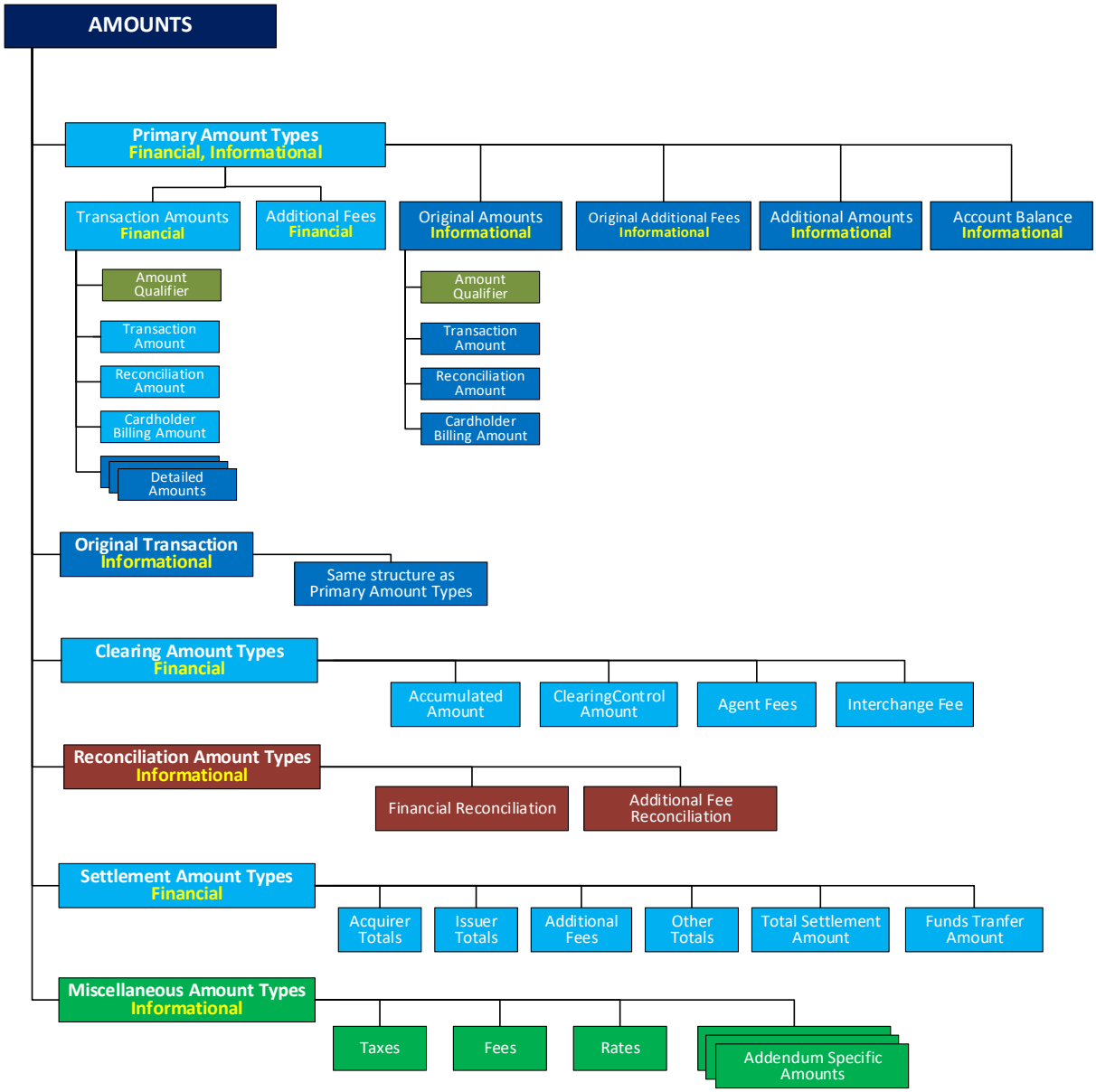
> The **PresentmentSequenceNumber** identifies a financial presentment in a sequence of different presentments related to a single authorisation; its value is incremented for each presentment in that sequence.

4 Amounts

This chapters discusses the different types of amounts and their usage as well as amounts relationship and usage of the currencies. The list is not exhaustive.

The different amounts are depicted in the diagram below.

4.1 Overview diagram



In general, the *primary* amount types as well as *clearing*, *reconciliation* and *settlement* are part of *Transaction* <*Tx*> component.

Some amounts components are also present in non-financial messages but only for informational purposes.

The amounts depicted as *financial* have a financial business impact on the parties and are part of the clearing and settlement processes. These amounts are also subject for reconciliation, but the reconciliation amounts have no financial impact themselves.

4.2 Relationship between financial impact amounts

Relationship between different amount fields and how they are used to calculate reconciliation and settlement totals is described in the table here below. The amount components that are not included in this table will not be part of the Financial Reconcliliation totals nor Settlement totals.

Amount type	Relationship
TransactionAmount TxAmts/<TxAmt>/<Amt>	This amount present in the messages with financial impact (Financial, Reversal, Chargeback) is used to populate the financial reconciliation totals and the settlement totals.
DetailedAmount TxAmts/<DtldAmt>/<Amt>	This amount is already included in Transaction Amount. This amount is not used to calculate the reconciliation totals nor in the settlement totals.
ReconciliationAmount TxAmts/<RcncltnAmt>/ <Amt>	The TransactionAmount/Amount set in the individual messages that have financial impact (Financial, Reversal, Chargeback) are used to calculate the reconciliation amount. The aggregation of these amounts will be set in the Reconciliation message.
AdditionalFees <AddtlFees>/<Amt>	Additional fees are not part of the TransactionAmount. Fees can be set for each individual message. The fees are directly exchanged between parties, they are included in the Settlement. In version 2, the fees are not aggregated in the Additional Fee Reconciliation part of the Reconciliation message.

Note: Reconciliation and settlement messages contain total positive, negative, and net amounts per currency.

Primary Amount Types

The class of amounts related to the transaction amount, bearing financial impact for the card account.

4.3 General concepts

4.3.1 Amount format

Format	Digits
minInclusive	0
totalDigits	18
fractionDigits	5

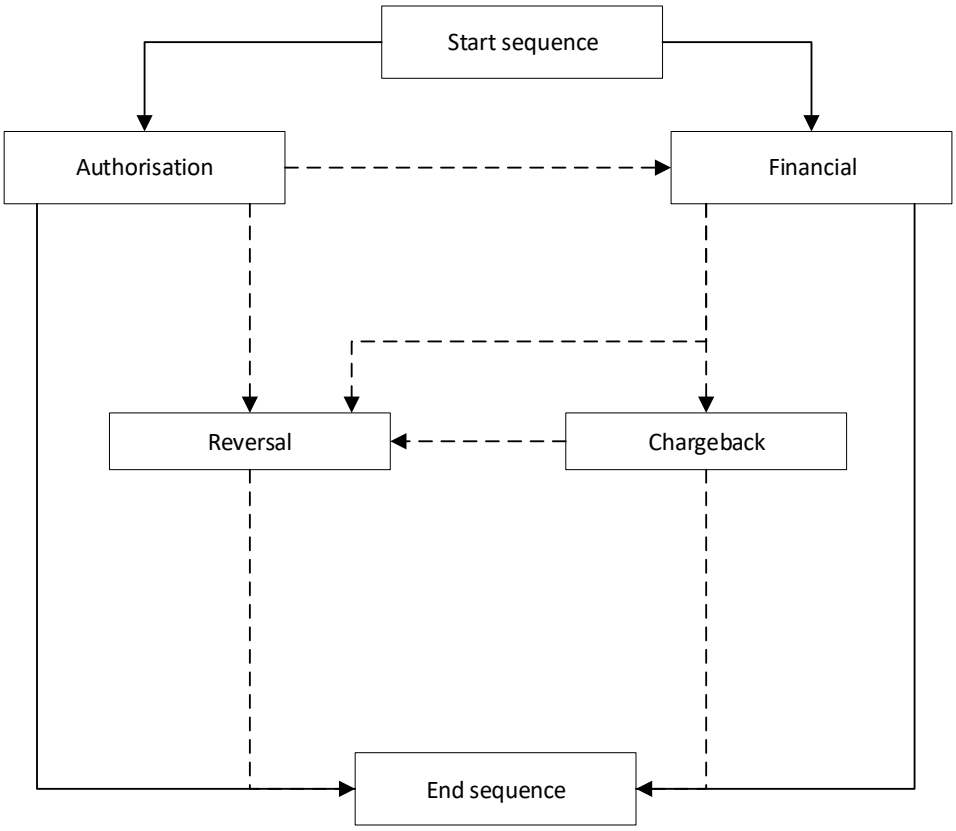
Valid amounts:
The trailing zeros are not populated, e.g.:
Amount of 500 will be represented as 500.0,
Amount of 500.100 will be represented as 500.1,
Amount of 500.10001 will be represented as 500.10001,
Amount of 1234567890112.3456(7)

Invalid amounts:
Amount of 500.000001 is not allowed.
Amount of -500.0 is not allowed.
Amount of 12345678901123.3456(7)

4.4 Amounts in card payment messages

4.4.1 Overview diagram

ATICA MDR Part1 describes several payment scenarios. The general process of the payment transaction is represented by the diagram below:



Usage of amount components in the scenarios resulting from this process (e.g. partial approval or partial reversal) requires populating both the *TxAmts.TxAmt* and *OrgnlTxAmts.TxAmt* as explained in the tables below:

4.4.2 Amounts in AuthorisationInitiation / AuthorisationResponse message

Response messages echo the amounts from request messages, except for the cases of partial approval or decline.

Amount description	Transaction Amounts/Amount Qualifier	Transaction Amounts/Transaction Amount/Amount	Transaction Amounts/Original Transaction Amounts/Transaction Amount/ Amount
AuthorisationInitiation			
Original (first authorisation)	N/A, DFLT, ESTM, MAXI, PRXY, DPST, RESD	Transaction amount	--
Original (preauthorisation)	N/A, DFLT, MAXI, PRXY	Transaction amount	--
Incremental	N/A, DPST, RESD, ESTM	incremental amount	Originally authorised amount
Resubmission (repeat)	Same as original	Transaction amount	--
AuthorisationResponse			
Full approval	Echo	Originally requested amount	--
Partial approval	Echo	Approved amount	Originally requested amount
Decline/reject	Echo	Originally requested amount	
Proxy management: The request contains AmountQualifier = Proxy Amount =1	Proxy	Actual value defined for the specific market	Originally requested amount = 1

4.4.3 Amounts in FinancialInitiation / FinancialResponse message

Amount description	Transaction Amounts/Amount Qualifier	Transaction Amounts/Transaction Amount/Amount	Transaction Amounts/Original Transaction Amounts/Transaction Amount/ Amount
FinancialInitiation			
First presentment	ACTL	Transaction amount	
Resubmission (repeat)	Same as original	Transaction amount	--

FinancialResponse			
Full approval	Echo	Originally requested amount	--
Partial approval	Echo	Approved amount	Originally requested amount
Decline/reject	Echo	Originally requested amount	

4.4.4 Amounts in Reversal messages

AmountQualifier is the echo of the one from the original transaction.

Type of reversal	**Transaction Amounts/Amount Qualifier**	**Transaction Amounts/Transaction Amount/Amount**	**Transaction Amounts/Original Transaction Amounts/Transaction Amount/ Amount**
Full	N/A or Any	Amount to be fully reversed	Originally authorised amount including any incremental amounts
Partial	N/A or Any	Amount to be partially reversed	Originally authorised amount including any incremental amounts

4.4.5 Amounts in Chargeback message

Type of chargeback	**Transaction Amounts/Amount Qualifier**	**Transaction Amounts/Transaction Amount/Amount**	**Transaction Amounts/Original Transaction Amounts/Transaction Amount/ Amount**
Full	ACTL	Amount to be fully charged back	Original presentment amount
Partial	ACTL	Amount to be partially charged back	Original presentment amount

5 Authorisation Message

5.1 Authorisation description

An authorisation is an approval or guarantee of funds given by the card issuer to the acquirer. The acquirer seeks authorisation from the card issuer or advises the card issuer of authorisation already given, by means of the authorisation message. Authorisation is not intended to permit the application of the approved transaction amount to the card holder's account for billing or posting.

The authorisation flow includes Initiation and Response messages.

AuthorisationInitiation is used by Acquirers or the Agent to request approval of the transaction, or to convey the result of stand-in process as advice.

AuthorisationResponse is used by Issuers, or the Agent, to give the result of the approval processing.

a) The following types of authorisations are defined (not exhaustive):

1) original authorisation, i.e. the first or only authorisation.
2) incremental authorisation, i.e. when one or more previous authorisations were approved and a further authorisation is required for an additional amount.
3) Delayed Authorisation, when an authorisation was not possible at the time of the transaction although the card was read and the card details are captured and the authorisation postponed. (e.g. In-flight purchases, cruise ships)

   …Context/PointOfServiceContext/DelayedAuthorisationIndicator

b) The following types of authorisation decisions are defined;

1) full approval, i.e. where the card issuer indicates approval of the requested amount.
2) partial approval, i.e., where the card issuer indicates approval of an amount less than the originally requested amount.
3) declined or rejected, i.e. where the request for approval is declined or the authorisation request message is rejected.

5.2 Authorisation Message Rules

The following applies to all authorisation messages.

a) An authorisation initiation message (*Request* at *MessageFunction* in *AuthorisationInitiation*) is sent used when the process cannot complete at the point of service until the authorisation response message (*Request* at *MessageFunction* in *AuthorisationResponse*) is received indicating the action to be taken.

b) An authorisation response message (*Request* at *MessageFunction* in *AuthorisationResponse*) is sent in response to an authorisation initiation message (*Request* at *MessageFunction* in *AuthorisationInitiation*). It indicates the approval or guarantee of funds or the action to be taken as specified in the *Action* data component.

c) An authorisation initiation message (*Advice* at *MessageFunction* in *AuthorisationInitiation*) is sent to advise the card issuer of an authorisation that has completed at the point of service.

d) An authorisation response message (*Advice* at *MessageFunction* in *AuthorisationResponse*) is sent in response to an authorisation advice initiation message to adjust the open-to-buy.

e) An authorisation initiation message (*Notification* at *MessageFunction* in *AuthorisationInitiation*) is sent to notify the card issuer to adjust the open-to-buy.

5.3 Authorisation Message Type Identifiers

Underneath table defines the message type identifiers that may be used in an authorisation transaction.

Message	Purpose	Message Definition	Message Type	From	To
Authorisation request initiation	Requests an authorisation	AuthorisationInitiation	Request	Acquirer	Issuer
Authorisation request response	Carries the answer to an authorisation request message	AuthorisationResponse	Request	Issuer	Acquirer
Authorisation advice initiation	Advice of an authorisation carried out on behalf of the card issuer	AuthorisationInitiation	Advice	Acquirer	Issuer
Authorisation advice response	Carries the answer to an authorisation advice message	AuthorisationResponse	Advice	Issuer	Acquirer
Authorisation notification initiation	Notifies of an authorisation action	AuthorisationInitiation	Notification	Acquirer	Issuer
Authorisation captured advice initiation	Advice of an authorisation processed by an Agent Forwards the result of the authorisation by the Issuer	AuthorisationInitiation	CaptureAdvice	Agent	Acquirer
Authorisation captured advice response	Carries the answer to an authorisation advice message	AuthorisationResponse	Agent	Acquirer	Agent
Authorisation captured notification initiation	Notifies of an authorisation processed by an Agent Forwards the result of the authorisation by the Issuer	AuthorisationInitiation	CaptureNotification	Agent	Acquirer

6 Financial messages

6.1 Financial message description

A financial message permits the application of the approved transaction amount to the cardholder's account for billing or posting. The acquirer seeks the approval from the card issuer, by means of the financial message.

The Financial flow includes Initiation and Response messages.

FinancialInitiation is used by an Acquirer or an Agent to an Issuer or an Agent to request approval of the transaction, or to convey the result of stand-in process as advice.

FinancialResponse is used by an Issuer or an Agent to an Acquirer or an Agent, to give the result of the approval processing.

a) The following types of financial message are defined;

1) First presentment either previously authorized or not, i.e. original or only financial presentment.
2) Representment, i.e. to partially, or to wholly, recover funds previously charged back by the card issuer.

b) The following types of financial message decisions are defined:

1) full approval, i.e. where the response from the card issuer indicates approval of the originally requested amount.
2) partial approval, i.e. where the card issuer indicates approval of an amount less than the originally requested amount.
3) declined or rejected, i.e. where the request for approval is declined or the financial request message is rejected.

6.2 Financial Message Rules

The following applies to all financial messages.

a) A financial request message (*Request* at *MessageFunction* in *FinancialInitiation*) is sent when the process cannot complete at the point of service until the response message is received indicating the action to be taken.

b) A financial request response message (*Request* at *MessageFunction* in *FinancialResponse*) is sent in response to a financial request message (*FinancialInitiation*). A financial request response message (*FinancialResponse*) indicates the approval or guarantee of funds or the action to be taken as specified in the *Action code* data element.

c) A financial advice message (Advice at MessageFunction in FinancialInitiation) is sent to inform the card issuer of a financial presentment that has completed at the point of service.

d) A financial advice response message (Advice at MessageFunction in FinancialResponse) is sent in response to a financial advice message (Advice at MessageFunction in FinancialInitiation).
A financial advice response message (Advice at MessageFunction in FinancialResponse) indicates if the card issuer accepts or rejects the transfer of financial liability.

e) A financial notification message (*Notification* at *MessageFunction* in *FinancialResponse*) is sent to inform the card issuer of a financial presentment that has completed at the point of service.

7 Reversal messages

7.1 Reversal message description

A reversal is the partial or complete nullification of the effects of a previous authorisation, financial message, chargeback message that cannot be processed as instructed, i.e. is undeliverable, is cancelled or the acquirer times out waiting for a response.

The following applies to all reversals.

a) A reversal shall not be reversed.

b) Authorisation, Financial messages can be reversed.

The Reversal flow includes Initiation and Response messages.

ReversalInitiation is used by Acquirers or the Agent to initiate the reversal of the transaction to convey the result of cancellation, time out or etc.

ReversalResponse is used by Issuers or the Agent, to give the result of the reversal processing.

7.2 Reversal Message Rules

Reversals shall use the request, advice, capture advice, capture notification or notification messages as message functions since the activity has already occurred.

The following applies to all reversal messages;

a) A reversal request message should be sent, i.e. when a card holder requests to cancel a previous transaction.

b) A reversal advice message (*Advice* at *MessageFunction* in *ReversalInitiation*) or reversal notification message (*Notification* at *MessageFunction* in *ReversalInitiation*) is sent by an acquirer whenever a previous authorisation, financial messages, etc., cannot be processed as instructed, i.e., is undeliverable, is cancelled, or the acquirer times out waiting for a response.

   1) the Message reason code data element is used to indicate the reason for the reversal.

   2) the Transaction Amount data element in a reversal advice or notification message contains the amount to be reversed that shall be less than or equal to the original Amount transaction.

   3) the TransactionType shall be the same as presented in the original request or advice message. If the original request or advice message was a debit, the reversal also indicates debit. if the original request or advice message was a credit, the reversal also indicates a credit.

c) A reversal advice response message (*Advice* at *MessageFunction* in *ReversalResponse*) is sent in response to a reversal advice message (*Advice* at *MessageFunction* in *ReversalInitiation*). A reversal advice shall not be declined except for specific reasons.

8 Retrieval and Retrieval Fulfilment messages

8.1 Retrieval and Retrieval Fulfilment message description

A retrieval is an activity (by the card acceptor, acquirer, or relevant agent) needed to support a card issuer who has determined that a transaction information needs to be examined before a potential chargeback is sent or to satisfy another need of the card issuer or cardholder. The *Message reason code* is used to indicate the specific reason for the retrieval. Only a card issuer can send a retrieval request.

RetrievalInitiation is used by an Issuer to request the transaction information.

RetrievalResponse is sent in response to a retrieval request advice by an Acquirer, or the Agent.

A retrieval fulfilment is where an acquirer has successfully retrieved the requested information or where the reason it was not provided is advised. The Function code is used to indicate the result to the card issuer. Only an acquirer can send a retrieval fulfilment.

RetrievalFulfilmentInitiation is sent by an Acquirer to answer if the requested information can be provided.

RetrievalFulfilmentResponse is sent in response to a retrieval fulfilment advice by an Issuer or an Agent.

The following applies to all chargebacks.

a) A card issuer initiates a *RetrievalInitiation* with a certain reason(s).

b) A card issuer may request to retrieve the information to show that the original transaction is absolutely correct. The cases that a card issuer can request it are defined by each scheme or in a bilateral agreement between the issuer and the acquirer.

c) When an acquirer receives a retrieval message with the *Message Type* as "*Advice*", the acquirer sends the response to the issuer in *RetrievalResponse*.

d) The acquirer can initiate *RetrievalFulfilmentInitiation* to indicate whether or not the acquirer has successfully retrieved the requested information.

e) When the issuer receives a retrieval fulfilment message with the *Message Type* as "*Advice*", the issuer sends the response to the acquirer in *RetrievalFulfilmentResponse*.

8.2 Retrieval and Retrieval Fulfilment message Rules

The following applies to *Retrieval* messages when used for retrieval.

a) A *RetrievalInitiation* message with "*Advice*" or "*Notification*" in *Message Type* is sent when the card issuer requires sight of a transaction information document.

b) An *RetrievalResponse* message is sent in response to an *RetrievalInitiation* message with "*Advice*" in *Message Type*. The *RetrievalResponse* message indicates receipt of the receipt of the *RetrievalInitiation* message. It is not a retrieval fulfilment.

The following applies to *RetrievalFulfilment* messages when used for retrieval fulfilment.

c) An *RetrievalFulfilmentInitation* message is sent by the acquirer to indicate the results of the retrieval fulfilment.

d) An *RetrievalFulfilmentResponse* message is sent by the card issuer in response to a *RetrievalFulfilmentInitiation* message only if "*Advice*" is set in *Message Type*.

9 Chargeback messages

9.1 Chargeback message description

A chargeback is the partial or complete nullification of a previous financial presentment or financial accumulation presentment when the card issuer determines that a customer dispute exists, or that an error or violation of rules has been committed.

The Chargeback flow includes Initiation and Response messages.

ChargebackInitiation is used by an Issuer to convey the result of charged back by a cardholder or an Issuer.

ChargebackResponse is sent in response to a chargeback advice message by an Acquirer, or the Agent.

The following applies to all chargebacks.

c) A card issuer initiates a chargeback with a certain reason(s).

d) A chargeback can be generated only if the original transaction had financial impact on the cardholder's net position. A chargeback shall not be used to cancel a balance inquiry, account transfer and authorisation.

e) To cancel, either partially or completely, a previous chargeback that was submitted in error, the card issuer shall initiate a subsequence chargeback containing *Reversal Advice* or *Reversal Notification* at *Message Function* in *Header* and Original Transaction information of the chargeback transaction to cancel in *Original Transaction Structure* to point the previous chargeback.

f) If the transaction that is being charged back requires a response, this response message can be sent before the chargeback transaction is generated.

g) A card issuer may chargeback an original transaction plus any subsequent representment(s) submitted by the acquirer. A separate chargeback transaction can be used for each.

h) This part of ATICA specifies no limits on the time frame or the number of chargebacks and representments that may be exchanged between an acquirer and a card issuer.

9.2 Chargeback message Rules

Chargeback uses the advice or notification message since the activity has already occurred. The following applies to all chargeback messages.

a) A chargeback advice message or chargeback notification message, is sent when the card issuer determines that a chargeback is appropriate and valid, in which case:

   1) the Message reason code data element is used to indicate the reason for the chargeback.

   2) the Amount transaction data element in chargeback advice or notification message can be the amount to be charged back and shall be less than or equal to the Amount of the previous presentment.

b) A chargeback advice response message is sent in response to a chargeback advice message. A chargeback shall not be declined except for specific reasons, although the acquirer may represent the original transaction.

c) When the chargeback data is an error, the status will be shown in the *Processing Result*, and detailed error information will be sent separately as an ***Error Message***.

Amount in chargeback message:

Type of chargeback	Amount transaction	Original amount transaction
Full	Amount charged back	-
Partial	Amount charged back	Original transaction amount

10 Error messages

10.1 Error messages description

Error messages allow for the details of specific errors to be returned to the message sender in Error message or Error details component in the corresponding Response message.

Error messages are also used in case of a rejection of a message

10.2 Error messages message rules

The following applies to all Error messages.

a) An Error notification message shall be sent in response to any previously submitted message:

1) the message function shall be Error notification if the error condition results in the notification of a message containing errors. The error contents can be either detailed in the Error detail component within the Error message itself or in the Processing result component of the corresponding Response message. It shall contain the invalid data and/or any additional information that will assist in correcting the error condition.

2) the message function shall be Reject notification if the error condition results in the rejection of a message. The error contents can be either detailed in the Error detail component within the Error message itself or in the Processing result component of the corresponding Response message. It shall contain the copy of the original message in the Original Message component that is rejected and if available the invalid data and/or any additional information that will assist in correcting the error condition.

11 Batch Management message

11.1 Batch Management description

Batch management allows transaction details to be sent as a series of messages. Control is maintained by the Batch Management type, which may be sent periodically, within the transmission of a batch.

The same technique can be employed between the card issuer and acquirer where the card issuer is required to send individual messages, such as chargeback messages.

11.2 Batch Management rules

The following applies to batch management messages.

a) To perform batch management, the requestor must send a BatchManagementInitiation with BatchManagementType=STRT and receive a BatchManagementResponse before initiating the batch transmission.

b) Series of messages

   1) A unique message sequence number of the sub-elements is provided by the Message Sequence Number.

   2) The BatchIdentification data within the message provides a unique identifier for the batch

c) To complete batch management, the requestor must send a BatchManagementInitiation with BatchManagementType=ENDB and receive a BatchManagementResponse to confirm completion of the batch transmission

12 Reconciliation

12.1 Reconciliation process

Reconciliation is the exchange of totals and/or counts of messages within a specific session between two interchanging parties (Acquirer, Issuer or Agent) to reach agreement on financial totals or message counts. Any of those parties can initiate the exchange. The reconciliation process is crucial for ensuring completeness and integrity of the transaction exchanges. The standard however does not define a process for resolution of the differences between the parties.

12.2 Reconciliation message rules

Below table provides a summary of the combination between the Message function, Reconciliation Function and Reconciliation Type:

Message Function	Name	Reconciliation Function	Name	ReconciliationType	
REQU	Request	INQR	Inquire	PART CheckPoint	FINL Final
		INCU	InitiateCutover		
ADVC	Advice	CNVY	Convey		
		INCU	InitiateCutover		
		INQR	Inquire		
NOTI*	Notification	CNVY	Convey		

The following applies to all reconciliation messages.

Two types of reconciliation are defined.

a) A checkpoint reconciliation shall be indicated by the Reconciliation type *Check Point*
A checkpoint reconciliation period shall be identified with the *CheckpointReference*. Any message initiated after completion of the message indicating a checkpoint shall contain the new *CheckpointReference*.

b) A final reconciliation shall be indicated by the Reconciliation type *Final*

An authorisation request initiation message (*Request* at *MessageFunction* in *AuthorisationInitiation*)

a) A reconciliation initiation message (*Request/Advice* at *MessageFunction*, *Inquire* at *ReconciliationFunction*), shall be sent to request reconciliation totals (number and value).

b) A reconciliation response message (*Request/Advice* at *MessageFunction*, *Inquire* at *ReconciliationFunction*) shall be sent in response to a reconciliation initiation message. A reconciliation response message shall contain the requested totals, if available, and shall indicate one of the following results:

1) totals provided, i.e. all amounts and number data elements shall be returned with the values from the institution sending the reconciliation response message (*Request/Advice* at *MessageFunction*, Inquire at *ReconciliationFunction*);
2) totals not available, i.e. all amount and number data elements shall be returned with zero values;

3) the totals shall be used to indicate the originating institution's position as either acquirer or card issuer (but not both) as defined by the message type identifier.

c) A reconciliation initiation message (*Advice* at *MessageFunction*, Convey at *ReconciliationFunction*) shall be sent to ask for a confirmation of totals (number and value).

d) A reconciliation response message (*Advice* at *MessageFunction*, *Convey* at *ReconciliationFunction*) shall be sent in response to a reconciliation initiation message (*Advice* at *MessageFunction*) and shall indicate one of the followingresults:

1) reconciled, in balance, i.e. only the Amount net reconciliation data element shall be returned in the reconciliation response message (*Advice* at *MessageFunction, Convey* at *ReconciliationFunction*));

2) reconciled, out of balance, i.e. all amount and number data elements shall be returned with the values from the institution sending the reconciliation response message (*Advice* at *MessageFunction, Convey* at *ReconciliationFunction*);

3) totals not available, i.e. all amount and number data elements shall be returned with zero values.

e) A reconciliation initiation message (*Notification* at *MessageFunction*, *Convey* at *ReconciliationFunction*) shall be used to provide totals (number andvalue). The totals contained in the reconciliation initiation message (*Notification* at *MessageFunction*, *Convey* at *ReconciliationFunction*) shall indicate an originating institution's position as either an acquirer or card issuer (but not both) as defined by the message type identifier. The reconciliation initiation message (*Notification* at *MessageFunction*, *Convey* at *ReconciliationFunction*) may require acknowledgement.

f) A reconciliation initiation message (*Request/Advice* at *MessageFunction*, *Initiate Cutover* at *ReconciliationFunction*), shall be sent to request reconciliation totals (number and value).

g) A reconciliation response message (*Request/Advice* at *MessageFunction*, *Initiate Cutover* at *ReconciliationFunction*) shall be sent in response to a reconciliation initiation message.

12.3 Reconciliation Amounts calculation overview

Chargebacks have a financial impact and should be computed within reconciliation totals. Specific rules for amounts calculation may be covered in community or bilateral MIG.

12.3.1 Calculation of reconciliation totals with financial impact for *ReconciliationTotals/FinancialReconciliation*

All amounts in the reconciliation messages are in the currency of reconciliation.

If the currency of reconciliation is different from that of the transaction, then the reconciliation amounts and fees shall be populated for each currency separately.

Amounts from individual *FinancialInitiation*, *ReversalInitiation*, *ChargebackInitiation* shall be grouped separately according to each category type (*Financial*, *Reversals* and *Chargebacks* and other types which are agreed bilaterally).

The fees reconciliation is performed separately (see “Reconciliation amount calculation” chapter below).

A repeat message is not added to reconciliation totals unless the original message was not received.

Transaction Type = 00 to 1Z means that the issuer (cardholder) account is debited, Transaction Type = 20 to 2Z means that the the issuer (cardholder) account is credited.

12.3.1.1 Financials reconciliation

Procedure for calculation of *TotalAmount* of financial messages:

- *FinancialReconciliation Amount with Type = FNCL and ReconciliationImpact = DEBT* is populated with the total sum of:

FinancialInitiation.ReconciliationAmount.Amount where Transaction Type = 00 to 1Z

- *FinancialReconciliation Amount with Type = FNCL and ReconciliationImpact = CRDT* is populated with the total sum of:

  FinancialInitiation.ReconciliationAmount.Amount where Transaction Type = 20 to 2Z

12.3.1.2 Reversals reconciliation

Procedure for calculation of reversals reconciliation total amount:

- *FinancialReconciliation.Amount with FinancialReconciliation.Type = RVSL and FinancialReconciliation.ReconciliationImpact = CRDT* is populated with the total sum of:

  ReversalInitiation.ReconciliationAmount.Amount where Transaction Type = 00 to 1Z

- *FinancialReconciliation.Amount with FinancialReconciliation.Type = RVSL and FinancialReconciliation.ReconciliationImpact = DEBT* is populated with the total sum of:

  ReversalInitiation.ReconciliationAmount.Amount where Transaction Type = 20 to 2Z

12.3.1.3 Reconciliation of presentment Chargebacks and reversal chargebacks

Procedure for calculation of chargebacks reconciliation total amount: for this example of calculation chargeback of a presentment and reversal of a chargeback are grouped together.

- *FinancialReconciliation.Amount with FinancialReconciliation.Type = CGBK and FinancialReconciliation.ReconciliationImpact = CRDT is populated with the sum of:*

  (*ChargebackInitiation.ReconciliationAmount amounts where Transaction Type = 00 to 1Z*

  PLUS

  ChargebackInitiation (Message Function = RVRA (Reversal Advice) or RVNO (Reversal Notification))ChargebackInitiation.ReconciliationAmount amounts where Transaction Type = 20 to 2Z)

- *FinancialReconciliation.Amount with Type = CGBK and FinancialReconciliation.ReconciliationImpact = DEBT* is populated with the total sum of:

  (*ChargebackInitiation.ReconciliationAmount amounts where Transaction Type = 20 to 2Z*

  PLUS

  ChargebackInitiation (Message Function = RVRA (Reversal Advice) or RVNO (Reversal Notification)) ChargebackInitiation.ReconciliationAmount amounts where Transaction type = 00 to 1Z)

12.3.2 Reconciliation amount calculation - *AdditionalFeeReconciliation*

The *AdditionalFee.FeeAmount.Amount* present in each message is used to compute the fees reconciliation.

Calculate the total sum of:

*AdditionalFee.FeeAmount.*Amount with CreditDebit=Debit *(in reconciliation currency)*

And populate it to:

AdditionalFeeReconciliation.Amount with *ReconciliationImpact = DEBT*

Calculate the total sum of:

*AdditionalFee.Amount.*Amount with CreditDebit=Credit *(in reconciliation currency)*

And populate it to:

AdditionalFeeReconciliation.Amount with *ReconciliationImpact = CRDT*

12.3.3 Reconciliation amount calculation - *NetReconciliationAmount*

Calculation of *NetReconciliationAmount.*

Calculate the total sum of:

AdditionalFeeReconciliation amount with *ReconciliationImpact = CRDT*

FinancialReconciliation amount with Type = FNCL and ReconciliationImpact = CRDT

FinancialReconciliation amount with Type = RVSL and ReconciliationImpact = CRDT

FinancialReconciliation amount with Type = CBGK and ReconciliationImpact = CRDT

Less the sum of:

AdditionalFeeReconciliation amount with *ReconciliationImpact = DEBT*

FinancialReconciliation amount with Type = FNCL and ReconciliationImpact = DEBT

FinancialReconciliation amount with Type = RVSL and ReconciliationImpact = DEBT

FinancialReconciliation amount with Type = CBGK and ReconciliationImpact = DEBT

Equals:

NetReconciliationAmount with the *Credit Debit indicator* indicating the amount to be debited or credited.

13 File Action

13.1 File Action description

A file action is used to add, change, delete or replace a file or record or inquire into a file or perform card administration, e.g. report lost or stolen cards. The Data record data element can be used to convey specific file action record or file information.

File action notification/notification acknowledgement, instruction/instruction acknowledgment messages are the key components of the file transfer process.

FileActionInitiation is used by any entity to transfer, inquire, or perform an action of non-financial information

FileActionResponse is used by any entity to give the result of the processing

13.2 File Action message rules

The following applies to all file action messages.

a) A file action request message (*Request* at *MessageFunction* in *FileActionInitiation*) is used when a file action is required. The File Action Type and the File Action Scope data elements can be used to indicate the type of file action required.

b) A file action request response message (*Request* at *MessageFunction* in *FileActionResponse*) is sent in response to a file action request message (*Request* at *MessageFunction* in *FileActionInitiation*). The File Action Type and the File Action Scope data elements specify the action taken or to be taken.

c) A file action advice message (*Advice* at *MessageFunction* in *FileActionInitiation*) is used to inform the Destination of a file action that has been completed.

d) A file action advice response message (*Advice* at *MessageFunction* in *FileActionResponse*) is sent in response to a file action advice message (*Advice* at *MessageFunction* in *FileActionInitiation*). A file action advice response message (*Advice* at *MessageFunction* in *FileActionResponse*) indicates the Destination's response to the file action advice message (*Advice* at *MessageFunction* in *FileActionInitiation*).

e) A file action notification message (*Notification* at *MessageFunction* in *FileActionInitiation*) is used to inform the Destination of a file action that has been completed.

f) A file action notification response message (*Notification* at *MessageFunction* in *FileActionResponse*) is sent in response to a file action notification message (*Notification* at *MessageFunction* in *FileActionInitiation*).

g) A file action instruction message (*Instructure* at *MessageFunction* in *FileActionInitiation*) shall be used to inform the receiver of a file action to be completed. The sender can periodically specify that the receiver acknowledges the receipt of the most recently sent group of instruction messages (*Instructure* at *MessageFunction* in *FileActionInitiation*).

14 Settlement basics

- Clearing is the process of collecting transactions exchanged between the originator and the destination parties. This process results in the information about all the transactions over a given period is delivered to the destination.

- Reconciliation is the process of calculating and exchanging information about the sums and counts of the transactions and messages exchanged between the parties during the agreed period, in most cases directly related to the clearing process.

- Settlement is the process of calculating and determining the net financial position of each party for all the transactions that have been cleared during the agreed period.

14.1 Settlement service

Settlement service is used by the parties to settle transactions between them in each business arrangement, e.g. related to an area, country or other relationship.

All parties in a business arrangement must use the same settlement service.

14.2 Card programme

Card programme information is used to identify the card program used for the transaction or brand related to the transaction. In settlement context this allows for grouping the transactions made with specific characteristics, e.g. related to card scheme, card type (consumer, commercial) or other parameters.

14.3 Settlement parties

Settlement parties include the parties involved in exchanges of the card transactions as well as parities responsible for transfers of funds and respective reporting.

- Acquirer – the party responsible for payments towards the acceptors
- Issuer – the party providing funding for the settlement
- Agent – intermediary party or parties involved in the settlement process
- Settlement agent – the party responsible for collecting the information about the settlement totals and initiating financial transfers between the parties and other settlement agents
- Financial institution (FI) – the party responsible for transferring the funds between the settlement agents

14.4 Transaction currency vs. settlement currency

Transactions may occur in different currencies depending on the location of the acceptor accepting a card payment. The transaction currency may differ from the cardholder currency. Moreover, the transaction funds may be settled through yet another currency which is the settlement currency.

All parities participating in a settlement service must use the same settlement currency.

14.5 Settlement date

Settlement date is the day when the periodical exchange of funds occurs. In most cases the settlement dates are predefined either on the calendar basis or in the *transaction day + n days* principle.

14.6 Issuer totals

Totals related to the issuer relationship. Mostly used by the card schemes to group the transactions according to the issuer.

14.7 Acquirer totals

Totals related to the acquirer relationship. Mostly used by the card schemes to group the transactions according to the acquirer.

14.8 Other totals

Other totals contain the transactions that are neither related to an issuer nor acquirer. These may include fee collection and funds disbursement transactions.

14.9 Net settlement

Net settlement is a concept of the settlement process employing the “netting” which is a way of calculating the amounts being owed by a given party subtracting their incoming payments from the outgoing thus effectively minimizing the amount of money exchanged.

This contrasts with the gross settlement arrangement where every transaction is separately settled between the parties.

14.10 Settlement positions

The net settlement process results in the settlement positions for the parties which are the amounts owed by or owed to the parties. The settlement positions are specific for the settlement service, card programme, settlement date and settlement currency as well as the settlement type – final position or interim position.

14.11 Funds transfer amount

Amount resulting from the net settlement calculation which the settlement agent is instructing the FI to be transferred.

15 Pre-authorisation

An acquirer may allow specific merchants (e.g. Automated Fuel Dispenser (AFD), Vending machines, etc.) to process Pre-Authorisation.

Pre-Authorisation is an authorisation for a maximum amount or proxy, where a Pre-Authorisation Completion contains the actual amount. This occurs within a specified (relatively short) timeframe upon completion of the transaction at the point of service.

This use case is typically the AFD payment use case, but is also open to other use cases on the market.

Each message shall set specific indicators as below.

15.1 Pre-Authorisation Initiation

Initiation of a pre-authorisation is performed using an AuthorisationInitiation (MessageFunction = Request) message with following characteristics:

- Transaction Attribute = Pre-Authorisation
- Amount Qualifier = Maximum or Proxy

Note: Amount value that is sent when the Proxy qualifier is set depends on the market and country: it can be one single unit of amount.

15.2 Update Pre-Authorisation

This step is optional.

Note: Single message will not use the update of pre-authorisation

Update of a pre-authorisation **may be** performed using an *AuthorisationInitiation* (MessageFunction = Advice) message with following characteristics:

- Transaction Attribute = Completion or not set
- Amount Qualifier = Actual

Alternate flow is to update a pre-authorisation by using a **Reversal Initiation** (MessageFunction = Advice) message with following characteristics:

- Transaction Attribute = Completion or not set
- Amount Qualifier = Actual

15.3 Completion Pre-Authorisation

Completes the pre-authorisation initiation for the final amount.

15.3.1 Single

Completion of a pre-authorisation is performed using a:

- ***FinancialInitiation*** (MessageFunction = Advice) (refer to MDR part 1 chapter 6.2.5)

Characteristics of the FinancialInitiation are:

- Transaction Attribute = Pre-Authorisation Completion
- Amount Qualifier = Actual

15.3.2 Dual

Completion of a pre-authorisation is performed using a:

- ***FinancialInitiation*** (MessageFunction = Notification) if update was done with an Advice (refer to MDR part 1 chapter 6.2.8)
- ***FinancialInitiation*** (MessageFunction = Advice) if no update was done (refer to MDR part 1 chapter 6.2.7)

Characteristics of the FinancialInitiation are:

- Transaction Attribute = Pre-Authorisation Completion
- Amount Qualifier = Actual

Notes:

*The length of the period where the approval code will be valid may be required in Pre-Authorisation Time Limit.

*When the transaction is completed, the acquirer shall send FinancialInitiation for the completed amount.

*When a transaction which has been authorized using Pre-Authorisation is subsequently cancelled, the pre-authorisation amount will need to be reversed.

16 Reservation

Reservation is when an acquirer allows specific merchants (e.g. reservation of vehicle rentals and lodging) to process pre-authorisation for an estimated amount for a longer period of time, such as the duration of the reservation. This timeframe is rule dependent and will be defined by the specific implementation.
This also provides estimated amount for which the cardholder can be debited in case of no-shows.

This use case is typically for Reservation and Rental, but is also open to other use cases on the market.

Each message shall set specific indicators as below.

16.1 Pre-Authorisation Initiation

Initiation of a pre-authorisation is performed using an AuthorisationInitiation (MessageFunction = Request) message with following characteristics:

- Transaction Attribute = Pre-Authorisation or Prepayment or not set
- Amount Qualifier = Estimated or Reserved

16.2 Update Pre-Authorisation

Update of a pre-authorisation is an adjustement (decrease of increase the amount).

It can be performed using an

- ***AuthorisationInitiation*** (MessageFunction = Request, Advice)

Transaction Attribute = Incremental

Amount Qualifier = Estimated or Reserved or Actual

- ***Reversal Initiation*** (MessageFunction = Advice)

Transaction Attribute = Completion or not set

Amount Qualifier = Actual

16.3 Completion Pre-Authorisation

Completion of a pre-authorisation is performed using:

- ***FinancialInitiation*** (MessageFunction = Notification)
- ***FinancialInitiation*** (MessageFunction = Advice)

Characteristics of the ***FinancialInitiation*** are:

- Transaction Attribute = Completion or Pre-Authorisation Completion (when the initial authorisation transaction attribute is Pre Authorisation) or not set
- Amount Qualifier = Actual

16.4 Other exchanges linked to Reservation

16.4.1 Delayed Charge

This is a merchant-initiated transaction per terms agreed by the cardholder, possibly linked to a previous transaction, whereby the merchant needs to initiate a payment after original services are rendered.

(e.g. After check-out, after vehicle rental return)

…Transaction/MessageReason

16.4.2 No Show

When a customer did not use or cancel the agreed upon service (e.g. Lodging, vehicle rental).

…AddendumData/VehicleRental/RentalInvoice/NoShowIndicator

…AddendumData/Lodging/Summary/NoShowIndicator

*This can be sent as AuthorisationInitiation, FinancialInitiation, or separately as AddendumInitiation.

16.5 Deferred payment

This is a transaction whereby the merchant and the consumer have agreed to make actual payment on a future date for goods or services already rendered.
…Transaction/TransactionAttribute =DFRD

17 e-Commerce

17.1 Identification of the e-commerce environment, main characteristics

Electronic commerce transactions are identified by using the following indicator:
Body/Context/PointOfServiceContext/E Commerce Indicator set to True.

The commonly other used values for Body/Context/PointOfServiceContext/CardDataEntryMode are the following:

[DFLE] “Account Data”: the credentials that are registered by the merchant are used to identify the card, PAN is not entered manually by the cardholder at the time of the transaction.

[MLEY] “Manual Entry”: PAN is entered manually by the cardholder on the merchant web page.

There are two types of electronic commerce transactions for which some specific rules are specified on some markets. This data helps to know whether the customer is “online” so the merchant has capability to authenticate them.

Body/Context/TransactionContext/Transaction Initiator is set to:

[CUST] “Customer” to identify that the transaction is initiated by the customer.

[MERC] “Merchant” to identify that the transaction is initiated by the merchant.

Body/Context/PointOfServiceContext/E commerce data component is iterative to convey additional data related to Ecommerce.

17.2 Verification of 3DS-based authentication

“Three Domain Secure” is one method used to secure remote transactions.

EMV Co standard provides one framework for its implementation.

Target of 3DS process at the issuer side is to verify the authentication elements coming from the acceptance domain. One proof of the authentication and extra information about the 3DS process itself are grouped together and conveyed in the Verification component.

To facilitate the use of the component, for 3DS EMV transactions:

Body/Context/Verification/Type is populated with ThreeDS : “THDS”

Body/Context/Verification/VerificationInformation/Type is populated with the “Field name” as defined in the EMV 3-D Secure Protocol and Core Functions Specification.
Body/Context/Verification/VerificationInformation/Value is populated with the value in the corresponding format field.

18 Example of a message implementation

Below is an example of a message implementation defined by the following elements (not all are mandatory):

First two rows of the following table are mentioned in the use case definition in MDR Part 1. The following data elements further specify the purpose and interpretation of the message contents.

Element	Example
Message class	AuthorisationInitiation/AuthorisationResponse (echo)
Message function	REQU Request
TransactionType	00 GoodsAndServicesPull
AccountFrom	20 Credit facility – default
AccountTo	00 Default – unspecified
AdditionalService	LOYT Loyalty Services
TransactionAttribute	INCR Incremental (to previous authorisation)
MessageReason	1505 Online forced (ICC)
SpecialProgrammeQualification	Qualification for incentive or other related programmes.

Additionally, following elements are used in response message (not all are mandatory):

Element	Example
Result	PRCS Processed
ApprovalCode	123456 Exactly 6 characters
ResultDetails	08 Honour with identification
ActionRequired	Yes
ActionType	CNTI Contact Issuer
AdditionalAction	DISP Display Message

TransactionType: generic description of the service through a code list

- Example in Authorisation: 01 (Cash)
- Example in FileAction: 72 (Activation)

AdditionalService: complementary description of the service through a code list (only used in Authorisation and Financial)

Example in Authorisation: DCCV when the TransactionType is 00 (Goods and service)

AdditionalService will contain the type= Other National / Other Private. In this case “Other Type” is populated with the value to be used as additional service.

TransactionAttribute: additional value to detail the step in a processed service (only used in Authorisation and Financial)

Example in Authorisation: “Initial” when the TransactionType is “Pre-Authorisation”

MessageReason: actual reason for which the message is sent

- Example in Authorisation: “FloorLimit”
- Example in Reversal: “TimeOut”
- Example in FileAction: “StolenCard”

19 References

The complete catalogue of official ISO 20022 messages, including the Message Definition Reports and XML Schemas, is available on the ISO 20022 official Web site (**http://www.iso20022.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/