



EHl Guide (JSON version)

Version: 5.1

12 September 2025

Publication number: JSON EHI-5.1-9/12/2025

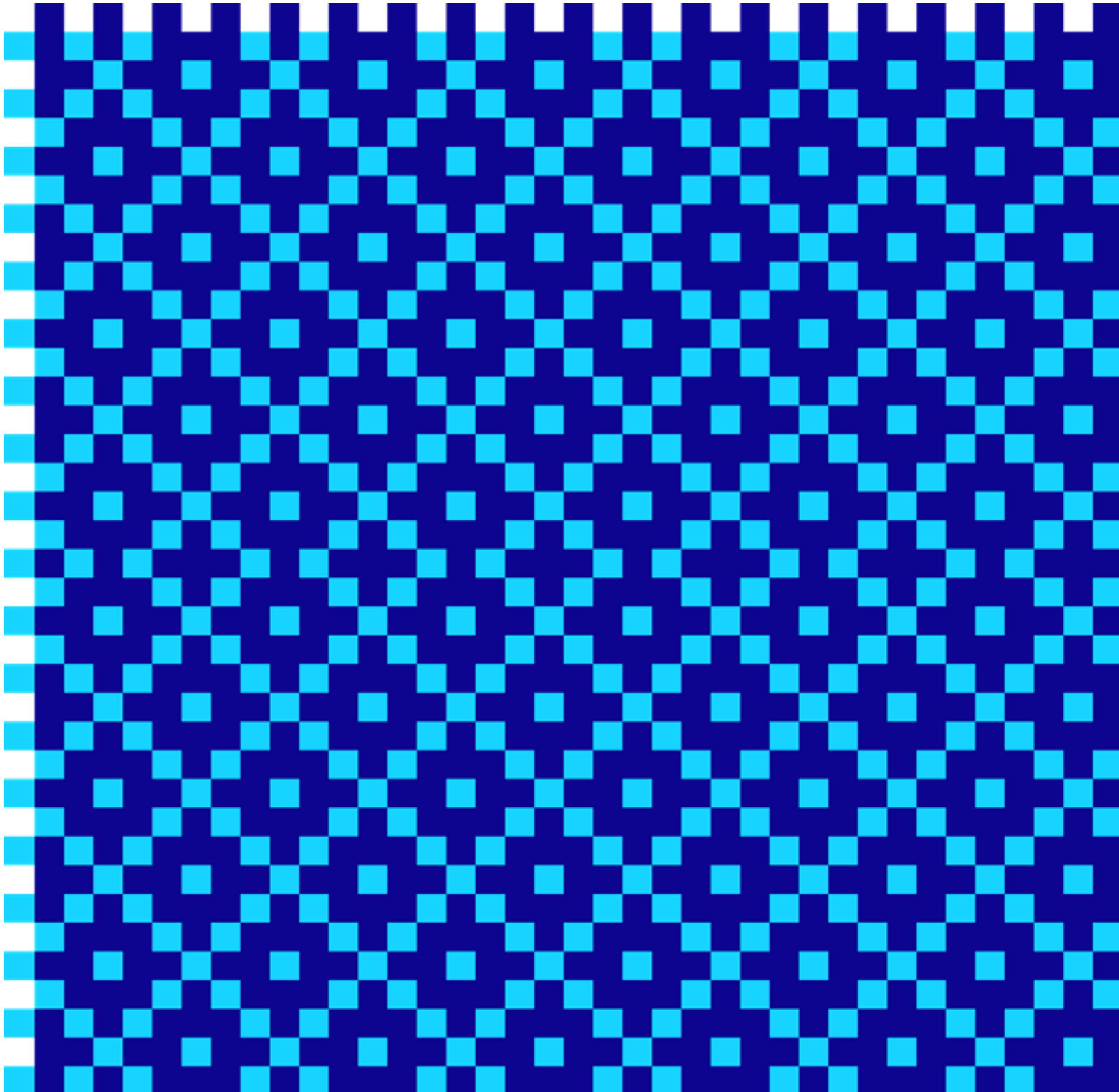
For the latest technical documentation, see the [Documentation Portal](#).

Thredd, Kingsbourne House, 229-231 High Holborn, London, WC1V 7DA

Support Email: occ@thredd.com

Support Phone: +44 (0) 203 740 9682

© Thredd 2025





Copyright

© Thredd 2025

The material contained on this website is copyrighted and owned by Thredd Ltd together with any other intellectual property in such material.

The material contained in this guide is copyrighted and owned by Thredd Ltd together with any other intellectual property in such material.

Except for personal and non-commercial use, no part of this guide may be copied, republished, performed in public, broadcast, uploaded, transmitted, distributed, modified or dealt with in any manner at all, without the prior written permission of Thredd Ltd., and, then, only in such a way that the source and intellectual property rights are acknowledged.

To the maximum extent permitted by law, Thredd Ltd shall not be liable to any person or organisation, in any manner whatsoever from the use, construction or interpretation of, or the reliance upon, all or any of the information or materials contained on this website.

To the maximum extent permitted by law, Thredd Ltd shall not be liable to any person or organisation, in any manner whatsoever from the use, construction or interpretation of, or the reliance upon, all or any of the information or materials contained in this guide.

The information in these materials is subject to change without notice and Thredd Ltd. assumes no responsibility for any errors.



About this Guide

This guide describes the Thredd External Host Interface (EHI) and provides technical specifications on how to integrate your systems to EHI. The guide provides details of how to receive and respond to EHI messages in JSON format.

Document Scope

You should read this guide if you are using EHI for payment transaction authorisation and/or subscription to the EHI real-time payment transaction data feed.

Target Audience

This guide is aimed at developers who need to integrate their applications to Thredd, using EHI.

What's Changed?

If you want to find out what's changed since the previous release, see the [Document History](#) section. For a list of EHI fields available with each of the supported EHI versions, see [EHI Versions](#).



Document	Description
Cards API Documentation	For customers using the Thredd REST-based Cards API, see the Cards API Docs Website .
Thredd Portal Guide	Describes how to use the Thredd Portal to manage your cards and transactions.
Smart Client Guide	Describes how to use the legacy Thredd Smart Client desktop application to manage your account.
Chargeback Guide	Describes the chargeback process and options for managing chargebacks.
Transaction XML Reporting Guide	Describes the transaction XML report sent to Program Managers.
3D Secure Guide	Describes the Thredd 3D Secure service.
Card Transaction System Guide	Describes how to submit card test transactions in the UAT environment.

Tip: For the latest technical documentation, see the [Documentation Portal](#).



SECTION 1: GETTING STARTED

You should read this section if you are new to the External Host Interface (EHI) and want to understand how it works and how EHI is configured.

Topics covered in this section:

- [About this Guide](#)
- [Overview](#)
- [EHI Operating Modes](#)
- [EHI Connection and Messages](#)
- [Card Payment Networks](#)
- [Transaction Flow Scenarios](#)
- [EHI Data Feeds](#)
- [Transaction Types](#)
- [EHI Version Control](#)
- [EHI Configuration Options](#)
- [Integration Steps](#)

Tip: To find out how to integrate your external host system to EHI, see [Processing EHI Transactions](#).

Tip: If you are upgrading your EHI version and want to understand what has changed, see [Document History](#).



1.1 Overview

The External Host Interface (EHI) offers a way to exchange transactional data between the Thredd processing system and the Program Manager's externally hosted systems. All transaction data processed by Thredd is transferred to the external host system via EHI in real time. EHI provides two main functions:

- A real-time transaction notification data feed
- Payment authorisation control

Note: Thredd supports Visa, Mastercard and Discover networks, as well as smaller networks that use the Mastercard Network Exchange (MNE), such as STAR and Pulse¹. EHI provides a single, unified message format, which enables your systems to receive and process messages using a single integration to Thredd. For more information, see [Card Payment Networks](#).

1.1.1 Real-time Transactional Data Feed

Thredd receives global real-time card and payment-related notifications from the card schemes (Visa and Mastercard networks). These notifications are merged into a single Thredd message format which your systems can process. The real-time notifications are sent via a secure connection to the external host URL endpoint



Thredd will process your response and respond to the card scheme (Mastercard or Visa). The authorisation decision process is in real time.



1.2 EHI Version Control

The currently supported EHI versions are 3.x, 4.x and 5.x.

Thredd creates new EHI versions for new functionality as required by scheme and regulatory changes and/or client enhancements. These are available to all customers.

We distinguish between the following versions:

- *Major versions*: add significant functionality (e.g. MDES). Examples of major version numbers are: 3.0, 4.0, 5.0, and so on.
- *Minor versions*: add minimal functionality (e.g. new field). Examples of minor version numbers are: 4.1, 4.2, 4.3, 5.1. and so on.
- *Document versions*: reflect guide update. Examples of document version numbers are: 4.0.1, 4.0.2, 4.0.3. and so on.

We send notifications out to you whenever there is a new release. To **upgrade your EHI version**, contact your Thredd account manager or Thredd implementation manager.



1.3 EHI Operating Modes

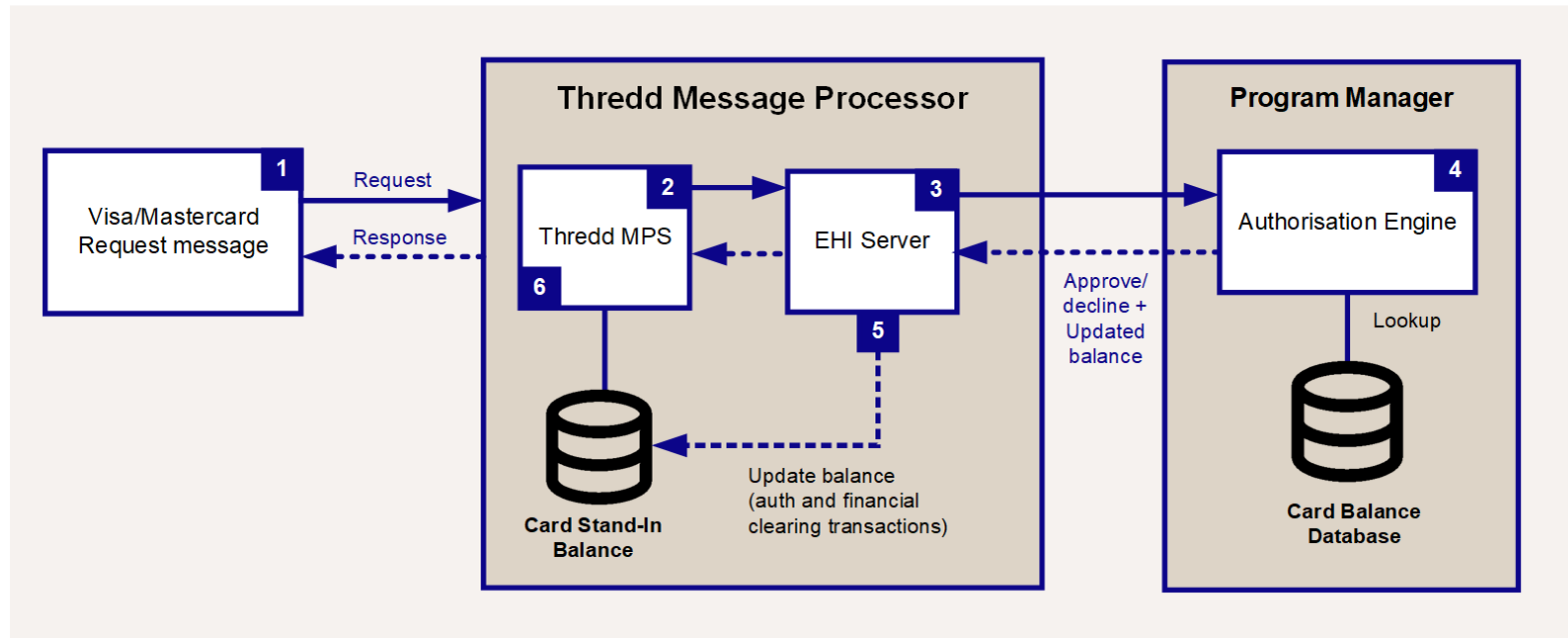
For authorisation types of transactions, the external host interface can operate in one of five supported modes. See the table below:

Setup type	Mode	Who Authorises?	Who Maintains the Balance?	Thredd Authorisation Stand-In	Details
Gateway Processing	1	External Host	External Host	No	Your systems maintain the balance and perform authorisation.
Cooperative Processing	2	Thredd	Thredd / External Host	Yes	Thredd maintains the balance and performs authorisation. You can override an approval decision. In Approval with Load your systems can update the Thredd-maintained balance.
Full Service Processing	3	Thredd	Thredd	No	Thredd maintains the balance and performs authorisation. You receive a read-only response.
Gateway Processing with STIP	4	External Host	External Host	Yes	Your systems maintain the balance and perform authorisation. Thredd provides Stand-In authorisation if the external host is unavailable.



- EHI response messages (see [Update_Balance](#) field).
- Balance Update Web Service (see [ws_BalanceUpdate](#) in the *Web Services Guide*). Note that this functionality is currently only available via SOAP Web Services.

In EHI Mode 4, all transactions (authorisations and financials) updates the Thredd stand-in balance.





For details of the fields included in each type of transaction, see [EHI Data Feeds](#).

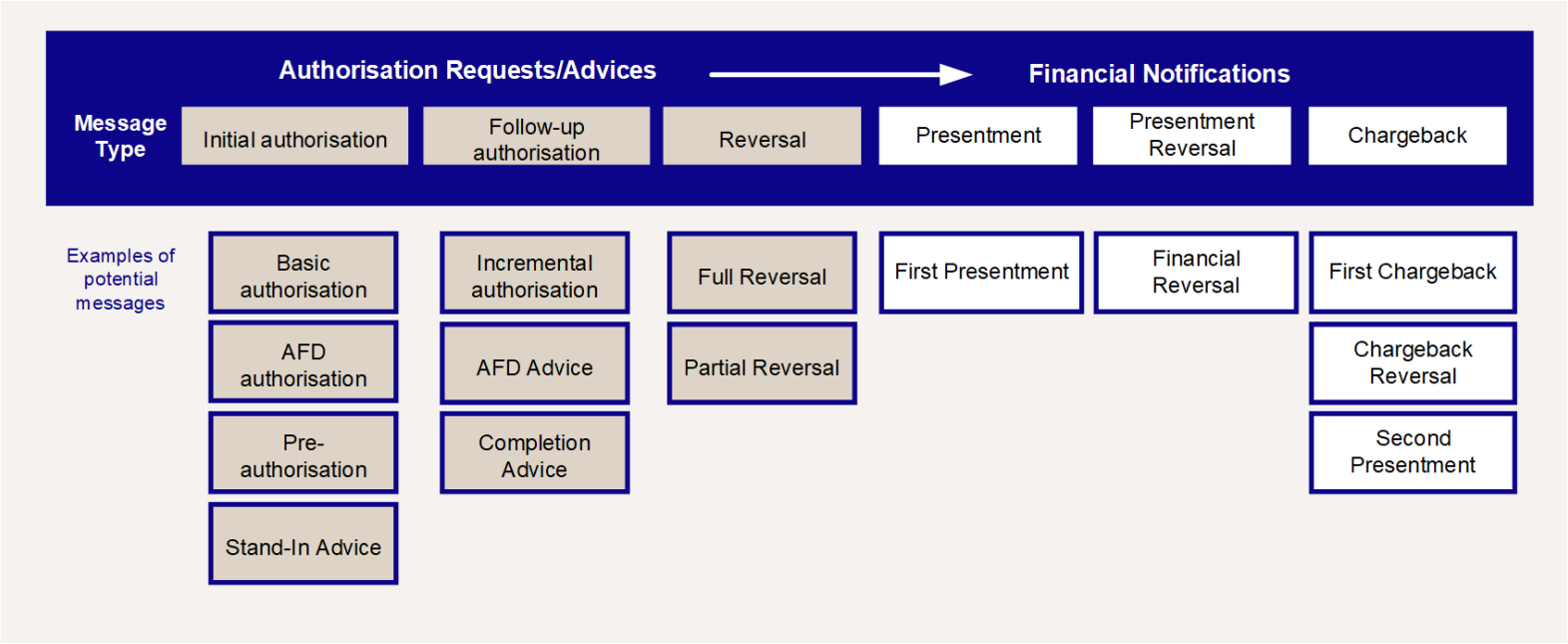
1.4.2 Transmission Control Protocol (TCP) Connection

For each message sent via the EHI, the following occurs:

1. The EHI makes a TCP connection to the external host URL configured for your program. Thredd is always the TCP client, the external host (your system) is always the TCP server.
2. Using the HTTP POST method, EHI sends an HTTP message to this TCP connection with the SOAP XML message as the message body. The message body is XML encoded to the UTF-8 format. See <https://www.w3.org/TR/REC-xml> for specifications.
3. The response from your system must be a valid HTTP response with the HTTP response body containing the valid XML response data.
4. In the production environment EHI uses Thredd's Trust Framework, which leverages the **Mutual Transport Layer (mTLS)** protocol. .

1.4.3 mTLS Connection

EHI uses Thredd's Secure Connectivity Framework, which leverages the **Mutual Transport Layer (mTLS)** protocol.





1.6 Transaction Flow Scenarios

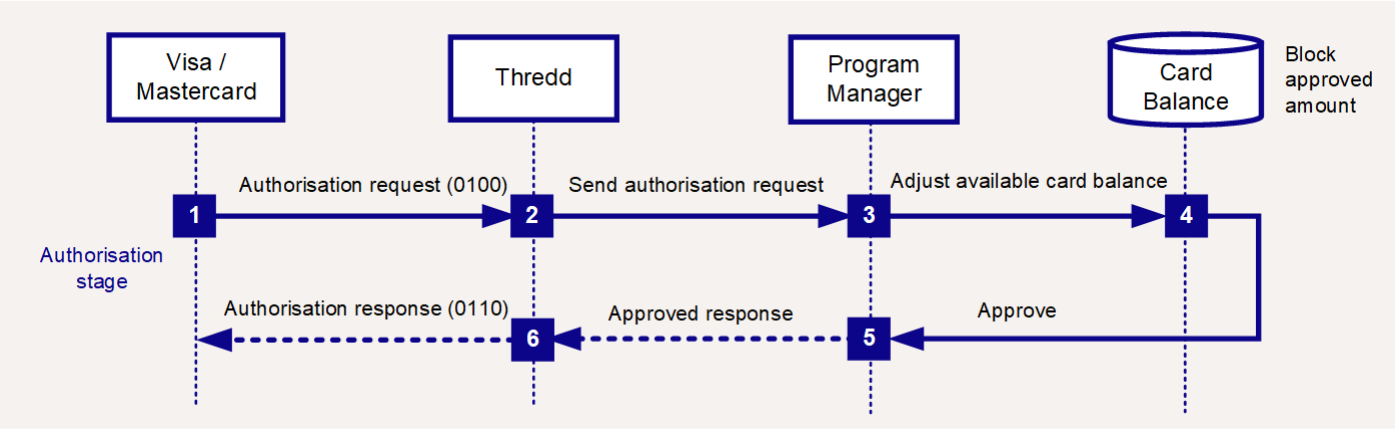
This section provides examples of typical transaction flows with Message Transaction IDs (MTIDs) for Dual Message systems. This provides a flavour of the type of messages you can expect to receive from the Thredd system.

1.6.1 Authorisations

Authorisation is the stage in a transaction life-cycle where a merchant requests approval for a card payment amount. If the authorisation is approved, the amount is ring-fenced on the card. Typically, the merchant then has 7-10 days to request the transfer of the authorised funds (although this can be up to 28 days for certain merchant category codes). For additional information see [What are Authorisations and how do they work?](#)

Authorisation with Approve

The following scenario illustrates a typical approve journey for EHI modes 1, 2 and 4 (Gateway and Cooperative Processing).



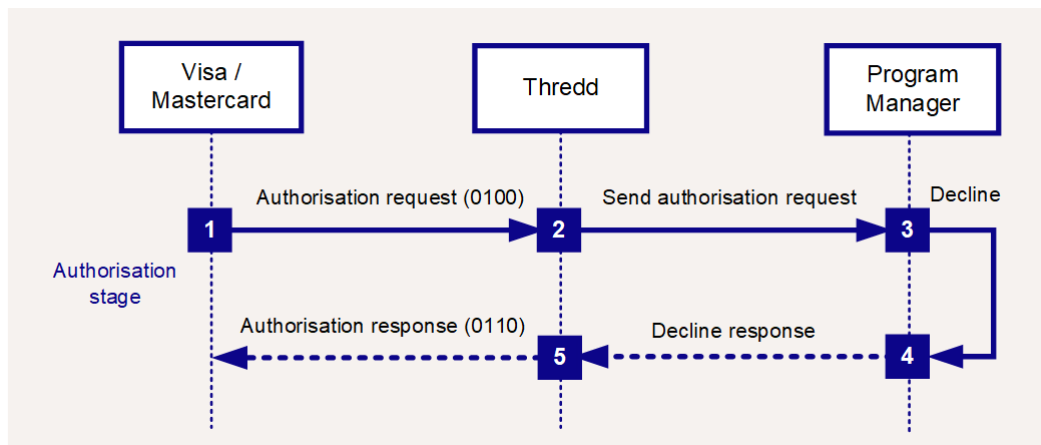


Figure 9: Authorisation Flow - Decline

1. The scheme sends an 0100 authorisation request to Thredd.
2. Thredd carries out validation checks and sends the request to the external host (Program Manager).
3. The Program Manager declines the request.
4. The Program Manager returns a declined response, for example:

*"Responsestatus": "05" and "Acknowledgement": "1".**

5. Thredd responds to the



Message Type	Description	How to Process	How you should respond
	generates this based on the expiry date configured for the card.	the Thredd API)	



Authorisations Not Acknowledged (Auths_NotAcknowledged)	Financials Acknowledged (Financials_Acknowledged)	Financials Not Acknowledged (Financials_NotAcknowledged)
Loads Acknowledged (LoadsUnloads_Acknowledged)	Loads Not Acknowledged (LoadsUnloads_NotAcknowledged)	
Balance Adjustment Expiry Acknowledged (BalanceAdjustExpiry_Acknowledged)	First Transaction ID (FirstTxn_ID)	



- Do the cards support the functionality and behaviour you expect?
- Are the CHIP profiles on the cards correctly set up? (i.e., is the card working and card validation working as expected.)
- Is your mobile app behaving as expected? (e.g., displaying real-time details of card status, card transactions and account balances.)
- If you are using mobile tokenisation services such as VDEP and MDES have you tested different use case scenarios?
- Have you tested other components of your card service, such as:
 - Recurring payments
 - Fees
 - Cardholder authentication (3D Secure)
 - Exception flows, such as reversals, refunds and chargebacks.
- Have you checked the cardholder journey from an end-to-end perspective?

Thredd Card Tests

Thredd runs a set of generic pavement tests where we check a range of card functionality, such as:

- Keys are set up correctly
- Both contact and contactless card transactions are working
- The card usage groups set up for your programme are declining as expected
- General authorisations are being received, refunds are working correctly, authorisation requests are being declined and approved as expected.



SECTION 2: PROCESSING EHI TRANSACTIONS

You should read this section to understand how to integrate your external host system to EHI.

This section covers the following topics:

- Processing EHI Transactions
- Transaction Matching
- Data Types
- GetTransaction Message
- Cut_Off Messages

Tip: If you are new to EHI and want to understand how EHI works and the available configuration options, see the [Getting Started](#) section.



```
<CVV2_Result>400</CVV2_Result>
<CurBalance_GPS_STIP>0</CurBalance_GPS_STIP>
<AvlBalance_GPS_STIP>100</AvlBalance_GPS_STIP>
</GetTransactionResult>
```

Note: You should only respond with `Update_Balance = '1'` (update balance) if you have received both `Balance_Sequence` and `Balance_Sequence_Ext` in the EHI request message. These balance sequence fields are only relevant to real-time transactions where balance adjustments can be made in the response. If you receive a financial notification and need to adjust the Thredd-held balance, you must use the `Balance Update Web Service` or Adjust Balance Cards API.



Transport for London (United Kingdom of GB and NI) Merchant Transactions

Transport for London (TFL) (London's Public Transport network) have various waivers on authorisation requirements, in order to permit more offline transactions and for amounts larger than usually permitted, after a single approved authorisation request.

Example 2

The following example is from Mastercard:

(All of the below have the same [traceid_lifecycle](#))

- MTID=0100 authorisation for GBP 6.60, which was approved
- MTID=1240 presentment for GBP 6.60
- MTID=1240 presentment for GBP 5.80
- MTID=1240 presentment for GBP 5.30

Note: Rules differ for UK and non-UK BIN ranges. Check with your Issuer for the latest network rules.



2.4 JSON Data Types and EHI Fields

This section describes the data types of the fields in the JSON message. You should use these details together with the information on [Data Types](#) for EHI GetTransaction messages.

Note: The Request field values are case sensitive.

Note: The EHI feed does not send JSON fields to externally-hosted systems where the values for the field data are blank. This is different from EHI feeds with XML fields where the feed can send blank values.

Request Fields

Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
Acquirer_id_DE32	String		N/A
ActBal	Number	Y	Y
Additional_Amt_DE54	String		N/A



Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
Note	String		N/A
POS_Data_DE22	String		N/A
POS_Data_DE61	String		N/A
POS_Termnl_DE41	String		N/A
POS_Time_DE12	String		N/A
Proc_Code	String		N/A
Resp_Code_DE39	String		N/A
Ret_Ref_No_DE37	String		N/A
Settle_Amt	Number	Y	Y
Settle_Ccy	String		N/A
Status_Code	String		N/A
Token	Number	Y	Y
Trans_link	String		N/A
Txn_Amt</			



Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
MTID	String		N/A
ProductID	Number	Y	Y
Record_Data_DE120	String		N/A
SubBIN	Number	Y	Y
TLogIDOrg	Number	Y	Y
VL_Group	String		N/A
Dom_Fee_Fixed	Number	Y	Y
Non_Dom_Fee_Fixed	Number	Y	Y
Fx_Fee_Fixed	Number	Y	Y
Other_Fee_Amt	Number	Y	Y
Fx_Fee_Rate	Number	Y	Y
Dom_Fee_Rate	Number	Y	Y
Non_Dom_Fee_Rate	Number		



Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
dest_bank_account	String		N/A
GPS_POS_Capability	String		N/A
GPS_POS_Data	String		N/A
Acquirer_Reference_Data_031	String		N/A
Response_Source	String		N/A
Response_Source_Why	Number	Y	Y
Message_Source	String		N/A
Message_Why	Number	Y	Y
traceid_lifecycle	String		N/A
Balance_Sequence	Number	Y	N
Balance_Sequence_ExtHost	Number	Y	N
PaymentToken_id	Number	Y	Y
PaymentToken_creator	String		



Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
PaymentToken_activationMethod	Number	Y	Y
ICC_System_Related_Data_DE55 This field applies to both Visa and Mastercard.	String		N/A
Merch_Name	String		N/A
Merch_Street	String		N/A
Merch_City	String		N/A
Merch_Region	String		N/A
Merch_Postcode	String		N/A
Merch_Country	String		N/A
Merch_Tel	String		N/A
Merch_URL	String		N/A
Merch_Name_Other	String		N/A
Merch_Net_id	String		N/A
Merch_Tax_id	String		



Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
Clearing_Process_Date	String		N/A
Settlement_Date	String		N/A
DCC_Indicator	Number	Y	Y
multi_part_txn	Number	Y	Y
multi_part_txn_final	Number	Y	Y
multi_part_number	Number	Y	Y
multi_part_count	Number	Y	Y
SettlementIndicator	String		N/A
Network_TxnAmt_To_BillAmt_Rate	String		N/A
Network_TxnAmt_To_BaseAmt_Rate	String		N/A
Network_BaseAmt_To_BillAmt_Rate	String		N/A
POS_Date_DE13	String		N/A



Field	JSON Data Type	Null Allowed	Compatible with IEEE 754 binary-64 (double precision) ?
ClearingFileID	String		N/A
Network_Fraud_Data	String		N/A
SenderData	String		N/A
ReceiverData	String		N/A
AuthenticationAmountUpper	Number	Y	Y
AuthenticationCurrency	String		N/A
AuthenticationMerchantHash	String		N/A
FxProviderCardholderRate	Number		Y
FxProviderBookedRate	Number		Y

Note: For more information on JSON data types, see the [RFC 8259 JSON specification](#).

Response Fields



Number range and precision

For each field of JSON data type *Number*, refer to the field description in [GetTransaction Message Fields](#), which describes the field length range and maximum value you need to support. This is important to ensure no loss of information, especially for the fields with potentially large values.

Tip: Some of the above fields of JSON data type *Number* have a bigger range than the RFC 8259 data type of IEEE 754 binary-64 (double precision) which is described for numbers. If using the JavaScript language to process JSON, then for whole numbers (of Thredd data type *N(min.max)*) the JavaScript data type *bigint* would be a suitable alternative to the IEEE 754 binary-64 number data type.

Example

The [Matching_Txn_ID](#) field is a number of type N(1,16), with a minimum length of 1 digits and maximum length of 16 digits. Your systems must be able to handle a value containing up to 16 digits.

