



Tokenisation Service Guide

Version: 2.2

28 May 2026

Publication number: TSG-2.2-5/28/2026

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

Thredd, 33 Kingsway, London, WC2B 6UF, UK

Support Email: occ@thredd.com

Support Phone: +44 (0) 203 740 9682

© Thredd 2026





Copyright

© Thredd 2026

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 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 Mastercard and Visa token services and how Thredd supports tokenisation. It explains how to set up and process tokens on the Thredd system.

Document Scope

This guide describes the process of implementing and managing the Digital Wallet product on a Visa or Mastercard programme and is aimed at any new or existing Thredd customers who wish to add this functionality and enable token-based mobile payments for their cardholders.

Target Audience

This guide is intended for Thredd clients (Program Managers) who have prior knowledge of the Card Payments industry and are interested in integrating the Mastercard and Visa token services into their programme.

What's Changed?

If you want to find out what's changed since the previous release, see the [Document History](#) section.

How to use this Guide

If you are new to tokenisation and want to understand how it works, see [How Tokenisation Works](#).

To find out about the steps involved in implementing a tokenisation project, see [Implementing a Tokenisation Project](#).

For Thredd token service configuration options, see [Thredd Configuration Options](#).

Note: In this guide we reference two Thredd API options available for managing your cards and tokens: the Thredd SOAP-based web services (XML format), or the REST-based Cards API (JSON format).

Related Documents

Refer to the table below for other documents which should be used in conjunction with this guide.

Document	Description
Web Services Guide	Provides details of the Thredd SOAP Web Services API.
Cards API Website	Provides details of the Thredd REST-based Cards API.
EHI Guide	Provides details of the Thredd External Host Interface (EHI).
Smart Client Guide	Describes how to use the Thredd Smart Client to manage your account.
Thredd Portal	Describes how to use the Thredd Portal, Thredd's new web application for managing your cards and transactions on the Thredd Platform.

The following documents are available from Visa and Mastercard:

Document	Description
Visa Token Service Guide	Describes the Visa token service. Available online at: Visa Token Service .



Document	Description
Mastercard Digital Enablement Service Guide	Describes the Mastercard token service. Available online at: Mastercard Developers .

Note: You may need to register with an account with Visa and Mastercard to access these sites.

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



1 Introduction

Tokenisation is a security technology which replaces the sensitive 16-digit permanent account number (PAN) that is typically embossed on a physical card with a unique payment token (a digital PAN or DPAN) that can be used in payments and prevents the need to expose or store actual card details. The DPAN is used to make purchases in the same way as a normal Financial PAN (FPAN).

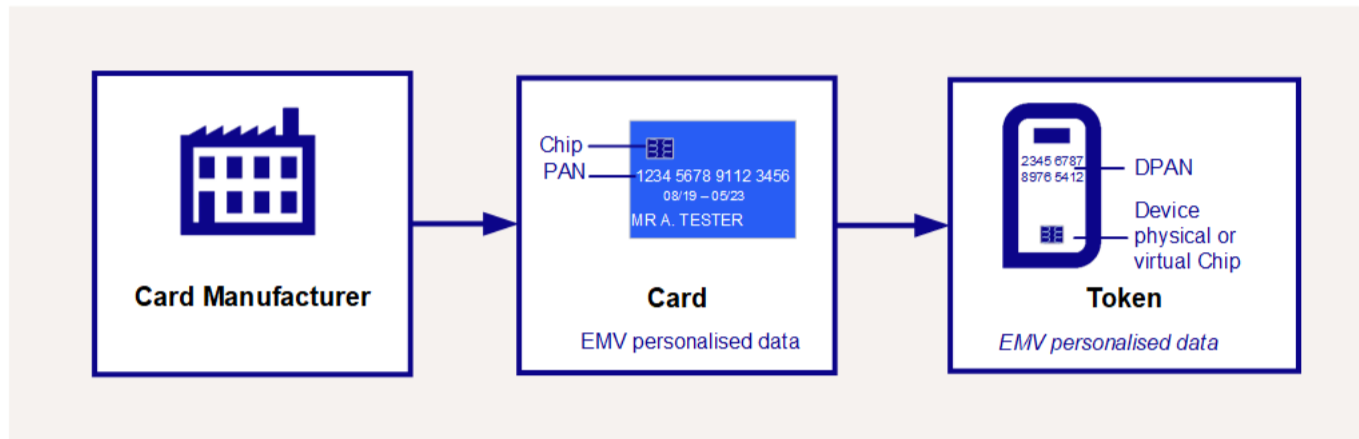


Figure 1: Tokenisation - converting a PAN to a DPAN

Tokenisation enables cardholders to access mobile wallet functionality – provided by companies such as Apple and Android – which allows payments to be made in store from a smart device such as a smartphone or tokenised device. Tokenisation also helps merchants to improve the security of online payment transactions by replacing the sensitive PAN card details with a token and storing this instead. The token can then be used for repeat or recurring payments.

Tokenisation is increasing the adoption of mobile wallet and other new payment technology and improving security across the industry. Its use will continue to grow as more merchants and issuers enable the service.

Both Mastercard and Visa offer a tokenisation service to card issuers. Mastercard offer the Digital Enablement Service (MDES) and Visa offer the Visa Token Service (VTS); Thredd refer to the Visa service as the Visa Digital Enablement Program (VDEP). Thredd supports both of these tokenisation services.

Note: Thredd do not share details of the FPAN or DPAN with Program Managers (Thredd clients). When a card is created on the Thredd system, we provide a unique public token that is linked to the card, and which can be used for queries and services on that card. The Thredd public token is for internal use only between Thredd and the Program Manager; it should not be confused with the payment token created during the tokenisation process described in this guide.



2 How Tokenisation Works

2.1 Who Participates in Tokenisation?

Tokenisation requires the following participants:

Cardholder

The cardholder enrolls with a mobile wallet provider or registers at an online merchant website.

Token Requestor

The token requestor initiates the request to convert your cardholder's Permanent Account Number (PAN) into a digital token. Token requestors can be mobile wallets (such as ApplePay) or online merchants (such as Netflix). Mastercard refer to the Token Requestor as the "Wallet Provider".

Token Service Provider (TSP)

The Token Service Provider is the party that generates the token and securely maps the PAN to a token. This is the Visa (VDEP) or Mastercard (MDES) systems that run the token service.

Issuer Host

The issuer host is Thredd, who receives the tokenisation request from Visa or Mastercard and decides on whether to approve or decline. During the implementation phase of the project, the issuer/Program Manager and Thredd work together to set up and create the token service.

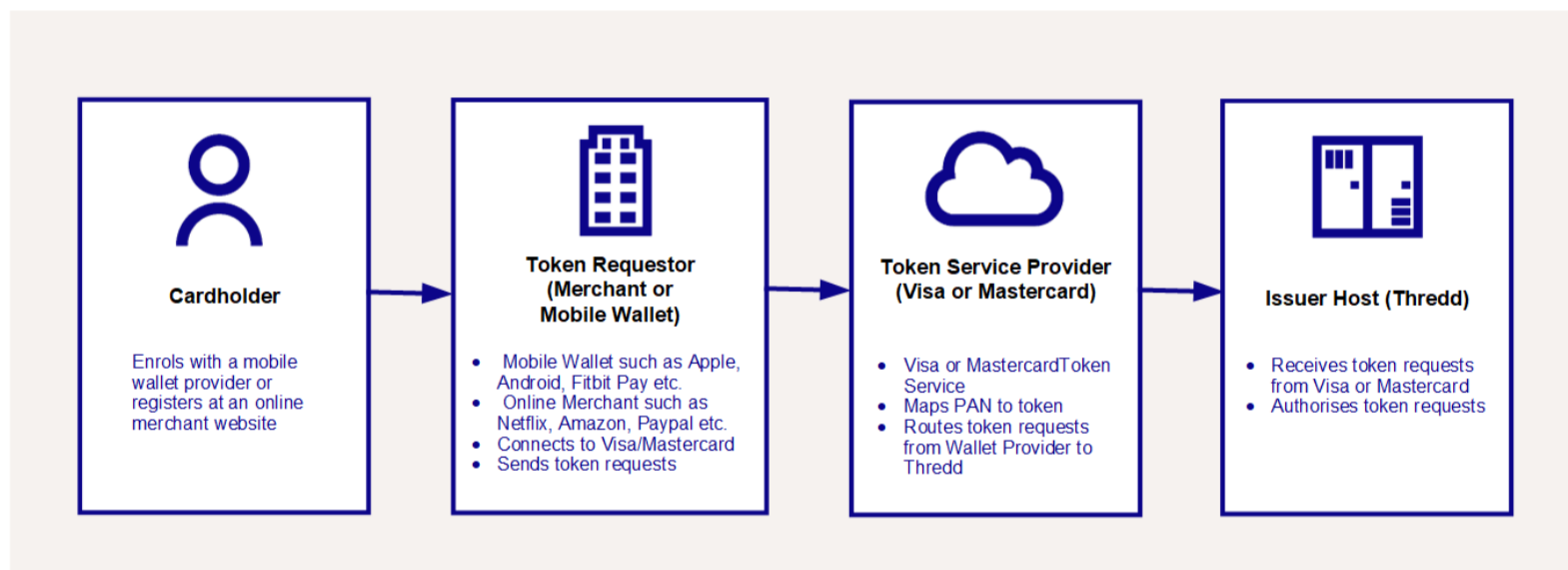


Figure 2: Participants in the Tokenisation Ecosystem

The Token Service Provider (Visa/Mastercard) receives token requests from the Token Requestor and sends them to Thredd for authorisation. There is no direct connection between Thredd and the Token Requestor during tokenisation and Thredd does not have the capability to act as a Token Requestor.

When using mobile wallet token requestors (such as Apple and Android), the Program Manager (Thredd customer) requires a separate commercial agreement with each of the three parties involved in tokenisation.

For online merchant tokenisation (i.e., for online payments), the card issuer does not need to have an existing relationship with the merchant.

2.2 Token Provisioning

Token provisioning is the act of creating and activating a digital token. The digital token is sometimes referred to as the DPAN, and is the same length as a normal 16-digit card financial PAN (FPAN).

This process must be completed first before the token can be used in transactions.

2.2.1 Token Provisioning Steps

Figure 3 provides a high-level overview of the token provisioning flow.

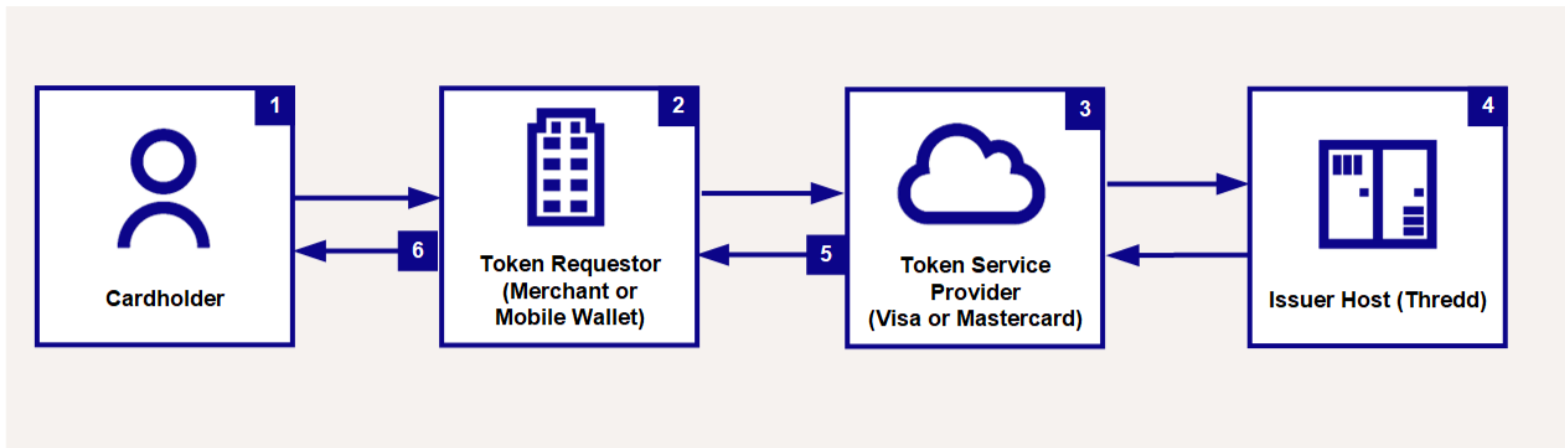


Figure 3: Token Provisioning Flow without Authentication

1. The cardholder enrolls their card with a token requestor (either an online merchant or a mobile Wallet provider).
2. The token requestor requests a new token from the token service provider (Visa/Mastercard).
3. The token service provider creates the payment token (DPAN), containing EMV and other card data, to replace the cardholder's FPAN. The token service provider sends a Token Activation Request (TAR) to the issuer host (Thredd).
4. Thredd decides if token activation can continue, based on the Thredd Configuration Options set up for your programme. (See Token Authorisation Options below.)
5. With Thredd approval the token service provider (Visa/Mastercard) activates the new payment token and sends the newly created token to the token requestor.
6. For an Online Merchant payment token, the token is stored for use on their website. For a Mobile Wallet payment token, it is installed on the phone for mobile Near Field Communication (NFC) use.

2.2.2 Token Authorisation Options

When Thredd returns a decision on the token request there are three options:

- Approve - token is active for use
- Approve with Authentication - additional authentication is required before the approved token can be used
- Decline - token is not approved.

The Thredd response code in the response triggers three different provisioning flows:

Thredd Response	Response Code	Provisioning Flow (Token Terminology)
Approve	00	Green Flow
Approve with Authentication	85	Yellow Flow
Decline	05	Red Flow

Each of these provisioning flows is described below.

2.2.3 Approve (Green Flow)

When Thredd receives the token activation request (TAR) and we approve, if cardholder authentication is not required, Thredd sends an approve message to the token service provider to create the token without further authentication¹.

Cardholder authentication is not required in the following circumstances:

- Authentication has already been performed (i.e., token is being push-provisioned; see [Push Provisioning](#))
- For an online merchant
- Based on the configuration for your Wallet Usage Group; see [Payment Token Usage Groups](#)

¹ Note that in some circumstances it is possible for a Program Manager or issuer to set up rules on Mastercard or Visa to ignore or overrule the Thredd response to a TAR. Please contact the card schemes for details.



Note: Your Thredd Wallet Usage Group configuration is used to determine the appropriate flow to trigger². Most Thredd Program Managers implement the approve with authentication flow.

2.2.4 Approve with Authentication (Yellow Flow)

When Thredd receives the token activation request (TAR) and we approve, if cardholder authentication is required, we send an approve with authentication message to the token service provider to create the token with cardholder authentication.

Cardholder authentication is only needed by mobile wallet token requestors (such as Apple and Android), where the cardholder is present at the time the card is being tokenised.

To authenticate a cardholder during token provisioning, the cardholder is provided with a One-Time Password (OTP) generated by the token service provider (Visa/Mastercard). The cardholder enters the OTP value into their mobile app for validation.

The Program Manager decides what delivery options are available to the cardholder for the OTP. These options can include:

- SMS text message to mobile phone
- Push notification/in-app notification
- Email
- Call centre (an operator reads out the passcode to the cardholder to enter; the passcode is available on Smart Client, via Thredd Web Services or the ThreddExternal Host Interface (EHI) and will expire after a limited period, such as 2 hours).

Note: Thredd currently only sends the OTP via SMS text message directly to the cardholder's mobile phone. For all other OTP methods, you will need to deliver the OTP to the cardholder. The OTP is always sent via EHI, even if Thredd also sends an SMS direct to the cardholder. The OTP can be viewed in Smart Client or retrieved via Web services.

Figure 4 below describes the Approve with Authentication (Yellow) flow.

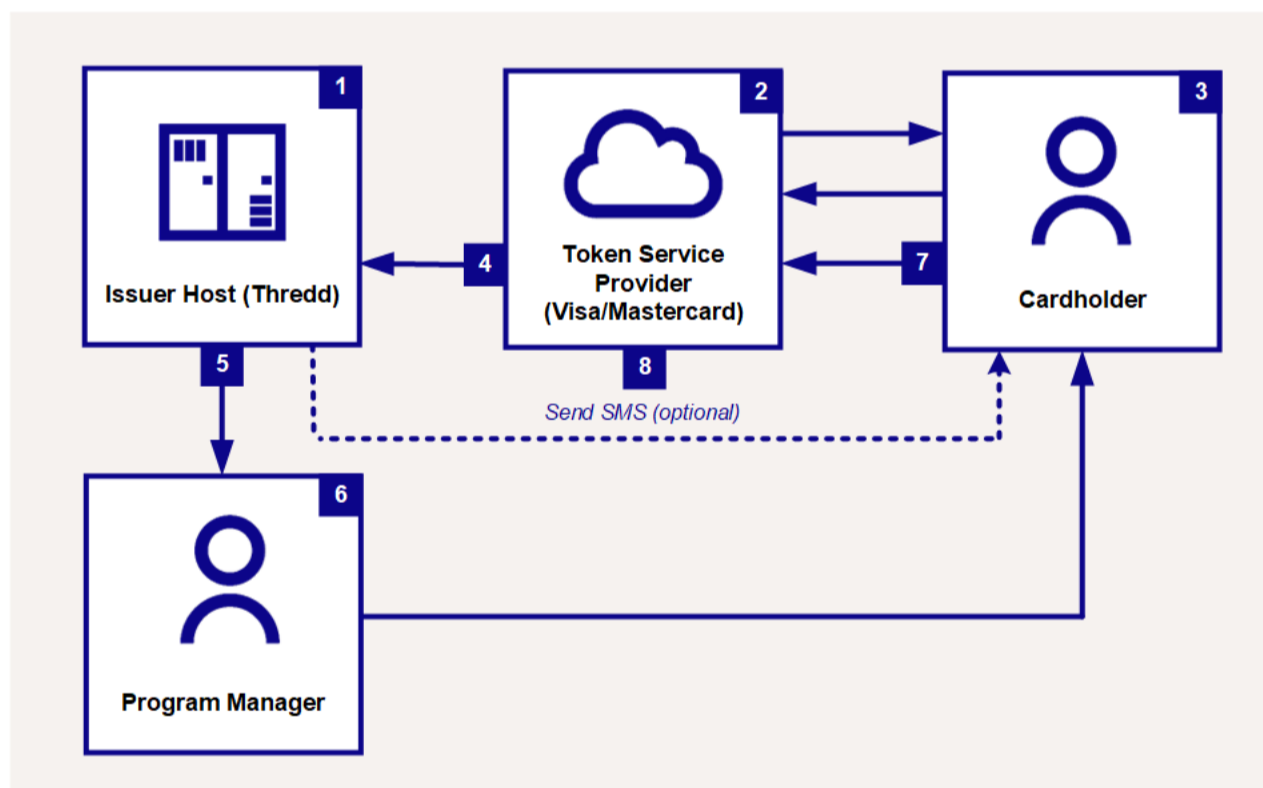


Figure 4: Cardholder Authentication During Token Provisioning

This flow commences after the token has been generated, but further user authentication is required before it can be used:

1. Thredd sends an 85 (approve with authentication) response to the token service provider (Visa/Mastercard). The response contains a list of cardholder verification methods (CVMs), based on the configuration of your Wallet Usage Group for your cards.
2. The token service provider sends a list of available cardholder verification methods to the cardholder.
3. The cardholder selects one of the verification methods shown on their mobile phone wallet application.
4. The Token Service Provider receives the method selection and sends the one-time password (OTP) to Thredd.
5. Thredd always sends the OTP over the External Host Interface (EHI) to the Program Manager.

² Your Wallet Usage Group can be viewed in Smart Client. If the token requestor is ApplePay, they populate the request with a score (Wallet device score and Wallet account score), which can be used to determine if further cardholder authentication is required. See Thredd Configuration Options.



6. If the cardholder selected the SMS option and you have requested that Thredd send the message on your behalf, then Thredd sends the OTP to the cardholder via SMS.

For other cardholder verification methods or where you have opted to send the SMS, the cardholder receives the OTP from your systems³.

7. The cardholder enters the OTP on their mobile device.

8. The token service provider validates the OTP.

Figure 4 above has been simplified to show the overall process. Token provisioning with authentication requires several messages between the card schemes (networks) that are a mixture of BASE I (ISO 8583) messages (for Visa and Mastercard) and APIs (for Visa only).

2.2.5 Decline (Red Flow)

When Thredd receives the token activation request (TAR) and we decline, we send a decline message to the Token Service Provider. This ends the token flow. The token requestor must request a new token.

2.2.6 Orange Flow

Orange Flow is for token requests from Apple that indicate if the request is high risk. Apple mandate that these requests must be authenticated using secure authentication methods or be declined. The Secure authentication methods are:

- In-App authentication where the application is tenured
- A call centre where additional fraud analysis is performed

Note: Orange Flow is set up during implementation, where you select the options on the PSF. Contact Implementations to discuss the best option for you.

In the event of a provisioning request being marked as Orange Flow, Thredd will return any of the above authentication methods that you have configured on the Payment Token Usage Wallet.

Note: In-App authentication cannot be returned if the provisioning requests started as In-App push provisioning. This is an Apple rule designed to stop a fraudulent actor with access to your app using it to start the flow and then authenticate themselves through the same channel.

Customers can choose to support secure authentication methods or to decline all requests flagged as Orange Flow. Apple recommend supporting secure authentication methods, however if you are not able to do this we have kept the possibility of declining all requests (In this scenario speak to Apple about getting your use case approved).

³ You must provide Thredd with the SMS text message to send to the customer. This message can contain dynamic fields. For details, check with your Implementation Manager. Thredd always sends the SMS to the phone number linked to the card on our systems (note that this may not be the same as the device which is being tokenised).



2.3 In-App Push Provisioning

In-App Push Provisioning enables a cardholder to request a token for their device directly from the Programme Manager's mobile app, removing the need to manually enter the PAN details into the mobile wallet Token Requestor's app.

Note: The cardholder must be logged into their account on their Programme Manager mobile app to be able to authenticate.

Since it originates from inside the mobile app, the Program Manager can pre-authenticate the cardholder through entry to their mobile app before a request for a token is sent to the Token Service Provider (Such as Visa or Mastercard). For information on the requirements for authenticating a cardholder during push provisioning, discuss with your mobile wallet Token Requestor.

Push provisioning requires you to share sensitive card data with the Token Service Provider using the mobile wallet Token Requestor. This data needs to be encrypted to standards of both the Token Service Provider and the mobile wallet Token Requestor. Thredd's role is to provide an encrypted payload that can be returned to your mobile app and passed into the relevant mobile wallet app.

Thredd provides an integrated solution through our In-App Provisioning endpoints to ensure that non PCI Compliance Level 1 customers can retrieve the PAN from the Thredd platform.

This integration enables you to retrieve the PAN and CVV from the Thredd platform. The data is encrypted and sent to the cardholder's mobile phone application to pass to the token requestor and then to the token service provider (Visa/Mastercard).

Note: Thredd's In-App Provisioning solution is also available to customers that are PCI compliant. Contact your Implementations Manager to ask about using this.

Figure 5 below describes the In-App Provisioning process.

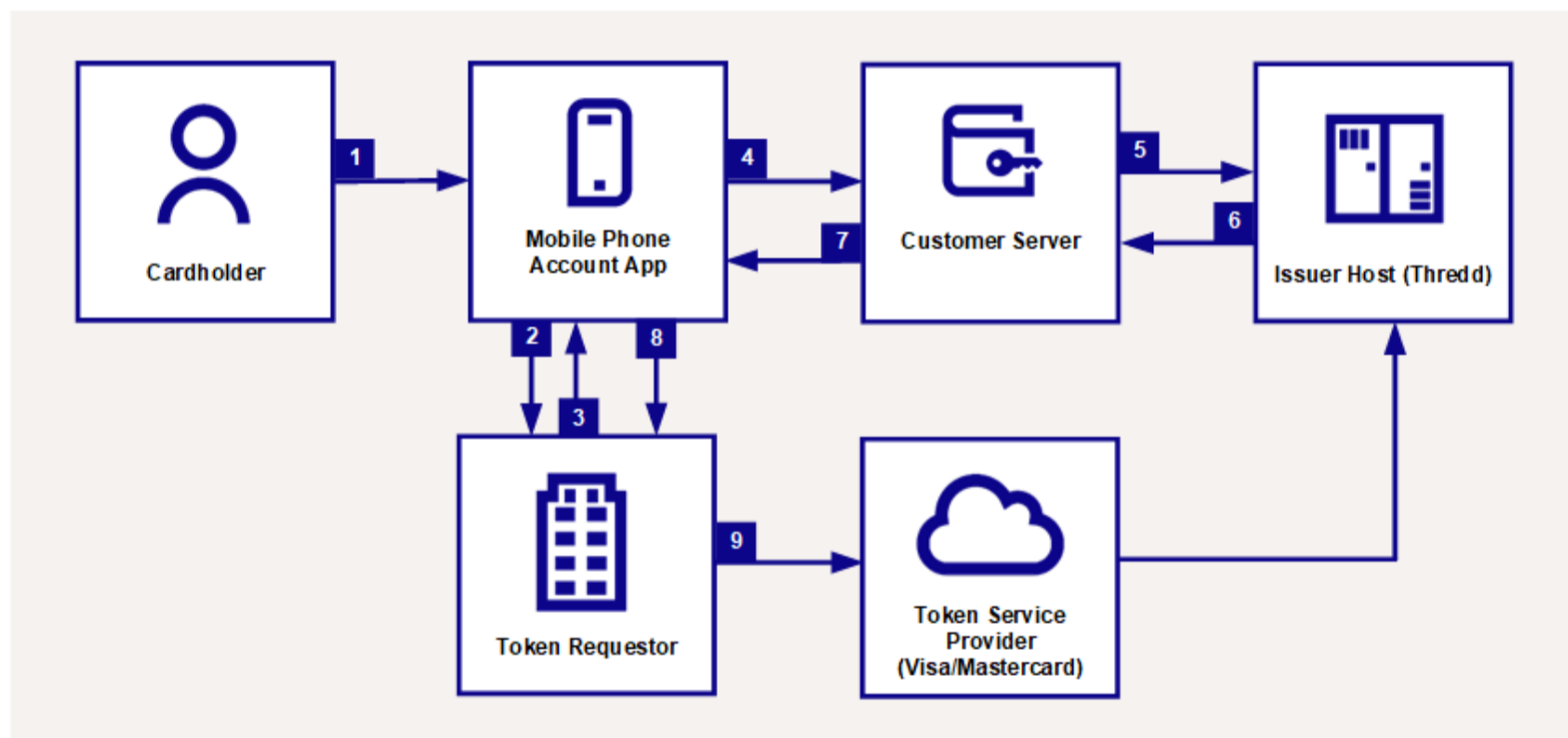


Figure 5: Integration for In-App Provisioning

1. The cardholder confirms the card to be added to their mobile phone application for your service.
2. Your mobile phone app requests encrypted card data for In-App Provisioning from the token requestor (e.g., Apple or Android).
3. The token requestor returns the data.
4. Your mobile app sends data to your server.
5. Using a valid REST API authorisation token, your server calls either the Thredd Apple, Google Wallet, or Samsung Pay endpoint.
6. Thredd creates an encrypted payload and returns it to your server.
7. The server sends the encrypted data to the mobile phone app.
8. The encrypted data is passed to the token requestor from the phone app.
9. The token requestor initiates provisioning with the Token Service Provider, which decrypts the card data and starts the token provisioning flow.



2.4 Token Requestors

The token requestor initiates the request to convert your cardholder's PAN into a token. There are two types of token requestors:

- Mobile Wallet Token Requestors - such as Apple and Android, who provide a token service via a smartphone or other mobile device that enables the cardholder to use their device for point of sale (POS) transactions
- Online Merchant Token Requestors - who tokenise a payment card for use in repeat or recurring payments on their website (such as Netflix, Domino's and PayPal). These are referred to by Card on File (COF) Token Requestors⁴.

As a Thredd customer, most of your implementation in a tokenisation service project will be focused on the Mobile Wallet Token Requestors, with whom you need to integrate directly.

You will also need to enable online merchant Token Requestors. You do not require a pre-existing relationship with the merchant. Since merchants replace the live PAN with a token, you do not need to store the PAN. The merchant sends only the token to the Token Service Provider who maps it back to the real PAN before sending to Thredd. This is done to improve card data security.

⁴ Card on File (COF) is also referred to by Mastercard as MDES for Merchants.



2.5 Visa Cloud Token Framework - Online Merchant Device Binding

Note: Mandatory for Visa only. Not applicable to Mastercard.

In October 2020, Visa launched the Visa Cloud Token Framework (CTF). This product is a precursor to supporting the EMVCo Secure Remote Commerce (SRC) functionality⁵. SRC aims to introduce a common e-checkout experience that cardholders will trust, called Click to Pay. CTF allows online Merchant Token Requestors to bind their previously created Card On File (COF) tokens with devices which they can authenticate belongs to their customer. The device binding process allows merchants to directly authenticate that the cardholder owns the device they are paying from.

How does it work?

The Online Merchant Token Requestors creates a COF token through the standard Token Provisioning flow (Green flow, without Authentication).

The token can then be bound to a device if the Online Merchant Requestor sends a follow up request to do so. During binding, the Online Merchant Token Requestors usually requires cardholder authentication, by sending an OTP to the cardholder. This cannot be done by in-app notification through an app (this is against the Visa rules and OTP standard). Methods such as SMS are still valid.

Note: When Thredd approves a device binding, the merchant can initiate authentication of the device at any stage. This means you may receive OTP messages (Activation Code Notifications) at any time over EHI and not just immediately following a TAR or Device Binding Request. These OTP messages must be sent to the cardholder. If configured, Thredd sends these via SMS on your behalf.

2.5.1 Binding an existing COF Token to a Device

Note that this is relevant to Visa only.

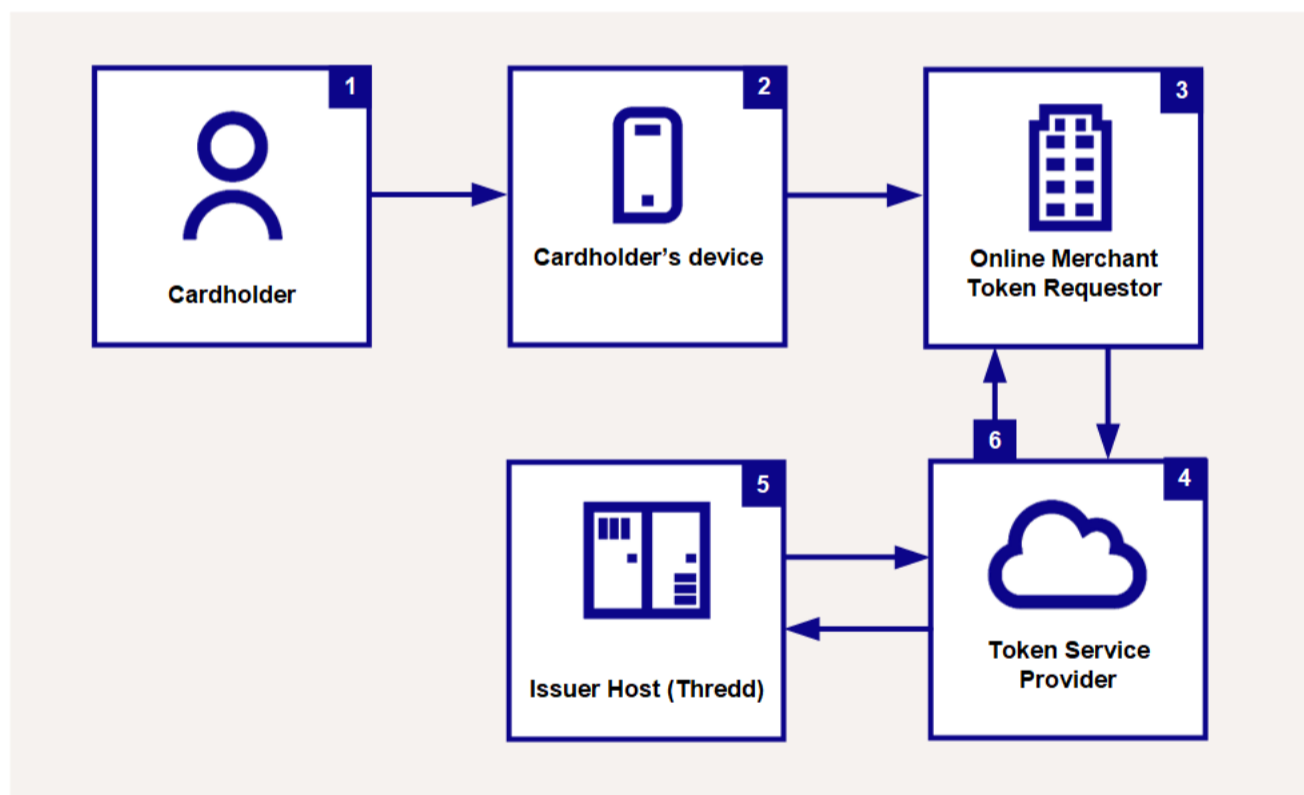


Figure 6: Device Binding Flow

1. The cardholder makes a purchase on their device.
2. The merchant identifies a new device on an existing Card on File (COF) token.
3. The merchant submits a device binding request.
4. The Token Service Provider (Visa/Mastercard) forwards the device binding request to Thredd.
5. Thredd provides a decision on the device binding request: Approve or Decline.
6. With approval, the merchant records the device binding for future purchases.

Cardholder authentication is not required in the following circumstances:

⁵ Precursor to Visa Secure EMVCo data.



- Authentication has already been performed (i.e., token is being push-provisioned; see [Push Provisioning](#))
- For an online merchant
- Based on the configuration for your Wallet Usage Group; see [Payment Token Usage Groups](#)

Note: Your Thredd Wallet Usage Group configuration is used to determine the appropriate flow to trigger⁶. Most Thredd Program Managers implement the approve with authentication flow.

⁶ Your Wallet Usage Group can be viewed in Smart Client. If the token requestor is ApplePay, they populate the request with a score (Wallet device score and Wallet account score), which can be used to determine if further cardholder authentication is required. See [Thredd Configuration Options](#).



2.6 Token Requestors

The token requestor initiates the request to convert your cardholder's PAN into a token. There are two types of token requestors:

- Mobile Wallet Token Requestors - such as Apple and Android, who provide a token service via a smartphone or other mobile device that enables the cardholder to use their device for point of sale (POS) transactions
- Online Merchant Token Requestors - who tokenise a payment card so that the token can be used for repeat payments or recurring payments on their website (e.g., such as Netflix, Domino's and PayPal). These are referred to by Card on File (COF) Token Requestors⁷.

As a Thredd customer, most of your implementation in a tokenisation service project will be focused on the Mobile Wallet Token Requestors, with whom you need to integrate directly.

You will also need to enable online merchant Token Requestors. You do not require a pre-existing relationship with the merchant. Since merchants replace the live PAN with a token, you do not need to store the PAN. The merchant sends only the token to the Token Service Provider who maps it back to the real PAN before sending to Thredd. This is done to improve card data security.

⁷ Card on File (COF) is also referred to by Mastercard as MDES for Merchants.



2.7 Visa Cloud Token Framework - Online Merchant Device Binding

Note: Mandatory for Visa only. Not applicable to Mastercard.

The device binding process allows merchants to directly authenticate that the cardholder owns the device they are paying from. The Online Merchant Token Requestor creates a COF token through the standard Token Provisioning flow, which can then be bound to a device. During binding, the Online Merchant Token Requestor usually requires cardholder authentication, by sending an OTP to the cardholder. This cannot be done by push notification through an app (this is against the Visa rules and OTP standard). Methods such as SMS are still valid.

Note: When Thredd approves a device binding, the merchant can initiate authentication of the device at any stage. This means you may receive OTP messages (Activation Code Notifications) at any time over EHI and not just immediately following a TAR or Device Binding Request. These OTP messages must be sent to the cardholder. If configured, Thredd sends these via SMS on your behalf.

2.7.1 Binding an existing COF Token to a Device

Note that this is relevant to Visa only.

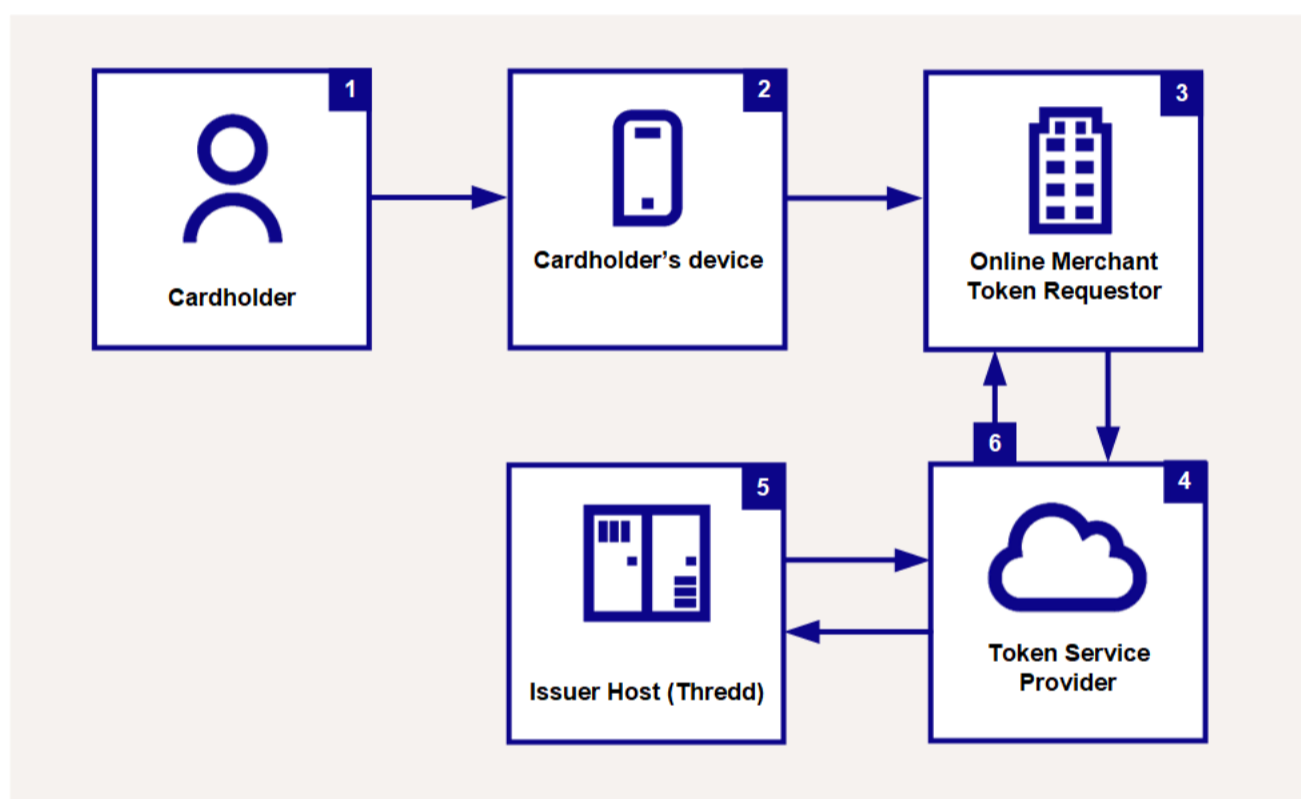


Figure 7: Device Binding Flow

1. The cardholder makes a purchase on their device.
2. The merchant identifies a new device on an existing Card on File (COF) token.
3. The merchant submits a device binding request.
4. The Token Service Provider (Visa/Mastercard) forwards the device binding request to Thredd.
5. Thredd provides a decision on the device binding request: Approve or Decline.
6. With approval, the merchant records the device binding for future purchases.



3 Transactions on a Token

Once the digital token (DPAN) has been created, it can be used in place of the card for payment authorisation transactions. Transactions on a token look like standard transactions on the card, but the payment token has additional data. Some of this data needs to be gathered and reported to token requestors such as Apple or Android.

3.1 Personalisation on a Device

Tokenisation of devices such as mobile phones and smart watches allows them to be used in the same way as physical cards. During tokenisation the mobile device is personalised. This is the process in which the device is marked with private data specific to that token and device. Personalisation is the same process as used on a physical card when chip data is added to the card prior to issuance.

Personalisation can either be done on the device or SIM card (known as Secure Element tokenisation) or in the cloud using Host Card Emulation (HCE).

- Apple Pay, which has access and control of the device and has pre-installed EMV chips that can be personalised, uses Secure Element (SE)
- Android, which is an operating system installed on various devices owned by other companies, uses HCE¹.

Other Mobile Wallet token requestors vary between SE and HCE and it is their decision which option is implemented. Thredd can support both SE and HCE mobile wallet token requestors.

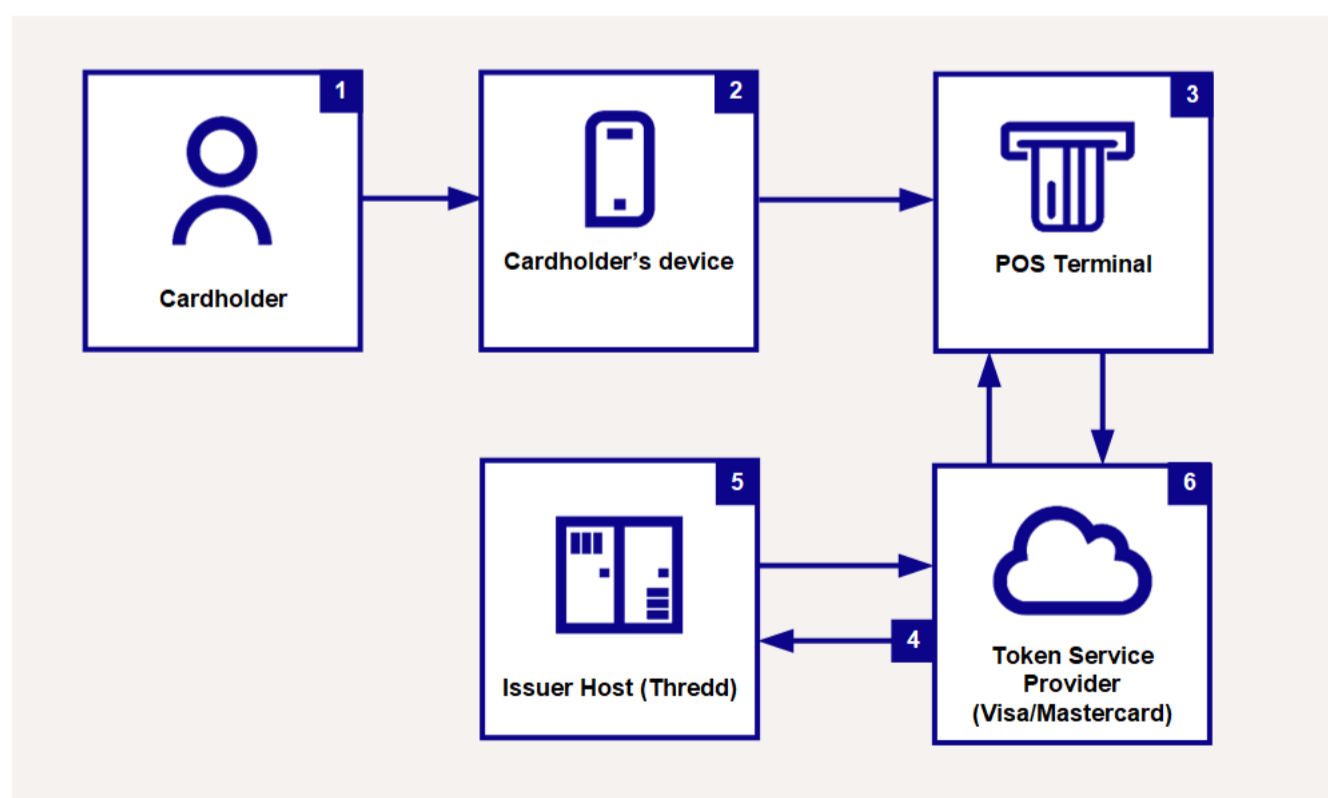
The data from the personalised device is transmitted to the Point of Sale (POS) terminal during an in-store transaction. The POS terminal then formats this into an authorisation request, which does not contain the real PAN, but uses the device token. This authorisation request is sent to Visa/Mastercard who maps the token back to the PAN before sending on to Thredd.

3.2 Visa/Mastercard Stand-In Processing

When setting up your programme configuration options on Visa or Mastercard, you must specify that they do not authorise Tokenisation Approval Requests (TARs) on behalf of Thredd. A TAR must always be generated by the token service provider and approved by Thredd². Thredd declines transactions on tokens that do not exist on the Thredd platform.

3.3 Making a Purchase using a Tokenised Device

Figure 7 below shows the flow for a tokenised device:



¹ An EMV program on the Android device manages transactions and communicates with a secure cloud host card emulator, where the keys for use for a transaction are generated.

² In scenarios where Visa/Mastercard can do Stand-In processing (STIP), they must not have any settings for your programme that pre-approves a tokenisation approval request (TAR); this must always be generated by the Token Requestor. If Thredd does not receive the TAR, we will decline transactions on the token.



Figure 8: Authorisation Request from Mobile Device

1. The cardholder makes an in-store purchase with an NFC-enabled tokenised device.
2. The device transmits personalised data to the POS terminal using the contactless interface.
3. The POS terminal generates the authorises the request, using the stored token (DPAN) and sends, via the merchant Acquirer, to the card scheme (Visa/Mastercard)³.
4. Visa/Mastercard maps the token back to the PAN (FPAN) and sends to Thredd for authorisation.
5. Thredd approves or declines the transaction.
6. Visa/Mastercard returns the authorisation response to the merchant. If approved, the merchant provides the goods to the cardholder.

³ The POS terminal treats the data received from the device in exactly the same way as data transmitted from a normal contactless card.



3.4 Making a Purchase using an Online website

For Online Merchant Token Requestors, the merchant uses the token associated with the cardholder to create and send an authorisation request. If it is the first time a card has been used with that merchant, then the tokenisation of the PAN will not have yet taken place; in this case the authorisation uses the real PAN initially and is then tokenised before storage. See *Figure 8* below.

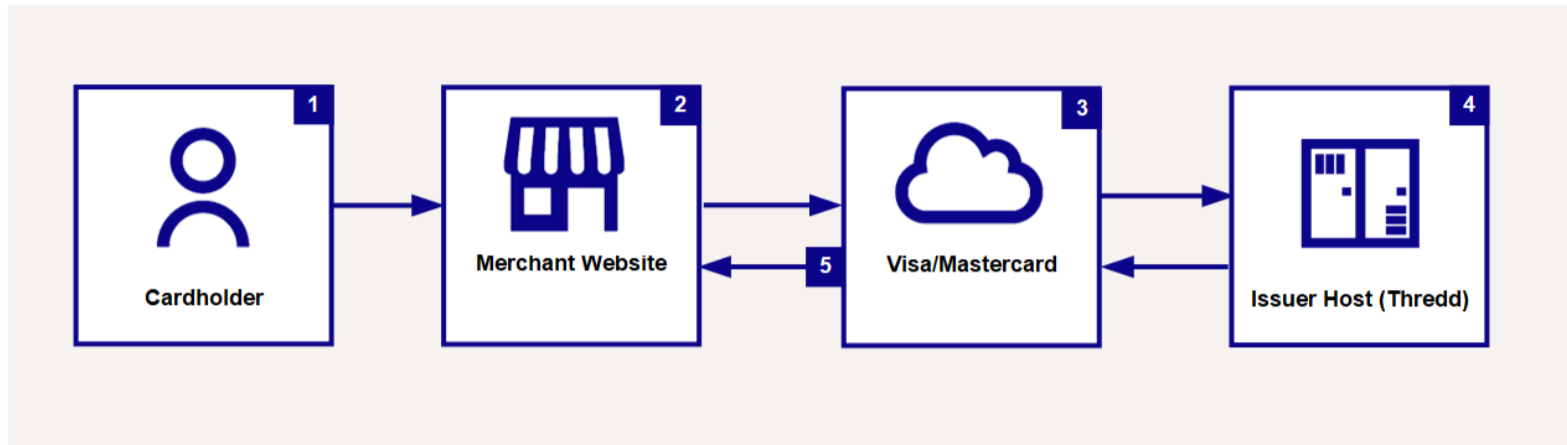


Figure 9: Authorisation Request from an Online website

1. The cardholder makes a payment on a merchant's website.
2. The merchant's systems generate the authorisation request using the stored token value and sends the request, via their Acquirer, to the card scheme (Visa/Mastercard).
3. Visa/Mastercard maps the stored token back to the PAN and sends to Thredd for authorisation.
4. Thredd approves or declines the transaction.
5. Visa/Mastercard returns the authorisation response to the merchant. If approved, the merchant provides the goods to the cardholder.



4 Implementing a Tokenisation Project

4.1 Steps in Enabling the Tokenisation Service

This section provides an indicative guide to the steps that you need to complete to enable the tokenisation service:

1. Sign up for the service
2. Complete requested forms
3. Configure your Thredd settings
4. Complete testing
5. Complete the Wallet Provider certification process

4.1.1 Step 1: Sign up for the Service

To enable the tokenisation service, you need to sign up with each of the following participants in the tokenisation flow:

- The Token Service Provider (Visa or Mastercard); for details, see the links below:
 - Visa Token Service: [Visa.co.uk](https://www.visa.co.uk): [Visa Token Service](#)
 - Mastercard Digital Enablement Service: [Mastercard.ie](https://www.mastercard.ie): [Digital Commerce-solutions](#)
- The Mobile Wallet Token Requestor(s) of your choice (e.g., Apple, Android, Fitbit, Samsung). for details, see the links below:
 - Apple Pay: developer.apple.com and [Apple Pay implementation](#)
 - Google Pay: pay.google.com
 - Samsung Pay: [samsung.com](https://www.samsung.com): [Samsung Pay](#)
- The Issuer Host (Thredd). Contact your Thredd Account Manager

You do not require a project or any pre-existing relationship with any Online Merchant Token Requestor (such as Netflix, PayPal, Amazon). As new Online Merchant Token Requestors are added to Visa/Mastercard, Thredd will continue to add these new merchants without further input from you to ensure you remain compliant.

Note: Thredd receives around 100-200 new Token Requestor updates a month from Visa. Mastercard add them to their generic MDES for merchant 3-digit token requestor code, so we do not need to update.

4.1.2 Step 2: Complete Requested Forms

Once you have signed up with Visa/Mastercard, your assigned Visa/Mastercard project manager or contact will send you a number of documents for completion. The Visa and Mastercard documents require Thredd input as they relate directly to the functionality on the Thredd platform. For details, check with your implementation Manager.

Note: Please ensure Thredd are involved in helping you complete the documents listed below.

Examples of Visa documents:

- Visa Token Service Program Information Form (PIF)
- Visa Token Service Digital Enrolment Form (DEF)

Examples of Mastercard documents:

- MasterCard BPMS guide (Parameter Worksheet)

Wallet provider documents:

- Complete the relevant wallet provider agreements and configuration forms. Thredd does not need to be involved in this process.



4.1.3 Step 3: Configure your Thredd settings

Once a project is open with Thredd, your Implementation Manager will work with you to understand how you want your token service programme to work.

You must complete the Thredd Digital Wallet Product Set Up Form (PSF) to confirm your tokenisation service configuration options. For details, see [Thredd Configuration Options](#).

If you want to receive tokenisation messages via the Thredd External Host Interface (EHI), in your Product Setup Form (PSF), ensure you tick the option to enable TAR transaction types. For details, see the [External Host Interface \(EHI\) Guide](#).

4.1.4 Step 4: Complete your internal testing

Complete internal pilot and pavement testing in the production environment. Get to know how your tokenisation app works and test against the wallet provider test scenarios.

Mastercard provide a Mastercard Test Facility (MTF), which can be used to test your MDES integration. MTF connects to the Thredd test environment. You can add your phone to MTF to send test tokenisation messages to Thredd. Please contact Mastercard to enable this service.

Visa provide a test service sandbox, which can be used to test outbound calls. For details, please contact Visa.

Note: Some integration work may be required on your end to integrate to the Mastercard or Visa test services. Many Thredd clients prefer to complete tests in the production environment.

4.1.5 Step 5: Complete the Wallet Provider certification process

Some Wallet providers, such as Apple Pay, have a formal certification process. Documentation for this is not available publicly, so Thredd recommends speaking to Apple Pay or your issuer in the first instance.

Google Pay does not have a formal certification process. Instead Google will send test scripts to you or your Issuer.

4.2 Implementing a Customised Token Service

Testing with Visa and Mastercard is not required if you are using the out of the box tokenisation service provided by Thredd.

Note: If you require non-standard functionality, you will need to raise a separate Thredd project (development work is required). Check with your account manager for details.



5 Tokenisation Configuration

This section provides details of the Thredd configuration options related to the tokenisation service.

5.1 Configuring Token Sub-Bin Ranges

IMPORTANT: For Program Managers implementing tokenisation on sub-BIN ranges, please note the restrictions in this section.

Visa systems can only support tokenising sub-BIN ranges split down to the 9th digit for Card on file (COF) and mobile wallet token requestors (such as ApplePay and GooglePay). If a Visa BIN is to be used for EU cross-border issuance and tokenisation, then only 10 countries can be used on that BIN before a new BIN is needed.

Mastercard and Thredd systems can support tokenisation down to the 10th digit of the sub-BIN range.

5.2 Card Usage Groups

If you are supporting tokenisation on your physical and virtual cards, please ensure that **Card Not Present (Manual Key Entry)** is enabled on your Card Usage Groups to allow your cards to be tokenised. You can specify this on the Thredd Product Setup Form (PSF). See the example below.

CARD USAGE RULES																																																																																																																							
Card Usage Group Name:		Thredd Code:																																																																																																																					
<table border="1"> <thead> <tr> <th>Allow? Y/N</th> <th>Card Acceptance Method (A)</th> </tr> </thead> <tbody> <tr><td>No</td><td>Unknown Acceptance Method</td></tr> <tr><td>No</td><td>Card Not Present (E-commerce)</td></tr> <tr><td>No</td><td>Card Not Present (Phone/Mail/Order)</td></tr> <tr><td>No</td><td>Card Not Present (Recurring)</td></tr> <tr><td>No</td><td>Card Not Present (Manual Key Entry)</td></tr> <tr><td>No</td><td>Mag Stripe transaction at Chip capable Terminal (Technical Fall Back)</td></tr> <tr><td>No</td><td>Mag stripe PAN entry - Common</td></tr> <tr><td>No</td><td>Chip PAN Entry - Offline PIN verification</td></tr> <tr><td>No</td><td>Chip PAN Entry - Online PIN verification</td></tr> <tr><td>No</td><td>Chip PAN Entry - Signature verification</td></tr> <tr><td>No</td><td>Chip PAN Entry - No Verification</td></tr> <tr><td>No</td><td>Cash withdrawal outside country of issue</td></tr> <tr><td>No</td><td>Cash withdrawal in currency other than card billing currency</td></tr> <tr><td>No</td><td>POS usage outside country of issue of a card</td></tr> <tr><td>No</td><td>POS usage in currency other than card billing currency</td></tr> <tr><td>No</td><td>Contactless EMV</td></tr> <tr><td>No</td><td>Manual Keyed Transaction at Chip capable Terminal</td></tr> <tr><td>No</td><td>Cardholder NOT present -Manual Key Entry</td></tr> <tr><td>No</td><td>Contactless MagStripe</td></tr> <tr><td>No</td><td>Card Not Present (Credential on File)</td></tr> <tr><td>No</td><td>Chip PAN Entry - no CVM Required</td></tr> <tr><td>No</td><td>Terminal indicates fallback Chip to Mag Stripe</td></tr> </tbody> </table>	Allow? Y/N	Card Acceptance Method (A)	No	Unknown Acceptance Method	No	Card Not Present (E-commerce)	No	Card Not Present (Phone/Mail/Order)	No	Card Not Present (Recurring)	No	Card Not Present (Manual Key Entry)	No	Mag Stripe transaction at Chip capable Terminal (Technical Fall Back)	No	Mag stripe PAN entry - Common	No	Chip PAN Entry - Offline PIN verification	No	Chip PAN Entry - Online PIN verification	No	Chip PAN Entry - Signature verification	No	Chip PAN Entry - No Verification	No	Cash withdrawal outside country of issue	No	Cash withdrawal in currency other than card billing currency	No	POS usage outside country of issue of a card	No	POS usage in currency other than card billing currency	No	Contactless EMV	No	Manual Keyed Transaction at Chip capable Terminal	No	Cardholder NOT present -Manual Key Entry	No	Contactless MagStripe	No	Card Not Present (Credential on File)	No	Chip PAN Entry - no CVM Required	No	Terminal indicates fallback Chip to Mag Stripe	<table border="1"> <thead> <tr> <th>Allow? Y/N</th> <th>Transaction Type (T)</th> </tr> </thead> <tbody> <tr><td>No</td><td>Purchase With Cashback (DE=09)</td></tr> <tr><td>No</td><td>Cash Advance (DE=17)</td></tr> <tr><td>No</td><td>Cash at ATM (DE=01)</td></tr> <tr><td>No</td><td>PIN Change ATM (DE=92 (M), 70(V))</td></tr> <tr><td>No</td><td>Balance Enquiry at ATM (DE=30)</td></tr> <tr><td>No</td><td>PIN Unblock via ATM (DE=91 (M), 72(V))</td></tr> <tr><td>No</td><td>Credits - Refunds (DE=20)</td></tr> <tr><td>No</td><td>Purchase of Goods & Services (DE=00)</td></tr> <tr><td>No</td><td>Visa Quasi- Cash (POS) transactions (DE=11)</td></tr> <tr><td>No</td><td>Credits Auth (DE=28)</td></tr> <tr><td>No</td><td>Original Credits (DE=26)</td></tr> <tr><td>No</td><td>Account Funding transaction (AFT) (DE=10)</td></tr> </tbody> </table>	Allow? Y/N	Transaction Type (T)	No	Purchase With Cashback (DE=09)	No	Cash Advance (DE=17)	No	Cash at ATM (DE=01)	No	PIN Change ATM (DE=92 (M), 70(V))	No	Balance Enquiry at ATM (DE=30)	No	PIN Unblock via ATM (DE=91 (M), 72(V))	No	Credits - Refunds (DE=20)	No	Purchase of Goods & Services (DE=00)	No	Visa Quasi- Cash (POS) transactions (DE=11)	No	Credits Auth (DE=28)	No	Original Credits (DE=26)	No	Account Funding transaction (AFT) (DE=10)	<table border="1"> <thead> <tr> <th>Allow? Y/N</th> <th>Verification Checks (V)</th> </tr> </thead> <tbody> <tr><td>No</td><td>Bypass Online PIN Check</td></tr> <tr><td>No</td><td>Bypass Expiry Date Check</td></tr> <tr><td>No</td><td>Bypass CVV2/CVC2 Check</td></tr> <tr><td>No</td><td>Blank CVV2 in Card not Present E-commerce</td></tr> <tr><td>No</td><td>Blank CVV2 in Card not Present Phone/Mail Order</td></tr> <tr><td>No</td><td>Blank CVV2 in Card not Present Recurring</td></tr> <tr><td>No</td><td>Blank CVV2 in Card not Present Manual Key Entry</td></tr> <tr><td>No</td><td>Allow Blank DE014</td></tr> <tr><td>No</td><td>Expiry date optional for Recurring Payments</td></tr> <tr><td>No</td><td>Bypass Card Status Check for Refund Authorisations</td></tr> </tbody> </table>	Allow? Y/N	Verification Checks (V)	No	Bypass Online PIN Check	No	Bypass Expiry Date Check	No	Bypass CVV2/CVC2 Check	No	Blank CVV2 in Card not Present E-commerce	No	Blank CVV2 in Card not Present Phone/Mail Order	No	Blank CVV2 in Card not Present Recurring	No	Blank CVV2 in Card not Present Manual Key Entry	No	Allow Blank DE014	No	Expiry date optional for Recurring Payments	No	Bypass Card Status Check for Refund Authorisations	<table border="1"> <thead> <tr> <th>Allow? Y/N</th> <th>Misc (M)</th> </tr> </thead> <tbody> <tr><td>No</td><td>If declined, force next EMV transaction online</td></tr> <tr><td>No</td><td>If Zero or negative balance, force next EMV transaction online</td></tr> <tr><td>No</td><td>Force next EMV transaction online</td></tr> <tr><td>No</td><td>Reset EMV counters to upper offline limits</td></tr> <tr><td>No</td><td>Transaction Alerts enabled</td></tr> <tr><td>No</td><td>Instant funding</td></tr> <tr><td>No</td><td>Override to allow International e-commerce</td></tr> <tr><td>No</td><td>Override to allow International Credential on File</td></tr> <tr><td>No</td><td>Instant Credit Gambling Payouts</td></tr> <tr><td>No</td><td>Faster Refunds Support</td></tr> </tbody> </table>	Allow? Y/N	Misc (M)	No	If declined, force next EMV transaction online	No	If Zero or negative balance, force next EMV transaction online	No	Force next EMV transaction online	No	Reset EMV counters to upper offline limits	No	Transaction Alerts enabled	No	Instant funding	No	Override to allow International e-commerce	No	Override to allow International Credential on File	No	Instant Credit Gambling Payouts	No	Faster Refunds Support
Allow? Y/N	Card Acceptance Method (A)																																																																																																																						
No	Unknown Acceptance Method																																																																																																																						
No	Card Not Present (E-commerce)																																																																																																																						
No	Card Not Present (Phone/Mail/Order)																																																																																																																						
No	Card Not Present (Recurring)																																																																																																																						
No	Card Not Present (Manual Key Entry)																																																																																																																						
No	Mag Stripe transaction at Chip capable Terminal (Technical Fall Back)																																																																																																																						
No	Mag stripe PAN entry - Common																																																																																																																						
No	Chip PAN Entry - Offline PIN verification																																																																																																																						
No	Chip PAN Entry - Online PIN verification																																																																																																																						
No	Chip PAN Entry - Signature verification																																																																																																																						
No	Chip PAN Entry - No Verification																																																																																																																						
No	Cash withdrawal outside country of issue																																																																																																																						
No	Cash withdrawal in currency other than card billing currency																																																																																																																						
No	POS usage outside country of issue of a card																																																																																																																						
No	POS usage in currency other than card billing currency																																																																																																																						
No	Contactless EMV																																																																																																																						
No	Manual Keyed Transaction at Chip capable Terminal																																																																																																																						
No	Cardholder NOT present -Manual Key Entry																																																																																																																						
No	Contactless MagStripe																																																																																																																						
No	Card Not Present (Credential on File)																																																																																																																						
No	Chip PAN Entry - no CVM Required																																																																																																																						
No	Terminal indicates fallback Chip to Mag Stripe																																																																																																																						
Allow? Y/N	Transaction Type (T)																																																																																																																						
No	Purchase With Cashback (DE=09)																																																																																																																						
No	Cash Advance (DE=17)																																																																																																																						
No	Cash at ATM (DE=01)																																																																																																																						
No	PIN Change ATM (DE=92 (M), 70(V))																																																																																																																						
No	Balance Enquiry at ATM (DE=30)																																																																																																																						
No	PIN Unblock via ATM (DE=91 (M), 72(V))																																																																																																																						
No	Credits - Refunds (DE=20)																																																																																																																						
No	Purchase of Goods & Services (DE=00)																																																																																																																						
No	Visa Quasi- Cash (POS) transactions (DE=11)																																																																																																																						
No	Credits Auth (DE=28)																																																																																																																						
No	Original Credits (DE=26)																																																																																																																						
No	Account Funding transaction (AFT) (DE=10)																																																																																																																						
Allow? Y/N	Verification Checks (V)																																																																																																																						
No	Bypass Online PIN Check																																																																																																																						
No	Bypass Expiry Date Check																																																																																																																						
No	Bypass CVV2/CVC2 Check																																																																																																																						
No	Blank CVV2 in Card not Present E-commerce																																																																																																																						
No	Blank CVV2 in Card not Present Phone/Mail Order																																																																																																																						
No	Blank CVV2 in Card not Present Recurring																																																																																																																						
No	Blank CVV2 in Card not Present Manual Key Entry																																																																																																																						
No	Allow Blank DE014																																																																																																																						
No	Expiry date optional for Recurring Payments																																																																																																																						
No	Bypass Card Status Check for Refund Authorisations																																																																																																																						
Allow? Y/N	Misc (M)																																																																																																																						
No	If declined, force next EMV transaction online																																																																																																																						
No	If Zero or negative balance, force next EMV transaction online																																																																																																																						
No	Force next EMV transaction online																																																																																																																						
No	Reset EMV counters to upper offline limits																																																																																																																						
No	Transaction Alerts enabled																																																																																																																						
No	Instant funding																																																																																																																						
No	Override to allow International e-commerce																																																																																																																						
No	Override to allow International Credential on File																																																																																																																						
No	Instant Credit Gambling Payouts																																																																																																																						
No	Faster Refunds Support																																																																																																																						
<p>Please note that all magnetic stripe ATM transactions are blocked by default. If you require this functionality enabled for your product then Thredd require explicit issuer sign-off.</p>																																																																																																																							

Figure 10: Card Usage Groups on the PSF

5.3 Payment Token Usage Groups

To configure your tokenisation usage groups, you need to fill in the Digital Wallet Product Set Up Form (PSF) and return to your Implementation Manager. The key configuration options, specific to the provisioning of a payment token, are found under two groups:

- Payment Token Usage Group: these are your default settings for all Token Requestors
- Payment Token Usage Wallet Groups: these are settings for specific Token Requestors (e.g., for Android and Apple)

To enable the payment token service, you will need at least one payment token usage group, which is set as the default group at a product level. See the example below.



WALLET CONFIGURATION

Wallet = Wallet provider such as Apple, Android, Samsung etc.
 Wallet Decision = A suggested decision from the wallet provider
 Device Score = A rating of the device (e.g. phone) assigned by the wallet provider. Thredd has no visibility of why the wallet provider selects these scores but will be based on things like device name being changed or multiple factory resets.
 Account Score = A rating of the account assigned by the wallet provider. Thredd has no visibility of why the wallet provider selects these scores but it is based on things like age of account, app activation, current fraud suspensions etc.

Payment Token Usage

Token Usage ID	Added by Thredd	Merchant Tokenisation	Yes
Token Usage Name		Click to Pay	Yes
Institution	Added by Thredd	Min. Tokens to Auth (min 1)	
Group Usage	Add details on 02. Card Usage Group	Min. Tokens to Decline (max 10)	

Payment Token Usage Wallet

Payment Token Usage	Fill in details on cell D14	Token provisioning existing setup	Yes
Token Usage Wallet	Apple	Approve Push Provisioning	Yes
Artwork		Orange Flow Option	Challenge
Dynamic	Select from drop down	Device Score Default	Default = 3
Default Decision	Default = A	Account Score Default	Default = 3
CVV2 Missing	Default = A	Wallet Decision is Authenticate	Default = A
AVS Missing	Default = A	Wallet Decision is Decline	Default = A
		Device Max Score Auth	Default = 3
		Device Max Score Decline	Default = 1
		Account Max Score Auth	Default = 3
		Account Max Score Decline	Default = 1
		Wallet Scores :	
		5 = Best	
		3 = Neutral	
		1 = Worst	

Authentication options *Please note Thredd can only send SMS, all other options are to be managed by the PM.

Mobile App	If Y: Enter Mobile App Ref
SMS OTP	If Y: Enter Mobile App Source Address
Call Centre	If Y: Enter Call Centre Tel Number
Automated Call Centre	If Y: Enter Call Centre Tel Number
Website URL	If Y: Enter Website URL
Email OTP	
Callback	Client to call the cardholder

Notification options

SMS Notifications	
Email Notifications	

Figure 11: Payment Token Usage Group

You then need to set up a payment token usage wallet group for each Token Requestor. See the example in Figure 12 below. Note that all COF Token Requestors are grouped into one payment token usage wallet group for ease of configuration.

Payment Token Usage	Fill in details on cell D14	Approve Push Provisioning	Select from drop down
Token Usage Wallet	Android	Device Score Default	Default = 3
If OWN wallet enter name:		Account Score Default	Default = 3
Artwork		Wallet Decision is Authenticate	Default = A
Dynamic	Select from drop down	Wallet Decision is Decline	Default = A
Default Decision	Default = A	Device Max Score Auth	Default = 3
CVV2 Missing	Default = A	Device Max Score Decline	Default = 1
AVS Missing	Default = A	Account Max Score Auth	Default = 3
		Account Max Score Decline	Default = 1
		Wallet Scores :	
		5 = Best	
		3 = Neutral	
		1 = Worst	

Authentication options *Please note Thredd can only send SMS, all other options are to be managed by the PM.

Mobile App	If Y: Enter Mobile App Ref
SMS OTP	If Y: Enter Mobile App Source Address
Call Centre	If Y: Complete SMS Setup Options tab
Automated Call Centre	If Y: Enter Call Centre Tel Number
Website URL	If Y: Enter Website URL
Email OTP	Email to be sent by the client
Callback	Client to call the cardholder

Notification options

SMS Notifications	
Email Notifications	

Figure 12: Payment Token Wallet Usage Group

For details of the fields in the Digital Wallet Product Set Up Form, see Thredd Configuration Options.

5.4 Thredd Configuration Options

The following table describes the available tokenisation service configuration options.

Settings in the Payment Token Usage Group apply for all Token Requestors, while settings in the Payment Token Wallet apply to individual Token Requestors (for example, Wallet Decision Auth).

Note: Online Merchant Token Requestors are provided as a single group (called MRCHTOKEN), so you cannot set different token logic for individual online merchant token requestors.¹

Parameter	Function	Suggestions
General Options		

¹ MRCHTOKEN is also referred to as M4M (by Mastercard) and Card on File (by Visa).



Parameter	Function	Suggestions
Group Usage	The default usage group that should be assigned to the wallet. (set at Payment Token Usage Group level)	If you want any different functionality for transactions on payment tokens than your existing physical or virtual cards, then specify a different usage group here to be used just for tokens. (For example, to prevent payment tokens to be used for ATMs with NFC enabled, or tokens to be used overseas.)
Artwork	The reference that Thredd should return to Visa/Mastercard for T&Cs and card art.	For Static card art: Leave blank (for Visa) or add a Mastercard reference. For Dynamic card art: Leave blank.
Dynamic	Whether the artwork is dynamic For details, see Dynamic vs. Static Card Art	If dynamic, select Yes. When you send Thredd a create card request (using the Thredd API) Thredd will pass the contents of the ProductRef field (SOAP web services) or the designId field (REST-based Cards API) to Visa/Mastercard.
Options for responding to a TAR request		
Default Decision	The default response that Thredd should provide when a TAR arrives.	Set to approve if you have scenarios where you want to approve a TAR without authentication. ² If you always want to authenticate the cardholder then set to authenticate. Set to decline if you are setting up a decline-only group. These groups are used to prevent individual cardholders from using the token service.
CVV2 Missing	The response code that Thredd should return if the CVV2 is missing from the TAR.	For Mobile Wallet tokenisation, set to approve or authenticate depending on your risk appetite. For the MRCHTOKEN wallet provider, this should be set to approve. Online merchant token activation requests must not decline for missing CVV2.
AVS Missing	The response code that Thredd returns if address data is missing from the TAR.	For Online Merchant tokenisation, this should be set to approve. For Mobile Wallet tokenisation, set to approve or authenticate depending on your risk appetite.
Wallet Decision Auth	The action Thredd should take if the incoming TAR from the token requestor recommends authenticate cardholder.	Set to approve or authenticate depending on your risk appetite.
Wallet Decision Decline	The action Thredd should take if the incoming TAR from the token requestor recommends decline.	Set to Approve, Authenticate or Decline depending on your risk appetite and the cardholder journey requirements.
Wallet and Device Scoring		
Wallet Device Score Default	The default score that Thredd should assign if there is no device score on the incoming TAR.	These scores are often missing (since many Token requestors do not provide a score). The default is 3, but you can set a higher or lower threshold, depending on your risk appetite. See Wallet Device and Account Scoring .
Wallet Account Score Default	The default score that Thredd should assign if there is no account score on the incoming TAR.	These scores are often missing if the Token requestor is not Apple. The default is 3, but you can set a higher or lower threshold, depending on your risk appetite.
Wallet Device	The maximum device score required to	The default is 3, but you can set a higher or lower threshold,

²Not applicable for Push Provisioning. Please see the setting for "Override approve-with-auth to Approve for in app provisioning" for further information on how to correctly set up push provisioning for Wallet Provider testing



Parameter	Function	Suggestions
Score Max Auth	trigger the Authenticate flow. Device scores are between 1 and 5.	depending on your risk appetite. See Wallet Device and Account Scoring .
Wallet Device Score Max Decline	The maximum device score required to trigger a Decline. Device scores are between 1 and 5.	Default is set to 1. Note that during internal pilots if you are adding and removing cards multiple times from Apple, the score may get low enough to cause declines.
Wallet Account Score Max Auth	The maximum wallet score to trigger the Authenticate flow. Wallet scores are between 1 and 5.	Usually set to 3 but you can set higher or lower threshold depending on your risk appetite.
Wallet account Score Max Decline	The maximum device score to trigger a Decline. Device scores are between 1 and 5.	Usually set to 1. However, during internal pilots if you are adding and removing cards multiple times from Apple the score may get low enough to cause declines.
Token frequency and overrides		
Min. Tokens to Auth	The number of tokens permitted before Thredd requests authentication. (set at Payment Token Usage Group level)	The number of existing tokens is specified in the incoming TAR request from the Token Service Provider.
Min. Tokens to Decline	The number of tokens permitted before Thredd declines further requests. (set at Payment Token Usage Group level only)	The number of existing tokens is specified in the incoming TAR request from the Token Service Provider.
Override-with-auth to Approve for in-app provisioning	Thredd can identify TARs where push provisioning has been used. In these requests the cardholder has already been authenticated so this option allows you to prevent a request for further authentication to be sent.	Always set Override Enabled for a better customer journey. If it is not enabled the cardholders will often need to authenticate twice. The override is required to pass Apple testing.
Default Wallet Provider Authentication	This relates to authentication on the payment token/DPAN. PSD2 requires cardholder authentication when: <ul style="list-style-type: none"> The transaction amount is over 50 EUR * The cumulative non-SCA value exceeds 150 EUR* More than five consecutive non-SCA transactions have been processed * The amount/value is configurable per client and currency Note: The Wallet provider should always handle the authentication and update the counters.	Options include: <ul style="list-style-type: none"> Authenticated - the payment token/wallet always does implicit cardholder authentication for each transaction performed on it. Not authenticated - no implicit cardholder authentication happens for transactions. (If PSD2 is enabled, then Thredd will track both contactless and e-commerce counters, and will request SCA if these limits are exceeded.) (This option can be set for an online merchant.) Do not apply PSD counters - the payment token/wallet does the PSD2 checking, and Thredd should not do any PSD2 checking for transactions. (This option should always be set for Mastercard. It is also recommended for Visa.)³ Note: PSD2 counters for physical cards is not affected by this setting. SCA counters for physical cards are a separate configuration parameter.
Options for cardholder authentication and token activation		
Activate: Call	The number to call if you want cardholders	Leave blank for no call centre, otherwise enter the phone number

³For Mastercard, Thredd do not receive the full authentication data to support this option. We also highly recommend you do not enable for Visa as this may lead to a poor customer journey (where the token is declined and the terminal prompts to insert a card).



Parameter	Function	Suggestions
Centre Tel Number	to be able to telephone a call centre to activate their payment token.	that Thredd should return to the token service provider. If you need different call centre numbers for different groups of cardholders, please set up a Payment Token Usage group for each number. Note: Your call centre staff can view the One-time passcode (OTP) required for activation in Smart Client. The OTP is also available via EHI or Thredd API.
Activate: Mobile App Reference	The name that the Wallet provider refers to your app as, and is displayed to the cardholder.	The name of the client that appears in the Apple or Google Pay app. This depends on your app build. Leave blank for no mobile app, otherwise enter the reference that Thredd returns to the token service provider.
App Source Address	The identifier that the network can use to tell the Wallet which app to send the cardholder to for verification.	<ul style="list-style-type: none"> • For Apple Wallet: Use the Adam ID of your app in the format of a string of 10 numbers. This must match the Adam ID added in MDES Manager. It can also be found by copying the numbers at the end of the clients app's App Store URL. For example, if the URL is <code>https://apps.apple.com/app/id1195168544</code>, the Adam ID is 1195168544. • For Google Wallet: Use the package name of the issuer's mobile app which is added in MDES Manager. For example: <code>com.mybank.bankingap</code>
Activate: Website URL	URL cardholders use to retrieve an OTP.	Enter the website URL you want cardholders to go to for their OTP.
Activate Email	Whether to activate email as an OTP delivery option	This option is required if you want email to be returned to the cardholder as an option for authentication; note that Thredd will not send the email directly to the cardholder and your systems will need to implement this: you can retrieve the OTP from EHI and handle with your own messaging and branding. If you are interested in Thredd sending emails directly, please raise with your Account Manager.
Activate SMS	Whether to activate SMS as an OTP delivery option.	If you want to send your own SMS, then enable this parameter, but do not configure a message.
Activate Call-back	Whether to activate call-back as an OTP delivery option	Thredd does not handle call-backs directly with a cardholder. If you want to provide this option, then your systems will need to retrieve the OTP from EHI and call your cardholder directly.
Token complete notification options		
Notify SMS	Whether you want Thredd to confirm via SMS to a cardholder that tokenisation is complete	Enable if required.

Note: For Thredd to return an Approve response code for a TAR (Green flow), all checks based on configuration and card must be approved. If only one check returns an authentication decision, then Thredd will request authentication (yellow flow)⁴; if only one check triggers a decline then Thredd will decline (red flow).

⁴ Excludes authentication for push-provisioned token requests, which only allow approve or decline responses.



5.4.1 Enabling Different Configuration Options per Card

Thredd provide the option to create multiple **Payment Token Usage Wallet Groups** and have different authentication options at a card level for different cards or card products. The Thredd Implementation Team set up these usage groups.

Below are some examples of why multiple Payment Token Wallet Usage Groups are used:

- To prevent individual cards from being tokenised, you can set up a “decline only” group (often done for fraud purposes).
- To set up different notification options such as different call centre numbers or to exclude SMS options from some clients.
- To set up different usage roles for tokenised cards, for example exclude MOTO payments or magnetic stripe card payments.
- To have different limits on the numbers of tokens that can be created before authentication and/or decline responses are sent.

How to change a card’s usage group

All token service products have a default Payment Token Usage Group. To change a card’s usage group using the Thredd API:

Using Web Services (SOAP)	Using Cards API (REST)
<ul style="list-style-type: none"> • When creating the card, use the Ws_CreateCard or when changing a card’s usage group at a later time, use the Ws_Change_Groups web service. • Enter the unique identifier into the <PaymentTokenUsageGroup> field. <p>For more information, see the Web Services Guide.</p>	<ul style="list-style-type: none"> • When creating the card, using the Create a Card endpoint, the default usage groups for the card product are linked to the card • You can use the Update Card Controls Group endpoint to change the card’s usage group <p>For more information, see the Cards API Website > Managing Card Control Groups.</p>

It is also possible to change a card’s usage group using Thredd Portal or Smart Client. For more information, see the [Thredd Portal Guide](#), or the [Smart Client Guide](#).

5.4.2 Dynamic vs. Static Card Art

There are two options to configure the card art that is displayed to the cardholder at the time of tokenisation:

- Static - the same artwork for the whole account range.
- Dynamic - artwork can vary at card level.

Configuration of artwork differs slightly between Visa and Mastercard.

Visa Configuration

You will need to configure your static card art per account range on the Visa Cardholder Metadata Manager (VCMM) online portal.

- To configure dynamic artwork, you must upload your card art options to VCMM against a [ProfileID](#). The [ProfileID](#) is always 32 characters long.
- When you create a card using the Thredd API, you must enter this Profile ID into the [ProductRef](#) field (SOAP [Card Create](#) web service) or the [designId](#) field (REST [Create Card](#) API).

Mastercard Configuration

Upload your artwork to the MDES Manager. Use the same product reference for your artwork as is used by your Card Manufacturer.

For static artwork, Thredd take the reference from your payment token wallet usage group. For dynamic artwork, Thredd take the reference from the [ProductRef](#) field (SOAP) or [designId](#) field (REST).

Note: the card art reference ([ProductRef](#) field for SOAP web services or [designId](#) field for REST-based Cards API) should be the same for both Visa/Mastercard and your Card Manufacturer.



5.4.3 Wallet Device and Account Scores

These scores are returned from Mobile Wallet Token Requestors such as Apple Pay, to reflect how trustworthy they consider the account and device. Scores are between 1-5. The higher the score, the more reliable the account or device is considered to be. 1 = least trusted, and 5 = most trustworthy.

You can use this score to determine how you want Thredd to respond to the Token Activation Request (TAR).

Example Thredd Settings

What is the maximum number which triggers approve with authentication (yellow flow)?

Wallet Device Score Max Auth = 3

What is the maximum number which triggers a decline (red flow)?

Wallet Device Score Max Decline = 1

If a device score of 4 is received, then Thredd approves. If a device score of 3 is received, then Thredd approves with authentication. If a device score of 1 is received, then Thredd declines.

5.4.4 Default Device and Account Scores

Some wallet providers do not return any device or account scores. In this case, you can configure Thredd to assign a default score that can be used in the Token Activation Request to determine how you want Thredd to respond to the Token Activation Request.

5.4.5 Device Binding Logic

Device binding requests follow a specific logic based on Visa’s best practice and mandates regarding how Thredd can process them so these settings cannot be changed. The processing flow is dependent on the reason code as follows:

- 3740 (Device binding request) - Always requests authentication
- 3749 (Device binding with FIDO intent) - Always requests authentication
- 3760 (Device binding - implicit green flow) - Always approval

Currently, Thredd device binding requests use the same authorisation logic as the Online Merchant token activation requests. If you want different authorisation logic between Online Merchant Token requests and Device Binding Requests, please raise with your Account Manager.

5.4.6 DPAN over FPAN Status

DPAN over FPAN status is an optional setting that specifies how Thredd should treat the Digital PAN (DPAN) and Financial PAN (FPAN) statuses during DPAN transactions:

- If DPAN over FPAN status is ENABLED - Thredd will check the DPAN status during a token transaction and disregard the underlying FPAN status
- If DPAN over FPAN status is DISABLED - Thredd will check BOTH the FPAN and DPAN statuses during token transactions

The default setting is disabled. To enable this setting, please raise a change request or speak to your Thredd Implementation Manager.

Note: We recommend that DPAN over FPAN status should not be enabled for Card on File token requestors. This should only be used for device-based tokens through wallet providers such as Apple and Android.

Changing the Token Status

The interaction between the card status and the payment token status depends on the type of payment token.

Payment Token Type	Authorisation Behaviour	Card Status Change
Wallet (for example, Apple Wallet or Google Wallet).	Authorisation response is determined by the status of the Wallet token being used.	Wallet token status is independent from the card status.

You can use the following Thredd APIs to change the status of your cards:



Using Web Services (SOAP)	Using Cards API (REST)
<ul style="list-style-type: none">• Ws_StatusChange - changes the status of a card (the FPAN status)• Ws_PaymentToken_StatusChange - changes the status of a digital payment token (the DPAN status) <p>Note that the card status is changed in both web services using the <NewStatCode> field. For more information, see the Web Services Guide.</p>	<ul style="list-style-type: none">• Update Card Status - changes the status of a card (the FPAN status)• Update Payment Token Status - changes the status of a digital payment token (the DPAN status) <p>For more information, see the Cards API Website.</p>



Impact on token transactions

If the DPAN over FPAN setting is enabled, then you must separately set the statuses of both the FPAN and DPAN using the Thredd API. See [Changing the FPAN and DPAN status](#).

If the DPAN over FPAN setting is disabled, then the DPAN status can be overridden in an authorisation by the FPAN status (as set using the Thredd API). This allows you to control the status used for authorisation using a single API.

Note: Do not use status code 41 if temporarily blocking a DPAN, since the Card Schemes (Networks) treat this as a permanent status. We recommend you use status code G1 instead, as this status is reversible.

5.5 Exchange of Keys

5.5.1 Visa Keys

Visa and Thredd exchange the following keys as part of the Visa Token service project. A set of Visa keys are unique per Bank Identification Number (BIN).

- Master Derivation Key (MDK) - used to validate chip cards. This key is per BIN and this process is the same as the key exchange for Visa STIP processing. Sharing the existing MDK with Visa is optional, as they can create a new one for tokenisation. Thredd can support either option.
- XPay Shared Secret (or HMAC key) - this is used by Visa to validate Visa inbound APIs that are sent by Thredd. This is shared per issuer via the Visa Developer Portal.
- XPay API Key - this is added to the URL by Thredd for Lifecycle Maintenance (outbound) APIs. This is shared per issuer via the Visa Developer Portal.
- JWE/JWS certificates - Private/Public key pairs used to sign or encrypt sensitive data within the APIs. The public certificates of these keys will be shared between Thredd and Visa during the implementation project.
- Key Identifier (KID) - Visa provides Thredd with the Key Identifier (KID) they assign to the Thredd JWE for Visa outbound APIs.
- Push Provisioning Shared Secret (or HMAC key) - this is used by Visa to validate Visa inbound payloads that are sent by Thredd. This is shared per issuer via the Visa Developer Portal.
- Push Provisioning API Key - this is added to the URL by Thredd for Push Provisioning payloads. This is shared per issuer via the Visa Developer Portal.
- Apple Web Services Key (WSDK) - This is used by Visa to validate Visa inbound payloads that are sent by Thredd. This key is already configured at Thredd and so does not need to be requested in a VTS project.

5.5.2 Mastercard Keys

The following keys are exchanged between Mastercard and Thredd.

- RSA Public key for TAV requests (Apple and Google) - This key is already configured at Thredd and so does not need to be requested in an MDES project
- RSA 2048-bit Public Key (Google) - A separate key is needed for each issuer who is onboarded. Mastercard need to share this with Thredd via the key management team email



5.6 Event Notifications

During token provisioning and status updates, Thredd receive multiple messages from the network. These can be sent to you through two routes: Webhooks or External Host Interface (EHI). The table below explains the events we can notify you about.

Message Type	EHI Processing Code	Webhook Event Code	Function	Usage
Token Activation Request (TAR)	330000	106	Request for a new token.	Counting token requests for reporting to the wallets. Identifying Orange flow requests (Webhooks only).
Activation Code Notification (ACN)	340000	107	Contains the OTP delivery message.	Client delivery of the OTP through a secure channel.
Token Complete Notification (TCN)	350000	108	Token created.	Counting successful token requests for reporting to the wallets. Confirm that the network has finished the tokenisation flow.
Token Event Notification (TEN)	360000	109	Token event notification (including activation).	Logging the status of a token to give cardholder correct options and information in the mobile app or website.
TLCM Failure Notification (Webhooks only)		110	Thredd was unable to update MDES or VTS with a DPANs new status.	Retrying any status changes that were not successfully processed by the Token Service Provider.

5.6.1 Webhooks

To receive tokenisation event notifications using Webhooks you can self-serve by using our Webhooks APIs. Using Webhooks, you can opt in to the message types that you want to receive.

You can utilise its REST API to create and update webhook endpoints, and specifying desired events for notification for the endpoints. Additionally, clients can retrieve historical or missed events on demand, update their details, and manage event subscriptions. Webhooks provide a robust platform for efficient event-driven interactions, retrieving notification details, subscribing to events, and re-sending notifications. The advantage of webhooks is that there is no persistent open connection to the system where you need to keep filtering for events that you are interested in. It is an asynchronous mechanism where you wait for the system to notify you. The HTTP payload will contain the details of the event.

Before using webhooks, you will need to setup a URL that can listen for any alerts. Contact the Thredd Application Support Team for help with this as your IP address must be whitelisted first. When the URL has been set up correctly, it must be registered using the Create Webhook endpoint.

For details of the Webhook message fields related to tokenisation, see [Introduction to Tokenisation Event Notifications](#).

5.6.2 EHI

You can receive tokenisation messages using Thredd's External Host Interface (EHI). To do so, in your Product Setup Form (PSF), tick the option to enable TAR transaction types. This sends all available token event notifications through the EHI (TAR, ACN, TCN, TEN). For more information, see the [EHI Guides](#).

For details of the EHI message fields related to tokenisation see [EHI Tokenisation Fields](#).



6 Implementing In App Provisioning

The following section details how to implement our in-app provisioning. There are four different types of in-app provisioning available:

- Apple Mastercard
- Apple Visa
- Google Mastercard
- Google Visa
- Samsung Visa



Step Three - Receive API response from Thredd

If the API request is successful, a 200 response is returned. See the following example.

```

{
  "encryptedPassData": "ew0KICAI TmFtZSI6ICII LA0KICAI Tm9uY2U0iAiIi-
wNCiAgIk5vb-
mNlU2l-
nbmF0dXJlI-
jogIiIsDQo-
gICJQcm-
ltYXJ5QWNjb3VudE51bWJl-
clByZWZpeCI6ICII LA0KICAI RW5jcml-
wdGVkUHJp-
bWFyeUFjY291b-
nR0dW1iZXIiOiB7DQo-
gICAgIkVuY3J5cHRlZFBheWxvYWQiOiB7DQo-
gICAgICAIUHVibGlsS2V5Rm-
luZ2VyUHJpb-
nQiOiAiIi-
wNCiAgICAgICJFb-
mNyeXB0ZWRLZXkiOiAiIi-
wNCiAgICAgICJPYWVwSGFz-
aGluZ0FsZ29y-
aXR0bSI6ICII LA0KICAgICAgIk12I-
jogIiIsDQo-
gICAgICAI RW5jcml-
wdGVkRGF0YSI6ICII dQo-
gICAg-
fQ0KICB9LA0KICAI TmV0d29y-
a05hbWUiOiAiTWZzdGVyY2FyZCI6ICII dQo-
"activationData": "ew0KICAI VmVyc2lvbiI6ICII i-
wNCiAgIktleUFsaWFzI-
jogIiIsDQo-
gICJTaW-
duYXR1cmVBbGd-
vcm-
l0aG0iOiAiUlNBLVNIQTII NiIsDQo-
gICJJb-
mNs-
dWRlZEZpZWxk-
c0luT3JkZXIiOiAiZGF0YVZh-
bGlkVW50aWxUaw1l-
c3RhbnB8YWNjb3VudE51bWJlcnxhY2NvdW50RXhwaXJ5IiwNCiAgICRhdGFwYXpZVFVudGlsVGltZXN0YW1wIjogIiIsDQo-
gICJTaWduYXR1cmUiOiAiIjogIiIsDQo-
"ephemeralPublicKey": "MzkzMDAwMDA="
}

```

Step Four - Send output data to Apple Wallet

To provision payments passes (cards) in your app, Thredd clients must initialise and invoke the `PKAddPaymentPassViewController` with a `PKAddPaymentPassRequestConfigurationObject`.



Implementing Visa with Apple Pay

The following section describes how to implement in-app provisioning for Visa with Apple Pay.

Pre-requisites

The pre-requisites for using Apple Pay with Visa are:

- An agreement with Apple to use the Apple Pay service.
- Your client iOS application requires com.apple.developer.payment-pass-provisioning entitlement.
- To exchange the Web Services key (WSDK) with Visa.

Step One - Retrieve input data from Apple Wallet

Extend `PKAddPaymentPassViewControllerDelegate`. This is where the certificate, nonce and nonce signature are provided by Apple to the Client application. Further details for objects that are required can be found on the [Apple Developer Documentation](#).

Step Two - Send API request to Thredd

When you have the data from Apple, use the Create Apple Wallet endpoint to send an API request to Thredd. See the following example request.

```
{
  "certificates": [
    "MIICYDCCA-
gagAwIBAgIUUKCe7rVr-
w/Sgst-
pLx4KPeLyRjCswCgYIKoZIzj0EAwIwADELMAKGA1UEBhMCVUsxDjAMBGNVBAgMBVN0YXRIMQ8wDQYDVQQHDAZMb25k-
b24xDzANBgNVBAoMB1RocmVkdEPMAG1UECwwGVGhyZWRkMRUYwFAYDQVQDDA1QUFRocmVkdZFN1YkNBMB4XDTE1MDIxMzE1MjAzMVoXDTM1MDIxMTE1MjAzMVowZzELMA-
KGA1UEBhMCVUsxDjAMBGNVBAgMBVN0YXRIMQ8wDQYDVQQHDAZMb25k-
b24xDzANBgNVBAoMB1RocmVkdEPMAG1UECwwGVGhyZWRkMRUYwFAYDQVQDDAxQUFRocmVkdZEx1YWYwWTATBgC-
qhkJOPQIBBg-
gqhkjOPQMBBwNCAATQE/gJiPV/b0xBy-
i4Fbr+UZbq7W5a7NmJlkXjIvBaiL5DoJQIM1-
maim-
cEXcuGxQg5ZGa78QVxZIC2QkUTBMYuko4GOMIGLMAKGA1UdEwQCAAwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBg-
grBgEFBQcDAjAPBgkqhkiG92NkBicEAgUAMB0GA1UdDgQWBBQ+D0z17sC8vBwb/g90X1uF2xmNhDAfBgNVHSMEGDAWgBTmiALoFDbCkZEInQd-
scUx+10NpVDABggqhkjOPQDDAgNIADBFAiEAtDZ3fLs2gcidvknZQs9uDoVv6/fyf5GQ4SkeddbSyaACICYczmRL0PFSgF905LKDOSVnLbs9TDK1RiLEELtH6ovN",
    "MIICYDCCA-
gagAwIBAgIUUKCe7rVr-
w/Sgst-
pLx4KPeLyRjCswCgYIKoZIzj0EAwIwADELMAKGA1UEBhMCVUsxDjAMBGNVBAgMBVN0YXRIMQ8wDQYDVQQHDAZMb25k-
b24xDzANBgNVBAoMB1RocmVkdEPMAG1UECwwGVGhyZWRkMRUYwFAYDQVQDDA1QUFRocmVkdZFN1YkNBMB4XDTE1MDIxMzE1MjAzMVoXDTM1MDIxMTE1MjAzMVowZzELMA-
KGA1UEBhMCVUsxDjAMBGNVBAgMBVN0YXRIMQ8wDQYDVQQHDAZMb25k-
b24xDzANBgNVBAoMB1RocmVkdEPMAG1UECwwGVGhyZWRkMRUYwFAYDQVQDDAxQUFRocmVkdZEx1YWYwWTATBgC-
qhkJOPQIBBg-
gqhkjOPQMBBwNCAATQE/gJiPV/b0xBy-
i4Fbr+UZbq7W5a7NmJlkXjIvBaiL5DoJQIM1-
maim-
cEXcuGxQg5ZGa78QVxZIC2QkUTBMYuko4GOMIGLMAKGA1UdEwQCAAwDgYDVR0PAQH/BAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBg-
grBgEFBQcDAjAPBgkqhkiG92NkBicEAgUAMB0GA1UdDgQWBBQ+D0z17sC8vBwb/g90X1uF2xmNhDAfBgNVHSMEGDAWgBTmiALoFDbCkZEInQd-
scUx+10NpVDABggqhkjOPQDDAgNIADBFAiEAtDZ3fLs2gcidvknZQs9uDoVv6/fyf5GQ4SkeddbSyaACICYczmRL0PFSgF905LKDOSVnLbs9TDK1RiLEELtH6ovN"
  ],
  "nonce": "c5846fb5",
  "nonceSignature": "4061d9d63ed34825f285d953274a6c5e06ebe011b-
f91d79660e1f7c6f6d21427abb3a62e6352e430abff987f6ec37e5dff9f3dbe40275156d03eeb594ab191d2792f37ef13ac528a65f56165c1d753463f"
}
```

Note: For more information on this endpoint, see [Apple In-App Push Provisioning](#) and [Apple In-App Push Provisioning - Field Descriptions](#).



Step Three - Receive API response from Thredd

If the API request is successful, a 200 response is returned. See the following example.

```
{
  "encryptedPassData": "ew0KICAI TmFtZSI6ICII LA0KICAI Tm9uY2Ui0iAiIi-
wNCiAgIk5vb-
mNlU2l-
nbmF0dXJlI-
jogIiIsDQo-
gICJQcm-
ltYXJ5QWVudE51bWJl-
clByZWZpeCI6ICII LA0KICAI RW5jcml-
wdGVkUHJp-
bWFyeUFjY291b-
nR0dW1iZXIi0iB7DQo-
gICAgIkVuY3J5cHRlZFBheWxvYWQi0iB7DQo-
gICAgICAIUHVibGlsS2V5Rm-
luZ2VyUHJpb-
nQi0iAiIi-
wNCiAgICAgICJFb-
mNyeXB0ZWRLZXki0iAiIi-
wNCiAgICAgICJPYWVwSGFz-
aGluZ0FsZ29y-
aXRobSI6ICII LA0KICAgICAgIk12I-
jogIiIsDQo-
gICAgICAI RW5jcml-
wdGVkRGF0YSI6ICII dQo-
gICAg-
fQ0KICB9LA0KICAI TmV0d29y-
a05hbWUi0iAiTWfzdGVyY2FyZCI sDQogICJQcm9kdWN0VHlwZSI6ICJERUZBVUxUX01BU1RFUknBUkQi LA0KICAI VmVyc2lvbiI6ICIxI g0KfQ==",
  "activationData": "ew0KICAI VmVyc2lvbiI6ICII 0Ii-
wNCiAgIktleUFsaWZlI-
jogIiIsDQo-
gICJTaW-
duYXR1cmVBbGd-
vcm-
l0aG0i0iAiUlNBLVNIQTII1NiIsDQo-
gICJJb-
mNs-
dWRlZlZpZWxk-
c0luT3JkZXIi0iAiZGF0YVZh-
bGlsVW50aWxUaw1l-
c3RhbnB8YWNjb3VudE51bWJlcnxhY2NvdW50RXhwaXJ5IiwNCiAgIkRhdGFwYXpZVudGlsVGltZXN0YW1wIjogIiIsDQogICJTaWduYXR1cmUi0iAiI g0KfQ==",
  "ephemeralPublicKey": "MzkzMDAwMDA="
}
```

Step Four - Send output data to Apple Wallet

To provision payment passes (cards) in your app, Thredd clients must initialise and invoke the [PKAddPaymentPassViewController](#) with a [PKAddPaymentPassRequestConfigurationObject](#).



Implementing Mastercard with Google Pay

The following section describes how to implement in-app provisioning for Mastercard with Google Pay.

Pre-requisites

The pre-requisites for using Mastercard with Google Pay are:

- An agreement with Google which grants permission to use the Google Push Provisioning API, and permits you to share all required data elements.
- Your apps must be added to the allow list for Google's Push Provisioning. See [API - Push Provisioning API Access](#) for more information.

Note: You must sign up and request access to view Google's documentation.

- Integrate the **Add to Google Pay** button into your mobile app.
- Obtain the Mastercard RSA 2048-bit Public Key and share with Thredd.
- Upload an RSA public key for TAV generation into the Key Management Portal application

Step One - Send API request to Thredd

When you have the data from Google, use the endpoint to send an API request to Thredd. Note if the billing object is left blank then the address associated with the publicToken for the card will be used instead.

```
{
  "billing": {
    "line1": "32 Eastern Drive",
    "line2": "Thurcroft",
    "city": "Sheffield",
    "countrySubdivision": "South Yorkshire",
    "postalCode": "S25 1AA",
    "country": "GBR"
  }
}
```

Note: For more information on this endpoint, see [Google In-App Push Provisioning](#) and [Google In-App Push Provisioning - Field Descriptions](#).

Step Two - Receive API response from Thredd

If the API request is successful, a 200 response is returned. See the following example.

```
{
  "opaquePaymentCard": "string",
  "last4digits": "string",
  "name": "string",
  "network": "Mastercard",
  "address": {
    "line1": "string",
    "line2": "string",
    "city": "string",
    "countrySubdivision": "string",
    "postalCode": "string",
    "country": "yIa"
  }
}
```

Step Three - Send API response to Google Push Provisioning API

After receiving a successful response, the information should be shared with Google using their Push Provisioning API.



Implementing Visa with Google Pay

The following section describes how to implement in-app provisioning for Visa with Google Pay.

Pre-requisites

The pre-requisites for using Visa with Google Pay are:

- An agreement with Google which grants permission to use the Google Push Provisioning API, and permits you to share all required data elements.
- Your apps must be added to the allow list for Google's Push Provisioning. See [API - Push Provisioning API Access](#) for more information.

Note: You must sign up and request access to view Google's documentation.

- Integrate the **Add to Google Pay** button into your mobile app.
- Exchange API Key with Visa and share with Thredd.
- Exchange shared secret with Visa and share with Thredd.

Step One - Send API request to Thredd

When you have the data from Google, use the endpoint to send an API request to Thredd. See the following example request.

```
{
  "clientWalletProvider": "string",
  "clientWalletAccountID": "string",
  "clientDeviceID": "string",
  "clientAppID": "string",
  "billing": {
    "line1": "string",
    "line2": "string",
    "city": "string",
    "countrySubdivision": "string",
    "postalCode": "string",
    "country": "gQc"
  }
}
```

Note: For more information on this endpoint, see [Google In-App Push Provisioning](#) and [Google In-App Push Provisioning - Field Descriptions](#).

Step Two - Receive API response from Thredd

If the API request is successful, a 200 response is returned. See the following example.

```
{
  "opaquePaymentCard": "string",
  "last4digits": "string",
  "name": "string",
  "network": "Mastercard",
  "address": {
    "line1": "string",
    "line2": "string",
    "city": "string",
    "countrySubdivision": "string",
    "postalCode": "string",
    "country": "yIa"
  }
}
```

Step Three - Send API response to Google Push Provisioning API

After receiving a successful response, the information should be shared with Google using their Push Provisioning API.



Implementing Visa with Samsung Pay

The following section describes how to implement in-app provisioning for Visa with Samsung Pay. This is similar to implementing Google pay for Visa, with some differences to the pre-requisites.

Pre-requisites

The pre-requisites for using Visa with Samsung Pay are:

- An agreement with Samsung which grants permission to use the Samsung Push Provisioning API, and permits you to share all required data elements.
- Integrate the **Add to Samsung Pay** button into your mobile app.
- Exchange API Key with Visa and share with Thredd.
- Exchange shared secret with Visa and share with Thredd.

Step One - Send API request to Thredd

When you have the data from Samsung, use the endpoint to send an API request to Thredd. See the following example request.

```
{
  "clientWalletProvider": "string",
  "clientWalletAccountID": "string",
  "clientDeviceID": "string",
  "clientAppID": "string",
  "billing": {
    "line1": "string",
    "line2": "string",
    "city": "string",
    "countrySubdivision": "string",
    "postalCode": "string",
    "country": "gQc"
  }
}
```

Note: For more information on this endpoint, see [Samsung In-App Push Provisioning](#) and [Samsung In-App Push Provisioning - Field Descriptions](#).

Step Two - Receive API response from Thredd

If the API request is successful, a 200 response is returned. See the following example.

```
{
  "opaquePaymentCard": "string",
  "last4digits": "string",
  "name": "string",
  "network": "Visa",
  "address": {
    "line1": "string",
    "line2": "string",
    "city": "string",
    "countrySubdivision": "string",
    "postalCode": "string",
    "country": "yIa"
  }
}
```

Step Three - Send API response to Samsung Push Provisioning API

After receiving a successful response, the information should be shared with Samsung using their Push Provisioning API.



7 Click to Pay

Click to Pay is a service that provides customers with a frictionless online experience during checkout. Issuers can register for this service on behalf of their customers. A merchant can display details of Click to Pay-registered cards by using only the cardholder's name and email/phone number during checkout. Thredd approves all Click to Pay tokenisation requests if Click to Pay is selected on your PSF.

Click to Pay uses the token instead of the card PAN at checkout, and avoids manual card entry for the cardholder. It also means there is a lower security risk, as the merchant never sees the cardholder's card details.

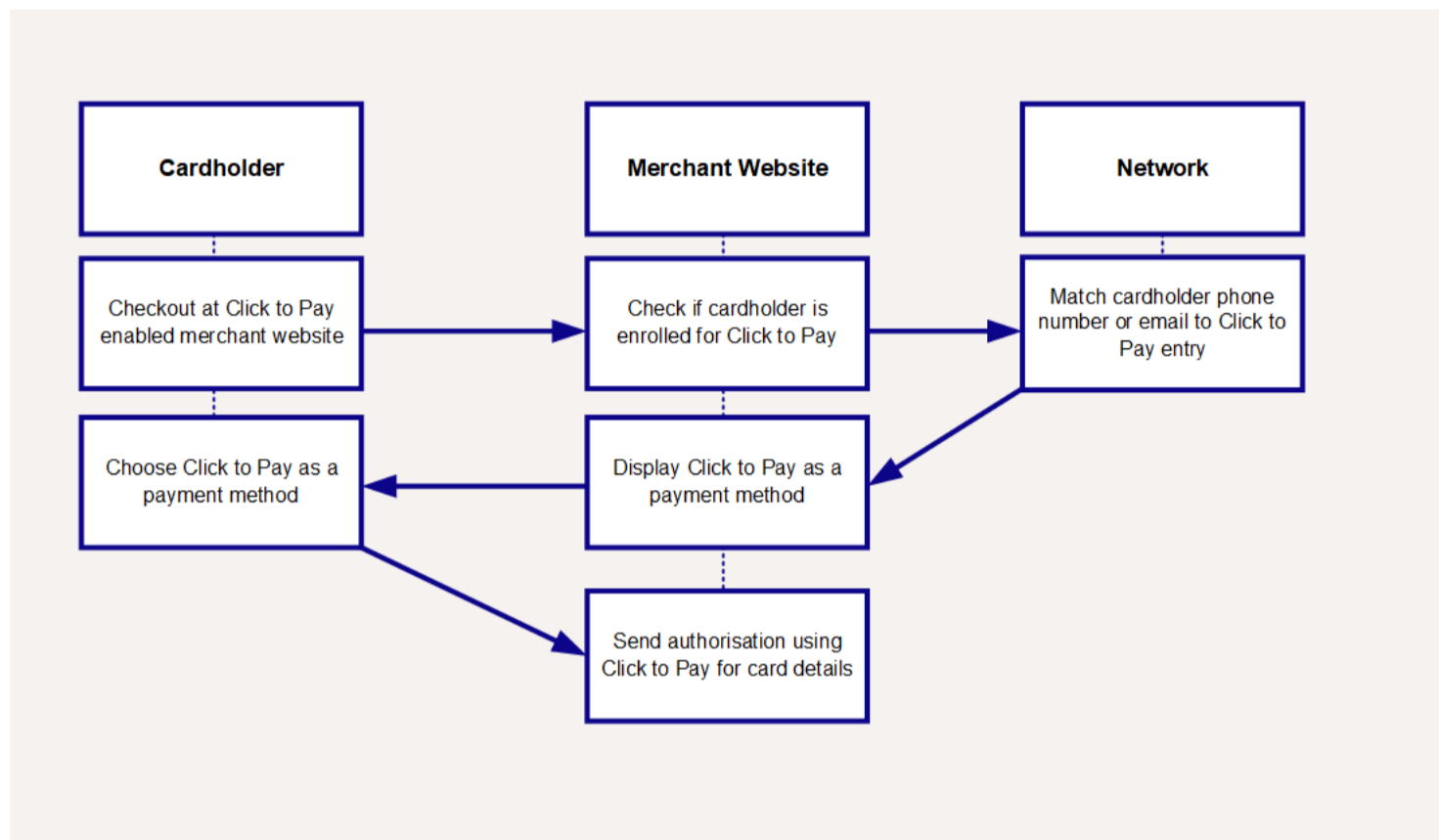


Figure 13: Flow diagram displaying how Click to Pay works

You can enrol cardholders onto Click to Pay using the Enroll Data endpoint. This endpoint enables you to register the new cardholder, phone number, email, and card linked to their token. If you want to add additional details to a previously enrolled card, you can use the Enroll Payment Instruments endpoint. Note you must register the cardholder using the Enroll Data endpoint before you can use the Enroll Payment Instruments endpoint.

Enroll Data

You can register a cardholder to use Click to Pay by making a POST request to the Enroll Data endpoint. For example:

```
{{base-url}}/ctp/api/v1/enrolldata
```

The POST body should include the details of the customer and their publicToken. The below is an example of what the body should look like:

```
{
  "customers": [
    {
      "customerDetails": {
        "billingAddress": {
          "city": "San Francisco",
          "state": "CA",
          "country": "USA",
          "postalCode": "94105",
          "addressLine1": "1000 Market Street",
          "addressLine2": "Building 56",
          "addressLine3": "Suite 101"
        },
        "customerReferenceId": "ImACustomer",
        "email": "jbloggs@email.com",
        "phone": "16504005555",
        "firstName": "John",

```



```
    "middleName": "Robert",
    "lastName": "Bloggs",
    "nationalIdentifiers": [
      {
        "type": "PASSPORT",
        "value": "A123456"
      }
    ],
  },
  "pubTokens": [
    {
      "value": "123456789"
    }
  ]
}
```

If successful, a 202 response is returned with the unique requestTraceId. See the following example of a successful response.

```
{
  "requestTraceId": "351562ba-83cf-11ee-b962-0242ac120002"
}
```

Note: For more information on the Enroll Data endpoint, see [Enroll Data](#) and [Enroll Data - Field Descriptions](#).

Enroll Payment Instruments

IMPORTANT: The Enroll Payment Instruments endpoint is for Visa cards only.

You can add more Click to Pay details to a card by making a POST request to the Enroll Payment Instruments endpoint. For example:

```
{{base-url}}/ctp/api/v1/enrollpaymentinstruments
```

The POST body should include the details of the customer, their publicToken and their billing address. The below is an example of what the body should look like.

```
{
  "billingAddress": {
    "city": "San Francisco",
    "state": "CA",
    "country": "USA",
    "postalCode": "94105",
    "addressLine1": "1000 Market Street",
    "addressLine2": "Building 56",
    "addressLine3": "Suite 101"
  },
  "customerReferenceId": "ImACustomer",
  "pubTokens": [
    {
      "value": "123456789"
    }
  ]
}
```

If successful, a 202 response is returned with the unique requestTraceId. See the following example of a successful response.

```
{
  "requestTraceId": "351562ba-83cf-11ee-b962-0242ac120002"
}
```

Note: For more information on the Enroll Payment Instruments endpoint, see [Enroll Payment Instruments](#) and [Enroll Payment Instruments - Field Descriptions](#).

Get Payment Instrument Information

IMPORTANT: The Get Payment Instrument Information endpoint is for Mastercard only.



The Get Payment Instrument Information endpoint enables you to return information on a payment instrument for a card enrolled onto Click to Pay. A payment instrument refers to the method or device through which a consumer can make a payment using the Click to Pay service.

You can return payment instrument information by making a GET request to the Get Payment Instrument Information endpoint, including the `customerReferenceId` and `publicToken`. For example:

```
{{base-url}}/ctp/api/v1/paymentinstrument/{customerReferenceId}/{publicToken}
```

If successful, a 200 response is returned with details of the payment instrument. For example.

```
{
  "intent": {
    "type": "PRODUCT_CODE",
    "value": "CLICK_TO_PAY"
  },
  "paymentInstrument": {
    "type": "CARD",
    "cardType": "CREDIT",
    "status": "ACTIVE",
    "issuerName": "Bank A",
    "nameOnCard": "John Doe",
    "accountNumber": "4111111145551140",
    "tokenDetails": {
      "tokenReferenceId": "DM4MMC1US000000a8513863213e4f1b99e4d016b4f61624"
    },
    "billingAddress": {
      "city": "San Francisco",
      "state": "CA",
      "country": "USA",
      "postalCode": "94105",
      "addressLine1": "1000 Market Street",
      "addressLine2": "Building 56",
      "addressLine3": "Suite 101"
    },
    "expirationDate": "2030-01",
    "paymentAccountReference": "5001DW4DS6FRSHVXVBC7HVP6JYAS6"
  }
}
```

Note: For more information on the Get Payment Instrument Information endpoint, see [Get Payment Instrument Data](#) and [Get Payment Instrument Information - Field Descriptions](#).

Manage Payment Instrument

The Manage Payment Instrument endpoint enables you to update existing payment instrument details, specifically the instruments billing address, created from the Enroll Payment Instrument endpoint.

You can manage a payment instrument by making a PUT request to the Manage Payment Instrument endpoint. For example:

```
{{base-url}}/ctp/api/v1/paymentinstrument
```

The Manage Payment Instrument request replaces the whole payment instrument object, so all fields are required in the request body. See the below example of a request:

```
{
  "customerReferenceId": 1234,
  "pubToken": 275797862,
  "billingAddress": {
    "city": "San Francisco",
    "state": "CA",
    "country": "USA",
    "postalCode": 94105,
    "addressLine1": "1000 Market Street",
    "addressLine2": "Building 56",
    "addressLine3": "Suite 102"
  },
}
```

If successful, a 202 response is returned with a unique `requestTraceld` when the card network is VISA, and a unique `srcCorrelationId` when the card network is Mastercard.



```
{
  "requestTraceId": "351562ba-83cf-11ee-b962-0242ac120002"
}
```

```
{
  "srcCorrelationId": "b17b686c-4759-4ade-a3b1-6b29c61b6774"
}
```

Note: For more information on the Manage Payment Instrument endpoint, see [Manage Payment Instrument](#) and [Manage Payment Instrument - Field Descriptions](#).

Manage Consumer Information

The Manage Consumer Information endpoint enables you to update consumer information created from the Enroll Data endpoint.

You can update consumer information by making a PUT request to the Manage Consumer Information endpoint. For example:

```
{{base-url}}/ctp/api/v1/consumerinformation
```

The request must consist of all fields, including any fields that do not require updating. If a field is missing from the request, the request fails.

See the below example of a request body.

```
{
  "customerDetails": {
    "nationalIdentifiers": [
      {
        "type": "PASSPORT",
        "value": "A123456"
      }
    ],
    "customerReferenceId": "63421837-d597-4f0f-89e4-930c1a7b9e85",
    "email": "alex123@hotmail.com",
    "phone": "16504005555",
    "firstName": "Alex",
    "middleName": "Robert",
    "lastName": "Miller",
    "countryCode": "USA"
  },
  "cardNetwork": "Visa"
}
```

If successful, a 202 response is returned with a unique `requestTraceId` when the card network is VISA, and a unique `srcCorrelationId` when the card network is Mastercard.

```
{
  "requestTraceId": "351562ba-83cf-11ee-b962-0242ac120002"
}
```

```
{
  "srcCorrelationId": "b17b686c-4759-4ade-a3b1-6b29c61b6774"
}
```

Note: For more information on the Manage Consumer Information endpoint, see [Manage Consumer Information](#) and [Manage Consumer Information - Field Descriptions](#).

Delete Consumer Information

You can remove consumer information by making a DELETE request to the Delete Consumer Information endpoint

```
{{base-url}}/ctp/api/v1/consumerinformation/{customerReferenceId}
```

If successful, a 202 response is returned with a unique `requestTraceId` when the card network is VISA, and a unique `srcCorrelationId` when the card network is Mastercard.

```
{
  "requestTraceId": "351562ba-83cf-11ee-b962-0242ac120002"
}
```



```
{  
  "srcCorrelationId": "b17b686c-4759-4ade-a3b1-6b29c61b6774"  
}
```

Note: For more information on the Delete Consumer Information endpoint, see [Delete Consumer Information](#).

Delete Payment Instrument

You can delete a payment instrument by making a DELETE request to the Delete Payment Instrument endpoint. For example

```
{{base-url}}/ctp/api/v1/paymentinstrument/{customerReferenceId}/{publicToken}
```

If successful, a 202 response is returned with a unique `requestTraceId` when the card network is VISA, and a unique `srcCorrelationId` when the card network is Mastercard

```
{  
  "requestTraceId": "351562ba-83cf-11ee-b962-0242ac120002"  
}
```

```
{  
  "srcCorrelationId": "b17b686c-4759-4ade-a3b1-6b29c61b6774"  
}
```



8 Token Provisioning Message Flows

For token provisioning there are several messages that are sent between the Token Service Provider and Thredd. These are a mixture of ISO 8583 (for VDEP and MDES) and JSON API (for VDEP only) formats. All ISO 8583 messages and One Time Passwords (OTPs) obtained are sent over EHI to the Program Manager¹.

Note: All EHI messages in the Token Provisioning flow (TAR/TEN/TCN) are advices only. This means that for all EHI modes you are not able to authorise TAR, TEN or TCN advices. You can use these messages to confirm to the cardholder the tokenisation status. They should never be used for payment authorisation approval or decline decisions.

Figures 12-15 below describe the Visa and Mastercard messages that are received for token provisioning requests. Note that since these are asynchronous messages, it is possible they may arrive out of sequence.

8.1 Message flow for Mastercard Token Provisioning (Green Flow)

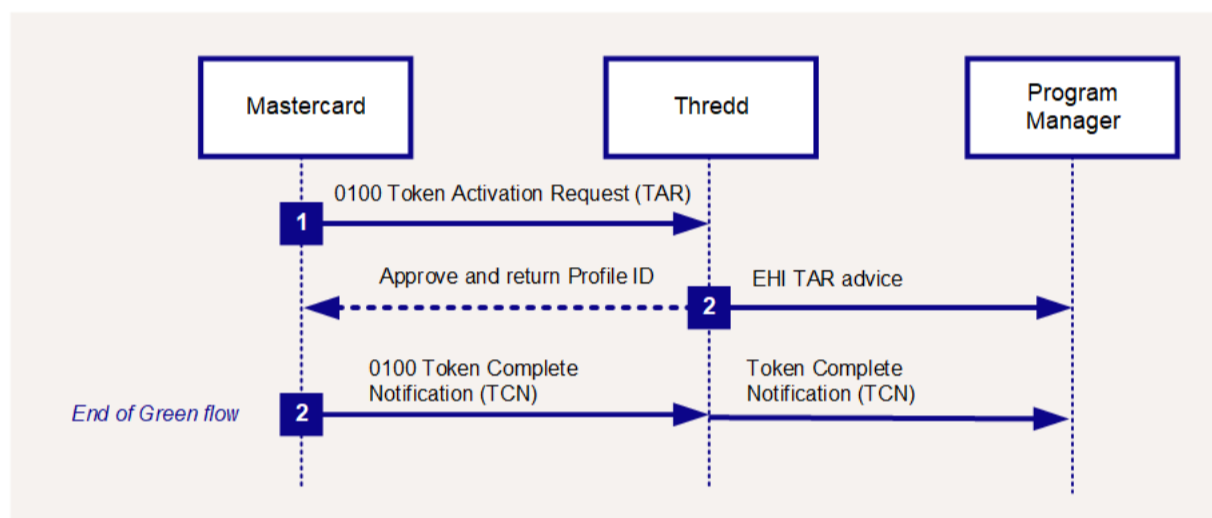


Figure 14: Mastercard Messages (Green Flow)

1. Mastercard sends an 0100 Token Activation Request (TAR).
2. Thredd returns an Approve response to Mastercard together with the Profile ID (if applicable) so that the token response displays the correct card art and T&Cs on the cardholder's mobile device screen.
Thredd forwards the TAR to the Program Manager, via EHI.
3. Mastercard sends an 0100 Token Complete Notification (TCN).
Thredd forward the TCN notification to the Program Manager, via EHI.

¹ ISO 8583 is the message format used for authorisation messages passed between Visa/Mastercard and Thredd.



8.2 Message flow for Mastercard Token Provisioning (Yellow Flow)

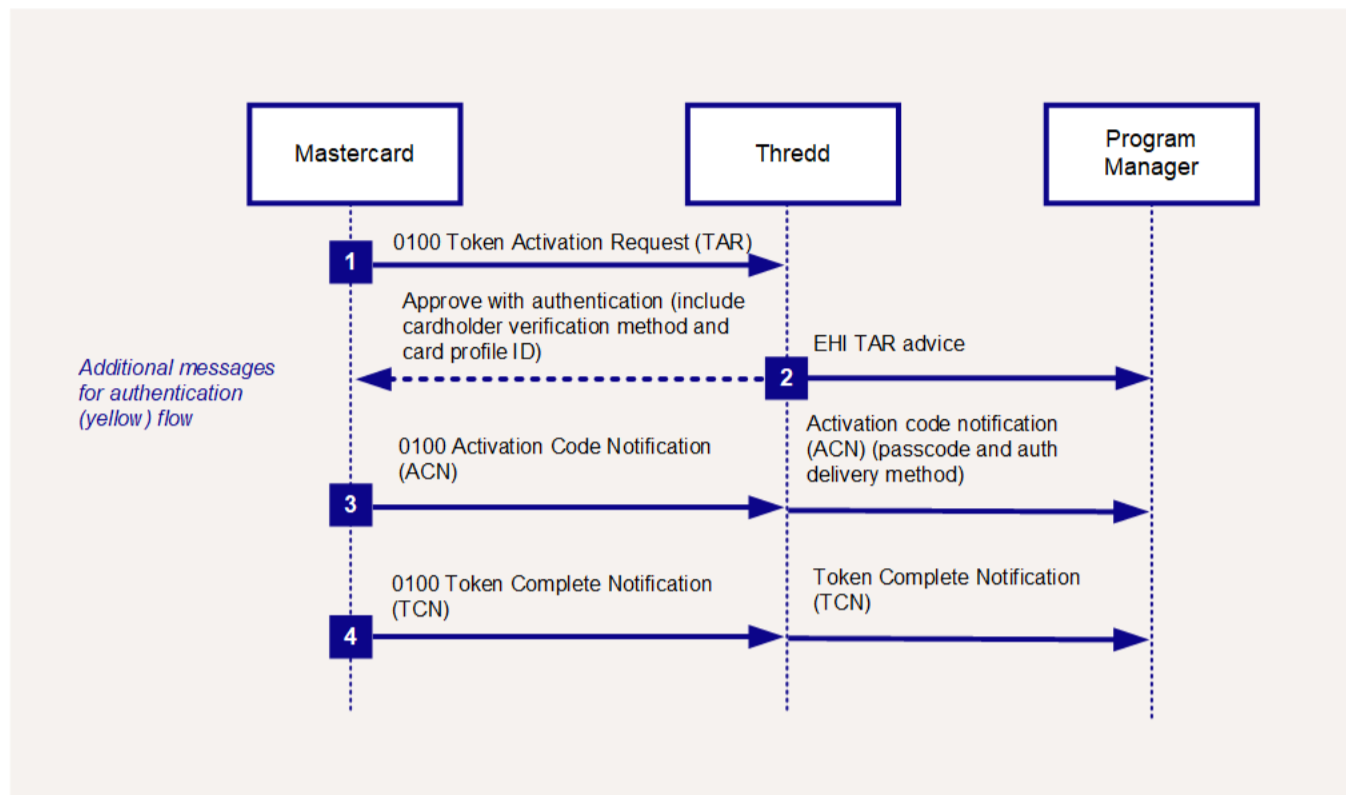


Figure 15: Mastercard Messages (Yellow Flow)

1. Mastercard sends an 0100 Token Activation Request (TAR).
2. Thredd returns an Approve with Authentication response to Mastercard. The response includes the available cardholder verification methods (e.g., SMS) and the Profile ID (if applicable).
Thredd forwards the TAR advice to the Program Manager, via EHI.
3. Mastercard sends an 0100 Activation Code Notification (ACN).
Thredd forward the ACN notification (plus the passcode and verification method) to the Program Manager, via EHI.
4. Mastercard sends an 0100 Token Complete Notification (TCN).
Thredd forward the TCN notification to the Program Manager, via EHI.

8.3 Message flow for Visa Token Provisioning (Green Flow)

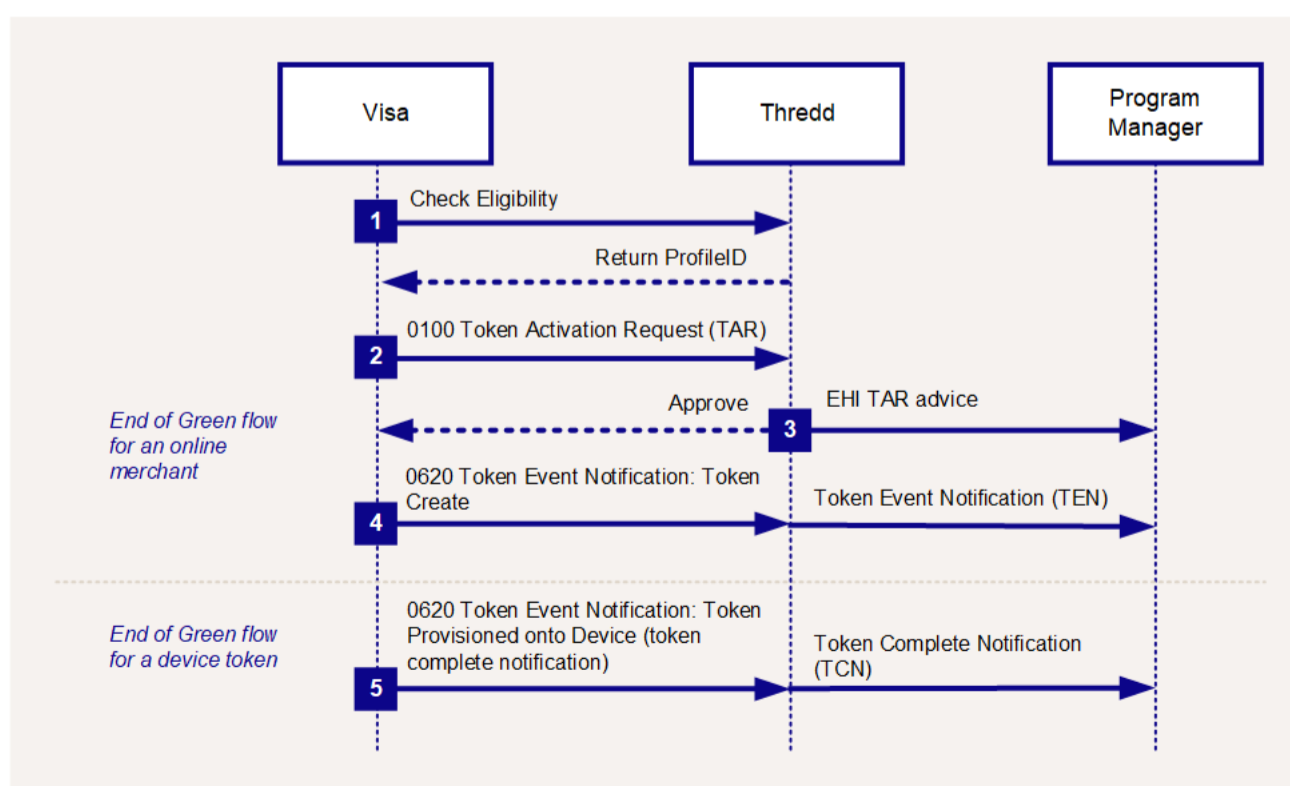


Figure 16: Visa Messages (Green Flow)



1. Visa sends a message to Thredd to check if the PAN is eligible for tokenisation.
Thredd returns the Profile ID (if applicable) so that the token response displays the correct card art and T&Cs on the cardholder's mobile device screen.
2. Visa sends an 0100 Token Activation Request (TAR) to Thredd.
3. Thredd returns an Approve response to Visa.
Thredd forwards the TAR to the Program Manager, via EHI.
4. Visa sends an 0620 Token Event Notification (TEN) to Thredd, to indicate the token has been created.
Thredd forwards the TEN notification to the Program Manager, via EHI.
5. For a token that is bound to a device, Visa sends an 0620 Token Event: Token Complete Notification (TCN), to indicate the token has been provisioned onto the device.
Thredd forwards the TCN notification to the Program Manager, via EHI.



8.4 Message flow for Visa Token Provisioning (Yellow Flow)

Note: This flow is only relevant to tokens that are bound to a mobile phone or other device.

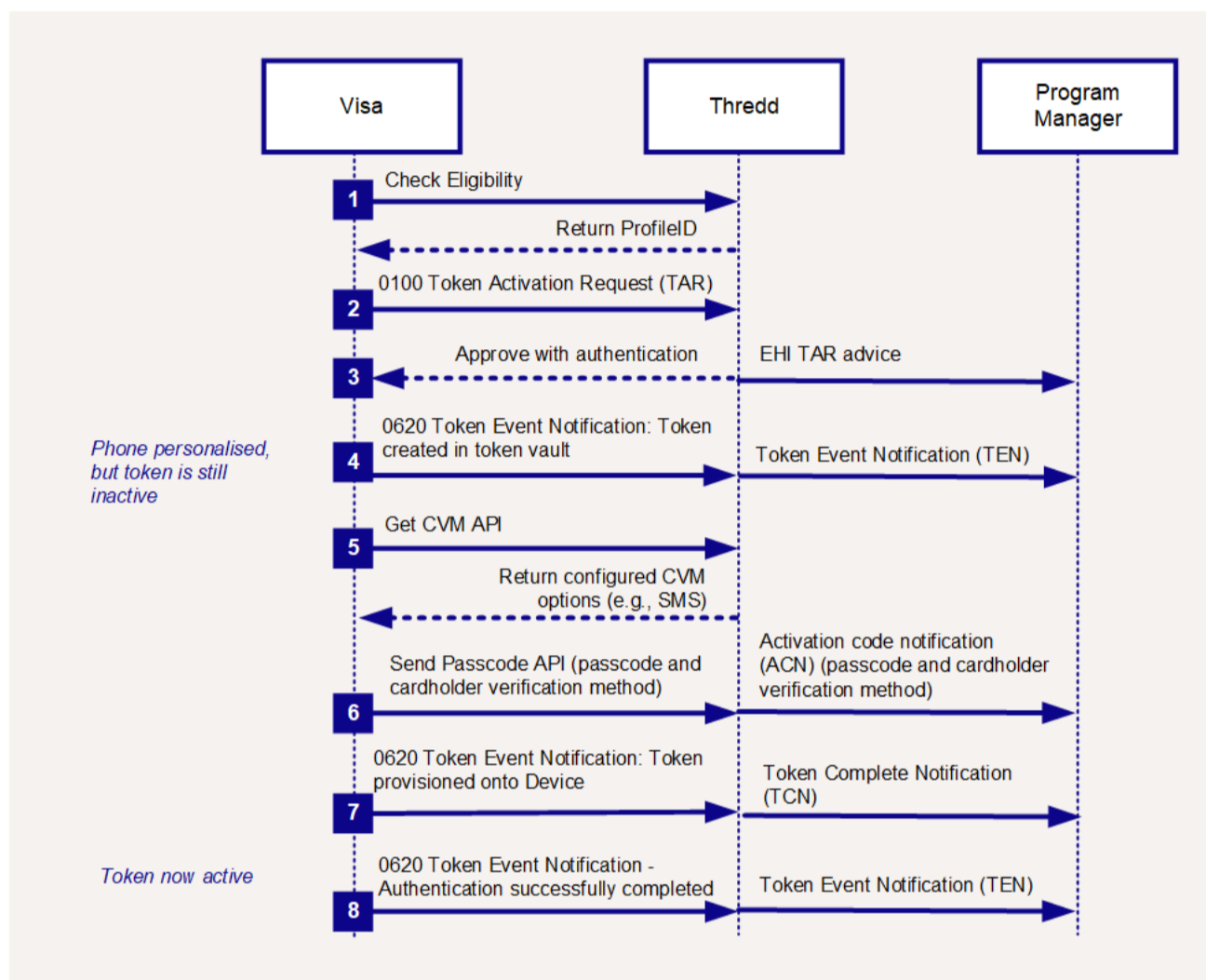


Figure 17: Visa Messages (Yellow Flow)

1. Visa sends a message to Thredd to check if the PAN is eligible for tokenisation. Thredd returns the Profile ID (if applicable) so that the token response displays the correct card art and T&Cs on the cardholder's mobile device screen.
2. Visa sends an 0100 Token Activation Request (TAR) to Thredd.
3. Thredd returns an Approve with Authentication response to Visa. Thredd forwards the TAR to the Program Manager, via EHI.
4. Visa sends an 0620 Token Event Notification (TEN) to indicate the token has been created (the token is not yet active)
5. Visa uses the Get CVM API to retrieve a list of available cardholder verification methods (CVMs) for this token from Thredd (i.e., methods such as SMS).
6. Visa uses their Send Passcode API to send the passcode and the user-selected cardholder verification method to Thredd. Thredd sends an Activation Code Notification (ACN) to the Program Manager, via EHI. The ACN contains the authentication passcode (One-time password) and user-selected verification method.
7. The OTP is delivered to the cardholder using their chosen verification method. Visa sends an 0620 Token Event: Token Complete Notification (TCN), to indicate the token has been provisioned onto the device. Thredd forward the TCN notification to the Program Manager, via EHI.
8. Visa sends an 0620 Token Event Notification (TEN) to Thredd, to indicate the token authentication was successful. The token is now active. Thredd forwards the TEN notification to the Program Manager, via EHI.

Note: Some Mobile Wallet token requestors require you to confirm to cardholders when the tokenisation process is complete or to follow up with cardholders when tokenisation has not been completed.

Note: The Token Complete Notification (TCN) sent over EHI currently indicates when the device is successfully provisioned. In some cases, later Token Event Notifications (TENs) can arrive once the cardholder is authenticated and Visa has activated the token, which represent the



actual end of the token provisioning flow.

The section [When to notify cardholders tokenisation is complete](#) below describes how you can identify the end of the tokenisation flow.

8.5 When to notify your cardholders that Tokenisation is complete

Mastercard sends Thredd a Token Completion Notification (TCN) which identifies when the tokenisation process is complete. We send you the Token Completion Notification ([ProcCode 350000](#)). You must notify your customers within 30 minutes of successful provisioning and activation of the token (an Apple requirement).

For Visa, currently the Token Completion Notification ([ProcCode 350000](#)) only represents the end of the tokenisation flow for Green Flow Device Tokenisation. Visa send a 620 message to indicate that the token is active. Thredd then send a Token Complete Notification (TCN) where the `paymenttoken_creatorStatus = A` (active). For details, see [Visa Tokenisation Messages](#).

8.6 Token Requestor Testing

Some Mobile Wallet Token Requestors require completion of testing before go-live. Please inform your Implementation Manager before this testing has started. Thredd do not receive updated test requirements from Mobile Wallet Token Requestors as we do not have a direct relationship with these parties.

Note: If you become aware of a recent change in the Apple or Android requirements, please contact your Account Manager or Implementation Manager before testing begins so Thredd can review.

8.7 Mastercard Provisioning Velocity Limits

Mastercard include provisioning velocity limits for MDES to enable the tackling of reprovisioning fraud in some public transport environments where authorisations are deferred. The provisioning velocity limits apply to a PAN and its associated device tokens.

How it Works

Mastercard blocks the provisioning if any of the following velocity limits are met:

- 4 provisions per day.
- 5 provisions for 7 days.
- 6 provisions for 30 days.

Velocity limit combinations are managed separately for each device associated with a PAN. For example, if the provisioning velocity limit is 6 provisions for 30 days, the monthly limit is applied to both a smart watch and a phone that is linked to the PAN.

Note: Thredd are not able to amend these limits. The limits are separate to the ones that Thredd manages internally.

Overriding the Limits

A customer service agent can override an individual provisioning velocity limit for each device and PAN combination for 30 minutes through a customer service portal. Alternatively, provisioning velocity limits can be overridden for a given PAN by adding to a whitelist in MDES Manager.

Opting in for Velocity Limit Provisioning

You can opt in for velocity limit provisioning by emailing digital.support@mastercard.com.

Note: Mastercard may also contact you if you have opted out of token reuse, and/or if fraud is detected on one or more devices.



9 Managing your Programme

This section describes how you can manage tokens on the Thredd platform and at the card scheme (network).

9.1 Token Lifecycle Management

Managing the status of merchant tokens and device tokens via Thredd's APIs on SOAP and REST. You can make changes to the tokenised PAN directly, or in some cases Thredd will automatically amend tokenised PANs when the linked PAN is changed.

Whenever a token is updated at Thredd we also inform the Token Service Provider so that they can update their systems, and the Token Requestor can accurately reflect the status of the token to cardholders. For details, see the section [Real-time Token Status Change](#).

Below is the list of supported use cases and the API services you can use to implement them.

9.1.1 API Use Cases

- Permanent block on payments from a lost or stolen device.
The cardholder has reported their device is permanently unavailable and so the payment token must be removed from the device. Used for lost, stolen or sold devices.
See [Update Device Tokens](#).
- Temporary block on payments from a device.
The device has been temporarily lost or the cardholder doesn't want to allow for use for payments (for example, phone in use by a family member or when travelling abroad).
See [Update Device Tokens](#).
- Update Merchant Tokens
Prevent a merchant from using their payment token (for example, for future or recurring transactions).
See [Update Merchant Tokens](#).
- Display payment tokens in-app.
Show a cardholder all their payment tokens, to manage them in the app.
See [View all Payment Tokens](#).
- Renew a card and transfer tokens.
Replace a physical or virtual card and ensure continuity of service for payment tokens. Used for expired, damaged or upgraded cards.
See [Renew a Card](#).
- Delete a card and linked tokens.
Close a cardholder's account or remove card functionality from their account.
See [Close a Card](#).
- Unbind a merchant token from a device.
Device binding allows a merchant token to be used with fewer authentications on a trusted device. Unbinding removes this if the device should no longer be trusted.
See [Unbind a merchant token from a device](#).
- Register for Click to Pay (Coming Soon)
Add a card into Click to Pay so that the cardholder can use their card at online merchants without typing in their card details.

9.1.2 Status Codes to use for Card Blocks

You should use the following Thredd status codes for temporary and permanent blocks:

- Temporary Block: "57 - Transaction not permitted to cardholder" or "62 - Restricted Card"
- Permanent Block: "46 - Account Closed" or "83 - Card Destroyed"

For details of additional card status codes, see the [Cards API Website: Card Status Codes](#).

Note: Changing the payment token to any of the permanent block statuses deletes it irreversibly. In this case, the cardholder will need to provision a new token on their device.



9.1.3 Update Device Tokens

Device tokens have statuses independent to the card and will need to be updated individually.

Use cases:

- Permanent block on payments from a lost or stolen device
- Temporary block on payments from a device
- Enabling a device only

Implementation:

Step	REST API	SOAP API
Identify the ID for the device token.	Get Card Payment Token	Payment Token Get
Update the status of the payment token.	Payment Token Status	Payment Token Status Change

9.1.4 Update Merchant Tokens

Merchant tokens (also called *Card on File* tokens) have the same status as the card, so when the card's status changes all merchant tokens linked to it have their status changed.

You can also change a merchant token status individually, however the next time the card status is changed the merchant token status will be aligned to the card.

Method 1: Update the card status using the public token (merchant tokens will change in sync):

Step	REST API	SOAP API
Update the card status.	Card Status	Update Card Status

Method 2: Update the merchant token status directly:

Step	REST API	SOAP API
Identify the ID for the device token.	Get Card Payment Token	Payment Token Get
Update the status of the payment token.	Payment Token Status	Payment Token Status Change

9.1.5 View all Payment Tokens

Display payment tokens linked to a Thredd public token.

Use cases:

- Display payment tokens in-app

Implementation:

Step	REST API	SOAP API
Return a list of all tokens linked to the specified public token.	Get Card Payment Token	Payment Token Get
Return a list of devices bound to a token. (For use on Visa (VDEP) service only.)	Get Card Payment Token Devices	Token Device Management

Tip: Payment tokens can also be viewed in Thredd Portal or Smart Client. Refer to the Thredd Portal Guide or Smart Client Guide.



9.1.6 Renew a Card (Move linked Tokens)

Renewing a card using these APIs will automatically move payment tokens from the old card to the new card. In the following circumstances the DPANs will NOT be transferred, and they will be deleted when the old card is placed in an irreversible status.

Use cases:

- Deleting a card and creating a new one
- Replacing an expiring card

Implementation:

Step	REST API	SOAP API
Renew or replace the card.	Card Renewal and Replacement	Card Renew
Activate the new card.	Update Card Status	Card Activate

Note: When a card is replaced or renewed Thredd send an API to the Token Service Provider in real-time informing them of the new PAN, CVV2 and Expiry which will then be passed to the Token Requestor.

9.1.7 Close a Card (Delete all linked Tokens)

Closing a card will automatically delete all payment tokens, this is the only way to delete all payment tokens at once, otherwise device tokens will need to be deleted individually.

Use cases:

- Delete a card and linked tokens

Implementation:

Step	REST API	SOAP API
Close the card record.	Card Status	Update Card Status

9.1.8 Unbind a Merchant Token from a Device

Merchants can bind a token to a device so that authentication is not required on future transactions (also referred to as a *Card on File* token). If a cardholder wants to opt out or their device is stolen, then you should unbind the device so that authentication is requested again.

Implementation:

Step	REST API	SOAP API
Identify the Device ID(s) linked to the payment token.	Get Card Payment Token Devices	Token Device Management
Unbind the device from the merchant token.	Unbind Payment Token	Token Device Management

Note: The unbind request triggers a real-time API call to the Token Service Provider. An approval action code (000) means the request has been successful on both the Thredd and Token Service Provider platforms. A failed response means that neither platform has been updated.

9.1.9 Response Codes (SOAP API)

When submitting a SOAP API request to change the status of a payment token, you may receive one of the following web service responses:



Response Code	Description										
000	The status change is successful on both Thredd and Token Service Provider platforms.										
213	Returned if you attempt to change to a status that is not compatible with the current status. For example, this error is returned in the following scenarios: <table border="1" data-bbox="478 632 1850 934"> <thead> <tr> <th>Current Status</th> <th>Changing to</th> </tr> </thead> <tbody> <tr> <td>Inactive</td> <td>Anything except Active</td> </tr> <tr> <td>Active</td> <td>Active</td> </tr> <tr> <td>Suspended</td> <td>Suspended</td> </tr> <tr> <td>Deleted</td> <td>Any status</td> </tr> </tbody> </table>	Current Status	Changing to	Inactive	Anything except Active	Active	Active	Suspended	Suspended	Deleted	Any status
Current Status	Changing to										
Inactive	Anything except Active										
Active	Active										
Suspended	Suspended										
Deleted	Any status										
654	Thredd received a status change request to block a payment token that is already blocked, deleted or deactivated.										
951	The status change is not successful at the card scheme and only the Thredd platform has been updated. If the first attempt at a status change is unsuccessful, Thredd recommends re-attempting the status change. Note: If subsequent attempts at a status change continue to be unsuccessful, contact Thredd for support.										
953	Occurs when Thredd were able to process the FPAN but not the DPAN, You should follow up with a Payment Token Status Change (Ws_Payment-Token_StatusChange) web service request (SOAP) or use the Update Payment Token Status endpoint (REST).										

9.1.10 Response Codes (REST API)

When submitting a REST API request to change the status of a payment token, you may receive one of the following responses:

Response Code	Description
200	The status change is successful on both Thredd and Token Service Provider platforms. For more information on the fields returned in the response, see the Cards API Website: Card Tokenisation .
401	Unauthorised access request
404	Resource not found

9.2 Adding and Updating your Card Art

When a token is created, the token requestor may wish to display it with the correct card art to help a cardholder identify their card.

During the tokenisation process Thredd responds to the card scheme with the card profile ID linked to the card. If no card profile ID is specified in the response from Thredd, the card scheme displays your default profile (card image artwork and terms & conditions) on the cardholder's mobile device screen. If you only have a single card product, you can use this default card profile, without requiring further updates on your end.

9.2.1 Adding Card Art Profiles

During the implementation process you can set up the default card art for your programme as described in the section [Token Configuration > Thredd Configuration Options > Dynamic vs. Static Card Art](#).

The card art that will be used is defined when the card is created, based on what is present in the [ProductRef](#) field (SOAP) or [designId](#) field (REST). This field is used for two purposes:



- When sending instructions to card manufacturers to identify the card image file in their systems to use
- To specify which card profile the card scheme (network) should use.

As a result, it is necessary to synchronise the IDs used at the card scheme and your card manufacturer.

With Mastercard you can define the card profile ID for tokenisation and Thredd recommends that you use the same card profile ID setup with your card manufacturer for image artwork.

Visa card profile IDs are generated by Visa in their Card Metadata Management (VCMM) tool. Visa returns a 32-character profile ID. This profile ID should be populated in the [ProductRef](#) field (SOAP) or [designId](#) field (REST) when creating cards and Thredd recommends that you use the same card profile ID when configuring image artwork with your card manufacturer.

Note: There may be situations where your card manufacturer is unable to update their identifier for image artwork to use the same identifier name for the Visa Profile ID. In these cases, please contact your Thredd implementation manager to discuss options.

9.2.2 Updating your Card Art

You can update the Card Art ID used for future tokenisation requests by updating the card record at Thredd. This can be done two ways:

- Raise a request to Thredd to populate the card profile IDs for the existing cards in your programme (Thredd uses a batch script to update the card profile ID linked to your card products).
- Use Thredd's API to update the profile ID for each card:
 - SOAP web services – use the *Update Cardholder Details V2* web service ([Ws_Update_Cardholder_Details_V2](#)). You must use the [ProductRef](#) field to add details of the card profile ID to be used. For details, see the [Web Services Guide > Update Cardholder Details V2](#).
 - REST-based Cards API – use the Update Cardholder Details endpoint. You must use the [designId](#) field to add details of the card profile ID to be used. For details, see the [Cards API Website > Update a Card](#).

For payment tokens that already exist, if you wish to change the card art you must use the card scheme's online portal: Visa Card Metadata Management (VCMM) for Visa or MDES Manager for Mastercard.

9.2.3 Terms and Conditions

During the tokenisation process a token requestor will display the issuer's terms and conditions for the cardholder to agree to. These are configured and updated at the card scheme's online portal: Visa Card Metadata Management (VCMM) for Visa or MDES Manager for Mastercard.



Appendix A: Device Scoring

Example of score configurations: 1-5, with 1 = least trusted, and 5 = most trustworthy.

Maximum Scores

Maximum scores which prompt whether we authenticate or decline.

Wallet Device Max Score Auth	3
Wallet Device Max Score Decline	1
Wallet Account Max Score Auth	3
Wallet Account Max Score Decline	1

If set to 0 = never authenticate or decline (use this if you do not want Thredd to use any of this logic).

Default Score

These options indicate what default score should be provided if no score is received from the Wallet Provider in the incoming TAR message. (Currently only Apple provide a device score)

Wallet Device Score Default	3
Wallet Account Score Default	3

In the above example, a value of 3 would result in authentication.



Appendix B: View the OTP on Smart Client

If a cardholder calls your call centre to retrieve the One Time Password, these are the steps your call centre staff need to follow.

1. Open the Smart Client Portal and select **View Transactions**.
2. Enter the cardholder's Thredd Token and search for the authorisation records.
3. Right-click the transaction and select **More Details > View Transaction Details** to display the **Transaction Details** screen.
4. Click the arrow to the right of the **Device** field. This displays the **Payment Token** screen.

Property	Value
Creator's token ref	
Linked Token	

Property	Value
Creator digi. ref	
Wallet Account Score	3
Wallet Device Score	3
Wallet risk table	
Thredd decision	Approve with Authentication
Thredd decision at	2023-07-05 09:50:59.963
Final decision	Approve with Authentication
Final decision by	Issuer Auth System (primary site)
Terms & Conditions	
PAN Source	Key Entry

Property	Value
Creator	Mastercard tokenization system (MDES)
Creator token ref	
Thredd token ref	8038
Token Expiry	2026-08-31
Token PAN	****9311
Token Type	Secure Element PAN
Wallet Provider	Apple
No. times replaced	0
Old Expiry Date	

Property	Value
Name	
ID	
IP address	
Device Language	
Location	32.160000 , 34.820000
Type	Mobile phone
End of phone number	8231
Firstname	
Lastname	
Wallet account hash	

Property	Value
Activation Code	977953
Activation Expires	2023-07-05 09:21:00.000 GMT/UTC
Activation Method	SMS to mobile
Activation Status	Unknown

Property	Value
Tokenised	<input checked="" type="checkbox"/>
Tokenisation Date	2023-07-05 09:51:27.677
Status(in Thredd)	00 - All Good
External Status	Deleted
Ext. Status set by	Cardholder
Ext. status changed	

Figure 18: Payment Token screen on Smart Client

5. The screen shows the payment token details supplied by MDES/VDEP, along with the decision process information. The One Time Password value is shown in the **Activation Code** field.
6. Once provided to the cardholder, they should be able to enter this into their Wallet app to authenticate.



Appendix C: EHI Tokenisation Fields

The table below lists the EHI message fields relevant to the tokenisation service.

Field	Description
PaymentToken_id	Unique Thredd token reference.
PaymentToken_creator	The token service provider (Mastercard or Visa).
PaymentToken_expdata	The expiry date of the token.
PaymentToken_type	The payment token type. Defines the technology the token is being held on.
PaymentToken_status	Indicates the status of the token. Please note, this can differ from the status of the PAN.
PaymentToken_creatorStatus	Indicates the status as set by the token service provider (Mastercard/Visa) and also on the device itself. This adds information around the progress of token setup along with whether a post-setup token is active or not. This field can contain “ ” or be empty when no value has been provided.
PaymentToken_wallet	The wallet provider (e.g., Apple Pay, Google Pay) the token is linked to.
PaymentToken_deviceType	The type of device the token is linked to (e.g., Mobile Phone, watch, tablet).
PaymentToken_lang	The language configured on the device linked to the token (if available).
PaymentToken_deviceTelNum	The telephone number of the device linked to the token.
PaymentToken_deviceIp	The IP address of the device linked to the token.
PaymentToken_deviceId	The device ID of the device linked to the token. (Also called the DE124 Payment Application Instance Id at Mastercard and SEIDs (Secure element ID) at Apple.)
PaymentToken_deviceName	The name of the device linked to the token.
PaymentToken_activationCode	The token activation code.
PaymentToken_activationExpiry	The token activation expiry date.
PaymentToken_activationMethod	The token activation method (e.g., 0=none; 1 = SMS)
PaymentToken_activationMethodData	The token activation method details (e.g., if the activation method is 1 for SMS, then provides the mobile phone number to send the SMS).

For more information, refer to the [EHI Guide](#).



```
<PaymentToken_deviceIp>192.0.0.8</PaymentToken_deviceIp>
<PaymentToken_deviceId>01234B234C1230011230054848300695D86E17C703548A4A</PaymentToken_deviceId>
<PaymentToken_deviceName>Test Apple Wa</PaymentToken_deviceName>
<PaymentToken_activationCode />
<PaymentToken_activationExpiry />
<PaymentToken_activationMethodData />
<PaymentToken_activationMethod>0</PaymentToken_activationMethod>
.....
</GetTransaction>
  </s:Body>
</s:Envelope>
```




```
"TXN_Time_DE07": "0605010001",
"Txn_Type": "A",
"Additional_Data_DE48": "063T260332733260101H06115019291999808020375130103001020291710418C ",
"Authorised_by_GPS": "Y",
"CU_Group": "PMT-CU-002",
"InstCode": "PMT",
"MTID": "0100",
"ProductID": 3004,
"SubBIN": 53759000,
"TLogIDOrg": 0,
"VL_Group": "PMT-VL-020",
"Dom_Fee_Fixed": 0.0000,
"Non_Dom_Fee_Fixed": 0.0000,
"Fx_Fee_Fixed": 0.0000,
"Other_Fee_Amt": 0.0000,
"Fx_Fee_Rate": 0.0000,
"Dom_Fee_Rate": 0.0000,
"Non_Dom_Fee_Rate": 0.0000,
"Additional_Data_DE124":
"195TAD00162036556451
                                1
                                C",
"SendingAttemptCount": 0
}
```

Response

Example response from the external host system:

```
{"Acknowledgement": "1"}
```




```
"PaymentToken_deviceType": "M",
"PaymentToken_lang": " ",
"PaymentToken_deviceTelNum": "7181",
"PaymentToken_deviceIp": "97E3830C",
"PaymentToken_deviceId": "0123456789E5C80020103080645305102E616B5CB01EB3DD6",
"PaymentToken_deviceName": "Stephanie's phone (3",
"Acquirer_id_DE32": "06015611",
"ActBal": 0.0000,
"Auth_Code_DE38": "166647",
"Avl_Bal": 0.0000,
"Bill_Amt": 0.0000,
"Bill_Ccy": "826",
"BlkAmt": 0.0000,
"FX_Pad": 0.0000,
"Fee_Fixed": 0.0000,
"Fee_Rate": 0.0000,
"MCC_Code": "5969",
"MCC_Desc": "Direct Marketing - Other",
"MCC_Pad": 0.0000,
"Merch_ID_DE42": "CARD ACCPT IDC",
"Merch_Name_DE43": "APPLE PAY St. Louis USA",
"POS_Data_DE22": "010",
"POS_Data_DE61": "10251090000084063368",
"Proc_Code": "350000",
"Resp_Code_DE39": "00",
"Settle_Amt": 0.0000,
"Settle_Ccy": "826",
"Status_Code": "00",
"Token": 374414990,
"Trans_link": "240604229542015611",
"Txn_Amt": 0.0000,
"Txn_CCY": "840",
"Txn_Ctry": "USA",
"Txn_Desc": "APPLE PAY St. Louis USA",
"Txn_GPS_Date": "2024-06-04 02:01:20.142",
"TXn_ID": 14414251623,
"Txn_Stat_Code": "A",
"TXN_Time_DE07": "0604010119",
"Txn_Type": "J",
"Additional_Data_DE48": "097T230221260310333540101C021653724083642754730304270706115011003027308020575130103001020291710418C ",
"Authorised_by_GPS": "N",
"CU_Group": "PMT-CU-001",
"InstCode": "PMT",
"MTID": "0620",
"ProductID": 4252,
"SubBIN": 51694000,
"TLogIDOrg": 0,
"VL_Group": "PMT-VL-001",
"Dom_Fee_Fixed": 0.0000,
"Non_Dom_Fee_Fixed": 0.0000,
"Fx_Fee_Fixed": 0.0000,
"Other_Fee_Amt": 0.0000,
"Fx_Fee_Rate": 0.0000,
"Dom_Fee_Rate": 0.0000,
"Non_Dom_Fee_Rate": 0.0000,
"Additional_Data_DE124": "192TCD001514838250601ZILCHCONS enStephanie's phone (31269d868a6a0394114a347cfe-f4ee0901e24060401010DAPLMC000026259835f4ba50cdd94aea8725b3beedde4545FAPLMC00002625988947bfff5eb134ffcae9822bf49031d09S",
"SendingAttemptCount": 0
}
```

Response

Example response from the external host system:

```
{"Responsestatus": "00", "CurBalance": 0.0, "AvlBalance": 0.0, "Acknowledgement": 1, "LoadAmount": 0.0, "Bill_Amt_Approved": 0.0, "Update_Balance": 0, "New_Balance_Sequence_ExtHost": 0, "CV2_Result": "", "AvlBalance_GPS_STIP": 0.0, "CurBalance_GPS_STIP": 0.0}
```



TEN (360000) Message

```
{
  "Network_Fraud_Data": "00199991",
  "FxProviderCardholderRate": 0.0,
  "Network_TxnAmt_To_BillAmt_Rate": "7878970:7",
  "POS_Date_DE13": "0000-00-00",
  "Traceid_Message": "BNET-20240603-BPDH0UE75",
  "Network_Currency_Conversion_Date": "2024-06-03",
  "Network_Transaction_ID": "BPDH0UE750603",
  "DCC_Indicator": 0,
  "multi_part_txn": 0,
  "multi_part_txn_final": 0,
  "auth_type": "0",
  "auth_expdate_utc": "2024-06-11 01:00:16.001",
  "Merch_Name": "Visa Provisioning Service",
  "Merch_City": "St. Louis",
  "Merch_Postcode": "63368",
  "Merch_Country": "USA",
  "GPS_POS_Capability": "00110000100000000000000000000000100000000230010",
  "GPS_POS_Data": "501800080000N000NNNNM0NUUXU",
  "Response_Source_Why": 0,
  "Message_Source": "CRDHLR",
  "Message_Why": 51,
  "traceid_lifecycle": "VIS1-20240604-304156486459325",
  "Balance_Sequence": 0,
  "Balance_Sequence_ExtHost": 0,
  "PaymentToken_id": 58098832,
  "PaymentToken_creator": "Visa-T",
  "PaymentToken_expdate": "2026-03-31",
  "PaymentToken_type": "SE",
  "PaymentToken_status": "00",
  "PaymentToken_creatorStatus": "D",
  "PaymentToken_wallet": "APPLE",
  "PaymentToken_deviceType": "M",
  "PaymentToken_lang": " ",
  "PaymentToken_deviceTelNum": "1504",
  "PaymentToken_deviceIp": "94FC8598",
  "PaymentToken_deviceId": "01234567896A8002119909991631144010A4926E8D6D1C36",
  "PaymentToken_deviceName": "iPhone",
  "Acquirer_id_DE32": "06015611",
  "ActBal": 0.0000,
  "Auth_Code_DE38": "161404",
  "Avl_Bal": 0.0000,
  "Bill_Amt": 0.0000,
  "Bill_Ccy": "826",
  "BlkAmt": 0.0000,
  "FX_Pad": 0.0000,
  "Fee_Fixed": 0.0000,
  "Fee_Rate": 0.0000,
  "MCC_Code": "5969",
  "MCC_Desc": "Direct Marketing - Other",
  "MCC_Pad": 0.0000,
  "Merch_ID_DE42": "CARD ACCPT IDC",
  "Merch_Name_DE43": "Visa Provisioning Service FR",
  "POS_Data_DE22": "010",
  "POS_Data_DE61": "10251090000084063368",
  "Proc_Code": "360000",
  "Resp_Code_DE39": "00",
  "Settle_Amt": 0.0000,
  "Settle_Ccy": "826",
  "Status_Code": "00",
  "Token": 155027420,
  "Trans_link": "240604919020015611",
  "Txn_Amt": 0.0000,
  "Txn_CCy": "840",
  "Txn_Ctry": "USA",
  "Txn_Desc": "Visa Provisioning Service FR",
  "Txn_GPS_Date": "2024-06-04 02:00:16.121",
}
```



```
"Txn_ID": 14414250789,
"Txn_Stat_Code": "A",
"TXN_Time_DE07": "0604010015",
"Txn_Type": "J",
"Additional_Data_DE48": "091T230221260310333480101C021653724083932605950304260306115011003027375130103001020291710418C ",
"Authorised_by_GPS": "N",
"CU_Group": "PMT-CU-001",
"InstCode": "PMT",
"MTID": "0620",
"ProductID": 4252,
"SubBIN": 51694000,
"TLogIDOrg": 0,
"VL_Group": "PMT-VL-001",
"Dom_Fee_Fixed": 0.0000,
"Non_Dom_Fee_Fixed": 0.0000,
"Fx_Fee_Fixed": 0.0000,
"Other_Fee_Amt": 0.0000,
"Fx_Fee_Rate": 0.0000,
"Dom_Fee_Rate": 0.0000,
"Non_Dom_Fee_Rate": 0.0000,
"Additional_Data_DE124": "068TVD00067408479624 2DAPLMC000026259862dc013463464968ac77488b5bd11e9d",
"SendingAttemptCount": 0
}
```

Response

Example response from the external host system:

```
{"Responsestatus": "00", "CurBalance": 0.0, "AvlBalance": 0.0, "Acknowledgement": 1, "LoadAmount": 0.0, "Bill_Amt_Approved": 0.0, "Update_Bal-
ance": 0, "New_Balance_Sequence_ExtHost": 0, "CVV2_Result": "", "AvlBalance_GPS_STIP": 0.0, "CurBalance_GPS_STIP": 0.0}
```



Appendix D: Visa Tokenisation Messages

The scenarios below describe how you can determine the end of the tokenisation flow on Visa.

Tip: When you get a message with payment token status of A, this means the token is active and ready to do transactions and should be the last message in the flow.

Scenario 1: Online Merchant Token Request - Green flow

Look for this information in the following EHI fields to identify the message flow and when you need to send a notification to your cardholder of successful provisioning.

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	BW or CF	330000 (TAR)	00	(omitted)	71	Approve
2	BW or CF	360000 (TEN)	00	A	71	Indicates end of Green flow. Do not notify cardholder ¹ .

Scenario 2: Mobile Wallet Token Requests with Green flow

For a mobile wallet: Look for this information in the following EHI fields to identify the message flow and what you need to do.

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	SE or CL	330000 (TAR)	00	(omitted)	71	Approve
2	SE or CL	360000 (TEN)	00	A	71	
3	SE or CL	350000 (TCN)	00	A	72	Last message in flow. Cardholder notification of successful provisioning can be sent here.

¹ These are used by e-commerce merchants who tokenise PANs for storage (e.g. Netflix) and the cardholder is not necessarily present so would be confused by a message confirming tokenisation and likely to consider it fraudulent.



Scenario 3: Mobile Wallet Token Requests Yellow flow with successful authentication

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	SE or CL	330000 (TAR)	85	(omitted)	71	Approve with authentication
2	SE or CL	360000 (TEN)	00	I	71	Token Event Notification
3	SE or CL	340000 (ACN)	00	I	0	Activation Code Network Message. Contains the OTP to verify the cardholder.
4	SE or CL	350000 (TCN)	00	I	72	In yellow flow - do not send messages using the TCN.
5	SE or CL	360000 (TEN)	00	A	73/74/75	Last message in flow. Cardholder notification of successful provisioning can be sent here.

Scenario 4: Mobile Wallet Token Requests with Yellow flow with unsuccessful authentication

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	SE or CL	330000 (TAR)	85	(omitted)	71	Approve with authentication
2	SE or CL	360000 (TEN)	00	I	71	Token Event Notification
3	SE or CL	340000 (ACN)	00	I	0	Activation Code Network Message. Contains the OTP to verify the cardholder.
4	SE or CL	350000 (TCN)	00	I	72	In yellow flow - do not send messages using



Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
						the TCN.
5	SE or CL	360000 (TEN)	00 or 06	I	53/54/55	Last message in flow. No notification of tokenisation completion as authentication was unsuccessful.



Appendix E: Mastercard Tokenisation Messages

The scenarios below describe how you can determine the end of the tokenisation flow on Mastercard.

Tip: When you get a message with payment token status of A, this means the token is active and ready to do transactions and should be the last message in the flow.

Scenario 1: Online Merchant Token Request - Green flow

Look for this information in the following EHI fields to identify the message flow and when you need to send a notification to your cardholder of successful provisioning.

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	BW or CF	350000 (TCN)	00	A	72	Indicates end of Green flow. Do not notify cardholder ¹ .

Scenario 2: Mobile Wallet Token Requests with Green flow

For a mobile wallet: Look for this information in the following EHI fields to identify the message flow and what you need to do.

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	SE or CL	330000 (TAR)	00	(omitted)	71	Approve
2	SE or CL	350000 (TCN)	00	A	72	Last message in flow. Cardholder notification of successful provisioning can be sent here.

Scenario 3: Mobile Wallet Token Requests Yellow flow with successful authentication

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	SE or CL	330000 (TAR)	85	(omitted)	71	Approve with authentication
2	SE or CL	340000 (ACN)	00	I	0	Activation Code Network Message. Contains the OTP to verify the cardholder.

¹These are used by e-commerce merchants who tokenise PANs for storage (e.g. Netflix) and the cardholder is not necessarily present so would be confused by a message confirming tokenisation and likely to consider it fraudulent.



Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
3	SE or CL	350000 (TCN)	00	A	72	Last message in flow. Cardholder notification of successful provisioning can be sent here.

Scenario 4: Mobile Wallet Token Requests with Yellow flow with unsuccessful authentication

Message Order	PaymentToken_Type	ProcCode	Resp_Code_DE39	PaymentToken_creatorStatus	Message_Why	Comments
1	SE or CL	330000 (TAR)	85	(omitted)	71	Approve with authentication
2	SE or CL	340000 (ACN)	00	I	0	Activation Code Network Message. Contains the OTP to verify the cardholder.
3	SE or CL	350000 (TCN)	00	A	72	Last message in flow. No notification of tokenisation completion as authentication was unsuccessful.



Appendix F: Apple Pay Tokens

Apple technology enables you to offer your customers an option to use their Apple devices to make contactless and secure e-commerce payments.

Your Mobile App can use an iOS API to view cards you've issued that have been provisioned onto the Apple device. You can use the Apple API to present your customer with their chosen card for payment. The cardholder can then provide consent for the transaction using Touch ID / Face ID or a passcode to make a payment. Upon completion of the payment, the cardholder is automatically returned back to your Mobile App.

Note: Apple require certification to use their service, which must be done directly with them. For further information, please refer to the Apple document: [Functional Requirements Direct NFC Access and Apple Pay](#).

For more information on Apple Pay services for developers, see the [Apple Pay Developer Website](#).

Apple Pay Token Provisioning

For details of token provisioning flows supported by Thredd, see [Token Provisioning Message Flows](#).

Token provisioning requests that are flagged as Apple Orange Flow and/or a Device Score = 1 indicate that more rigorous authentication of the cardholder is required. See [Apple Pay Orange Flow](#) below.

FPAN Reissue and Replacement

Apple Pay require that when the FPAN is reissued (due to expiration or lost/fraud replacement), the DPAN should continue to work without the need to re-provision the new physical card. The card in the Wallet displays the new FPAN number, if it has changed.

Note: By default, the FPAN and DPAN are connected, so if the FPAN is lost and the card status is changed this will also change the status of the DPAN. If you need the DPAN to continue to work while the FPAN is replaced you must split them, by enabling the `DPAN_over_FPAN` option. For details, see [DPAN over FPAN Status](#).

When replacing a card, you should always use the Card Renew ([Ws_Renew](#)) web service or the [Replace a Card](#) endpoint (REST). For details, see the [Web Services Guide](#) or the [Cards API Website](#).

Once the new card is activated, the update to the FPAN/DPAN details will be immediate.

FPAN Reissue and Replacement

Apple Pay require that when the FPAN is reissued (due to expiration or lost/fraud replacement), the DPAN should continue to work without the need to re-provision the new physical card. The card in the Wallet displays the new FPAN number, if it has changed.

Note: By default, the FPAN and DPAN are connected, so if the FPAN is lost and the card status is changed this will also change the status of the DPAN. If you need the DPAN to continue to work while the FPAN is replaced you must split them, by enabling the `DPAN_over_FPAN` option. For details, see [DPAN over FPAN Status](#).

When replacing a card, you should always use the Card Renew ([Ws_Renew](#)) web service or the [Replace a Card](#) endpoint (REST). For details, see the [Web Services Guide](#) or the [Cards API Website](#).

Once the new card is activated, the update to the FPAN/DPAN details will be immediate.



10 Appendix G: Realtime Token Status Change

This section provides more information on how the process of token status change works.

Real-time Token Status Change (Visa)

The following diagram shows the process of a token status change for Visa.

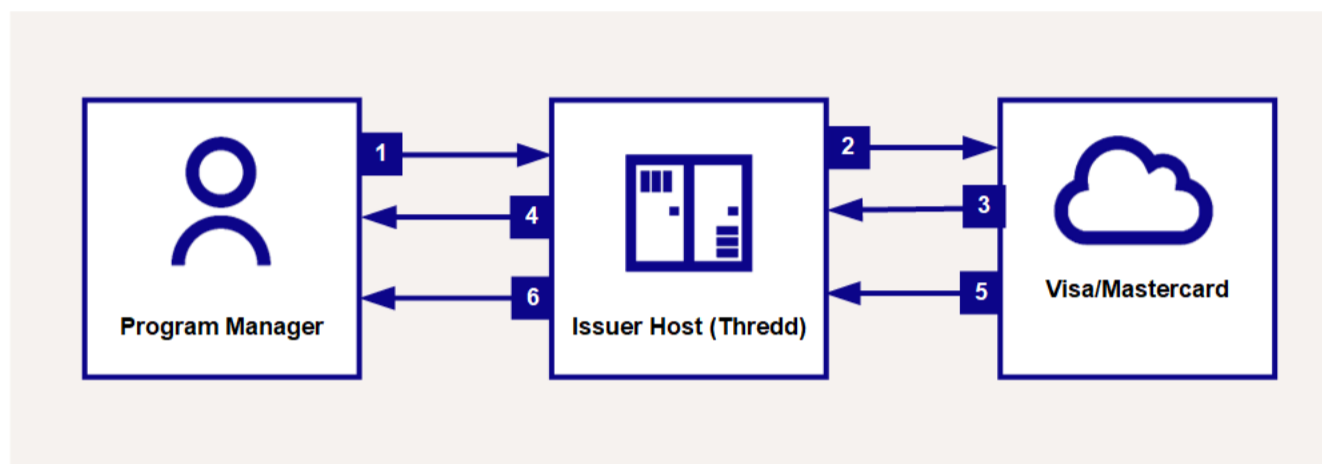


Figure 19: Real-time Token Status Change (Visa)

1. The Program Manager uses the Thredd web service to change the token status on the Thredd platform.
2. Thredd sends a request to Visa to update payment token status on their systems.
3. Visa responds with the token status update result.
4. Thredd confirms the status update in the web service response.
5. Visa sends a Token Event Notification (TEN) with the status change.
6. Thredd confirms the status update via EHI.

Real-time Token Status Change (Mastercard)

The following diagram shows the process of a token status change for Mastercard.

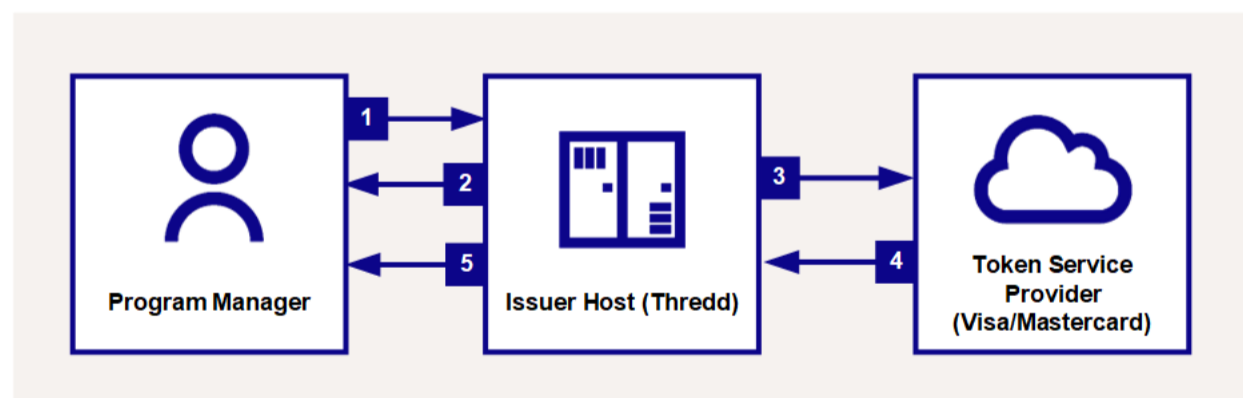


Figure 20: Token Status Change (Mastercard)

1. The Program Manager uses the Thredd web service to change the token status on the Thredd platform.
2. Thredd sends a request to Mastercard to update payment token status on their systems.
3. Mastercard responds with the token status update result.
4. Thredd confirms the status update in the web service response.
5. Mastercard sends a Token Event Notification (TEN) with the status change.
6. Thredd confirms the status update via EHI.



11 Appendix H: Card Status Codes

This section lists card status changes that trigger calls to the Token Service Provider (MDES/VDEP). For each card status change, the tables below list the corresponding DPAN status and the likely transaction status when the card is used.

Tip: For a full list of all card status codes and corresponding response codes, see the [Card Status and Response Codes Guide](#).

Web Services (SOAP) Status Codes

The following status codes can be set using the Thredd Web Services API:

Card status	Description	DPAN Status	Transaction status
00	All good/active	Active	Approve
04	Capture Card	Deactivated	Decline
05	Do not honour	Suspended	Decline
41	Lost card	Suspended	Decline
43	Stolen card (Capture)	Deactivated	Decline
46	Closed Account	Deactivated	Decline
57	Transaction not permitted to cardholder.	Suspended	Decline
59	Suspected fraud.	Suspended	Decline
62	Restricted card.	Suspended	Decline
63	Security violation.	Suspended	Decline
70	Cardholder to contact issuer.	Suspended	Decline
75	Allowable Number Of PIN Tries Exceeded	Suspended	Decline
83	Card Destroyed	Deactivated	Decline
98	Refund given to customer.	Suspended	Decline
99	Card Voided	Deactivated	Decline
G1	A short-term ¹ block ² which temporarily blocks card usage for all card transactions (excluding Credits and Refunds) for a short period.	Suspended	Decline
G2	Short-term full block (all transactions are blocked).	Suspended	Decline
G3	Long-term ³ block (excluding Credits and Refunds).	Suspended	Decline

¹ Use when you want merchants to try again. Visa guidelines instruct merchants to attempt up to 15 retries over 30 days. (If you expect the block to last longer than this, long-term may be more appropriate.)

² A card block will block all non-credit, Balance enquiry and tokenisation transactions. Refunds and Credits will be permitted.

³ Use when you don't want merchants to try again. Visa expect that the card should not return to the '00 Approve' state at all, or at least not within 30 days.



Card status	Description	DPAN Status	Transaction status
G4	Long-term full block (all transactions are blocked).	Suspended	Decline

For a full list of Card status codes , see the [Web Services Guide > Card Status Codes](#).

Cards API (REST) Status Codes

The following status codes can be set using the Thredd REST API:

Card status	Description	DPAN Status	Transaction status
00	All good/active	Active	Approve
04	Capture Card	Deactivated	Decline
05	Do not honour	Suspended	Decline
14	Invalid card	Suspended	Decline
41	Lost card	Suspended	Decline
43	Stolen card (Capture)	Deactivated	Decline
46	Closed account.	Deactivated	Decline
54	Card Expired	Deactivated	Decline
57	Transaction not permitted to cardholder.	Suspended	Decline
59	Suspected fraud.	Suspended	Decline
62	Restricted card.	Suspended	Decline
63	Security violation.	Suspended	Decline
70	Cardholder to contact issuer.	Suspended	Decline
75	Allowable Number Of PIN Tries Exceeded	Suspended	Decline
83	Card Destroyed	Deactivated	Decline
98	Refund given to customer.	Suspended	Decline
99	Card Voided	Deactivated	Decline
G1	A short-term ⁴ block ⁵ which temporarily blocks card usage for all card transactions (excluding Credits and Refunds) for a short period.	Suspended	Decline

⁴ Use when you want merchants to try again. Visa guidelines instruct merchants to attempt up to 15 retries over 30 days. (If you expect the block to last longer than this, long-term may be more appropriate.)

⁵ A card block will block all non-credit, Balance enquiry and tokenisation transactions. Refunds and Credits will be permitted.



Card status	Description	DPAN Status	Transaction status
G2	Short-term full block (all transactions are blocked).	Suspended	Decline
G3	Long-term ⁶ block (excluding Credits and Refunds).	Suspended	Decline
G4	Long-term full block (all transactions are blocked).	Suspended	Decline

⁶ Use when you don't want merchants to try again. Visa expect that the card should not return to the '00 Approve' state at all, or at least not within 30 days.



Frequently Asked Questions

Q. What is the role of Thredd in the tokenisation process?

Thredd are the issuing host and so approve or decline the tokenisation requests. Thredd plays an important role in connecting your program to the Token Service Providers (Mastercard/Visa), configuring the service and providing your systems with messages to support the tokenisation service. See [Who Participates in Tokenisation?](#)

Q. How do we start a project?

A project needs to be opened with the Token Service Providers (Visa/Mastercard) and with Thredd. Please discuss with your Account Manager.

Q. At what point does Thredd get involved?

Thredd needs to be involved when the Visa/Mastercard project is started, as we need to provide details in the documentation about the Thredd setup. See [Implementing a Tokenisation Project](#).

Q. What do we need to do as a Program Manager?

Essentially, you are the owner of the project and need to manage all parties involved in the setup of the service (Mobile wallet token requestors, token service providers and Thredd). See [Implementing a Tokenisation Project](#).

Q. How long does a project take?

To add tokenisation to an existing product typically takes approximately 3 months. This depends on many external factors and delays may occur in the live testing with Token Requestors.

Q. Why do we need EHI?

EHI is used to retrieve the One Time Passcode (OTP) used in authentication. This needs to be sent to the cardholder quickly and so cannot be sent via any reports. If you choose not to use EHI, you will only be able to use the Thredd SMS option to send the OTP to the cardholder. See [External Host Interface \(EHI\)](#).

Q. What is in-app provisioning?

In-app provisioning (also known as push provisioning) is a provisioning request originating from your mobile app so the cardholder does not have to enter their card details. This means that you can pre-authenticate the cardholder and choose to not authenticate when the provisioning request reaches Thredd. During In-App provisioning the cardholder will not enter their PAN and instead an encrypted payload must be sent to Apple to confirm the card details. When the In-App Provisioning API is called Thredd generate this payload using the cardholder details we have and the wallet inputs you provide, before returning it in the API response.

For more information, see [Implementing In App Provisioning](#).

Q. Are there any API calls we need to make?

Yes. See below:

Using Web Services (SOAP)	Using Cards API (REST)
<ul style="list-style-type: none"> • Ws_CreateCard – create card • Ws_Activate – activate card • Ws_Update_Cardholder_Details_V2 – update card • Ws_Payment_Token_Get – get the payment token • Ws_Token_Device_Management – manage the token device • Ws_Payment_Token_StatusChange – change the status of the payment token <p>For more information, refer to the Web Services Guide.</p>	<ul style="list-style-type: none"> • Create a Card – create card • Update Card Status – change card status to <i>active</i> • Update a Card – update card • Get Card Payment Token – get the payment token • Get Card Payment Token Devices – manage the token device • Update Payment Token Status – change the status of the payment token <p>For more information, see the Cards API Website.</p>

Q. Do we need to develop an app?

If you wish to support Mobile Wallet Token requestors, then an app is required. Please discuss with your chosen Token Requestors. You do not need an app for Online Merchant Token Requestors.



Q. On the PSF what does “override enabled/disabled” mean, what does it do?

This option on the Payment Setup Form (PSF) means that Thredd will override any logic that would send an authentication request to the cardholder when we detect that push provisioning has been carried out. Since the cardholder has already been authenticated during push provisioning, Thredd does not need to request further authentication.

This must be enabled to pass Apple testing and is a good cardholder journey for other token requestors. See [Thredd Configuration Options](#).

Q. What is the difference between VTS and VDEP?

They both refer to the same service. VTS is the Visa Token Service and VDEP is the Visa Digital Enablement Programme. You are required to sign a VDEP agreement with Visa when starting a new Visa Token Service integration.

Since VTS is also an abbreviation for the Visa Test Simulator (VTS), we use the term VDEP to avoid confusion.

Q. What’s the difference between a Token and a Payment Token?

Thredd refer to the 9-digit public token for use on Thredd systems as the Token or Public Token and the digitised tokens from the schemes is called a Payment Token.

Q. What’s the difference between a Token Requestor and a Wallet Provider?

These are used interchangeably between the schemes however Visa will more often use Token Requestor and Mastercard use Token requestor. Because the Mastercard Digital Enablement Service (MDES) was integrated first at Thredd you will often see references to token requestor.

Q. What is the difference between an FPAN and a DPAN?

These are Apple terms to specify which PAN is being discussed as following tokenisation there are two PANs for one card. The FPAN is the Funding PAN and refers to the original PAN on the card and the DPAN is the Device PAN and refers to the PAN personalised onto the device.

Q. Does Thredd know the DPAN?

Yes. Thredd receives and stores the DPAN during the provisioning process and validates it during subsequent transactions on that DPAN. IF Thredd does not receive the DPAN then it will decline transactions.



Glossary

This page provides a list of glossary terms used in this guide.

#

0100 Message

0100 Message Transaction Identifier (MTID). This is a Token Activation Request (TAR) message, requesting authorisation for the token creation.

0620 Message

0620 Message Transaction Identifier (MTID). This is a Token Event Notification (TEN) which indicates the token has been created.

3D Secure

3D Secure (3-domain structure), also known as a payer authentication, is a security protocol that helps to prevent fraud in online credit and debit card transactions. This security feature is supported by Visa and Mastercard and is branded as 'Verified by Visa' and 'Mastercard SecureCode' respectively.

A

ACN

Activation Code Network Message. The message sent to Thredd and also the Programme manager via EHI which contains the OTP to verify the cardholder.

Acquirer

Banking organisation and licensed scheme member that enables merchants to take card payments and send payment authorisation requests to the issuer using the card scheme's network.

Activation Code Notification

A message over EHI containing the OTP.

Authentication

This includes checks to confirm the cardholder identity, such as PIN, CVV2 and CAVV.

Authorisation

Stage where a merchant requests approval for a card payment by sending a request to the card issuer to check that the card is valid, and that the requested authorisation amount is available on the card. At this stage the funds are not deducted from the card.

Automated Fuel Dispenser (AFD)

Automatic fuel dispensers (AFDs) are used at petrol or gas stations for customer self-service fuel payments. Typically the customer inserts their card and enters a PIN number and the AFD authorises a fixed amount (e.g. £99). Once the final payment amount is known, the AFD may reverse the authorisation and/or request a second authorisation.

B

BIN

The Bank Identification Number (BIN) is the first four or six numbers on a payment card, which identifies the institution that issues the card.

BLOB

Binary Large Object file. A blob is a data type that can store binary data. It can be used to store images or other multimedia files.

C

Card Scheme (Network)

Card network, such as Mastercard, Visa or Discover, responsible for managing transactions over the network and for arbitration of any disputes.

Clearing File/Clearing Transaction

Thredd receive batch clearing files from the card networks, containing clearing transactions, such as presentments and network fees. The card issuer transfers the requested settlement amount to the acquirer and 'clears' the amount on the card, reducing the available



card balance accordingly.

COF Token

Card on File token request created by an online merchant.

COF Token Requestors

Online Merchant Token Requestors are referred to as Card on File (COF) Token Requestors. These are merchants who tokenise a payment card so that the token can be used for repeat payments or recurring payments on their website.

Counter

A counter under the PSD2 rules is used to track the number of transactions and cumulative amount before the cardholder is requested to authenticate using Strong Customer Authentication (SCA): for example, via PIN for a card or via 3D Secure authentication for an online transaction.

CVV2

The Card Verification Value (CVV) on a credit card or debit card is a 3 digit number on VISA, MasterCard and Discover branded credit and debit cards. Cardholder's are typically required to enter the CVV during any online or cardholder not present transactions. CVV numbers are also known as CSC numbers (Card Security Code), as well as CVV2 numbers, which are the same as CVV numbers, except that they have been generated by a 2nd generation process that makes them harder to guess.

D

Device Score

The score applied by the wallet provider defining the level of satisfaction the wallet provider has in the request being a genuine cardholder attempt, based on the wallet providers internal fraud parameters.

DPAN

Device PAN. The PAN value set up on the cardholder's device. This is not visible to the cardholder, but is the PAN used for the transactions as far as the merchant is concerned.

E

EHI

The External Host Interface (EHI) is a Thredd system that enables Thredd customers to receive and respond to real-time transaction data as well as financial messages.

EMV

EMV is a payment standard for smart payment cards, payment terminals and automated teller machines (ATMs). EMV is an acronym for "Europay, Mastercard, and Visa", the three companies which created the standard.

EMVCO

Organisation that facilitates worldwide interoperability and acceptance of secure payment transactions. Created by EuroPay, Mastercard and Visa.

External Host

The external system to which Thredd sends real-time transaction-related data. The URL to this system is configured within Thredd per programme or product. The Program Manager uses their external host system to hold details of the balance on the cards in their programme and perform transaction-related services, such as payment authorisation, transaction matching and reconciliation.

F

Fee Groups

Groups which control the card transaction authorisation fees, and other fees, such as recurring fees and Thredd web service API fees.

FPAN

Funding PAN. The true 16-digit PAN of the card, which Mastercard/Visa converts when authorisations come through to them from Acquirers on the DPAN.

G

Green Flow

This is an Apple term for a Token Provisioning request that is approved.



H

Hanging Filter

The period of time during which Thredd waits for an approved authorisation amount to be settled. This is defined at a Thredd product level. A typical default is 7 days for an auth and 10 days for a pre-auth.

I

Incremental Authorisation

A request for an additional amount on a prior authorisation. An incremental authorisation is used when the final amount for a transaction is greater than the amount of the original authorisation. For example, a hotel guest might register for one night, but then decide to extend the reservation for an additional night. In that case, an incremental authorisation might be performed in order to get approval for additional charges pertaining to the second night.

ISO 8583

The message format for BASE I/Authorisation messages between Thredd and the token service provider (Visa/Mastercard). This is the industry standard for authorisations.

Issuer (BIN sponsor)

Financial organisation and scheme member, licensed by the scheme to issue cards and process transactions using the scheme's network.

Issuer Host

This is the host connected directly to Visa/Mastercard for authorisation messages (i.e., Thredd).

M

MDES

The MasterCard Digital Enablement Service (MDES) is a data interchange platform for generating and managing secure digital payment tokens. It enables devices such as smartphones, smart watches, as well as merchants, to create a tokenised version of a Mastercard, which is specific to that device or merchant. Then the device/merchant can use the tokenised version of the card to perform transactions. The tokenised version of the card appears as just a normal Mastercard card number to the merchant and acquirer, and Mastercard will map the transactions onto the original cardholder Mastercard.

Merchant

The shop or store providing a product or service that the cardholder is purchasing. A merchant must have a merchant account, provided by their acquirer, in order to trade. Physical stores use a terminal or card reader to request authorisation for transactions. Online sites provide an online shopping basket and use a payment service provider to process their payments.

Merchant Category Code (MCC)

A unique identifier of the merchant, to identify the type of account provided to them by their acquirer.

MIP

Mastercard Interface Processor (MIP) The processing hardware and software system that interfaces with Mastercard's Global Payment System communications network.

Mobile Wallet Token Requestor

A token requestor connected to a mobile device.

MRCHTOKEN

Thredd name for the Wallet Provider group, representing Online Merchant Token Requestors. Also referred to as M4M (by Mastercard) and Card on File (by Visa).

N

NFC

Near Field Communication (NFC) is a technology that enables a device, such as a mobile phone or payment ring, to transmit data to a Point of Sale (POS) terminal, enabling contactless payments.



O

Offline Transaction

This is often used in scenarios where the merchant terminal is not required to request authorisation from the card issuer (for example for certain low risk, small value transactions used by airlines and transport networks). The card CHIP EMV determines if the offline transaction is permitted; if not supported, the terminal declines the transaction. Note: Since the balance on the card balance is not authorised in real-time, there is a risk that the card may not have the amount required to cover the transaction.

Online Merchant Token Requestor

A token requestor that is an e-commerce merchant.

Orange Flow

A token request for Apple Pay where Apple indicates if the request is high risk. As a result it must be authenticated through either a mobile app authenticator or a call centre with fraud checking.

OTP

One Time passcode/ Activation code which is sent to the cardholder for use in authenticating during token provisioning, during the setup of Google Pay, Apple Pay or other wallet on their device.

P

PAN

The card's 16-digit primary account number (PAN) that is typically embossed on a physical card.

Partial Amount Approval

Some acquirers support a partial amount approval for Debit or Prepaid payment authorisation requests. The issuer can respond with an approval amount less than the requested amount. The cardholder then needs to pay the remainder using another form of tender.

Payment Token

Thredd term for a MDES/VDEP token. This is used to differentiate between a Thredd public token and a MDES/VDEP token. Thredd use this in EHI and web service calls to identify a particular DPAN

Payment Token Usage Wallet

The default set of parameters Thredd will use to authorise a TAR.

PCI Compliance

The Payment Card Industry Data Security Standard (PCI DSS) is an information security standard for organisations that handle credit cards from the major card schemes. All Program Managers who handle customer card data must be compliant with this standard. See: https://www.pcisecuritystandards.org/pci_security/

Personalisation

The technical process of marking private data specific to a given card or device. The same terminology is used when putting private data on a chip card or a smart device.

Point of Sale (POS) Terminal

A hardware device for processing card payments at retail stores. The device has embedded software that is used to read the card's magnetic strip data.

Program Manager

A Thredd customer who manages a card program. The program manager can create branded cards, load funds and provide other card or banking services to their end customers.

PSD2

Payment Service Directive 2. PSD2 is an EU Directive which sets requirements for firms that provide payment services. It aims to improve consumer protection, make payments safer and more secure, and drive down the costs of payment services.

Public Token

The Thredd 9-digit token is a unique reference for the PAN. This is used between Thredd and clients to remove the need for Thredd clients to hold actual PANs.

Push Provisioning

The process of pre-authenticating the cardholder prior to a token request being sent to Visa.



R

Red Flow

This is an Apple term for a Token Provisioning request that is declined.

S

sFTP

Secure File Transfer Protocol. File Transfer Protocol (FTP) is a popular unencrypted method of transferring files between two remote systems. SFTP (SSH File Transfer Protocol, or Secure File Transfer Protocol) is a separate protocol packaged with SSH that works in a similar way but over a secure connection.

Smart Client

Smart Client is Thredd's legacy desktop application for managing your account on the Thredd Platform.

Strong Customer Authentication (SCA)

Authentication which is a combination of two factors of identification at checkout. Examples include something they know (such as a password or PIN), something they get (such as an OTP in a mobile phone or other device) or something they are (such as their fingerprint).

T

TAV

Tokenisation Authentication Value. Used as part of In-app provisioning process and is the encrypted message that contains the PAN details for Mastercard from the Programme Manager.

Thredd Portal

Thredd Portal is Thredd's new web application for managing your cards and transactions on the Thredd Platform.

TLS

Transport Layer Security (TLS) is a security protocol that provides privacy and data integrity for Internet communications. Implementing TLS is a standard practice for building secure web apps.

Token Activation Request (TAR)

Tokenisation Authorisation Request messages enable the issuer to provide a real-time decision as to whether the token service provider (MDES/VDEP) can digitise a card and designate a token on their behalf.

Token Complete Notification (TCN)

Tokenisation Complete Notification. Sent from Mastercard/Visa to Thredd and made available via EHI to the Programme Manager to confirm the setup of the token was successful (note: there may be further messages for activation).

Token Event Notification (TEN)

Tokenisation Event Notification. Informs the issuer of unsuccessful Activation Code entry attempts and subsequent invalidation of an Activation Code or when a token is suspended, resumed or de-activated.

Token Requestor

The token requestor initiates the request to convert your cardholder's Permanent Account Number (PAN) into a digital token. Token requestors can be mobile wallets (such as ApplePay) or online merchants (such as Netflix). Mastercard refer to the Token Requestor as the "Wallet Provider".

Token Service Provider (TSP)

The entity who stores the mapping between the PAN and the token. With the existing Thredd integration this would be Visa or Mastercard.

Triple DES

Triple DES (3DES or TDES), is a symmetric-key block cipher, which applies the DES cipher algorithm three times to each data block to produce a more secure encryption.

V

Validation

Checks to confirm the card is valid, such as CHIP cryptograms, mag-stripe data (if available) and expiry date



VDEP

Visa Digital Enablement Programme. Also called the Visa Tokenisation Service (VTS).

Visa Card Metadata Management (VCMM)

Online tool provided by Visa to enable card issuers to add artwork and terms & conditions for use on tokenised card images. For more information, see: <https://developer.visa.com/capabilities/token-service-provisioning>

VTS

Visa Tokenisation Service - is the Visa product name for tokenisation and equivalent of Mastercard's MDES (see MDES). Thredd refer to this service as the Visa Digital Enablement Program (VDEP).



Document History

Version	Date	Description	Revised by
2.2	27/11/2025	Added support for Mastercard Click to Pay. See Click to Pay .	JB
	20/11/2025	Included a description on Mastercard Provisioning Velocity Limits .	KD
	21/08/2025	Added support for Samsung Pay to In-App Provisioning. See Implementing In App Provisioning .	JB
	31/07/2025	Updated Event Notifications to include information on how to use webhooks for event notifications.	JB
	12/06/2025	Updated the Tokenisation Configuration page, including updating guidance on Payment Token groups, removing the Visa/Mastercard Rules section, updating references on the Mastercard Portal, and adding new device binding logic.	JB
	02/05/2025	New Click to Pay section added.	JB
2.1	25/02/2025	Updated the example request body for Apple In-App Provisioning. See Implementing In App Provisioning .	JB
	12/02/2025	Added references to Thredd Portal, our new web application for managing your cards and transactions.	JB
	30/01/2024	Added Orange Flow section to How Tokenisation Works page. Added Orange Flow to Glossary .	JB
	27/01/2025	New Appendix H: Card Status Codes – lists card status changes that trigger calls to Token Service Provider (MDES/VDEP).	WS
	16/01/2025	Corrections to JSON tokenisation examples: ACN, TCN and TEN messages should have <code>Txn_Type = J</code> . ACN messages should have an <code>MTID = 0120</code> , while TCN and TEN messages have an <code>MTID = 0620</code> . See Appendix C: EHI Tokenisation Fields .	WS
	16/10/2024	Rewrite to section Managing your Program , introducing new use cases and details of how to implement using either SOAP API or REST API. New Appendix G: Realtime Token Status Change . Revision to FAQ on using Thredd's In-app provisioning service.	WS
	21/08/2024	Added a new section with important information on Configuring Token Sub-Bin Ranges .	WS
	19/07/2024	Added section on implementing In-App provisioning . Updated Managing your Programme and How Tokenisation Works to support these changes.	JB
	02/07/2024	Updated Managing your Programme to update Mastercard update file frequency. Updated the company address.	JB
2.0	27/06/2024	Correction - updated JSON format 36000 tokenisation message from Mastercard to Visa. See Appendix C: EHI Tokenisation Fields .	JB
	10/06/2024	Added examples of EHI 34000, 35000 and 36000 tokenisation messages in JSON format. See Appendix C: EHI Tokenisation Fields .	WS
	04/04/2024	Updates to content to align with taxonomy updates on our Documentation Portal. Added details of using Thredd's REST-based Cards API to manage tokenised cards.	WS



Version	Date	Description	Revised by
	05/10/2023	Correction to text in Scenario 1 in Appendix E: Mastercard Tokenisation Messages . The EHI ProcCode received in Green Flow should be "350000 (TCN)".	WS
	29/09/2023	Updated Smart Client screen shot in Appendix B: View the OTP on Smart Client .	MW
	21/09/2023	Added details to clarify that the EHI field PaymentToken_creatorStatus can contain "" or be empty when no value has been provided. See Appendix C: EHI Tokenisation Fields .	WS
	01/09/2023	New response codes table and details added to the section Changing the status of a Payment Token .	WS
1.9	14/08/2023	Correction to the list of Visa documents that must be completed when implementing a tokenisation project through Visa. See Implementing a Tokenisation Project .	WS
	08/08/2023	Added a note to clarify that the Token Device Management web service (Ws-Token-Device-Management) is for use on the Visa (VDEP) service only. See Managing your Programme .	WS
	04/08/2023	Added JSON example EHI TAR Message. See Appendix C: EHI Tokenisation Fields .	JB
	07/06/2023	Updated Operations email address to be occ@thredd.com	MW
	27/04/2023	Guide rebrand to new company name and brand identity.	WS
1.8	13/02/2023	Revised wording around Apple Pay Orange flow support. See Appendix F: Apple Pay Tokens .	WS
	21/12/2022	Updated numbering in Table of Contents.	MW
1.8	22/12/2022	Added details of the real-time API for Mastercard clients. See Real-time Token Status Change (Mastercard) and FPAN Reissue and Replacement	MW
	01/12/2022	Updated the Copyright Statement.	
1.7	07/11/2022	Correction: In the Visa Tokenisation yellow flow, the final message PaymentToken_creatorStatus status should be "/" when provisioning was unsuccessful.	WS
	12/10/2022	Added a new section on Updating the Card Profile Linked to a Token . New Payment-Token Transactions setting enables you to decide whether the Thredd card status should be checked or ignored during transaction authorisations. See Thredd Configuration Options .	WS
1.6	03/08/2022	Updated the PDF page layout.	MW
	21/07/2022	New online version of the Tokenisation Service Guide is now available.	PC
	28/4/2022	Added details of Mastercard Real-Time Token to Token and PAN Lifecycle Management. See Real-time Token Status Change (Mastercard) .	MW
1.5	14/04/2022	Addition of Digiseq as a third-party push provisioning service provider .	WS
	28/04/2022	Correction: If the DPAN over FPAN setting is enabled , then you must separately set the statuses of both the FPAN and DPAN. See DPAN over FPAN Status .	
1.4	08/02/2022	Added information on the DPAN over FPAN Status option.	WS
	02/02/2022	Added note to clarify usage of cards status codes for temporary blocks. See Status Codes to use for Card Blocks .	
	01/04/2022		



Version	Date	Description	Revised by
		Added details of MRCHTOKEN and Apple Pay Orange flow. Added new Appendix on ApplePay Tokens . Added details of enabling Manual Key Entry in the Card Usage Groups used by tokenised cards.	
1.3	9/08/2021 25/08/2021 03/09/2021	New advice on Status Codes to use for Card Blocks . Updates to diagram and description in Section Message flow for Visa Token Provisioning (Yellow Flow) . Note added to section Token Provisioning Message Flows to highlight that tokenisation messages receive via EHI are advices only and should never be used for payment authorisation approval or decline decisions.	WS
1.2	28/06/2021	New Mastercard Tokenisation Messages .	WS
1.1	14/05/2021	Changes to section Thredd Configuration Options , and updates to Figures 9 and 10.	WS
1.0	29/03/2021	First version	WS
0.1	4/11/2020	Initial draft	SB



Contact Us

Please contact us if you have queries relating to this document. Our contact details are provided below.

Thredd UK Ltd.

Company registration number 09926803

Support Email: occ@thredd.com

Telephone: +44 (0) 203 740 9682

Our Head Office

33 Kingsway

London

WC2B 6UF

Technical Publications

If you want to contact our technical publications team directly, for queries or feedback related to this guide, you can email us at: docs@thredd.com.