# New Payments Platform

# API Framework

**Publication Version 5.0**

07 May 2021

NPP Australia Limited and SWIFT SCRL

# Contents

**Table of Figures**

## TERMS OF USE

The NPP API Framework is provided ‘AS IS’ without warranties of any kind, and NPP Australia Limited (NPPA) neither assumes nor accepts any liability for any errors or omissions contained in this document. **NPPA disclaims all representations and warranties, express or implied, including without limitation, implied warranties of merchantability, fitness for a particular purpose, title and non-infringement, as to the specifications set out in this document.**

NPPA makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to the NPP API Framework. NPPA has no responsibility to determine whether any implementation of the NPP API Framework may violate, infringe, otherwise exercise the patent, copyright, trademark, trade secret, know-how other intellectual property rights of any third party, and thus any person who implements any part of the NPP API Framework should make their own inquiries and obtain independent advice before implementing any part of this Framework.

Without limiting the foregoing, the NPP API Framework may provide for the use of technology which may be the subject matter of patents in several countries. Any person seeking to implement a solution based on the NPP API Framework is solely responsible for determining whether its activities require a licence to such technology. NPPA shall not be liable for any person’s infringement of intellectual property rights in connection with the NPP API Framework.

Third parties wishing to use NPPA intellectual property including the PayID trademark, should visit **https://payid.com.au/payid-for-business/**.

To learn more about PayTo visit **https://payto.com.au/**

PayID & PayTo are registered trademarks of NPP Australia Limited.

**Important Note:**
Designed to promote inter-operability and standardisation, the NPP API Framework defines the key technical approach and mandatory data attributes for NPP APIs, aligned to ISO 20022 standards. The NPP API Framework includes sample APIs which are included for the purposes of illustrating how the framework could be used by participating financial institutions. For more information regarding what APIs are available for third party use, please contact your financial institution.

***Please note that NPPA does not host an NPP API service or offer NPP APIs for third party use on the Platform. Participating financial institutions may make their proprietary NPP APIs available for use by third parties.***

# 1 Introduction

The New Payments Platform (NPP) is fast, flexible and data-rich payments infrastructure that enables Australian consumers, businesses and government agencies to make real-time payments between accounts at participating Australian financial institutions*.* NPP Australia Limited (NPPA) has engaged SWIFT to design, build and operate the NPP platform infrastructure.

This document has been jointly developed by NPPA and SWIFT to promote development of the NPP ecosystem and to assist NPP Participants, Third Party Service Providers and software developers with the development of API solutions for NPP transactions.

In order to maximise standardisation and interoperability, and to provide a consistent NPP experience, NPP Participants, Third Party Service Providers and software developers are encouraged to use the NPP API Framework as a guide for the development of their own API solutions for NPP. This document provides a consolidated point of reference and an expanded view for NPP Participants, Third Party Service Providers and software developers to further their adoption of API technologies for NPP.

NPPA does not mandate use of this Framework by participating financial institutions. NPP Participants that utilise this Framework for the development of open APIs or partner-based APIs are free to use this Framework, and to publicise such use, as they see fit.

**NPPA itself does not host an NPP API service or offer NPP APIs for third party use on the platform. Participating financial institutions may make their proprietary NPP APIs available for use by third parties**.

## 1.1 Purpose & Scope

The purpose of the NPP API Framework document is to provide guidance relevant to the design of APIs in the context of NPP. The Framework is designed to support and facilitate NPP Participants' open NPP APIs and partner-based NPP APIs, and to enable Third Party Service Providers and software developers to design payment services using the NPP. The NPP API Framework is intended to encourage innovation by establishing a set of minimum standards primarily for the benefit of Third Party Service Providers and software developers, which may obviate the need to build multiple customised APIs for interacting with each NPP Participant.

NPP Participants, Third Party Service Providers and software developers may identify different or additional considerations for their API design at a more detailed level which are not covered by this document.

Overlay Service Providers (**OSPs**) are encouraged to make available specific data usage or mandatory elements for use with the API framework. These elements are defined and maintained by each OSP and are the responsibility of that organisation.

The specific purpose of this document is to:

- Identify a common set of design principles and best practices that could be leveraged by the NPP community to reduce interoperability variation across core data exchanges when implementing API based technologies;
- At a high level, outline considerations to ensure the NPP community has a common reference of key API design considerations and how they may be approached by the industry; and
- Identify applicable standards for NPP data transfer, data constructs and security

## 1.2 Sample APIs

To provide Third Party Service Providers and software developers with an understanding of how to utilise the NPP API Framework, sample NPP APIs have been included as part of the framework documentation. The sample NPP APIs include usage guidelines in .pdf and excel format as well as a JSON example. The samples include the following:

- Look up of a PayID which will validate that the PayID exists and if valid return the associated short name and status - API: pain.a11.001.04 (Get Account Servicer by Alias);

  Please note that this API will be amended in the next version of the NPP API Framework as the BIC of the Payee Participant will no longer be returned in the API response.
- Submit a payment initiation request to process a payment - API: pain.a09.001.05 (Submit Payment);
- Confirm completion of payment - API: pain.a10.001.03 (Get Payment Status);
- Submit a request to return a previously submitted payment – API: camt.a09.001.02 (Cancel Payment);
- Confirm the status of the previously submitted Cancel Payment – API: pain.a48.001.01 (Get Cancellation Status)
- Submit a request to return a previously received payment – API: camt.a10.001.02 (Return Payment);
- Fetch payment details of a debit or credit from/to their account – API: camt.a11.001.04 (Payment Event Details); and
- Send notification of a payment event, debit or credit, from/to an account – API: camt.a12.001.03 (Event Notification).
- Initiate one or more interbank payments as individual debits or as a single batch debit – API: pain.a46.001.03 (Batch Payment Initiation)
- Confirm completion of batch payment – API: pain.a47.001.01 (Batch Payment Status)
- Fetch details of a batch of debits from their account – API: camt.a13.001.03 (Batch Payment Event Details)

**PayTo APIs:**

The following sample APIs can be used to perform certain functions related to the NPP PayTo service. The term ‘mandate’ refers to a PayTo agreement that is authorised by the Payer customer.

- Submit a request to create a Mandate – API: pain.a12.001.01 (Create Mandate)
- Get details of a specific mandate – API: pain.a13.001.01 (Get Mandate By ID)
- Submit a request to amend an existing mandate – API: pain.a14.001.01 (Amend Mandate)
- Confirm the validity of a mandate – API: pain.a15.001.01 (Validate Mandate)
- Retrieve the current status of a mandate – API: pain.a16.001.01 (Get Mandate Status)
- Submit a request to change the status of a mandate – API: pain.a17.001.01 (Change Mandate Status)
- Fetch details of a specific action performed on a mandate – API: pain.a22.001.01 (Get Mandate Action)

- Provide a resolution to a pending mandate action – API: pain.a23.001.01 (Resolve Mandate Action)
- Query on the MMS to retrieve a list of mandates that match a selection criteria – API: pain.a36.001.01 (Mandate Enquiry)

| API | Method | API Name | Description |
|---|---|---|---|
| pain.a09.001.05 | POST | /payment/submit | Submit Payment |
| pain.a10.001.03 | GET | /payment/{transaction_identification}/status | Get Payment Status |
| pain.a11.001.04 | GET | /accountServicer/alias | Get Account Servicer by Alias |
| camt.a09.001.02 | POST | /cancelPayment/submit | Cancel Payment |
| camt.a10.001.02 | POST | /returnPayment/submit | Return Payment |
| camt.a11.001.04 | GET | /paymentEventDetails/{notification_identification}/event | Payment Event Details |
| camt.a12.001.02 | POST | /paymentEventNotification/submit | Event Notification |
| camt.a13.001.03 | GET | /BatchEventDetails/{notification_identification}/event | Batch Payment Event Details |
| pain.a46.001.03 | POST | /payment/submitBatch | Batch Payment Initiation |
| pain.a47.001.01 | GET | /payment/batch/status | Batch Payment Status |
| pain.a48.001.01 | GET | /cancellations/{case_id}/status | Get Cancellation Status |

## PayTo APIs

| API | Method | API Name | Description |
|---|---|---|---|
| pain.a12.001.01 | POST | /Post/mandates | Create Mandate |
| pain.a13.001.01 | GET | /get/mandates/{mandateId} | Get Mandate By ID |
| pain.a14.001.01 | PATCH | /patch/mandates/{mandateId} | Amend Mandate |
| pain.a15.001.01 | GET | /get/mandates/{mandateId}/validity | Validate Mandate |
| pain.a16.001.01 | GET | /get/mandates/{mandateId}/status | Get Mandate Status |
| pain.a17.001.01 | PATCH | /patch/mandates/{mandateId}/status | Change Mandate Status |
| pain.a22.001.01 | GET | /get/mandates/actions/{actionId} | Get Mandate Action |

| | | | |
|---|---|---|---|
| pain.a23.001.01 | PATCH | /patch/mandates/actions/{actionId} | Resolve Mandate Action |
| pain.a36.001.01 | POST | /post/mandates/details | Mandate Enquiry |

## 1.3 Approach

This document describes a development framework and refers to other related reference material.

To facilitate use of the document, each section has sub-sections to highlight key themes or processes NPP Participants and third parties should consider in their development of API solutions for NPP. It is intended to be of use to both existing and prospective NPP Participants, Third Party Service Providers and software developers.

The use of the API Framework document as a source of information does not affect or alter:

(a) any rights or obligations of NPP Participants under the NPPA Regulations and NPP Procedures. For the avoidance of doubt, where there is any inconsistency between this document and the NPP Regulations and Procedures, or an NPPA-approved NPP design document, those documents prevail to the extent of the inconsistency; or

(b) the rights or obligations of NPP Participants as data controllers, to comply with privacy laws and to establish their own permission frameworks and requirements for secure data transfer.

## 1.4. API Sandbox

NPPA in conjunction with SWIFT have developed an API sandbox to help Third Party Service Providers and software developers to learn and test the NPP's capabilities via the available sample NPP APIs. External parties can build and test NPP based solutions in this independent environment.

To request access to the API Sandbox, please go to:
https://nppa-developer.swift.com/user/register

## 1.5 Glossary & abbreviations used in this document

| Term | Description |
|---|---|
| Addressing Service | Component of the NPP platform infrastructure that enables registration of customer account information and PayID (account proxy) information |
| API | Application Programming Interface |
| NPP Basic Infrastructure | NPP platform infrastructure that supports the processing of NPP payments |
| Connected Institutions | Connect directly to the NPP solely for the purposes of sending and receiving non-value messages such as payment initiation messages |

| Term | Description |
|---|---|
| Initiating Participant | An NPP Participant who sends payment initiation requests on behalf of a customer |
| FSS | Refers to the RBA Fast Settlement Service that settles NPP payments in real-time |
| JSON | JavaScript Object Notation |
| NPP Participant | Connects directly to the NPP for the purposes of clearing and settling NPP Payments |
| Mandate | Refers to a PayTo agreement that enables a payer customer to pre-authorise third parties to initiate payments from the customer's bank account |
| MMS | Refers to the Mandate Management Service which is a centralised access-controlled database managed by NPPA that stores mandate records |
| Overlay Service | Refers to a payment service, or payment-related service, using the NPP |
| Overlay Service Provider | Overlay Service Providers provide customised Overlay Services that sit on top of the NPP and utilise the underlying infrastructure. Overlay Service Providers may also be Connected Institutions |
| PayID | Refers to an alias record in the Addressing Service. The Addressing Service provides a mechanism that allows a registered PayID (acting as a unique identifier for a customer account) to be resolved to a bank account. A PayID can be one of four alias types; phone number, email address, ABN or Organisational Identifier |
| RESTful | Representational state transfer (REST) or RESTful web services |
| Third Party Service Provider | Payment service providers that are third parties (i.e. not any of the following: owner of the account, the account servicer or the account servicer's sponsoring Participant) |

# 2 NPP Open API Design Principles

## 2.1 Basics

The following key design principles incorporate both RESTful concepts and ISO20022 as the data standard and describes additional considerations in the development of APIs for use on the NPP.

### 2.1.1 RESTful APIs

Each NPP API should adhere to the RESTful API concepts as *the transfer standard* of choice.

A RESTful API is a set of Hypertext Transfer Protocol (HTTP) request messages, along with a definition of the structure of response messages, which is in a JavaScript Object Notation (JSON) format.

Overall, the priority should be to have an API that is simple to understand and easy to use. In instances where following RESTful principles would be convoluted and complex, the principles have not been followed.

References:

- The highest level Data Description Language used is the JSON Schema: **http://json-schema.org/**
- Best Practice has also been taken from the Data Description Language for APIs; JSON API: **http://jsonapi.org/**
- The Interface Description Language used is the Swagger Specification version 2.0 (also known as Open API): **http://swagger.io/**

### 2.1.2 ISO 20022

NPP API payloads should be designed based on existing ISO 20022 message elements and components, where available, as *the data standard*. Intended to decrease implementation time for developers to consume, these APIs support interoperability with the asynchronous NPP message formats currently used across the Platform.

The principles applied to the re-use of ISO message elements and components are:

- Where relevant – the API payloads should be flattened so that they are more developer friendly.
- Only elements that are required for the functioning of the API endpoint should be included in the API payload. API endpoints are defined for specific use-cases (not to be generically extensible for all use-cases). For example - only elements that are required for a single immediate payment initiation would be included in the Payment API payload.
- Support modification of ISO 20022 elements where the existing standard does not cater for an API context (such as filtering, pagination etc.). For example, latitude and longitude in decimal format - as this is how developers will work with latitude and longitude; or using simple types (e.g. a single date-time field) instead of a complex type (e.g. a choice field with a nesting of date and time).
- Consideration of *ISO 20022 and JSON: An Implementation Best Practices.* Includes pertinent modelling guides

### 2.1.3 Security Standards

Each NPP Participant is responsible for setting the security standards for Third Party Service Providers connecting to that NPP Participant via APIs.

NPPA recommends the use of global best practice in security standards where possible.

### 2.1.4 NPP Overlay Service Identification

In order to support in each common API request call, a mechanism to vary the API call attributes according to Overlay Service, the Service Level element (from <SvcLvl/Prtry> element in ISO 20022> is defined as the first element in the API request. The Service Level element is used to distinguish the specific messaging or overlay service under which an API (or XML message) is being used e.g.: npp.msg.01-x2p1.03 (example of an overlay service for basic messaging). It is optional to send Service Level information.

### 2.1.5 Status Codes

Each API may need to consider three status codes that serve different purposes:

- The HTTP Status Code reflects the outcome of the API call (the HTTP operation on the resource).
  E.g. 200 - OK , 400 - Bad Request, 405 - Method Not Allowed
- In the API content, where a status could be returned to reflect the outcome of the request. For example, the Status field in a Payment API payload could reflect the status of a specific payment that makes use of the ISO 20022 PaymentStatusCode code-list enumeration (external code list) to report status. E.g. CH11 - Creditor Identifier Incorrect
- Participant specific status codes. Each NPP Participant may have a specific set of status codes that it has defined for API connectivity to its own infrastructure.

### 2.1.6 Notification and Callback Events

The API Framework requires a number of events to be notified to the API user. The Framework notifies the user of an event and requires the user to call back to the NPP Participant so that credentials can be validated before details are shared. This allows the NPP Participant to protect the data security and integrity of the resource.

Events are created for the API User as a notification of activity related to a Payment or Batch Processing. Payment Event Notification (camt.a12) API allows an NPP Participant to notify the API user that a payment event has occurred in their account. The customer can then use the notification ID returned in Payment Event Notification and use Payment Event Details or Batch Event Details to fetch details about the payment or batch.

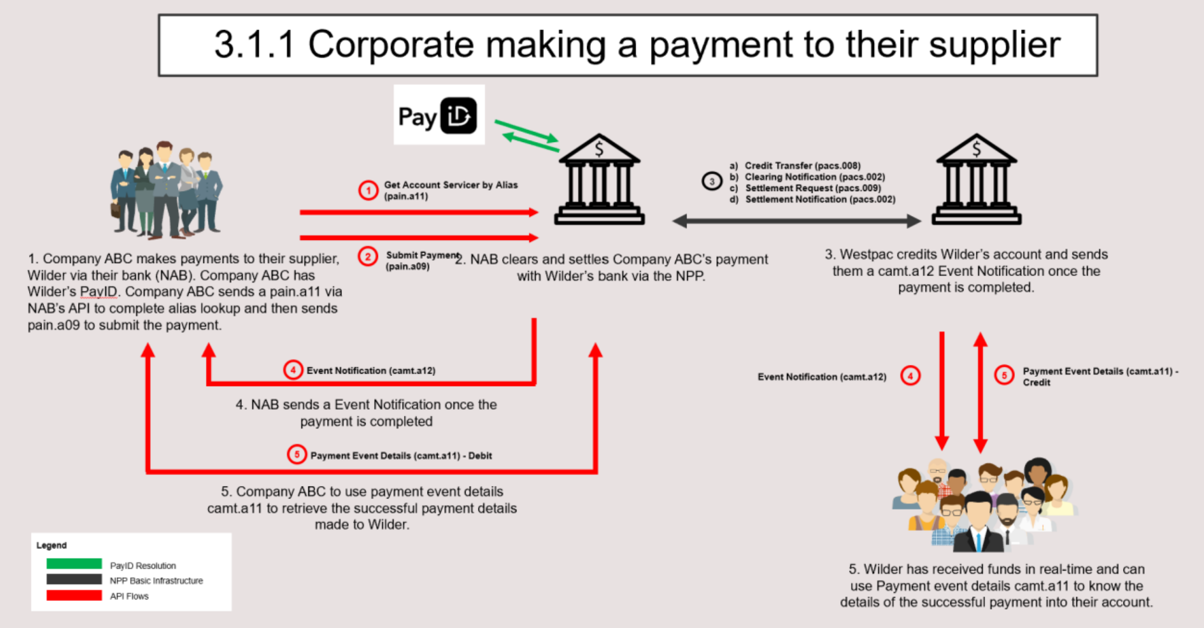
# 3 NPP Open API Use Cases

## 3.1 Payment Initiation

The Usage example (Figure 1) and payment flow (Figure 2 below) illustrates an NPP payment initiation process that *could* be realised as an NPP real-time Credit Transfer through the use of a series of common open APIs made available by NPP Participants and associated institutions. APIs are foreseen as complementary to other channels such as asynchronous messaging or web forms / mobile phone apps.

### 3.1.1 Usage Example

Figure 1 below illustrates the example of a corporate paying their supplier and the associated interactions and the work flow that might take place:



3.1.1 Corporate making a payment to their supplier

1. Company ABC makes payments to their supplier, Wilder via their bank (NAB). Company ABC has Wilder's PayID. Company ABC sends a pain.a11 via NAB's API to complete alias lookup and then sends pain.a09 to submit the payment.

2. NAB clears and settles Company ABC's payment with Wilder's bank via the NPP.

3. Westpac credits Wilder's account and sends them a camt.a12 Event Notification once the payment is completed.

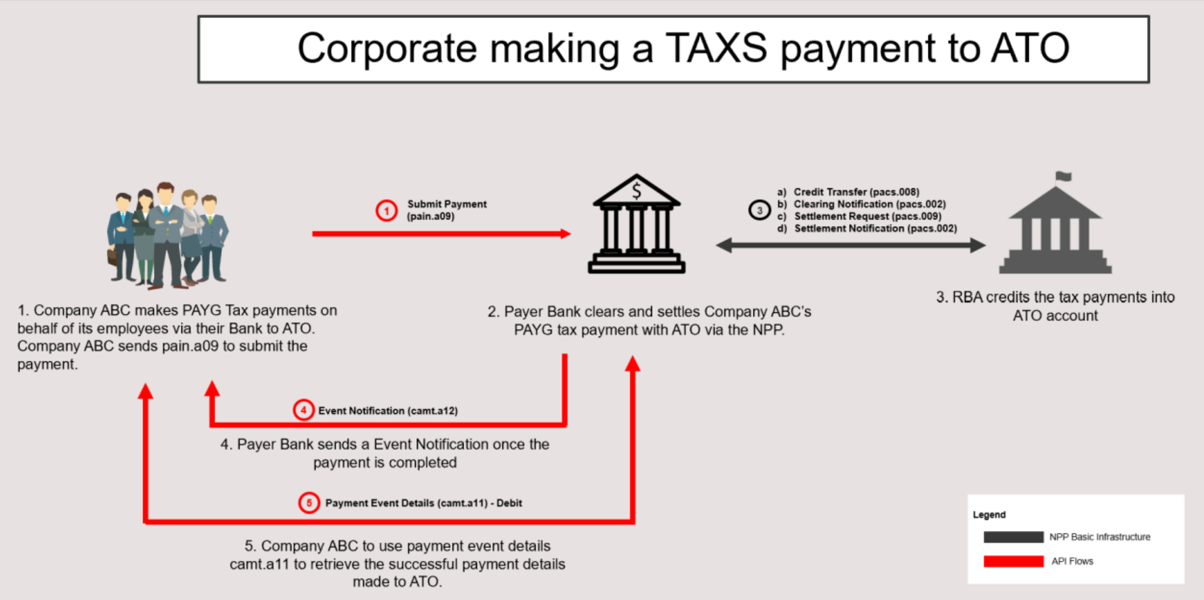
4. NAB sends a Event Notification once the payment is completed

5. Company ABC to use payment event details camt.a11 to retrieve the successful payment details made to Wilder.

5. Wilder has received funds in real-time and can use Payment event details camt.a11 to know the details of the successful payment into their account.

PayiD
1 Get Account Servicer by Alias (pain.a11)
2 Submit Payment (pain.a09)
3
a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
4 Event Notification (camt.a12)
5 Payment Event Details (camt.a11) - Debit
Event Notification (camt.a12) 4
5 Payment Event Details (camt.a11) - Credit
Legend
PayID Resolution
NPP Basic Infrastructure
API Flows

*Figure 1. Corporate making a payment to their supplier*



Corporate making a TAXS payment to ATO

1 Submit Payment (pain.a09)

3 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)

1. Company ABC makes PAYG Tax payments on behalf of its employees via their Bank to ATO. Company ABC sends pain.a09 to submit the payment.

2. Payer Bank clears and settles Company ABC's PAYG tax payment with ATO via the NPP.

3. RBA credits the tax payments into ATO account

4 Event Notification (camt.a12)

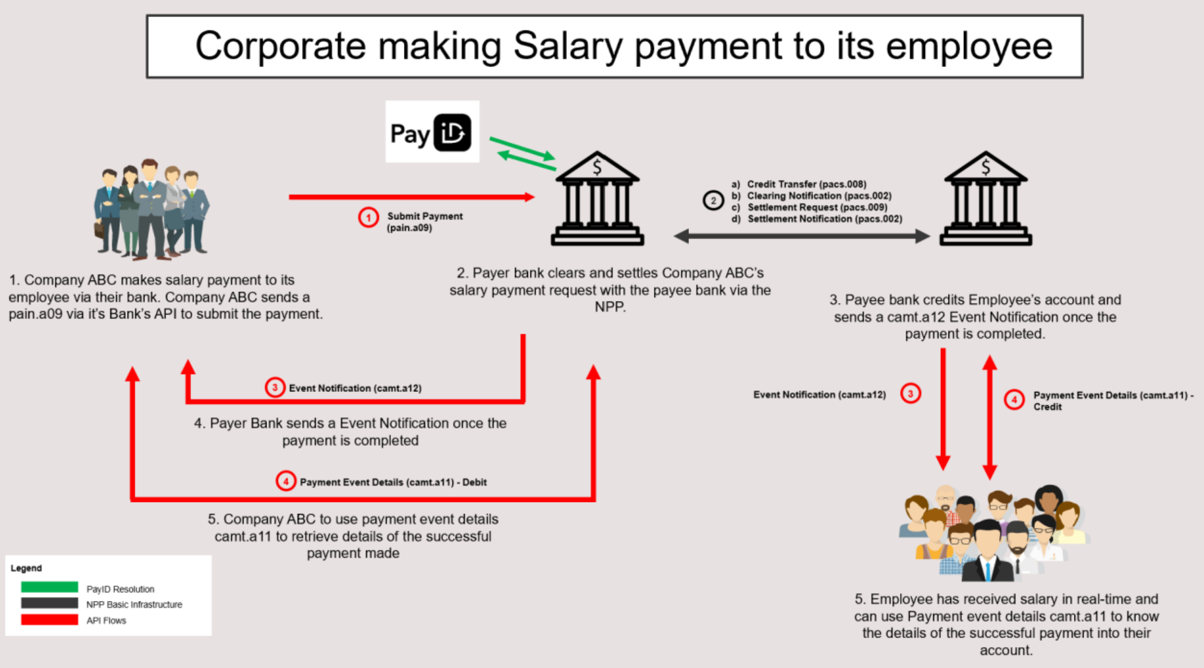
4. Payer Bank sends a Event Notification once the payment is completed

5 Payment Event Details (camt.a11) - Debit

5. Company ABC to use payment event details camt.a11 to retrieve the successful payment details made to ATO.

Legend
⬛ NPP Basic Infrastructure
🟥 API Flows

*Figure 2. Corporate making a Tax payment to the ATO*



Corporate making Salary payment to its employee

1 Submit Payment (pain.a09)

2 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)

1. Company ABC makes salary payment to its employee via their bank. Company ABC sends a pain.a09 via it's Bank's API to submit the payment.

2. Payer bank clears and settles Company ABC's salary payment request with the payee bank via the NPP.

3. Payee bank credits Employee's account and sends a camt.a12 Event Notification once the payment is completed.

3 Event Notification (camt.a12)

4. Payer Bank sends a Event Notification once the payment is completed

4 Payment Event Details (camt.a11) - Debit

5. Company ABC to use payment event details camt.a11 to retrieve details of the successful payment made

Event Notification (camt.a12) 3

4 Payment Event Details (camt.a11) - Credit

5. Employee has received salary in real-time and can use Payment event details camt.a11 to know the details of the successful payment into their account.

Legend
🟩 PayID Resolution
⬛ NPP Basic Infrastructure
🟥 API Flows

*Figure 3. Corporate making salary payments to employees*

### 3.1.2 Payment Flow Processing Steps

The sequence of processing steps in this section is provided as an illustrative example of a possible payment flow using APIs; it is a guide only and is not meant to be prescriptive.

Note: In addition to basic payments, the payment reflected below may also represent a Salary, Tax or Superannuation payment. In this case, the relevant classification of this payment should be used, e.g. SALA, TAXS or PENS.

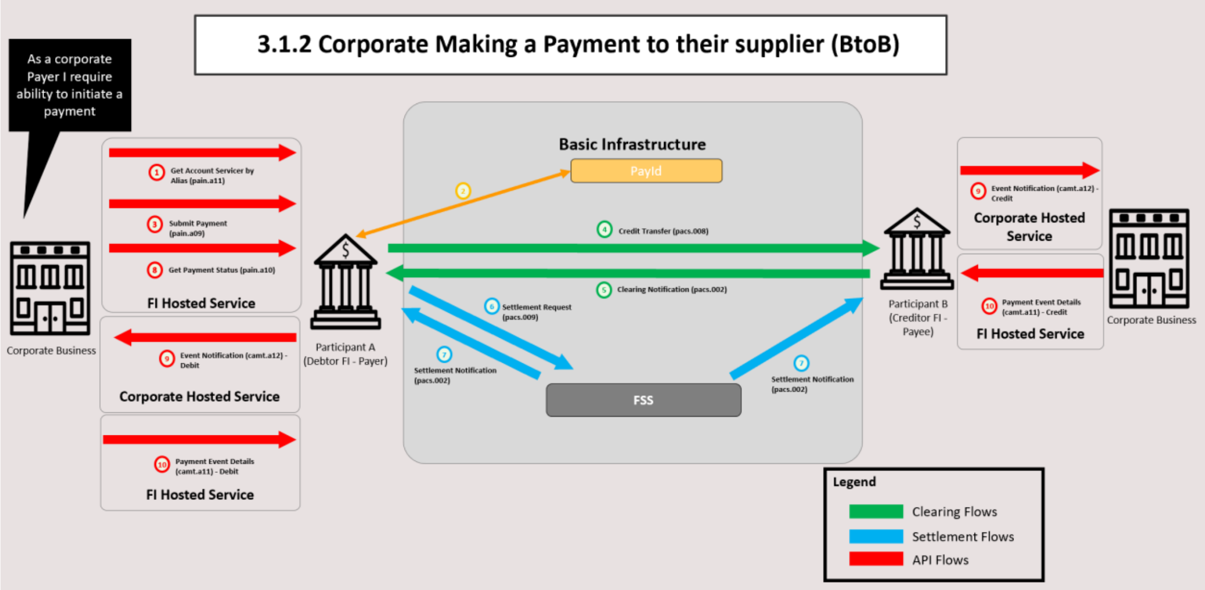

3.1.2 Corporate Making a Payment to their supplier (BtoB)
As a corporate Payer I require ability to initiate a payment
Corporate Business
1 Get Account Servicer by Alias (pain.a11)
3 Submit Payment (pain.a09)
8 Get Payment Status (pain.a10)
FI Hosted Service
9 Event Notification (camt.a12) - Debit
Corporate Hosted Service
10 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Participant A (Debtor FI - Payer)
Basic Infrastructure
PayId
2
4 Credit Transfer (pacs.008)
5 Clearing Notification (pacs.002)
6 Settlement Request (pacs.009)
7 Settlement Notification (pacs.002)
FSS
7 Settlement Notification (pacs.002)
Participant B (Creditor FI - Payee)
9 Event Notification (camt.a12) - Credit
Corporate Hosted Service
10 Payment Event Details (camt.a11) - Credit
FI Hosted Service
Corporate Business
Legend
Clearing Flows
Settlement Flows
API Flows

*Figure 4. Corporate making a payment to their supplier*

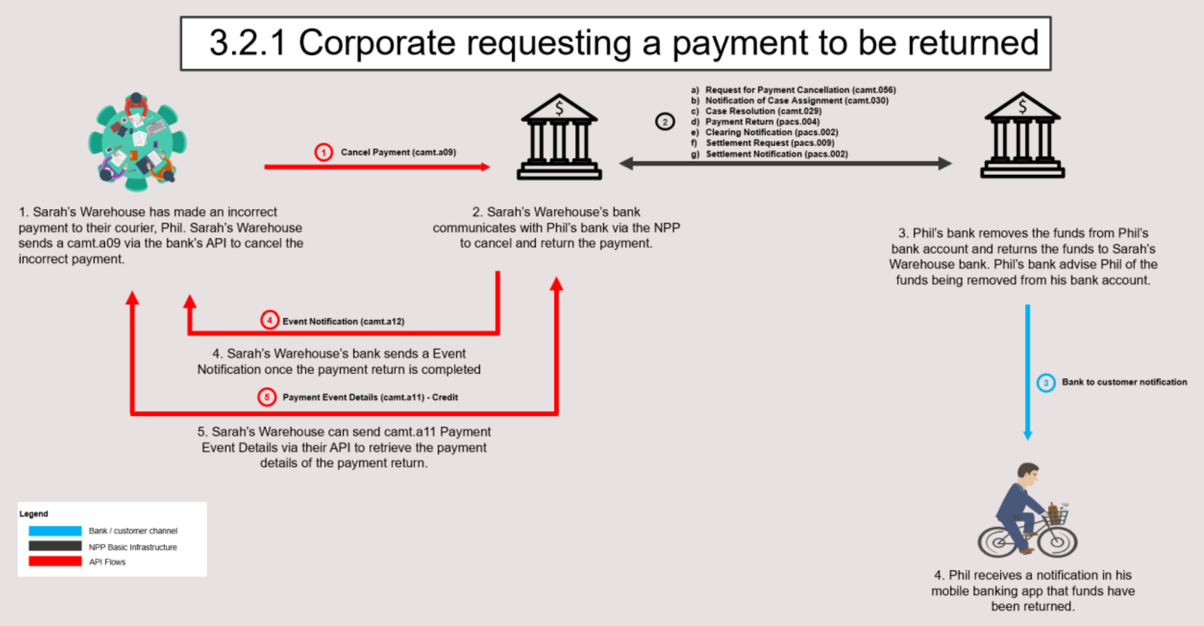
| Step | Description |
|---|---|
| 1-2 | Payer business initiates a check for a valid PayID via their NPP Participant before submitting a payment initiation |
| 3 | Payer business submits a payment initiation request to their NPP Participant |
| 4 | Payer's NPP Participant creates a clearing request with details from the payment initiation request and PayID which is routed via the NPP Basic Infrastructure to the Payee's NPP Participant |
| 5 | Payee's NPP Participant accepts clearing request and responds with a clearing notification |
| 6-7 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 8 | Payer's NPP Participant responds to Payment Status (successful / unsuccessful) to Payer |
| 9 | Payer's NPP Participant notifies Payer Business of a payment event in their account.<br>Payee's NPP Participant notifies Payee Business of a payment event in their account. |

| Step | Description |
|---|---|
| 10 | Payer Business gets details of a debit from their account from the Payer’s NPP Participant. Payee Business gets details of a credit to their account from the Payee’s NPP Participant. |

## 3.2 Cancel Payment

### 3.2.1 Usage Example

The figure below illustrates the example of a corporate requesting the return of an incorrect payment and the associated interactions and the work flow that might take place:



3.2.1 Corporate requesting a payment to be returned

1. Sarah’s Warehouse has made an incorrect payment to their courier, Phil. Sarah’s Warehouse sends a camt.a09 via the bank’s API to cancel the incorrect payment.

2. Sarah’s Warehouse’s bank communicates with Phil’s bank via the NPP to cancel and return the payment.

3. Phil’s bank removes the funds from Phil’s bank account and returns the funds to Sarah’s Warehouse bank. Phil’s bank advise Phil of the funds being removed from his bank account.

4. Sarah’s Warehouse’s bank sends a Event Notification once the payment return is completed

5. Sarah’s Warehouse can send camt.a11 Payment Event Details via their API to retrieve the payment details of the payment return.

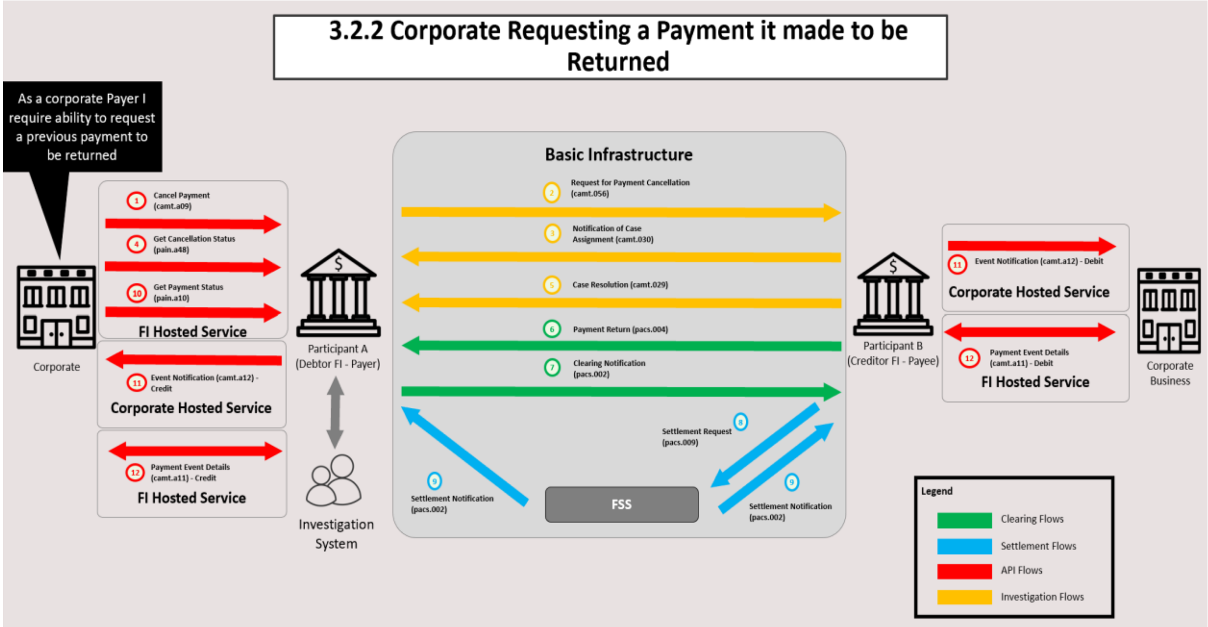
4. Phil receives a notification in his mobile banking app that funds have been returned.

1 Cancel Payment (camt.a09)
2
a) Request for Payment Cancellation (camt.056)
b) Notification of Case Assignment (camt.030)
c) Case Resolution (camt.029)
d) Payment Return (pacs.004)
e) Clearing Notification (pacs.002)
f) Settlement Request (pacs.009)
g) Settlement Notification (pacs.002)
3 Bank to customer notification
4 Event Notification (camt.a12)
5 Payment Event Details (camt.a11) - Credit
Legend
Bank / customer channel
NPP Basic Infrastructure
API Flows

*Figure 5. Corporate requesting a payment to be returned*

### 3.2.2 Payment Cancellation Flow Processing Steps

The sequence of processing steps in this section is provided as an illustrative example of a possible payment flow using APIs; it is a guide only and is not meant to be prescriptive.



3.2.2 Corporate Requesting a Payment it made to be Returned
As a corporate Payer I require ability to request a previous payment to be returned
Corporate
1 Cancel Payment (camt.a09)
4 Get Cancellation Status (pain.a48)
10 Get Payment Status (pain.a10)
FI Hosted Service
11 Event Notification (camt.a12) - Credit
Corporate Hosted Service
12 Payment Event Details (camt.a11) - Credit
FI Hosted Service
Participant A (Debtor FI - Payer)
Investigation System
Basic Infrastructure
2 Request for Payment Cancellation (camt.056)
3 Notification of Case Assignment (camt.030)
5 Case Resolution (camt.029)
6 Payment Return (pacs.004)
7 Clearing Notification (pacs.002)
8 Settlement Request (pacs.009)
9 Settlement Notification (pacs.002)
FSS
9 Settlement Notification (pacs.002)
Participant B (Creditor FI - Payee)
11 Event Notification (camt.a12) - Debit
Corporate Hosted Service
12 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Corporate Business
Legend
🟩 Clearing Flows
🟦 Settlement Flows
🟥 API Flows
🟨 Investigation Flows

*Figure 6. Corporate requesting a payment it made to be returned*

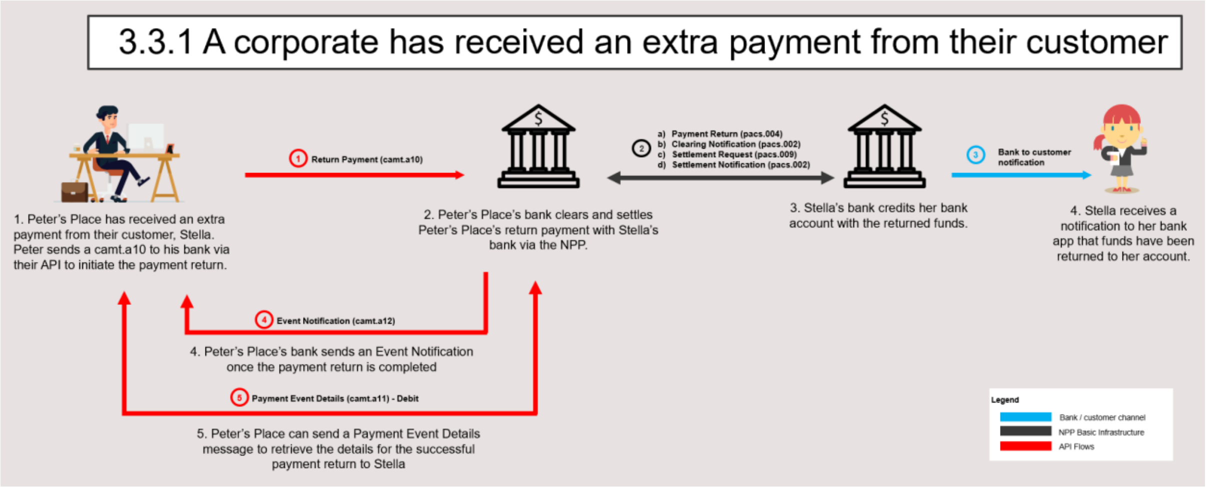
| Step | Description |
|---|---|
| 1 | Payer business submits a request to return a previously submitted payment to their NPP Participant |
| 2 | Payer's NPP Participant creates a payment cancellation request with details from the cancel payment request which is routed via the NPP Basic Infrastructure to the Payee's NPP Participant |
| 3, 5,6 | Payer's NPP Participant and Payee's NPP Participant will use Investigation messages and Payment return messages to process the Payment cancellation request |
| 4 | Payer's NPP Participant responds to Cancellation Status (Cancellation Pending / Cancellation Accepted / Cancellation Rejected / Return Pending / Return Rejected / Return Settled) |
| 7-8 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 9 | Payer's NPP Participant responds to Payment Status (successful / unsuccessful) to Payer |
| 10 | Payer's NPP Participant notifies Payer Business of a payment event in their account. |

| Step | Description |
|---|---|
| 11 | Payer Business gets details of a credit to their account from the Payer’s NPP Participant. |

## 3.3 Return Payment

### 3.3.1 Usage Example

The figure below illustrates the example of a corporate returning an extra payment from their customer and the associated interactions and the work flow that might take place:



3.3.1 A corporate has received an extra payment from their customer

1. Peter’s Place has received an extra payment from their customer, Stella. Peter sends a camt.a10 to his bank via their API to initiate the payment return.

2. Peter’s Place’s bank clears and settles Peter’s Place’s return payment with Stella’s bank via the NPP.

3. Stella’s bank credits her bank account with the returned funds.

4. Stella receives a notification to her bank app that funds have been returned to her account.

4. Peter’s Place’s bank sends an Event Notification once the payment return is completed

5. Peter’s Place can send a Payment Event Details message to retrieve the details for the successful payment return to Stella

1 Return Payment (camt.a10)
2
a) Payment Return (pacs.004)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
3 Bank to customer notification
4 Event Notification (camt.a12)
5 Payment Event Details (camt.a11) - Debit
Legend
Bank / customer channel
NPP Basic Infrastructure
API Flows

*Figure 7. A corporate receives an extra payment from their customer*

### 3.3.2 Return Payment Flow Processing Steps

The sequence of processing steps in this section is provided as an illustrative example of a possible payment flow using APIs; it is a guide only and is not meant to be prescriptive.

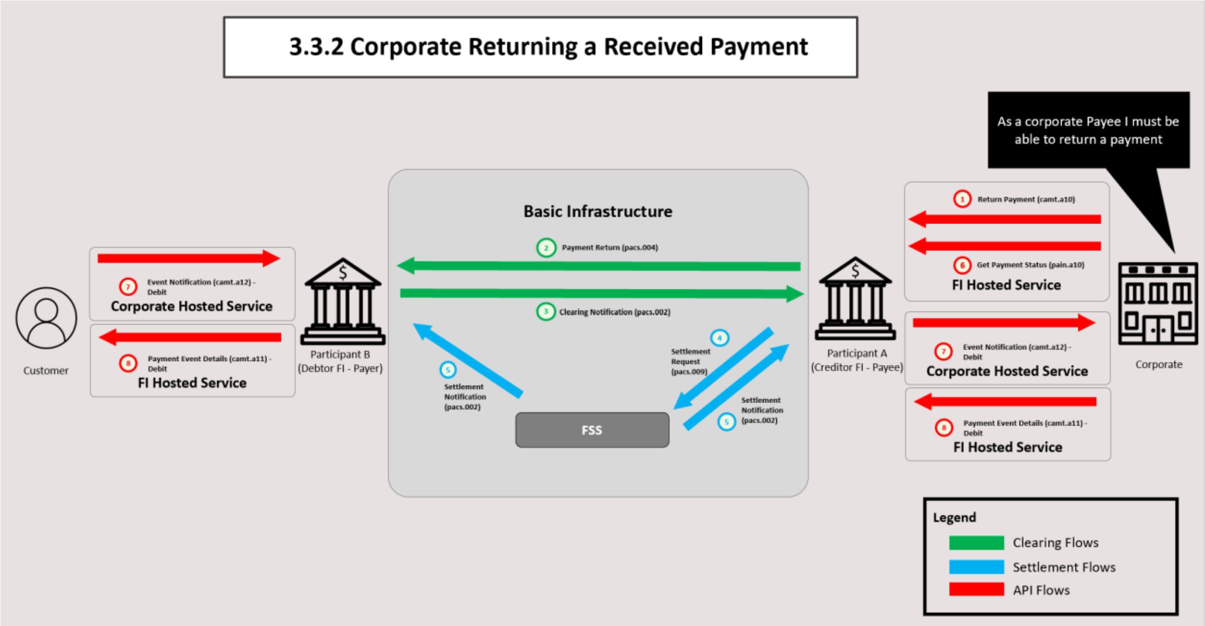

3.3.2 Corporate Returning a Received Payment
As a corporate Payee I must be able to return a payment
Customer
7 Event Notification (camt.a12) - Debit
Corporate Hosted Service
8 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Participant B (Debtor FI - Payer)
Basic Infrastructure
2 Payment Return (pacs.004)
3 Clearing Notification (pacs.002)
5 Settlement Notification (pacs.002)
4 Settlement Request (pacs.009)
Settlement Notification (pacs.002) 5
FSS
Participant A (Creditor FI - Payee)
1 Return Payment (camt.a10)
6 Get Payment Status (pain.a10)
FI Hosted Service
7 Event Notification (camt.a12) - Debit
Corporate Hosted Service
8 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Corporate
Legend
Clearing Flows
Settlement Flows
API Flows

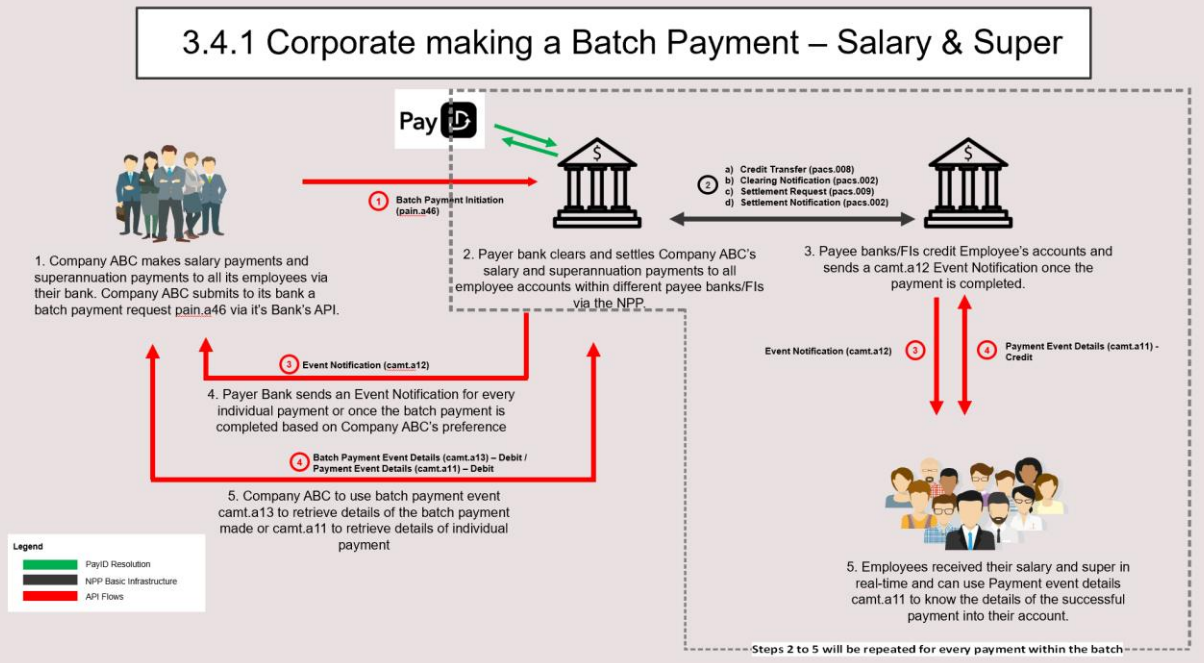
*Figure 8. Corporate returning a received payment*

| Step | Description |
|---|---|
| 1 | Payee business submits a request to return a previously received payment to their NPP Participant |
| 2-3 | Payee’s NPP Participant sends a Payment return request to the Payer’s NPP Participant and Payer’s NPP Participant sends a clearing notification for return to the Payee’s NPP Participant. |
| 4-5 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 6 | Payee’s NPP Participant responds to Payment Status (successful / unsuccessful) to Payee |
| 7 | Payee’s NPP Participant notifies Payee Business of a payment event in their account. |
| 8 | Payee Business gets details of a debit from their account from the Payee’s NPP Participant. |

## 3.4 Batch Payment Flow Processing Steps

### 3.4.1 Usage Example

The sequence of processing steps in this section is provided as an illustrative example of a possible batch payment flow using APIs; it is a guide only and is not meant to be prescriptive.



3.4.1 Corporate making a Batch Payment – Salary & Super

1. Company ABC makes salary payments and superannuation payments to all its employees via their bank. Company ABC submits to its bank a batch payment request pain.a46 via it's Bank's API.

2. Payer bank clears and settles Company ABC's salary and superannuation payments to all employee accounts within different payee banks/FIs via the NPP.

3. Payee banks/FIs credit Employee's accounts and sends a camt.a12 Event Notification once the payment is completed.

4. Payer Bank sends an Event Notification for every individual payment or once the batch payment is completed based on Company ABC's preference

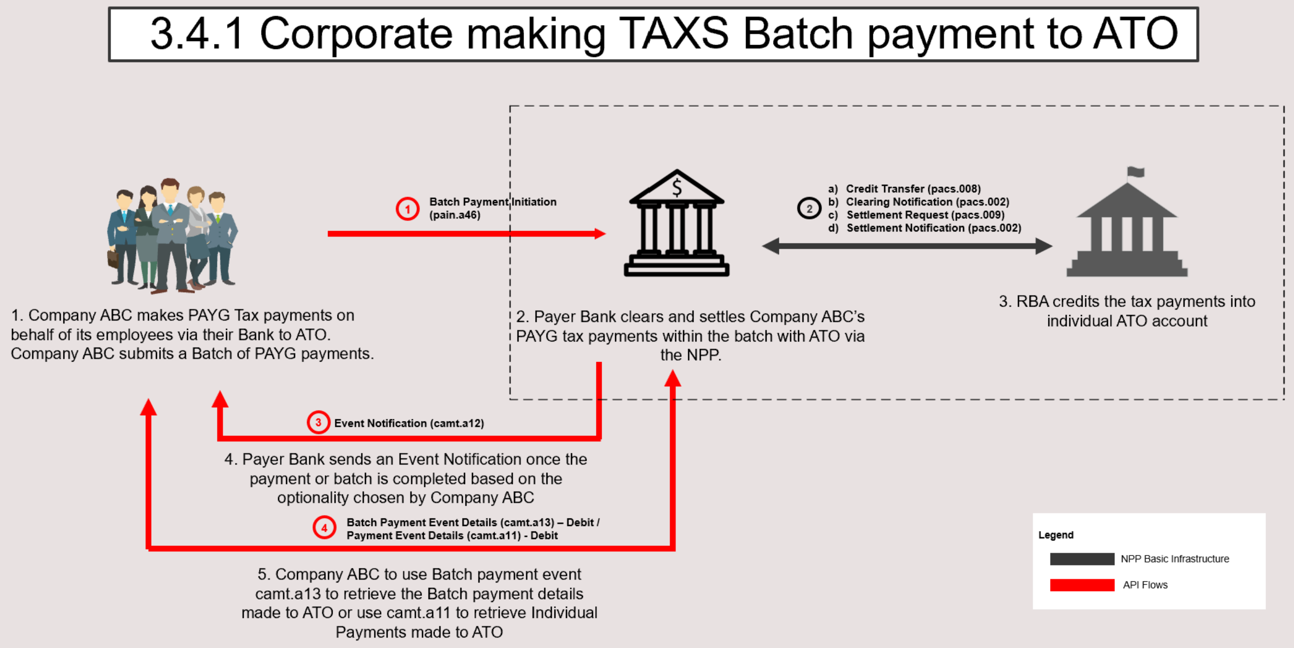
5. Company ABC to use batch payment event camt.a13 to retrieve details of the batch payment made or camt.a11 to retrieve details of individual payment

5. Employees received their salary and super in real-time and can use Payment event details camt.a11 to know the details of the successful payment into their account.

Steps 2 to 5 will be repeated for every payment within the batch

PayID
1 Batch Payment Initiation (pain.a46)
2 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
Event Notification (camt.a12) 3
4 Payment Event Details (camt.a11) - Credit
3 Event Notification (camt.a12)
4 Batch Payment Event Details (camt.a13) – Debit / Payment Event Details (camt.a11) – Debit
Legend
PayID Resolution
NPP Basic Infrastructure
API Flows

*Figure 9. Corporate making a batch payment*



3.4.1 Corporate making TAXS Batch payment to ATO

1. Company ABC makes PAYG Tax payments on behalf of its employees via their Bank to ATO. Company ABC submits a Batch of PAYG payments.

2. Payer Bank clears and settles Company ABC's PAYG tax payments within the batch with ATO via the NPP.

3. RBA credits the tax payments into individual ATO account

4. Payer Bank sends an Event Notification once the payment or batch is completed based on the optionality chosen by Company ABC

5. Company ABC to use Batch payment event camt.a13 to retrieve the Batch payment details made to ATO or use camt.a11 to retrieve Individual Payments made to ATO

1 Batch Payment Initiation (pain.a46)
2 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
3 Event Notification (camt.a12)
4 Batch Payment Event Details (camt.a13) – Debit / Payment Event Details (camt.a11) - Debit
Legend
NPP Basic Infrastructure
API Flows

*Figure 10. Corporate making a tax batch payment to the ATO*

### 3.4.2 Batch Payment Flow Processing Steps

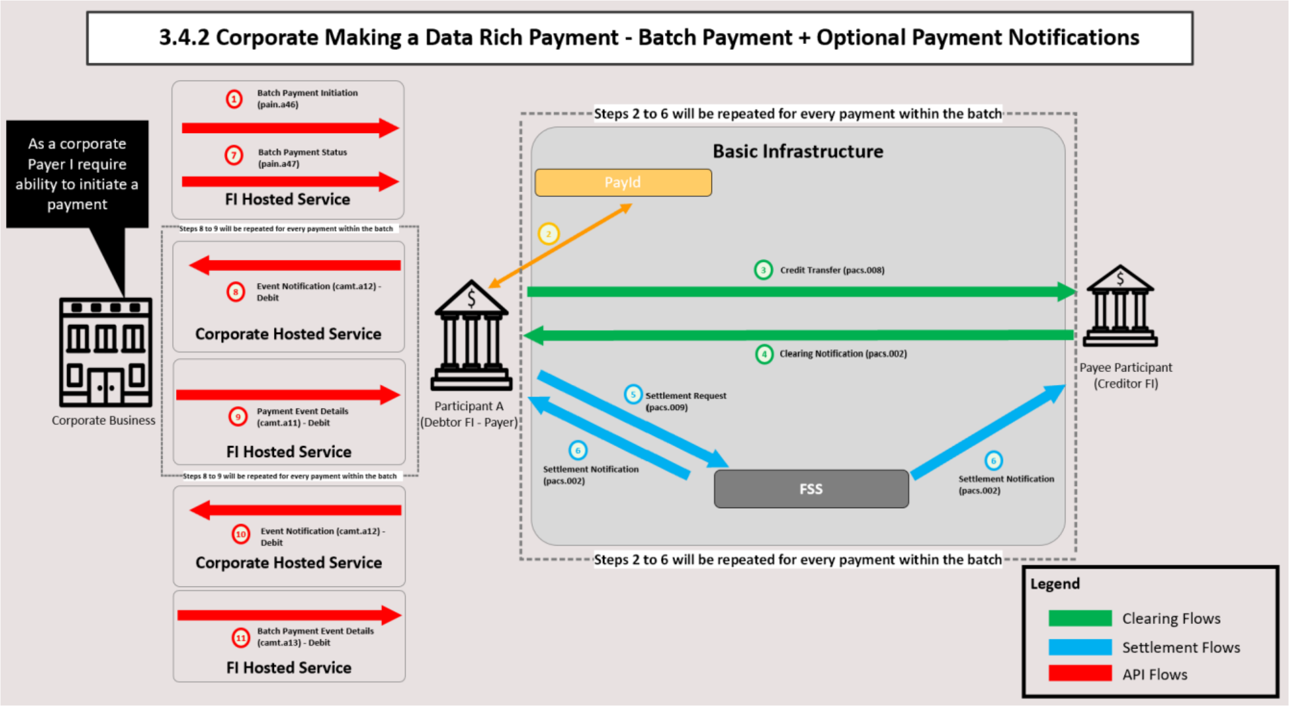

3.4.2 Corporate Making a Data Rich Payment - Batch Payment + Optional Payment Notifications
As a corporate Payer I require ability to initiate a payment
Corporate Business
1 Batch Payment Initiation (pain.a46)
7 Batch Payment Status (pain.a47)
FI Hosted Service
Steps 8 to 9 will be repeated for every payment within the batch
8 Event Notification (camt.a12) - Debit
Corporate Hosted Service
9 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Steps 8 to 9 will be repeated for every payment within the batch
10 Event Notification (camt.a12) - Debit
Corporate Hosted Service
11 Batch Payment Event Details (camt.a13) - Debit
FI Hosted Service
Participant A (Debtor FI - Payer)
Steps 2 to 6 will be repeated for every payment within the batch
Basic Infrastructure
PayId
2
3 Credit Transfer (pacs.008)
4 Clearing Notification (pacs.002)
5 Settlement Request (pacs.009)
6 Settlement Notification (pacs.002)
FSS
6 Settlement Notification (pacs.002)
Steps 2 to 6 will be repeated for every payment within the batch
Payee Participant (Creditor FI)
Legend
Clearing Flows
Settlement Flows
API Flows

*Figure 11. Corporate making a data-rich payment*

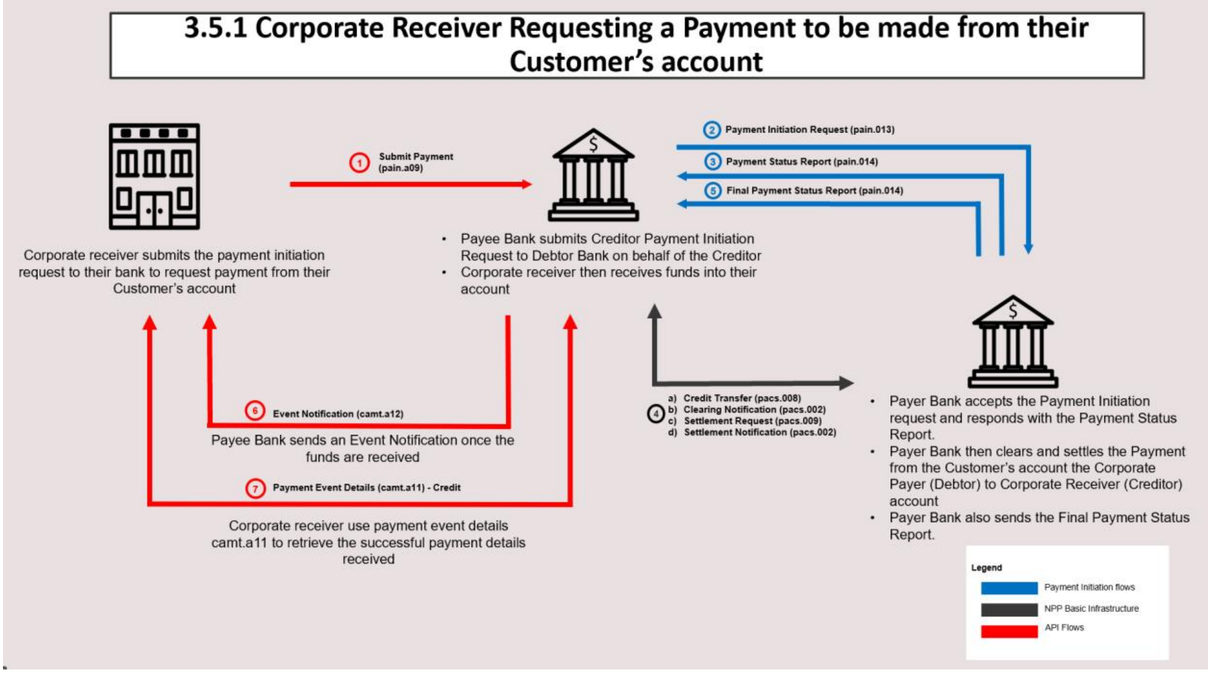
| Step | Description |
|---|---|
| 1-2 | Payer business assembles a batch of payments (e.g. for salary, TAXS tax, or Superannuation payments), and when required initiates a check for a valid PayID for a payee via their NPP Participant before submitting a batch payment initiation.<br>Payer business submits a batch payment initiation request to their NPP Participant |
| 3 | Payer's NPP Participant de-batches the submission and creates a clearing request with details from each transaction in the batch payment initiation request and PayID which is routed via the NPP Basic Infrastructure to the Payee's NPP Participant |
| 4 | Payee's NPP Participant accepts clearing request and responds with a clearing notification |
| 5-6 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 7 | Payer's NPP Participant responds with Batch Payment Status (successful / unsuccessful) to Payer |
| 8 | Payer business can optionally choose to be notified either after every individual payment or just once after the entire batch has been processed.<br>Payer Participant notifies Payer Business for every single payment within the batch. Payer Participant sends a notification for each payment event. |

| Step | Description |
|---|---|
| 9 | Payer Business gets details of individual debits from their account from the Payer’s NPP Participant. |
| 10 | Payer’s NPP Participant notifies Payer Business of a Batch Payment Event reporting debits to their account(s). Payee’s NPP Participant notifies Payee Business of a payment event reporting a credit to their account. |
| 11 | Payer Business gets details of batch debit(s) from their account from the Payer’s NPP Participant. Payee Business gets details of a credit to their account from the Payee’s NPP Participant. |

## 3.5 Requesting a Payment

### 3.5.1 Usage Example

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for requesting a payment to be made using APIs; it is a guide only and is not meant to be prescriptive.



3.5.1 Corporate Receiver Requesting a Payment to be made from their Customer’s account

Corporate receiver submits the payment initiation request to their bank to request payment from their Customer’s account

- Payee Bank submits Creditor Payment Initiation Request to Debtor Bank on behalf of the Creditor
- Corporate receiver then receives funds into their account

Payee Bank sends an Event Notification once the funds are received

Corporate receiver use payment event details camt.a11 to retrieve the successful payment details received

- Payer Bank accepts the Payment Initiation request and responds with the Payment Status Report.
- Payer Bank then clears and settles the Payment from the Customer’s account the Corporate Payer (Debtor) to Corporate Receiver (Creditor) account
- Payer Bank also sends the Final Payment Status Report.

1 Submit Payment (pain.a09)
2 Payment Initiation Request (pain.013)
3 Payment Status Report (pain.014)
5 Final Payment Status Report (pain.014)
4 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
6 Event Notification (camt.a12)
7 Payment Event Details (camt.a11) - Credit
Legend
Payment Initiation flows
NPP Basic Infrastructure
API Flows

*Figure 12. Corporate receives a request for payment from their customer's account*

## 3.5.2 Requesting a Payment Flow Processing Steps

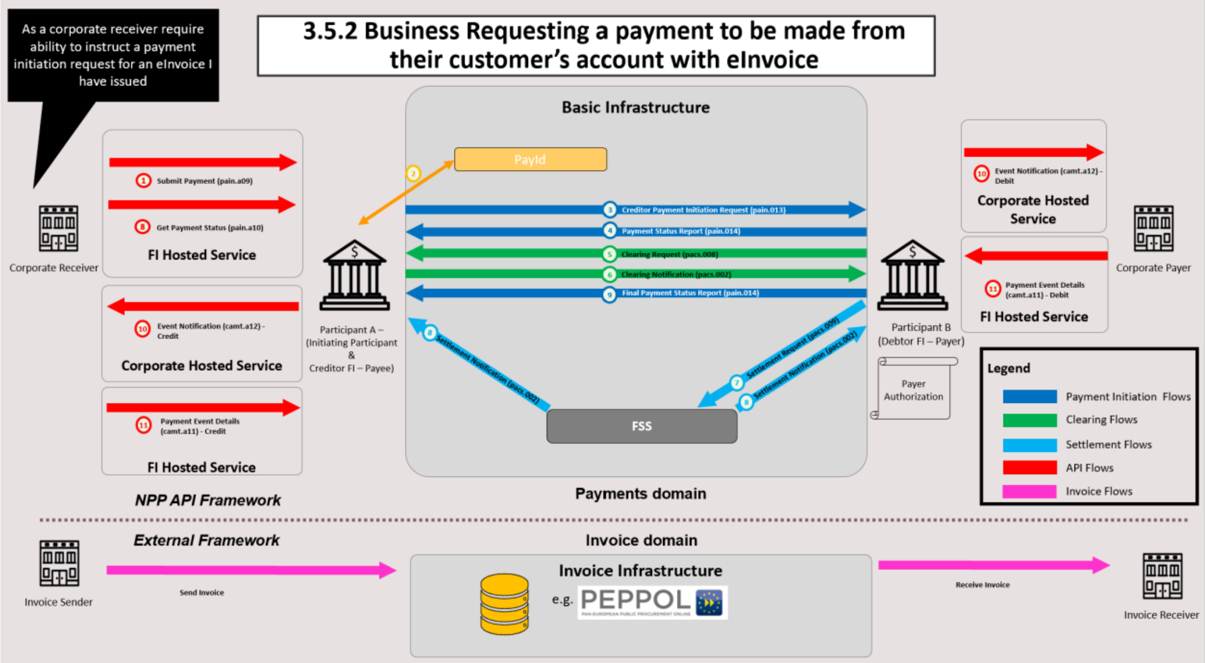

3.5.2 Business Requesting a payment to be made from their customer's account with eInvoice
As a corporate receiver require ability to instruct a payment initiation request for an eInvoice I have issued
Corporate Receiver
1 Submit Payment (pain.a09)
8 Get Payment Status (pain.a10)
FI Hosted Service
10 Event Notification (camt.a12) - Credit
Corporate Hosted Service
11 Payment Event Details (camt.a11) - Credit
FI Hosted Service
Participant A – (Initiating Participant & Creditor FI – Payee)
Basic Infrastructure
2
PayId
3 Creditor Payment Initiation Request (pain.013)
4 Payment Status Report (pain.014)
5 Clearing Request (pacs.008)
6 Clearing Notification (pacs.002)
9 Final Payment Status Report (pain.014)
8 Settlement Notification (pacs.002)
7 Settlement Request (pacs.009)
8 Settlement Notification (pacs.002)
FSS
Participant B (Debtor FI – Payer)
Payer Authorization
10 Event Notification (camt.a12) - Debit
Corporate Hosted Service
11 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Corporate Payer
Legend
Payment Initiation Flows
Clearing Flows
Settlement Flows
API Flows
Invoice Flows
NPP API Framework
Payments domain
External Framework
Invoice domain
Invoice Sender
Send Invoice
Invoice Infrastructure
e.g. PEPPOL
Receive Invoice
Invoice Receiver

*Figure 13. Business requests a payment to be made from their customer's account with eInvoice*

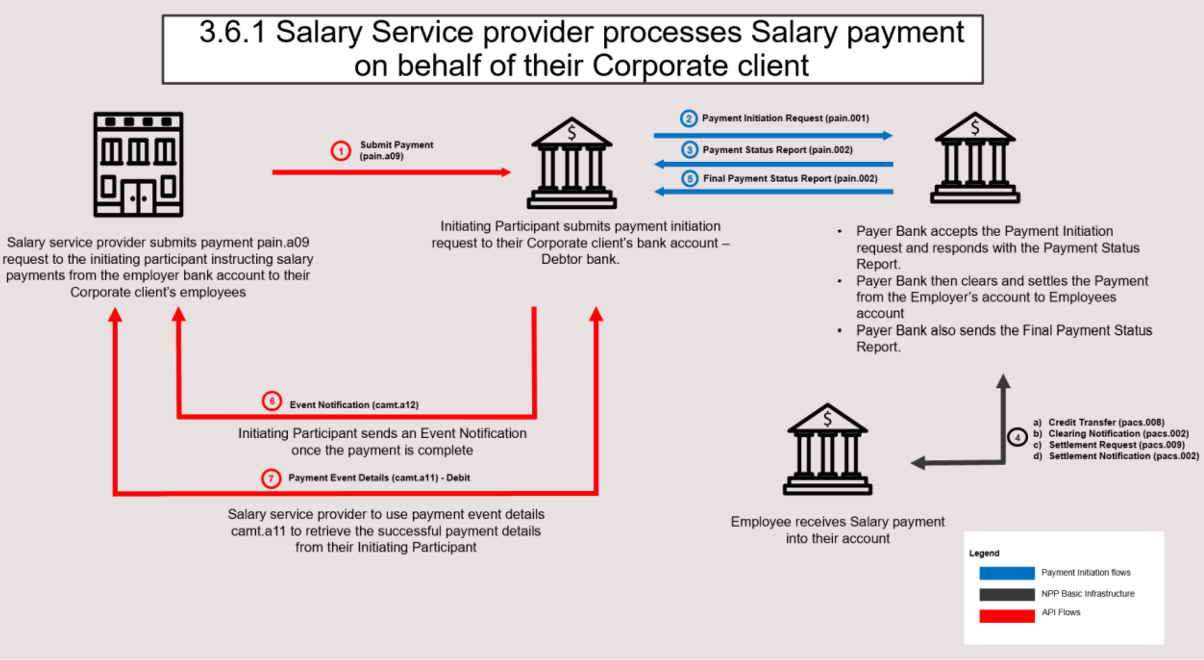
| Step | Description |
|---|---|
| 1-2 | Payee business submits a payment initiation request to its Initiating Participant also the Payee Participant. Initiating Participant initiates a check for a valid PayID before creating a payment initiation |
| 3 | Initiating Participant creates and sends a Creditor Payment Initiation Request to the Payer's NPP Participant |
| 4 | Payer's NPP Participant accepts the Creditor Payment Initiation Request and sends an Initial Payment Status Report to the Initiating Participant |
| 5 | Payer's NPP Participant then creates a clearing request with details from the Creditor Payment Initiation Request and PayID which is routed via the NPP Basic Infrastructure to the Payee's NPP Participant |
| 6 | Payee's NPP Participant accepts clearing request and responds with a clearing notification |
| 7-8 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 9 | Payer's NPP Participant sends Final Payment Status Report with the status of latest payment outcome to Initiating Participant |
| 10 | Payer's NPP Participant notifies Payer of a payment event in their account. Payee's NPP Participant notifies Payee Business of a payment event in their account. |

| Step | Description |
|---|---|
| 11 | Payee Business gets details of a credit to their account from the Payee's NPP Participant. Payer gets details of a debit from their account from the Payer's NPP Participant. |

## 3.6 Instructing a Payment

### 3.6.1 Usage Example

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for instructing a payment to be made using APIs; it is a guide only and is not meant to be prescriptive.


3.6.1 Salary Service provider processes Salary payment on behalf of their Corporate client

Salary service provider submits payment pain.a09 request to the initiating participant instructing salary payments from the employer bank account to their Corporate client's employees

Initiating Participant submits payment initiation request to their Corporate client's bank account – Debtor bank.

- Payer Bank accepts the Payment Initiation request and responds with the Payment Status Report.
- Payer Bank then clears and settles the Payment from the Employer's account to Employees account
- Payer Bank also sends the Final Payment Status Report.

Initiating Participant sends an Event Notification once the payment is complete

Salary service provider to use payment event details camt.a11 to retrieve the successful payment details from their Initiating Participant

Employee receives Salary payment into their account

1 Submit Payment (pain.a09)
2 Payment Initiation Request (pain.001)
3 Payment Status Report (pain.002)
5 Final Payment Status Report (pain.002)
4
a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
6 Event Notification (camt.a12)
7 Payment Event Details (camt.a11) - Debit
Legend
Payment Initiation flows
NPP Basic Infrastructure
API Flows

*Figure 14. Third party service processes salary payments on behalf of their client*

### 3.6.2 Instructing a Payment Flow Processing Steps

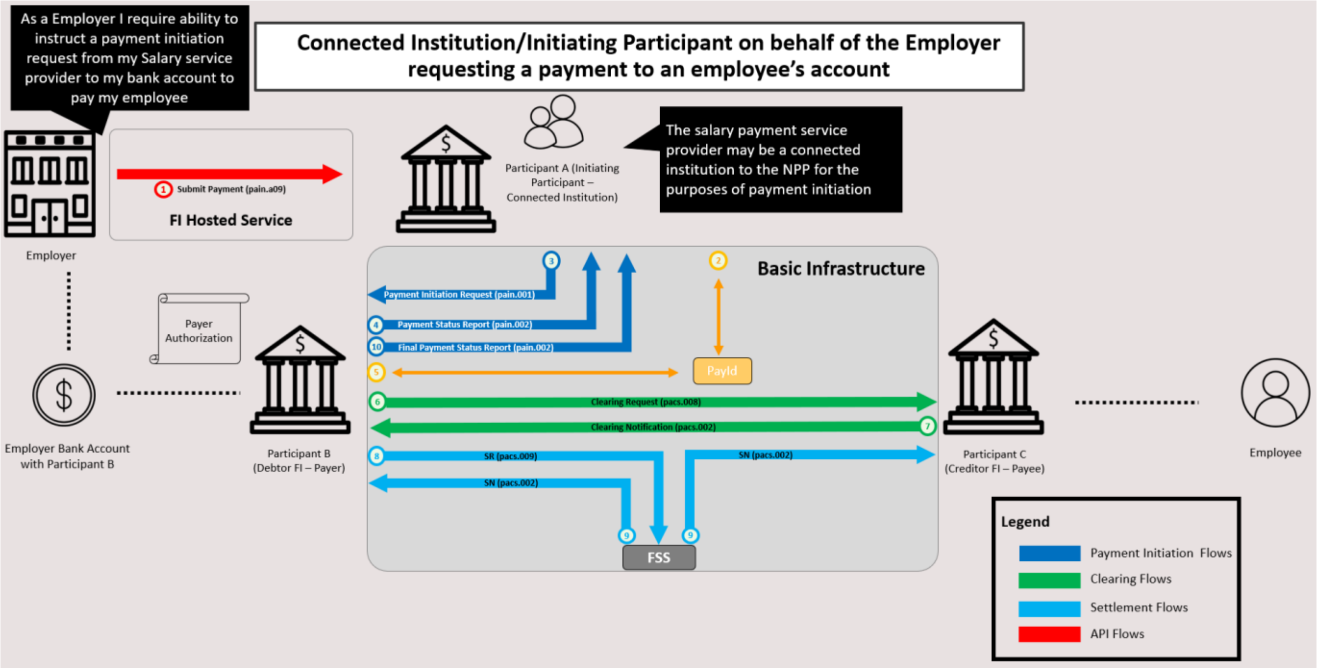

As a Employer I require ability to instruct a payment initiation request from my Salary service provider to my bank account to pay my employee
Connected Institution/Initiating Participant on behalf of the Employer requesting a payment to an employee's account
Employer
1 Submit Payment (pain.a09)
FI Hosted Service
Participant A (Initiating Participant – Connected Institution)
The salary payment service provider may be a connected institution to the NPP for the purposes of payment initiation
Basic Infrastructure
3
Payment Initiation Request (pain.001)
4
Payment Status Report (pain.002)
10
Final Payment Status Report (pain.002)
2
5
PayId
6
Clearing Request (pacs.008)
Clearing Notification (pacs.002)
7
8
SR (pacs.009)
SN (pacs.002)
SN (pacs.002)
9
9
FSS
Payer Authorization
Employer Bank Account with Participant B
Participant B (Debtor FI – Payer)
Participant C (Creditor FI – Payee)
Employee
Legend
Payment Initiation Flows
Clearing Flows
Settlement Flows
API Flows

*Figure 15. Corporate instructs third party to make a payment on their behalf*

| Step | Description |
|---|---|
| 1-2 | Salary service provider submits a payment initiation request to their Initiating Participant. Initiating Participant initiates a check for a valid PayID before creating a payment initiation |
| 3 | Initiating Participant creates and sends a Payment Initiation Request to the Payer's NPP Participant |
| 4 | Payer's NPP Participant accepts the Payment Initiation Request and sends an Initial Payment Status Report to the Initiating Participant |
| 5 | Payer's NPP Participant initiates a check for a valid PayID before creating a Clearing request |
| 6 | Payer's NPP Participant then creates a clearing request with details from the Payment Initiation Request and PayID which is routed via the NPP Basic Infrastructure to the Payee's NPP Participant |
| 7 | Payee's NPP Participant accepts clearing request and responds with a clearing notification |
| 8-9 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 10 | Payer's NPP Participant sends Final Payment Status Report with the status of latest payment outcome to Initiating Participant |

| Step | Description |
|---|---|
| 11 | Initiating Participant notifies Salary service provider of a payment event in their Corporate client’s account. |
| 12 | Salary service provider gets details of the payment from their Corporate client’s account from the Initiating Participant. |

## 3.7 Mandate Creation

### 3.7.1 Usage Example

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for instructing a Mandate to be created using APIs; it is a guide only and is not meant to be prescriptive

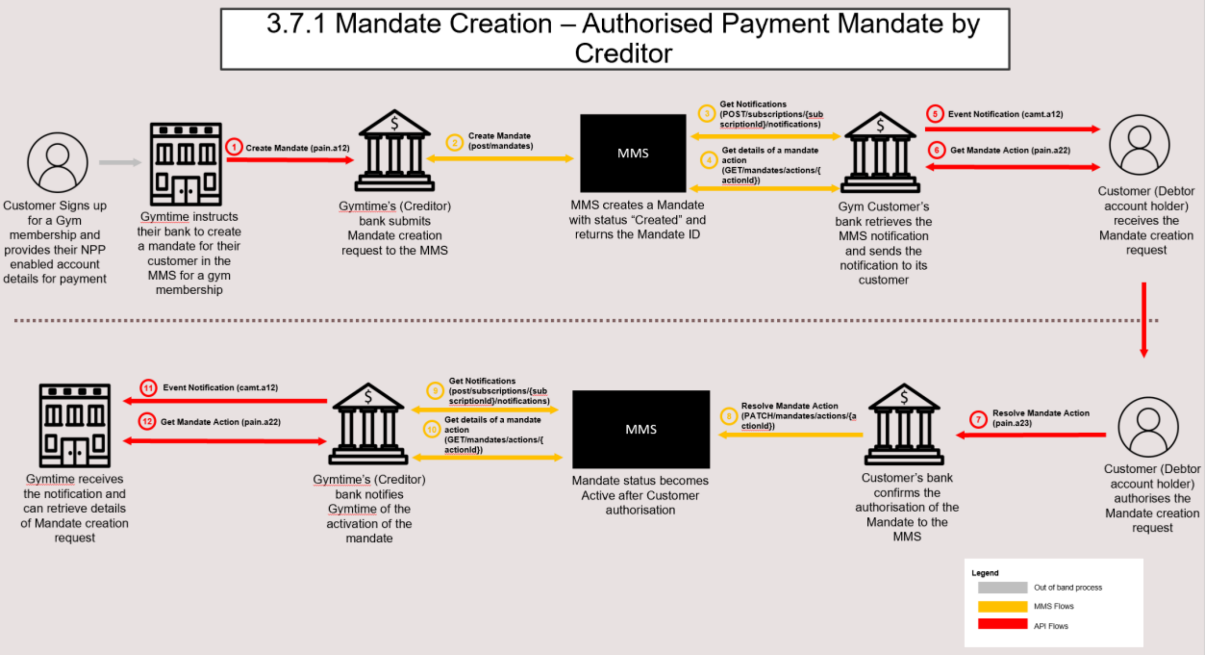

3.7.1 Mandate Creation – Authorised Payment Mandate by Creditor
Customer Signs up for a Gym membership and provides their NPP enabled account details for payment
Gymtime instructs their bank to create a mandate for their customer in the MMS for a gym membership
1 Create Mandate (pain.a12)
Gymtime's (Creditor) bank submits Mandate creation request to the MMS
2 Create Mandate (post/mandates)
MMS
MMS creates a Mandate with status “Created” and returns the Mandate ID
3 Get Notifications (POST/subscriptions/{subscriptionId}/notifications)
4 Get details of a mandate action (GET/mandates/actions/{actionId})
Gym Customer's bank retrieves the MMS notification and sends the notification to its customer
5 Event Notification (camt.a12)
6 Get Mandate Action (pain.a22)
Customer (Debtor account holder) receives the Mandate creation request
7 Resolve Mandate Action (pain.a23)
Customer (Debtor account holder) authorises the Mandate creation request
Customer's bank confirms the authorisation of the Mandate to the MMS
8 Resolve Mandate Action (PATCH/mandates/actions/{actionId})
MMS
Mandate status becomes Active after Customer authorisation
9 Get Notifications (post/subscriptions/{subscriptionId}/notifications)
10 Get details of a mandate action (GET/mandates/actions/{actionId})
Gymtime's (Creditor) bank notifies Gymtime of the activation of the mandate
11 Event Notification (camt.a12)
12 Get Mandate Action (pain.a22)
Gymtime receives the notification and can retrieve details of Mandate creation request
Legend
Out of band process
MMS Flows
API Flows

*Figure 16. Mandate creation*

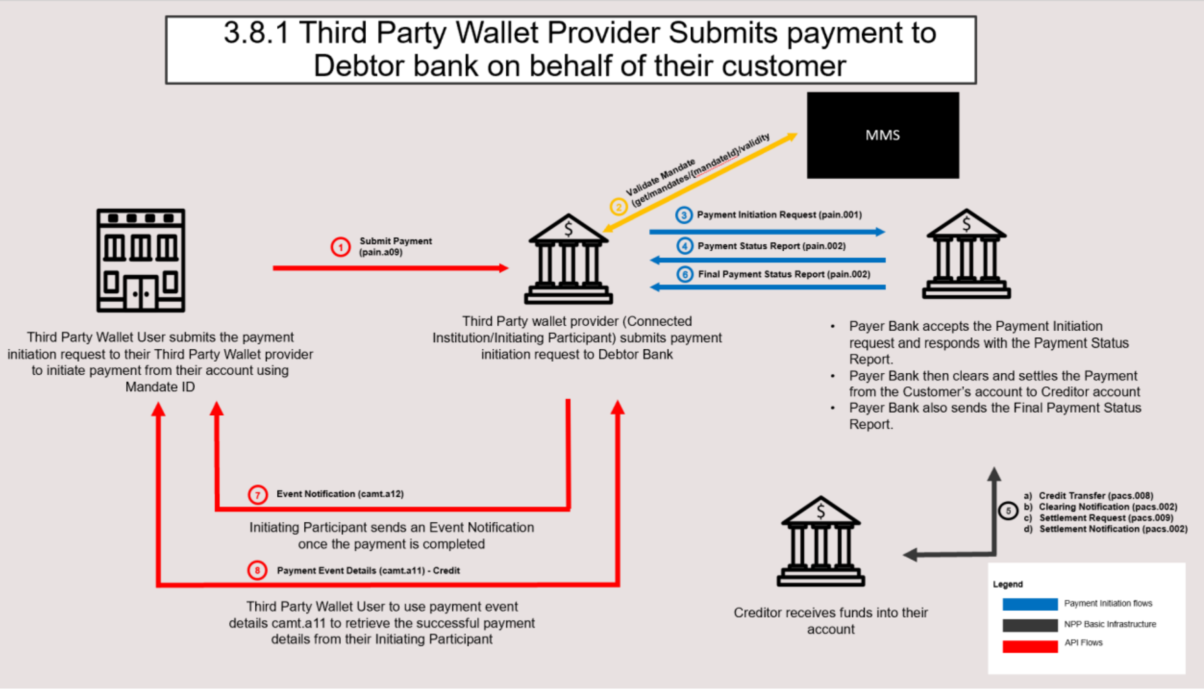
| Step | Description |
|---|---|
| 1 | Customer signs up for a Gym membership and the merchant (Creditor) instructs their bank to create a mandate for their customer in the Mandate Management Service (MMS) |
| 2 | Creditor bank submits Mandate creation request to the MMS |
| 3 | MMS creates a mandate with status ’Created’ and returns the Mandate ID to the Merchant |
| 4 | Customer’s bank (Debtor bank) retrieves the MMS notification and sends the event notification to the customer |

| | |
|---|---|
| 5-6 | Customer retrieves the mandate creation request pending for authorisation and authorises the mandate |
| 7 | Debtor bank confirms Customer authorisation of the mandate to the MMS |
| 8 | MMS updates the mandate status from ‘Created’ to ‘Active’ |
| 9 | Creditor bank retrieves the MMS notification confirming the activation of the mandate and sends an event notification to the Merchant |
| 10 | Merchant retrieves the details of the authorised mandate creation request from the Creditor bank |

## 3.8 Instructing a Mandate Payment – Initiating Participant is not the Creditor Bank

### 3.8.1 Usage Example

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for instructing a payment to be made using APIs; it is a guide only and is not meant to be prescriptive.



3.8.1 Third Party Wallet Provider Submits payment to Debtor bank on behalf of their customer

Third Party Wallet User submits the payment initiation request to their Third Party Wallet provider to initiate payment from their account using Mandate ID

Third Party wallet provider (Connected Institution/Initiating Participant) submits payment initiation request to Debtor Bank

- Payer Bank accepts the Payment Initiation request and responds with the Payment Status Report.
- Payer Bank then clears and settles the Payment from the Customer's account to Creditor account
- Payer Bank also sends the Final Payment Status Report.

Initiating Participant sends an Event Notification once the payment is completed

Third Party Wallet User to use payment event details camt.a11 to retrieve the successful payment details from their Initiating Participant

Creditor receives funds into their account

MMS
1 Submit Payment (pain.a09)
2 Validate Mandate (get/mandates/{mandateId}/validity
3 Payment Initiation Request (pain.001)
4 Payment Status Report (pain.002)
6 Final Payment Status Report (pain.002)
5 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
7 Event Notification (camt.a12)
8 Payment Event Details (camt.a11) - Credit
Legend
Payment Initiation flows
NPP Basic Infrastructure
API Flows

*Figure 17. Third party wallet provider submits payment on behalf of customer*

### 3.8.2 Instructing a Mandate Payment Flow Processing Steps (On behalf of the Debtor)

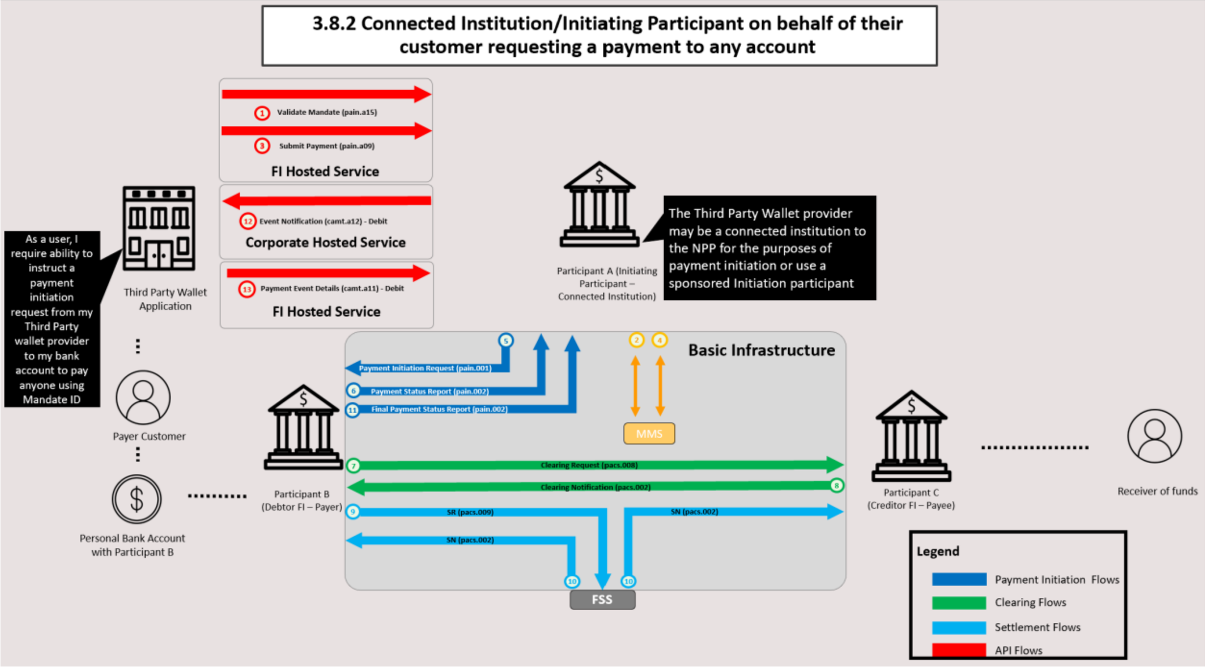
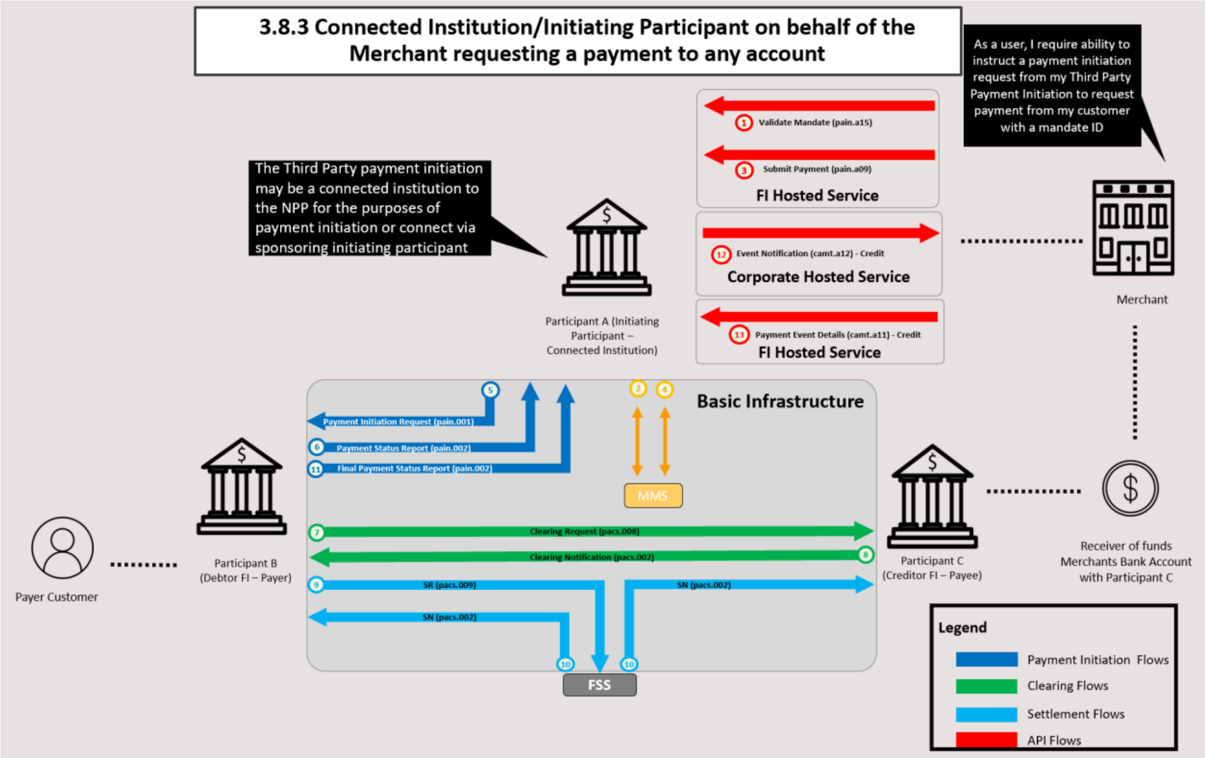

3.8.2 Connected Institution/Initiating Participant on behalf of their customer requesting a payment to any account
1 Validate Mandate (pain.a15)
3 Submit Payment (pain.a09)
FI Hosted Service
12 Event Notification (camt.a12) - Debit
Corporate Hosted Service
13 Payment Event Details (camt.a11) - Debit
FI Hosted Service
As a user, I require ability to instruct a payment initiation request from my Third Party wallet provider to my bank account to pay anyone using Mandate ID
Third Party Wallet Application
Payer Customer
Personal Bank Account with Participant B
Participant B (Debtor FI – Payer)
Participant A (Initiating Participant – Connected Institution)
The Third Party Wallet provider may be a connected institution to the NPP for the purposes of payment initiation or use a sponsored Initiation participant
Basic Infrastructure
5 Payment Initiation Request (pain.001)
6 Payment Status Report (pain.002)
11 Final Payment Status Report (pain.002)
2
4
MMS
7 Clearing Request (pacs.008)
8 Clearing Notification (pacs.002)
9 SR (pacs.009)
SN (pacs.002)
SN (pacs.002)
10
10
FSS
Participant C (Creditor FI – Payee)
Receiver of funds
Legend
Payment Initiation Flows
Clearing Flows
Settlement Flows
API Flows

*Figure 18. Initiating a mandate payment*

| Step | Description |
|---|---|
| 1-2 | Third Party Wallet application user instructs a payment initiation request and the Third Party Wallet application checks the validity of the mandate already established with their user in the MMS |
| 3 | After successful validation of the Mandate, Third Party Wallet application submits a payment initiation request to their Third Party wallet provider (Connected Institution/Initiating Participant) to initiate payment from their account using the Mandate ID |
| 4-5 | Third party wallet provider validates the Mandate and submits a payment initiation request to the Payer’s NPP Participant (Debtor bank). |
| 6 | Debtor Bank accepts the Payment Initiation Request and sends an Initial Payment Status Report to the Initiating Participant |
| 7 | Debtor Bank then creates a clearing request with details from the Payment Initiation Request which is routed via the NPP Basic Infrastructure to the Payee’s NPP Participant (Creditor Bank) |
| 8 | Creditor Bank accepts clearing request and responds with a clearing notification |
| 9-10 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 11 | Debtor Bank sends Final Payment Status Report with the status of latest payment outcome to Initiating Participant |

|  |  |
|---|---|
| 12 | Initiating Participant notifies Third Party Wallet user of a payment event from the user's account |
| 13 | Third Party Wallet user can retrieve details of the payment event from the Initiating Participant |

### 3.8.3 Instructing a Mandate Payment Flow Processing Steps (On behalf of the Creditor)



3.8.3 Connected Institution/Initiating Participant on behalf of the Merchant requesting a payment to any account

As a user, I require ability to instruct a payment initiation request from my Third Party Payment Initiation to request payment from my customer with a mandate ID

The Third Party payment initiation may be a connected institution to the NPP for the purposes of payment initiation or connect via sponsoring initiating participant

1 Validate Mandate (pain.a15)
3 Submit Payment (pain.a09)
FI Hosted Service
12 Event Notification (camt.a12) - Credit
Corporate Hosted Service
13 Payment Event Details (camt.a11) - Credit
FI Hosted Service
Merchant
Participant A (Initiating Participant – Connected Institution)
Basic Infrastructure
5
Payment Initiation Request (pain.001)
6 Payment Status Report (pain.002)
11 Final Payment Status Report (pain.002)
2
4
MMS
7 Clearing Request (pacs.008)
8 Clearing Notification (pacs.002)
9 SR (pacs.009)
SN (pacs.002)
SN (pacs.002)
10
10
FSS
Payer Customer
Participant B (Debtor FI – Payer)
Participant C (Creditor FI – Payee)
Receiver of funds Merchants Bank Account with Participant C
Legend
Payment Initiation Flows
Clearing Flows
Settlement Flows
API Flows

*Figure 19. Initiating a mandate payment on behalf of a creditor*

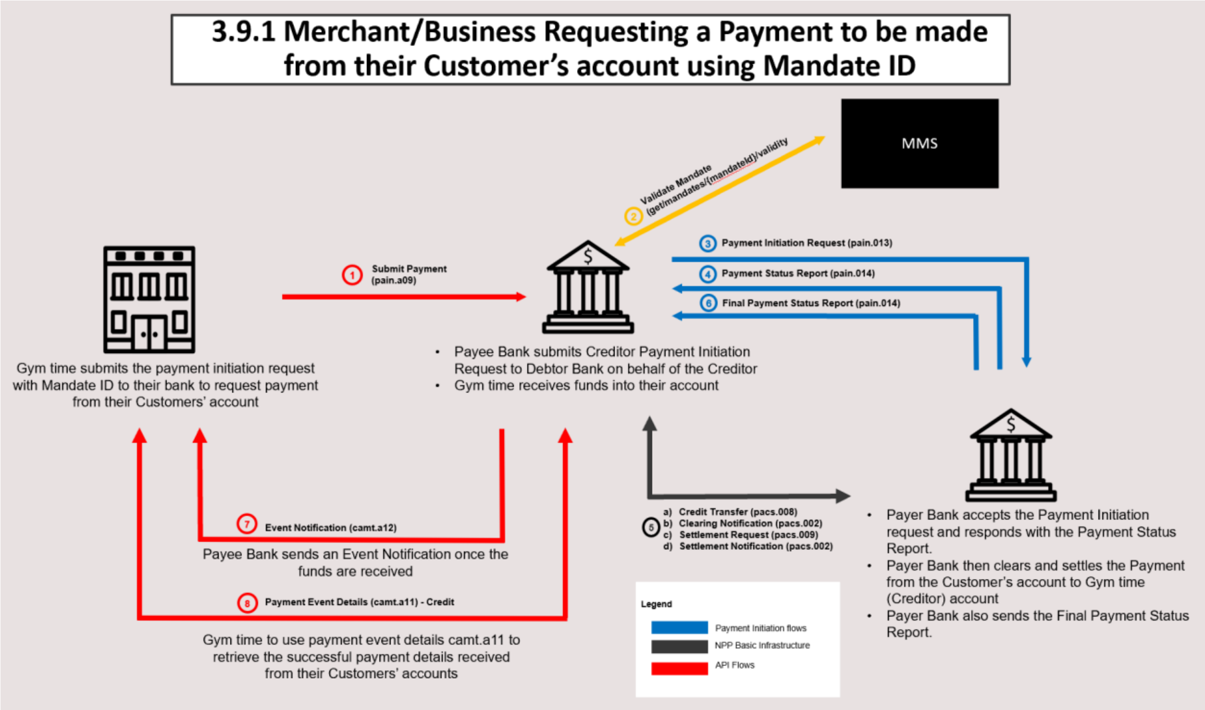
| Step | Description |
|---|---|
| 1-2 | Merchant checks the validity of the mandate already established with their customer in the MMS |
| 3 | After successful validation of the Mandate, Merchant submits a payment initiation request to a Third Party payment service provider (Connected Institution/Initiating Participant) to initiate payment from their Customer's account (Debtor) using the Mandate ID into the Merchant's account (Creditor) |
| 4-5 | Third party payment service provider validates the Mandate and submits a payment initiation request to the Payer customer's NPP Participant (Debtor bank). |
| 6 | Debtor Bank accepts the Payment Initiation Request and sends an Initial Payment Status Report to the Initiating Participant |
| 7 | Debtor Bank then creates a clearing request with details from the Payment Initiation Request which is routed via the NPP Basic Infrastructure to the Payee's NPP Participant (Creditor Bank) |

| | |
|---|---|
| 8 | Creditor Bank accepts clearing request and responds with a clearing notification |
| 9-10 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 11 | Debtor Bank sends Final Payment Status Report with the status of latest payment outcome to Initiating Participant |
| 12 | Initiating Participant notifies the merchant of a payment event into the Merchant’s account |
| 13 | Merchant retrieves details of the payment event from the Initiating Participant |

## 3.9 Instructing a Mandate Payment – Initiating Participant is the Creditor Bank

### 3.9.1 Usage Example

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for instructing a payment to be made using APIs; it is a guide only and is not meant to be prescriptive.


3.9.1 Merchant/Business Requesting a Payment to be made from their Customer’s account using Mandate ID

Gym time submits the payment initiation request with Mandate ID to their bank to request payment from their Customers’ account

- Payee Bank submits Creditor Payment Initiation Request to Debtor Bank on behalf of the Creditor
- Gym time receives funds into their account

- Payer Bank accepts the Payment Initiation request and responds with the Payment Status Report.
- Payer Bank then clears and settles the Payment from the Customer’s account to Gym time (Creditor) account
- Payer Bank also sends the Final Payment Status Report.

Payee Bank sends an Event Notification once the funds are received

Gym time to use payment event details camt.a11 to retrieve the successful payment details received from their Customers’ accounts

MMS
2 Validate Mandate (get/mandates/{mandateId}/validity
1 Submit Payment (pain.a09)
3 Payment Initiation Request (pain.013)
4 Payment Status Report (pain.014)
6 Final Payment Status Report (pain.014)
5 a) Credit Transfer (pacs.008)
b) Clearing Notification (pacs.002)
c) Settlement Request (pacs.009)
d) Settlement Notification (pacs.002)
7 Event Notification (camt.a12)
8 Payment Event Details (camt.a11) - Credit
Legend
Payment Initiation flows
NPP Basic Infrastructure
API Flows

*Figure 20. Organisation requesting a payment to be made from their customer's account using a mandate ID*

### 3.9.2 Instructing a Mandate Payment Flow Processing Steps

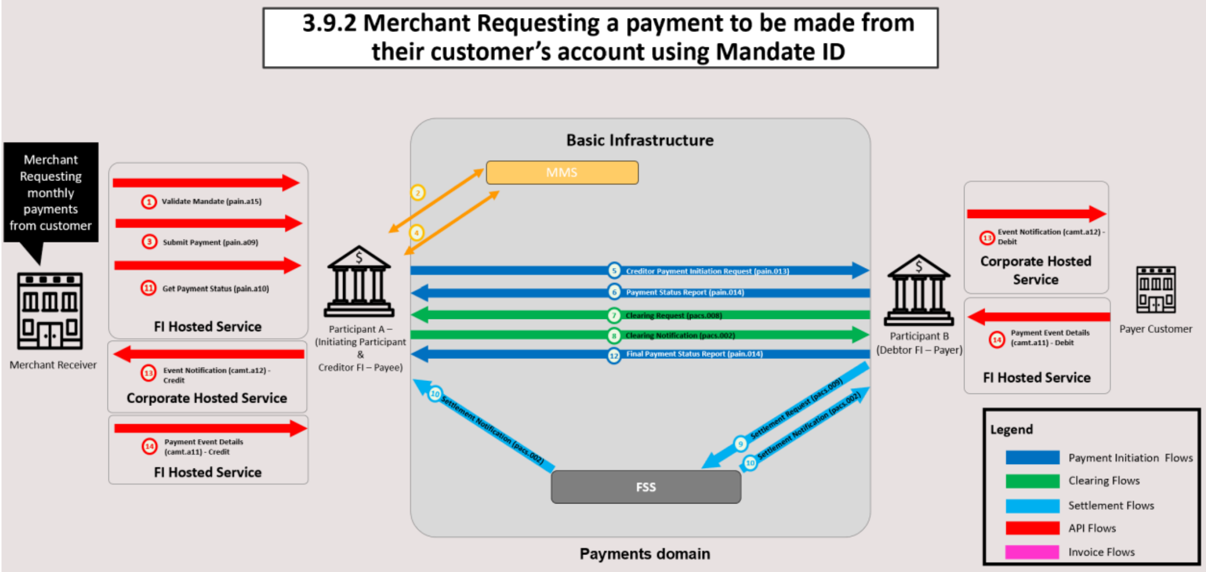

3.9.2 Merchant Requesting a payment to be made from their customer's account using Mandate ID
Merchant Requesting monthly payments from customer
Merchant Receiver
1 Validate Mandate (pain.a15)
3 Submit Payment (pain.a09)
11 Get Payment Status (pain.a10)
FI Hosted Service
13 Event Notification (camt.a12) - Credit
Corporate Hosted Service
14 Payment Event Details (camt.a11) - Credit
FI Hosted Service
Participant A – (Initiating Participant & Creditor FI – Payee)
Basic Infrastructure
MMS
2
4
5 Creditor Payment Initiation Request (pain.013)
6 Payment Status Report (pain.014)
7 Clearing Request (pacs.008)
8 Clearing Notification (pacs.002)
12 Final Payment Status Report (pain.014)
10 Settlement Notification (pacs.002)
9 Settlement Request (pacs.009)
10 Settlement Notification (pacs.002)
FSS
Payments domain
Participant B (Debtor FI – Payer)
13 Event Notification (camt.a12) - Debit
Corporate Hosted Service
14 Payment Event Details (camt.a11) - Debit
FI Hosted Service
Payer Customer
Legend
Payment Initiation Flows
Clearing Flows
Settlement Flows
API Flows
Invoice Flows

*Figure 21. Merchant requests a payment to be made from their customer's account using the mandate ID*

| Step | Description |
|---|---|
| 1-2 | Merchant checks the validity of the mandate already established with their customer in the MMS |
| 3 | After successful validation of the Mandate, Merchant submits a payment initiation request to the Creditor Bank (Initiating Participant) to initiate payment from their Customer's account (Debtor) using the Mandate ID into the Merchant's account (Creditor) |
| 4-5 | Creditor Bank validates the Mandate and submits a payment initiation request to the Payer customer's NPP Participant (Debtor bank). |
| 6 | Debtor Bank accepts the Payment Initiation Request and sends an Initial Payment Status Report to the Initiating Participant |
| 7 | Debtor Bank then creates a clearing request with details from the Payment Initiation Request which is routed via the NPP Basic Infrastructure to the Creditor Bank |
| 8 | Creditor Bank accepts clearing request and responds with a clearing notification |
| 9-10 | NPP Basic Infrastructure and FSS complete settlement processing and sends confirmations to both NPP Participants |
| 11 | Merchant can optionally check the status of the final payment |
| 12 | Debtor Bank sends Final Payment Status Report with the status of latest payment outcome to Initiating Participant |
| 13 | Initiating Participant/Creditor Bank notifies the merchant of a payment event into the Merchant's account |

| | |
|---|---|
| 14 | Merchant retrieves details of the payment event from the Initiating Participant |

## 3.10 Mandate Amendment

### 3.10.1 Usage Example - Mandate Amendment – Unilateral

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for Unilateral Mandate Amendment to be made using APIs; it is a guide only and is not meant to be prescriptive.

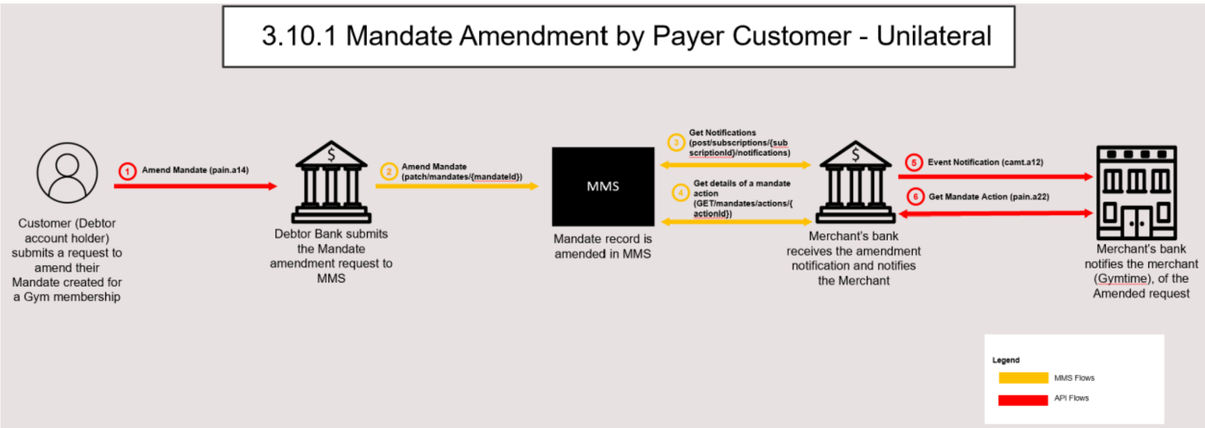
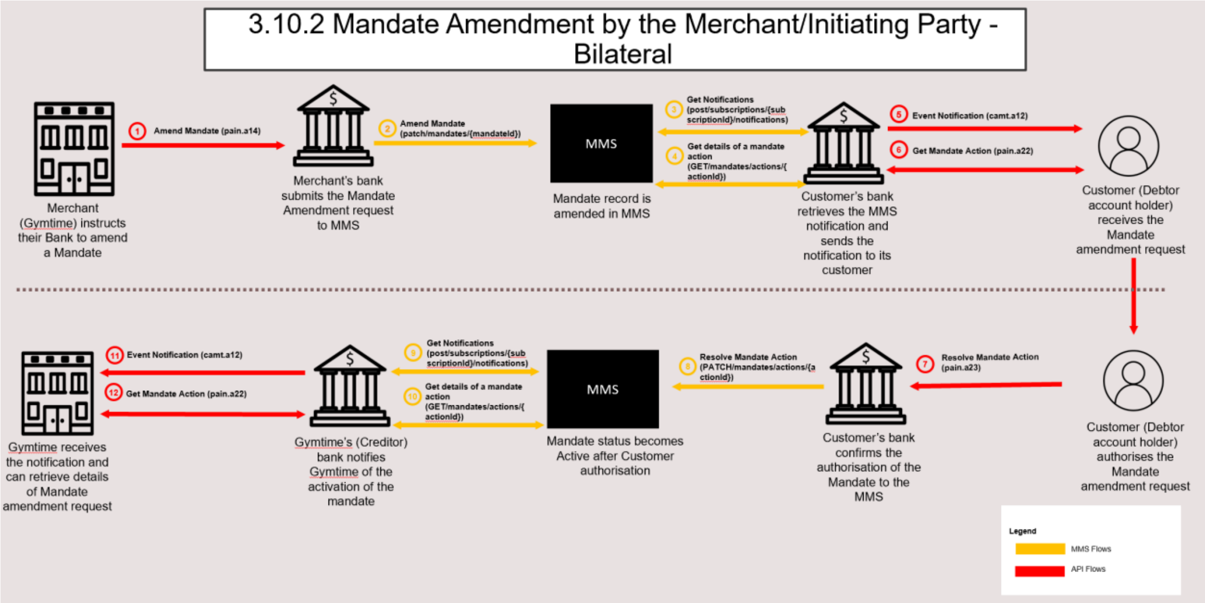

3.10.1 Mandate Amendment by Payer Customer - Unilateral
1 Amend Mandate (pain.a14)
2 Amend Mandate (patch/mandates/{mandateId})
3 Get Notifications (post/subscriptions/{subscriptionId}/notifications)
4 Get details of a mandate action (GET/mandates/actions/{actionId})
5 Event Notification (camt.a12)
6 Get Mandate Action (pain.a22)
MMS
Customer (Debtor account holder) submits a request to amend their Mandate created for a Gym membership
Debtor Bank submits the Mandate amendment request to MMS
Mandate record is amended in MMS
Merchant's bank receives the amendment notification and notifies the Merchant
Merchant's bank notifies the merchant (Gymtime), of the Amended request
Legend
MMS Flows
API Flows

*Figure 22. Mandate amendment*

| Step | Description |
|---|---|
| 1 | Customer submits a request to their Bank to amend a Mandate that has already been created |
| 2 | Debtor Bank submits the Mandate amendment request to the MMS and MMS updates the Mandate record as per the request submitted by the Debtor Bank |
| 3-4 | Merchant's bank (Creditor) receives the amendment notification and sends an event notification to the Merchant |
| 5 | Merchant retrieves the details of the amendment made to the Mandate record from the Creditor bank |

### 3.10.2 Usage Example - Mandate Amendment – Bilateral

The sequence of processing steps in this section is provided as an illustrative example of a possible flow for Bilateral Mandate Amendment to be made using APIs; it is a guide only and is not meant to be prescriptive.


3.10.2 Mandate Amendment by the Merchant/Initiating Party - Bilateral

Merchant (Gymtime) instructs their Bank to amend a Mandate
1 Amend Mandate (pain.a14)
Merchant's bank submits the Mandate Amendment request to MMS
2 Amend Mandate (patch/mandates/{mandateId})
MMS
Mandate record is amended in MMS
3 Get Notifications (post/subscriptions/{sub scriptionId}/notifications)
4 Get details of a mandate action (GET/mandates/actions/{ actionId})
Customer's bank retrieves the MMS notification and sends the notification to its customer
5 Event Notification (camt.a12)
6 Get Mandate Action (pain.a22)
Customer (Debtor account holder) receives the Mandate amendment request
Customer (Debtor account holder) authorises the Mandate amendment request
7 Resolve Mandate Action (pain.a23)
Customer's bank confirms the authorisation of the Mandate to the MMS
8 Resolve Mandate Action (PATCH/mandates/actions/{a ctionId})
MMS
Mandate status becomes Active after Customer authorisation
9 Get Notifications (post/subscriptions/{sub scriptionId}/notifications)
10 Get details of a mandate action (GET/mandates/actions/{ actionId})
Gymtime's (Creditor) bank notifies Gymtime of the activation of the mandate
11 Event Notification (camt.a12)
12 Get Mandate Action (pain.a22)
Gymtime receives the notification and can retrieve details of Mandate amendment request
Legend
MMS Flows
API Flows

*Figure 23. Mandate amendment by the merchant*

| Step | Description |
|---|---|
| 1 | Merchant (Creditor) instructs their Bank to amend a Mandate that has been created with their customer in the MMS |
| 2 | Merchant's bank (Creditor bank) submits a Mandate amendment request to the MMS |
| 3-4 | Customer's bank (Debtor bank) retrieves the MMS notification and sends the event notification to the customer |
| 5-6 | Customer retrieves the mandate amendment request pending for authorisation and authorises the request |
| 7 | Debtor bank confirms Customer authorisation of the mandate amendment to the MMS and MMS updates the mandate status to 'Active' |
| 8-9 | Creditor bank retrieves the MMS notification confirming activation of the mandate and sends an event notification to the Merchant |
| 10 | Merchant retrieves the details of the authorised mandate amendment request from the Creditor bank |

## 3.11 Sample API Documentation

The associated zip file "NPP API Framework Sample Schema V5" contains the following sample APIs

- **API: pain.a09.001.05 (Submit Payment)**
  pain.a09.001.05.pdf
  pain.a09.001.05.xls
  pain.a09.001.05.schema.json

- **API: pain.a10.001.03 (Get Payment Status**)

pain.a10.001.03.pdf
pain.a10.001.03.xls
pain.a10.001.03.schema.json

- **API: pain.a11.001.04 (Get Account Servicer by Alias)**
  pain.a11.001.04.pdf
  pain.a11.001.04.xls
  pain.a11.001.04.schema.json

- **API: camt.a09.001.02 (Cancel Payment)**
  camt.a09.001.02.pdf
  camt.a09.001.02.xls
  camt.a09.001.02.schema.json

- **API: camt.a10.001.02 (Return Payment**)
  camt.a10.001.02.pdf
  camt.a10.001.02.xls
  camt.a10.001.02.schema.json

- **API: camt.a11.001.03 (Payment Event Details)**
  camt.a11.001.03.pdf
  camt.a11.001.03.xls
  camt.a11.001.03.schema.json

- **API: camt.a12.001.03 (Event Notification)**
  camt.a12.001.03.pdf
  camt.a12.001.03.xls
  camt.a12.001.03.schema.json

- **API: pain.a46.001.03 (Batch Payment Initiation)**
  pain.a46.001.03.pdf
  pain.a46.001.03.xls
  pain.a46.001.03.schema.json

- **API: pain.a47.001.01 (Batch Payment Status**)
  pain.a47.001.01.pdf
  pain.a47.001.01.xls
  pain.a47.001.01.schema.json

- **API: camt.a13.001.03 (Batch Payment Event Details**)
  camt.a13.001.03.pdf
  camt.a13.001.03.xls
  camt.a13.001.03.schema.json

- **API: pain.a48.001.02 (Get Cancellation Status**)
  pain.a48.001.01.pdf
  pain.a48.001.01.xls
  pain.a48.001.01.schema.json

- **API: pain.a12.001.01 (Create Mandate)**
  pain.a12.001.01.pdf
  pain.a12.001.01.xls
  pain.a12.001.01.schema.json

- **API: pain.a13.001.01 (Get Mandate By ID)**
  pain.a13.001.01.pdf
  pain.a13.001.01.xls
  pain.a13.001.01.schema.json

- **API: pain.a14.001.01 (Amend Mandate)**
  pain.a14.001.01.pdf
  pain.a14.001.01.xls
  pain.a14.001.01.schema.json

- **API: pain.a15.001.01 (Validate Mandate)**
  pain.a15.001.01.pdf
  pain.a15.001.01.xls
  pain.a15.001.01.schema.json

- **API: pain.a16.001.01 (Get Mandate Status)**
  pain.a16.001.01.pdf
  pain.a16.001.01.xls
  pain.a16.001.01.schema.json

- **API: pain.a17.001.01 (Change Mandate Status)**
  pain.a17.001.01.pdf
  pain.a17.001.01.xls
  pain.a17.001.01.schema.json

- **API: pain.a22.001.01 (Get Mandate Action)**
  pain.a22.001.01.pdf
  pain.a22.001.01.xls
  pain.a22.001.01.schema.json

- **API: pain.a23.001.01 (Resolve Mandate Action)**
  pain.a23.001.01.pdf
  pain.a23.001.01.xls
  pain.a23.001.01.schema.json

- **API: pain.a36.001.01 (Mandate Enquiry)**
  pain.a36.001.01.pdf
  pain.a36.001.01.xls
  pain.a36.001.01.schema.json