Central Bank of The United Arab Emirates

UAEIPP Overlay Service

Interface Specifications - Front End APIs for Merchants

Document Code:

UAEIPP-PLT-Overlay Services-Interface Specifications-FrontEnd APIs for merchants payments

Version October 2024

CONFIDENTIAL

Control

Document Sign-Off Sheet

Document Sign-Off Sheet				
Name	Position	Comments	Signature	Date

Document History

Author	Date	Version	Reason for new Version
Central Bank of The United Arab Emirates	June 2022	1.0	First version
Central Bank of The United Arab Emirates	August 2022	1.1	Updated Error codes Updated definition in swagger (.zip files)
Central Bank of The United Arab Emirates	October 2022	1.2	Updated Error codes Updated definition in swagger (.zip files)
Central Bank of The United Arab Emirates	February 2023	1.3	• registerQrCodeChannel, amended documentation • swagger split in section 2.15 as per sr1046 • Removed "01252 QR CODE NOT FOUND" • Checked a typo: "129 RECIPIENT IBAN DOES NO BELONG TO THE BANK" • Clarified the following Errors: ○ 01255 THE REVERSAL IS NOT COHERENT WITH THE PAYMENT Clarification to be added in the doc: Mismatch of data between reversal and original payment ○ 01258 TRANSACTION ID NOT COHERENT Clarification to be added in the doc: Mismatch of transaction ID
Central Bank of The United Arab Emirates	March 2023	1.4	Added the idsct in the following APIs: • checkStatusQrCodeChannel • checkOnlineRequestToPayStatus Clarification added on the EMV Qr Code
Central Bank of The United Arab Emirates	April 2023	1.4.1	Clarification added on the Qr Code
Central Bank of The United Arab Emirates	May 2023	1.5	Release 2.1 – Acquirers updates • Added new common fields in the request header documented at the paragraph 2.4.1 to manage the use of the APIs by technical provider and technical provider as acquirer actors • Added new error codes to cover bad scenarios which involve technical provider and technical provider as acquirer • Added new paragraph 3.15.9 in the Appendix on technical provider and technical provider as acquirer

Author	Date	Version	Reason for new Version
Central Bank of The United Arab Emirates	May 2023	1.5.1	• Common field: added a note for the Not Mandatory fields linked to the Provider/Acquirers • Added error code "01182 – The APP-ID is not valid on Refund API" • Clarifications on Reversal
Central Bank of The United Arab Emirates	July 2023	1.5.2	• Amended path of the API checkOnlineReversalStatus with correct microservice (inquiry-payment) • Added a "when" in the pre-authorisation to clarify the polling flow • Clarification on paymentid – Delete QrCode
Central Bank of The United Arab Emirates	Aug 2023	1.5.3	• CR57 - Added appId field in the request header of the following APIs (even in swaggers "business-payment-ms QrCode 1.1" and "business-payment-ms RTP 1.2"): ○ verifyReversal ○ confirmReversal ○ Refund ○ verifyOnlineBusinessReversal ○ confirmOnlineBusinessReversal ○ Refund (Online) • Added clarifications to the EMV Static QR Code String structure – Consumer and Merchant scenario; • Typo in 2.1.6.1 sequence diagram description • Clarification on Acquirer/provider • Updated error code 01076 description • Add clarification on the refund for Provider/Acquirers • Replaced post with put in confirmBusinessReversalAPI • Clarification on link - GenerateQrCode
Central Bank of The United Arab Emirates	Oct 2023	1.5.4	• Typo in the description of error code 03032 and aligned to the specs for Status and finalizePaymentChannel • Removed error code 03027 from finalise payment channel + check reversal status • Added clarification on ID "26" Dynamic QR Code Structure and on the paymentcategory • Added ID "08", sub-field of ID "62" to contain paymentCategory. FIxed typo in Merchant City
Central Bank of The United Arab Emirates	Dec 2023	1.5.5	• Clarification on payment object – Detele Qr Code • Updated cashDeshId and shopId as mandatory in Register QR Code API (updated even in swagger business-payment-ms QrCode 1.2) to be aligned with the BE API • Added new example for EMV standard Dynamic QrCode • CR 51 – Updated expected values in impacted APIs for the field proxy.type. Changes implemented for CR 53 (2024 R2). Added the following fields for the APIs: refund; verifyBusinessReversalCHANNEL;

Author	Date	Version	Reason for new Version
			confirmBusinessReversalCHANNEL; verifyOnlineBusinessReversal; confirmOnlineBusinessReversal • deviceOSVersion. • deviceModel • deviceId • deviceIpAddress • country • timeZone
Central Bank of The United Arab Emirates	Apr 2024	1.5.6	• CR36 Added the error code "The field {0} doesn't match any of the expected values" (available from 2024 R4) for the APIs registerQrCode, refund, sendOnlineRequestToPay • CR53 – 2024 R2: o Added the error codes "Country not authorised" and "Country information mandatory for the payment" for the API refund, verifyBusinessReversal, confirmBusinessReversal, verifyOnlineBusinessReversal, confirmOnlineBusinessReversal, o Replaced TBD with the codes 432 and 433 for the errors "Country not authorised" and "Country information mandatory for the payment" • Added clarification on RTP flow after 2024 R3 (when PGS will be released)
Central Bank of The United Arab Emirates	May 2024	1.5.7	• NN-1969 Added clarification about Merchants temporarily blacklisted after three customer rejections within 10 minutes to prevent spam for sendOnlineRequestToPay API • Added disclaimer in Endpoint section
Central Bank of The United Arab Emirates	June 2024	1.5.8	• Added error code 01383 Creditor not found to Send Online RTP and Refund • Aligned Refund response response to the swagger • Added in API Verify Reversal the fields: •Payment.idSct •Payment.mobile •consentId •receiverNominative •receiverIban •senderIban
Central Bank of The United Arab Emirates	August 2024	1.5.9	• CR_119 – R3 Added possibility to send a RTP from Physical Shop and generate a Qr Code from Online Shop. Removed error code 01328 from SendOnlineRTP • Aligned description 01076 "Bank Account not found"

Author	Date	Version	Reason for new Version

Central Bank of The United Arab Emirates	October 2024	1.5.10	• Fix on the type of the following fields in all APIs:
			  • payment.paymentRefId
			  • payment.shopId
			  • paymentId
			  • cashDeskId
			• Added error code 01130 in Send Online Request To Pay.
			• Modified description 01182

TABLE OF CONTENT

INDEX OF FIGURES

GENERAL INFORMATION

Definitions

Acronym/Term	Text explaining the definition
API	Application Programming Interface
APP	Application (mobile)
B2P	Business To Person (refund)
HTTP	Hyper Text Transfer Protocol
IBAN	International Bank Account Number
JWT	Json Web Token
JWS	Json Web Signature
OAUTH	Open Authorization
P2B	Person To Business
P2P	Person To Person
PSP	Payment Service Providers
RTP/R2P	Request To Pay
REST	Representational State Transfer
OAUTH	Open Authorization

Standards

Standard	Description	Link or Document
EMVCo	Merchant Presented QR Code Specification Standard	QR Code Specification for Payment Systems (EMV QRCPS)
ISO 4217	Codes for the representation of currencies and funds	https://www.iso.org/iso-4217-currency-codes.html
ISO 3166	Codes for the representation of currencies	https://www.iso.org/obp/ui/#search

EXECUTIVE SUMMARY

The aim of this document is to describe the services (APIs) that the UAEIPP Overlay Service system makes available to Participant Participants' Clients through the API Gateway. These APIs allow the Participants to integrate UAEIPP Overlay Service functionalities in their own systems (can be Participants' Channels or Payment Gateway).

The functionalities offered by these APIs concern:

1. Merchant services
2. Request to Pay Services P2B (merchants' side)
3. Request to Pay Services Qr Code P2B
4. Transaction Enquiry services

1. INTRODUCTION

The solution has been designed as a multi-layer application architecture. The Mobile Server and the API Gateway act as network front-end (Network Layer), the Engine and the Gateway, acting as back-end.

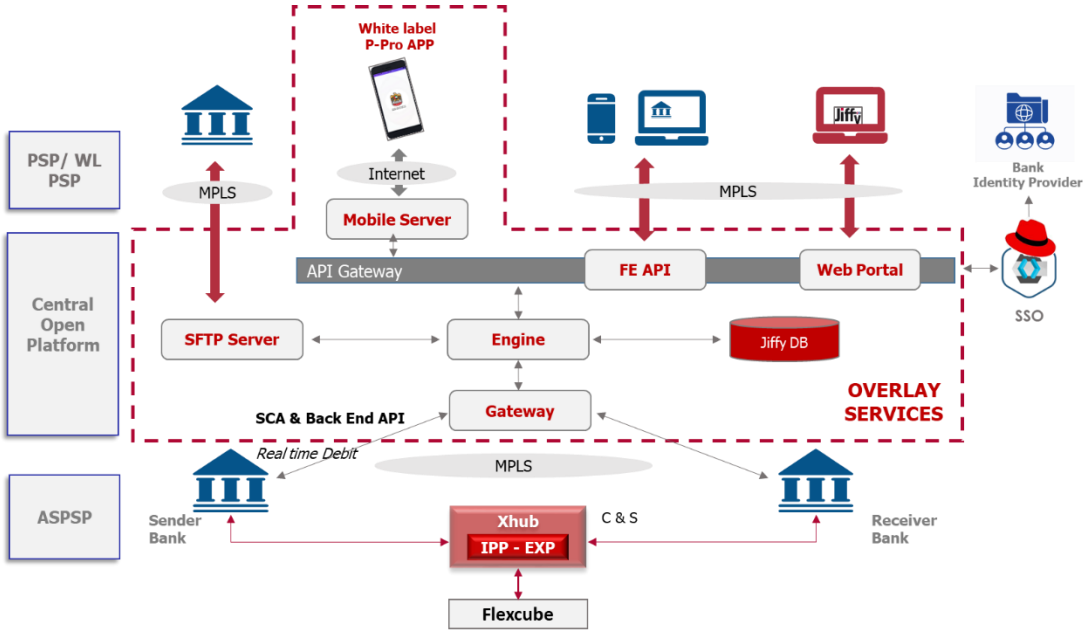


Figure 1 High-level Architecture

The IPP core platform is represented by UAEIPP Core solution (for details, kindly refer to UAEIPP Core Service Participant Functional Specification).

The White Label APP is the component that can be used by Participants' customers to execute the operations provided by the service.

The Mobile Server behaves as a Presentation Layer for the White Label App. In other words, it communicates with UAEIPP Overlay Service Engine, and it provides to the White Label App all the information and data necessary to be displayed on a specific screen

The API Gateway exposes Engine's RESTful JSON API manager to Participants or merchants, who can retrieve the necessary APIs to integrate the solution through their own payment gateways.

The Engine represents the core Platform and exposes customer management functions and financial operation functions through a workflow manager. Proxy Database is directly connected to the Engine, containing all the information about customers and transactions.

The Gateway connects the Engine to Participants' financial services (Back End APIs exposed by Participants).

UAEIPP Overlay Service is based on APIs (not ISO20022 messages) created following the guidelines of ISO20022 White Paper.

1.1. Integration Layers

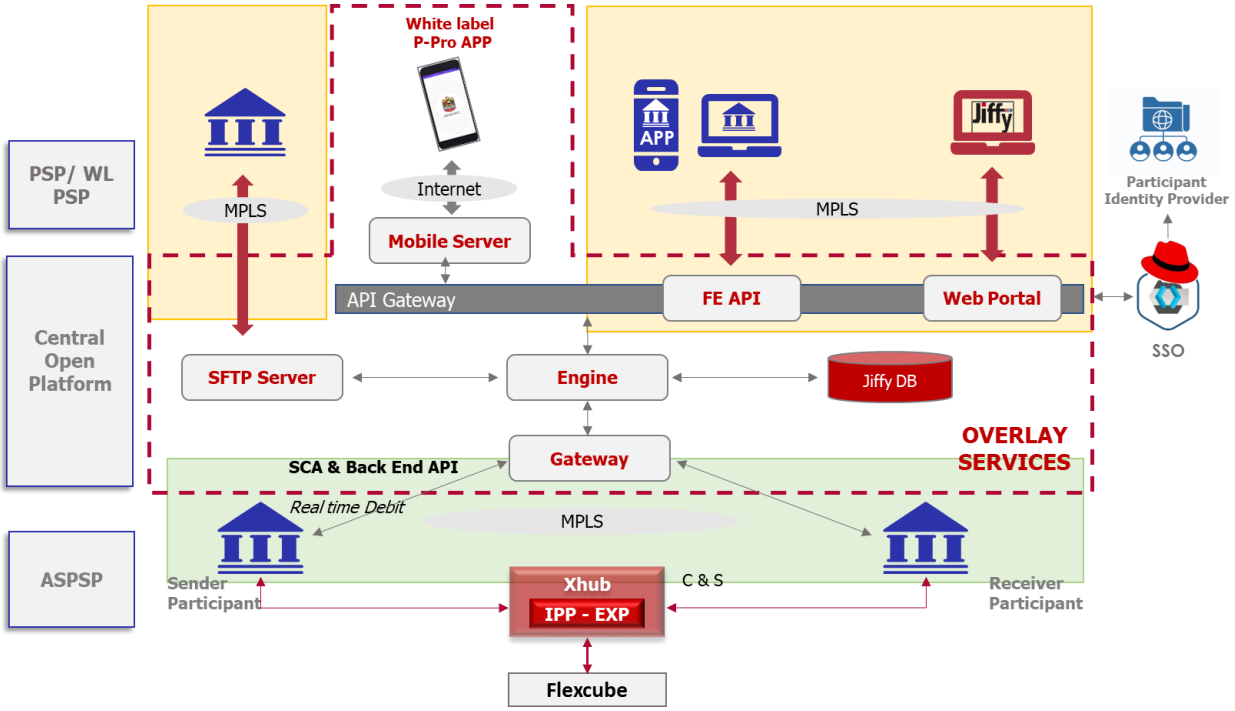


Figure 2 Integration Layers

In the above pictures are underlined different points of integration, from PSP and ASPSP points of view.

In the yellow boxes are the Front-End integration layers. Participants can use different channels to manage customers' data. The APIs Gateway exposes the APIs to Participants, and the participants can also use the Web Portal to allow setup and maintenance of end customers.

Bulk Operations are also possible for setup and maintenance, through the SFTP Server.

Moreover, the APIs Gateway exposes the APIs to Participants to allow payments management. These functionalities can be integrated directly through the participants or (optionally) through a delegated entity:

- Interface for QR Code generation and pre-authorization (in-store merchants)
- Interface for online Request to pay (e-commerce/billers)

Another layer of Front-End integration is the Mobile Bank APP. If the White Label APP is not adopted, Participants can have two possible integration components:

1. SDK to be integrated into the mobile banking APP.
2. APIs to be integrated by Participants through their own server.

The green box is the so-called Back-end integration. Participants act as servers, exposing a set of APIs (specifications will be provided in Back End Interface specification documents) to initiate payments through the UAEIPP Overlay Service Gateway. These APIs allow:

- Participant SCA integration
- Balance inquiry
- Participant account verification and pre-authorization
- Account debit
- Fund reversal

Different integration layers are indicated in the overall solution.

1.2. Functional Design

This document will describe the APIs used by the Participant , Acquirer and Provider to integrate UAEIPP Overlay Service functionalities in their own channels/Payment Gateway.

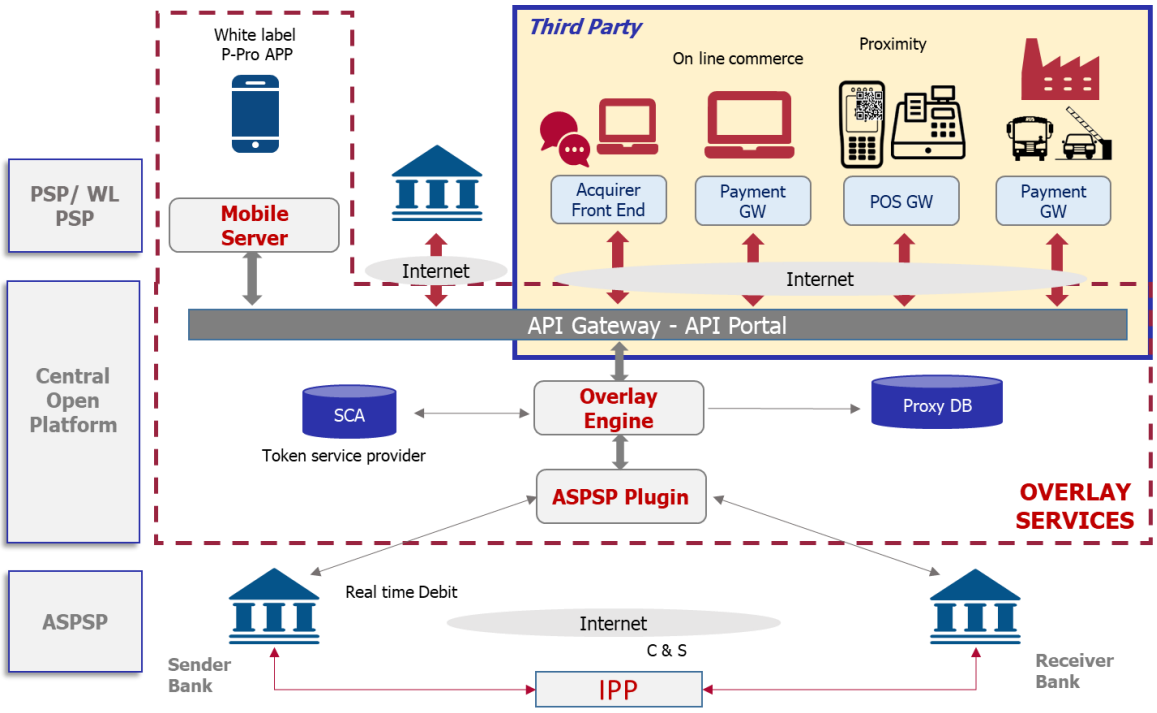


Figure 3 Front End APIs for Merchants' Services

These Front End APIs describes:

- Request to pay through Qr Code. Including all the APIs for managing the Qr Code
- Request to pay through Proxies. Including all the APIs for managing the payments' requests
- Request to pay for pre-authorization. Including all the APIs for managing the payments' requests
- Refund, disbursement, and reversal APIs.

All the APIs presented in this document has a pre-requisite: the merchants must be enrolled by the participants or technical provider as acquirer and the consumers/buyers must be enrolled by the participants. For the Enrolment APIs kindly refer to the *UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance* document.

The document is divided into two macro-areas:

- Chapter 2 includes all the flows based on the dynamic Qr Code.
- Chapter 3 includes all the flows based on the RTP.

1.2.1. Actors

A brief description of the actors involved in the process.

1. **UAEIPP Participant:**
   - The Merchant has a bank account with the Participant;
   - The Participant can enrol the merchants;
   - The Participant can use Merchants' Payments API
   - The Participant is involved in all the Payments, with BE API integration
   - The Participant is responsible for the Merchants' authorisations flow (SCA process)
2. **UAEIPP Technical Provider (providerType = "PROVIDER"):**
   - The Provider can use Merchants' Payments API
   - The Provider cannot enrol merchants
   - The Provider is not integrating the BE API for Payment initiation (the only BE API to be implemented is the Notification API – Please refer to the BE API Interface specifications)
   - For the Refund processes, or in general, for the flows where the merchant account must be debited, the Provider must start the SCA flow to obtain the Participant Authorisation
3. **UAEIPP Acquirer (providerType = "ACQUIRER")**
   - The Acquirer can use Merchants' Payments API
   - The Acquirer can enrol merchants, but the Merchant's Account belongs to a participant
   - The Acquirer is not integrating the BE API for Payment initiation (the only BE API to be implemented is the Notification API – Please refer to the BE API Interface specifications)
   - For the Refund processes, or in general, for the flows where the merchant account must be debited, the Acquirer must start the SCA flow to obtain the Participant Authorisation

4. **Buyer/Customer:**

Participant's customer (person) enrolled in the service, with an active APP.

5. **Merchant/Professional:**

Merchant Customer/Professional enrolled in the UAEIPP Overlay Service platform enabled to perform P2B services.

6. **API Gateway:**

API Gateway interfaces the platform with Participants.

7. **Participant Bank's Mobile Server:**

Participant's Mobile Server connected as a client to the API Gateway with a server-to-server approach.

1.2.2. Character set and Input Validation Rules

The fields of type "string" in the body request and response accept character of standard UTF-8, Fada characters are supported.
For the following entities, when requested as input field of any of the endpoint of the front-end API, further validations are applied on specific fields.

1.2.2.1. IBAN

IBAN checks: after removing "-" and blanks, all IBAN inserted are verified to ensure that they respect following rules:

- **not empty:** string is not empty
- **SEPA Area Country:** IBAN starts with ISO 3166-1 alpha-2 code of SEPA Area

- **IBAN Length:** Iban is max 23 character length
- **Check number:** check number is valid for provided Iban

1.2.2.2. Mobile Number

Mobile number checks: after removing "-" and blanks, all mobile numbers inserted are verified to ensure that they respect following rules:

- **not empty**: string is not empty
- **UAE mobile number**: all special characters, white spaces and 0 must be removed from the mobile number: +971 054-1768094 becomes +971541768094.
- **International mobile number** starts with + followed by varying number of digits. Max length expected is 30. All special characters and white spaces are removed from the mobile number (e.g., Italian mobile number +39 331-123 45 67 becomes +393311234567).

1.2.2.3. Document legend

Required: Values that column “Req.” of requests and response parameters can assume.

Value	Description
Y	YES- mandatory, to be set always in request or always present in response
N	not mandatory
C	conditional, present at specific condition described in field description
E	echo of the field presented in request (response body only)

1.3. Error Result Code

The following paragraphs describe error handling and the error outcomes.

All the responses with status HTTP 4xx and 5xx will be considered as errors. HTTP version supported 1.1

General response code:

- 200 – OK response
- 201 – OK Created response
- 400 – Bad request
- 401 - Unauthorized
- 500 - Internal Server Error

1.3.1. Result Code Categories

CODE	Description
00xxx	Positive result. This category includes all the codes that show a successful execution of the requested service
01xxx	Negative result. This category includes all the error events related to the app errors during the request elaboration.
02xxx	Negative result. This category includes all the error events related to input request fields wrongly filled in.
03xxx	Negative result. This category includes all the error events related to permissions/authorizations and, therefore, the service requested cannot be executed.

1.3.2. Result Code Mapping

Some messages have the placeholder {0} or {1} that will be replaced with a more specific value regarding the single API call.

For example, in case of blank input for field “IBAN”, the response message will be “The field IBAN is not filled in”; or in case of a field “name” too long, the response message will be “The field Name has not a valid format [must have at most 35 characters]”.

For each API are described the possible errors’ code.

1.3.2.1. Common result code categories

CODE	DESCRIPTION
00xxx	Positive result. This category includes all the codes that show a successful execution of the requested service
01xxx	Negative result. This category includes all the error events related to the app errors during the request elaboration.
02xxx	Negative result. This category includes all the error events related to input request fields wrongly filled in.
03xxx	Negative result. This category includes all the error events related to permissions/authorizations and, therefore, the service requested cannot be executed.

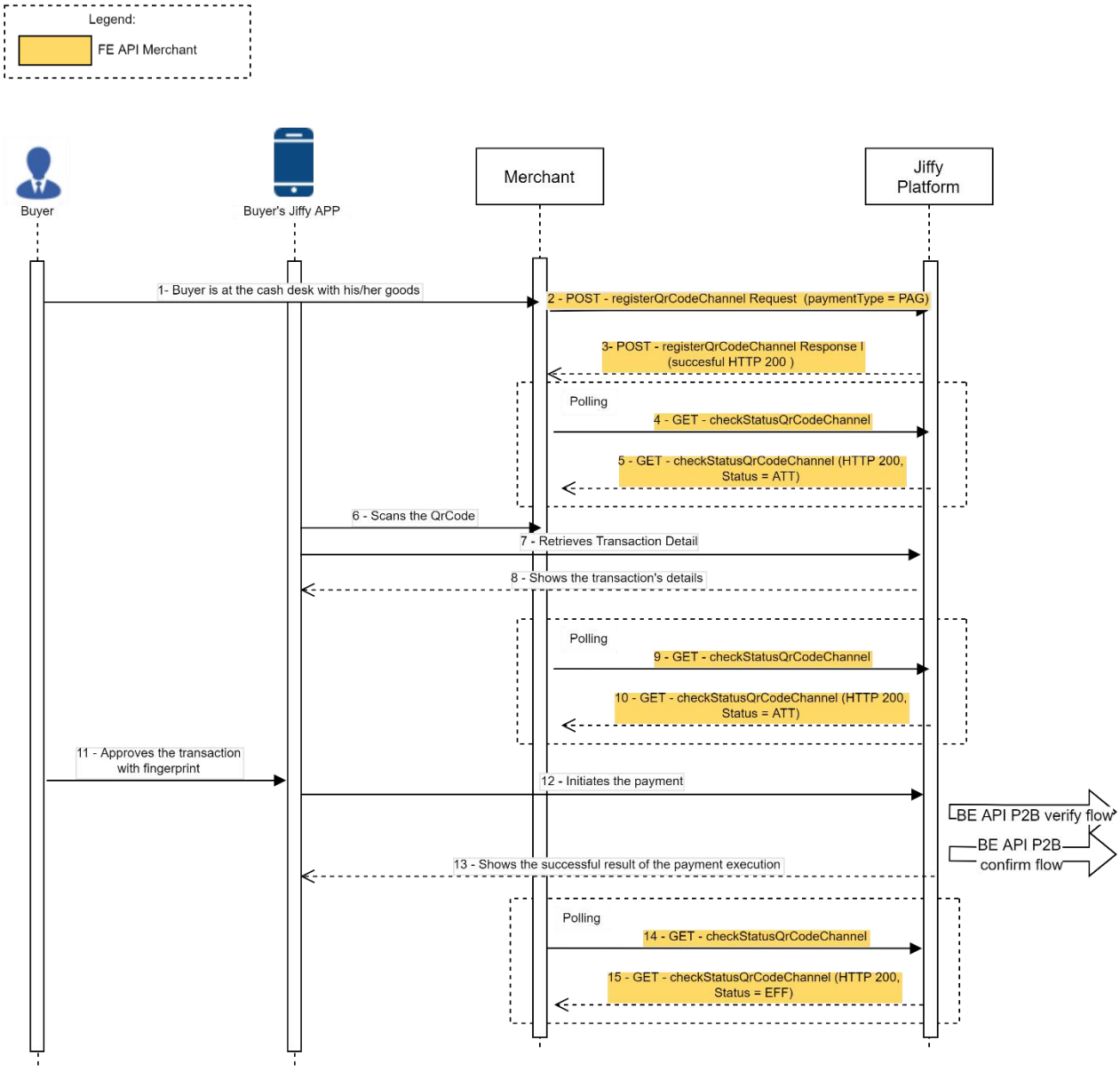
2. SEQUENCE DIAGRAMS – QR CODE

In this section are presented the sequence diagrams, describing the functional flows for different scenarios.

The flows indicate by way for example also possible steps that participant can decide to take to allow customers to use these APIs in their own systems. The participants are fully responsible of the integration.

2.1. Happy Flow Sequence Diagrams

2.1.1. Qr Code Payments



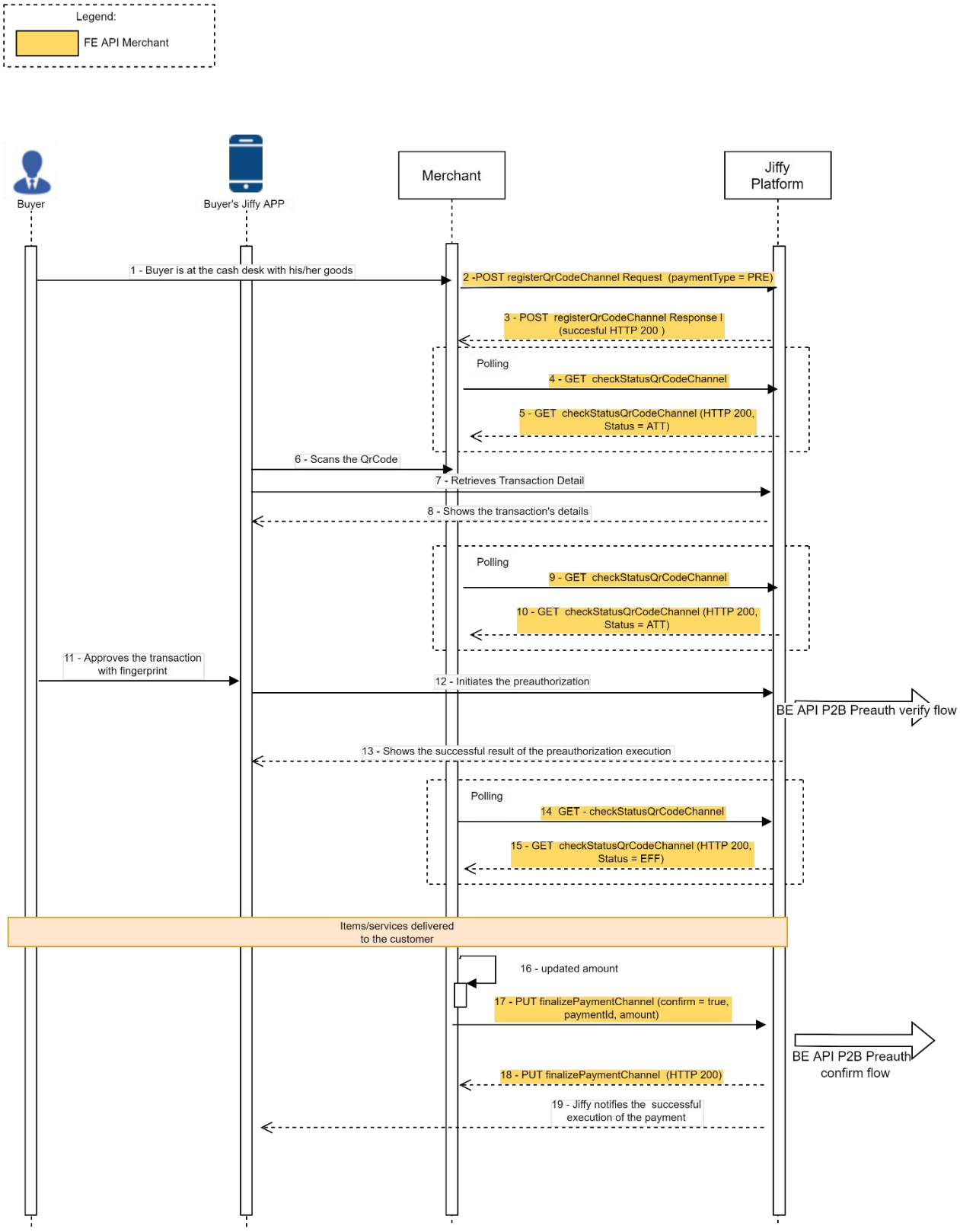
2.1.1.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the *UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance* document). The buyer must have activated the APP (White Label or Mobile Bank APP)

1. Buyer at cash desk chooses to pay with UAEIPP Overlay Service.

2. Merchant requires a QRCode to manage the payment invoking POST registerQrCodeChannel with attribute in request
a. paymentType = PAG,
3. UAEIPP Overlay Service platform replies by returning a QRCodeId (Merchant renders the response in a valid QRCode, see appendix for further details).
4. Merchant starts polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
5. UAEIPP Overlay Service platform replies with status “ATT” (Merchant keeps on polling).
6. Buyer scans with UAEIPP Overlay Service app QrCode Merchant rendered on a display.
7. Buyer’s UAEIPP Overlay Service app queries payment details from UAEIPP Overlay Service Platform.
8. UAEIPP Overlay Service platform replies with payment details to Buyer’s UAEIPP Overlay Service app.
9. Merchant keeps on polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
10. UAEIPP Overlay Service platform replies with status “ATT” (Merchant keeps on polling).
11. Buyer approves payment with UAEIPP Overlay Service app and provide biometrics (or pin).
12. Buyer’s UAEIPP Overlay Service app forwards payment confirmation to UAEIPP Overlay Service platform. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
13. UAEIPP Overlay Service Platform successfully orchestrates payment flows with participants (both Buyer’s and Merchant’s). then returns to Buyer’s UAEIPP Overlay Service app successful result of the payment. Refer to *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
14. Merchant keeps on polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
15. UAEIPP Overlay Service platform replies with status “EFF”; Merchant now knows that payment has been executed (Merchant stops polling).

2.1.2. Qr Code preauthorization Confirmed



2.1.2.1. Sequence diagram description

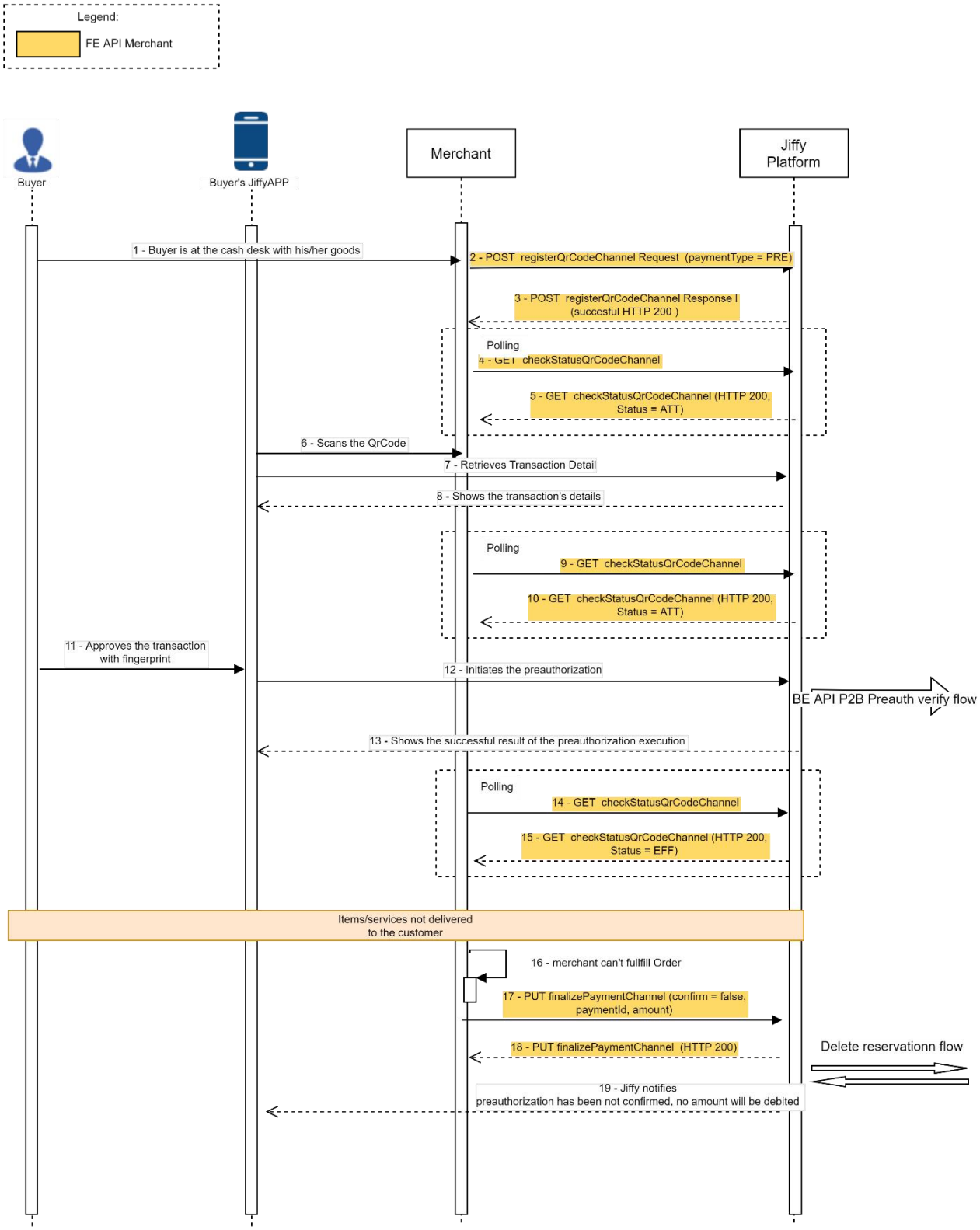
Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP).

1. Buyer at cashdesk choose to pay with UAEIPP Overlay Service.
2. Merchant requires a QRCode to manage the preauthorized payment invoking POST registerQrCodeChannel with attribute in request

a. paymentType = PRE,

3. UAEIPP Overlay Service platform replies by returning a QRCodeId (Merchant renders the response in a valid QRCode, see appendix for further details).
4. Merchant starts polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
5. UAEIPP Overlay Service platform replies with status “ATT” (Merchant keeps on polling).
6. Buyer scans with UAEIPP Overlay Service app QrCode Merchant rendered on a display.
7. Buyer’s UAEIPP Overlay Service app queries payment details from UAEIPP Overlay Service Platform.
8. UAEIPP Overlay Service platform replies with payment details to Buyer’s UAEIPP Overlay Service app.
9. Merchant keeps on polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
10. UAEIPP Overlay Service platform replies with status “ATT” (Merchant keeps on polling).
11. Buyer approves preauthorized payment with UAEIPP Overlay Service app and provide biometrics (or pin).
12. Buyer’s UAEIPP Overlay Service app forwards preauthorized payment confirmation to UAEIPP Overlay Service platform.
13. UAEIPP Overlay Service Platform successfully orchestrates preauthorized payment verify flows with participants (both Buyer’s and Merchant’s) then returns to Buyer’s UAEIPP Overlay Service app successful result of the preauthorized payment.
14. Merchant Keeps on polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
15. When UAEIPP Overlay Service platform replies with status “EFF”; Merchant now knows that preauthorized payment has been executed (Merchant stops polling).
16. Merchant provides goods and services to Buyer.
17. Merchant prepares order, preauthorized amount needs to be updated in case not same. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
18. Merchant notifies to UAEIPP Overlay Service platform to confirm preauthorized payment (the one approved at step 11) invoking PUT finalizePaymentChannel with parameter confirm = true, providing new amount (lower or equal than the one approved at step 11).
19. UAEIPP Overlay Service Platform successfully orchestrates preauthorized payment confirm flows with participants (both Buyer’s and Merchant’s) then returns to Buyer’s UAEIPP Overlay Service app successful result of the preauthorized payment amount update.
20. UAEIPP Overlay Service Platform notifies to Buyer’s UAEIPP Overlay Service app final amount.

2.1.3. Qr Code Preauthorization not confirmed



2.1.3.1. Sequence diagram description

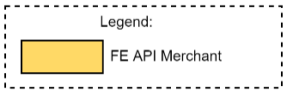
Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP).

1. Buyer at cashdesk choose to pay with UAEIPP Overlay Service
2. Merchant requires a QRCode to manage the preauthorized payment invoking POST registerQrCodeChannel with attribute in request

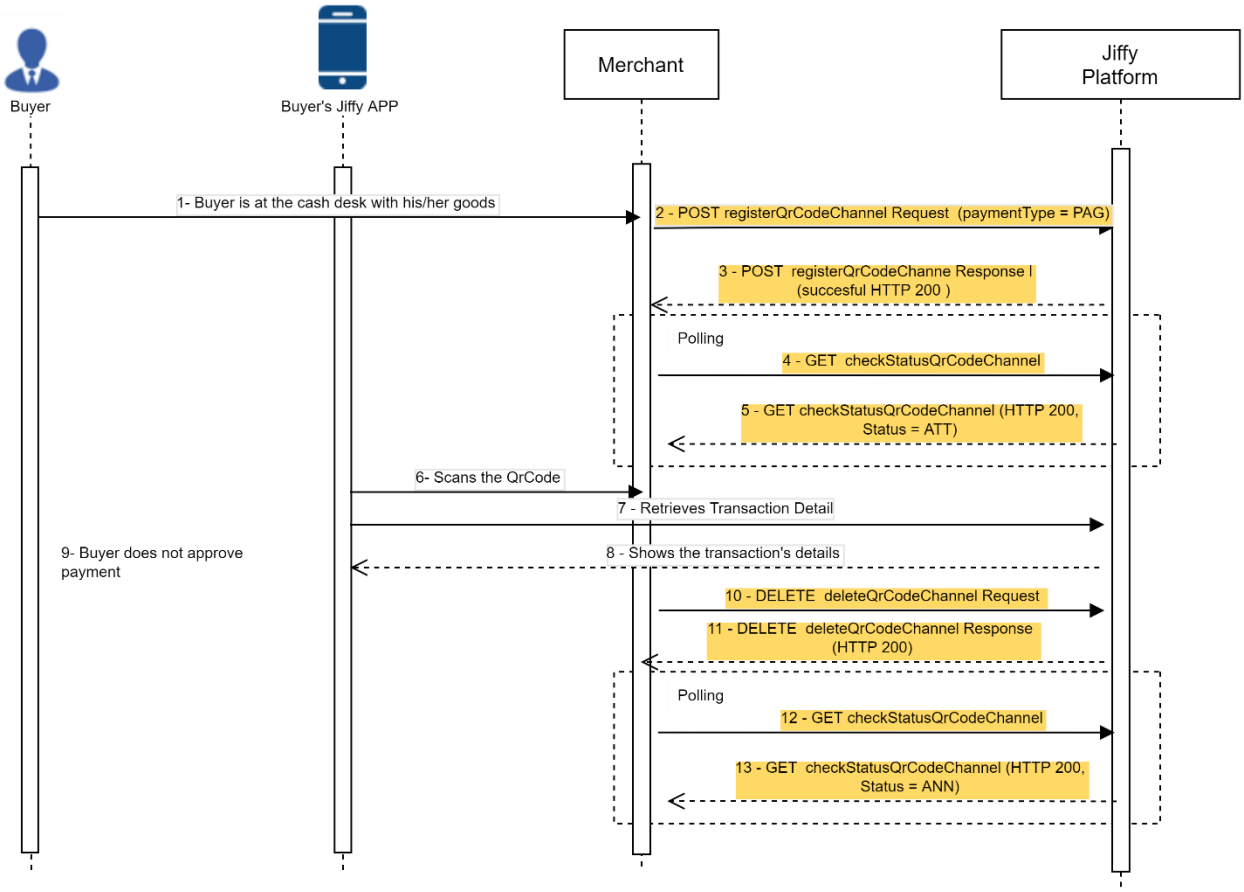
a. paymentType = PRE,

3. UAEIPP Overlay Service platform replies by returning a QRCodeId (Merchant renders the response in a valid QRCode, see appendix for further details).
4. Merchant starts polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
5. UAEIPP Overlay Service platform replies with status “ATT” (Merchant keeps on polling).
6. Buyer scans with UAEIPP Overlay Service app QrCode Merchant rendered on a display.
7. Buyer’s UAEIPP Overlay Service app queries payment details from UAEIPP Overlay Service Platform.
8. UAEIPP Overlay Service platform replies with payment details to Buyer’s UAEIPP Overlay Service app.
9. Merchant keeps on polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
10. UAEIPP Overlay Service replies with status “ATT” (Merchant keeps on polling).
11. Buyer approves preauthorized payment with the APP and provide biometrics (or pin).
12. Buyer’s APP forwards preauthorized payment confirmation to UAEIPP Overlay Service.
13. UAEIPP Overlay Service successfully orchestrates preauthorized payment verify flows with participants (both Buyer’s and Merchant’s) then returns to Buyer’s UAEIPP Overlay Service app successful result of the preauthorized payment.
14. Merchant Keeps on polling UAEIPP Overlay Service platform to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
15. UAEIPP Overlay Service platform replies with status “EFF”, Merchant now knows that preauthorized payment has been executed (Merchant stops polling).
16. Merchant can’t fulfil order or customer does not buy goods; preauthorization needs to be unconfirmed
17. Merchant notifies to UAEIPP Overlay Service to cancel preauthorized payment (the one approved at step 11) invoking PUT finalizePaymentChannel with parameter confirm = false.
18. UAEIPP Overlay Service successfully orchestrates Delete reservation flow with Buyer’s participant.
19. UAEIPP Overlay Service notifies to Buyer’s APP that preauthorization has not been confirmed and no amount will be debited.

2.1.4. Qr Code Delete: Qr Code payment not confirmed



QRcode delete can be used optionally by merchant to cancel a QRcode scanned but not approved by the Buyer.

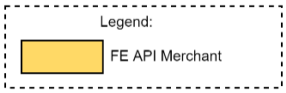


2.1.4.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchants must have generated a Qr Code, this scenario is valid for both payment and preauthorized payments, steps from 1 to 8 are equal to the ones at Qr Code Payments Sequence Diagram.

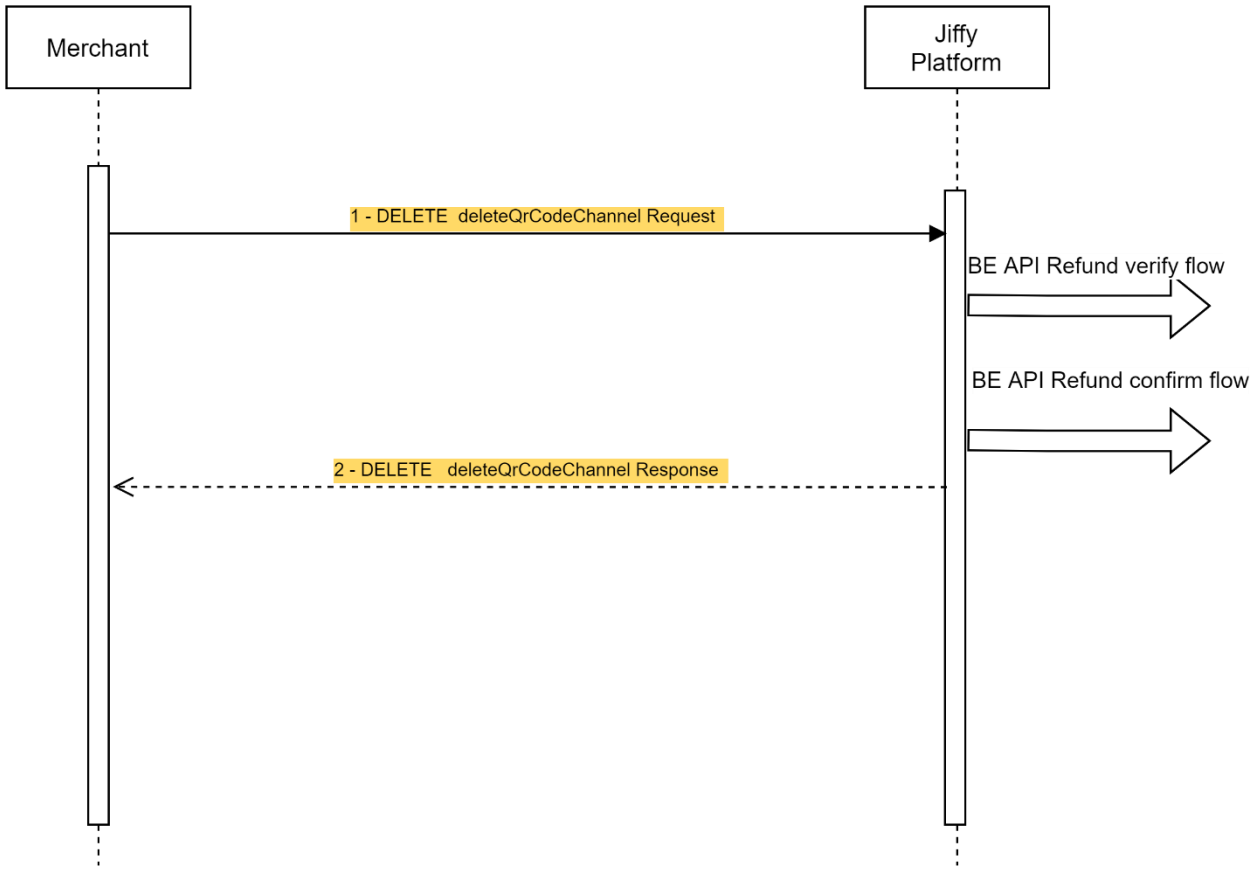
9. Buyer does not approve payment.
10. For some business, reason Merchant decides to cancel the payment request represented by the QRCode then invokes DELETE deleteQrCodeChannel. This scenario occurs before Buyer scans QRCode with the APP (e.g.: merchants enter wrong amount in POS or ECR hence needs to cancel payment request pressing any cancel button on POS or ECR).
11. UAEIPP Overlay Service replies with successful response, QRCode has been deleted.
12. Merchant keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
13. UAEIPP Overlay Service replies with status “ANN” because QRCode has been deleted (Merchant stops polling).

2.1.5. Qr Code: Payment or Preauthorized Confirmed



QRcode delete can be used by merchant to cancel a QRcode, scanned and approved by the Buyer.

In this scenario Refund, flow will be triggered to the Participants.



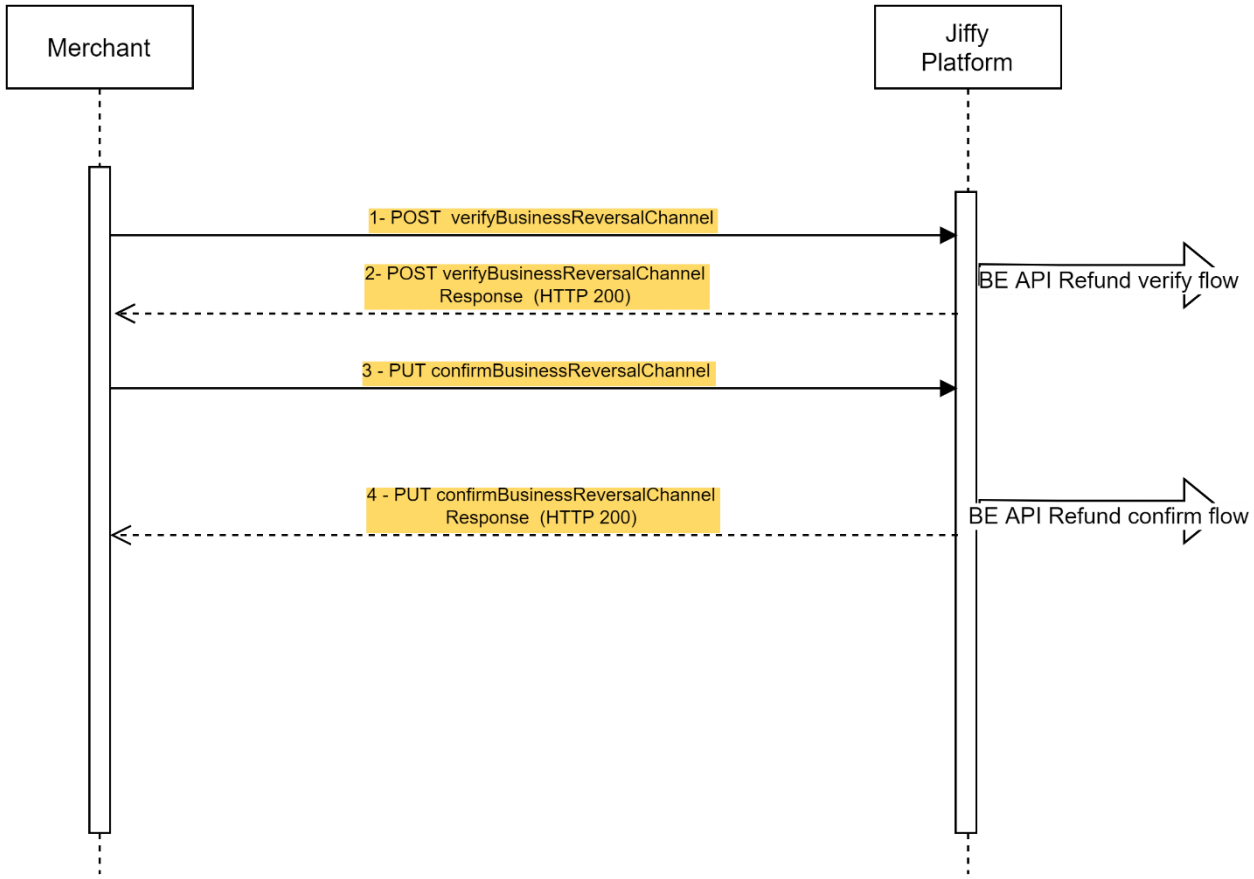
2.1.5.1. **Sequence diagram description**

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant must have generated the Qr Code (Qr Code Payments Sequence Diagram) and the Buyer has accepted it.
This scenario is valid for both payment and preauthorized payments and may be run alternatively to Reversal, because once run after a QrCode Payment or QrCode Preauthorized payment has been successfully ended, UAEIPP Overlay Service orchestrates Refund flows with participants.
Note: as the Delete QrCode triggers the BE API for the Refund process, the platform must refresh the merchant token considering the following validations:

- The Participant account to be debited.
- The last refresh token registered in the DB for the specific participant to be debited.

1. For some business reason Merchant decides to cancel the QRCode payment then invokes DELETE deleteQrCodeChannel. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
2. UAEIPP Overlay Service runs Refund verify flows with participants (both Merchant's and Buyer's one) then replies to successful response.

2.1.6. Reversal



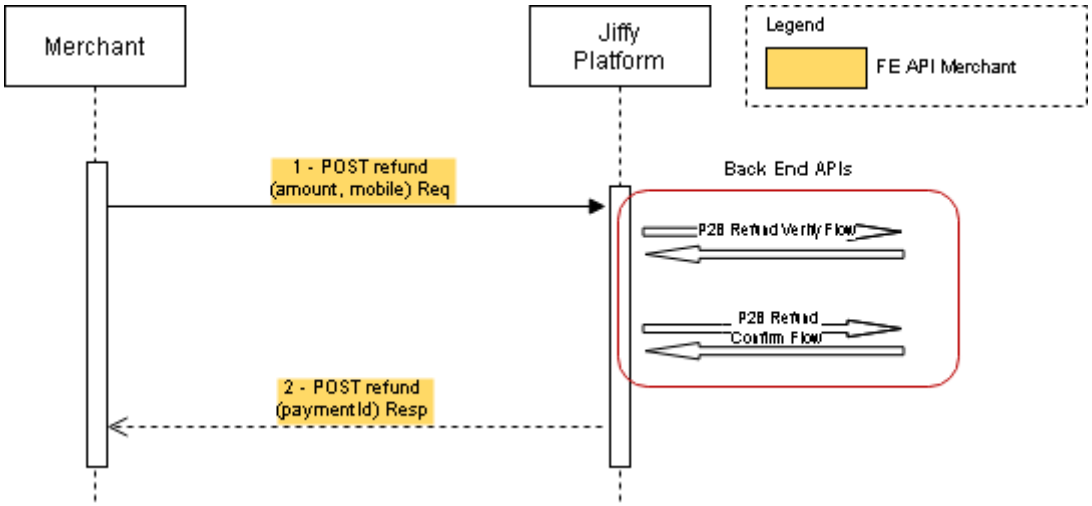
2.1.6.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant must have generated the Qr Code (Qr Code Payments Sequence Diagram) and the Buyer has accepted it.
The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Reversal, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes POST verifyBusinessReversalChannel to verify that a payment exists on the platform and has been previously approved. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
2. UAEIPP Overlay Service runs Refund verify flows with participants (both Merchant's and Buyer's one) then replies to successful response.

3. Merchant invokes PUT confirmBusinessReversalChannel to confirm the reversal for the previously verified approved payment. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
4. UAEIPP Overlay Service runs Refund confirm flows with participants (both Merchant's and Buyer's one) then replies successful response.

2.1.7. Refund



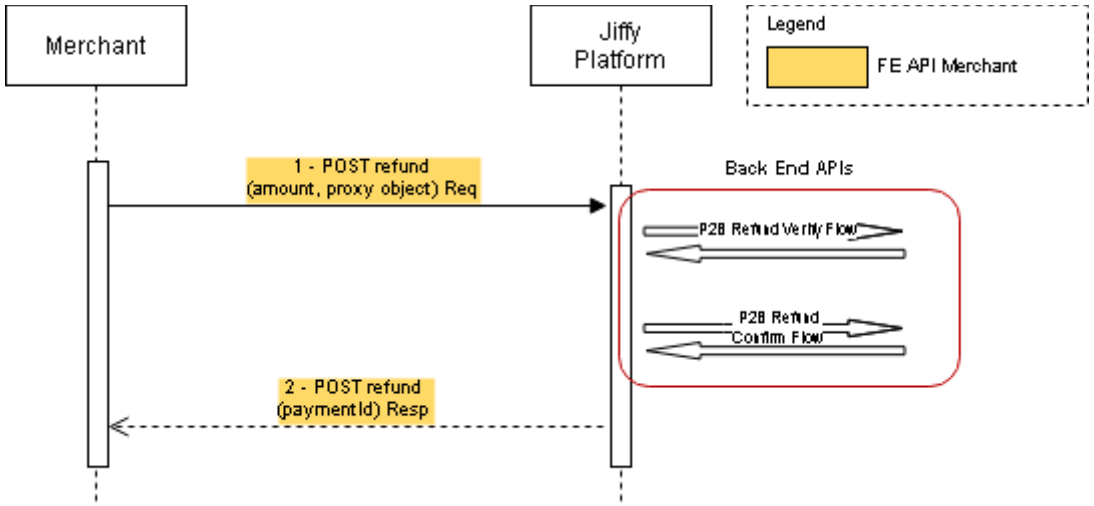
2.1.7.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The merchant wants to refund the customer, there is no need of any previous transaction information (different from the Reversal flow).

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Refund, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's mobile number, without entering any reference to the original payment of the buyer. After that, the verify and confirm BE APIs are sequentially called in order to perform the refund transaction from merchant's bank to buyer's bank.
2. If the refund transaction has been successfully processed and orchestrated, the UAEIPP Overlay Service replies to the Merchant with a positive message meaning that the buyer's bank account will be credited.

2.1.8. Refund with a proxy



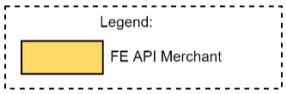
2.1.8.1. Sequence diagram description

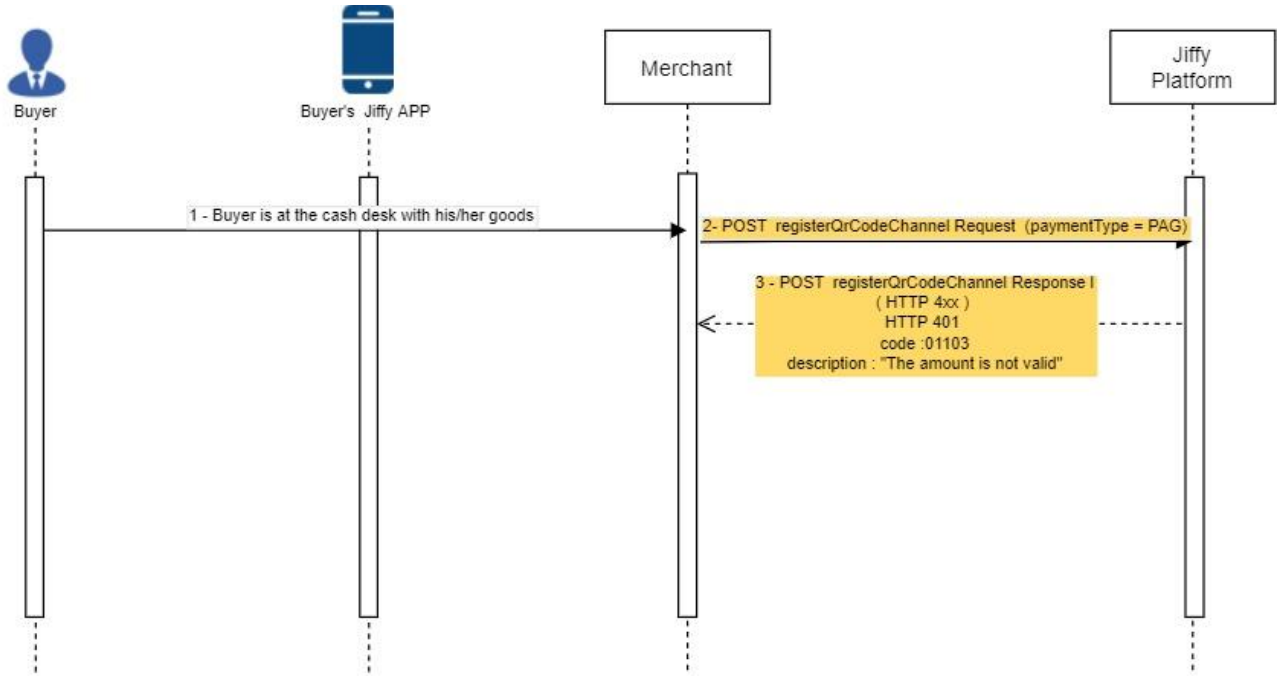
Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The merchant wants to refund the customer, there is no need of any previous transaction information (different from the Reversal flow)
The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Refund, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's proxy inside its object (email or document-id), without entering any reference to the original payment of the buyer. After that, the verify and confirm BE APIs are sequentially called to perform the refund transaction from merchant's participant to buyer's participant.
2. If the refund transaction has been successfully processed and orchestrated, the UAEIPP Overlay Service replies to the Merchant with a positive message meaning that the buyer's participant account will be credited.

2.2. Unhappy Flow Sequence Diagrams

2.2.1. QrCode Payments or Preauthorized payments: Merchant can't generate QRCode





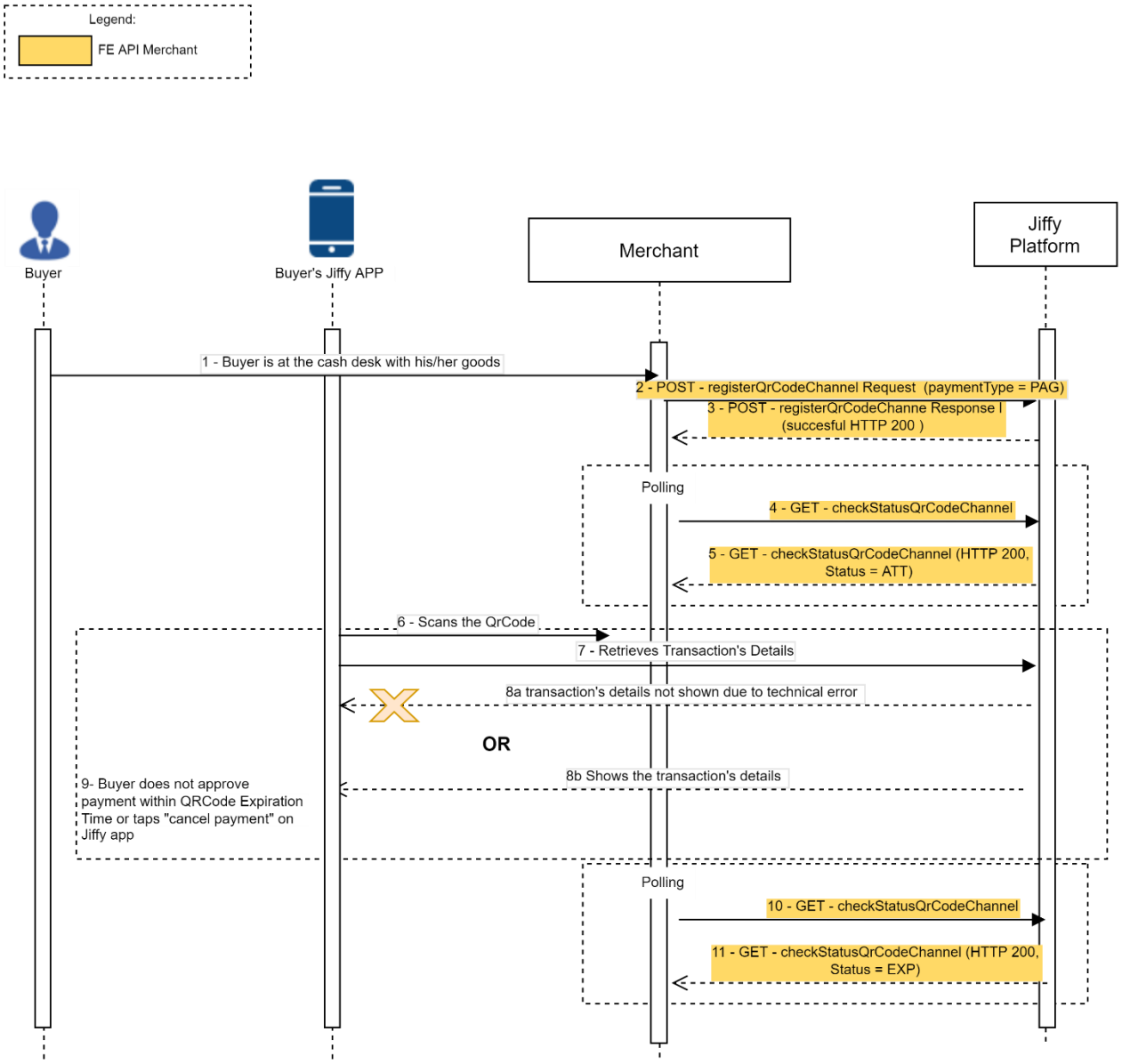
2.2.1.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled into UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP).

1. Buyer at cashdesk choose to pay with UAEIPP Overlay Service.
2. Merchant requires a QRCode to manage the payment invoking POST registerQrCodeChannel with attribute in request
    a. paymentType = PAG,
3. UAEIPP Overlay Service replies by returning an unsuccessful response.

This scenario is applied in case, at step 2, paymentType = PRE.

2.2.2. QrCode Payments or Preauthorized payments: Buyer cancels / does not approve Payment (QRCode expires)



2.2.2.1. **Sequence diagram description**

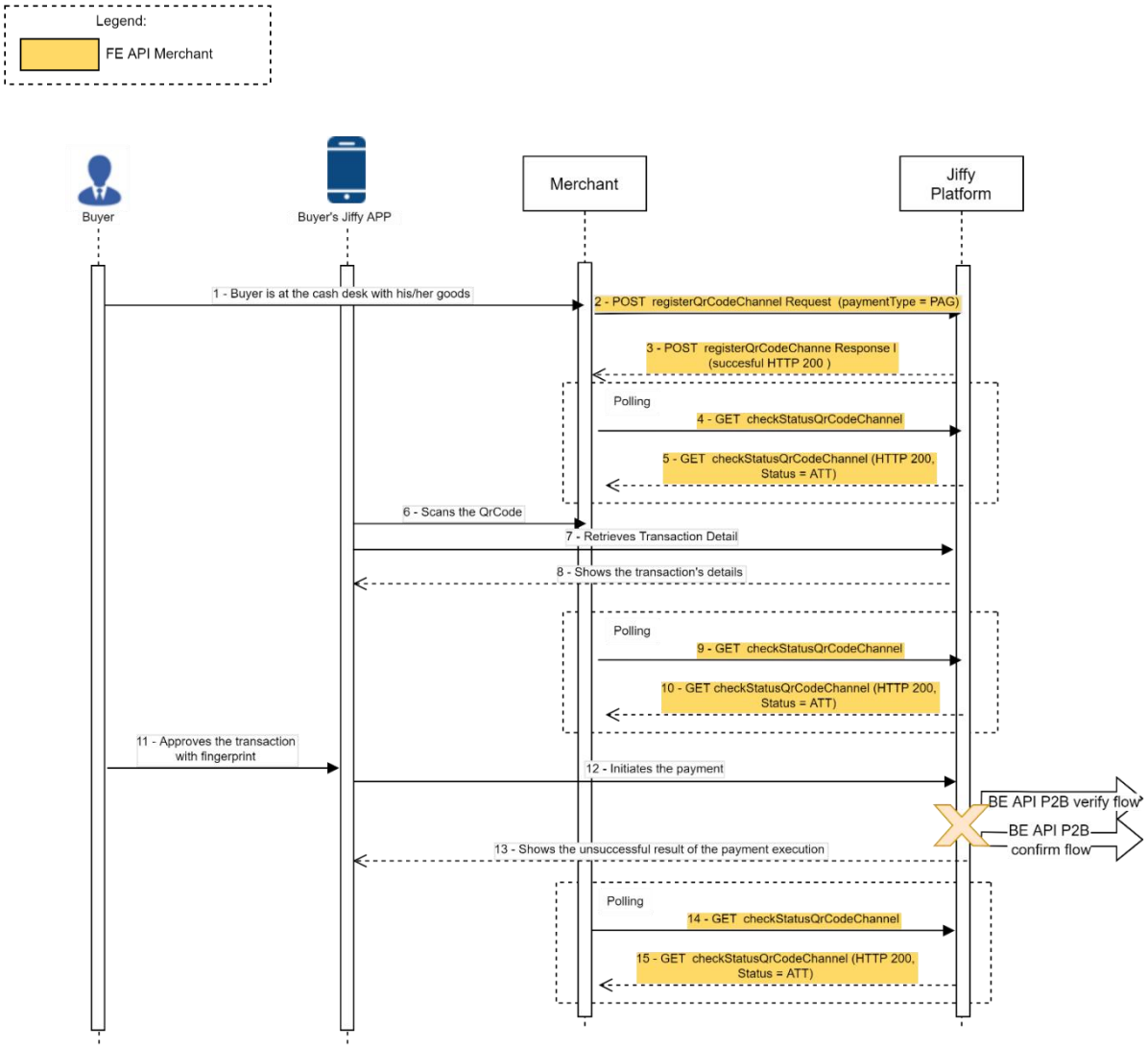
Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP).

1. Buyer at cashdesk choose to pay with UAEIPP Overlay Service.
2. Merchant requires a QRCode to manage the payment invoking POST registerQrCodeChannel with attribute in request
    a. paymentType = PAG,
3. UAEIPP Overlay Service replies by returning a QRCodeId (Merchant renders the response in a valid QRCode, see appendix for further details).
4. Merchant starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
5. UAEIPP Overlay Service replies with status “ATT” (Merchant keeps on polling).

6. Buyer scans with the APP QrCode Merchant rendered on a display.
7. Buyer's APP queries payment details from UAEIPP Overlay Service.
8. UAEIPP Overlay Service
    a. replies with payment details to Buyer's APP but details are not shown on Buyer's UAEIPP Overlay Service due to technical error.
    OR
    b. replies with payment details to Buyer's APP , then (9)Buyer does not approve or taps "cancel payment" on the APP.
9. Buyer does not approve or taps "cancel payment" on UAEIPP Overlay Service APP.
10. Merchant Keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
11. UAEIPP Overlay Service replies with status "EXP", Merchant now knows that payment has not been executed (Merchant stops polling).

This scenario is applied in case, at step 2, paymentType = PRE.

2.2.3. QrCode Payments: UAEIPP Overlay Service flows with banks is unsuccessful



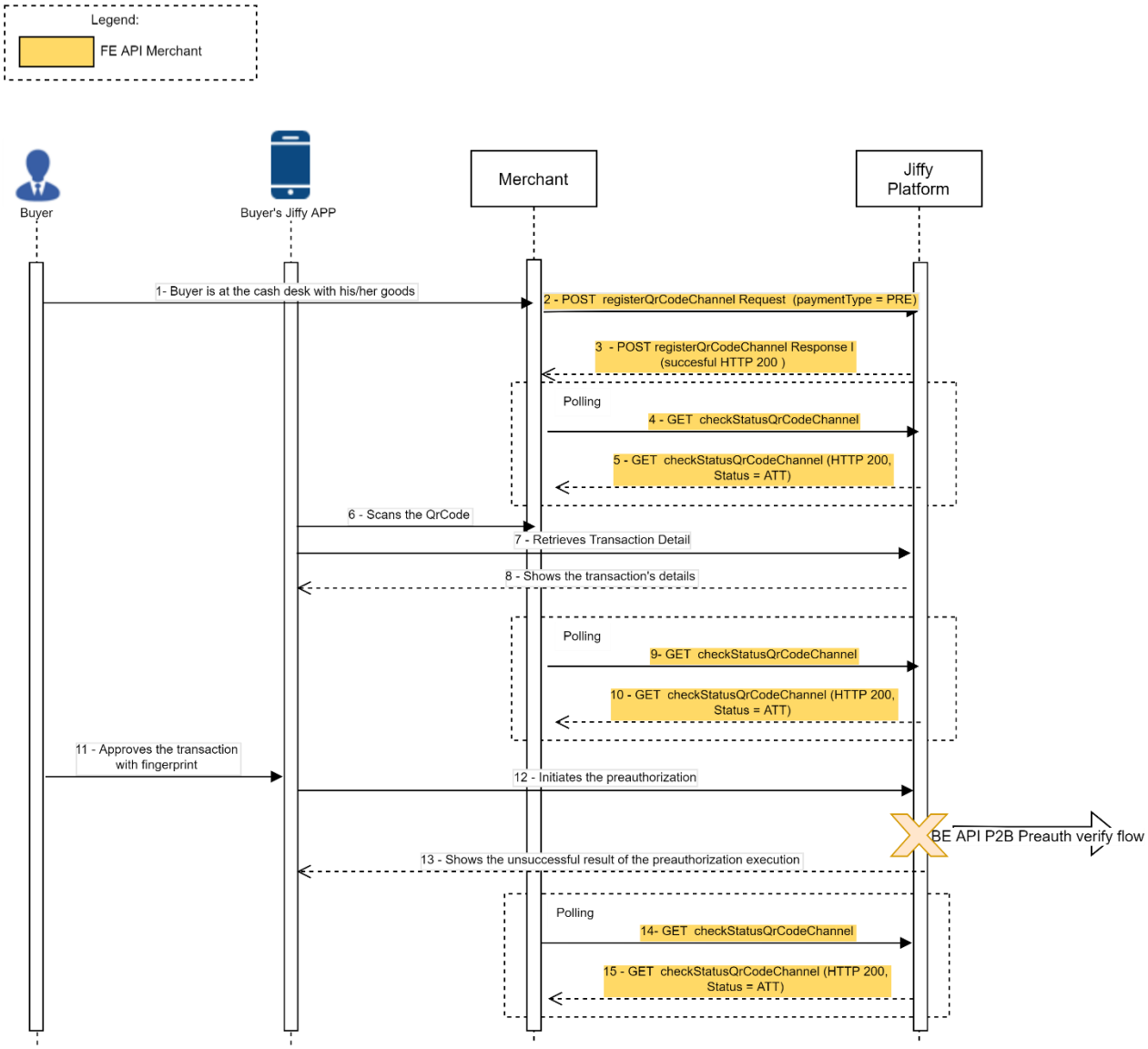
2.2.3.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP).

1. Buyer at cashdesk choose to pay with UAEIPP Overlay Service.
2. Merchant requires a QRCode to manage the preauthorized payment invoking POST registerQrCodeChannel with attribute in request
    a. paymentType = PRE,
3. UAEIPP Overlay Service replies by returning a QRCodeId (Merchant renders the response in a valid QRCode, see appendix for further details).
4. Merchant starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
5. UAEIPP Overlay Service replies with status “ATT” (Merchant keeps on polling).
6. Buyer scans with the UAEIPP Overlay Service APP QrCode Merchant rendered on a display.
7. Buyer’s APP queries payment details from UAEIPP Overlay Service.
8. UAEIPP Overlay Service replies with payment details to Buyer’s UAEIPP Overlay Service APP.
9. Merchant keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
10. UAEIPP Overlay Service replies with status “ATT”(Merchant keeps on polling).
11. Buyer approves preauthorized payment with the APP and provide biometrics (or pin).
12. Buyer’s APP forwards preauthorized payment confirmation to UAEIPP Overlay Service. Refer to the UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration document for details on APIs to be exposed by the Participants for Payment Initiation.
13. UAEIPP Overlay Service unsuccessfully orchestrates preauthorized payment verify flows with participants (both Buyer’s and Merchant’s) then returns to Buyer’s Overlay Service APP unsuccessful result of the preauthorized payment.
14. Merchant Keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
15. UAEIPP Overlay Service replies with status “ATT” (Merchant keeps on polling).

Even if UAEIPP Overlay Service did not run flows with participants successfully, QRCode is still in status “ATT”, because Buyers itself or another Buyer with the APP could try to scan the QRCode again (within QRcode Expiration time).

2.2.4. QrCode Preauthorization: UAEIPP Overlay Service flow with participants is unsuccessful



2.2.4.1. **Sequence diagram description**

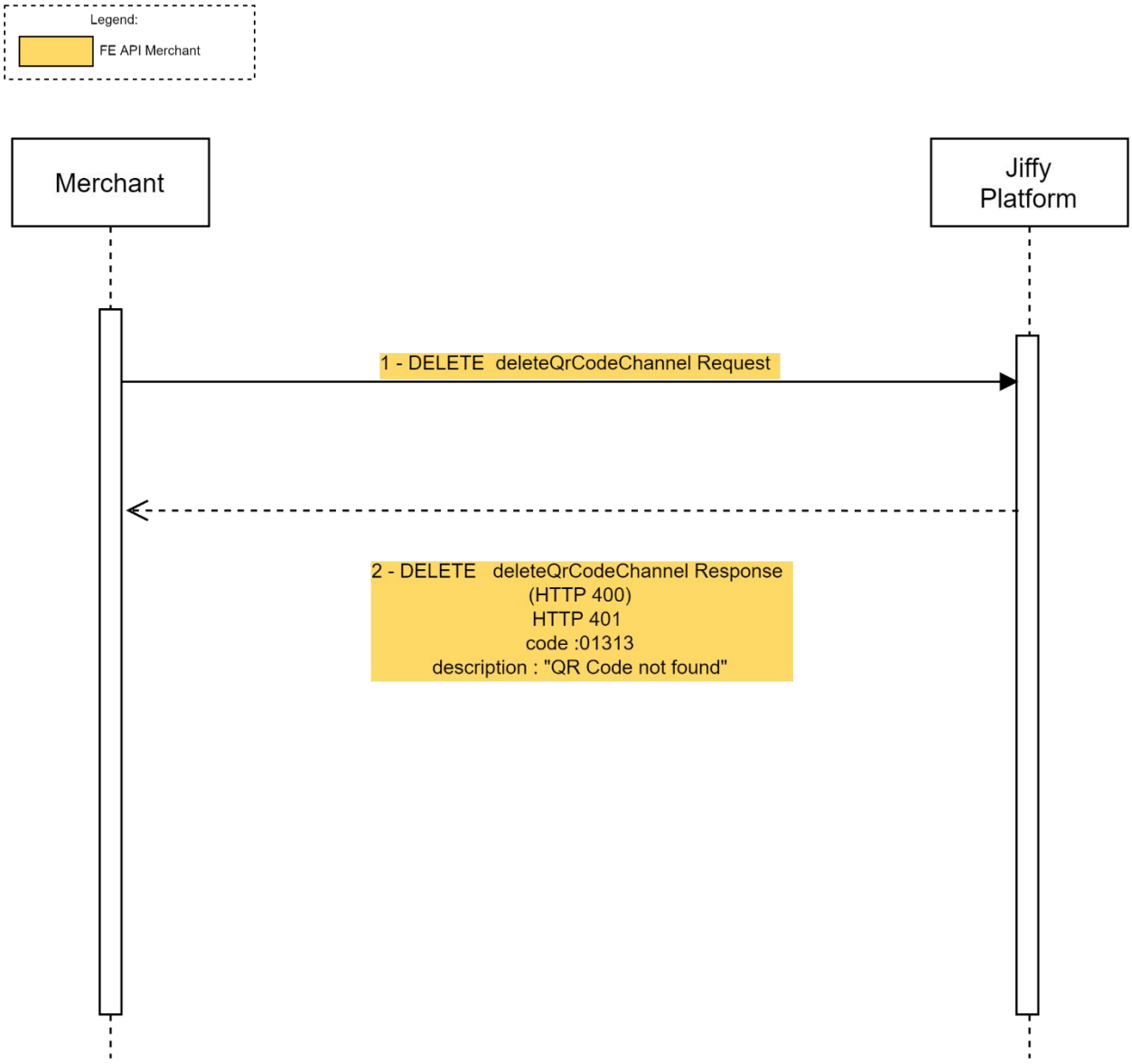
Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to Customers' Setup and Maintenance Interface Specifications). The buyer must have activated the APP (White Label or Mobile Bank APP).

1. Buyer at cashdesk choose to pay with UAEIPP Overlay Service.
2. Merchant requires a QRCode to manage the preauthorized payment invoking POST registerQrCodeChannel with attribute in request
    a. paymentType = PRE,
3. UAEIPP Overlay Service replies by returning a QRCodeId (Merchant renders the response in a valid QRCode, see appendix for further details).
4. Merchant starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
5. UAEIPP Overlay Service replies with status "ATT" (Merchant keeps on polling).
6. Buyer scans with the APP QrCode Merchant rendered on a display.
7. Buyer's APP queries payment details from UAEIPP Overlay Service.
8. UAEIPP Overlay Service replies with payment details to Buyer's UAEIPP Overlay Service APP.

9. Merchant keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
10. UAEIPP Overlay Service replies with status “ATT”(Merchant keeps on polling).
11. Buyer approves preauthorized payment with the APP and provide biometrics (or pin).
12. Buyer’s UAEIPP Overlay Service APPUAEIPP Overlay Service APP forwards preauthorized payment confirmation to UAEIPP Overlay Service. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
13. UAEIPP Overlay Service unsuccessfully orchestrates preauthorized payment verify flows with participants (both Buyer’s and Merchant’s) then returns to Buyer’s APP unsuccessful result of the preauthorized payment.
14. Merchant Keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking GET checkStatusQRCodeChannel.
15. UAEIPP Overlay Service replies with status “ATT” (Merchant keeps on polling).

Even if UAEIPP Overlay Service did not run flows with participants successfully, QRCode is still in status “ATT”, because Buyers itself or another Buyer with the APP could try to scan the QRCode again (within QRcode Expiration time).

2.2.5. QrCode Delete: delete QrCode request fails with UAEIPP Overlay Service



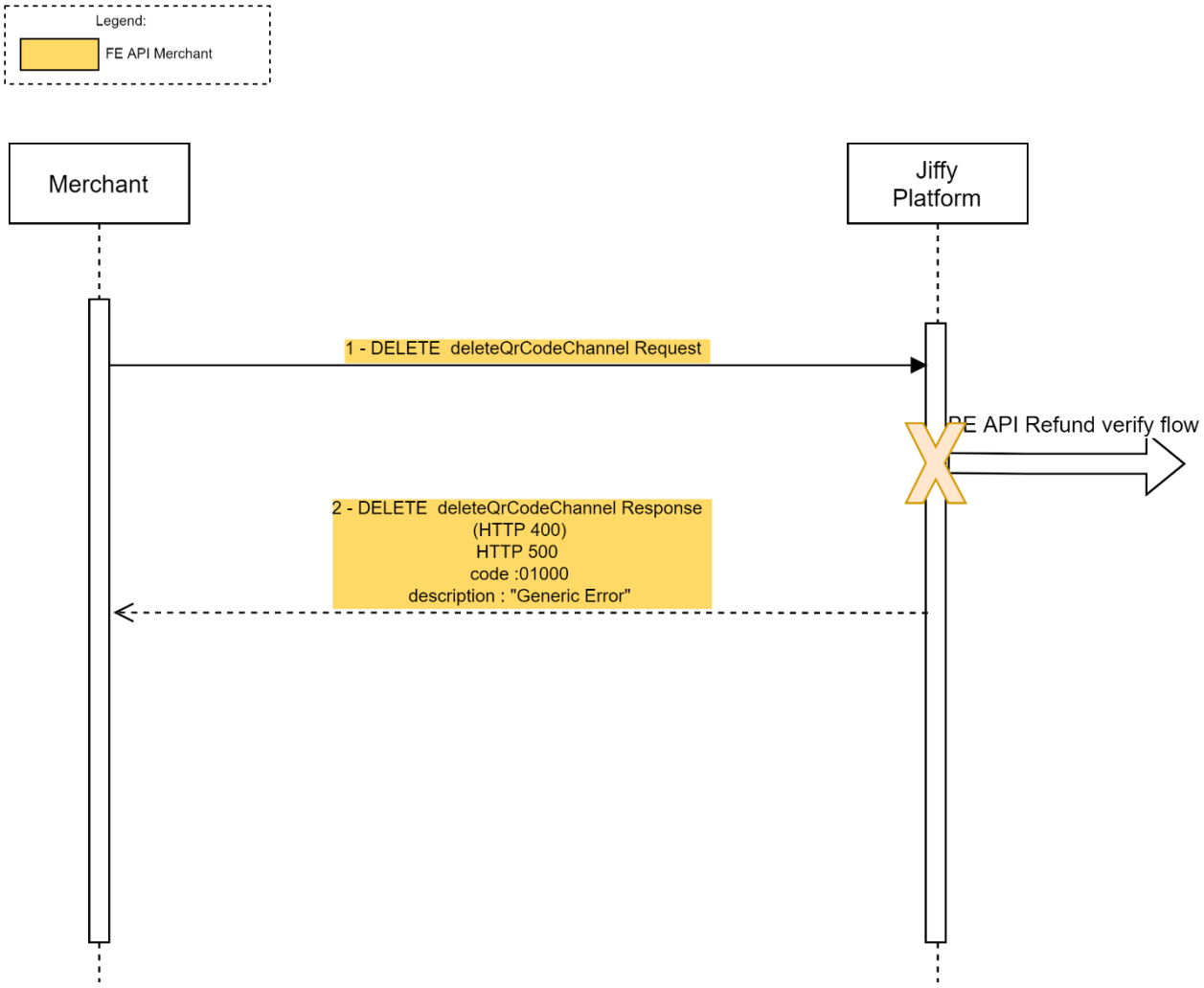
2.2.5.1. **Sequence diagram description**

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant must have generated the Qr Code (Qr Code Payments Sequence Diagram) and the Buyer has accepted it.

This scenario is valid for both payment and preauthorized payments.

1. For some business reason Merchants decide to cancel the payment then invokes DELETE deleteQrCodeChannel.
2. UAEIPP Overlay Service replies to unsuccessful response due to technical errors(e.g.: QRCodeId not found, missing fields or invalid field format in request).

2.2.6. QrCode Delete: delete QrCode request fails due to Refund verify flows with banks is unsuccessful



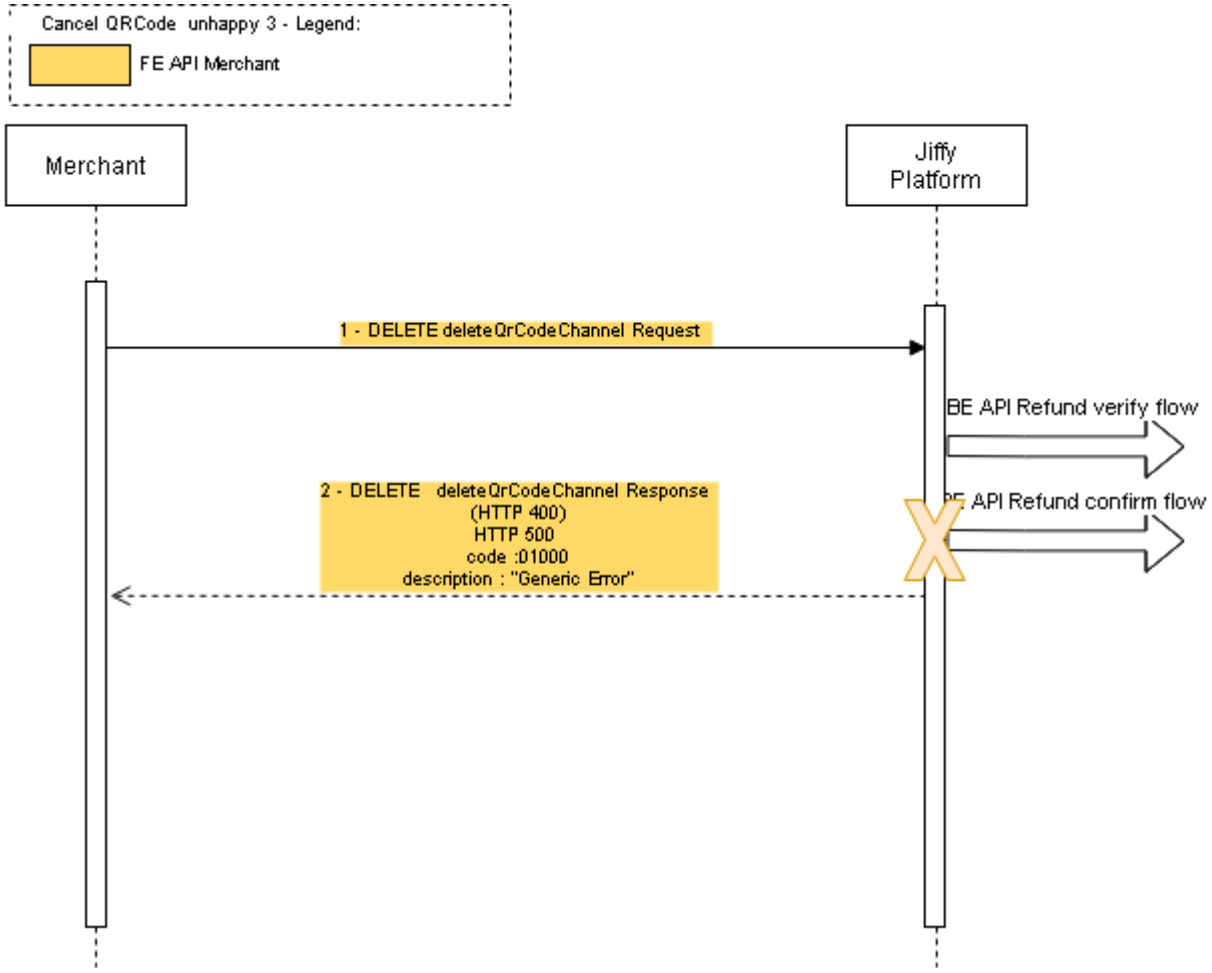
2.2.6.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant must have generated the Qr Code (Qr Code Payments Sequence Diagram) and the Buyer has accepted it.

This scenario is valid for both payment and preauthorized payments.

1. For some business reason Merchants decide to cancel the payment then invokes DELETE deleteQrCodeChannel. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration document* for details on APIs to be exposed by the Participants for Payment Initiation.
2. UAEIPP Overlay Service unsuccessfully runs Refund verify flows with banks (both Merchant's and Buyer's one), then replies to unsuccessful response to Merchant.

2.2.7. QrCode Delete: UAEIPP Overlay Service Refund Confirm flows with banks is unsuccessful



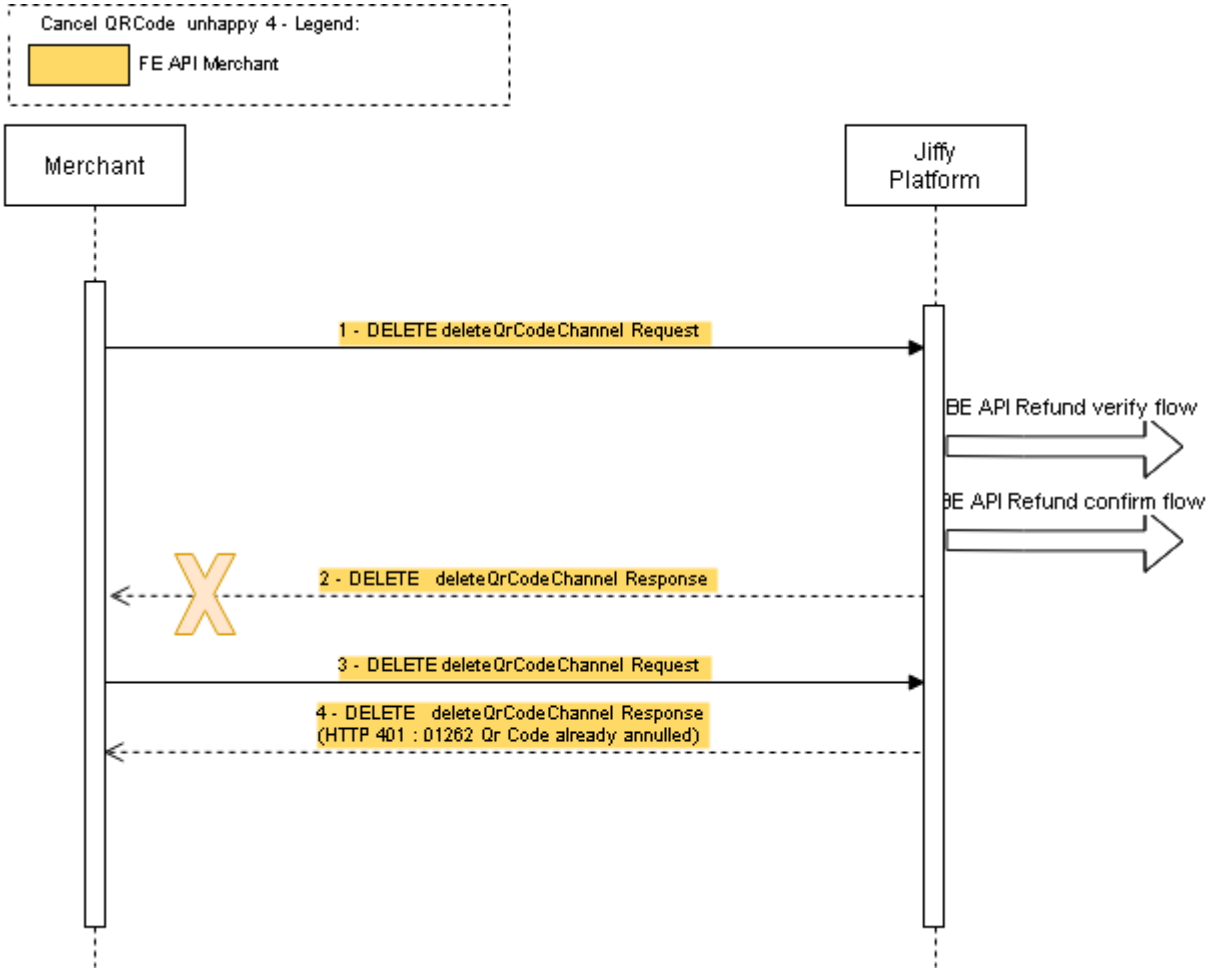
2.2.7.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant must have generated the Qr Code (Qr Code Payments Sequence Diagram) and the Buyer has accepted it.

This scenario is valid for both payment and preauthorized payments.

1. For some business reason Merchants decide to cancel the payment then invokes DELETE deleteQrCodeChannel. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on Refund API to be exposed by the Participants for Payment Initiation (Refund Process).
2. UAEIPP Overlay Service unsuccessfully runs Refund Confirm flows with banks after successful verify flow (both Merchant's and Buyer's one), then replies to unsuccessful response to Merchant.

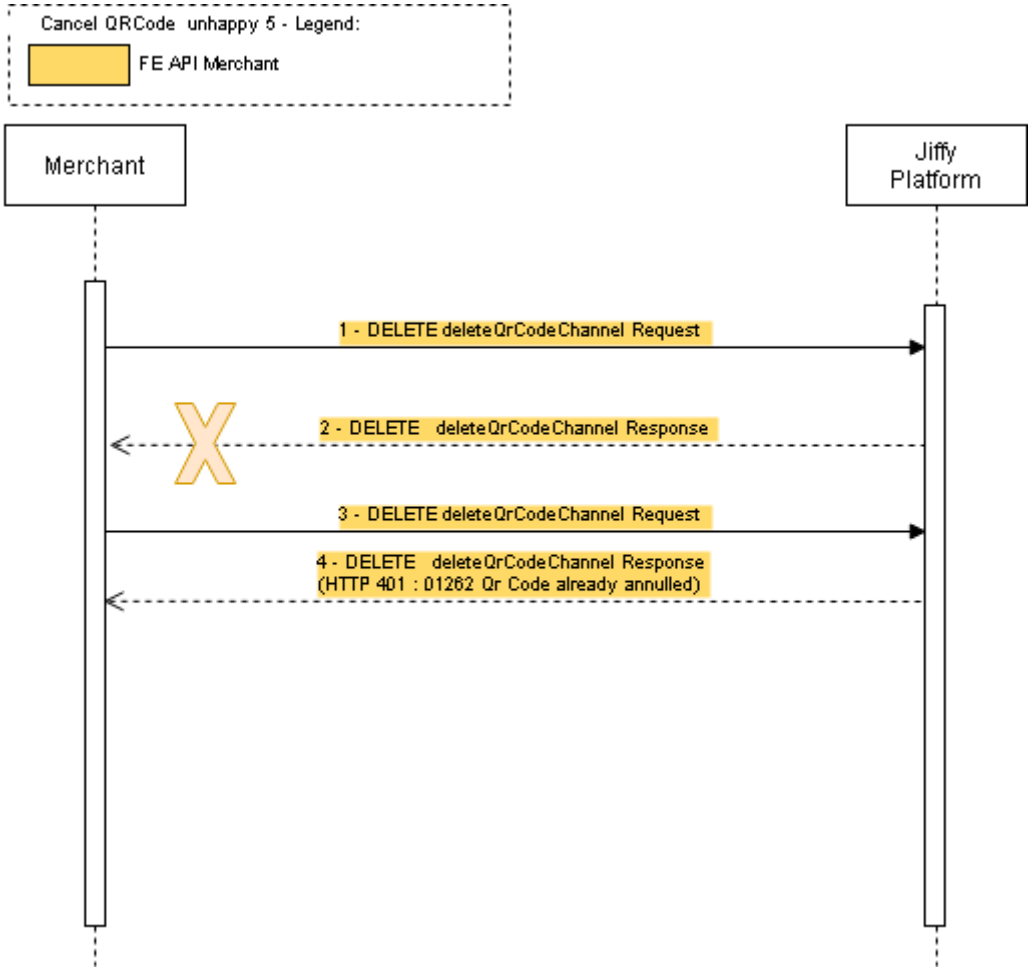
2.2.8. QrCode Delete: UAEIPP Overlay Service can't deliver a response to the merchant while the QrCode was already scanned by the buyer



2.2.8.1. Sequence diagram description

1. For some business reason Merchant decides to cancel the payment then invokes DELETE deleteQrCodeChannel, while the buyer already has scanned it.
2. UAEIPP Overlay Service successfully executes Refund Verify Flow and Refund Confirm flow with banks.
3. Merchant does not receive any response to request at point 1 from the platform due to technical issues. Merchant's request fails due to timeout.
4. At any point in time Merchant retries the DELETE operation.
5. As the previous deleteQrCodeChannel was successful, as per description at point 2, UAEIPP Overlay Service replies "QrCode is already canceled".

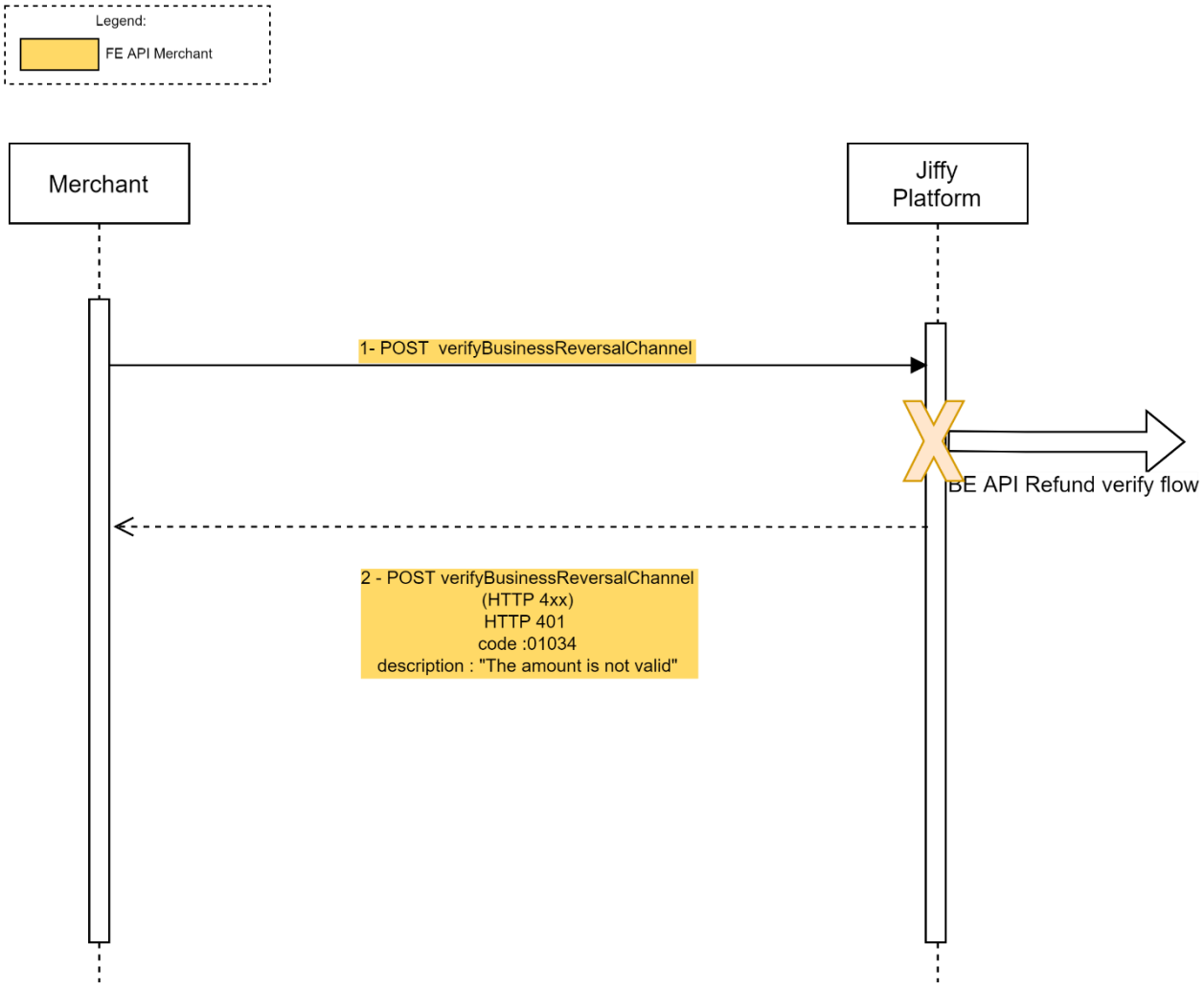
2.2.9. QrCode Delete: UAEIPP Overlay Service can't deliver a response to the merchant



2.2.9.1. **Sequence diagram description**

1. For some business reason Merchant decides to cancel the QrCode then invokes DELETE deleteQrCodeChannel, while the buyer hasn't scanned it yet.
2. Merchant does not receive any response to request at point 1 from UAEIPP Overlay Service due to technical issues. Merchant's request fails due to timeout.
3. At any point in time Merchant retries the DELETE operation.
4. UAEIPP Overlay Service replies "QrCode is already canceled".

2.2.10. Reversal: UAEIPP Overlay Service verify flows with banks is unsuccessful



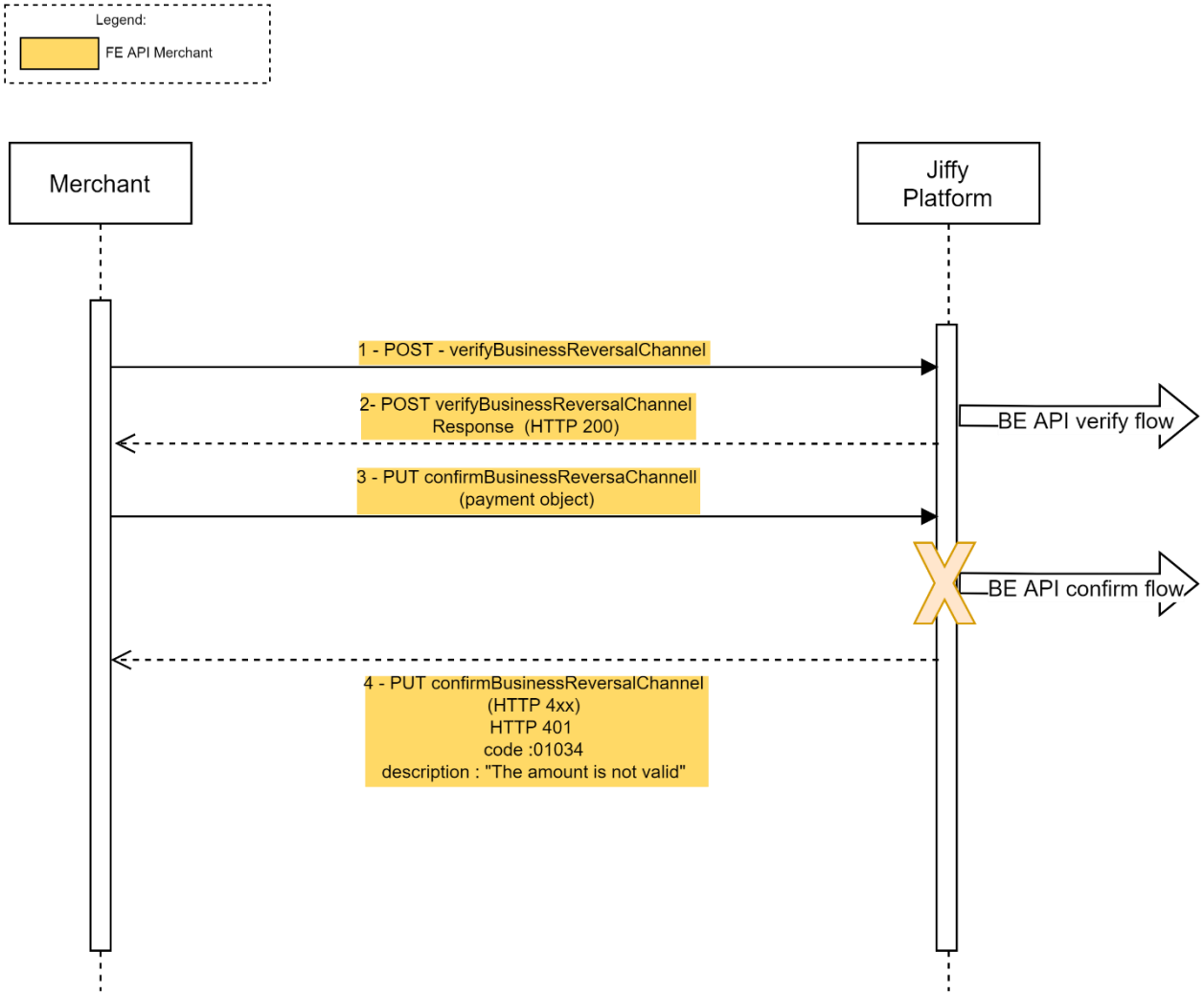
2.2.10.1. **Sequence diagram description**

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant decides to start the reversal process.

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Reversal, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes Post verifyBusinessReversalChannel to verify that a payment exists on the platform and has been previously approved. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
2. UAEIPP Overlay Service unsuccessfully runs Refund verify flows with participants (both Merchant's and Buyer's one) then replies to unsuccessful response to Merchant.

2.2.11. Reversal: UAEIPP Overlay Service Reversal Confirm flows with banks is unsuccessful



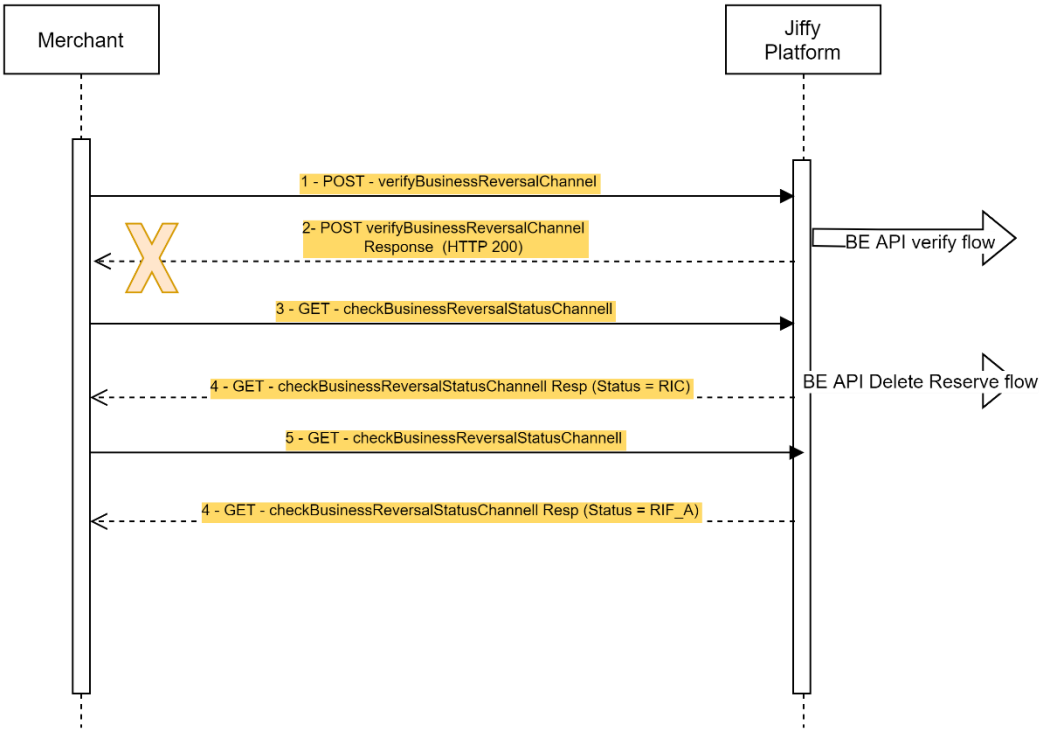
2.2.11.1. **Sequence diagram description**

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant must reverse the payment to the buyer.

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Reversal, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes Post verifyBusinessReversalChannel to verify that a payment exists on the platform and has been previously approved. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
2. UAEIPP Overlay Service runs Refund verify flows with participants(both Merchant's and Buyer's one) then replies to successful response.
3. Merchant invokes PUT confirmBusinessReversalChannel to confirm the reversal from the previously verified approved payment. Refer to the *UAEIPP Overlay Service Interface Specifications - Back-End APIs for Participants integration* document for details on APIs to be exposed by the Participants for Payment Initiation.
4. UAEIPP Overlay Service unsuccessfully runs Refund confirm flows with participants (both Merchant's and Buyer's one) then replies to unsuccessful response to Merchant.

2.2.12. Reversal: UAEIPP Overlay Service can't send verifyReversal response to the merchant



2.2.12.1. Sequence diagram description

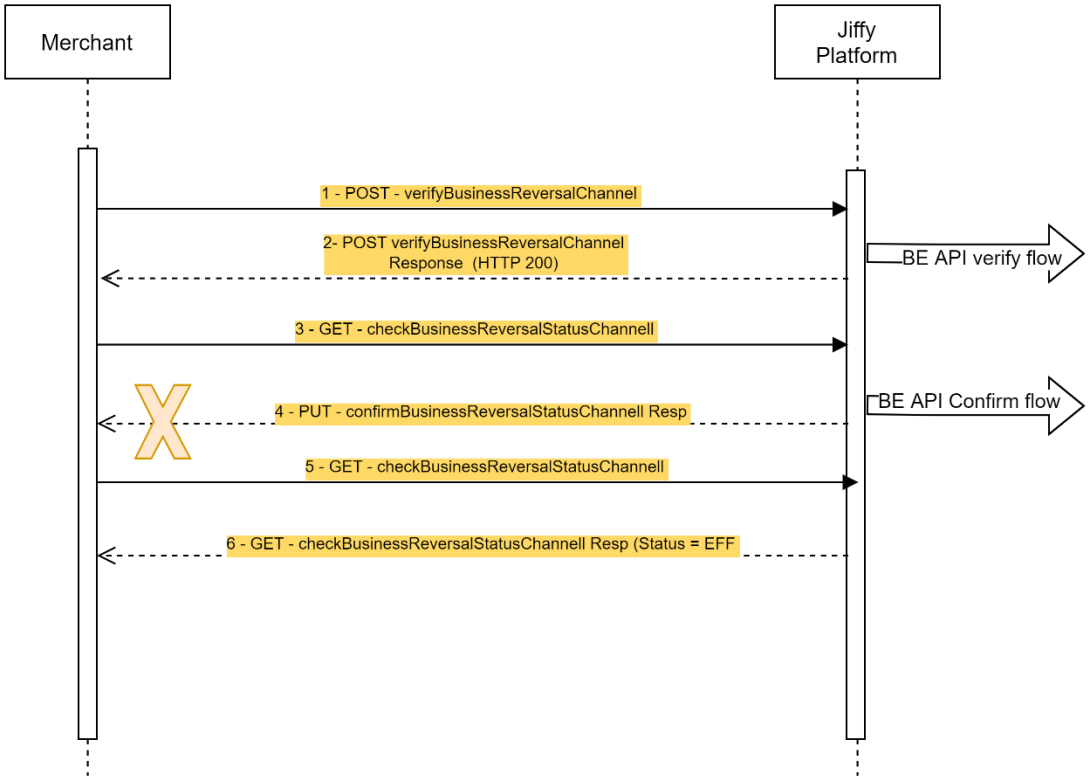
Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The buyer must have activated the APP (White Label or Mobile Bank APP). The merchant decides to start the reversal process.

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Reversal, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes Post verifyBusinessReversalChannel to verify that a payment exists on the platform and has been previously approved.
2. UAEIPP Overlay Service runs Refund verify flows with participant (Merchant's) then receives a successful response. For a technical reason (example: timeout) Merchant does not receive any response from UAEIPP Overlay Service.

3. Merchant invokes GET checkBusinessReversalStatusChannel to check what has happened.
4. UAEIPP Overlay Service returns response with attribute status= “RIC”. This means that UAEIPP Overlay Service has received verifyBusinessReversalChannel request from Merchant and successfully executed Refund Verify flow with participants and is waiting Merchant to confirm reversal operation through PUT confirmBusinessReversalChannel, but Merchant actually can’t invoke confirmBusinessReversalChannel operation because Merchant did not receive any response at point 2 containing “paymentId” (unique identifier of this reversal operation) to be mandatorily provided in confirmBusinessReversalChannel request .
5. UAEIPP Overlay Service executes delete reserve flow with Merchant’s Participant after verify-reserve expiration time.
6. If Merchant tries again with GET checkBusinessReversalStatusChannel, after delete reservation. UAEIPP Overlay Service returns response with attribute status= “RIF_A”. This means that the operation has not been processed by UAEIPP Overlay Service due to technical errors. Merchant will not be debited and have to repeat the reversal operation from the beginning.

2.2.13. Reversal: UAEIPP Overlay Service can’t send confirmReversal response to the merchant



2.2.13.1. Sequence diagram description

1. Merchant invokes Post verifyBusinessReversalChannel to verify that a payment exists on the platform and has been previously approved.
2. UAEIPP Overlay Service runs Refund verify flows with Merchant’s participant then replies to successful response.

3. Merchant invokes confirmBusinessReversalChannel providing paymentId received at point 2 to confirm and trigger UAEIPP Overlay Service to orchestrate refund confirm flow between merchant's participant and buyer's participant.
4. Orchestration ends successfully but for a technical reason the response cannot be communicated to the Merchant
5. Merchant invokes GET checkBusinessReversalStatusChannel to check what has happened.
6. UAEIPP Overlay Service returns response with attribute status
    a. "EFF": this means that UAEIPP Overlay Service received and processed confirmBusinessReversalChannel request from Merchant, executing successfully Refund confirm flow with participants. Buyer will be refunded.

       OR

    b. "DA_STR": this means that UAEIPP Overlay Service received and processed confirmBusinessReversalChannel request from Merchant, but Refund confirm flow with participants is still running (and is going to end successfully within minutes, depending on platform workload). Buyer will be refunded.
        i. Merchant executes again GET checkBusinessReversalStatusChannel at any point in time to check that status of the reversal will change to "EFF" , to be sure that refund confirm flow with participants has successfully ended.

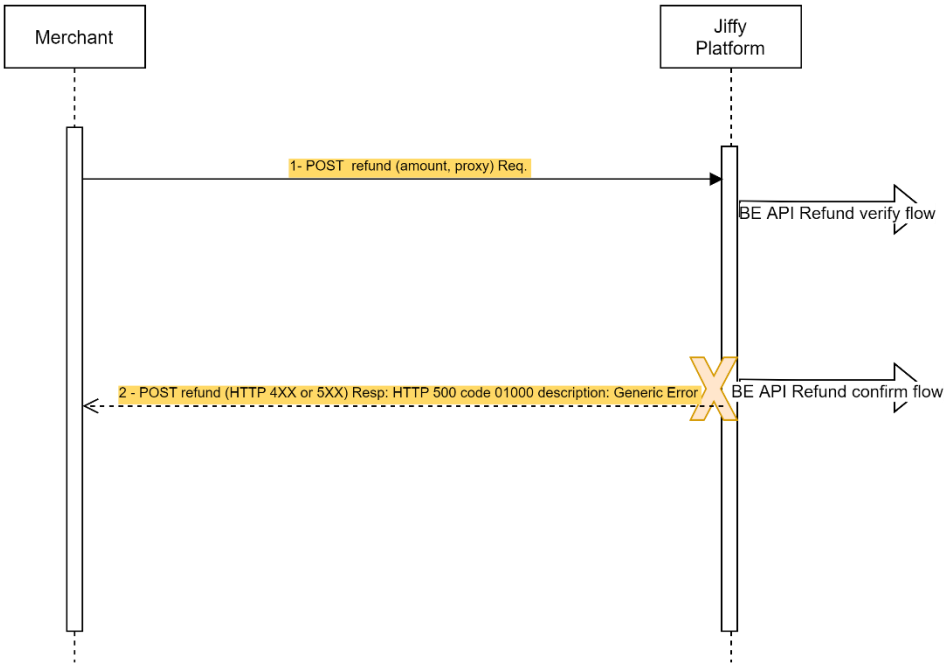
Sequence description above assumes that Refund confirm flow with participants will be successful, but in case this flow with participants is not successful sequence above may change from point 5 as follows:

5. Merchant invokes GET checkBusinessReversalStatusChannel to check what has happened.
6. UAEIPP Overlay Service returns response with attribute status
    a. "ERR": this means that, verify Flow with participants ended successfully then UAEIPP Overlay Service received and processed checkBusinessReversalStatusChannel request from Merchant, executing not successfully Refund confirm flow with participants. Buyer will not be refunded.

       OR

    b. "RIC": this means that, verify Flow with participants ended successfully then UAEIPP Overlay Service received and processed confirmBusinessReversalChannel request from Merchant, but Refund confirm flow with participants is still running .
       Merchant may execute GET checkBusinessReversalStatusChannel at any point in time to check that status of the reversal will change to "ERR".

2.2.14. Refund: UAEIPP Overlay Service Refund flows with banks is unsuccessful



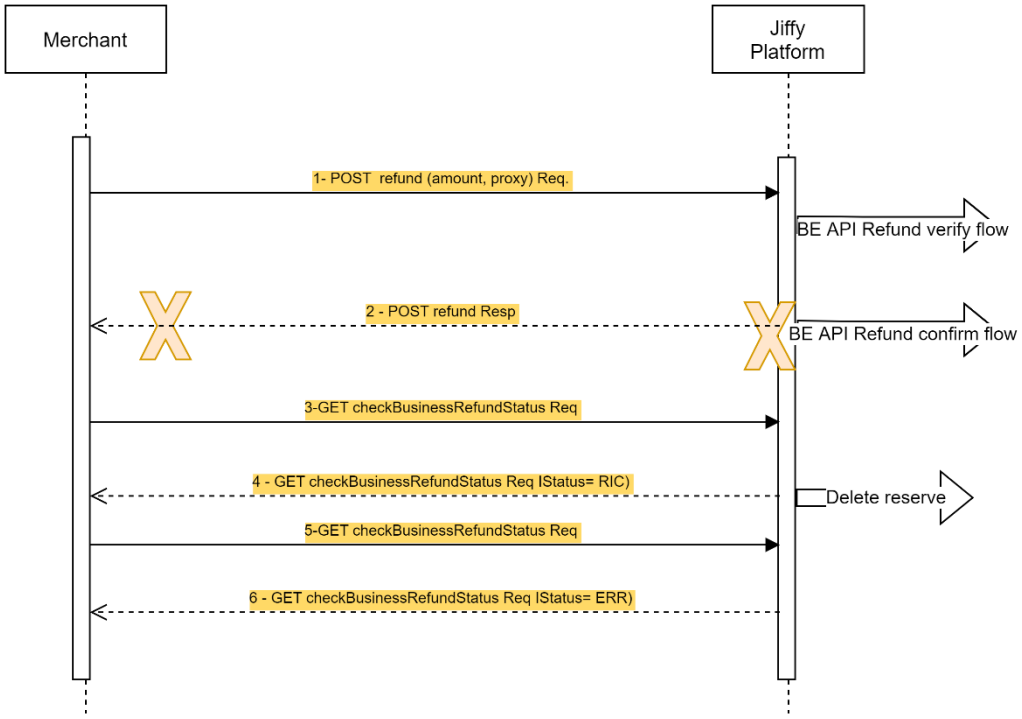
2.2.14.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The merchant wants to refund the customer, there is no need of any previous transaction information (different from the Reversal flow). The refund cannot be completed.

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Refund, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's mobile number, without entering any reference to the original payment of the buyer.
2. The orchestration of the refund transaction ends unsuccessfully due to a technical problem.
3. Merchant may execute GET checkBusinessRefundStatus at any point in time to check that status of the refund is "ERR".

2.2.15. Refund:UAEIPP Overlay Service Refund Response never arrives after a successful verify step



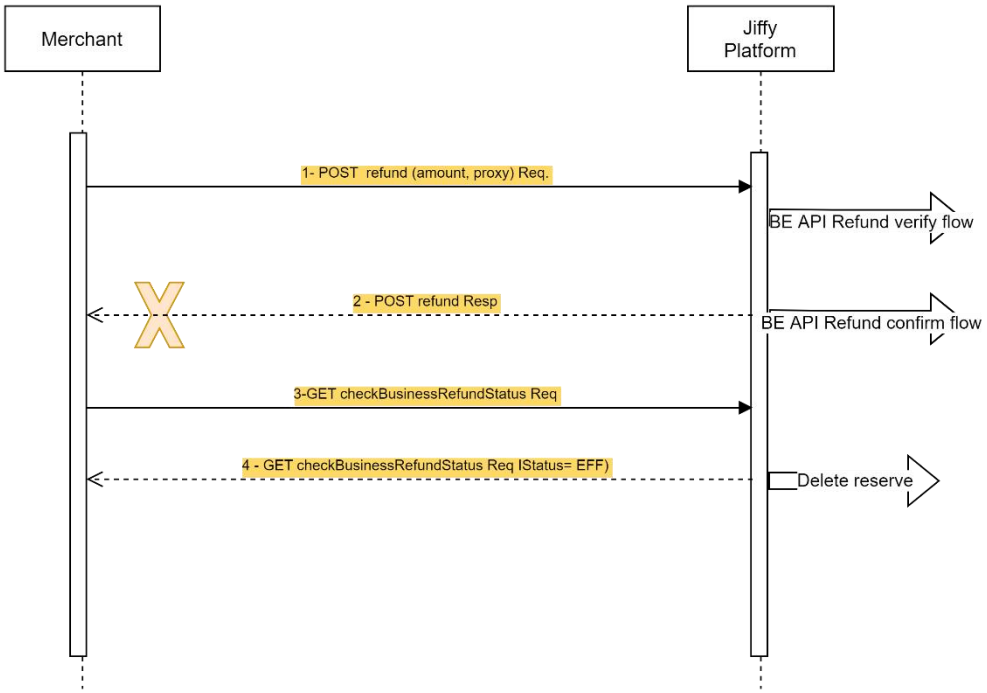
2.2.15.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The merchant wants to refund the customer, there is no need of any previous transaction information (different from the Reversal flow). The refund cannot be completed.

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Refund, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's proxy without entering any reference to the original payment of the buyer.
2. After a successful verify step, there is something wrong with the confirm step and the response of the refund API never arrives. For BE API, refer to Participants' Interface Specification Document.
3. In order to know what has happened with the refund, Merchant invokes checkBusinessRefundStatus.
4. UAEIPP Overlay Service returns response with attribute status= "RIC" that means that the verify step was successfully processed.
5. After fund reserve expiration time passed, UAEIPP Overlay Service triggers delete reservation flow with Merchant's participant.
6. After successful execution of delete reservation flow, when Merchant invokes checkBusinessRefundStatus, response outlines that refund status is ERR (Participant error) as now it is clear that confirmation flow wasn't successfully processed, and Marchant needs to repeat the refund operation from the beginning.

2.2.16. Refund: UAEIPP Overlay Service Refund Response never arrives (timeout)



2.2.16.1. Sequence diagram description

Pre-requisite: both the merchant and the buyer must be enrolled in UAEIPP Overlay Service (refer to the UAEIPP Overlay Service Interface Specifications - Front-End APIs for setup and maintenance document). The merchant wants to refund the customer, there is no need of any previous transaction information (different from the Reversal flow). The refund cannot be completed

The merchant must be active on one Channel. In order to be active, the merchant must have already concluded the SCA. It is not mandatory to re-perform the SCA every time the merchant wants to start the Refund, (e.g. it is not needed, if the refresh token provided during the SCA is still valid). For details, please refer to UAEIPP Overlay Serv Interf Specs - SCA Process

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's proxy, without entering any reference to the original payment of the buyer.
2. After a successful verify and confirm step, the response of the refund API never arrives.
3. In order to know what has happened with the refund, Merchant invokes checkBusinessRefundStatus.
4. UAEIPP Overlay Service returns response with attribute status= "EFF" that means that the refund was successfully executed.

2.3. Endpoints

Operation	Endpoint	Method	Purpose of the API
registerQrCodeChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/qr-code	POST	This API allows registering a dynamic QRCode to request a payment. The QRCode will have a limited temporal validity"
checkStatusQrCodeChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/qr-code/status	GET	The API is used by the merchant to check the status of the payment triggered by an input QR Code.

deleteQrCodeChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tags/{merchantTag}/qr-code	DELETE	"This API allows deleting a dynamic QR Code generated by the API registerQrCode. If the payment had been already concluded, it starts a reversal payment operation
verifyBusinessReversalChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/business/reversal	POST	This API allows verifying the reversal for any P2B payment. It returns the reversal identifier, for a single P2B only a single reversal payment can be made.
confirmBusinessReversalChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/business/reversal	PUT	This API allows confirming a previous verified reversal payment. All the data in the request must be consistent with the verifybusinessreversalchannel step"
finalizePaymentChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/ tag/{merchantTag}/finalize/payment	PUT	This API allows finalizing a preauthorized payment. Merchant can update the amount, compared to the one provided in the original transaction
refund	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund	POST	This API allows a merchant to make a refund to a certain buyer. The API is also available for disbursement operations
checkBusinessReversalStatusChannel	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/business/reversal/status	GET	This API allows checking the reversal execution status .
checkBusinessRefundStatus	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund/status	GET	This API allows checking the execution status of a requested refund.

As general rule, if in Response to the Front End API the Participant receives unexpected fields, the same can be ignored.

2.4. Common Fields

In this paragraph we describe all the common attributes used in every call. They are applied to all the endpoints.

HTTP version supported 1.1

2.4.1. Request

2.4.1.1. Path Parameter

Name	Type	Sample Value	Req.	Scope	Description
groupCode	String Max 5	99999	Y	Functional	Parent participant, Technical Service Provider or Technical Service Provider as Acquirer code of the active merchant. In case the caller of the API is a Technical Service Provider or Technical Service Provider as Acquirer all these 3 headers must be also used: 1. providerType 2. participantGroupCode 3. participantBankCode

bankCode	String Max 5	99999	Y	Functional	Sub participant, Technical Service Provider or Technical Service Provider as Acquirer code of the active merchant. In case the caller of the API is a Technical Service Provider or Technical Service Provider as Acquirer all these 3 headers must be also used: 1. providerType 2. participantGroupCode 3. participantBankCode

2.4.1.2. **Request Header**

Name	Type	Sample Value	Req.	Scope	Description
X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique Client RequestID. It must present a constraint that allows distinguishing the calling institution.
language	String 2	EN	N	Functional	Code that defines which language to use for result messages. Standard used is the ISO 639-1 2 letter codes. Example values: “IT” “EN” “DE”.
timestamp	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}\+\d{2}:\d{2}	2023-09-22T23:50:56.193+01:00	N	Security	Execution date/time of the call to the service. The timeZone in this field can be different of the value included in the field timeZone.
Authorization	String	Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJtZV9UMk9XTmFUZ0JHUk4tVmZ3R3RWcnNBMHUyR2VFc0Z2OEFqNkctcVZ3In0…	Y	Security	Access bearer token
x-jws-signature	String	detached signature	Y	Security	Describe the digital signature or message authentication code (MAC) applied to the Payload.
Content-Type	String Max 16	application/json	N	Functional	Entity header is used to indicate the media type of the resource.
JWT	String	JWT	N	Security	Note: the field should not be implemented for R1 as all the merchants will be connected

					through a Participant Channel
Additional-customer-authorization	String	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IlhGRDVxbXdYeVB5bTk1dGFGR0szIn0.eyJpc3MiOiJodHRwczovL2p3dC1pZHAuamlmZnkuc2lhLmV1IiwianRpIjoidG52NXpCMFdiM2FMQ2dYa2ZFTEUiLCJoY2kiOiIyNzQwQUQ3MDMzM0FBMjY3M0YyOURDMThGNTFBMjdGMEVFMjM3QTRBMDQwREE5MDgyMjdCNTkwNkU0NTI1RkVGIiwiZXhwIjoxNjAwMjYzMTA2ODM4LCJvcG4iOiJQMlAiLCJhbXQiOiI1MDAwIiwiY2N5IjoiRVVSIiwiYWlkIjoiODcyMzM4NDcxNSIsInNpYiI6IklFNjCT0ZJOTAwMDE3OTI4NjUxIiwicmliIjoiSUxMEJPRkk5MDAwMTc3MjU4ODciLCJzcG4iOiIrMzkzMzUxMjEyMTIzIiwicnBuIjoiKzM5MzM0NzQ4Nzg3NiIsIm9zZCI6IklPUyIsIm9zdiI6IjE0IiwibXNnIjoiWW91IGFyZSBzZW5kaW5nIDUwLjAwIEVVUiB0byBNYXIqKiogUG9sKioqIHdpdGggeW91ciBhY2NvdW50ICoqKio4NjUxIn0.s4yL6NTf3AbduAcYdYf3OI2F43orNwWibRPjLKpquhI	N	Security	Additional optional authorization data generated by Merchant's clients (to authorize refund/reversal/disbursement) and forwarded by UAEIPP Overlay Service to Participant Participants. This field is structured as a AJWT
providerType	String	ACQUIRER	C	Functional	Code that defines the caller Technical Service Provider Type. Expected value: 1. ACQUIRER 2. PROVIDER This field has to be populated with "ACQUIRER" whether the API is called by a Technical Service Provider as Acquirer and with "PROVIDER" in case it is called by a Technical Service Provider (see definition of the actors in the paragraph 1.2.1)
participantGroupCode	String Max 5	99999	C	Functional	Parent bank code of the participant. This field has to be populated with the Participant group code if the API is called by a Technical Service Provider as acquirer or Technical Service Provider (see Appendix paragraph 3.15.9). In case of Participants calling the API, the field is not expected.
participantBankCode	String Max 5	99999	C	Functional	Sub bank code of the participant. This field has to be populated with the Participant bak code if the API is called by a Technical Service Provider as acquirer or Technical Service

					Provider (see Appendix paragraph 3.15.9). In case of Participants calling the API, the field is not expected.

2.4.2. Response

2.4.2.1. Common Response Header

Name	Type	Sample Value	Req.	Scope	Description
x-jws-signature	String	detached signature	Y	Security	Describe the digital signature or message authentication code (MAC) applied to the Payload.
Content-Type	String Max 16	application/json	N	Functional	Entity header is used to indicate the media type of the resource.
X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

2.4.2.2. Common Response Body

Name	Type	Sample Value	Req.	Scope	Description
result	Object	NA	Y	Functional	This object contains information on the result of the execution of the service.
result.code	String Max 5	00000	Y	Functional	Result code.
result.result	Boolean	True	Y	Functional	Result of the request (true=positive, false=negative).
result.message	String Max 100	Positive result	Y	Functional	Represents the description of the result.
result.X-Request-ID	String	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the

	Max 100				same value as the result - X-Request-ID present in the respective request.

2.5. Error handling

2.5.1. Common response header

Name	Type	Sample Value	Req.	Scope	Description
x-jws-signature	String	detached signature	Y	Security	Describe the digital signature or message authentication code (MAC) applied to the Payload.
Content-Type	String Max 16	application/json	N	Functional	Entity header is used to indicate the media type of the resource.
X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

2.5.2. Common response body

Name	Type	Sample Value	Req.	Scope	Description
result	Object	NA	Y	Functional	This object contains information on the result of the execution of the service.
result.code	String Max 5	00000	Y	Functional	Result code.
result.result	Boolean (True/False)	True	Y	Functional	Result of the request. Expected values: 1. True 2. False
result.message	String Max 150	Positive result	Y	Functional	Represents the description of the result.
result.X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

2.6. [POST] Register QR Code

This chapter describes the additional fields available in the endpoint related to the “registerQrCodeChannel” service; whereas common fields are applied to all the endpoints.

2.6.1. Description

This API allows registering a dynamic QR Code with a limited temporal validity, this QR Code can be used by a customer to start a payment. QRcodes created with this operation last 5 minutes (see appendix for further details).

2.6.2. Business Scenario

The merchant creates the dynamic QR Code to let the buyer make the payment. The QrCode can be generated from Physical or Online Shop.

2.6.2.1. Direct outcome

- Dynamic QR code created and ready to be shared with the buyer with the aim to perform the payment.

2.6.2.2. Related outcome

N.A

2.6.3. URL

POST
/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/qr-code

2.6.4. Operation

registerQrCodeChannel

2.6.5. Request

2.6.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields

2.6.5.2. Request Header

For the request headers, refer to the common ones described in the paragraph Common Fields.

2.6.5.3. Request Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	This object contains the data about the payment to be executed
payment.amount	Number Max 9 (Decimal part: 2)	4.50	Y	Payment amount
payment.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
payment.reason	String Max 140	Coffee x4	N	Payment reason
payment.shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	Y	Identification code of the cash desk of the merchant's shop.
paymentCategory	String Max 15	01	N	Payment Category of the payment: Expected values: 1. "01" = Bill payment 2. "02" = Prepaid top-up 3. "03" = Purchase Note: this field can be provided or not. If provided, the same must be reported in the EMV

				string. Please refer to the dedicated paragraph.
paymentType	String Max 15	PAG	Y	Category of the payment: Expected values: 1. "PAG" = Standard payment 2. "PRE" = Preauthorized payment in Case paymentCategory is 01 "Bill payment" MUST be set to PAG
qrCodeTransactionId	String Max 50	ABCD09876	Y	Transaction identifier for this operation created by the Merchant.
payment.categoryPurpose	String Max 35	CCP	Y	Category Purpose Code in a proprietary form. For the list of allowed values, please refer to the dedicated document From 2024 R4

2.6.6. Response

2.6.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.6.6.2. Response Body

Name	Type	Sample Value	Req.	Description
qrCodeId	String Max 16	z4E0k	Y	Qr code identifier
link	String Max 50	https://p.bcmt.en/?	Y	QR Code's link. This field must be not considered as the platform is generating an EMV standard Qrcode. Please refer to Par: "Generate UAEIPP Overlay Service valid dynamic QRCode"

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.6.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01027	Merchant not registered	FALSE	401
01034	The amount is not valid	FALSE	401

01037	Transaction id already used	FALSE	401
01076	Bank account not found	FALSE	401
01146	Payment reason not valid	FALSE	401
01281	Currency mismatch, payment currency: {0}, bank account currency: {1}	FALSE	400
01300	Merchant not found	FALSE	401
01301	Shop not found	FALSE	401
01302	Cash Desk not found	FALSE	401
01346	Currency mismatch, bank account currency: {0}, payment currency: {1}	FALSE	400
01365	Bank account blocked; it is impossible to proceed with the request	FALSE	401
01371	Merchant not authorized	FALSE	401
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
02002	The field {0} is not filled in or has an invalid format	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
02005	The field {0} doesn't match any of the expected values	FALSE	400
03004	Merchant not enabled for payment method:{0}	FALSE	400
03016	The operation is not allowed	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

2.7. [PUT] Finalize Payment Channel

This chapter describes the additional fields available in the endpoint related to the “finalizePaymentChannel ” service, whereas common fields are applied to all the endpoints.

2.7.1. Description

This API allows the merchant customer to finalize a pre-authorized payment - requested by Merchant with registerQrCodeChannel operation having “paymentType” set to “PRE” and confirmed by buyer via White Label App - with the updated value of the amount, once the items/services have been delivered to the buyer.

2.7.2. Business Scenario

The merchant customer needs to finalize a pre-authorized payment by confirming it. Through this service, the merchant customer can also refuse the payment by setting the “confirm” query parameter to “false”.

2.7.2.1. Direct outcome

- Merchant customer finalizes a pre-authorized payment.

2.7.2.2. Related outcome

N.A

2.7.3. URL

PUT

/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/finalize/payment

2.7.4. Operation

finalizePaymentChannel

2.7.5. Request

2.7.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identification code .
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant Customer ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

2.7.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
confirm	Boolean	true	Y	Indicates if the merchant customer confirms or refuses the payment. Expected values: “true” = confirm the payment “false” = refuse the payment

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.7.5.3. Header

For the request headers, refer to the common ones described in the paragraph Common Fields.

2.7.5.4. Request Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	123456	Y	Internal identification code of the payment generated by the system, returned by the verifyPayment API. The paymentid can be retrieved in the CheckQrCodestatus API
payment.amount	Number Max 9 (Decimal part: 2)	70	Y	Updated amount of the payment after the items/services have been delivered to the buyer.
payment.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
payment.reason	String Max 140	Soccer shoes	N	Payment reason.
payment.shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	N	Identification code of the cash desk of the shop.

2.7.6. Response

2.7.6.1. Response Header

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.7.6.2. Response Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	123456	E	Internal identification code of the payment generated by the system.
payment.amount	Number Max 9 (Decimal part: 2)	70	E	Updated amount of the payment after the items/services have been delivered to the buyer.
payment.currency	String Max 3	AED	Y	Currency of the payment's amount. ISO 4217

payment.fees	Number	0	N	Transaction fees, additional charge of the service expressed in euro. Its value is set to zero by default. As it is reserved for future use, it cannot be not implemented as the value will be always 0.
payment.totalAmt	Number Max 9 (Decimal part: 2)	70	N	Total fees and amount. As it is reserved for future use, it cannot be implemented. The value will be always the same as payment.amount.
payment.reason	String Max 140	Soccer shoes	E	Payment reason
payment.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	E	Identification code of the cash desk of the shop.

2.7.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
00023	Payment taken in charge, check the correct conclusion	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01030	Communication Problems with Gateway	FALSE	401
01034	The amount is not valid	FALSE	401
01041	Invalid QR code	FALSE	401
01047	The payment to be finalized does not match the payment initiated	FALSE	401
01054	Payment refused, try again	FALSE	401
01061	The operation is not allowed	FALSE	401
01067	No payments found	FALSE	401
01076	Bank account not found	FALSE	401
01146	Payment reason not valid	FALSE	401
01300	Merchant not found	FALSE	401
01301	Shop not found	FALSE	401
01302	Cash Desk not found	FALSE	401
01367	Preauthorization expired.	FALSE	400

01371	Merchant not authorized	FALSE	401
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
03004	Merchant not enabled for payment method:{0}	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400

2.8. [GET] Check QR Code Status

This chapter describes the additional fields available in the endpoint related to the "checkStatusQrCodeChannel" service; whereas common fields are applied to all the endpoints.

2.8.1. Description

The API is used by the merchant customer to check the status of the payment triggered starting from registerQrCodeChannel operation successful execution.

Merchant can invoke this method in polling to retrieve updates on the status of the payment in order to know the successful execution on the payment or if the Qr Code is still valid. For details on the flow, please refer to the Qr Code Payments sequence diagrams.

Polling frequency suggested is one invocation every 5 seconds.

2.8.2. Business Scenario

The Merchant Customer needs to know the status of a newly generated QR Code.

2.8.2.1. Direct outcome

- QR code status information provided.

2.8.2.2. Related outcome

N.A

2.8.3. URL

GET
/inquiry-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/qr-code/status

2.8.4. Operation

checkStatusQrCodeChannel

2.8.5. Request

2.8.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.8.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
qrCodeId	String Max 16	z4E0k	N	QR code identification code

2.8.5.3. Request Header

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.8.6. Response

2.8.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.8.6.2. Response Body

Name	Type	Sample Value	Req.	Description
QrCodeRequestStatus	String	EFF	Y	Status of the qr Code. Expected values: • "**EFF**" - Request paid by the buyer; • "**ATT**" - Request in pending status to be paid by the buyer; • "**EXP**" - Request expired; • "**ANN**" - QrCode canceled by Merchant
paymentId	Number Max 16	224466	C	Identification code of the payment generated by UAEIPP Overlay Service, once the payment flows with the participants are triggered and ended successfully. This will be populated only when status is "EFF"
lastUpdateDate	Date	2020-08-03T17:45:20.000+00:00	C	Date in which RTP shifted to "EFF" status.
paymentIdSct	string	0003P62000030934171	C	This value is the unique reference for the transaction. It is returned only if the Status is "EFF"

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.8.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
00043	QR Code annulled	TRUE	200
00044	QR Code expired	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01027	Merchant not registered	FALSE	401
01251	QR code not found	FALSE	401
01371	Merchant not authorized	FALSE	401

02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

2.9. [DELETE] Delete QR Code

This chapter describes the additional fields available in the endpoint related to the “deleteQrCodeChannel” service; whereas common fields are applied to all the endpoints.

2.9.1. Description

This API allows deleting a dynamic QR Code generated by the API "registerQrCodeChannel", and, if the payment is already concluded (status **Executed**), it starts a reversal payment operation[1].

2.9.2. Business Scenario

The Merchant customer decides to delete the dynamic QR Code.

2.9.2.1. Direct outcome

- Dynamic QR code deleted.

2.9.2.2. Related outcome

N.A

2.9.3. URL

DELETE
/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/qr-code

2.9.4. Operation

deleteQrCodeChannel

2.9.5. Request

2.9.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.9.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
qrCodeId	String Max 16	z4E0k	Y	QR code identification code.
qrCodeTransactionId	String Max 50	ABCD09876	Y	Transaction identifier for this operation created by the Merchant .identification

[1] When Jiffy starts a reversal invokes Merchant’s participant services to initiate a SCT from merchant’s account to buyer’s account.

				code , provided in registerQrCodeChannel operation

2.9.5.3. Request Header

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.9.6. Response

2.9.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.9.6.2. Response Body

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all reverted payment data. The platform returns this response body to DELETE QrCode only in case the API has been invoked after buyer successfully executed the payment.
payment.paymentId	Number Max 16	123456	C	Internal identification code of the payment generated by the system. This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet)
payment.amount	Number Max 9 (Decimal part: 2)	70	C	Updated amount of the payment after the items/services have been delivered to the buyer. This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet
payment.currency	String Max 3	AED	C	Currency of the payment's amount. ISO 4217 This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet
payment.fees	Number	0	C	Transaction fees, additional charge of the service expressed in euro. Its value is set to zero by default. As it is reserved for future use, it can be not expected, the value will be always 0. This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet
payment.totalAmt	Number Max 9 (Decimal part: 2)	70	C	Total fees and amount. As it is reserved for future use, it can be not expected, the value will be always the same as payment.amount. This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet
payment.reason	String Max 140	Soccer shoes	C	Payment reason (if present in the original request). This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet
payment.shopId	Number Min 5 Max 16	10001	C	Identification code of the shop issued by the platform. This field is not provided in response if the payment is still not processed (e.g. the payer has not paid yet
payment.cashDeskId	Number Min 8 Max 16	10000001	C	Identification code of the cash desk of the shop. This field is not provided in response if

				the payment is still not processed (e.g. the payer has not paid yet

2.9.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
00046	QrCode annulled and all related payments have been reversed	TRUE	200
00047	QrCode annulled but not all related payments have been reversed	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01251	QR code not found	FALSE	401
01259	Transaction id already used for the channel	FALSE	400
01261	A static QrCode cannot be annulled	FALSE	400
01262	QR code already annulled	FALSE	401
01263	The QrCode cannot be annulled	FALSE	401
01277	Cannot annull qrCode now, retry later	FALSE	400
01300	Merchant not found	FALSE	401
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

2.10. [POST] Verify Reversal

This chapter describes the additional fields available in the endpoint related to the "verifyBusinessReversalChannel" service; whereas common fields are applied to all the endpoints.

2.10.1. Description

This API allows starting a reversal for a previous P2B payment received. For any P2B payment only a single reversal payment can be made.

In case of positive result, the API returns the paymentId of the reversal representing an instance of the reversal that may be confirmed through the dedicated API (confirmBusinessReversalChannel) to give the amount of the previously confirmed payment to the buyer.

The reversed amount can be equal or lower than the original payment amount (it cannot be greater).

2.10.2. Business Scenario

The Merchant Customer needs to start a reversal transaction to a buyer.

2.10.2.1. Direct outcome

- The checks on the reversal are completed and therefore the reversal itself is ready to be confirmed.

2.10.2.2. Related outcome

N.A

2.10.3. URL

POST
/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/business/reversal

2.10.4. Operation

verifyBusinessReversalChannel

2.10.5. Request

2.10.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.10.5.2. Request Header

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Scope	Description
appId	String 64	03704576A8D81E96BF0C5D8E8953F4298842EC7A1803884A3EEC96C11C290099	Y	Security	Unique identification code generated by the Participant's Mobile Server related to the APP installed by the merchant. This field must be populated with the appId of the channel from which the merchant is operating
deviceOSVersion	String Max 255	Ios 10.5.26	N	Security	Operating system of the device.
deviceModel	String Max 255	ABCD1234	N	Security	Code associated with the device model.
deviceId	String Max 255	ABCD1234	N	Security	Unique identification code of the device.
deviceIpAddress	String Max 255	192.168.1.1	N	Security	IP address associated with the device.
country	String 2	AE	N	Security	Sender Country Code defined according to the standard ISO-3166 Alpha2.
timeZone	String PATTERN: +hh:mm or -hh:mm	+01:00	N	Security	Time zone on sender's side registered by the device. This Timezone can be different of the Timezone included in Timestamp field.

2.10.5.3. Request Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	This object contains the data about the payment where the reversal needs to be executed
payment.amount	Number Max 9 (Decimal part: 2)	4.50	Y	Payment amount
payment.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
payment.reason	String Max 140	Coffee x4	N	Payment reason
payment.paymentRefId	Number Max 16	224455	Y	Identification code of the payment to reverse.
payment.shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by the platform.

payment.cashDeskId	Number Min 8 Max 16	10000001	N	Identification code of the cash desk of the merchant's shop.
merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.
merchantTrxRefId	String Max 50	123456	Y	Transaction identifier for the original payment to be refunded generated by the Merchant Customer.

2.10.6. Response

2.10.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.10.6.2. Response Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	This object contains the data about the payment to be executed
payment.paymentId	Number Max 16	224466	C	Identification code of the reversal operation, returned if verify result is successful
payment.amount	Number Max 9 (Decimal part: 2)	4.50	E	Payment amount
payment.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217
payment.fees	Number Max 9 (Decimal part: 2)	0.34	Y	Transaction fees, additional charge of the service expressed in euro.
payment.totalAmt	Number Max 9 (Decimal part: 2)	4.84	Y	Total amount of the transaction: fees + amount.
payment.reason	String Max 140	Coffee x4	E	Payment reason
payment.paymentRefId	Number Max 16	224455	E	Identification code of the payment to reverse.
payment.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	E	Identification code of the cash desk of the merchant's shop.
payment.idSct	String 23	P2B03P20120231046321664	Y	Unique transaction Identifier code, it is also provided in Credit Transfer flow for reconciliation

				purposes. It must be provided in TransactionID in pacs.008. Format: "alias+ x-idempotency-key" The value depends on the transaction type. Original transactionID of the CT previously settled (only for refund flow or implicit reversal flow)
payment.mobile	String Max 30	9,71548E+11	Y	Mobile number of the customer in the UAEIPP Overlay Service, it MUST start with international prefix and plus symbol, in Query Parameters, MUST be encoded as %2B. It can be the mobile number of designated contact for Merchant. Not mandatory for non-professionals. In case of Professional (R2) with App activated , this will trigger invalidate APP-ID In case of Merchant's mobile number; this field can be conditional with the merchantTAG. In Request Body and in case of a not enrolled user the platform will require additional fields. In Response Body it can be obfuscated.
consentId	String Max 128	512b7509-3b7a-4ebb-ba56-91e43d11b80c	Y	An authentication identifier used to represent and track the user's consent in relation to the Strong Customer Authentication (SCA) implemented by the bank. This identifier is included in the JSON Web Token (JWT) that is generated during the authentication process via the mobile app. The consentID is transparently passed from the authentication engine to the server/mobile app and can also be used for non-professional merchants, although its implementation for refunds is not currently provided in the app.
receiverNominative	String Max 100	Marks & Spencer	Y	Name and surname of the creditor. It could be merchant name in case of a P2B or B2B transaction. More details on the masking name rules are

				described in the dedicated paragraph of the Appendix.
receiverIban	String Max 35 Chars pattern: "[a-zA-Z0-9]+"	AE070331234567890569883	Y	International Participant Account Number (IBAN) identifies the payer participant account.
senderIban	String Max 35 Chars pattern: "[a-zA-Z0-9]+"	AE070331234567890569883	Y	International Participant Account Number (IBAN) identifies the payer participant account.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.10.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code	Note
00000	Positive result	TRUE	200	
00023	Payment taken in charge, check the correct conclusion	TRUE	200	
00045	RTP cancelled	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01027	Merchant not registered	FALSE	401	
01030	Communication Problems with Gateway	FALSE	401	
01034	The amount is not valid	FALSE	401	
01041	Invalid QR code	FALSE	401	
01044	Shop not found	FALSE	401	
01047	The payment to be finalized does not match the payment initiated	FALSE	401	
01054	Payment refused, try again	FALSE	401	
01061	The operation is not allowed	FALSE	401	
01067	No payments found	FALSE	401	
01076	Bank account not found	FALSE	401	
01144	The recipient has no receiving bank account set up	FALSE	401	
01146	Payment reason not valid	FALSE	401	
01191	Bank account blocked	FALSE	401	

01228	No RTPs found	FALSE	400	
01255	The reversal is not coherent with the payment	FALSE	400	Mismatch of data between reversal and original payment
01257	The transactionRefId used is not coherent with the transactionId of the payment	FALSE	400	
01259	Transaction id already used for the channel	FALSE	400	
01281	Currency mismatch, payment currency: {0}, bank account currency: {1}	FALSE	400	
01300	Merchant not found	FALSE	401	
01301	Shop not found	FALSE	401	
01302	Cash Desk not found	FALSE	401	
01313	The reversal amount is higher than the one of the payment to which is referred	FALSE	400	
01321	The payment is not reversible	FALSE	400	
01322	The payment was not executed to the selected merchant	FALSE	400	
01323	The payment was not executed to the selected shop	FALSE	400	
01367	Preauthorization expired.	FALSE	400	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03004	Merchant not enabled for payment method:{0}	FALSE	400	
03018	The bank is not enabled	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	

03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	
432	Country not authorised	FALSE	400	From 2024 R2
433	Country information mandatory for the payment	FALSE	400	From 2024 R2

2.11. [PUT] Confirm Reversal

This chapter describes the additional fields available in the endpoint related to the "confirmBusinessReversalChannel" service; whereas common fields are applied to all the endpoints.

2.11.1. Description

This API allows confirming a previous verified reversal payment, all the data in the request must be consistent with the verify step.

2.11.2. URL

PUT
/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/business/reversal

2.11.3. Business Scenario

The Merchant Customer needs to confirm a reversal transaction to a buyer.

2.11.3.1. Direct outcome

- Fund transfer from merchant account to buyer account is requested to merchant's participant.

2.11.3.2. Related outcome

N.A

2.11.4. Operation

confirmBusinessReversalChannel

2.11.5. Request

2.11.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.11.5.2. Request Header

Name	Type	Sample Value	Req.	Scope	Description
appId	String 64	03704576A8D81E96BF0C5D8E8953F4298842EC7A1803884A3EEC96C11C290099	Y	Security	Unique identification code generated by the Participant's Mobile Server related to the APP installed by the merchant. This field must be populated with the appId of the channel from which the merchant is operating
deviceOSVersion	String Max 255	Ios 10.5.26	N	Security	Operating system of the device.
deviceModel	String Max 255	ABCD1234	N	Security	Code associated with the device model.
deviceId	String Max 255	ABCD1234	N	Security	Unique identification code of the device.
deviceIpAddress	String Max 255	192.168.1.1	N	Security	IP address associated with the device.
country	String 2	AE	N	Security	Sender Country Code defined according to the standard ISO-3166 Alpha2.
timeZone	String PATTERN: +hh:mm or -hh:mm	+01:00	N	Security	Time zone on sender's side registered by the device. This Timezone can be different of the Timezone included in Timestamp field.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.11.5.3. Request Body

Name	Type	Sample Value	Req.	Description
merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.
merchantTrxRefId	String Max 50	123456	Y	Transaction identifier for the original payment to be refunded generated by the Merchant Customer.
payment	Object	N.A.	Y	This object contains the data about the payment where the reversal needs to be executed
payment.paymentId	Number Max 16	224466	Y	Identification code of the reversal, returned by the verifyBusinessReversalChannel API.

payment.amount	Number Max 9 (Decimal part: 2)	4.50	Y	Payment amount
payment.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
payment.reason	String Max 140	Coffee x4	N	Payment reason
payment.paymentRefId	Number Max 16	224455	Y	Identification code of the payment to reverse.
payment.shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	N	Identification code of the cash desk of the merchant's shop.

2.11.6. Response

2.11.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.11.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the reversal transaction. Possible values are: 'EFF' = Executed; 'DA_STR' = To be transferred (the request has been accepted by UAEIPP Overlay Service, waiting to be forwarded to the participants); 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Participant error 'RIF_A' = Refused automatically by the platform
payment	Object	N.A.	Y	This object contains the data about the payment to be reversed

payment.paymentId	Number Max 16	224466	E	Identifier of the payment
payment.amount	Number Max 9 (Decimal part: 2)	4.50	E	Payment amount
payment.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217
payment.fees	Number Max 9 (Decimal part: 2)	0.34	Y	Transaction fees, additional charge of the service expressed in euro. As it is reserved for future use, it can be not expected. The value will be always 0.
payment.totalAmt	Number Max 9 (Decimal part: 2)	4.84	Y	Total amount of the transaction: fees + amount. As it is reserved for future use, it can be not expected. The value will be always the same as payment.amount.
payment.reason	String Max 140	Coffee x4	E	Payment reason
payment.paymentRefId	Number Max 16	224455	E	Identification code of the payment to refund.
payment.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	E	Identification code of the cash desk of the merchant's shop.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.11.6.3. **Result Codes**

CODE	MESSAGE	RESULT	HTTP Code	note
00000	Positive result	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01027	Merchant not registered	FALSE	401	

01030	Communication Problems with Gateway	FALSE	401	
01034	The amount is not valid	FALSE	401	
01061	The operation is not allowed	FALSE	401	
01067	No payments found	FALSE	401	
01146	Payment reason not valid	FALSE	401	
01191	Bank account blocked	FALSE	401	
01255	The reversal is not coherent with the payment	FALSE	400	Mismatch of data between reversal and original payment
01258	Transaction id not coherent	FALSE	400	Mismatch of transaction ID
01300	Merchant not found	FALSE	401	
01301	Shop not found	FALSE	401	
01302	Cash Desk not found	FALSE	401	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03004	Merchant not enabled for payment method:{0}	FALSE	400	
03018	The bank is not enabled	FALSE	400	
101	Unknown user	FALSE	401	

102	Sender IBAN does not belong to the bank	FALSE	401	
103	Sender IBAN is blocked	FALSE	401	
104	Insufficient funds in sender account	FALSE	401	
105	Authorization denied by bank	FALSE	401	
106	Invalid request (eg. Incorrect timestamp format, service not available, etc.).	FALSE	400	
107	Rejection due to internal Payer Bank checks (eg. Fraud checks)	FALSE	401	
108	TransactionId missing in SCT initiation request	FALSE	401	
109	Rejection due to internal Payee Bank checks (eg. Fraud checks)	FALSE	401	
110	Bank is not able to provide balance information	FALSE	401	
111	Invalid client certificate presented	FALSE	401	
112	x-jws-signature mismatch	FALSE	401	
113	Bearer token mismatch Client	FALSE	401	
114	Too many request	FALSE	401	
115	Bank has revoked the consent	FALSE	401	

116	Method non allowed	FALSE	401	
117	Request not acceptable	FALSE	401	
118	Unsupported media type	FALSE	401	
119	Sender bank account is closed	FALSE	401	
120	Recipient bank account is not valid	FALSE	401	
121	Bank's limit reached	FALSE	401	
122	Sender bank code is not valid	FALSE	401	
123	Recipient bank code is not valid	FALSE	401	
124	Consent is not valid	FALSE	401	
125	Recipient IBAN is blocked	FALSE	401	
126	Sender IBAN is not valid	FALSE	401	
127	Recipient IBAN is not valid	FALSE	401	
128	Recipient bank account is closed	FALSE	401	
129	Recipient IBAN does not belong to the bank	FALSE	401	
130	Payer Bank Service Unavailable	FALSE	401	
131	Payee Bank Service Unavailable	FALSE	401	
432	Country not authorised	FALSE	400	From 2024 R2
433	Country information mandatory for the payment	FALSE	400	From 2024 R2

999	Generic error	FALSE	500	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	

2.12. [POST] Refund

This chapter describes the additional fields available in the endpoint related to the “refund” service, whereas common fields are applied to all the endpoints.

2.12.1. Description

This API allows Merchants to execute refunds.

Merchant must indicate the mobile number of the buyer receiving the refund and the amount to be refunded. Otherwise, he can use a proxy (email or a document-id).
Buyer must be a Customer enrolled to the platform with the App installed and active.

2.12.2. Business Scenario

Possible scenarios:

- The merchant customer needs to refund a customer (retail refund type);

2.12.2.1. Direct outcome

•The buyer’s participant account is credited.

2.12.2.2. Related outcome

N.A

2.12.3. URL

POST
/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund

00001 = Participant A -> client_id= client_bank_a -> auth : client_id + assertion jwt

00002 = Participant B -> client_id= client_bank_b -> auth : client_id + assertion jwt

/business-payment-ms/services/groups/00001/banks/00001/bank-user/user01/tag/merchant01/refund

2.12.4. Operation

refund

2.12.5. Request

2.12.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.12.5.2. Request Header

Name	Type	Sample Value	Req.	Scope	Description
appId	String 64	03704576A8D81E96BF0C5D8E8953F4298842EC7A1803884A3EEC96C11C290099	Y	Security	Unique identification code generated by the Participant's Mobile Server related to the APP installed by the merchant. This field must be populated with the appId of the channel from which the merchant is operating
deviceOSVersion	String Max 255	Ios 10.5.26	N	Security	Operating system of the device.
deviceModel	String Max 255	ABCD1234	N	Security	Code associated with the device model.
deviceId	String Max 255	ABCD1234	N	Security	Unique identification code of the device.
deviceIpAddress	String Max 255	192.168.1.1	N	Security	IP address associated with the device.
country	String 2	AE	N	Security	Sender Country Code defined according to the standard ISO-3166 Alpha2.
timeZone	String PATTERN: +hh:mm or -hh:mm	+01:00	N	Security	Time zone on sender's side registered by the device. This Timezone can be different of the Timezone included in Timestamp field.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.12.5.3. Request Body

Name	Type	Sample Value	Req.	Description
refund	Object	N.A.	Y	Object that contains all the refund transaction's data.
refund.amount	Number Max 9	70.97	Y	Refund amount.
refund.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
refund.reason	String	wrong bill amount	N	Refund reason.

	Max 140			
refund.type	Enum(String)	disbursement	N	It defines the scope of the refund transaction values can be: 1. “disbursement”; 2. “retail”. Value is not mandatory and in case is not provided the platform will consider refund type implicitly “retail”.
refund.mobile	String Max 30	+353837892848	C	Mobile number of the buyer receiving the refund. This must be a mobile number of a Buyer Customer enrolled to the UAEIPP Overlay Service by one of the Participants. The Buyer Customer must have the app installed on his device and activated with this mobile number.
refund.proxy	Object	N.A.	C	Object related to the proxy used by the buyer alternatively to the mobile number for receiving the payment.
proxy.type	Enum	email	Y	Type of the proxy of the buyer to be used for making/receiving payments. Expected values: email; document-id proxy.type is as per enrolment. document-id and email are case insensitive
proxy.value	String	john.wick@gmail.com	Y	Value of the proxy.
refund.paymentTime	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}	2012-11-25T23:50:56.193	N	Execution date of the request.
refund.shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by UAEIPP Overlay Service
refund.cashDeskId	Number Min 8 Max 16	10000001	N	Identifier of the cash desk of the shop.

merchantTrxId	String Max 50	123456	Y	Transaction identifier of the refund generated by the Merchant Customer.
refund.categoryPurpose	String Max 35	CCP	Y	Category Purpose Code in a proprietary form. For the list of allowed values, please refer to the dedicated document From 2024 R4

2.12.6. Response

2.12.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.12.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the refund transaction. Possible values are: 'EFF' = Executed; 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Participant error
refund	Object	N.A.	Y	Object that contains all the refund transaction's data.
refund.paymentId	Number Max 16	123456	Y	Identification code of the refund generated by UAEIPP Overlay Service
refund.amount	Number Max 9	70.97	E	Refund amount.
refund.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217
refund.reason	String Max 140	wrong bill amount	E	Payment reason
refund.mobile	String Max 30	+353837892848	E	Mobile number of the buyer receiving the refund. This must be a mobile number of a Buyer Customer enrolled to the UAEIPP Overlay Service by one of the Participant. The Buyer

				Customer must have the app installed on his device and activated with this mobile number.
refund.proxy	Object	N.A.	E	Object related to the proxy used by the buyer alternatively to the mobile number for receiving the payment.
proxy.type	Enum	email	E	Type of the proxy of the buyer to be used for making/receiving payments. Expected values: email; document-id proxy.type is as per enrolment. document-id and email are case insensitive
proxy.value	String	john.wick@gmail.com	E	Value of the proxy.
refund.paymentTime	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}	2012-11-25T23:50:56.193	E	The execution date of the request.
refund.fees	Number	0	N	It will always be zero. As it is reserved for future use, it can be not expected.
refund.totalAmt	Number Max 9	70	N	It will always match with amount. As it is reserved for future use, it can be not expected.
refund.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by UAEIPP Overlay Service.
refund.cashDeskId	Number Min 8 Max 16	10000001	E	Identifier of the cash desk of the shop.
refund.type	Enum(String)	disbursement	E	It defines the scope of the refund transaction values can be: 3. “disbursement”; 4. “retail”. Value is not mandatory and in case is not provided the platform will consider refund type implicitly “retail”.

merchantTrxId	String Max 50	123456	E	Transaction identifier of the refund generated by the Merchant Customer.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.12.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code	
00000	Positive result	TRUE	200	
00023	Payment taken in charge, check the correct conclusion	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01018	No default bank account found	FALSE	401	
01027	Merchant not registered	FALSE	401	
01030	Communication Problems with Gateway	FALSE	401	
01033	Cash Desk not found	FALSE	401	
01034	The amount is not valid	FALSE	401	
01044	Shop not found	FALSE	401	
01075	Customer not found	FALSE	401	
01146	Payment reason not valid	FALSE	401	
01182	The APP-ID is not valid	FALSE	401	
01300	Merchant not found	FALSE	401	
01301	Shop not found	FALSE	401	
01302	Cash Desk not found	FALSE	401	
01346	Currency mismatch, bank account currency: {0}, payment currency: {1}	FALSE	400	
01347	{0} is not a managed currency	FALSE	400	
01348	No active bank account found with currency {0}	FALSE	400	
01365	Bank account blocked, it is impossible to proceed with the request	FALSE	401	
01371	Merchant not authorized	FALSE	401	
01383	Creditor not found	FALSE	400	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	

02001	The field {0} has not a valid format [{1}]	FALSE	400	
02005	The field {0} doesn't match any of the expected values	FALSE	400	
02010	At least one of the following fields is mandatory: {0}	FALSE	400	
03018	The bank is not enabled	FALSE	400	
101	Unknown user	FALSE	401	
102	Sender IBAN does not belong to the bank	FALSE	401	
103	Sender IBAN is blocked	FALSE	401	
104	Insufficient funds in sender account	FALSE	401	
105	Authorization denied by bank	FALSE	401	
106	Invalid request (eg. Incorrect timestamp format, service not available, etc.).	FALSE	400	
107	Rejection due to internal Payer Bank checks (eg. Fraud checks)	FALSE	401	
108	TransactionId missing in SCT initiation request	FALSE	401	
109	Rejection due to internal Payee Bank checks (eg. Fraud checks)	FALSE	401	
110	Bank is not able to provide balance information	FALSE	401	
111	Invalid client certificate presented	FALSE	401	
112	x-jws-signature mismatch	FALSE	401	
113	Bearer token mismatch Client	FALSE	401	
114	Too many request	FALSE	401	
115	Bank has revoked the consent	FALSE	401	
116	Method non allowed	FALSE	401	
117	Request not acceptable	FALSE	401	
118	Unsupported media type	FALSE	401	
119	Sender bank account is closed	FALSE	401	
120	Recipient bank account is not valid	FALSE	401	
121	Bank's limit reached	FALSE	401	
122	Sender bank code is not valid	FALSE	401	
123	Recipient bank code is not valid	FALSE	401	
124	Consent is not valid	FALSE	401	
125	Recipient IBAN is blocked	FALSE	401	

126	Sender IBAN is not valid	FALSE	401	
127	Recipient IBAN is not valid	FALSE	401	
128	Recipient bank account is closed	FALSE	401	
129	Recipient IBAN does not belong to the bank	FALSE	401	
130	Payer Bank Service Unavailable	FALSE	401	
131	Payee Bank Service Unavailable	FALSE	401	
999	Generic error	FALSE	500	
432	Country not authorised	FALSE	400	From 2024 R2
433	Country information mandatory for the payment	FALSE	400	From 2024 R2
TBD	The categoryPurpose provided is invalid	FALSE	400	From 2024 R4
04004	no {0} proxy customer found	FALSE	400	
04005	mobile numbers mismatch	FALSE	400	
04008	{0} is not a valid proxy type	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	

2.13. [GET] Check Reversal Status

This chapter describes the additional fields available in the endpoint related to the "checkBusinessReversalStatusChannel" service, whereas common fields are applied to all the endpoints.

2.13.1. Description

This API is called by the merchant customer to check the execution status of a requested reversal.

The platform provides "checkBusinessReversalStatusChannel" to let Merchants recover the actual result of reversal in case Merchants do not receive response due to technical issues. This operation may be used by merchant at any time, after "verifyBusinessReversalChannel" and "confirmBusinessReversalChannel" requests.

2.13.2. Business Scenario

The merchant customer needs to check the status of a reversal operation.

2.13.2.1. Direct outcome

- The status of the reversal transaction is delivered to the merchant customer.

2.13.2.2. Related outcome

N.A

2.13.3. URL

GET
/inquiry ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/business /reversal/status

2.13.4. Operation

checkBusinessReversalStatusChannel

2.13.5. Request

2.13.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.13.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description

merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.

2.13.5.3. Request Header

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.13.5.4. Request Body

N.A.

2.13.6. Response

2.13.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.13.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the reversal transaction. Possible values are: 'EFF' = Executed; 'DA_STR' = To be transferred (the request has been accepted by UAEIPP Overlay Service, waiting to be forwarded to the participants); 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Participant error 'RIF_A' = Refused automatically by the platform

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.13.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code	note
00000	Positive result	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01067	No payments found	FALSE	401	
01258	Transaction id not coherent	FALSE	400	Mismatch of transaction ID

01300	Merchant not found	FALSE	401	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03018	The bank is not enabled	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	

2.14. [GET] Check Refund Status

This chapter describes the additional fields available in the endpoint related to the “checkBusinessRefundStatus” service, whereas common fields are applied to all the endpoints.

2.14.1. Description

This API is called by the merchant customer to check the execution status of a requested refund.

The platform provides “checkBusinessRefundStatus” to let Merchants recover the actual result of refund operations in case Merchants do not receive response due to technical issues. This operation may be used by merchant at any time, after “refund” requests.

2.14.2. Business Scenario

The merchant customer needs to check the status of a refund operation.

2.14.2.1. Direct outcome

- The status of the refund transaction is delivered to the merchant customer.

2.14.2.2. Related outcome

N.A

2.14.3. URL

GET
/business-payment-ms/services /groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund/status

2.14.4. Operation

checkBusinessRefundStatus

2.14.5. Request

2.14.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.14.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.

2.14.5.3. Request Header

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.14.5.4. Request Body

N.A.

2.14.6. Response

2.14.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

2.14.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the refund transaction. Possible values are: 'EFF' = Executed; 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Participant error

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

2.14.6.3. **Result Codes**

CODE	MESSAGE	RESULT	HTTP Code	note
00000	Positive result	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01067	No payments found	FALSE	401	
01258	Transaction id not coherent	FALSE	400	Mismatch of transaction ID
01300	Merchant not found	FALSE	401	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03018	The bank is not enabled	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	

2.15. SWAGGER

See file: “business payment ms QrCode” and “inquiry ms QrCode”

2.16. APPENDIX

2.16.1. QrCode Expiration Time

QRCodes created by merchants with registerQrCodeChannel lasts 5 minutes. This parameter is configurable (1 day , configurable) .
After that time

- QRCode goes to status “expired”.
- checkStatusQrCodeChannel will return “00002” “QR Code expired” with payment.status= Expired.

2.16.2. Preauthorization Expiration Time

After a registerQrCodeChannel with paymentType set to “PRE” (= Preauthorized payment has been requested successfully by Merchant’s Payment Gateway and confirmed by Buyer via APP), Merchant has 24 hours,1 day (starting from successful response sent to registerQrCodeChannel request, configurable) to invoke finalizePaymentChannel with “confirm” query parameter set to “true” (= confirm the payment with the actual amount).
After that time

1. Payment request goes to status “expired”
    a. finalizePaymentChannel will return “01037” “Preauthorization is expired”.
    b. checkStatusQrCodeChannel will return the payment with status “EXP” - Request expired.
2. platform in background will trigger a flow with participants to delete reservation made on buyer participant account.

2.16.3. Preauthorization not confirmed

After a registerQrCodeChannel with “paymentType” set to “PRE” (= Preauthorized payment has been requested successfully by Merchant’s Payment Gateway and confirmed by Buyer via APP), in case Merchant invokes finalizePaymentChannel with “confirm” query parameter set to “false” (= refuse the payment), platform in background will trigger a flow with participants to delete reservation made on buyer participant account.

2.16.4. Reversal verify expiration time

After a verifyBusinessReversalChannel has been requested successfully, Merchant has 5 minutes (starting from successful response sent to verifyBusinessReversalChannel request, configurable) to invoke confirmBusinessReversalChannel.

After that time

1. The platform in background will trigger a flow with participant to delete reservation made on Merchant’s participant account.
2. Any execution of the checkBusinessReversalStatusChannel will return a response with status attribute = 'RIF_A' - Refused automatically by the platform.

2.16.5. Reversal and refund timeouts

The Platform provides “checkBusinessReversalStatusChannel” and “checkBusinessRefundStatus” to let Merchants recover the actual result of reversal and refund operations in case Merchant does not receive response due to technical issues. These operations may be used by merchant at any time, after “verifyBusinessReversalChannel”, “confirmBusinessReversalChannel” and “refund” requests.

The platform relies on Participants’ APIs to reply within 20 seconds when “P2B Refund VERIFY FLOW” and “P2B Refund CONFIRM FLOW” are executed (see section below for description of these flows), then Merchant may decide its own timeout definition according to inner business logic.

“P2B Refund VERIFY FLOW” and “P2B Refund CONFIRM FLOW” are executed with participants to support both Reversal and Refunds described in this document. In case of Reversal in **“**P2B Refund VERIFY FLOW” and “P2B Refund CONFIRM FLOW” the platform will forward to participants full references of the original transaction to be reverted.

2.16.6. Polling frequency (GET checkStatusQrCodeChannel)

Payment Gateway can invoke max 10 times per minute this endpoint in order to retrieve payment status.

2.16.7. Generate UAEIPP Overlay Service valid dynamic QRCode

In order to generate valid QRCode Merchant must encode an alphanumeric string created following EMV standard.

Basing on the EMV standard, each field in the QR Code String is composed by 3 elements:

- An **ID**, defined by the standard;
- ID value **Length**, which corresponds to the number of digits composing the ID value;
- The ID **value.**

Here follows the minimal list of mandatory fields used to generate a Merchant Dynamic QR Code EMV Compliant:

ID name	ID	Note
Payload Format Indicator	00	Value “01”, defined by the EMV standard
Point of Initiation Method	01	Value “12”, defined by the EMV standard, which identifies the dynamic QR Code
Merchant Account Information	26	Field composed by the Globally Unique Identifier (see row below)
Merchant Account Information - Globally Unique Identifier	00	Optional Sub-field of the Merchant Account Information. This field has to be valued with the Merchant Tag. Although this field is required by the EMV standard, the System will not consider it.
Merchant Category Code	52	Field valued with the Merchant Category Code
Transaction Currency	53	Field valued with “784”, which corresponds to AED in ISO 4217 numeric standard
Transaction Amount	54	Amount of the transaction, mandatory in the dynamic QR Code
Country Code	58	Field to be valued with ISO 3166 standard value
Merchant Name	59	Field valued with the Merchant Company name, as it is registered on the UAEIPP Overlay System
Merchant City	60	Field valued with the Merchant City

Additional Data Field	62	Additional data field, used for custom purpose. In our solution, it is composed by the fields depicted in the following rows.
Additional Data Field – Bill Number	01	Optional Sub-field of the Additional Data field, maximum 25 characters long. It represents the unique identifier of the generated QR Code, composed by following values: • merchantTag: value that uniquely identifies the specific merchant in the platform, included in the url of the POST registerQrCodeChannel; • qrCodeId value received in response to POST registerQrCodeChannel; • RFU value, always null; • shopId passed in request body toPOST registerQrCodeChannel
Additional Data Field – Purpose of Transaction	08	Optional Sub-field of the Additional Data field, maximum 25 characters long. It represents the Payment category of the transaction. Expected values: 1. "01" for Bill payment; 2. "02" for Prepaid top-up; 3. "03" for Purchase. Note: the field is not mandatory. If the Merchant, while creating the QR Code, is providing the field, this must be reported in the verifyPayment and confirmPayment.
CRC	63	Field calculated by the system, once entered all the fields. This field indicates the correctness of the entire EMV string

Please note: each field composing the Additional Data Field – Bill Number (ID "01", inside ID "62"), should be followed by the hashtag "#".
The string to be encoded must have the format below:

merchantTag#qrCodeId##shopId#

since the RFU is null, it is not represented in the string. Only the hashtag following the RFU field is represented.

EMV String Example Standard Explanation

EMV string example (Merchant tag in bold) to be used in the QrCode image generation.

00020101021226110007**RX268XZ**52049999530378454031**100**5802AE5912**MerchantTest**6005DUBAI62290121**RX268XZ#vZj03**##10001#0802016304E32C

Fields description:

ID	Length	Value	Field example - Result	Fixed value*
00	02	01	000201	Y
01	02	12	010212	Y
26	11	0007**RX268XZ**	26110007**RX268XZ**	N

00 (part of ID26)	07	**RX268XZ**	0007**RX268XZ**	N
52	04	9999	52049999	N
53	03	784	5303784	Y
54	03	100	5403100	N
58	02	AE	5802AE	Y
59	12	MerchantTest	5912MerchantTest	N
60	05	DUBAI	6005DUBAI	N
62	29	0121**RX268XZ**#vZj03##10001#080201	62290121**RX268XZ**#vZj03##10001#080201	N
01 (part of the ID62)	21	**RX268XZ**#vZj03##10001#	0121**RX268XZ**#vZj03##10001#	N
08 (part of the ID62)	02	01	080201	N
63	04	E32C	6304E32C	N

*as per the current implementation, the currency can be only AED and the ISO country code can be only AE

2.16.8. Participants Flows

Following flows to briefly explain how UAEIPP Overlay Service interacts with participants to execute a Payment, or preauthorization or Reversal.

P2B VERIFY FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to verify Buyer and Merchant IBANs in order to execute a payment. As final result this flows reserve fund on Buyer Account.
P2B CONFIRM FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to initiate a payment after a previously successful run P2B VERIFY FLOW
P2B Preauth VERIFY FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to verify Buyer and Merchant IBANs in order to execute a Preauthorized payment. As final result this flows reserve fund on Buyer Account.
P2B Preauth CONFIRM FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to confirm final amount of a previously successfully authorized preauthorization with P2B Preauth VERIFY FLOW
P2B Refund VERIFY FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to verify Buyer and Merchant IBANs in order to execute a refund or a reversal. As final result this flows reserve fund on Merchant Account.
P2B Refund CONFIRM FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to initiate a reversal or a refund after a previously successful run P2B Refund VERIFY FLOW
Delete Reservation FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service to unblock funds previously reserved after successfully P2B VERIFY FLOW or P2B Preauth VERIFY FLOW or P2B Refund VERIFY FLOW

2.16.9. Proxy Management Principles

In this paragraph, the main principles related to the Proxy management of customers are described. Currently, the proxies identify only the consumers.

The proxy types that are currently supported for consumers by the UAEIPP Overlay Service platform are the following:

1. "e-mail";
2. "document-id"
3. Phone number.

The following sections describe the principles valid for each proxy type for consumers.

2.16.9.1. **E-mail**

For the proxy with type "e-mail" the following rules must be followed:

1. **Mobile number relationship**: the e-mail address of a consumer is linked **one to one** with the mobile number. One mobile number cannot have more than one email address. Therefore, it's not possible to register the same e-mail with different mobile numbers;
2. **Payments Scenarios availability**: the e-mail address can be used in P2P Send Money and Request to Pay scenarios (Split Bill transactions excluded), P2B Requests to Pay sent by the merchant (including e-commerce scenarios) and B2P payment scenarios (refund and disbursement), alternatively to the mobile number;
3. **Default Account**: the default account linked to the email is equal to the default of the mobile number, due to the unique relationship with the mobile number. If the participant account at the mobile number level is updated, the change is also reflected to the default of the e-mail address;
4. **E-mail format**: the e-mail must be set in Latin characters only, Arabic characters are not supported in the e-mail input field and must comply with the standard format, e.g. example@domain.com;
5. **E-mail creation**: the e-mail address can be registered manually by the user by using the APP after activation. It cannot be defined in the enrolment phase by the participant. The e-mail address requires to be validated by the user as described in the next principle;
6. **E-mail validation**: in order to complete the successful registration of the e-mail address, the user must validate it through the OTP approach, by inserting the OTP code in the APP once the customer has received it in his/her email's inbox. In case the customer's device is changed, but the phone number is the same, the email address does not need to be redefined and validated;
7. **E-mail update**: the e-mail can be updated by the customer and to complete the operation it has to be validated as explained in step 6. If the customer has not validated the e-mail, he is not reachable with this proxy;
8. **E-mail deletion**: the e-mail address cannot be deleted directly by the customer, only through Participants channels (FE API and Web Portal) can be performed.

2.16.9.2. **Document-ID**

For the proxy with type "document-id", that is meant as a unique datum of an individual, the following rules must be followed:

1. **Mobile number relationship**: the document-id of a consumer can be linked to **one or more** mobile numbers, but **not vice versa** (1 mobile number cannot be linked to more than one document-id at the same time). The platform will not allow this duplication. It can be possible only in case of recycled numbers;
2. **Payments Scenarios availability**: the document-id can be used alternatively to the mobile number as an identifier of the UAEIPP Overlay Service user in P2B (including e-commerce) and B2P (refund and disbursement) scenarios;
3. **Default Account**: because of the 1-N relationship with the mobile number, the document-id needs a dedicated default account, in order to let the user make/receive payments using the document-id as a proxy.

The default account at the document-id level is the same at the mobile number level during enrolment, the first account of the list is set as default also at the document-id level). After the enrolment, when the customer activates the APP, he can change the default account by using the APP that will call the proper API (updateProxyDefaultBankAccount – API functionality described in SDK Interface Specifications and Front End APIs for Participants' Channels). The default account at the document-id level, can, therefore, differ from the one set at the mobile number level. If the Participant blocks a default account, the consumer is not reachable with the linked proxy until the customer selects another participant account as default. The consumer, to change the default account must select one of the participant accounts connected to the APP he/she is using. For consumers, no automatic change of the default account can be done by the participant or the UAEIPP Overlay Service platform;

4. **Document-ID format**: compared to the e-mail address, the document-id has not a specific format since can be equal to the Emirates ID or to the Passport ID. When a customer has both the documents (EmiratesID and PassportID), the participants must enrol the customer with the EmiratesID;
5. **Document-ID creation**: the document-id can be created only during the enrolment phase through the Participant channels (FE API, Web Portal and Bulk Operations) since it is a mandatory user's datum;
6. **Document-ID validation**: compared to the e-mail address, the document-id does not require any validation from the user, it is under the responsibility of the Participant to certify the datum;
7. **Document-ID update**: the document-id cannot be updated by the user, neither by using Participant channels (FE API, Web Portal and Bulk Operations), since it is a unique identifier of the user. If a customer change his document-Id, he must be unenrolled and re-enrolled (security reasons);
8. **Document-ID deletion**: the document-id cannot be deleted by the user, neither through Participant channels (FE API, Web Portal and Bulk Operations).

2.16.10. Technical Service Provider

In this paragraph are described the main principles, for merchants, related to the Technical Service Provider and Technical Service Provider as Acquirer. Currently, merchant APIs can be called not only by Participant Banks but also by a Technical Service Provider or by a Technical Service Provider as Acquirer.

The main difference from a Technical Service Provider (providerType = "PROVIDER") and a Technical Service Provider as Acquirer (providerType = "ACQUIRER") is that the first one is only be able to call operation FE APIs while the second one can also call, in addition to the first one, maintenance FE APIs.

For a Technical Service Provider, or for a Technical Service Provider as Acquirer, it's possible to call a Merchant API using their own Bank Code and Group Code as path parameters and these 3 headers must also be populated (For more information about the request headers, refer to the common ones described in the paragraph Common Fields):

- providerType
- participantGroupCode
- participantBankCode

These are the conditions for a a Technical Service Provider, or for a Technical Service Provider as Acquirer, to be able to make a FE API call:

- The parent and sub code indicated as path parameters must match a corresponding, valid and active Technical Service Provider, or Technical Service Provider as Acquirer, and this must also be enabled to the providerType indicated in the header field
- The participantGroupCode and participantBankCode headers must match a corresponding, valid and active Participant Bank
- The Participant Bank and Technical Service Provider, or Technical Service Provider as Acquirer, must be associated

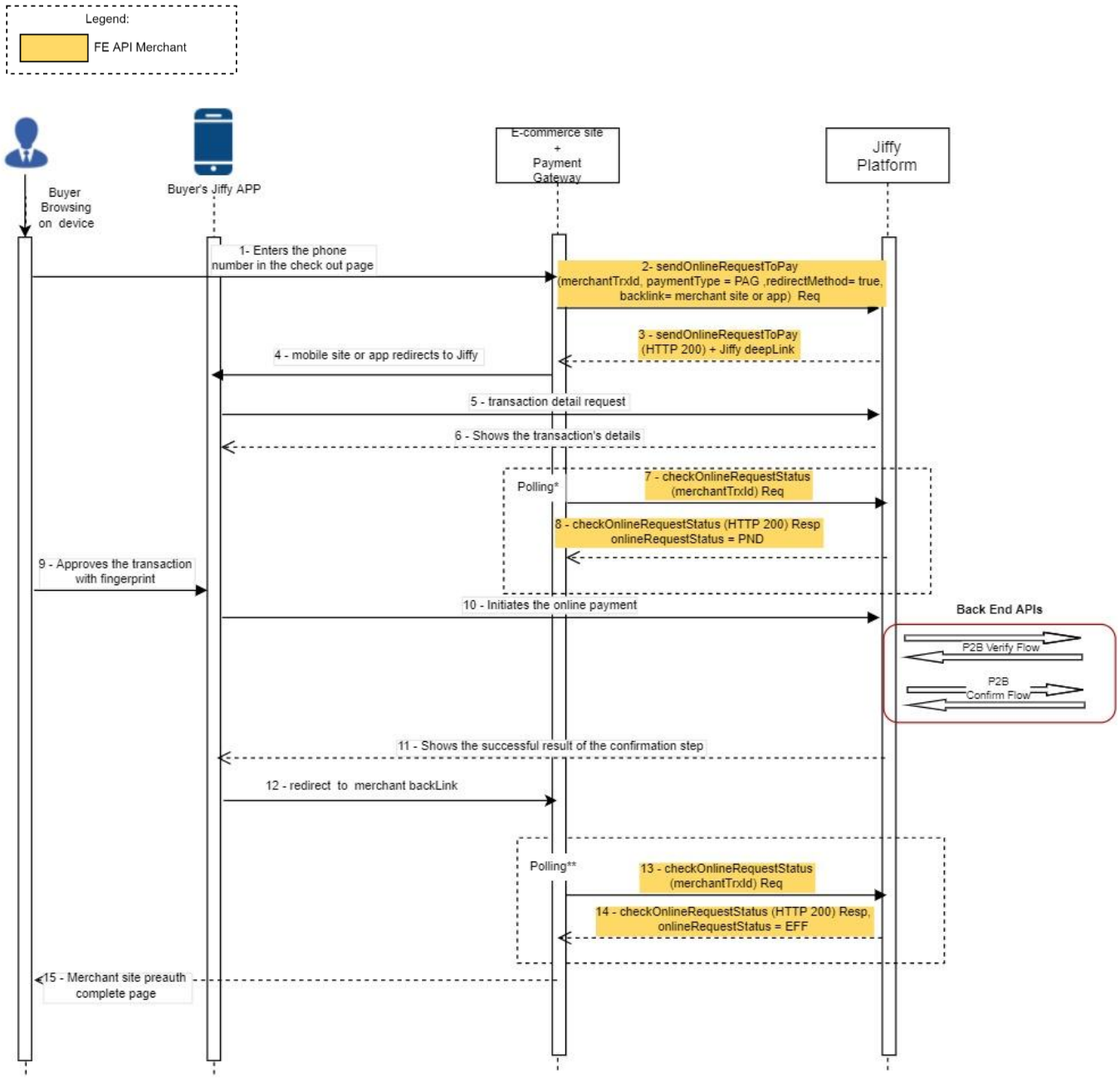
- The Technical Service Provider, or Technical Service Provider as Acquirer, must be assigned to the Merchant

3. SEQUENCE DIAGRAMS – REQUEST TO PAY

In this chapter there is a brief overview of the functionalities available through Front End API interface to and small merchant customers for executing P2E payment transactions. API Gateway makes available following APIs to allow merchant customers through the Payment Gateway to perform online payment transactions (e-commerce payment scenario).

3.1. Happy Flows

3.1.1. Payment Flow (redirect)



Polling

Merchant's Payment Gateway may start polling Jiffy Platform invoking checkOnlineRequestStatus after receiving http 200 in response to sendOnlineRequestToPay.
* Polling continues when checOnlineRequestStatus response contains status "PND"

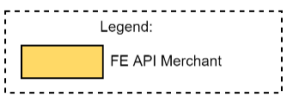
**(happy flow) Polling Stops when checOnlineRequestStatus response contains status "EFF" .

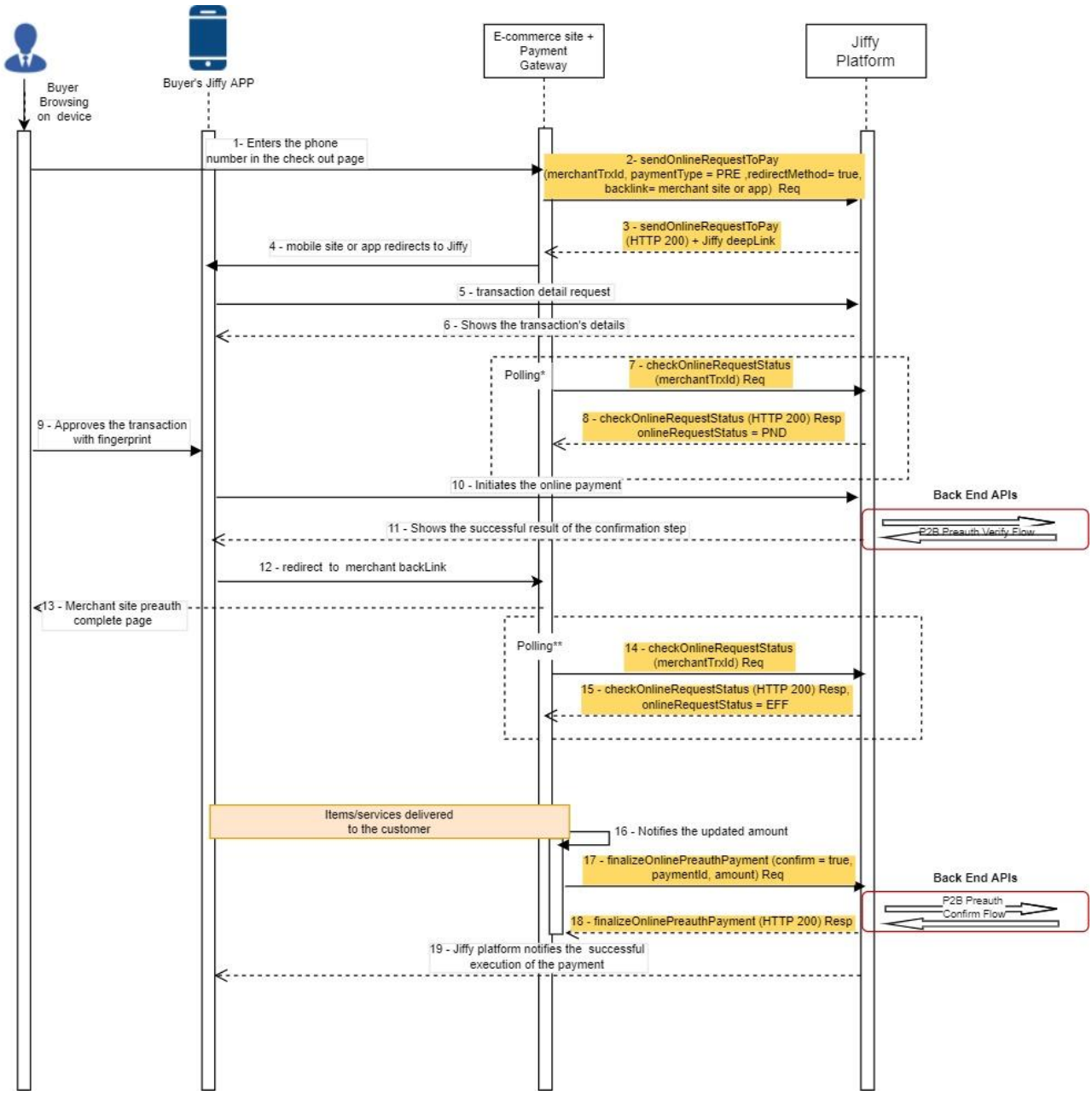
All other statuses do not requires Gateway to keep on polling.

3.1.1.1. Sequence diagram description

1. Buyer is browsing Merchant app or mobile website choose UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow with redirect, Merchant app or mobile website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PAG ,
    b. redirectMethod= true,
    c. backlink= merchant site or app url,
3. UAEIPP Overlay Service replies by returning UAEIPP Overlay Service deeplink
4. Merchant app or mobile website redirects buyer to UAEIPP Overlay Service APP
5. at launch, the APP requests transaction detail to UAEIPP Overlay Service
6. UAEIPP Overlay Service provides details to UAEIPP Overlay Service App.
7. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus
8. UAEIPP Overlay Service replies with status “PND” (app or mobile website keeps on polling)
9. Buyer authorizes payment with SCA on the UAEIPP Overlay Service App.
10. UAEIPP Overlay Service APP triggers UAEIPP Overlay Service to orchestrate payment flows between buyer’s participant and merchant participant
11. Flow has finished successfully, complete payment message is displayed on UAEIPP Overlay Service APP
12. The APP redirects buyer to app or mobile website following backlink provided at point 2
13. Merchant app or mobile website keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus
14. UAEIPP Overlay Service replies with status “EFF” (Buyer confirmed payment and UAEIPP Overlay Service payment orchestration went good)
15. Merchant app or mobile website displays successful payment message

3.1.2. Pre-authorisation Flow (redirect)




Buyer Browsing on device
Buyer's Jiffy APP
E-commerce site + Payment Gateway
Jiffy Platform
1- Enters the phone number in the check out page
2- sendOnlineRequestToPay (merchantTrxId, paymentType = PRE ,redirectMethod= true, backlink= merchant site or app) Req
3 - sendOnlineRequestToPay (HTTP 200) + Jiffy deepLink
4 - mobile site or app redirects to Jiffy
5 - transaction detail request
6 - Shows the transaction's details
Polling*
7 - checkOnlineRequestStatus (merchantTrxId) Req
8 - checkOnlineRequestStatus (HTTP 200) Resp onlineRequestStatus = PND
9 - Approves the transaction with fingerprint
10 - Initiates the online payment
Back End APIs
P2B Preauth Verify Flow
11 - Shows the successful result of the confirmation step
12 - redirect to merchant backLink
13 - Merchant site preauth complete page
Polling**
14 - checkOnlineRequestStatus (merchantTrxId) Req
15 - checkOnlineRequestStatus (HTTP 200) Resp, onlineRequestStatus = EFF
Items/services delivered to the customer
16 - Notifies the updated amount
17 - finalizeOnlinePreauthPayment (confirm = true, paymentId, amount) Req
Back End APIs
P2B Preauth Confirm Flow
18 - finalizeOnlinePreauthPayment (HTTP 200) Resp
19 - Jiffy platform notifies the successful execution of the payment

Polling

Merchant's Payment Gateway may start polling Jiffy Platform invoking checkOnlineRequestStatus after receiving http 200 in response to sendOnlineRequestToPay.
* Polling continues when checOnlineRequestStatus response contains status "PND"

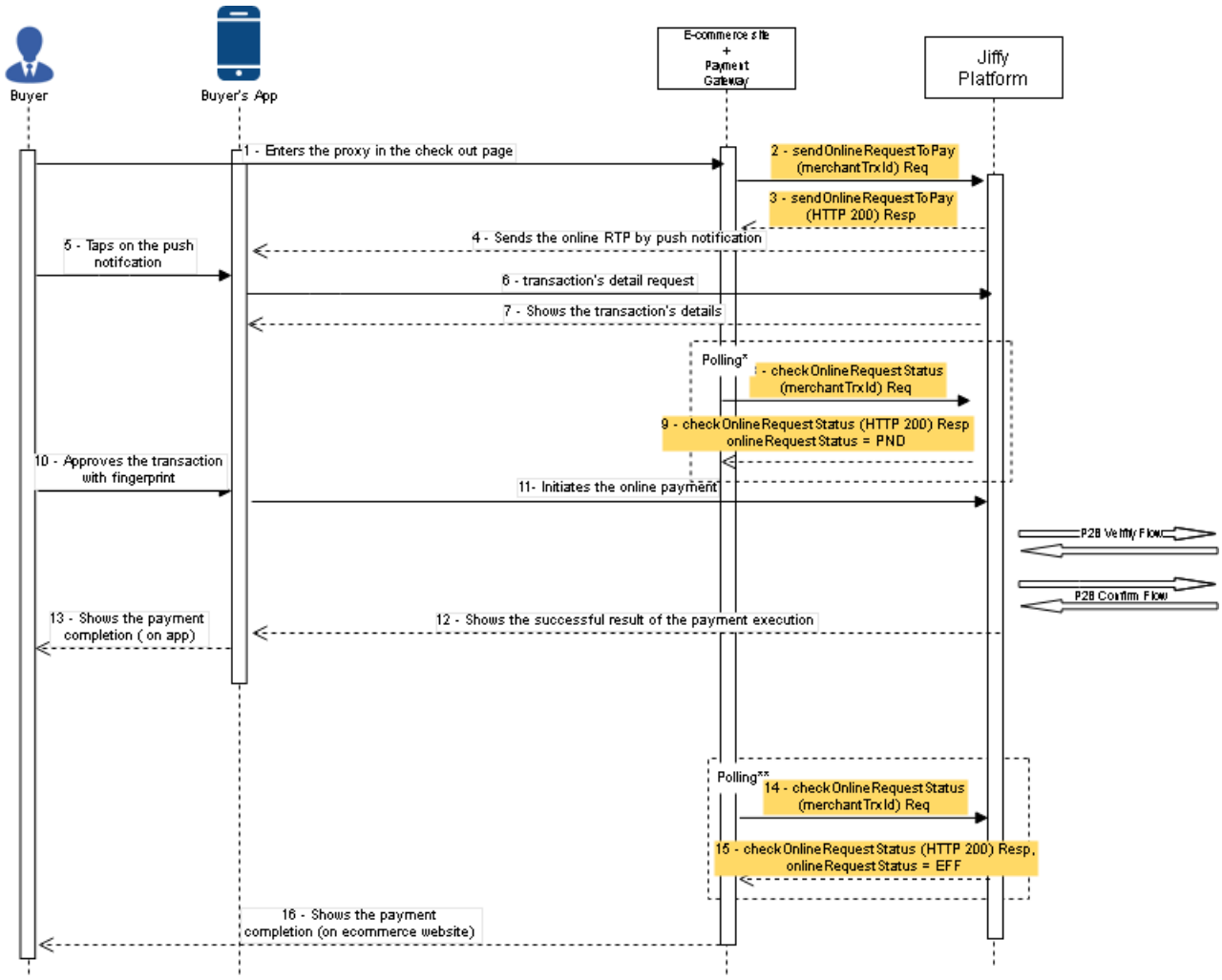
**(happy flow) Polling Stops when checOnlineRequestStatus response contains status "EFF" .

All other statuses do not requires Gateway to keep on polling.

3.1.2.1. Sequence diagram Description

1. Buyer's browsing Merchant app or mobile website choose UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow with redirect, Merchant app or mobile website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
   a. paymentType = PRE ,
   b. redirectMethod= true,
   c. backlink= merchant site or app url,
3. UAEIPP Overlay Service replies returning UAEIPP Overlay Service deeplink
4. Merchant app or mobile website redirects buyer to UAEIPP Overlay Service APP
5. at launch, the APP requests transaction detail to UAEIPP Overlay Service
6. UAEIPP Overlay Service provides details to the APP
7. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus
8. UAEIPP Overlay Service replies with status "PND" (app or mobile website keeps on polling)
9. Buyer authorizes Preauthorized Payment with SCA
10. UAEIPP Overlay Service APP triggers UAEIPP Overlay Service to orchestrate preauthorized payment verify flow between buyer's participant and merchant participant . At the end of this flow buyer's account is not debited but funds are reserved (amount of funds reserved is equal to payment.amount)
11. Flow has finished successfully, complete preauthorization payment message is displayed on UAEIPP Overlay Service APP
12. The APP redirects buyer to app or mobile website following backlink provided at point 2
13. Merchant app or mobile website displays successful payment message
14. Merchant app or mobile website keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus
15. When UAEIPP Overlay Service replies with status "EFF" (Buyer confirmed preauthorized payment and UAEIPP Overlay Service payment orchestration went good)
16. After conclusion of business process, merchant may update actual amount to be debited on buyer's account
17. Merchant invokes finalizeOnlinePreauthPayment with queryParameter "confirm" set to "true" and actual amount in request body then UAEIPP Overlay Service orchestrates preauthorized payment confirmation flow between buyer's participant and merchant participant. At the end of this flow buyer's account is debited with actual amount.
18. UAEIPP Overlay Service returns positive response to merchant to notify that payment with actual amount has been debited
19. UAEIPP Overlay Service notifies to Buyer's APP execution of the preauthorized payment with actual amount

3.1.3. Proxy Payment Flow (no redirect Push Notification)

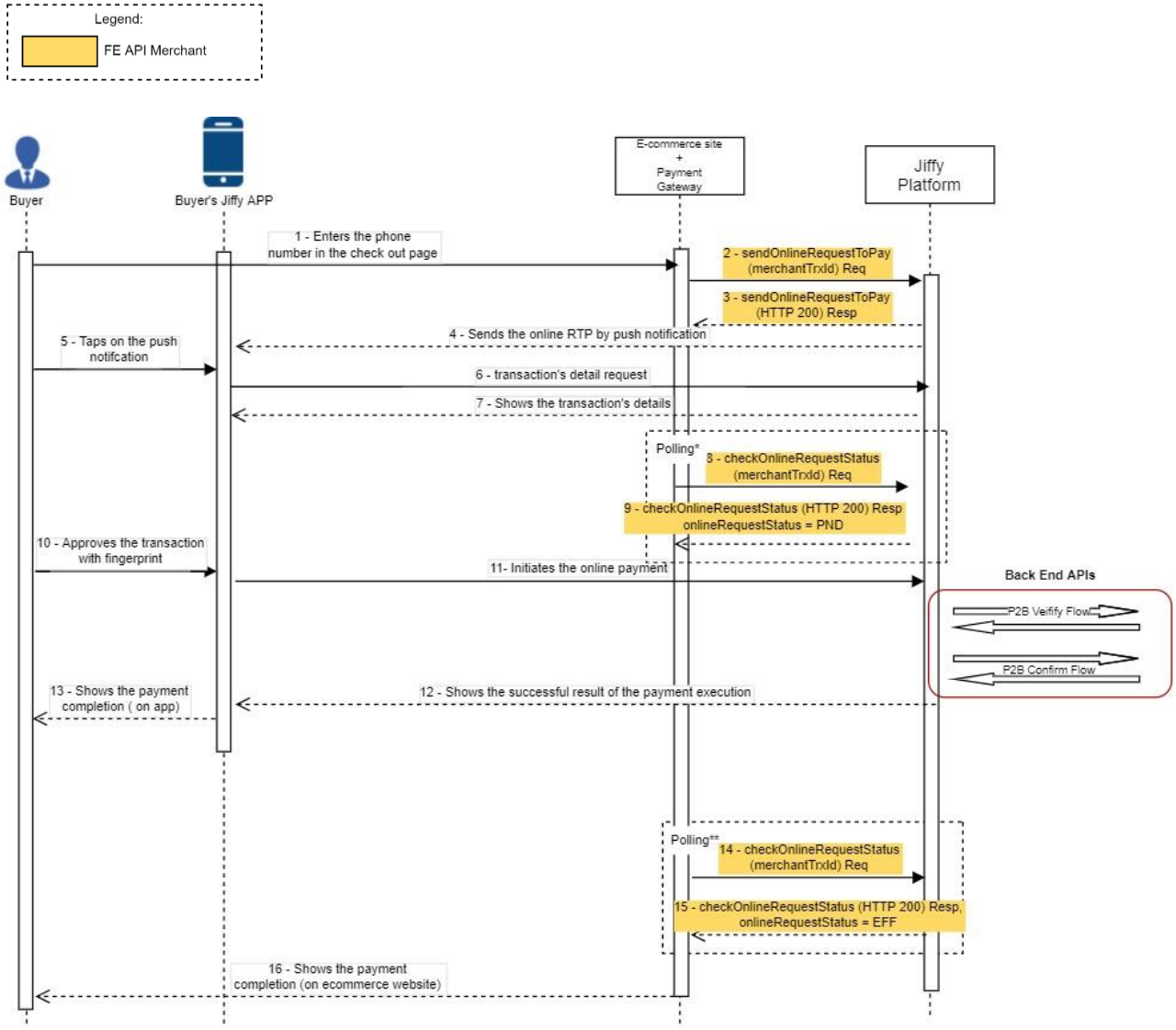


3.1.3.1. Sequence diagram description

1. Buyer is browsing Merchant website, chooses UAEIPP Overlay Service as payment method in checkout page, provides the proxy (email or document-id).
2. In order to trigger payment flow with notification, Merchant website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PAG,
    b. redirectMethod= false,
3. UAEIPP Overlay Service replies returning positive response, meaning that a request to pay has been created and Merchant website can start polling to retrieve payment status (waiting for buyer to confirm payment).
4. UAEIPP Overlay Service sends a Request to pay notification to the APP.
5. Buyer taps on notification.
6. The APP requests transaction detail to UAEIPP Overlay Service.
7. UAEIPP Overlay Service provides details to the APP.
8. Merchant website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
9. UAEIPP Overlay Service replies with status “PND”(Merchant website keeps on polling).
10. Buyer authorizes payment with SCA.

11. UAEIPP Overlay Service APP triggers UAEIPP Overlay Service to orchestrate payment flows between buyer's participant and merchant participant.
12. Flow has finished successfully, complete payment message is notified to the APP.
13. Successful payment message is displayed on the APP.
14. Merchant website keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
15. UAEIPP Overlay Service replies with status "EFF"(Buyer confirmed payment and UAEIPP Overlay Service payment orchestration went well).
16. Merchant website displays successful payment message.

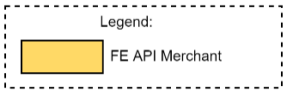
3.1.4. Payment Flow (no redirect Push Notification)

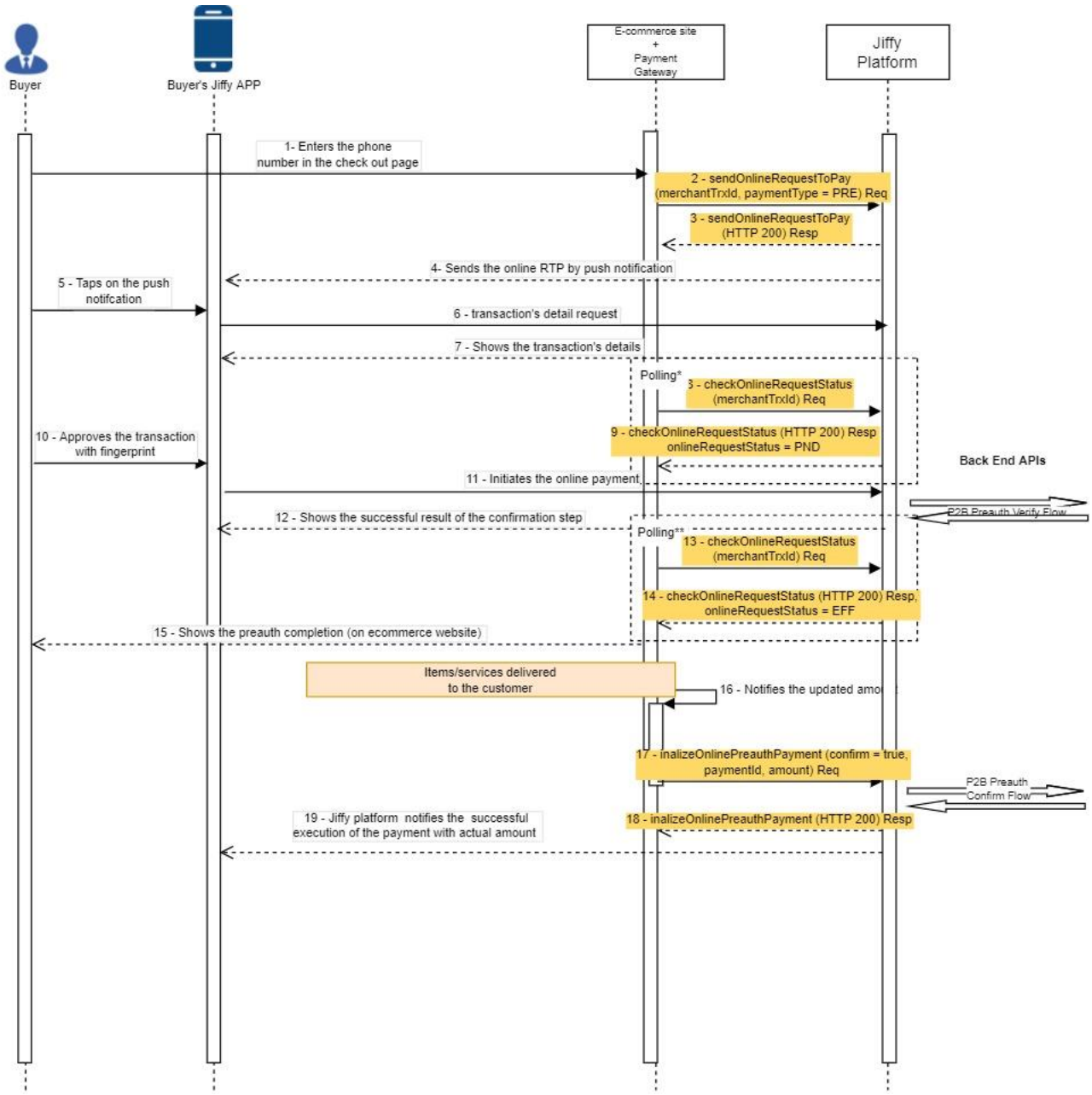


3.1.4.1. Sequence diagram description

2. Buyer is browsing Merchant website, chooses UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
3. In order to trigger payment flow with notification, Merchant website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    b. paymentType = PAG,
    c. redirectMethod= false,
4. UAEIPP Overlay Service replies returning positive response, meaning that a request to pay has been created and Merchant website can start polling to retrieve payment status (waiting for buyer to confirm payment).
5. UAEIPP Overlay Service sends a Request to pay notification to UAEIPP Overlay Service APP.
6. Buyer taps on notification.
7. UAEIPP Overlay Service APP requests transaction detail to UAEIPP Overlay Service.
8. UAEIPP Overlay Service provides details to UAEIPP Overlay Service APP.
9. Merchant website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
10. UAEIPP Overlay Service replies with status “PND”(Merchant website keeps on polling).
11. Buyer authorizes payment with SCA.
12. UAEIPP Overlay Service APP triggers UAEIPP Overlay Service to orchestrate payment flows between buyer’s participant and merchant participant.
13. Flow has finished successfully, complete payment message is notified to UAEIPP Overlay Service APP.
14. Successful payment message is displayed on UAEIPP Overlay Service APP.
15. Merchant website keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
16. UAEIPP Overlay Service replies with status “EFF”(Buyer confirmed payment and UAEIPP Overlay Service payment orchestration went well).
17. Merchant website displays successful payment message.

3.1.5. Preauthorization Flow (no redirect - push notification)



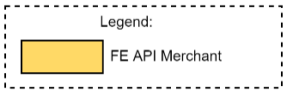


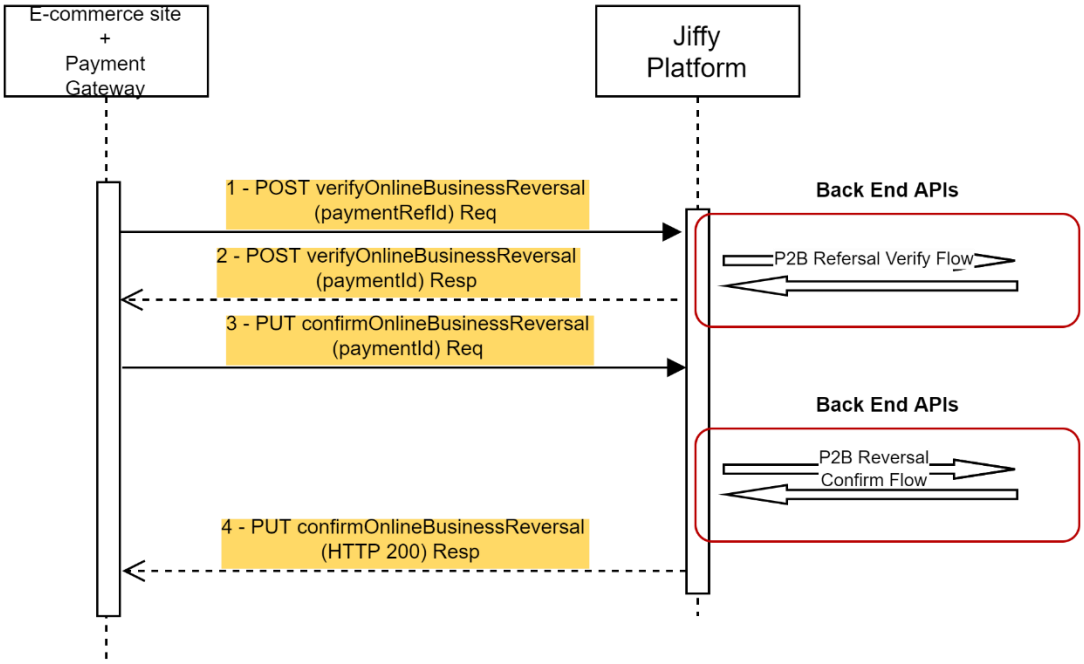
3.1.5.1. Sequence diagram Description

1. Buyer is browsing Merchant website, chooses UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow with redirect, Merchant website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PRE ,
    b. redirectMethod= false,
3. UAEIPP Overlay Service replies returning positive response, meaning that a request to pay has been created and Merchant website can start polling to retrieve payment status (waiting for buyer to confirm preauthorized payment).
3. UAEIPP Overlay Service sends a Request to pay notification to UAEIPP Overlay Service APP.

5. Buyer taps on notification.
6. Buyer APP requests transaction detail to UAEIPP Overlay Service.
7. Buyer APP provides details to UAEIPP Overlay Service.
8. Merchant website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
9. UAEIPP Overlay Service replies with status “PND” (Merchant website keeps on polling).
10. Buyer authorizes payment with SCA.
11. The APP triggers UAEIPP Overlay Service to orchestrate preauthorized payment verify flow between buyer’s participant and merchant participant. At the end of this flow buyer’s account is not debited but funds are reserved. Amount of funds reserved is equal to amount of the Request to Pay at point 2.
12. Flow has finished successfully, complete preauthorization payment message is displayed on the APP.
13. Merchant website keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
14. UAEIPP Overlay Service replies with status “EFF” (Buyer confirmed preauthorized payment and UAEIPP Overlay Service payment orchestration went well).
15. Merchant website displays successful preauthorized payment message.
16. After conclusion of business process, merchant may update actual amount to be debited on buyer’s account.
17. Merchant invokes finalizeOnlinePreauthPayment with queryParameter “confirm” set to “true” and actual amount in request body then UAEIPP Overlay Service orchestrates preauthorized payment confirmation flow between buyer’s participant and merchant participant. At the end of this flow buyer’s account is debited with actual amount.
18. UAEIPP Overlay Service returns positive response to merchant to notify that payment with actual amount has been debited.
19. UAEIPP Overlay Service notifies to buyer’s app execution of the preauthorized payment with actual amount.

3.1.6. Reversal Flow

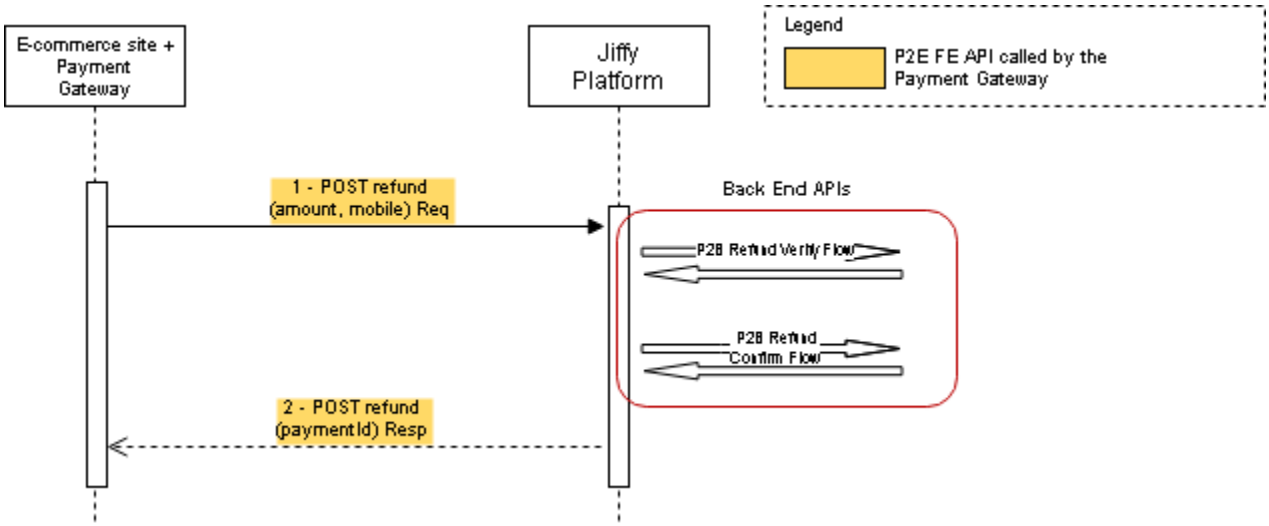




3.1.6.1. Sequence diagram description

1. Merchant invokes verifyOnlineBusinessReversal providing reference to payment to revert.
2. In case Payment exists and was successfully processed UAEIPP Overlay Service provides a positive response and a new paymentId representing the reversal to be confirmed.
3. Merchant invokes confirmOnlineBusinessReversal providing paymentId received at point 2 to confirm and trigger UAEIPP Overlay Service to orchestrate refund flow between merchant's participant and buyer's participant.
4. Orchestration ends successfully then UAEIPP Overlay Service replies to merchant a positive response meaning that buyer will be credited.

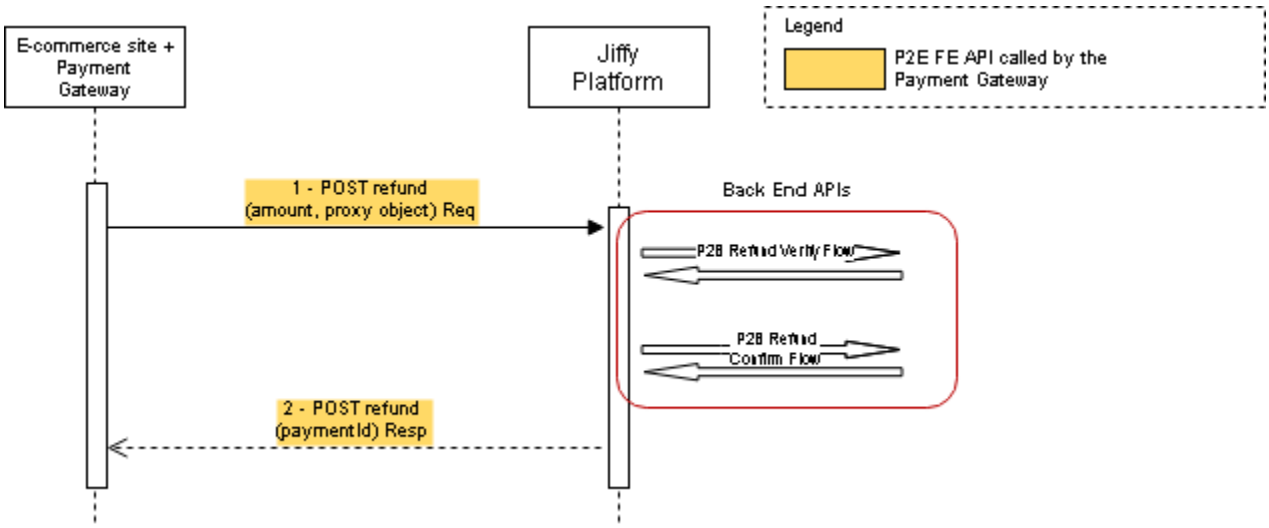
3.1.7. Refund Flow



3.1.7.1. Sequence diagram description

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's mobile number. After that, verify and confirm flows with participants are sequentially called in order to perform the refund transaction from merchant's participant to buyer's participant.
2. Refund Verify and Refund Confirm Flows have been successfully executed, then UAEIPP Overlay Service replies to the merchant with a positive message meaning that the buyer's participant account will be credited.

3.1.8. Proxy Refund Flow



3.1.8.1. Sequence diagram description

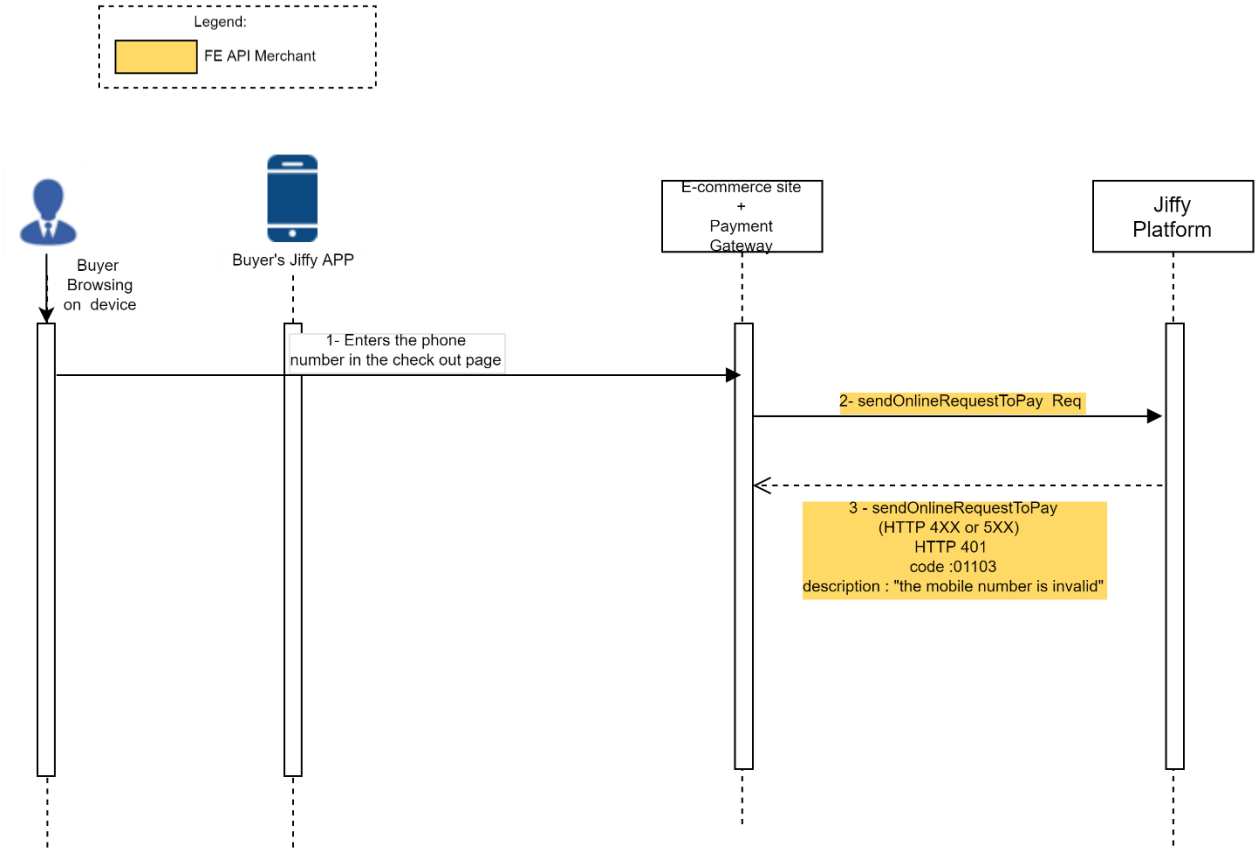
3. Merchant invokes the "refund" API providing the amount to be refunded and the proxy data of the buyer (e.g. email or document-id). After that, verify and confirm flows with participants are sequentially called in order to perform the refund transaction from merchant's participant to buyer's participant.
4. Refund Verify and Refund Confirm Flows have been successfully executed, then UAEIPP Overlay Service replies to the merchant with a positive message meaning that the buyer's participant account will be credited.

3.2. Unhappy Flows

3.2.1. Unhappy Flow: sendOnlineRequestToPay response is negative

This unhappy flow description can be applied:

- Simple e-commerce payments (both redirect and push notification)
- Pre-authorised payments (both redirect and push notification)



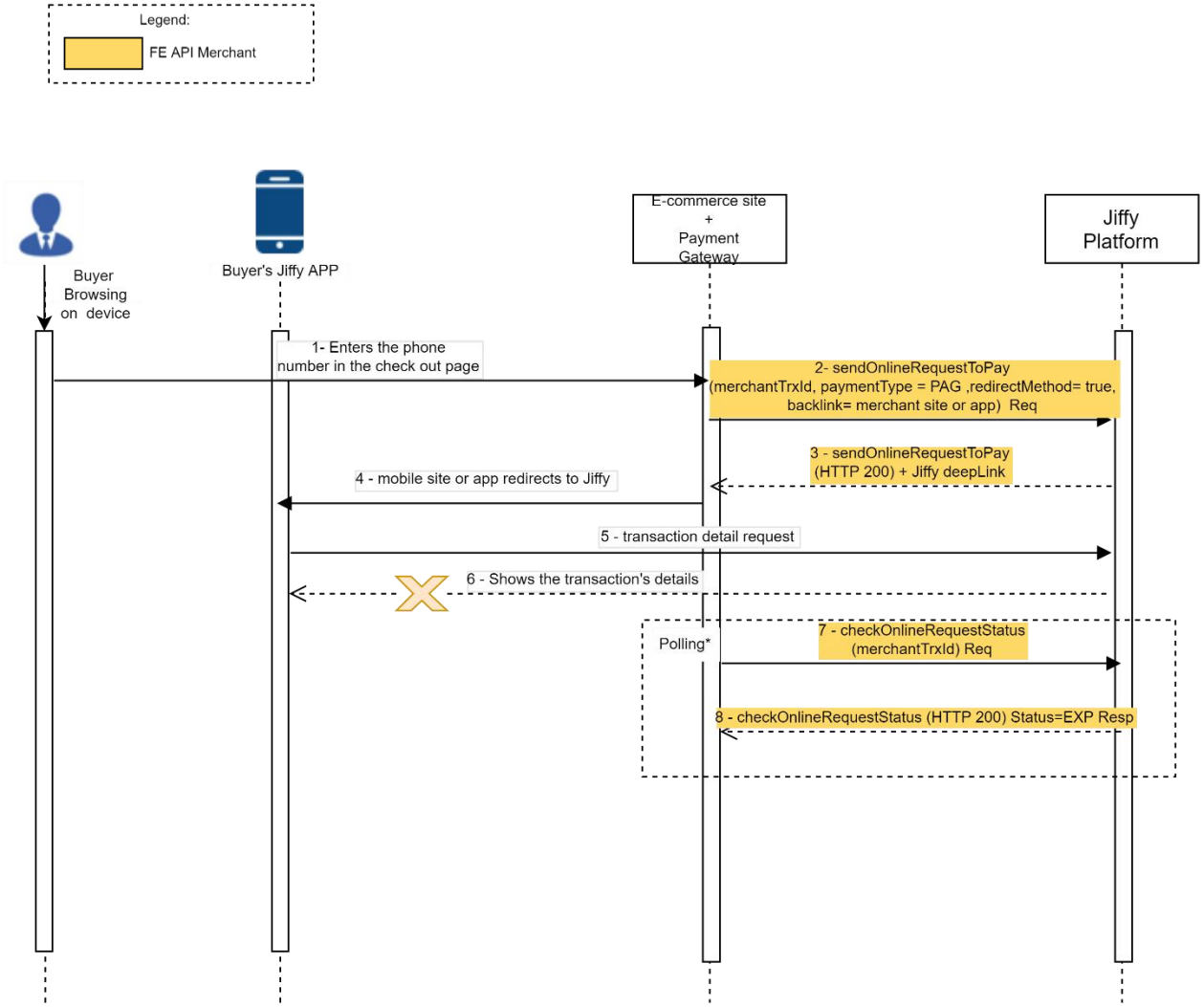
3.2.1.1. Sequence diagram description

1. Buyer is browsing Merchant website, chooses UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow, Merchant website invokes sendOnlineRequestToPay on UAEIPP Overlay Service.
3. An error occurs and UAEIPP Overlay Service replies returning negative response (e.g.: HTTP 401, code :01103, description : "the mobile number is invalid". Error management on e-commerce is up to the merchant).

3.2.2. Unhappy Flow: Buyer's request for transaction details ends with error

This unhappy flow description can be applied:

- Simple e-commerce payments (both redirect and push notification)
- Pre-authorised payments (both redirect and push notification)



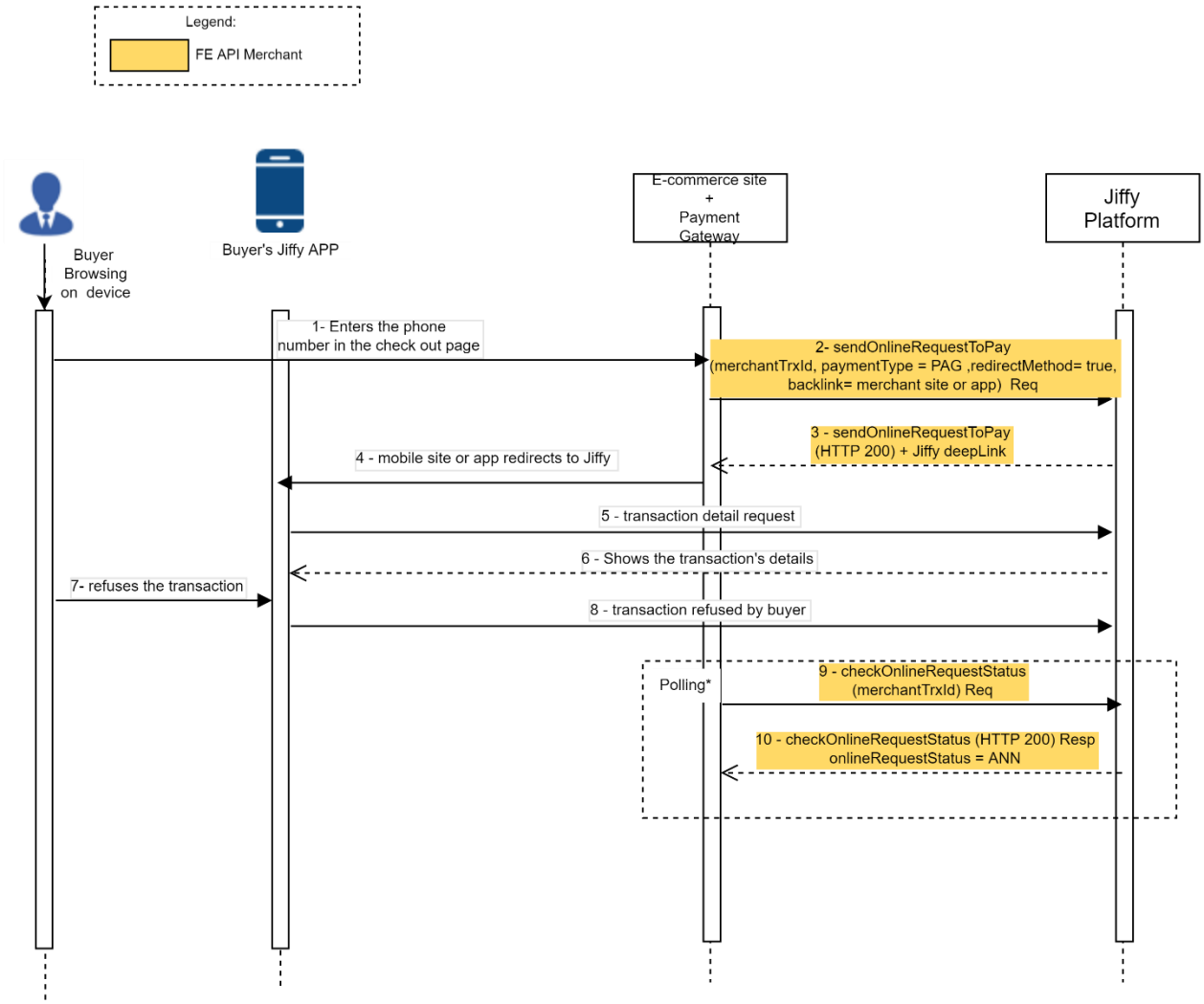
3.2.2.1. Sequence diagram description

1. Buyer is browsing Merchant app or mobile website, chooses UAEIPP Overlay Service as payment method in checkout page, and provides mobile number.
2. In order to trigger payment flow with redirect, Merchant app or mobile website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PAG,
    b. redirectMethod= true
    c. backlink= merchant site or app url,
3. UAEIPP Overlay Service replies by returning UAEIPP Overlay Service deeplink.
4. Merchant app or mobile website redirects buyer to the APP (in case of no redirect, push notification will be sent to UAEIPP Overlay Service App).
5. at launch, UAEIPP Overlay Service APP requests transaction detail to UAEIPP Overlay Service.
6. UAEIPP Overlay Service has problem to provide details to UAEIPP Overlay Service App.
7. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus, until request to pay is not expired, merchant receives response from UAEIPP Overlay Service with status “PND”.
8. After request to pay expires UAEIPP Overlay Service replies with status “EXP” to request at point 7.

3.2.3. Unhappy Flow: Buyer refuses request to pay

This unhappy flow description can be applied:

- Simple e-commerce payments (both redirect and push notification)
- Pre-authorised payments (both redirect and push notification)



3.2.3.1. Sequence Diagram Description

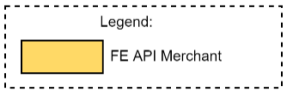
1. Buyer is browsing Merchant app or mobile website, chooses UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow with redirect, Merchant app or mobile website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PAG,
    b. redirectMethod= true
    c. backlink= merchant site or app url,
3. UAEIPP Overlay Service replies by returning UAEIPP Overlay Service deeplink.
4. Merchant app or mobile website redirects buyer to the APP (in case of no redirect, push notification will be sent to UAEIPP Overlay Service App).
5. at launch, UAEIPP Overlay Service APP requests transaction detail to UAEIPP Overlay Service.
6. UAEIPP Overlay Service provides details to UAEIPP Overlay Service App.
7. The buyer refuses the payment.

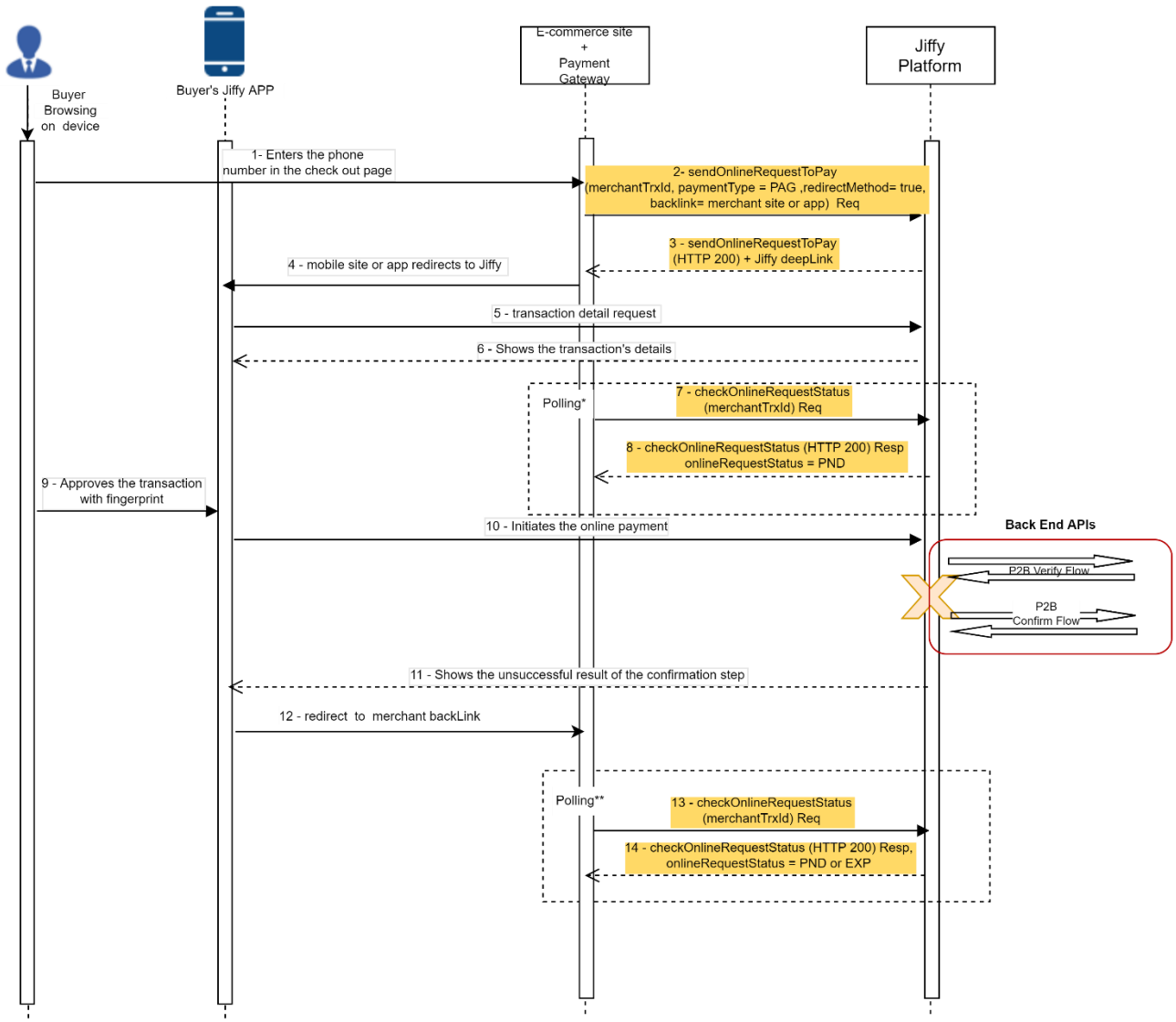
8. The UAEIPP Overlay Service APP communicates with UAEIPP Overlay Service and lets it know that the payment was refused by the buyer.
9. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus, until request to pay is not expired or buyer does not refuse to pay, merchant receives response from UAEIPP Overlay Service with status “PND”.
10. UAEIPP Overlay Service replies with status “ANN”.

3.2.4. Unhappy Flow: Verify (or confirm) flow with participants ends unsuccessfully

This unhappy flow description can be applied:

- Simple e-commerce payments (both redirect and push notification)
- Pre-authorised payments (both redirect and push notification)





3.2.4.1. Sequence diagram description

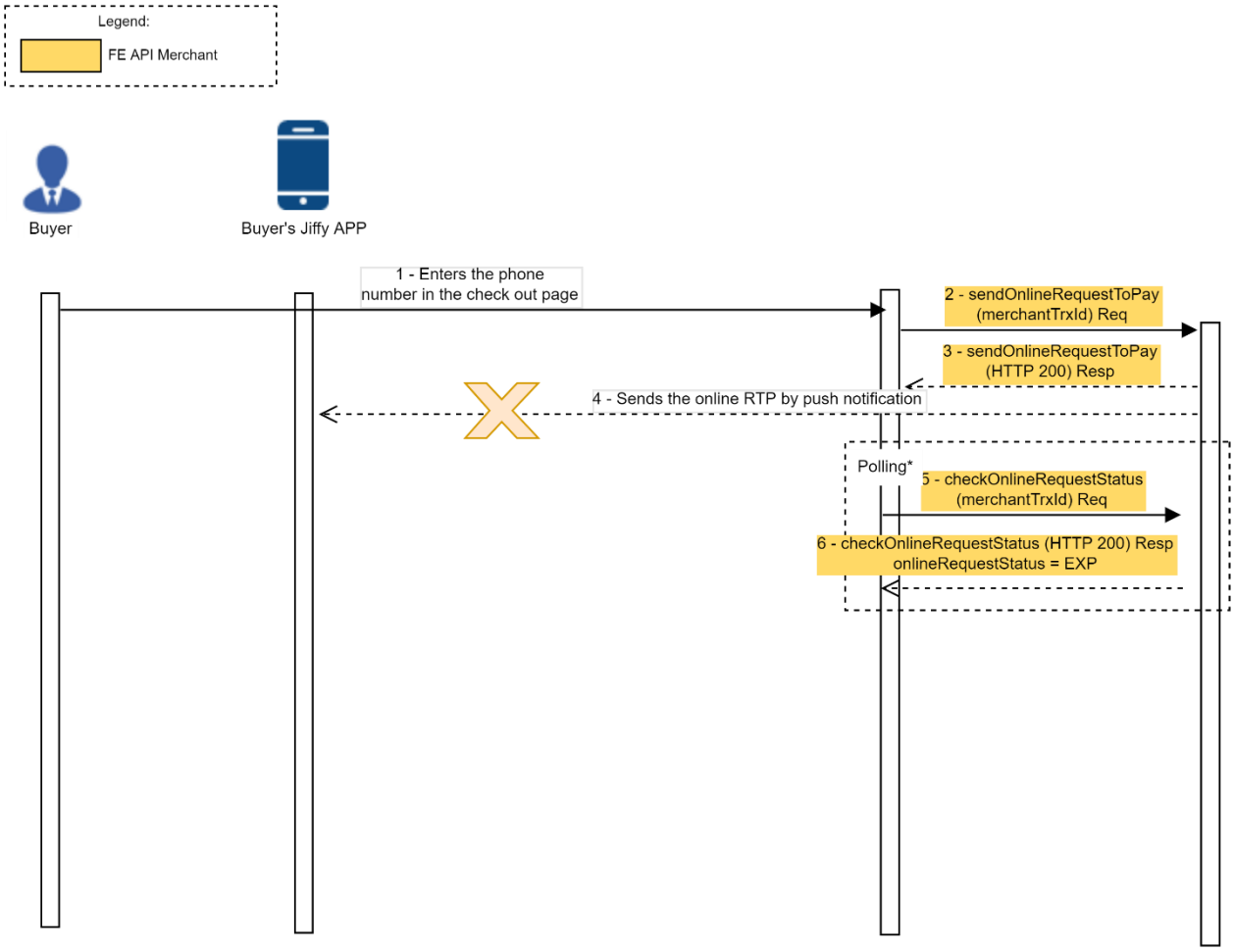
1. Buyer is browsing Merchant app or mobile website, chooses UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow with redirect, Merchant app or mobile website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PAG ,
    b. redirectMethod= true,
    c. backlink= merchant site or app url,
3. UAEIPP Overlay Service replies by returning UAEIPP Overlay Service deeplink.
4. Merchant app or mobile website redirects buyer to UAEIPP Overlay Service APP(in case of no redirect, push notification will be sent to UAEIPP Overlay Service App).
5. at launch, UAEIPP Overlay Service APP requests transaction detail to UAEIPP Overlay Service.
6. UAEIPP Overlay Service provides details to UAEIPP Overlay Service App.
7. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
8. UAEIPP Overlay Service replies with status “PND” (app or mobile website keeps on polling)
9. Buyer authorize payment with SCA on the UAEIPP Overlay Service App.

10. UAEIPP Overlay Service APP triggers UAEIPP Overlay Service to orchestrate payment flows between buyer's participant and merchant participant.
11. Flow has finished unsuccessfully, payment failure message is displayed on UAEIPP Overlay Service APP.
12. UAEIPP Overlay Service APPUAEIPP Overlay Service APP redirects buyer to app or mobile website following backlink provided at point 2. In case request to pay is not expired, buyer can find it in its request to pay list in UAEIPP Overlay Service APPUAEIPP Overlay Service APP and retry to pay it (in case of no redirect, there will not be any redirect back).
13. Merchant app or mobile website keeps on polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus, until request to pay is not expired or buyer does not refuse to pay, merchant receives response from UAEIPP Overlay Service with status "PND".
14. UAEIPP Overlay Service replies with status "PND" (if the online request to pay is still valid) or "EXP" (if online request to pay has expired).

3.2.5. Unhappy Flow: a problem with sending the notification to the app

This unhappy flow description can be applied:

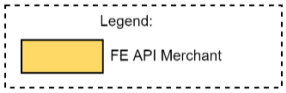
- Simple e-commerce payments push notification
- Pre-authorised payments push notification

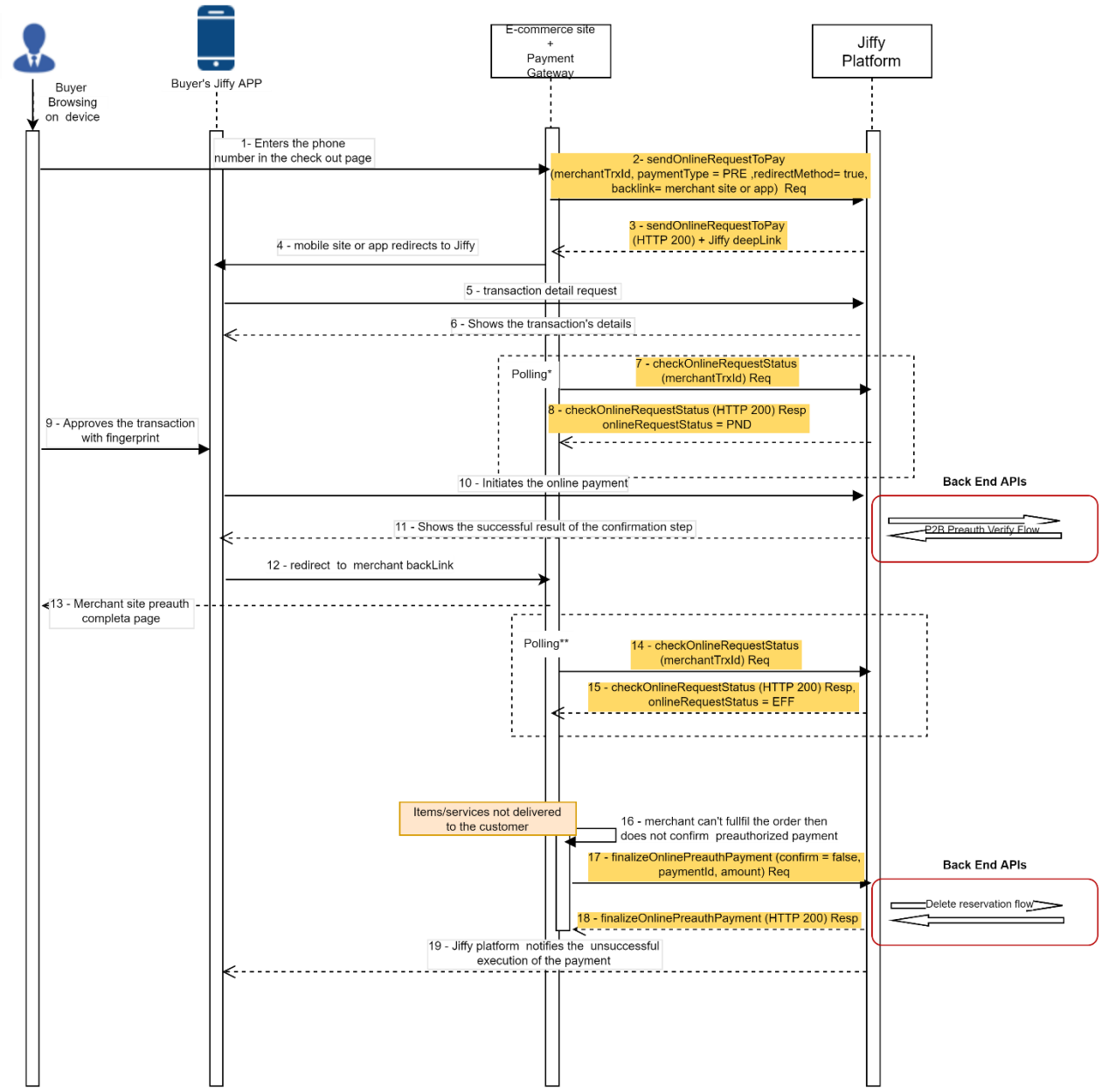


3.2.5.1. Sequence diagram description

1. Buyer is browsing Merchant website, chooses UAEIPP Overlay Service as payment method in checkout page, provides mobile number.
2. In order to trigger payment flow with notification , Merchant website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
   paymentType = PAG,
   redirectMethod= false
3. UAEIPP Overlay Service replies returning positive response, meaning that a request to pay has been created and Merchant website can start polling to retrieve payment status (waiting for buyer to confirm payment).
4. UAEIPP Overlay Service sends a Request to pay notification to UAEIPP Overlay Service APPUAEIPP Overlay Service APP but there is a technical problem with the notification itself.
5. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus, until request to pay is not expired merchant receives response from UAEIPP Overlay Service with status “PND”.
6. UAEIPP Overlay Service replies with status “EXP”.

3.2.6. Unhappy Flow : Merchant doesn’t finalize a preauthorized transaction



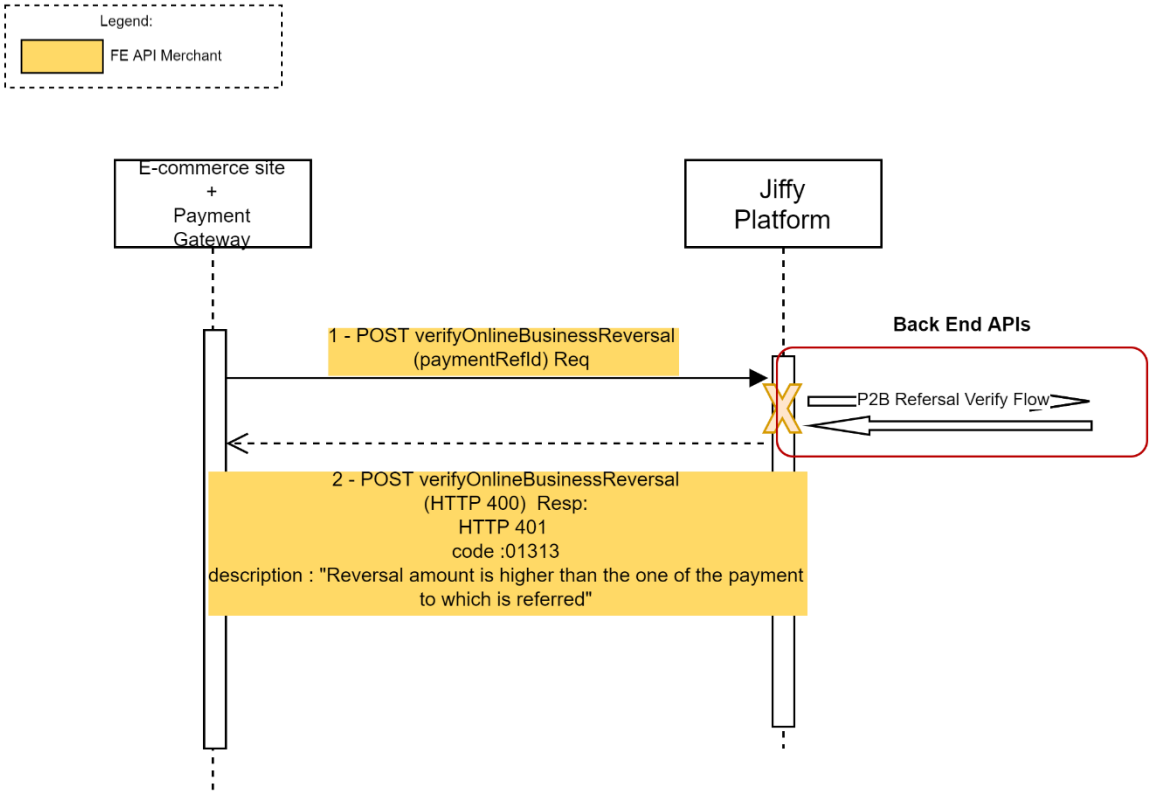


3.2.6.1. Sequence diagram description

1. Buyer is browsing Merchant app or mobile website, chooses UAEIPP Overlay Service as payment method in checkout page, and provides mobile number.
2. In order to trigger payment flow with redirect, Merchant app or mobile website invokes sendOnlineRequestToPay on UAEIPP Overlay Service with :
    a. paymentType = PRE,
    b. redirectMethod= true,
    c. backlink= merchant site or app url,
3. UAEIPP Overlay Service replies returning UAEIPP Overlay Service deeplink.
4. Merchant app or mobile website redirects buyer to UAEIPP Overlay Service APP(in case of no redirect, push notification will be sent to UAEIPP Overlay Service App).
5. At launch, UAEIPP Overlay Service APPUAEIPP Overlay Service APP requests transaction detail to UAEIPP Overlay Service.

6. UAEIPP Overlay Service provides details to UAEIPP Overlay Service APP.
7. Merchant app or mobile website starts polling UAEIPP Overlay Service to get the status of payment requested at point 2 by invoking checkOnlineRequestStatus.
8. UAEIPP Overlay Service replies with status “PND” (app or mobile website keeps on polling).
9. Buyer confirms preauthorization.
10. UAEIPP Overlay Service APP triggers UAEIPP Overlay Service to orchestrate preauthorized payment verify flow between buyer’s participant and merchant participant. At the end of this flow buyer’s account is not debited but funds are reserved (amount of funds reserved is equal to payment.amount , described in 6.1.1.13).
11. Flow has finished successfully, complete preauthorization payment message is displayed on UAEIPP Overlay Service APP.
12. UAEIPP Overlay Service APPUAEIPP Overlay Service APP redirects buyer to app or mobile website following backlink provided at point 2(in case of non-redirect, there won’t be any redirect back).
13. Merchant app or mobile website displays successful preauthorization message.
14. Merchant app or mobile website keeps on polling UAEIPP Overlay Service to get the status of payment. requested at point 2 by invoking checkOnlineRequestStatus.
15. UAEIPP Overlay Service replies with status “EFF” (Buyer confirmed preauthorized payment and UAEIPP Overlay Service payment orchestration went well).
16. After conclusion of business process, merchant may update actual amount to be debited on buyer’s account.
17. Merchant invokes finalizeOnlinePreauthPayment with queryParameter “confirm” set to “false”.
18. UAEIPP Overlay Service returns positive response to merchant to notify that payment has been cancelled.
19. UAEIPP Overlay Service notifies to Buyer’s UAEIPP Overlay Service APP that the preauthorized payment was not successful.

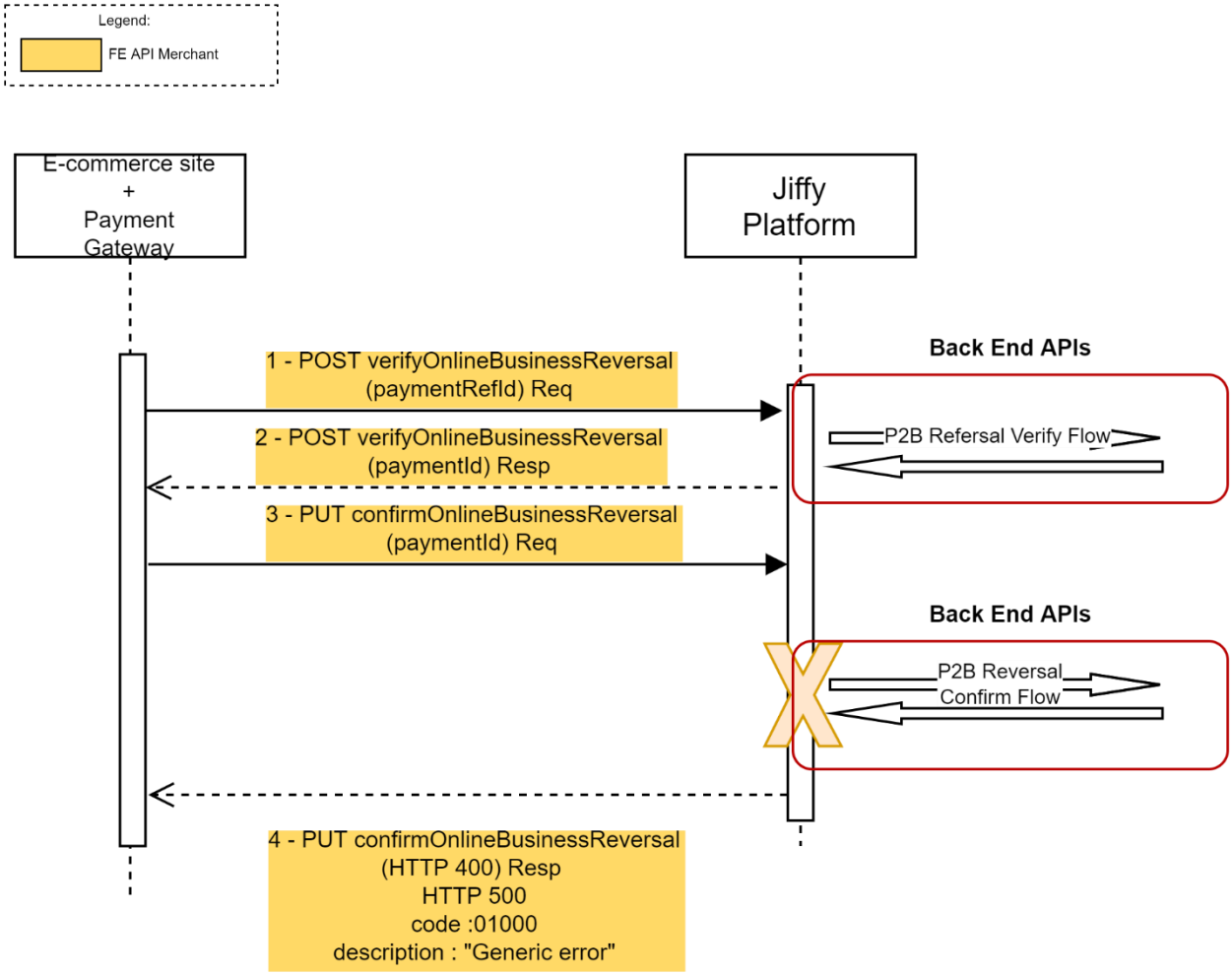
3.2.7. Unhappy Flow: a problem with verify step of a reversal operation



3.2.7.1. **Sequence diagram description**

1. Merchant invokes verifyOnlineBusinessReversal providing reference to payment to revert.
2. In case Payment doesn't exist on the platform or was unsuccessfully processed, UAEIPP Overlay Service provides a negative response, and the reversal operation cannot be completed (e.g.: HTTP 401 code :01313 description : Reversal amount is higher than the one of the payment to which is referred").

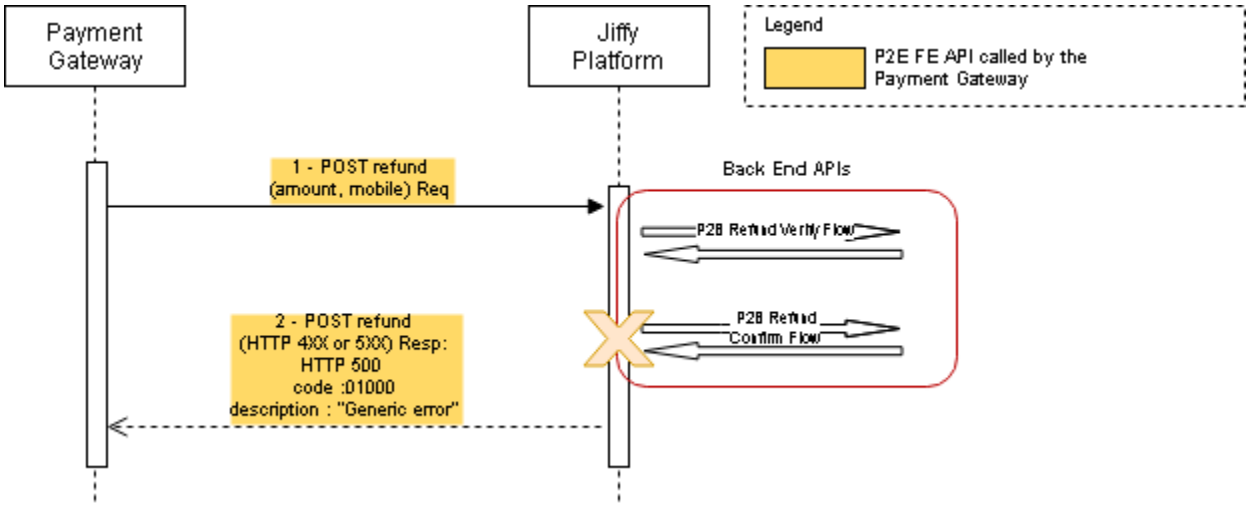
3.2.8. Unhappy Flow: a problem with confirm step of a reversal operation



3.2.8.1. Sequence diagram description

1. Merchant invokes verifyOnlineBusinessReversal providing reference to payment to revert.
2. In case Payment exists and was Refund verify flow with participants has been successfully processed UAEIPP Overlay Service provides a positive response and a new paymentId representing the reversal to be confirmed.
3. Merchant invokes confirmOnlineBusinessReversal providing paymentId received at point 2 to confirm and trigger UAEIPP Overlay Service to orchestrate refund confirm flow between merchant's participant and buyer's participant.
4. Orchestration ends unsuccessfully due to a technical problem related to the confirmation step.

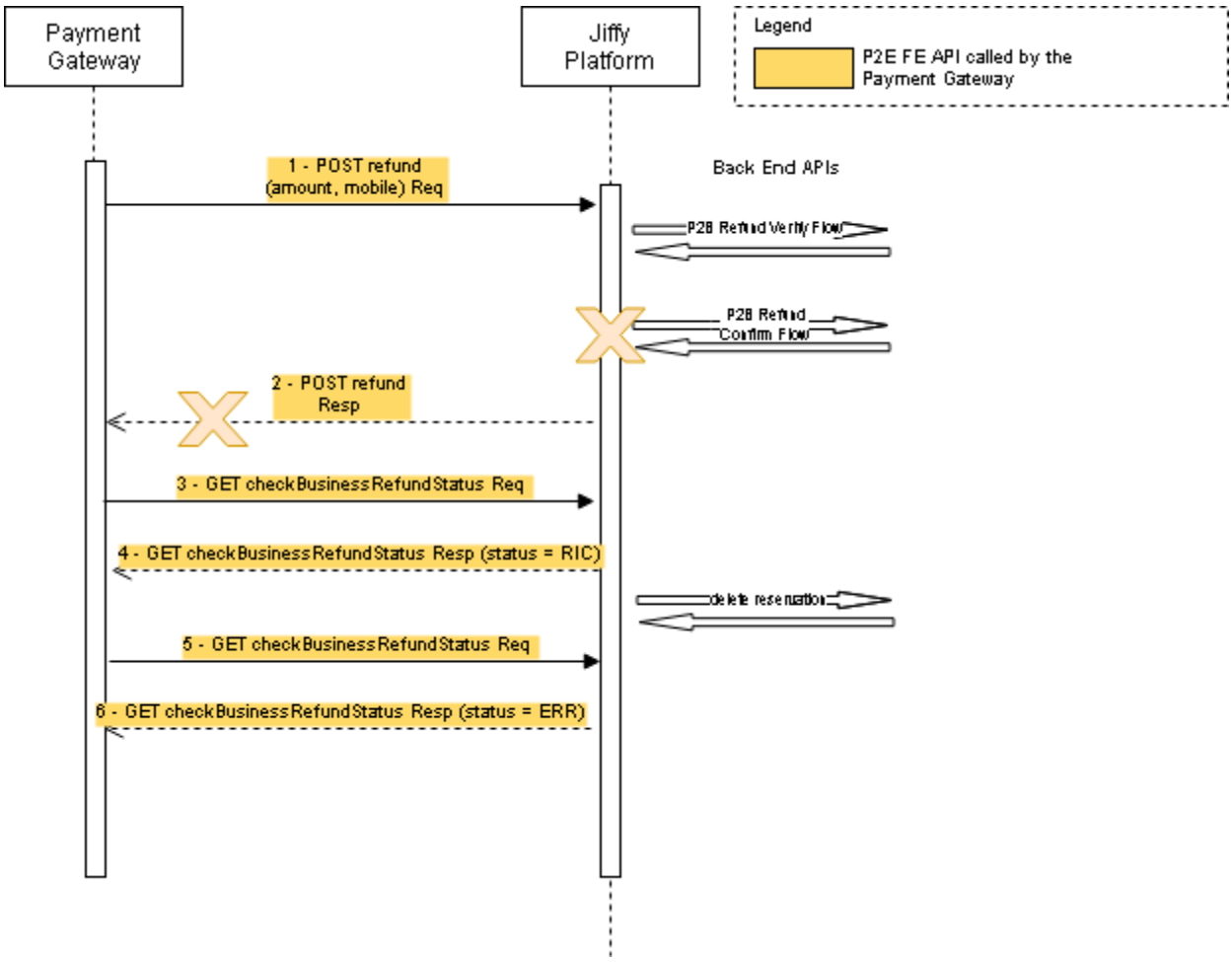
3.2.9. Unhappy Flow: a problem with participant refund flows, after Refund request



3.2.9.1. Sequence diagram description

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's mobile number, without entering any reference to the original payment of the buyer.
2. The orchestration of the refund transaction ends unsuccessfully due to a technical problem.
3. Merchant may execute GET checkBusinessRefundStatus at any point in time to check that status of the reversal is "ERR".

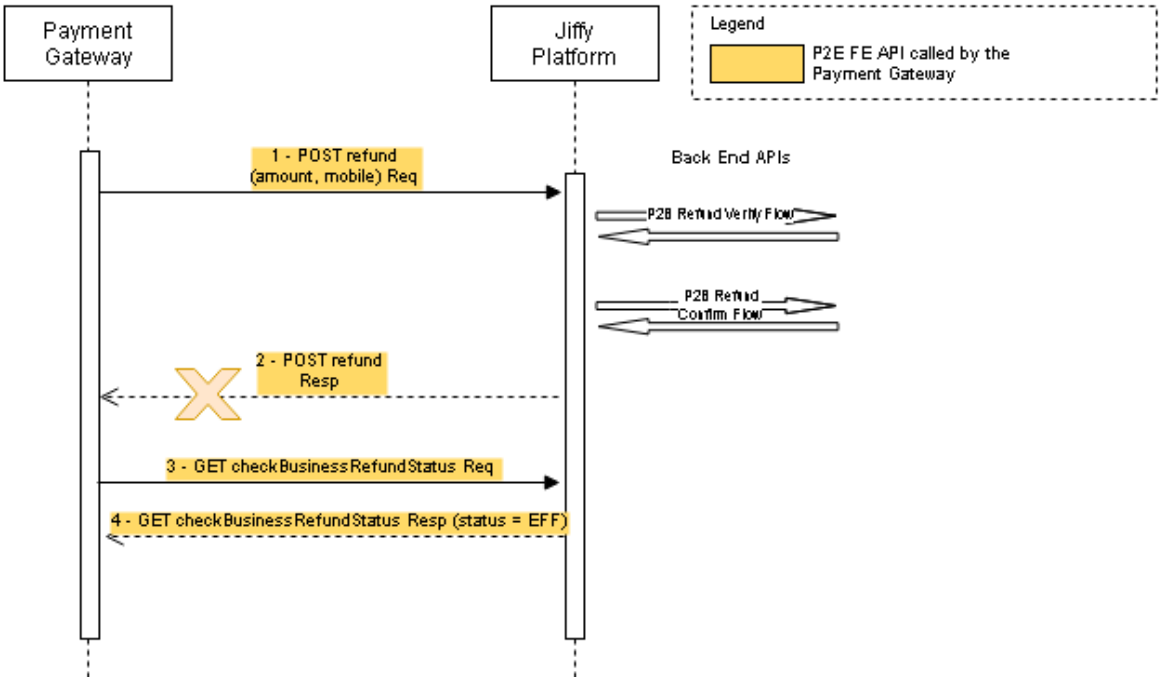
3.2.10. Unhappy Flow: a problem with refund response after a successful verify step



3.2.10.1. Sequence diagram description

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's mobile number, without entering any reference to the original payment of the buyer.
2. After a successful verify step, there is something wrong with the confirm step and also the response of the refund API never arrives.
3. In order to know what has happened with the refund, Payment Gateway invokes checkBusinessRefundStatus.
4. UAEIPP Overlay Service returns response with attribute status= "RIC" that means that the verify step was successfully processed.
5. After fund reserve expiration time passed, UAEIPP Overlay Service triggers delete reservation flow with Merchant's participant.
6. After successful execution of delete reservation flow, when Merchant Payment Gateway invokes checkBusinessRefundStatus, response outlines that refund status is ERR (Bank error) as now it is clear that confirmation flow wasn't successfully processed and Marchant need to repeat the refund operation from the beginning.

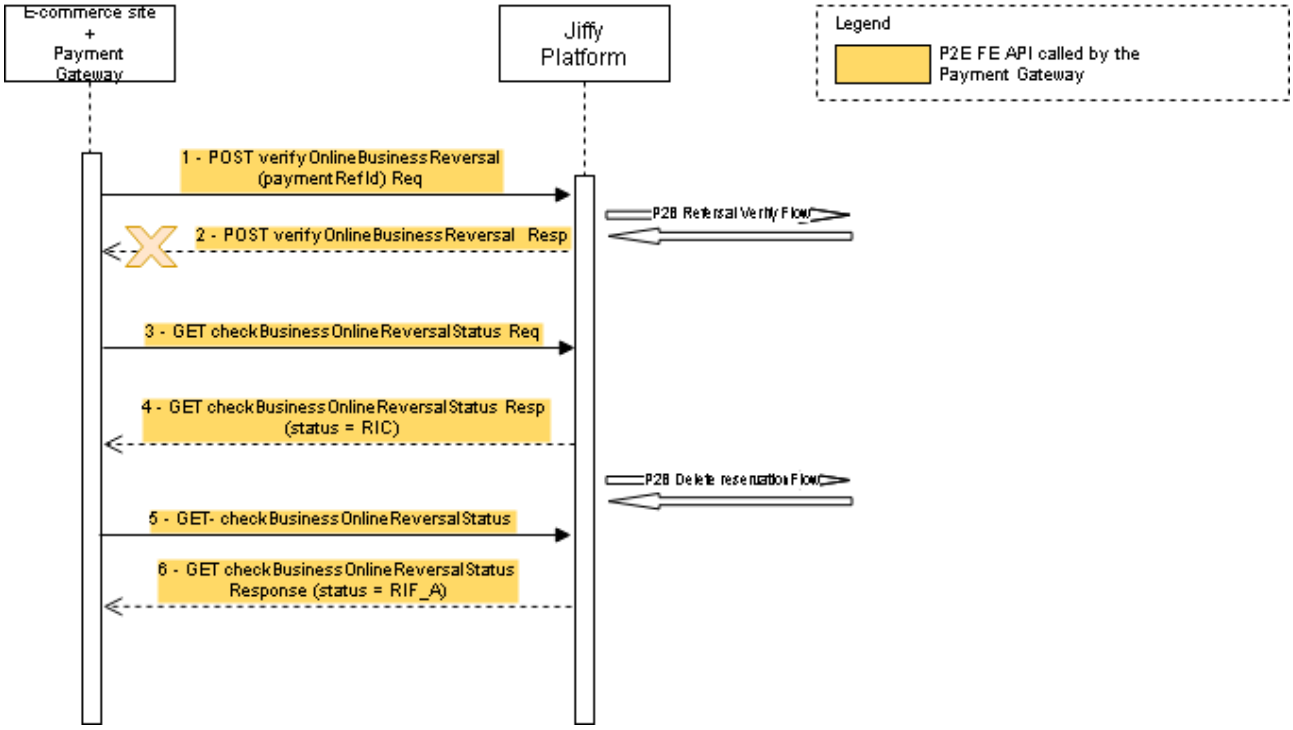
3.2.11. Unhappy Flow: a problem with refund response (timeout)



3.2.11.1. Sequence diagram description

1. Merchant invokes the "refund" API providing the amount to be refunded and the buyer's mobile number, without entering any reference to the original payment of the buyer.
2. After a successful verify and confirm step the response of the refund API never arrives.
3. In order to know what has happened with the refund, Payment Gateway invokes checkBusinessRefundStatus.
4. UAEIPP Overlay Service returns response with attribute status= "EFF" that means that the refund was successfully executed.

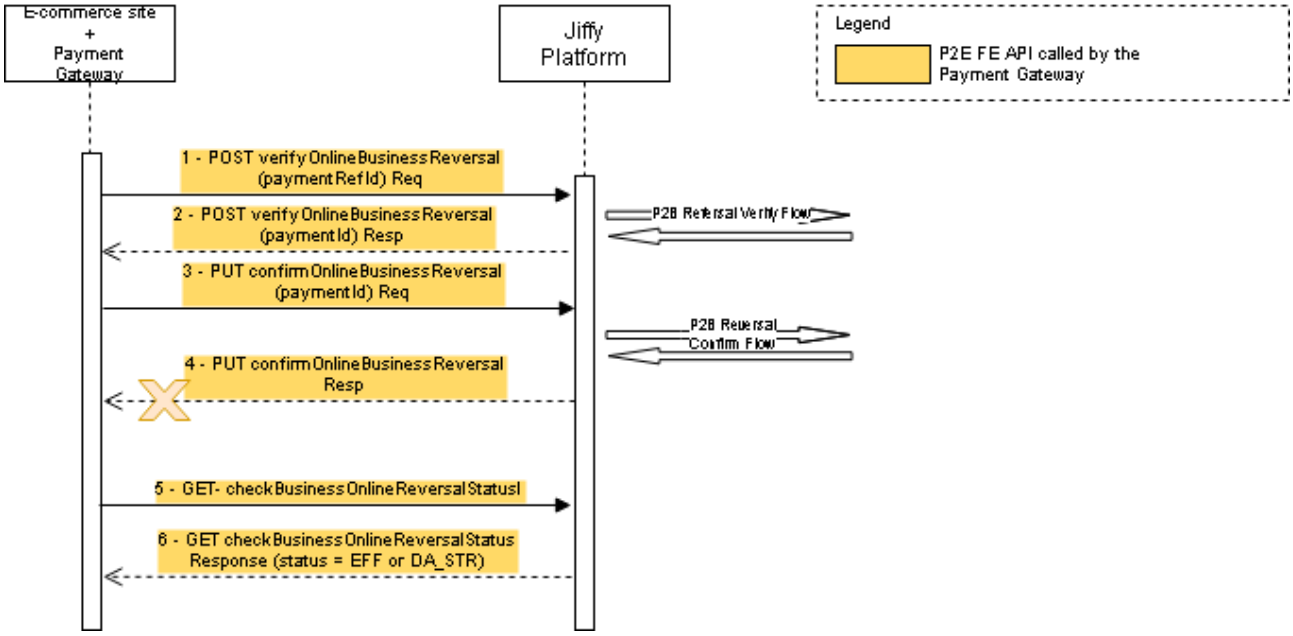
3.2.12. Unhappy Flow: a problem with verifyReversal response (timeout)



3.2.12.1. **Sequence diagram description**

1. Payment Gateway invokes Post verifyOnlineBusinessReversal to verify that a payment exists on the platform and has been previously approved.
2. UAEIPP Overlay Service runs Refund verify flows with participants (both Merchant's and Buyer's one) then replies successful response. For a technical reason (example: timeout) Merchant does not receive any response from UAEIPP Overlay Service.
3. Merchant's Payment Gateway invokes GET checkBusinessOnlineReversalStatusl to check what has happened.
4. UAEIPP Overlay Service returns response with attribute status= "RIC". This means that UAEIPP Overlay Service has received verifyOnlineBusinessReversal request from Merchant and successfully executed Refund Verify flow with participants and is waiting Merchant to confirm reversal operation through PUT confirmOnlineBusinessReversal, but Merchant actually can't invoke confirmOnlineBusinessReversal operation because Merchant did not receive any response at point 2 containing "paymentId" (unique identifier of this reversal operation) to be mandatorily provided in confirmOnlineBusinessReversal request .
5. UAEIPP Overlay Service executes delete reserve flow with Merchant's Participant after verify-reserve expiration time.
6. If the Payment Gateway tries again with GET checkBusinessOnlineReversalStatus, after delete reservation. UAEIPP Overlay Service returns response with attribute status= "RIF_A". This means that the operation has not been processed by UAEIPP Overlay Service due to technical errors. Merchant will not be debited and have to repeat the reversal operation from the beginning.

3.2.13. Unhappy Flow: a problem with confirmReversal response (timeout)



3.2.13.1. **Sequence diagram description**

1. Payment Gateway invokes Post verifyOnlineBusinessReversal to verify that a payment exists on the platform and has been previously approved.
2. UAEIPP Overlay Service runs Refund verify flows with participants (both Merchant's and Buyer's one) then replies successful response.
3. Merchant invokes confirmOnlineBusinessReversal providing paymentId received at point 2 to confirm and trigger UAEIPP Overlay Service to orchestrate refund confirm flow between merchant's participant and buyer's participant.
4. Orchestration ends successfully but for a technical reason the response cannot be communicated to the Payment Gateway
5. Payment Gateway invokes GET checkBusinessOnlineReversalStatus to check what has happened.
6. UAEIPP Overlay Service returns response with attribute status
    a. "EFF": this means that UAEIPP Overlay Service received and processed confirmOnlineBusinessReversal request from Merchant, executing successfully Refund confirm flow with participants. Buyer will be refunded.

       OR

    b. "DA_STR": this means that UAEIPP Overlay Service received and processed confirmOnlineBusinessReversal request from Merchant, but Refund confirm flow with participants is still running (and is going to end successfully within minutes, depending on platform workload). Buyer will be refunded.
        i. Merchant execute again GET checkBusinessOnlineReversalStatus at any point in time to check that status of the reversal will change to "EFF" , to be sure that refund confirm flow with participants has successfully ended.

Sequence description above assumes that Refund confirm flow with participants will be successful, but in case this flow with participants is not successful sequence above may change from point 5 as follows:

5. Payment Gateway invokes GET checkBusinessOnlineReversalStatus to check what has happened.
6. UAEIPP Overlay Service returns response with attribute status

c. “ERR”: this means that, verify Flow with participants ended successfully then UAEIPP Overlay Service received and processed confirmOnlineBusinessReversal request from Merchant, executing not successfully Refund confirm flow with participants. Buyer will not be refunded.

   OR

d. “RIC”: this means that, verify Flow with participants ended successfully then UAEIPP Overlay Service received and processed confirmOnlineBusinessReversal request from Merchant, but Refund confirm flow with participants is still running .
   i. Merchant may execute GET checkBusinessOnlineReversalStatus at any point in time to check that status of the reversal will change to “ERR”.

3.3. Endpoints

Operation	Endpoint	Method	Purpose of the API
sendOnlineRequestToPay	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/request-to-pay	POST	This API allows sending an online request to pay to a customer. From 2024 R3, Overlay Service will support a new RTP flow (PGS) and this API can be substitute by the RTP described in FE API for Participants Channels
checkOnlineRequestStatus	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/request-to-pay/status	GET	This API is called by the Payment Gateway on behalf of the merchant customer to check the execution status of an online request to pay. From 2024 R3, Overlay Service will support a new RTP flow (PGS) and this API can be substitute by the RTP described in FE API for Participants Channels
verifyOnlineBusinessReversal	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/business/reversal	POST	This API allows making the checks on the reversal for the sender and the receiver on P2E transactions, if passed returns the identification code of the reversal transaction.
confirmOnlineBusinessReversal	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/business/reversal	PUT	This API allows confirming a previous verified reversal transaction of a P2E payment, all the data in the request must be consistent with the verify step.
finalizeOnlinePreauthPayment	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/request-to-pay	PUT	This API allows confirming an online pre-authorized payment.
refund	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund	POST	This API allows a merchant to execute refunds.
checkBusinessOnlineReversalStatus	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/business/reversal/status	GET	This API allows checking the execution status of a requested reversal.
checkBusinessRefundStatus	/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund/status	GET	This API allows checking the execution status of a requested refund.

3.4. Common Fields

In this paragraph all the common attributes used in every call are described. They are applied to all the endpoints.

HTTP version supported 1.1

3.4.1. Request

3.4.1.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
groupCode	String Max 5	99999	Y	Functional	Parent participant, Technical Service Provider or Technical Service Provider as Acquirer code of the active merchant. In case of Technical Service Provider or Technical Service Provider as Acquirer all these 3 headers must be also used: 1. providerType 2. participantGroupCode 3. participantBankCode
bankCode	String Max 5	99999	Y	Functional	Sub participant, Technical Service Provider or Technical Service Provider as Acquirer code of the active merchant. In case of Technical Service Provider or Technical Service Provider as Acquirer all these 3 headers must be also used: 1. providerType 2. participantGroupCode 3. participantBankCode

3.4.1.2. Request Header

Name	Type	Sample Value	Req.	Scope	Description
X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique Client RequestID. It must present a constraint that allows distinguishing the calling institution.
language	String 2	EN	N	Security	Code that defines which language to use for result messages. Standard used is the ISO 639-1 2 letter codes.

					Example values: “IT” “EN” “DE”.
timestamp	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}\+\d{2}:\d{2}	2023-09-22T23:50:56.193+01:00	N	Security	Execution date/time of the call to the service. The timeZone in this field can be different of the value included in the field timeZone.
Authorization	String	Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJtZV9UMk9XTmFUZ0JHUk4tVmZ3R3RWcnNBMHUyR2VFc0Z2OEFqNkctcVZ3In0…	Y	Security	Access bearer token
x-jws-signature	String	detached signature	Y	Security	Describe the digital signature or message authentication code (MAC) applied to the Payload.
Content-Type	String Max 16	application/json	N	Functional	Entity header is used to indicate the media type of the resource.
Additional-customer-authorization	String	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IlhGRDVxbXdYeVB5bTk1dGFGR0szIn0.eyJpc3MiOiJodHRwczovL2p3dC1pZHAuamlmZnkuc2lhLmV1IiwianRpIjoidG52NXpCMFdiM2FMQ2dYa2ZFTEUiLCJoY2kiOiIyNzQwQUQ3MDMzM0FBMjY3M0YyOURDMThGNTFBMjdGMEVFMjM3QTRBMDQwREE5MDgyMjdCNTkwNkU0NTI1RkVGIiwiZXhwIjoxNjAwMjYzMTA2ODM4LCJvcG4iOiJQMlAiLCJhbXQiOiI1MDAwIiwiY2N5IjoiRVVSIiwiYWlkIjoiODcyMzM4NDcxNSIsInNpYiI6IklFNjJCT0ZJOTAwMDE3OTI4NjUxIiwicmliIjoiSUUxMEJPRkk5MDAwMTc3MjU4ODciLCJzcG4iOiIrMzkzMzUxMjEyMTIzIiwicnBuIjoiKzM5MzM0NzQ4Nzg3NiIsIm9zZCI6IklPUyIsIm9zdiI6IjE0IiwibXNnIjoiWW91IGFyZSBzZW5kaW5nIDUwLjAwIEVVUiB0byBNYXIqKiogUG9sKioqIHdpdGggeW91ciBhY2NvdW50ICoqKio4NjUxIn0.s4yL6NTf3AbduAcYdYf3OI2F43orNwWibRPjLKpquhI	N	Security	Additional optional authorization data generated by Merchant’s clients (to authorize refund/reversal/disbursement) and forwarded by the platform to Participant Participants. This field is structured as a AJWT
providerType	String	ACQUIRER	C	Functional	Code that defines the caller Technical Service Provider Type. Expected value: 1. ACQUIRER 2. PROVIDER This field has to be populated with “ACQUIRER” whether the API is called by a Technical Service Provider as Acquirer and with “PROVIDER” in case it is called by a Technical Service

					Provider (see definition of the actors in the paragraph 1.2.1)
participantGroupCode	String Max 5	99999	C	Functional	Parent bank code of the participant. This field has to be populated with the Participant group code if the API is called by a Technical Service Provider as acquirer or Technical Service Provider (see Appendix). In case of Participants calling the API, the field is not expected.
participantBankCode	String Max 5	99999	C	Functional	Sub bank code of the participant. This field has to be populated with the Participant bank code if the API is called by a Technical Service Provider as acquirer or Technical Service Provider (see Appendix). In case of Participants calling the API, the field is not expected.

3.4.2. Response

3.4.2.1. **Header**

Name	Type	Sample Value	Req.	Scope	Description
x-jws-signature	String	detached signature	Y	Security	Describe the digital signature or message authentication code (MAC) applied to the Payload.
Content-Type	String Max 16	application/json	N	Functional	Entity header is used to indicate the media type of the resource.
X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

3.4.2.2. Common Response Body

Name	Type	Sample Value	Req.	Scope	Description
result	Object	NA	Y	Functional	This object contains information on the result of the execution of the service.
result.code	String Max 5	00000	Y	Functional	Result code.
result.result	Boolean	True	Y	Functional	Result of the request (true=positive, false=negative).
result.message	String Max 100	Positive result	Y	Functional	Represents the description of the result.
result.X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

3.5. Error handling

3.5.1. Common response header

Name	Type	Sample Value	Req.	Scope	Description
x-jws-signature	String	detached signature	Y	Security	Describe the digital signature or message authentication code (MAC) applied to the Payload.
Content-Type	String Max 16	application/json	N	Functional	Entity header is used to indicate the media type of the resource.
X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

3.5.2. Common Response Body

Name	Type	Sample Value	Req.		Description

				Scope	
result	Object	NA	Y	Functional	This object contains information on the result of the execution of the service.
result.code	String Max 5	00000	Y	Functional	Result code.
result.result	Boolean (True/False)	True	Y	Functional	Result of the request. Expected values: • True • False
result.message	String Max 150	Positive result	Y	Functional	Represents the description of the result.
result.X-Request-ID	String Max 100	b45d94ce-57ca-4e05-b67b-de11e8799fed	Y	Functional	Unique identifier of the invocation. It takes on the same value as the result - X-Request-ID present in the respective request.

3.6. [POST] Send Online Request To Pay

This chapter describes the additional fields available in the endpoint related to the “sendOnlineRequestToPay” service; whereas common fields are applied to all the endpoints.

3.6.1. Description

This API allows the Payment Gateway to send an online request to pay to the buyer on behalf of the merchant customer. The buyer involved in the online transaction is identified by his/her mobile number that needs to be specified in the request of this API or by his/her proxy (email or document-id). In addition, the Payment Gateway must provide its identification code of the transaction represented in this API by the “merchantTrxId”.

This service can be called also for sending online requests to pay for preauthorized transactions by inserting the value “PRE” in the “paymentType” field.

If a customer rejects three requests from the same merchant within a 10-minute timeframe, that merchant will be temporarily blacklisted. During this period, the merchant will lose the ability to send further payment requests to the specific customer for 10 minutes. This control measure is implemented to protect the customer and mitigate excessive spam of money requests. When this scenario occurs the error “01049 - Merchant blocked for payment of this specific buyer” is returned by the API.

The RTP can be generated from Physical or Online Shop.

3.6.2. Business Scenario

The merchant customer needs to send an online payment requests to his/her customer.

3.6.2.1. Direct outcome

- Online request to pay sent to the buyer.

3.6.2.2. Related outcome

N.A

3.6.3. URL

POST

/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/request-to-pay

3.6.4. Operation

sendOnlineRequestToPay

3.6.5. Request

3.6.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identifier code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant customer ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.6.5.2. Request Header

For the request headers, refer to the common ones described in the paragraph Common Fields.

3.6.5.3. Request Body

Name	Type	Sample Value	Req.	Description
merchantTrxId	String Max 50	123456	Y	Transaction identifier for this operation created by the Payment Gateway.
amount	Number Max 9 (Decimal part: 2)	70	Y	Payment amount.
currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
mobile	String Max 30	+971837892848	C	Mobile number of the request to pay's recipient.
proxy	Object	N.A.	C	Object related to the proxy used by the buyer to receive the request to pay to eb used alternatively to the mobile number.
proxy.type	Enum	email	Y	Type of the proxy of the buyer to be used for receiving RTPs. Expected values: email; document-id

				proxy.type is as per enrolment.document-id and email are case insensitive
proxy.value	String	john.wick@gmail.com	Y	Value of the proxy.
reason	String Max 140	Soccer shoes	N	Payment reason.
paymentCategory	String Max 15	01	N	Payment Category of the payment: Expected values: 1. "01" = Bill payment (default) 2. "02" = Prepaid top-up 3. "03" = "Purchase"
paymentType	String Max 15	PAG	Y	Category of the payment: Expected values: 1. "PAG" = Standard payment (default) 2. "PRE" = Preauthorized payment in Case paymentCategory is 01 "Bill payment" MUST be set to PAG
backMethod	String Max 7	URL_WEB	C	Type of the back function method after payment completion to be addressed to the buyer. Expected values are: • "URL_WEB" = if the method returns a web url • "APP_MER" = if the method returns to the merchant's APP For cross device case this field will not be filled. in Case paymentCategory is 01 "Bill payment" this field should not be passed then it will be ignored
backLink	String	https://www.onlineshopping.com/pay	C	Based on the value of the field 'backMethod', this field shows the link that allows the buyer to go back to the merchant customer's web site (backMethod = URL_WEB) or to the online merchant customer's APP (backMethod = APP_MER). For cross device case this field will not be filled. in Case paymentCategory is 01 "Bill payment" this field should not be passed then it will be ignored

redirectMethod	Boolean	true	Y	Flag that indicates if the redirect method to the Mobile APP is through the deep-link (true) or by push notification (false). in Case paymentCategory is 01 “Bill payment” this must not be passed this must be set to false
quickPaymentFlag	Boolean	false	N	Flag that indicates if the online request to pay is linked to a quick payment (true) or not (false). Always FALSE.
validShippingAddress	Boolean	true	N	This flag is used in case of a Quick Payment and indicates if the default shipping address received in response to the API.
shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by UAEIPP Overlay Service.
cashDeskId	Number Min 8 Max 16	10000001	Y	Identifier of the cash desk of the shop.
categoryPurpose	String Max 35	CCP	Y	Category Purpose Code in a proprietary form. For the list of allowed values, please refer to the dedicated document From 2024 R4

The following matrix explains fields redirecMethod, backMethod, backLink and deepLink work in different device type cases.

Device type	**Request field population**			**Response field population**	**Note**
	redirectMethod	backMethod	backLink	Deeplink in Response	
Different Device (E-commerce web site to the app)	False	N.A. A push notification will be sent to the buyer’s mobile		N.A. A push notification will be sent to the buyer’s mobile	
Same device (Mobile) - App E-commerce to the app	True	APP_MER	Filled with ecommerce app deeplink	Returned	

Same device (Mobile) - mobile E-commerce web site to the app	True	URL_WEB	Filled with URL Web of Ecommerce Website	Returned	
Same device (Mobile) - mobile E-commerce web site /app E-commerce to the app	False	N.A. A push notification will be sent to the buyer's mobile		N.A. A push notification will be sent to the buyer's mobile	In case e-commerce does not implement redirect method

3.6.6. Response

3.6.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.**Error! Reference s ource not found.**

3.6.6.2. Response Body

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Description
deepLink	String(255)	https://p.bcmt.en/	C	Deep link to the buyer's mobile APP if the "redirectMethod" field is set as "true".
shippingAddressId	Integer Max 16	23356	N	Unique identifier code of the address generated by the platform, used only for Quick Payments Scenario.

3.6.6.3. Result code

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01017	The customer is not registered	FALSE	401
01027	Merchant not registered	FALSE	401
01034	The amount is not valid	FALSE	401
01037	Transaction id already used	FALSE	401

01044	Shop not found	FALSE	401
01049	Merchant blocked for payment of this specific buyer	FALSE	401
01076	Bank account not found	FALSE	401
01080	There is none active mobile for the customer	FALSE	401
01130	It is not possible to send an RTP to the same number as yours	FALSE	401
01146	Payment reason not valid	FALSE	401
01247	No active online shop found for the merchant	FALSE	401
01248	The customer has disabled the reception of online RTP	FALSE	401
01281	Currency mismatch, payment currency: {0}, bank account currency: {1}	FALSE	400
01300	Merchant not found	FALSE	401
01301	Shop not found	FALSE	401
01302	Cash Desk not found	FALSE	401
01330	Shipping address not found	FALSE	401
01346	Currency mismatch, bank account currency: {0}, payment currency: {1}	FALSE	400
01365	Bank account blocked, it is impossible to proceed with the request	FALSE	401
01368	Email not found	FALSE	401
01371	Merchant not authorized	FALSE	401
01383	Creditor not found	FALSE	400
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
02002	The field {0} is not filled in or has an invalid format	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
02005	The field {0} doesn't match any of the expected values	FALSE	400
02010	At least one of the following fields is mandatory: {0}	FALSE	400
03004	Merchant not enabled for payment method:{0}	FALSE	400
03016	The operation is not allowed	FALSE	400
03018	The bank is not enabled	FALSE	400
04004	no {0} proxy customer found	FALSE	400

04005	mobile numbers mismatch	FALSE	400
04008	{0} is not a valid proxy type	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

3.7. [GET] Check Online Request Status

This chapter describes the additional fields available in the endpoint related to the "checkOnlineRequestStatus" service; whereas common fields are applied to all the endpoints.

3.7.1. Description

This API is called in polling by the Payment Gateway on behalf of the merchant customer to check the execution status of an online request to pay uniquely identified by its merchantTrxId after the confirmation step of the payment on buyer's side.

Below the list of the possible values of the RTP's status:

- "EFF" - Request paid by the buyer;
- "PND" - Request in pending status to be paid by the buyer;
- "EXP" - Request expired;
- "ANN_A" - Request cancelled automatically by the system. This status can be returned to the Merchant in case his shop has been disabled.
- "ANN_M" - Request cancelled manually by the buyer;

3.7.2. Business scenario

The merchant customer needs to check the status of an online request to pay.

3.7.2.1. Direct outcome

- The status of the online request to pay is delivered to the merchant customer through the Payment Gateway.

3.7.2.2. Related outcome

N.A.

3.7.3. URL

GET

/inquiry-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/request-to-pay/status

3.7.4. Operation

checkOnlineRequestToPayStatus

3.7.5. Request

3.7.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identifier code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant customer ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common fields.

3.7.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
merchantTrxId	String	2333444	Y	Transaction identifier for this operation created by the Payment Gateway.

3.7.5.3. Request Header

For the request headers, refer to the common ones described in the paragraph Common Fields.

3.7.6. Response

3.7.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

3.7.6.2. Response Body

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Description
onlineRequestStatus	String	EFF	Y	Status of the online request to pay. Expected values: • "**EFF**" - Request paid by the buyer; • "**PND**" - Request in pending status to be paid by the buyer; • "**EXP**" - Request expired; • "**ANN_A**" - Request cancelled automatically by the system; • "**ANN_M**" - Request cancelled manually by the buyer;
paymentId	Number Max 16	123456	C	Internal identifier of the payment generated by the system. PaymentId is returned when the status shift to "EFF" or in the case of a preauthorised payment when the status is "PND" but the buyer has approved the payment. This should be saved by Merchant because must be provided as value of attribute "paymentRefId" in verifyOnlineBusinessReversal Operation
lastUpdateDate	Date	2020-08-03T17:45:10.000+00:00	C	Date in which RTP shifted to "EFF" status.
paymentIdSct	string	0003P62000030934171	C	This value is the unique reference for the transaction. It is returned only if the Status is "EFF"

3.7.6.3. **Result code**

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01027	Merchant not registered	FALSE	401
01228	No RTPs found	FALSE	400
01250	The RTP is not associated with the merchant	FALSE	401
01371	Merchant not authorized	FALSE	401
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

3.8. [POST] Verify Online Business Reversal

This chapter describes the additional fields available in the endpoint related to the "verifyOnlineBusinessReversal" service; whereas common fields are applied to all the endpoints.

3.8.1. Description

This API allows starting a reversal for a previous P2E payment received. It makes the participant account and funds checks on the reversal for the merchant customer and the buyer, likewise other verify steps.

For any P2E payment only a single reversal payment can be made.

In case of positive result, the API returns the paymentId of the reversal: this id must be used to confirm the reversal through the dedicated API (confirmOnlineBusinessReversal).

3.8.2. Business Scenario

The merchant customer needs to initiate a reversal payment to his/her customer.

3.8.2.1. Direct outcome

- The reversal process is initiated, the participant accounts and funds checks on merchant customer's and buyer's sides are completed.

The reversed amount can be equal or lower than the original payment amount (it cannot be greater).

3.8.2.2. Related outcome

N.A

3.8.3. URL

POST
/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/business/reversal

3.8.4. Operation

verifyOnlineBusinessReversal

3.8.5. Request

3.8.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identifier code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant customer ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.8.5.2. Request Header

Name	Type	Sample Value	Req.	Scope	Description
appId	String 64	03704576A8D81E96BF0C5D8E8953F4298842EC7A1803884A3EEC96C11C290099	Y	Security	Unique identification code generated by the Participant's Mobile Server related to the APP installed by the merchant. This field must be populated with the appId of the channel from which the merchant is operating
deviceOSVersion	String Max 255	Ios 10.5.26	N	Security	Operating system of the device.
deviceModel	String Max 255	ABCD1234	N	Security	Code associated with the device model.
language	String 2	EN	N	Security	Code that defines which language to use for result messages. Standard used is the ISO 639-1 2 letter codes. Example values: "IT" "EN" "DE".
deviceId	String Max 255	ABCD1234	N	Security	Unique identification code of the device.
deviceIpAddress	String Max 255	192.168.1.1	N	Security	IP address associated with the device.
country	String 2	AE	N	Security	Sender Country Code defined according to the standard ISO-3166 Alpha2.
timestamp	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}\+\d{2}:\d{2}	2023-09-22T23:50:56.193+01:00	N	Security	Execution date/time of the call to the service. The timeZone in this field can be different of the value included in the field timeZone.
timeZone	String PATTERN: +hh:mm or -hh:mm	+01:00	N	Security	Time zone on sender's side registered by the device. This Timezone can be different of the Timezone included in Timestamp field.

For the remaining request headers, refer to the common ones described in the paragraph Common Fields.

3.8.5.3. Request Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.amount	Number Max 9	70	Y	Payment amount

	(Decimal part: 2)			
payment.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
payment.reason	String Max 140	Soccer shoes	N	Payment reason
payment.paymentRefId	Number Max 16	014455	Y	Identifier of the payment to revert. This identifier value is related to the identifier generated by the system and returned to merchant in response body of the operation checkOnlineRequestToPayStatus (paymentId).
payment.shopId	Number Min 5 Max 16	10001	N	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	N	Identifier of the cash desk of the shop.
payment.merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Payment Gateway.
payment.merchantTrxRefId	String Max 50	123456	Y	Transaction identifier for the original payment to be refunded generated by the Payment Gateway.

3.8.6. Response

3.8.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

3.8.6.2. Response Body

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	123456	Y	Identifier of the payment related to the reversal transaction.
payment.amount	Number Max 9 (Decimal part: 2)	70	E	Payment amount
payment.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217
payment.fees	Number	0	N	Transaction fees, additional charge of the service expressed in euro. Its value is set to zero by default. Reserved for future use, it must not be implemented.

payment.totalAmt	Number Max 9 (Decimal part: 2)	70	N	Total fees and amount. Reserved for future use, the value will be always the same as payment.amount.
payment.reason	String Max 140	Soccer shoes	E	Payment reason
payment.paymentRefId	Number Max 16	22333	E	Identifier of the payment to refund.
payment.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	E	Identifier of the cash desk of the shop.

3.8.6.3. **Result code**

CODE	MESSAGE	RESULT	HTTP Code	
00000	Positive result	TRUE	200	
00023	Payment taken in charge, check the correct conclusion	TRUE	200	
00045	RTP cancelled	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01027	Merchant not registered	FALSE	401	
01030	Communication Problems with Gateway	FALSE	401	
01034	The amount is not valid	FALSE	401	
01041	Invalid QR code	FALSE	401	
01044	Shop not found	FALSE	401	
01047	The payment to be finalized does not match the payment initiated	FALSE	401	
01054	Payment refused, try again	FALSE	401	
01061	The operation is not allowed	FALSE	401	
01067	No payments found	FALSE	401	
01076	Bank account not found	FALSE	401	
01144	The recipient has no receiving bank account set up	FALSE	401	
01146	Payment reason not valid	FALSE	401	

01191	Bank account blocked	FALSE	401	
01228	No RTPs found	FALSE	400	
01255	The reversal is not coherent with the payment	FALSE	400	Mismatch of data between reversal and original payment
01257	The transactionRefId used is not coherent with the transactionId of the payment	FALSE	400	
01259	Transaction id already used for the channel	FALSE	400	
01281	Currency mismatch, payment currency: {0}, bank account currency: {1}	FALSE	400	
01300	Merchant not found	FALSE	401	
01301	Shop not found	FALSE	401	
01302	Cash Desk not found	FALSE	401	
01313	The reversal amount is higher than the one of the payment to which is referred	FALSE	400	
01321	The payment is not reversible	FALSE	400	
01322	The payment was not executed to the selected merchant	FALSE	400	
01323	The payment was not executed to the selected shop	FALSE	400	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03004	Merchant not enabled for payment method:{0}	FALSE	400	
03018	The bank is not enabled	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	
432	Country not authorised	FALSE	400	From 2024 R2

433	Country information mandatory for the payment	FALSE	400	From 2024 R2

3.9. [PUT] Confirm Online Business Reversal

This chapter describes the additional fields available in the endpoint related to the “confirmOnlineBusinessReversal” service; whereas common fields are applied to all the endpoints.

3.9.1. Description

This API allows confirming a previous verified reversal transaction of a P2E payment, all the data in the request must be consistent with the verify step. The “paymentId” field of the request has to be filled with the identifier of the payment generated by the previous API “verifyOnlineBusinessReversal”.

3.9.2. Business scenario

The merchant customer needs to finalize a reversal operation, after the verifyOnlineBusinessReversal service.

3.9.2.1. Direct outcome

- The merchant customer confirms the reversal transaction.

3.9.2.2. Related outcome

N.A.

3.9.3. URL

PUT

/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/business/reversal

3.9.4. Operation

confirmOnlineBusinessReversal

3.9.5. Request

3.9.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identifier code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant customer ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.9.5.2. Request Header

Name	Type	Sample Value	Req.	Scope	Description
appId	String 64	03704576A8D81E96BF0C5D8E8953F4298842EC7A1803884A3EEC96C11C290099	Y	Security	Unique identification code generated by the Participant's Mobile Server related to the APP installed by the merchant. This field must be populated with the appId of the channel from which the merchant is operating
deviceOSVersion	String Max 255	Ios 10.5.26	N	Security	Operating system of the device.
deviceModel	String Max 255	ABCD1234	N	Security	Code associated with the device model.
language	String 2	EN	N	Security	Code that defines which language to use for result messages. Standard used is the ISO 639-1 2 letter codes. Example values: "IT" "EN" "DE".
deviceId	String Max 255	ABCD1234	N	Security	Unique identification code of the device.
deviceIpAddress	String Max 255	192.168.1.1	N	Security	IP address associated with the device.
country	String 2	AE	N	Security	Sender Country Code defined according to the standard ISO-3166 Alpha2.
timestamp	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}\+\d{2}:\d{2}	2023-09-22T23:50:56.193+01:00	N	Security	Execution date/time of the call to the service. The timeZone in this field can be different of the value included in the field timeZone.
timeZone	String PATTERN: +hh:mm or -hh:mm	+01:00	N	Security	Time zone on sender's side registered by the device. This Timezone can be different of the Timezone included in Timestamp field.

For the remaining request headers, refer to the common ones described in the paragraph Common Fields.

3.9.5.3. Request Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	22333444	Y	Identifier of the payment, returned by the verifyOnlineBusinessReversal API.

payment.amount	Number Max 9 (Decimal part: 2)	70	Y	Payment amount
payment.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
payment.reason	String Max 140	Soccer shoes	N	Payment reason
payment.paymentRefId	Number Max 16	014455	Y	Identifier of the payment to divert. This datum is related to the identifier generated by the system (paymentId) of the transaction.
payment.shopId	Number Min 5 Max 16	10001	N	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	N	Identifier of the cash desk of the shop.
payment.merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Payment Gateway.
payment.merchantTrxRefId	String Max 50	123456	Y	Transaction identifier for the original payment to be refunded generated by the Payment Gateway.

3.9.6. Response

3.9.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

3.9.6.2. Response Body

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the refund transaction. Possible values are: 'EFF' = Executed; 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Bank error
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	123456	Y	Identifier of the payment related to the reversal.
payment.amount	Number Max 9 (Decimal part: 2)	70	Y	Payment amount
payment.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217

payment.fees	Number	0	N	Transaction fees, additional charge of the service expressed in euro, it will be always 0. As it is reserved for future use, it can be not expected.
payment.totalAmt	Number Max 9 (Decimal part: 2)	70	N	Total fees and amount. As it is reserved for future use, it can be not expected. The value will be always the same as payment.amount.
payment.reason	String Max 140	Soccer shoes	E	Payment reason
payment.paymentRefId	Number Max 16	22333	E	Identifier of the payment to divert.
payment.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	E	Identifier of the cash desk of the shop.

3.9.6.3. **Result code**

For common result's codes refer to the dedicated Chapter.

CODE	MESSAGE	RESULT	HTTP Code	note
00000	Positive result	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01027	Merchant not registered	FALSE	401	
01030	Communication Problems with Gateway	FALSE	401	
01034	The amount is not valid	FALSE	401	
01061	The operation is not allowed	FALSE	401	
01067	No payments found	FALSE	401	
01146	Payment reason not valid	FALSE	401	
01191	Bank account blocked	FALSE	401	
01255	The reversal is not coherent with the payment	FALSE	400	Mismatch of data between reversal and original payment
01258	Transaction id not coherent	FALSE	400	Mismatch of transaction ID
01300	Merchant not found	FALSE	401	
01301	Shop not found	FALSE	401	

01302	Cash Desk not found	FALSE	401	
01367	Preauthorization expired.	FALSE	400	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03004	Merchant not enabled for payment method:{0}	FALSE	400	
03018	The bank is not enabled	FALSE	400	
101	Unknown user	FALSE	401	
102	Sender IBAN does not belong to the bank	FALSE	401	
103	Sender IBAN is blocked	FALSE	401	
104	Insufficient funds in sender account	FALSE	401	
105	Authorization denied by bank	FALSE	401	
106	Invalid request (eg. Incorrect timestamp format, service not available, etc.).	FALSE	400	
107	Rejection due to internal Payer Bank checks (eg. Fraud checks)	FALSE	401	
108	TransactionId missing in SCT initiation request	FALSE	401	
109	Rejection due to internal Payee Bank checks (eg. Fraud checks)	FALSE	401	
110	Bank is not able to provide balance information	FALSE	401	
111	Invalid client certificate presented	FALSE	401	
112	x-jws-signature mismatch	FALSE	401	
113	Bearer token mismatch Client	FALSE	401	
114	Too many request	FALSE	401	
115	Bank has revoked the consent	FALSE	401	
116	Method non allowed	FALSE	401	

117	Request not acceptable	FALSE	401	
118	Unsupported media type	FALSE	401	
119	Sender bank account is closed	FALSE	401	
120	Recipient bank account is not valid	FALSE	401	
121	Bank's limit reached	FALSE	401	
122	Sender bank code is not valid	FALSE	401	
123	Recipient bank code is not valid	FALSE	401	
124	Consent is not valid	FALSE	401	
125	Recipient IBAN is blocked	FALSE	401	
126	Sender IBAN is not valid	FALSE	401	
127	Recipient IBAN is not valid	FALSE	401	
128	Recipient bank account is closed	FALSE	401	
129	Recipient IBAN does not belong to the bank	FALSE	401	
130	Payer Bank Service Unavailable	FALSE	401	
131	Payee Bank Service Unavailable	FALSE	401	
432	Country not authorised	FALSE	400	From 2024 R2
433	Country information mandatory for the payment	FALSE	400	From 2024 R2
999	Generic error	FALSE	500	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	

3.10. [PUT] Finalize Online Preauthorized Payment

This chapter describes the additional fields available in the endpoint related to the “finalizeOnlinePreauthPayment” service; whereas common fields are applied to all the endpoints.

3.10.1. Description

This API allows the merchant customer through the Payment Gateway to finalize an online pre-authorized payment with the updated vale of the amount once the items/services have been delivered to the buyer.

3.10.2. Business scenario

The merchant customer needs to finalize an online pre-authorized payment by confirming it. Through this service, the Payment Gateway can also cancel the payment on behalf of the merchant customer by setting the “confirm” query parameter to “false”.

In case “confirm” query parameter is set to “false” UAEIPP Overlay Service will unblock funds reserved on buyer account.

3.10.2.1. Direct outcome

- Merchant customer finalizes an online pre-authorized payment or cancels the pre-auth payment.

3.10.2.2. Related outcome

N.A.

3.10.3. URL

PUT

/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/request-to-pay

3.10.4. Operation

finalizeOnlinePreauthPayment

3.10.5. Request

3.10.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active Merchant customer's identifier code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant customer ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.10.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
confirm	Boolean	true	Y	Indicates if the merchant customer confirms or refuses the payment. Expected values: “true” = confirm the payment “false” = refuse the payment

3.10.5.3. Request Header

For the request headers, refer to the common ones described in the paragraph Common Fields.

3.10.5.4. Request Body

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	123456	Y	Internal identifier of the payment generated by the system.
payment.amount	Number Max 9 (Decimal part: 2)	70	Y	Updated amount of the payment after the items/services have been delivered to the buyer.
payment.currency	String Max 3	AED	N	Currency of the payment’s amount. ISO 4217
payment.reason	String Max 140	Soccer shoes	N	Payment reason.
payment.shopId	Number Min 5 Max 16	10001	N	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	N	Identifier of the cash desk of the shop.
payment.merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.

3.10.6. Response

3.10.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

3.10.6.2. Response Body

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

Name	Type	Sample Value	Req.	Description
payment	Object	N.A.	Y	Object that contains all the payment data.
payment.paymentId	Number Max 16	123456	E	Internal identifier of the payment generated by the system.

payment.amount	Number Max 9 (Decimal part: 2)	70	E	Updated amount of the payment after the items/services have been delivered to the buyer.
payment.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217
payment.fees	Number	0	N	Transaction fees, additional charge of the service expressed in euro. Its value is set to zero by default. As it is reserved for future use, it can be not expected.
payment.totalAmt	Number Max 9 (Decimal part: 2)	70	N	Total fees and amount. As it is reserved for future use, it can be not expected. The value will be always the same as payment.amount.
payment.reason	String Max 140	Soccer shoes	E	Payment reason
payment.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by the platform.
payment.cashDeskId	Number Min 8 Max 16	10000001	E	Identifier of the cash desk of the shop.

3.10.6.3. **Result code**

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
00023	Payment taken in charge, check the correct conclusion	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01030	Communication Problems with Gateway	FALSE	401
01034	The amount is not valid	FALSE	401
01041	Invalid QR code	FALSE	401
01047	The payment to be finalized does not match the payment initiated	FALSE	401
01054	Payment refused, try again	FALSE	401
01061	The operation is not allowed	FALSE	401
01067	No payments found	FALSE	401
01076	Bank account not found	FALSE	401
01146	Payment reason not valid	FALSE	401
01300	Merchant not found	FALSE	401
01301	Shop not found	FALSE	401

01302	Cash Desk not found	FALSE	401
01367	Preauthorization expired.	FALSE	400
01371	Merchant not authorized	FALSE	401
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
03004	Merchant not enabled for payment method:{0}	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

3.11. [POST] Refund

This chapter describes the additional fields available in the endpoint related to the "refund" service, whereas common fields are applied to all the endpoints.

3.11.1. Description

This API allows Merchants executing refunds.

Merchant must indicate the mobile number of the buyer receiving the refund and the amount to be refunded. Otherwise, he can use a proxy (email or a document-id).
Buyer must be a Customer enrolled to the platform with the App installed and active.

3.11.2. Business scenario

Possible scenarios:

- The merchant customer needs to refund a customer (retail refund type);
- The corporate needs to pay a salary (disbursement refund type)

3.11.2.1. Direct outcome

- The buyer's participant account is credited.

3.11.2.2. Related outcome

N.A.

3.11.3. URL

POST

/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund

3.11.4. Operation

refund

3.11.5. Request

3.11.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.11.5.2. **Request Header**

Name	Type	Sample Value	Req.	Scope	Description
appId	String 64	03704576A8D81E96BF0C5D8E8953F4298842EC7A1803884A3EEC96C11C290099	Y	Security	Unique identification code generated by the Participant's Mobile Server related to the APP installed by the merchant. This field must be populated with the appId of the channel from which the merchant is operating
deviceOSVersion	String Max 255	Ios 10.5.26	N	Security	Operating system of the device.
deviceModel	String Max 255	ABCD1234	N	Security	Code associated with the device model.
language	String 2	EN	N	Security	Code that defines which language to use for result messages. Standard used is the ISO 639-1 2 letter codes. Example values: "IT" "EN" "DE".
deviceId	String Max 255	ABCD1234	N	Security	Unique identification code of the device.
deviceIpAddress	String Max 255	192.168.1.1	N	Security	IP address associated with the device.
country	String 2	AE	N	Security	Sender Country Code defined according to the standard ISO-3166 Alpha2.
timestamp	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}\+\d{2}:\d{2}	2023-09-22T23:50:56.193+01:00	N	Security	Execution date/time of the call to the service. The timeZone in this field can be different from the value included in the field timeZone.
timeZone	String PATTERN: +hh:mm or -hh:mm	+01:00	N	Security	Time zone on sender's side registered by the device. This Timezone can be different from the Timezone included in Timestamp field.

For the remaining request headers, refer to the common ones described in the Common Fields.

3.11.5.3. **Request Body**

Name	Type	Sample Value	Req.	Description
refund	Object	N.A.	Y	Object that contains all the refund transaction's data.
refund.amount	Number Max 9	70.97	Y	Refund amount.

refund.currency	String Max 3	AED	N	Currency of the payment's amount. ISO 4217
refund.reason	String Max 140	wrong bill amount	N	Refund reason.
refund.type	Enum(String)	disbursement	N	It defines the scope of the refund transaction values can be: 5. "disbursement"; 6. "retail". Value is not mandatory and in case is not provided the platform will consider refund type implicitly "retail".
refund.mobile	String Max 30	+971837892848	C	Mobile number of the buyer receiving the refund. This must be a mobile number of a Buyer Customer enrolled to the UAEIPP Overlay Service by one of the Participants. The Buyer Customer must have the app installed on his device and activated with this mobile number.
refund.proxy	Object	N.A.	C	Object related to the proxy used by the buyer alternatively to the mobile number for receiving the payment.
proxy.type	Enum	email	Y	Type of the proxy of the buyer to be used for making/receiving payments. Expected values: email; document-id proxy.type is as per enrolment. document-id and email are case insensitive
proxy.value	String	john.wick@gmail.com	Y	Value of the proxy.
refund.paymentTime	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}	2012-11-25T23:50:56.193	N	Execution date of the request.

refund.shopId	Number Min 5 Max 16	10001	Y	Identification code of the shop issued by UAEIPP Overlay Service
refund.cashDeskId	Number Min 8 Max 16	10000001	N	Identifier of the cash desk of the shop.
merchantTrxId	String Max 50	123456	Y	Transaction identifier of the refund generated by the Merchant Customer.
refund.categoryPurpose	String Max 35	CCP	Y	Category Purpose Code in a proprietary form. For the list of allowed values, please refer to the dedicated document From 2024 R4

3.11.6. Response

3.11.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Response Header.

3.11.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the refund transaction. Possible values are: 'EFF' = Executed; 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Bank error
refund	Object	N.A.	Y	Object that contains all the refund transaction's data.
refund.paymentId	Number Max 16	123456	Y	Identification code of the refund generated by UAEIPP Overlay Service
refund.amount	Number Max 9	70.97	E	Refund amount.
refund.currency	String Max 3	AED	E	Currency of the payment's amount. ISO 4217
refund.reason	String	wrong bill amount	E	Payment reason

	Max 140			
refund.mobile	String Max 30	+353837892848	E	Mobile number of the buyer receiving the refund. This must be a mobile number of a Buyer Customer enrolled to the UAEIPP Overlay Service by one of the Participants. The Buyer Customer must have the app installed on his device and activated with this mobile number.
refund.proxy	Object	N.A.	E	Object related to the proxy used by the buyer alternatively to the mobile number for receiving the payment.
proxy.type	Enum	email	E	Type of the proxy of the buyer to be used for making/receiving payments. Expected values: email; document-id proxy.type is as per enrolment. document-id and email are case insensitive
proxy.value	String	john.wick@gmail.com	E	Value of the proxy.
refund.paymentTime	Timezone PATTERN: \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}.\d{3}	2012-11-25T23:50:56.193	E	The execution date of the request.
refund.fees	Number	0	N	It will always be zero. As it is reserved for future use, it can be not expected.
refund.totalAmt	Number Max 9	70	N	It will always match with amount. As it is reserved for future use, it can be not expected. The value will be always the same as payment.amount.

refund.shopId	Number Min 5 Max 16	10001	E	Identification code of the shop issued by UAEIPP Overlay Service.
refund.cashDeskId	Number Min 8 Max 16	10000001	E	Identifier of the cash desk of the shop.
merchantTrxId	String Max 50	123456	E	Transaction identifier of the refund generated by the Merchant Customer.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.11.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code	
00000	Positive result	TRUE	200	
00023	Payment taken in charge, check the correct conclusion	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01018	No default bank account found	FALSE	401	
01027	Merchant not registered	FALSE	401	
01030	Communication Problems with Gateway	FALSE	401	
01033	Cash Desk not found	FALSE	401	
01034	The amount is not valid	FALSE	401	
01044	Shop not found	FALSE	401	
01075	Customer not found	FALSE	401	
01146	Payment reason not valid	FALSE	401	
01300	Merchant not found	FALSE	401	
01301	Shop not found	FALSE	401	
01302	Cash Desk not found	FALSE	401	
01346	Currency mismatch, bank account currency: {0}, payment currency: {1}	FALSE	400	
01347	{0} is not a managed currency	FALSE	400	

01348	No active bank account found with currency {0}	FALSE	400	
01365	Bank account blocked, it is impossible to proceed with the request	FALSE	401	
01371	Merchant not authorized	FALSE	401	
02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
02005	The field {0} doesn't match any of the expected values	FALSE	400	
02010	At least one of the following fields is mandatory: {0}	FALSE	400	
03018	The bank is not enabled	FALSE	400	
101	Unknown user	FALSE	401	
102	Sender IBAN does not belong to the bank	FALSE	401	
103	Sender IBAN is blocked	FALSE	401	
104	Insufficient funds in sender account	FALSE	401	
105	Authorization denied by bank	FALSE	401	
106	Invalid request (eg. Incorrect timestamp format, service not available, etc.).	FALSE	400	
107	Rejection due to internal Payer Bank checks (eg. Fraud checks)	FALSE	401	
108	TransactionId missing in SCT initiation request	FALSE	401	
109	Rejection due to internal Payee Bank checks (eg. Fraud checks)	FALSE	401	
110	Bank is not able to provide balance information	FALSE	401	
111	Invalid client certificate presented	FALSE	401	

112	x-jws-signature mismatch	FALSE	401	
113	Bearer token mismatch Client	FALSE	401	
114	Too many request	FALSE	401	
115	Bank has revoked the consent	FALSE	401	
116	Method non allowed	FALSE	401	
117	Request not acceptable	FALSE	401	
118	Unsupported media type	FALSE	401	
119	Sender bank account is closed	FALSE	401	
120	Recipient bank account is not valid	FALSE	401	
121	Bank's limit reached	FALSE	401	
122	Sender bank code is not valid	FALSE	401	
123	Recipient bank code is not valid	FALSE	401	
124	Consent is not valid	FALSE	401	
125	Recipient IBAN is blocked	FALSE	401	
126	Sender IBAN is not valid	FALSE	401	
127	Recipient IBAN is not valid	FALSE	401	
128	Recipient bank account is closed	FALSE	401	
129	Recipient IBAN does not belong to the bank	FALSE	401	
130	Payer Bank Service Unavailable	FALSE	401	
131	Payee Bank Service Unavailable	FALSE	401	
432	Country not authorised	FALSE	400	From 2024 R2
433	Country information mandatory for the payment	FALSE	400	From 2024 R2
TBD	The categoryPurpose provided is invalid	FALSE	400	From 2024 R4
999	Generic error	FALSE	500	
04004	no {0} proxy customer found	FALSE	400	
04005	mobile numbers mismatch	FALSE	400	

04008	{0} is not a vaild proxy type	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	

3.12. [GET] Check Online Reversal Status

This chapter describes the additional fields available in the endpoint related to the "checkBusinessOnlineReversalStatus" service, whereas common fields are applied to all the endpoints.

3.12.1. Description

This API is called by the Payment Gateway on behalf of merchant customer to check the execution status of a requested reversal.

The platform provides "checkBusinessOnlineReversalStatus" to let Merchants recover the actual result of reversal in case they do not receive response due to technical issues. This operation may be used by merchant at any time, after "verifyOnlineBusinessReversal" and "confirmOnlineBusinessReversal" requests.

3.12.2. Business scenario

The merchant customer needs to check the status of a reversal operation.

3.12.2.1. Direct outcome

- The status of the reversal transaction is returned to the merchant customer.

3.12.2.2. Related outcome

N.A.

3.12.3. URL

GET

/inquiry-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/online/business/reversal/status

3.12.4. Operation

checkBusinessOnlineReversalStatus

3.12.5. Request

3.12.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description

bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.12.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.

3.12.5.3. Request Header

For the request headers, refer to the common ones described in the paragraph Common Fields.

3.12.5.4. Request Body

N.A.

3.12.6. Response

3.12.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Fields.

3.12.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the reversal transaction. Possible values are: 'EFF' = Executed; 'DA_STR' = To be transferred (the request has been accepted by UAEIPP Overlay Service, waiting to be forwarded to the participants); 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Bank error 'RIF_A' = Refused automatically by the platform

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.12.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code
00000	Positive result	TRUE	200
01000	Generic error	FALSE	500
01001	Negative result	FALSE	500
01027	Merchant not registered	FALSE	401
01253	Reversal not found	FALSE	401
01371	Merchant not authorized	FALSE	401
02000	The field {0} is not filled in	FALSE	400
02001	The field {0} has not a valid format [{1}]	FALSE	400
03018	The bank is not enabled	FALSE	400
03024	The Acquirer or Provider is not enabled	FALSE	400
03028	There is no association between the Acquirer or Provider and participant	FALSE	400
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400
03035	Provider Type is not valid	FALSE	400

3.13. [GET] Check Refund Status

This chapter describes the additional fields available in the endpoint related to the “checkBusinessRefundStatus” service, whereas common fields are applied to all the endpoints.

3.13.1. Description

This API is called by the Payment Gateway on behalf of the merchant customer to check the execution status of a requested refund.

The platform provides “checkBusinessRefundStatus” to let Merchants recover the actual result of refund operations in case Merchants do notreceive response due to technical issues. This operation may be used by merchant at any time, after “refund” requests.

3.13.2. Business scenario

The merchant customer needs to check the status of a refund operation (retail or disbursement).

3.13.2.1. Direct outcome

- The status of the refund transaction is delivered to the merchant customer.

3.13.2.2. Related outcome

N.A.

3.13.3. URL

GET

/business-payment-ms/services/groups/{groupCode}/banks/{bankCode}/bank-user/{bankUserId}/tag/{merchantTag}/refund/status

3.13.4. Operation

checkBusinessRefundStatus

3.13.5. Request

3.13.5.1. Path Parameters

Name	Type	Sample Value	Req.	Scope	Description
bankUserId	String Max 16	AA1234567890	Y	Functional	Active customer's identification code.
merchantTag	String Max 7	UB776WH	Y	Functional	Merchant ID issued by UAEIPP Overlay Service. The TAG is a unique ID.

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.13.5.2. Query Parameters

Name	Type	Sample Value	Req.	Description
merchantTrxId	String Max 50	7891011	Y	Transaction identifier for this operation created by the Merchant Customer.

3.13.5.3. Request Header

For the request headers, refer to the common ones described in the Common Fields.

3.13.5.4. Request Body

N.A.

3.13.6. Response

3.13.6.1. Response Header

For the response headers, refer to the common ones described in the paragraph Common Response Header.

3.13.6.2. Response Body

Name	Type	Sample Value	Req.	Description
status	String	EFF	Y	Status of the refund transaction. Possible values are: 'EFF' = Executed; 'RIC' = Requested (verify step has been executed but not the confirm one); 'ERR' = Bank error

For the remaining parameters, refer to the common ones described in the paragraph Common Fields.

3.13.6.3. Result Codes

CODE	MESSAGE	RESULT	HTTP Code	note
00000	Positive result	TRUE	200	
01000	Generic error	FALSE	500	
01001	Negative result	FALSE	500	
01067	No payments found	FALSE	401	
01258	Transaction id not coherent	FALSE	400	Mismatch of transaction ID
01300	Merchant not found	FALSE	401	
01371	Merchant not authorized	FALSE	401	

02000	The field {0} is not filled in	FALSE	400	
02001	The field {0} has not a valid format [{1}]	FALSE	400	
03018	The bank is not enabled	FALSE	400	
03024	The Acquirer or Provider is not enabled	FALSE	400	
03028	There is no association between the Acquirer or Provider and participant	FALSE	400	
03032	The Provider or Acquirer is not enabled for this merchant	FALSE	400	
03035	Provider Type is not valid	FALSE	400	

3.14. SWAGGER

See file: “business payment ms RTP” and “inquiry ms RTP”

3.15. APPENDIX

3.15.1. Request to pay Expiration Time

After a sendOnlineRequestToPay with paymentType set to “PRE” or “PAG” Buyer has 2 minutes to retrieve payment details and confirm the payment before the request to pay itself expires.
Request to pay expiration time is configurable.

3.15.2. Preauthorization Expiration Time

After a sendOnlineRequestToPay with paymentType set to “PRE” (= Preauthorized payment has been requested successfully) by Merchant’s Payment Gateway and confirmed by Buyer via APP, Merchant has 24 hours, 1 day (starting from successful response sent to sendOnlineRequestToPay request, configurable) to invoke finalizeOnlinePreauthPayment with “confirm” query parameter set to “true” (= confirm the payment with the actual amount.
After that time

1. Payment request goes to status “expired”
    a. finalizeOnlinePreauthPayment will return “01037” “Preauthorization is expired”.
    b. checkOnlineRequestStatus will return the payment with status “EXP” - Request expired;
2. platform in background will trigger a flow with participants to delete reservation made on buyer’s participant account.

3.15.3. Preauthorization not confirmed

After a sendOnlineRequestToPay with “paymentType” set to “PRE” (= Preauthorized payment has been requested successfully) by Merchant’s Payment Gateway and confirmed by Buyer via APP, in case Merchant invokes finalizeOnlinePreauthPayment with “confirm” query parameter set to “false” (= refuse the payment) platform in background will trigger a flow with participants to delete reservation made on buyer’s participant account.

3.15.4. Reversal verify expiration time

After a verifyOnlineBusinessReversal has been requested successfully by Merchant’s Payment Gateway Merchant has 5 minutes (starting from successful response sent to verifyOnlineBusinessReversal request, configurable) to invoke confirmOnlineBusinessReversal.

After that time:

1. The platform in background will trigger a flow with participants to delete reservation made on Merchant’s bank account.
2. Any execution of the checkBusinessOnlineReversalStatus will return a response with status attribute = 'RIF_A' - Refused automatically by the platform.

3.15.5. Reversal and refund timeouts

UAEIPP Overlay Service provides “checkBusinessOnlineReversalStatus” and “checkBusinessRefundStatus” to let Merchants recover the actual result of reversal and refund operations in case Merchant does not receive response due to technical issues. These operations may be used by merchant at any time, after “verifyOnlineBusinessReversal”, “confirmOnlineBusinessReversal” and “refund” requests.

UAEIPP Overlay Service relies on Participants' APIs to reply within 20 seconds when "P2B Refund VERIFY FLOW" and "P2B Refund CONFIRM FLOW" are executed (see section below for description of these flows), then Merchant may decide its own timeout definition according to inner business logic.

"P2B Refund VERIFY FLOW" and "P2B Refund CONFIRM FLOW" are executed with participants to support both Reversal and Refunds described in this document. In case of Reversal in "P2B Refund VERIFY FLOW" and "P2B Refund CONFIRM FLOW" the platform will forward to banks full references of the original transaction to be reverted.

3.15.6. Polling frequency (GET CheckOnLineRequestStaus)

Payment Gateway can invoke max 10 times per minute this endpoint in order to retrieve payment status.

3.15.7. Participant Flows

Following flows to briefly explain how UAEIPP Overlay Service interacts with participants to execute a Payment, or preauthorization or Reversal

P2B VERIFY FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to verify Buyer and Merchant IBANs In order to execute a payment. As final result this flows reserve fund on Buyer Account.
P2B CONFIRM FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to initiate a payment after a previously successful run P2B VERIFY FLOW
P2B Preauth VERIFY FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to verify Buyer and Merchant IBANs In order to execute a Preauthorized payment. As final result this flows reserve fund on Buyer Account.
P2B Preauth CONFIRM FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to confirm final amount of a previously successfully authorized preauthorization with P2B Preauth VERIFY FLOW
P2B Refund VERIFY FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to verify Buyer and Merchant IBANs In order to execute a refund or a reversal. As final result this flows reserve fund on Merchant Account.
P2B Refund CONFIRM FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to initiate a reversal or a refund after a previously successful run P2B Refund VERIFY FLOW
Delete Reservation FLOW	Orchestration of Participants API Endpoints invocations run by UAEIPP Overlay Service platform to unblock funds previously reserved after successfully P2B VERIFY FLOW or P2B Preauth VERIFY FLOW or P2B Refund VERIFY FLOW

3.15.8. Proxy Management Principles

In this paragraph, the main principles related to the Proxy management of customers are described. Currently, the proxies identify only the consumers.

The proxy types that are currently supported for consumers by the UAEIPP Overlay Service platform are the following:

1. "e-mail";
2. "document-id"

The following sections describe the principles valid for each proxy type for consumers.

3.15.8.1. **E-mail**

For the proxy with type "e-mail" the following rules must be followed:

9. **Mobile number relationship**: the e-mail address of a consumer is linked **one to one** with the mobile number. One mobile number cannot have more than one email address. Therefore, it's not possible to register the same e-mail with different mobile numbers;

10. **Payments Scenarios availability**: the e-mail address can be used in P2P Send Money and Request to Pay scenarios (Split Bill transactions excluded), P2B Requests to Pay sent by the merchant (including e-commerce scenarios) and B2P payment scenarios (refund and disbursement), alternatively to the mobile number;
11. **Default Account**: the default account linked to the email is equal to the default of the mobile number, due to the unique relationship with the mobile number. If the participant bank account at the mobile number level is updated, the change is also reflected to the default of the e-mail address;
12. **E-mail format**: the e-mail must be set in Latin characters only, Arabic characters are not supported in the e-mail input field and must comply with the standard format, e.g. example@domain.com;
13. **E-mail creation**: the e-mail address can be registered manually by the user by using the APP after activation. It cannot be defined in the enrolment phase by the participant. The e-mail address requires to be validated by the user as described in the next principle;
14. **E-mail validation**: in order to complete the successful registration of the e-mail address, the user must validate it through the OTP approach, by inserting the OTP code in the APP once the customer has received it in his/her email's inbox. In case the customer's device is changed, but the phone number is the same, the email address does not need to be redefined and validated;
15. **E-mail update**: the e-mail can be updated by the customer and to complete the operation it has to be validated as explained in step 6. If the customer has not validated the e-mail, he is not reachable with this proxy;
16. **E-mail deletion**: the e-mail address cannot be deleted directly by the customer, only through Participants channels (FE API and Web Portal) can be performed.

3.15.8.2. Document-ID

For the proxy with type "document-id", that is meant as a unique datum of an individual, the following rules must be followed:

9. **Mobile number relationship**: the document-id of a consumer can be linked to **one or more** mobile numbers, but **not vice versa** (1 mobile number cannot be linked to more than one document-id at the same time). The platform will not allow this duplication. It can be possible only in case of recycled numbers;
10. **Payments Scenarios availability**: the document-id can be used alternatively to the mobile number as an identifier of the UAEIPP Overlay Service user in P2B (including e-commerce) and B2P (refund and disbursement) scenarios;
11. **Default Account**: because of the 1-N relationship with the mobile number, the document-id needs a dedicated default account, in order to let the user make/receive payments using the document-id as a proxy. The default account at the document-id level is the same at the mobile number level during enrolment, the first account of the list is set as default also at the document-id level). After the enrolment, when the customer activates the APP, he can change the default account by using the APP that will call the proper API (updateProxyDefaultBankAccount – API functionality described in SDK Interface Specifications and Front End APIs for Participants' Channels). The default account at the document-id level, can, therefore, differ from the one set at the mobile number level. If the Participant blocks a default account, the consumer is not reachable with the linked proxy until the customer selects another participant account as default. The consumer, to change the default account must select one of the participant accounts connected to the APP he/she is using. For consumers, no automatic change of the default account can be done by the participant or the UAEIPP Overlay Service platform;
12. **Document-ID format**: compared to the e-mail address, the document-id has not a specific format since can be equal to the Emirates ID or to the Passport ID. When a customer has both the documents (EmiratesID and PassportID), the participants must enrol the customer with the EmiratesID;
13. **Document-ID creation**: the document-id can be created only during the enrolment phase through the Participant channels (FE API, Web Portal and Bulk Operations) since it is a mandatory user's datum;
14. **Document-ID validation**: compared to the e-mail address, the document-id does not require any validation from the user, it is under the responsibility of the Participant to certify the datum;

15. **Document-ID update**: the document-id cannot be updated by the user, neither by using Participant channels (FE API, Web Portal and Bulk Operations), since it is a unique identifier of the user. If a customer change his document-Id, he must be unenrolled and re-enrolled (security reasons);
16. **Document-ID deletion**: the document-id cannot be deleted by the user, neither through Participant channels (FE API, Web Portal and Bulk Operations).

3.15.9. Technical Service Provider and Technical Service provider as acquirer

For a Technical Service Provider (providerType=PROVIDER), or a Technical Service Provider as Acquirer (providerType=ACQUIRER), it's possible to call a Merchant API using their own parent and sub code as path parameters but the following 3 additional fields in the APIs header must also be populated (for more details see paragraph 3.4.1):

- providerType
- participantGroupCode
- participantBankCode

In addition:

- The parent and sub code indicated as API path parameters must match with a valid and active Technical Service Provider or Technical Service Provider as Acquirer.
- The participantGroupCode and participantBankCode fields in the API header must match with a valid and active UAEIPP Participant Bank.
- The UAEIPP Participant and Technical Service Provider, or Technical Service Provider as Acquirer, must be associated each other in the UAEIPP Overlay platform.
- The Technical Service Provider, or Technical Service Provider as Acquirer, can invoke merchant APIs only for Merchants associated to them in the UAEIPP Overlay platform.
- Even if a payment is initiated from a Technical Service Provider or Technical Service Provider as Acquirer (e.g acquirer generates an online RTP on behalf of a merchant) through routing functionality supported by UAEIPP Overlay platform the payment is always processed by the Participant for both sender and receiver.