Soramitsukhmer | ソラミツ Co., Ltd

FAST - Message Flow and Web Service

The purpose of this document is to outline the message flows and web services for the NBC FAST project.

Change history

Date	Version	Purpose of modification	Updated by
2021-01-12	0.1.0	Initial version	Phuylai OEUNG
2021-03-03	0.1.1	Modify Customer Account Inquiry (STP) message flow	Sokphyrum KIM
2021-06-02	0.1.2	Updated Customer Account Inquiry (STP) - 2.1.2 Web Service section b) Token Request and c) Receiver Account Request	Sovannak Beng Sokphyrum Kim
2021-09-06	0.1.3	Added Reference, 2.1 Customer Account Inquiry Message Flow, Updated 2.1.2 Web Service, diagram of 2.2 Customer Account Inquiry (Full STP Transfer) point 5, 2.6 Account-based Transfer (Full STP - Failed at CM), 2.7 Account-based Transfer (Full STP - Failed at FI), updated makeAcknowledgement to makeAcknowledgment	Sokphyrum Kim Sodanheang Vandy Sokkun Norn
2021-09-13	0.1.4	Updated title text "Receiver Account Request" to "Customer Account Inquiry request" section 2.1.2.b.2 & 2.1.c.2, Retries timeout of CM (60s, 60s, 60s) in section 2.6 & 2.7, Added "ability to pay" validation criteria in section 2.5 point #3, section 2.14.2.a added note in the description,	Sokphyrum Kim
2021-09-23	0.1.4.1	Updated the inconsistency format of <ReqdExctnDt> in sample request and response	Sokphyrum Kim
2021-10-26	0.1.4.2	Updated sample of the request in makeReverseTransaction (FAST to Sender Participant) section	Sokphyrum Kim Sodanheang Vandy

Contents

1 Introduction

1.1 Purpose of the document

This document specifies the main message flows and its corresponding web services for the FAST payment system. This document is intended for the development and business teams.

1.2 Product vision

The current FAST is a fund transfer and payment service in Khmer Riel across financial institutions, enabling customers to send and receive funds instantly. Operation of FAST is available 24/7, yet settlement is conducted on a business day at a time specified by NBC. This version of FAST will bring new features to the existing FAST system by adding manual/automatic straight-through processing for account-based and non-account-based fund transfer, failure tolerance mechanism, optimized configuration granularity, dynamic report/log, and new UI based on RFT (Real FAST Transfer) system. Besides, the new FAST system is supposed to support multiple currency types, KHR as well as USD. More currency can be added in the future.

1.3 Reference

1. Client Module/Adapter Installation Guideline:
   Fast Client Module Setup Guide

2 Overview of the message flow

The following diagram illustrates the electronic information exchange between the sender bank, FAST system, and the receiver bank.

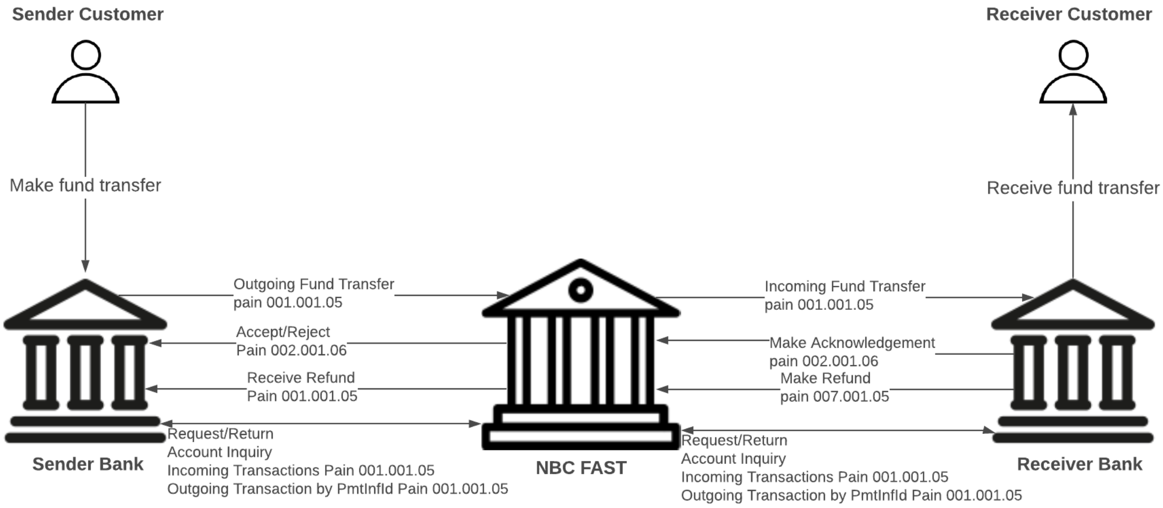


Fig.1 The overview of the message flows

Participant bank CBS communicates with FAST using a software module named Client Module (CM). This CM is mainly integrated for the STP option. This adapter is installed in the client environment (Participant bank) and supports the ISO 20022 financial message. The following lists down the supported messages in the system:

- Pain 001.001.05 - fund transfer message
- Pain 002.001.06 - accept, reject, and acknowledgment message
- Pain 007.001.05 - Refund message

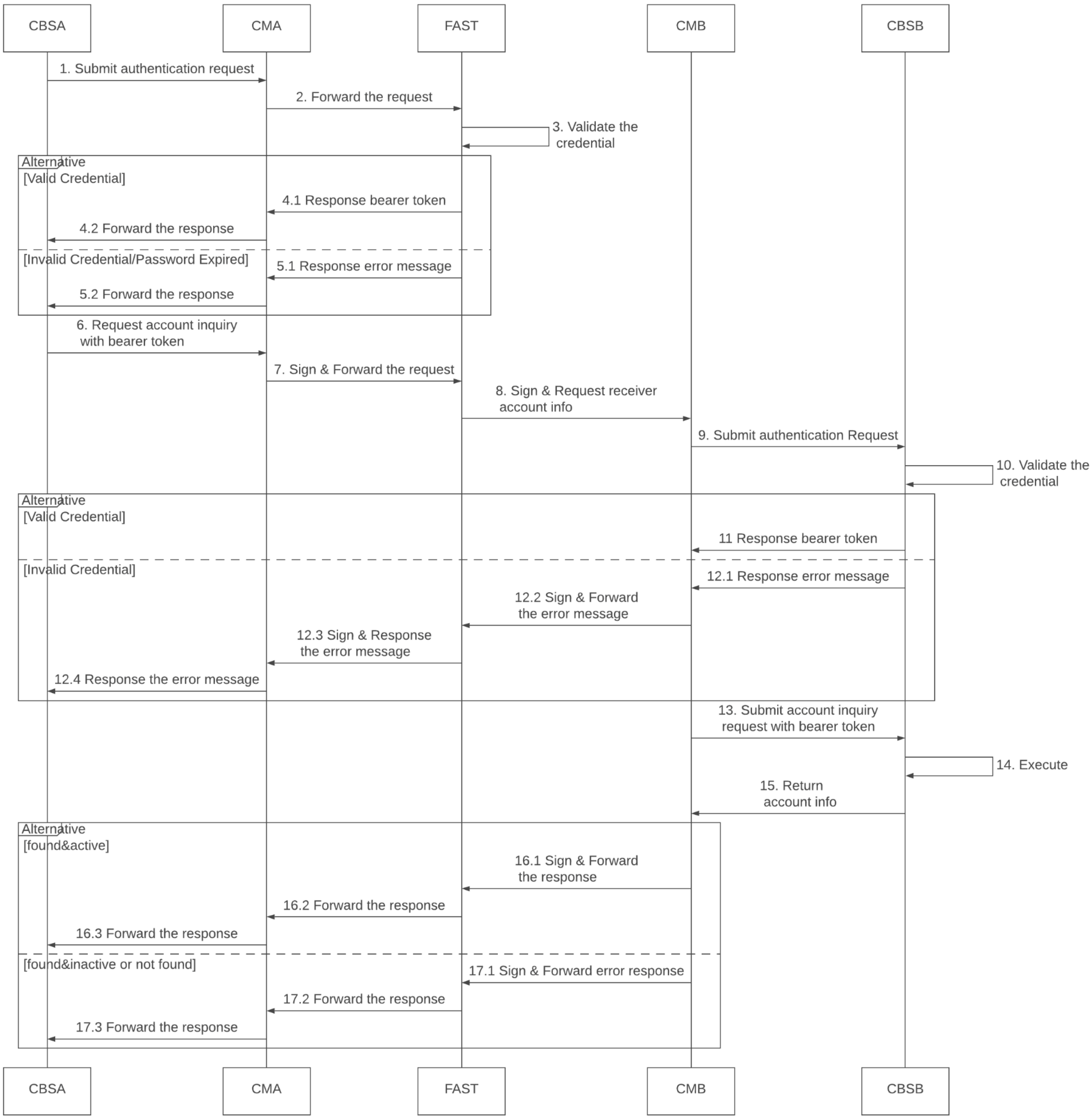
Furthermore, FAST provides web services for participant banks to inquire about its account information, incoming transactions, and outgoing transactions.

It is worth noting that we store the following in the CM for signing and validating procedure:

- Store private key and digital certificate for FI to sign the outgoing batch
- Store Digital certificate of FAST to validate the incoming batch from FAST system

2.1 Customer Account Inquiry

2.1.1 Message Flow



CBSA
CMA
FAST
CMB
CBSB
1. Submit authentication request
2. Forward the request
3. Validate the credential
Alternative
[Valid Credential]
4.1 Response bearer token
4.2 Forward the response
[Invalid Credential/Password Expired]
5.1 Response error message
5.2 Forward the response
6. Request account inquiry with bearer token
7. Sign & Forward the request
8. Sign & Request receiver account info
9. Submit authentication Request
10. Validate the credential
Alternative
[Valid Credential]
11 Response bearer token
[Invalid Credential]
12.1 Response error message
12.2 Sign & Forward the error message
12.3 Sign & Response the error message
12.4 Response the error message
13. Submit account inquiry request with bearer token
14. Execute
15. Return account info
Alternative
[found&active]
16.1 Sign & Forward the response
16.2 Forward the response
16.3 Forward the response
[found&inactive or not found]
17.1 Sign & Forward error response
17.2 Forward the response
17.3 Forward the response
CBSA
CMA
FAST
CMB
CBSB

1. Bank A CBS (CBSA) submits authentication REST API requests to Bank A Client Module (CMA). CMA forwards request to FAST Core Engine (CE)
2. FAST CE validates the credentials. If the credentials are valid then the bearer token is a response from FAST CE to CBSA via CMA.
3. CBSA submits a customer account Inquiry REST API request with a valid bearer token to CMA. CMA signs the request using the private key of the FI and sends it to FAST CE.
4. Fast CE signs and submits the request to CMB.
5. Firstly, CMB submits authentication requests to CBSB. CBSB validates the credentials and responses the bearer token to CMB.
6. CMB submits the account inquiry request to CBSB with a valid bearer token. CBSB executes the request and responds to the receiver account information to CMB.

7. CMB signs and forwards response back to FAST CE. The response is forwarded to CBSA via CMA.

Note:

- From CMA to FAST CE, CMA calls the endpoint mentioned on the section 2.1.2.c
- From FAST CE to CMB, the endpoint is defined in CM's configuration, parameter *CM_ACI_ENDPOINT (For more detail, please refer to reference document in section 1.3)*

2.1.2 Web Service

a) Description

The participant bank invokes this REST API to acquire the receiver customer's account information before proceeding with the transfer procedure. Doing so will enable quick response to the error concerning the receiver account, such as account not found, invalid account, inactive account,etc., to improve the transfer performance.

b) From CBSA to CMA

1. Token Request

username	FAST System STP user's Username that NBC provides to Sender Bank
password	FAST System STP user's Password that NBC provides to Sender Bank
rememberMe	true/false
Method	POST
Endpoint	http://{ClientModule-domain-name}/api/v1/auth

Note:

- ***{ClientModule-domain-name}*** can be a combination of Client module IP and Port i.e *{CM-IP}:{CM-Port}*.
- ***{CM-IP}*** is an accessible IP address to each Client Module of the participant.
- ***{CM-Port}*** is the CM_SERVER_PORT parameter value that is configured in each Client module of the participant.

Sample of the request

```
{
"username":"BANK1",
"password":"password1",
"rememberMe":true
}
```

Response

status	Status code, error code, error message, error warning
data	null if error; token if success

Sample of the response for a successful request

```
{
"status": {
"code": 0,
"errorCode": null,
"error": null,
"warning": null
},
"data": {
"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoicG9zdGluZyIsInVzZXJuYW
1lIjoic2VyZXl3aXRuZXNzN
yIsInBhc3N3b3JkIjoiNUtRZWNYaVdteUtWY1l3Vms3WEZ6eDJuOUFTbURQcGJUc005dFVGQ
WNkVFRrYWt
aMWZMIiwidXNlclByb2ZpbGVUb2tlbiI6ImV5SmhiR2NpT2lKSVV6VXhNaUo5LmV5SnpkV0l
pT2lKelpYSmxlW
GRwZEc1bGMzTTNJaXdpWVhWMGFDSTZJbEpQVEVWWZlZWTkZVaUlzSW1WNGNDSTZNVFUzTkRZM
U1U
a3dOMzAuYU8tVGRXcHZ5RXJhSUVKc2JIVnBFTXVsY3QtWmhpZzh0eUdEV0phWFNGZk1HTXMy
TUIzYkY
2OTM3Q3FEUGZDN2lEVzVZYjByN21pOGxnd08taHMzcHciLCJpYXQiOjE1NzQ2NDkzMTcsImV
4cCI6MTU3
NDczNTcxN30.kFQ84edOAUUlKfwAAw0dYHGfwuAbRyVmOtPtNdk61S0"
}
```

Sample of the response for a failure request (Invalid authentication)

```
"status": {
"code": 4,
"errorCode": 11,
"error": "Authentication error. Please try again.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (User's password is expired)

```
"status": {
"code": 4,
"errorCode": 187,
"error": "User's password is expired. Please update your password.",
"warning": ""
},
"data": null
}
```

2. Customer Account Inquiry Request

Auth Bearer Token	Token FAST gets from Auth
account_no	Account number to inquire
Method	POST
Endpoint	http://{ClientModule-domain-name}/api/v1/account-inquiry

Sample of the request

```
{
"account_no": "123456789",
"participant_code": "ABAAKHPP"
}
```

Response

status	Status code, error code, error message, error warning
data	null if error; account information if success

Sample of the response for a successful request (Account is found)

```
{
"status": {
"code": 0,
"errorCode": null,
"error": null,
"warning": null
},
"data": {
"result": "success",
"product_code": "FAST"
"account_data": {
```

```
"account": "123456789",
"account_name": "John Smith"
"account_currency": "USD",
"bank_code": "ABAAKHPP",
"account_type": "01"
}
}
```

Sample of the response for a failure request (Account not found)

```
{
"status": {
"code": 1,
"errorCode": 65,
"error": "The account Number is not found. Please check the entered
data.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Expired token)

```
{
"status": {
"code": 4,
"errorCode": 10,
"error": "JWT token has expired. Please, log in again.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Participant bank code not found)

```
{
"status": {
"code": 1,
"errorCode": 184,
"error": "Receiver’s Participant Code is not found. Please check the
entered data.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (CM/CBS is unreachable)

```
{
"status": {
"code": 1,
"errorCode": 41,
"error": "CBS is not available",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (FAST is unreachable)

```
{
"status": {
"code": 1,
"errorCode": 40,
"error": "TPS is not available",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Account Inquiry setting is Inactive. FI Admin set Inactive status of Account Inquiry in CBS Setting.)

```
{
"status": {
"code": 1,
"errorCode": 182,
"error": "The receiver bank's account inquiry setting status is
inactive.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Some other system errors eg. services have not been started, network issues, database issues, and so on)

```
{
"status": {
"code": 1,
"errorCode": 131,
"error": "Internal server error",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Invalid Signature.)

```
{
"status": {
"code": 1,
"errorCode": 183,
"error": "There is invalid signature with response and request.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Invalid Account Format.)

```
{
"status": {
"code": 1,
"errorCode": 185,
"error": "Account number is invalid format. The account number should
contain [0-9].",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Expired Certificate.)

```
{
"status": {
"code": 1,
"errorCode": 186,
"error": "Certificate to verify the request is expired. Please update
the certificate.",
"warning": ""
},
"data": null
}
```

c) From CMB to CBSB

1. Token Request

username	Username of the CBS system in the receiver bank system
password	Password of the CBS system in the receiver bank system
rememberMe	true/false
Method	POST
Endpoint	http://{CBS-domain-name}/api/v1/auth

Note:

- ***{CBS-domain-name}*** *is configured via parameter* ***{CM_ACI_ENDPOINT}***
- User credential is configured via parameters ***{CM_ACI_USER_NAME} &*** ***{CM_ACI_PASSWORD}*** in the Client module of each participant.

Sample of the request

```
{
"username":"BANK1",
"password":"password1",
"rememberMe":true
}
```

Response

status	Status code, error code, error message, error warning
data	null if error; token if success

Sample of the response for a successful request

```
{
"status": {
"code": 0,
"errorCode": null,
"error": null,
"warning": null
},
"data": {
"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoicG9zdGluZyIsInVzZXJuYW
1lIjoic2VyZXl3aXRuZXNzN
yIsInBhc3N3b3JkIjoiNUtRZWNYaVdteUtWY1l3Vms3WEZ6eDJuOUFTbURQcGJUc005dFVGQ
WNkVFRrYWt
aMWZMIiwidXNlclByb2ZpbGVUb2tlbiI6ImV5SmhiR2NpT2lKSVV6VXhNaUo5LmV5SnpkV0l
pT2lKelpYSmxlW
GRwZEc1bGMzTTNJaXdpWVhWMGFDSTZJbEpQVEVWZlZWTkZVaUlzSW1WNGNDSTZNVFUzTkRZM
U1U
a3dOMzAuYU8tVGRXcHZ5RXJhSUVKc2JIVnBFTXVsY3QtWmhpZzh0eUdEV0phWFNGZk1HTXMy
TUIzYkY
2OTM3Q3FEUGZDN2lEVzVZYjByN21pOGxnd08taHMzcHciLCJpYXQiOjE1NzQ2NDkzMTcsImV
4cCI6MTU3
NDczNTcxN30.kFQ84edOAUUlKfwAAw0dYHGfwuAbRyVmOtPtNdk61S0"
}
```

Sample of the response for a failure request (Invalid authentication)

```
"status": {
"code": 4,
"errorCode": 11,
"error": "Authentication error. Please try again.",
"warning": ""
},
"data": null
}
```

2. Customer Account Inquiry Request

Auth Bearer Token	Token CBS gets from Auth
account_no	Account number to inquire
Method	POST
Endpoint	http://{CBS-domain-name}/api/v1/account

Sample of the request

```
{
"account_no": "123456789"
}
```

Response

status	Status code, error code, error message, error warning
data	null if error; account information if success

Sample of the response for a successful request (Account is found)

```
{
"status": {
"code": 0,
"errorCode": null,
"error": null,
"warning": null
},
"data": {
"result": "success",
"product_code": "FAST"
"account_data": {
"account": "123456789",
"account_name": "John Smith"
"account_currency": "USD",
"bank_code": "ABAAKHPP",
"account_type": "01"
}
}
```

Sample of the response for a failure request (Account not found)

```
{
"status": {
"code": 1,
"errorCode": 65,
"error": "The account Number is not found. Please check the entered
data.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Expired token)

```
{
"status": {
"code": 4,
"errorCode": 10,
"error": "JWT token has expired. Please, log in again.",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (CM/CBS is unreachable)

```
{
"status": {
"code": 1,
"errorCode": 41,
"error": "CBS is not available",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (FAST is unreachable)

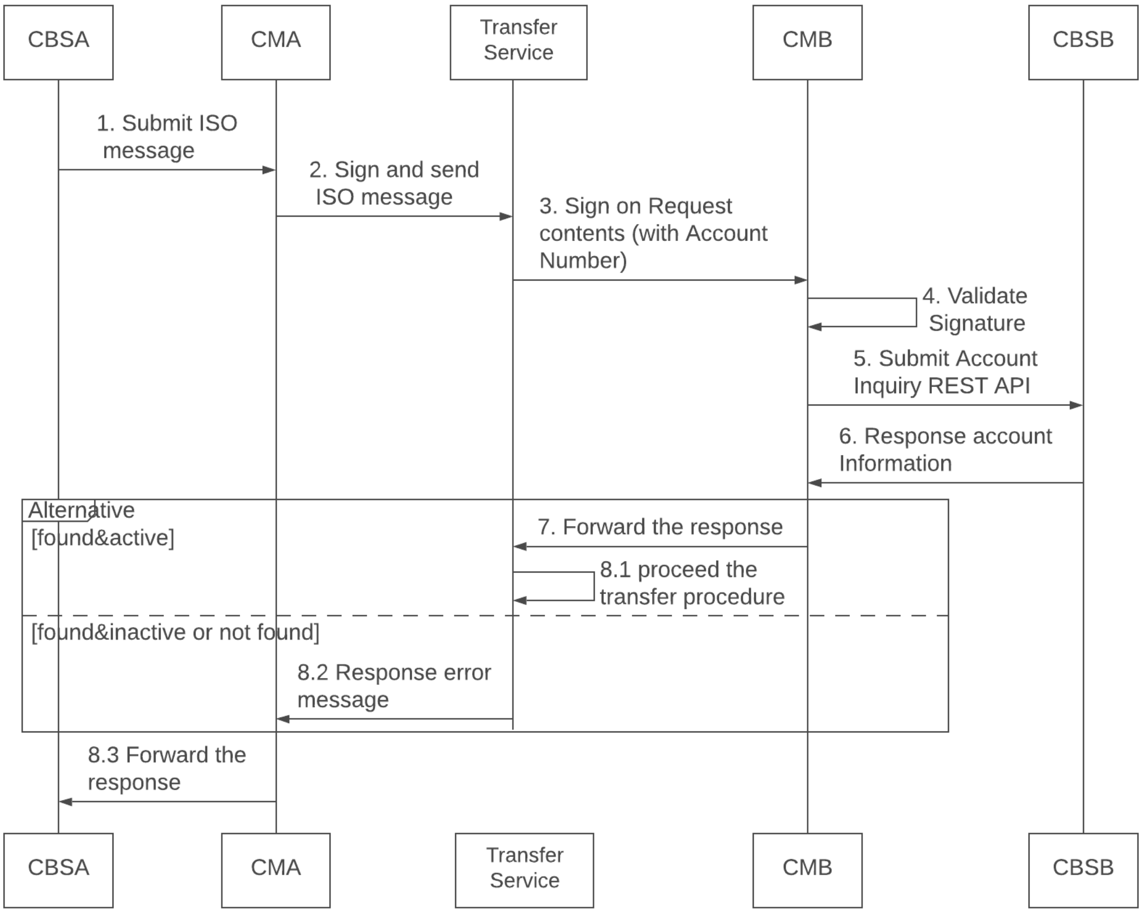
```
{
"status": {
"code": 1,
"errorCode": 40,
"error": "TPS is not available",
"warning": ""
},
"data": null
}
```

Sample of the response for a failure request (Some other system errors eg. services have not been started, network issues, database issues, and so on)

```
{
"status": {
"code": 1,
"errorCode": 131,
"error": "Internal server error",
"warning": ""
},
"data": null
}
```

2.2 Customer Account Inquiry (Full STP Transfer)

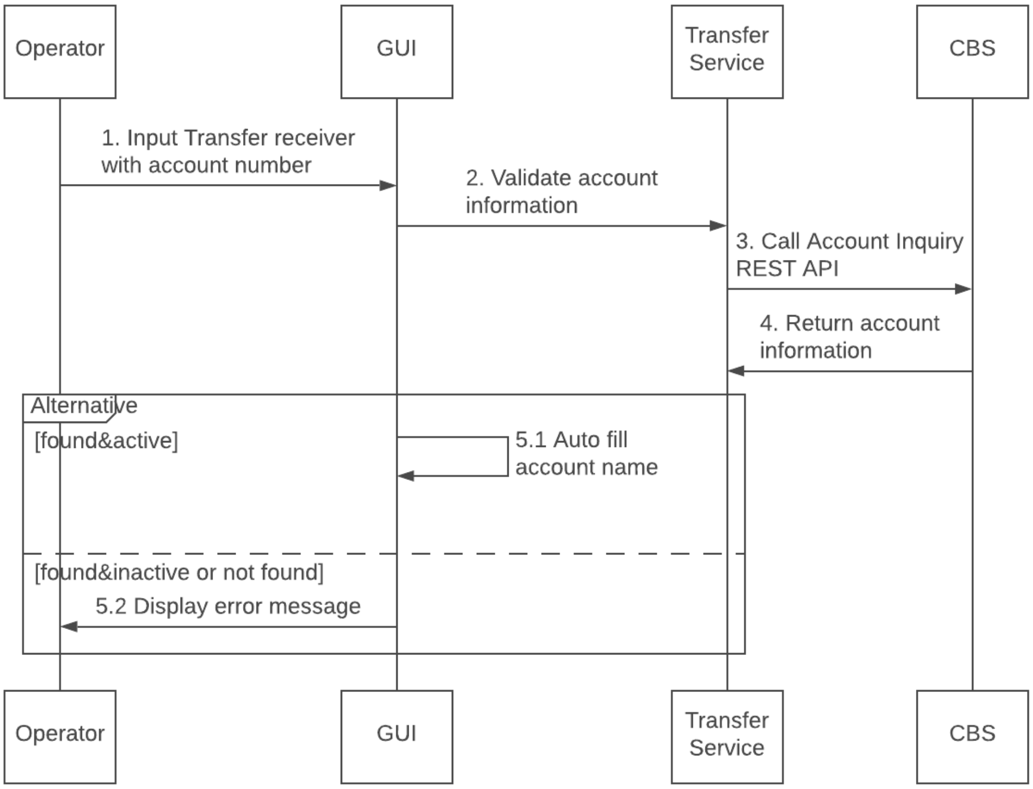
Message Flow


CBSA
CMA
Transfer Service
CMB
CBSB
1. Submit ISO message
2. Sign and send ISO message
3. Sign on Request contents (with Account Number)
4. Validate Signature
5. Submit Account Inquiry REST API
6. Response account Information
Alternative
[found&active]
7. Forward the response
8.1 proceed the transfer procedure
[found&inactive or not found]
8.2 Response error message
8.3 Forward the response
CBSA
CMA
Transfer Service
CMB
CBSB

1. CBS of Bank A submits a fund transfer message ISO 20022 (pain 001.001.05) via SOAP web service into Bank A Client Module (CMA)
2. Bank A CM hashes and signs the message using the private key of the FI and sends to FAST Core Engine (CE) Transfer Service
3. Transfer Service applies all the validation rules on the message and checks if it complies (message WSDL, digital signature, receiver account, receiver bank status, currency, operation limit, ability to pay). For receiver account validation, Transfer Service calls the *Customer Account Inquiry API* of the receiver Bank B via Bank B Client Module B to check the status of the receiver bank and the validity of the receiver customer account
4. If it is valid (including others validations), Transfer Service responses a corresponding successful message to sender Bank A using pain 002.001.06 with transaction status (Txsts) of **Accepted Settlement in Process (ACSP)** and proceed with the transfer procedure (*See 2.3 Account-based Transfer (Full STP - Success)*)
5. If it is invalid, Transfer Service responds with a corresponding error message to Sender Bank A using pain 002.001.06 message with transaction status (Txsts) of **Rejected**

(RJCT). The message also provides the additional message offering the reason for the rejection

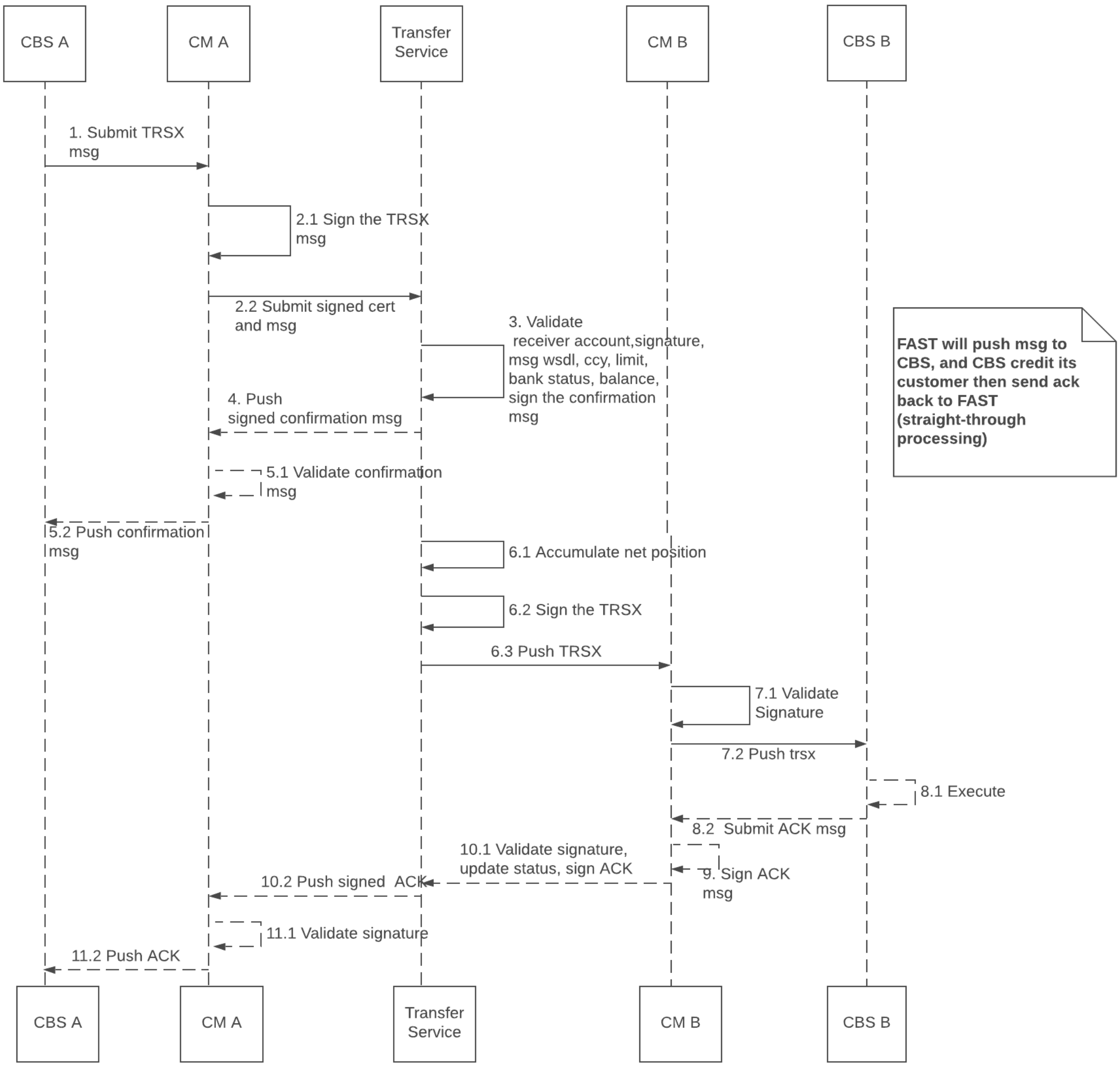
2.3 Customer Account Inquiry (Manual Transfer)


Operator
GUI
Transfer Service
CBS
1. Input Transfer receiver with account number
2. Validate account information
3. Call Account Inquiry REST API
4. Return account information
Alternative
[found&active]
5.1 Auto fill account name
[found&inactive or not found]
5.2 Display error message
Operator
GUI
Transfer Service
CBS

1. The operator user of Bank A creates a fund transfer using FAST GUI. He/She selects a receiver bank and inputs receiver customer's Account Number
2. The FAST client will call the customer account inquiry API of the receiver bank to check the validity of the receiver customer account (Please refer to 2.1.2)
3. If it is valid, FAST clients will trigger the autofill of account name
4. If it is invalid, FAST client responses a corresponding error message to GUI offering the reason for the rejection

2.4 Account-based Transfer (Full STP - Success)

2.4.1 Message Flow


CBS A
CM A
Transfer Service
CM B
CBS B
1. Submit TRSX msg
2.1 Sign the TRSX msg
2.2 Submit signed cert and msg
3. Validate receiver account,signature, msg wsdl, ccy, limit, bank status, balance, sign the confirmation msg
FAST will push msg to CBS, and CBS credit its customer then send ack back to FAST (straight-through processing)
4. Push signed confirmation msg
5.1 Validate confirmation msg
5.2 Push confirmation msg
6.1 Accumulate net position
6.2 Sign the TRSX
6.3 Push TRSX
7.1 Validate Signature
7.2 Push trsx
8.1 Execute
8.2 Submit ACK msg
10.1 Validate signature, update status, sign ACK
9. Sign ACK msg
10.2 Push signed ACK
11.1 Validate signature
11.2 Push ACK
CBS A
CM A
Transfer Service
CM B
CBS B

1. CBS of Bank A submits a fund transfer message using ISO 20022 format (*pain 001.001.05*) by calling **[makeFullFundTransfer]** via SOAP web service into Bank A CM
2. Bank A CM hashes the message and signs it using the private key of the FI and sends it to FAST Core Engine (CE) Transfer Service (The private key is stored in the FI's CM)
3. Transfer Service applies all the validation rules on the message and checks if it complies (message WSDL, digital signature, receiver account, receiver bank status, currency, operation limit, ability to pay)
4. If it is valid, Transfer Service hashes and signs a confirmation message *pain 002.001.06* with transaction status (Txsts) of **Accepted Settlement In Process (ACSP)** using FAST system private key, and sends it to sender Bank A CM. In case of invalid, please refer to *2.4 Account-based Transfer (Full STP - Rejected At Core)*

5. CM of Bank A will validate the signature of the received confirmation message and send it to its own CBS
6. Transfer Service will accumulate the net position, signs the transfer message (outgoing batch) using the FAST private key and send to CM of Bank B
7. CM of Bank B validates the signature on the message and sends it to CBS of Bank B by calling **[makeFullFundTransfer]** via SOAP
8. CBS of Bank B bears the responsibility to validate and verify the incoming message. If it is valid, Bank B has to credit the fund into its customer account and notify the customer. At this stage, the transaction is counted as a successful transaction. After crediting, Bank B sends an acknowledgment message *pain 002.001.06* to its CM by calling **[makeAcknowledgment]** via SOAP
9. CM of Bank B signs on the message and sends to FAST Transfer Service
10. Transfer Service updates the transaction status to **Acknowledged by FI**, signs the acknowledgement message using the FAST private key and sends it to the CM of Bank A
11. CM of Bank A validates the message and pushes to its own CBS by calling **[makeAcknowledgment]** via SOAP (Please refer to 2.15)

2.4.2 Web Services

makeFullFundTransfer (Sender Participant to FAST)

a) Description

The participant bank invokes this API to make fund transfer in a full automation process. In such that the transaction is signed and sent to the FAST Transfer Service through the sender Client Module (CM), and it will be received at the receiver bank CBS after passing all the validation rules of the FAST Transfer Service.

Request

Parameter	Description
cm_user_name	Username of the CM which makes the fund transfer
cm_password	Password of the CM which makes the fund transfer
iso_message	pain.001.001.05
import_file_name	Name of the xml file which stores batch transaction

Sample of the request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
      <soapenv:Header/>
            <soapenv:Body>
```

```
                   <fas:makeFullFundTransfer>
                         <cm_user_name>bred</cm_user_name>
                         <cm_password>aA12345@</cm_password>
                         <iso_message><![CDATA[<?xml version="1.0"
encoding="UTF-8" standalone="yes"?>
                         <Document
xsi:schemaLocation="xsd/pain.001.001.05.xsd"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                         <CstmrCdtTrfInitn>
                    <GrpHdr>
                        <MsgId>CRTBREDKHPPXX000001912</MsgId>
                        <CreDtTm>2019-09-02T09:45:50.000+07:00</CreDtTm>
                        <NbOfTxs>1</NbOfTxs>
                        <CtrlSum>1000</CtrlSum>
                        <InitgPty>
                            <Nm>Bred Bank Cambodia Plc HQ-001</Nm>
                        </InitgPty>
                    </GrpHdr>
                    <PmtInf>
                        <PmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1912</PmtInfId>
                        <PmtMtd>TRF</PmtMtd>
                        <BtchBookg>false</BtchBookg>
                        <ReqdExctnDt>2019-09-02T09:45:50</ReqdExctnDt>
                        <Dbtr>
                            <Nm>Vannak</Nm>
                        </Dbtr>
                        <DbtrAcct>
                            <Id>
                                <Othr>
                                    <Id>000000000001</Id>
                                </Othr>
                            </Id>
                            <Ccy>USD</Ccy>
                        </DbtrAcct>
                        <DbtrAgt>
                            <FinInstnId>
                                <BICFI>BREDXXXX</BICFI>
                            </FinInstnId>
                        </DbtrAgt>
                        <CdtTrfTxInf>
                            <PmtId>
                                <InstrId>PRINKHPPXXXX/AAAA0139</InstrId>

<EndToEndId>001000008600027-00002896422111</EndToEndId>
                            </PmtId>
                            <Amt>
                                <InstdAmt Ccy="USD">1000</InstdAmt>
                            </Amt>
                            <ChrgBr>CRED</ChrgBr>
                            <CdtrAgt>
                                <FinInstnId>
                                    <BICFI>PRINXXXX</BICFI>
                                </FinInstnId>
                            </CdtrAgt>
                            <Cdtr>
                                <Nm>Leng Alisa</Nm>
                            </Cdtr>
                            <CdtrAcct>
                                <Id>
                                    <Othr>
                                        <Id>000000000002</Id>
                                    </Othr>
                                </Id>
                            </CdtrAcct>
                            <Purp>
                                <Cd>GDDS</Cd>
                            </Purp>
                            <RmtInf>
                                <Ustrd>RCVD</Ustrd>
                                <Strd>
                                    <RfrdDocInf>
                                        <Tp>
                                            <CdOrPrtry>
                                                <Cd>CINV</Cd>
                                            </CdOrPrtry>
                                        </Tp>
                                        <Nb>987-AC</Nb>
                                        <RltdDt>2020-09-02</RltdDt>
                                    </RfrdDocInf>
                                    <AddtlRmtInf>Phnom
Penh/AAAA0139</AddtlRmtInf>
                                </Strd>
                            </RmtInf>
                        </CdtTrfTxInf>
                    </PmtInf>
                         </CstmrCdtTrfInitn>
                         </Document>]]></iso_message>
                   <import_file_name></import_file_name>
            </fas:makeFullFundTransfer>
      </soapenv:Body>
</soapenv:Envelope>
```

b) Response

return	pain.002.001.06

Sample of the response for a successful request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeFullFundTransferResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPBREDKHPPXX000000160</MsgId>
            <CreDtTm>2020-09-05T10:57:33.707</CreDtTm>
            <InitgPty>
                        <Nm>Bred Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>BREDKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>PRINKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-09-02T09:45:50.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1912</OrgnlPmtInfId>
            <TxInfAndSts>
<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                    <TxSts>ACSP</TxSts>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Vannak</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeFullFundTransferResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeFullFundTransferResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPBREDKHPPXX000000170</MsgId>
            <CreDtTm>2020-09-05T10:59:09.441</CreDtTm>
            <InitgPty>
                        <Nm>Bred Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>BREDKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
```

```
                    <BICFI>PRINKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-09-02T09:45:50.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1912</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                    <TxSts>RJCT</TxSts>
                    <StsRsnInf>
                        <Orgtr>
                            <id>
                                <OrgId>
                                    <AnyBIC>BREDKHPPXX<AnyBIC>
                                <OrgId>
                            </id>
                        </Orgtr>
                        <AddtlInf>REJECTED AT CORE DUE TO: [This
original payment info id is already existed.]</AddtlInf>
                    </StsRsnInf>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="EUR">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Vannak</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeFullFundTransferResponse>
   </SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

makeFullFundTransfer (FAST to Receiver Participant)

a) Description

The FAST system invokes this API to make fund transfer in a full automation process. In such that the transaction is signed and sent by the FAST Transfer Service to the receiver bank CBS through the receiver Client Module (CM).

b) Request

Parameter	**Description**
cm_user_name	Username of the FAST in the receiver participant bank CBS
cm_password	Password of the FAST in the receiver participant bank CBS
iso_message	pain.001.001.05
import_file_name	Name of the xml file which stores batch transaction

Sample of the request

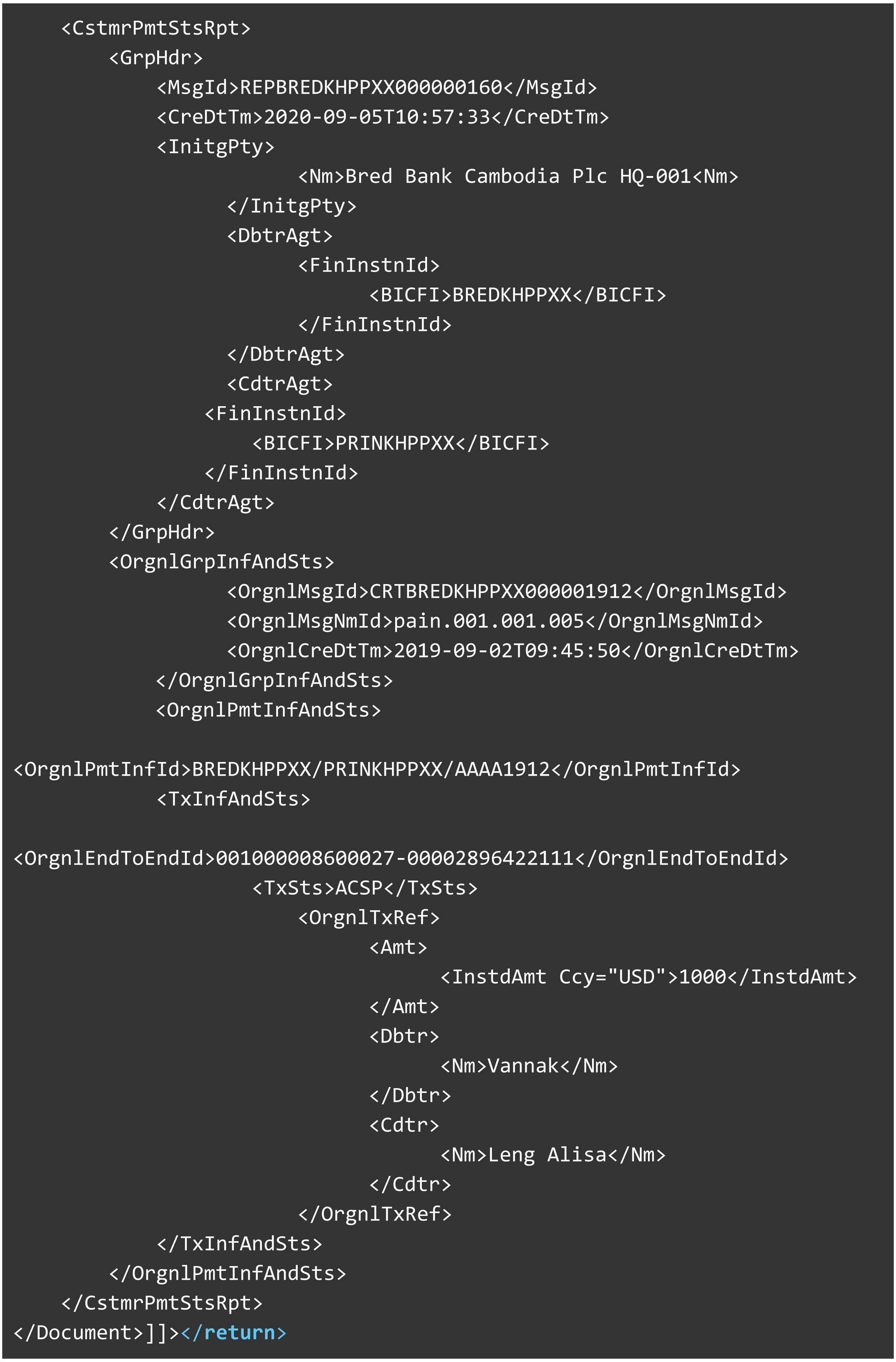
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
      <soapenv:Header/>
            <soapenv:Body>
                  <fas:makeFullFundTransfer>
                        <cm_user_name>bred</cm_user_name>
                        <cm_password>aA12345@</cm_password>
                        <iso_message><![CDATA[<?xml version="1.0"
encoding="UTF-8" standalone="yes"?>
                        <Document
xsi:schemaLocation="xsd/pain.001.001.05.xsd"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                        <CstmrCdtTrfInitn>
                    <GrpHdr>
                        <MsgId>CRTBREDKHPPXX000001912</MsgId>
                        <CreDtTm>2019-09-02T09:45:50</CreDtTm>
                        <NbOfTxs>1</NbOfTxs>
                        <CtrlSum>1000</CtrlSum>
                        <InitgPty>
                            <Nm>Bred Bank Cambodia Plc HQ-001</Nm>
                        </InitgPty>
                    </GrpHdr>
                    <PmtInf>

<PmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1912</PmtInfId>
                        <PmtMtd>TRF</PmtMtd>
                        <BtchBookg>false</BtchBookg>
                        <ReqdExctnDt>2019-09-02T09:45:50</ReqdExctnDt>
                        <Dbtr>
                            <Nm>Vannak</Nm>
                        </Dbtr>
                        <DbtrAcct>
                            <Id>
                                <Othr>
                                    <Id>000000000001</Id>
                                </Othr>
                            </Id>
                            <Ccy>USD</Ccy>
                        </DbtrAcct>
                        <DbtrAgt>
                            <FinInstnId>
                                <BICFI>BREDXXXX</BICFI>
                            </FinInstnId>
                        </DbtrAgt>
                        <CdtTrfTxInf>
                            <PmtId>
                                <InstrId>PRINKHPPXXXX/AAAA0139</InstrId>

<EndToEndId>001000008600027-00002896422111</EndToEndId>
                            </PmtId>
                            <Amt>
                                <InstdAmt Ccy="USD">1000</InstdAmt>
                            </Amt>
                            <ChrgBr>CRED</ChrgBr>
                            <CdtrAgt>
                                <FinInstnId>
                                    <BICFI>PRINXXXX</BICFI>
                                </FinInstnId>
                            </CdtrAgt>
                            <Cdtr>
                                <Nm>Leng Alisa</Nm>
                            </Cdtr>
                            <CdtrAcct>
                                <Id>
                                    <Othr>
                                        <Id>000000000002</Id>
                                    </Othr>
                                </Id>
                            </CdtrAcct>
                            <Purp>
                                <Cd>GDDS</Cd>
                            </Purp>
                            <RmtInf>
                                <Ustrd>RCVD</Ustrd>
                                <Strd>
                                    <RfrdDocInf>
                                        <Tp>
                                            <CdOrPrtry>
                                                <Cd>CINV</Cd>
                                            </CdOrPrtry>
                                        </Tp>
                                        <Nb>987-AC</Nb>
                                        <RltdDt>2020-09-02</RltdDt>
                                    </RfrdDocInf>
                                    <AddtlRmtInf>Phnom 
Penh/AAAA0139</AddtlRmtInf>
                                </Strd>
                            </RmtInf>
                        </CdtTrfTxInf>
                    </PmtInf>
                        </CstmrCdtTrfInitn>
                        </Document>]]></iso_message>
                  <import_file_name></import_file_name>
            </fas:makeFullFundTransfer>
      </soapenv:Body>
</soapenv:Envelope>
```

c) Response

return	pain.002.001.06

Sample of the response for a successful request

```
<SOAP-ENV:Envelope 
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeFullFundTransferResponse 
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
```



```
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPBREDKHPPXX000000160</MsgId>
            <CreDtTm>2020-09-05T10:57:33</CreDtTm>
            <InitgPty>
                        <Nm>Bred Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>BREDKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>PRINKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>
                  <OrgnlCreDtTm>2019-09-02T09:45:50</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1912</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                    <TxSts>ACSP</TxSts>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Vannak</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
```

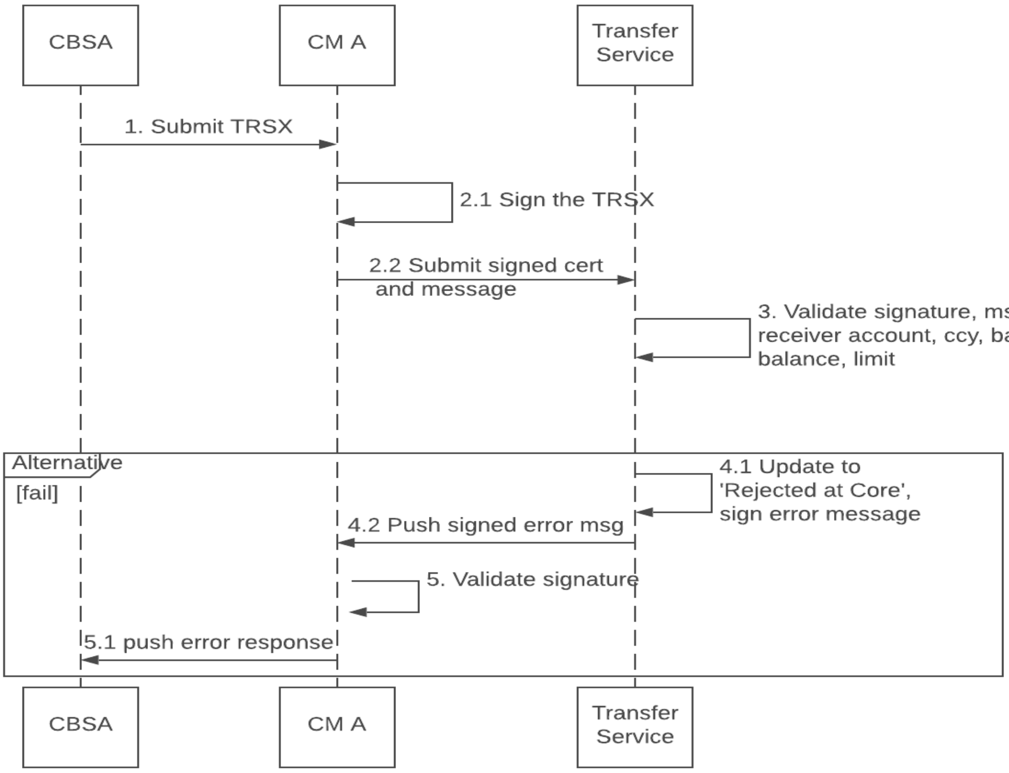
```
      </ns3:makeFullFundTransferResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeFullFundTransferResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPBREDKHPPXX000000170</MsgId>
            <CreDtTm>2020-09-05T10:59:09</CreDtTm>
            <InitgPty>
                        <Nm>Bred Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>BREDKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>PRINKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>
                  <OrgnlCreDtTm>2019-09-02T09:45:50</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>
<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1912</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                    <TxSts>RJCT</TxSts>
                    <StsRsnInf>
                        <Orgtr>
```

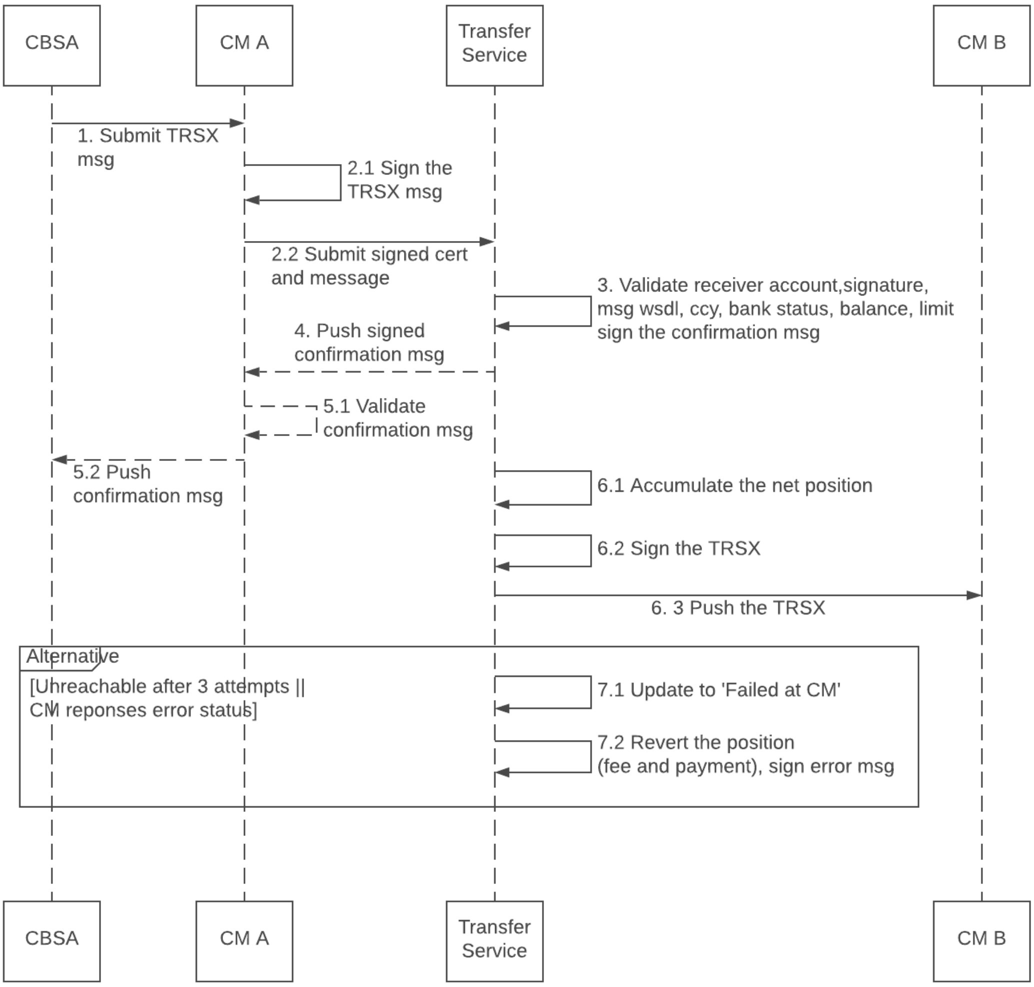
```
                            <id>
                                <OrgId>
                                    <AnyBIC>BREDKHPPXX<AnyBIC>
                                <OrgId>
                            </id>
                        </Orgtr>
                        <AddtlInf>REJECTED AT CORE DUE TO: [This
original payment info id is already existed.]</AddtlInf>
                    </StsRsnInf>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="EUR">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Vannak</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeFullFundTransferResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.5 Account-based Transfer (Full STP - Rejected At Core)


CBSA
CM A
Transfer Service
1. Submit TRSX
2.1 Sign the TRSX
2.2 Submit signed cert and message
3. Validate signature, msg wsdl, receiver account, ccy, bank status, balance, limit
Alternative
[fail]
4.1 Update to 'Rejected at Core', sign error message
4.2 Push signed error msg
5. Validate signature
5.1 push error response
CBSA
CM A
Transfer Service

1. CBS of Bank A submits a fund transfer message using ISO 20022 format (*pain 001.001.05)* by calling **[makeFullFundTransfer]** via SOAP web service into Bank A CM
2. Bank A CM hashes and signs on the message using the private key stored in CM sends it to FAST Core Engine (CE) Transfer Service
3. Transfer Service applies all the validation rules on the message and checks if it complies (message WSDL, digital signature, receiver account, receiver bank status, currency, operation limit, ability to pay)
4. If it is invalid, the transaction and the message are stored in the system with status **Rejected At Core**. Transfer Service responds with a corresponding error message to sender Bank A CM using *pain 002.001.06* with transaction status (Txsts) of **Rejected (RJCT)**.
5. CM of Bank A validates the signature of the error message and sends it to its CBS to credit the balance back to the their customer

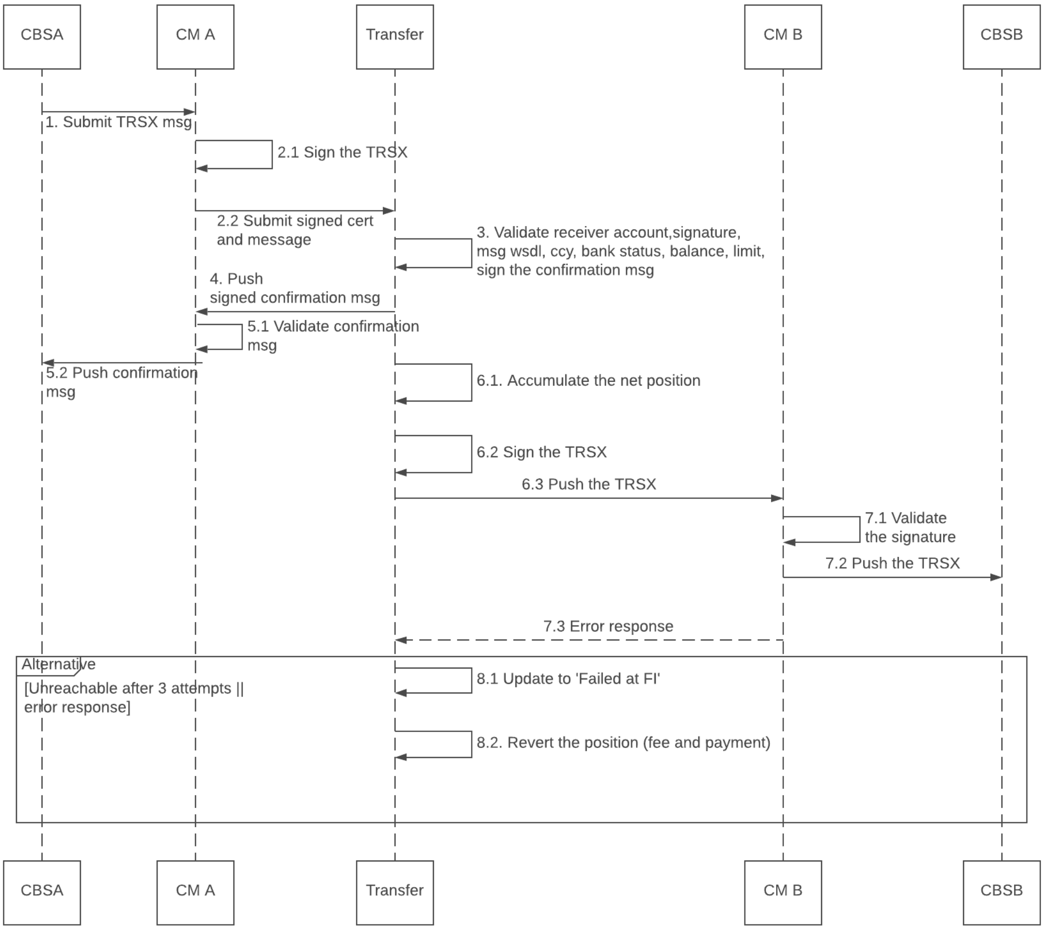
2.6 Account-based Transfer (Full STP - Failed at CM)


CBSA
CM A
Transfer Service
CM B
1. Submit TRSX msg
2.1 Sign the TRSX msg
2.2 Submit signed cert and message
3. Validate receiver account,signature, msg wsdl, ccy, bank status, balance, limit sign the confirmation msg
4. Push signed confirmation msg
5.1 Validate confirmation msg
5.2 Push confirmation msg
6.1 Accumulate the net position
6.2 Sign the TRSX
6. 3 Push the TRSX
Alternative
[Unreachable after 3 attempts || CM reponses error status]
7.1 Update to 'Failed at CM'
7.2 Revert the position (fee and payment), sign error msg
CBSA
CM A
Transfer Service
CM B

1. Please refer to *2.3 Account-based Transfer (Full STP - Success)* for the description of the message flow from step 1 to step 6.3
7. If Bank B CM is unreachable after three back-off retries (60s, 60s, 60s) or the CM of bank B responds with an error status, the transaction is considered to be **Failed at CM**. Transfer Service bears the responsibility to reverse the fund transfer. The transaction status is updated to **Failed at CM**, and the net position of fee and inter-payment are reversed.

   Note: To check the latest status of the transaction, sender bank pulls Outgoing Transfers

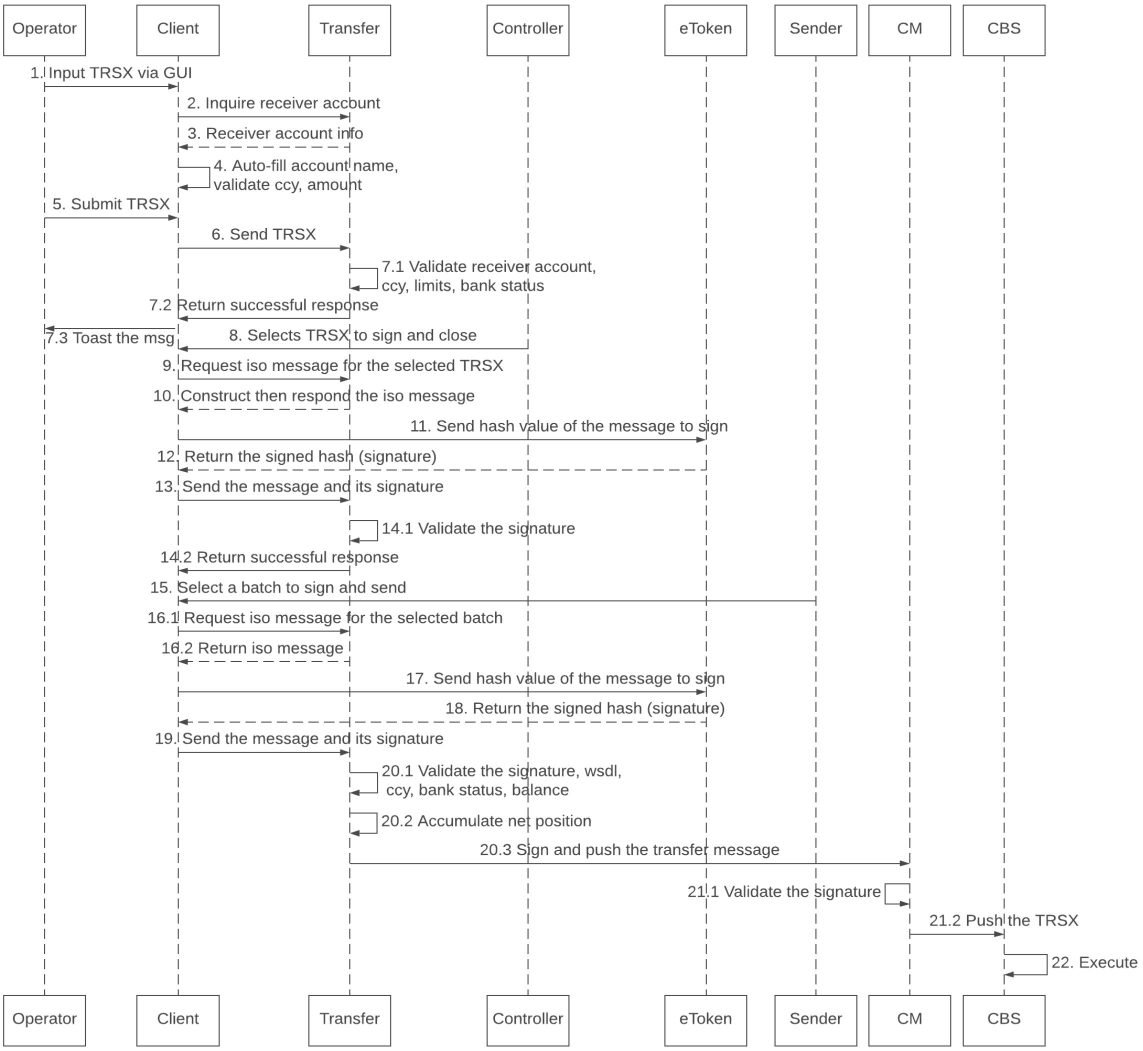
2.7 Account-based Transfer (Full STP - Failed at FI)


CBSA
CM A
Transfer
CM B
CBSB
1. Submit TRSX msg
2.1 Sign the TRSX
2.2 Submit signed cert and message
3. Validate receiver account,signature, msg wsdl, ccy, bank status, balance, limit, sign the confirmation msg
4. Push signed confirmation msg
5.1 Validate confirmation msg
5.2 Push confirmation msg
6.1. Accumulate the net position
6.2 Sign the TRSX
6.3 Push the TRSX
7.1 Validate the signature
7.2 Push the TRSX
7.3 Error response
Alternative
[Unreachable after 3 attempts || error response]
8.1 Update to 'Failed at FI'
8.2. Revert the position (fee and payment)
CBSA
CM A
Transfer
CM B
CBSB

1. Please refer to *2.3 Account-based Transfer (Full STP - Success)* for the description of the message flow from step 1 to step 7.2
7. CBS of Bank B is unreachable after three retries (60s, 60s, 60s), or the CBS of bank B responds with an error status, the transaction is considered to be **Failed at FI**. Bank B CM forwards the response to Transfer Service
8. Transfer Service bears the responsibility to reverse the fund transfer. The transaction status is updated to **Failed at FI**, and the net position of fee and inter-payment are reversed.

   Note: To check the latest status of the transaction, sender bank pulls Outgoing Transfers

2.8 Account-based Transfer (Semi STP-Manual Input, Straight-through Output-Success)

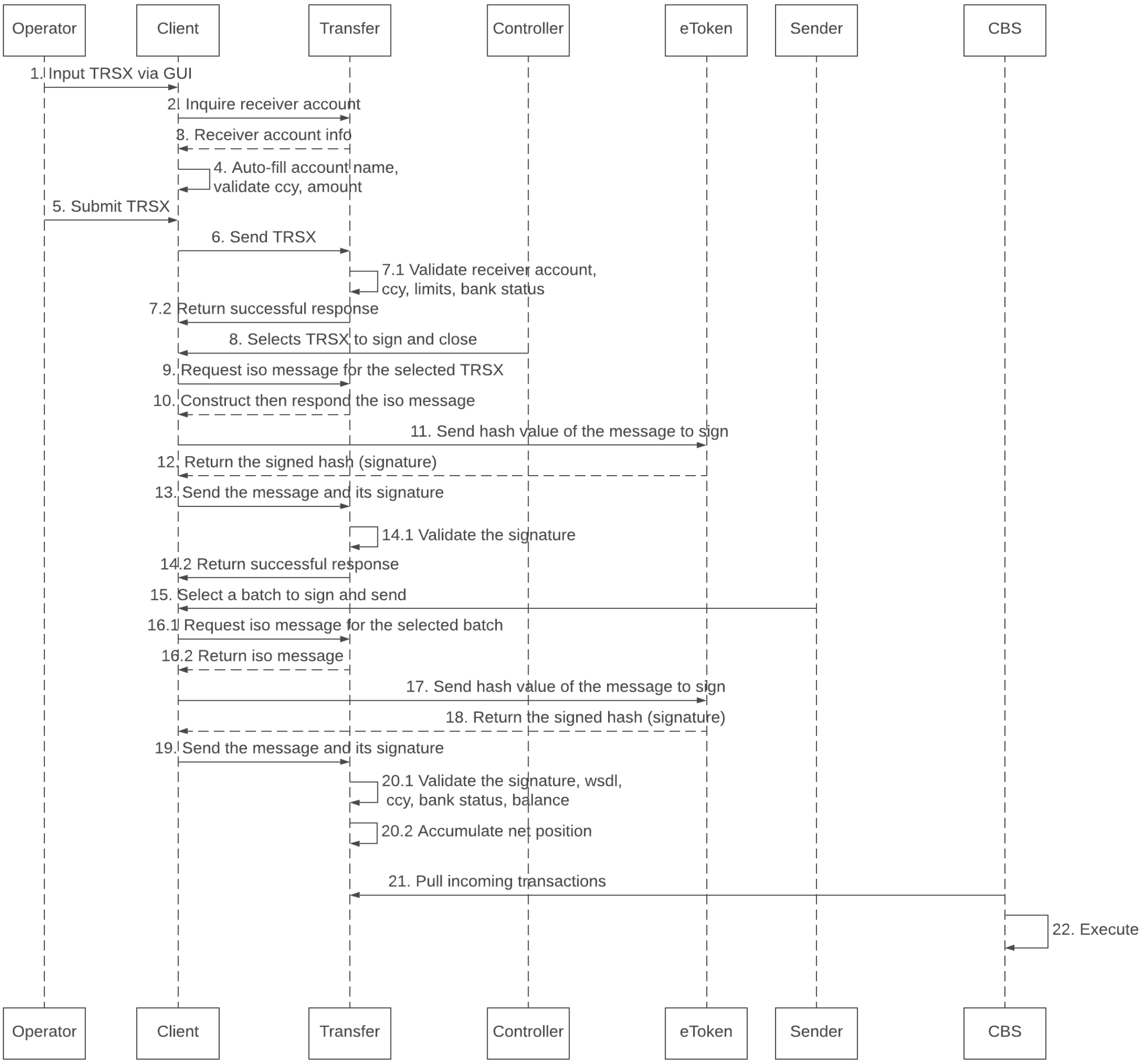


Operator
Client
Transfer
Controller
eToken
Sender
CM
CBS
1. Input TRSX via GUI
2. Inquire receiver account
3. Receiver account info
4. Auto-fill account name, validate ccy, amount
5. Submit TRSX
6. Send TRSX
7.1 Validate receiver account, ccy, limits, bank status
7.2 Return successful response
7.3 Toast the msg
8. Selects TRSX to sign and close
9. Request iso message for the selected TRSX
10. Construct then respond the iso message
11. Send hash value of the message to sign
12. Return the signed hash (signature)
13. Send the message and its signature
14.1 Validate the signature
14.2 Return successful response
15. Select a batch to sign and send
16.1 Request iso message for the selected batch
16.2 Return iso message
17. Send hash value of the message to sign
18. Return the signed hash (signature)
19. Send the message and its signature
20.1 Validate the signature, wsdl, ccy, bank status, balance
20.2 Accumulate net position
20.3 Sign and push the transfer message
21.1 Validate the signature
21.2 Push the TRSX
22. Execute
Operator
Client
Transfer
Controller
eToken
Sender
CM
CBS

1. The operator goes to Account-based Transfer screen and inputs the transfer information
2. Once the Operator inputs the receiver account number, the FAST client will call the Receiver Customer Account Inquiry API to retrieve the account information. To simplify the diagram, the account inquiry at receiver CBS flow is omitted, please refer to 2.2 Customer Account Inquiry (Manual Transfer) for the details
3. Transfer Service responds back the account information to the client
4. The client auto-fills the account name, validate ccy and the amount (against the operation limits)
5. Operator clicks Submit button
6. The client sends this transaction to Transfer Service

7. Transfer Service applies the validation rules on the transaction and checks if it complies (receiver account, receiver bank status, currency, operation limit). And, the Transfer Service responds to the client of the result, and a successful message is toasted for the operator
8. The controller can view the pending transactions on the Pending Transactions screen. He/she can click multiple transactions to sign and close using eToken
9. After clicking Sign and Close, the client will request the Transfer Service to form the corresponding ISO message pain 001.001.05
10. Transfer Service constructs the message and returns it to the client
11. The client communicates with the eToken plugin to send the message to eToken for signing. Pre-condition: Controller has inputted the right pin for the eToken
12. After successfully signing, the signed hash (digital signature) will be sent back to the client
13. The client sends the original message and its digital signature to Transfer Service
14. The Transfer Service validates the signature of the message
15. The sender can view the closed batches on the Closed Batches screen. He/she can click one batch to sign and send using eToken
16. After clicking Sign and Send, the client will request the Transfer Service to form the corresponding ISO message. Transfer Service returns the message to the client
17. The client communicates with the eToken plugin to send the message to eToken for signing. Pre-condition: Sender has inputted the right pin for the eToken
18. After successfully signing, the signed hash (digital signature) will be sent back to the client
19. The client sends the original message and its digital signature to Transfer Service
20. Transfer Service applies all the validation rules on the message and checks if it complies (message WSDL, digital signature, ability to pay). If it is valid, the net positions of the banks in the batch are accumulated, and the status of the transaction is updated to **Signed and Sent**. Then, Transfer Service constructs and signs the iso messages pain 001.001.05 (outgoing) to the corresponding CM of the receiver banks
21. CM of the receiver bank B validates the signature and pushes it to its CBS by calling **[makeFullFundTransfer]** via SOAP
22. CBS of Bank B bears the responsibility to validate and verify the incoming message. If it is valid, Bank B has to credit the fund into its customer account and notify the customer. Please refer to the 2.14 Acknowledgement (STP via API) for the subsequent flow

2.9 Receiver Bank Pulls Incoming Transfers (Manual Input - Pull Incoming)

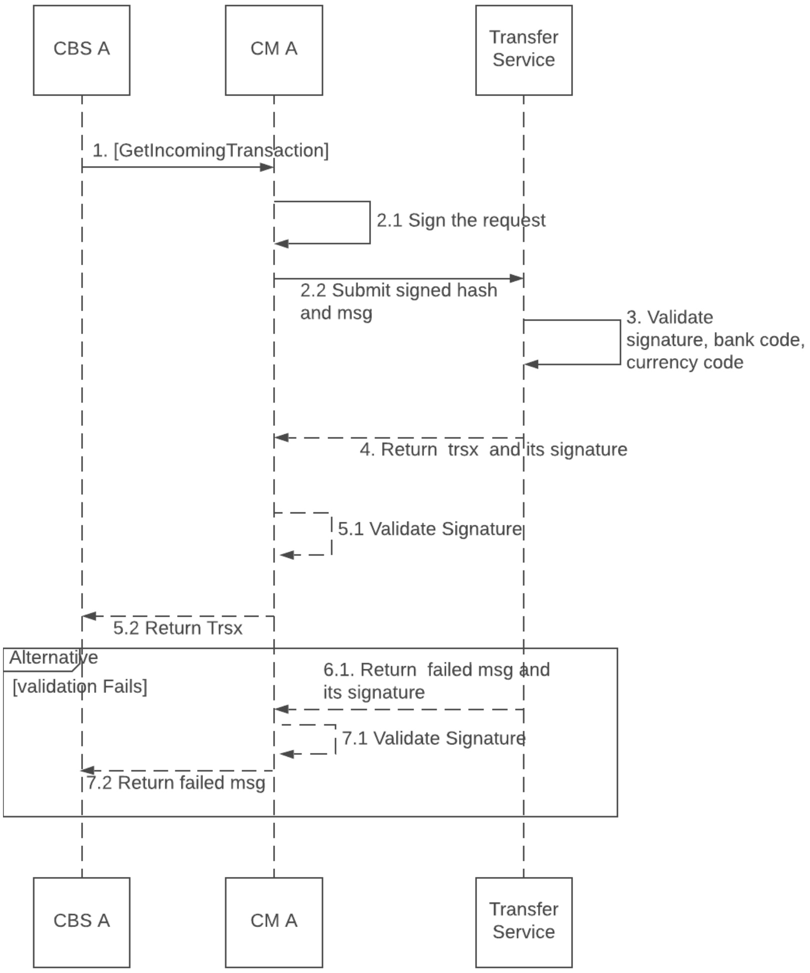

Operator
Client
Transfer
Controller
eToken
Sender
CBS
1. Input TRSX via GUI
2. Inquire receiver account
3. Receiver account info
4. Auto-fill account name, validate ccy, amount
5. Submit TRSX
6. Send TRSX
7.1 Validate receiver account, ccy, limits, bank status
7.2 Return successful response
8. Selects TRSX to sign and close
9. Request iso message for the selected TRSX
10. Construct then respond the iso message
11. Send hash value of the message to sign
12. Return the signed hash (signature)
13. Send the message and its signature
14.1 Validate the signature
14.2 Return successful response
15. Select a batch to sign and send
16.1 Request iso message for the selected batch
16.2 Return iso message
17. Send hash value of the message to sign
18. Return the signed hash (signature)
19. Send the message and its signature
20.1 Validate the signature, wsdl, ccy, bank status, balance
20.2 Accumulate net position
21. Pull incoming transactions
22. Execute
Operator
Client
Transfer
Controller
eToken
Sender
CBS

1. Please refer to *2.7 Account-based Transfer (Semi STP - Manual Input, Straight-through Output - Success)* for the description of the message flow from step 1 to step 20.2
21. CBS of Bank B fetches the incoming transactions, which are not yet acknowledged, via SOAP web service by calling **[getIncomingTransaction]** from FAST Transfer Service in the format of *pain 001.001.05*
22. CBS of Bank B bears the responsibility to validate and verify the incoming message. If it is valid, Bank B has to credit the fund into its customer account and notify the customer. Please refer to the 2.14 Acknowledgement (STP via API) for the subsequent flow

Remark: 2.10 illustrates the process flow of the pull incoming transaction

2.10 Pull Incoming Transfer [getIncomingTransaction]

2.10.1 Message Flow



1. CBS of Bank A submits a request go get incoming transactions by calling [getIncomingTransaction] via SOAP web service into Bank A CM
2. Bank A CM hashes and signs on the request using the private key stored in CM sends it to FAST Core Engine (CE) Transfer Service
3. Transfer Service applies all the validation rules on the message and checks if it complies (message WSDL, digital signature, currency, bank code)
4. If it is valid, Transfer Service will return the signed and sent transactions of the bank with current business day to Bank A CM, and updates those transactions to **Received at FI**
5. Bank A CM validates the message signature and pushes to CBS, and thus CBS can credit their customer accordingly. To finalize the transaction flow, CBS also need to [**makeAcknowledgment**] over the transactions it received via SOAP web service
6. If it is invalid, Transfer Service will construct an error message and respond to Bank A CM
7. Bank A CM validates the signature and returns the failed message to the CBS

2.10.2 Web Service

a) Description

The receiver participant bank invokes this API to fetch the incoming transactions from the FAST Transfer Service. The receiving message contains multiple transactions that are not yet acknowledged. The transaction date will only take the current date of the system.

"instruction_ref" in the tag <PmtInfId></PmtInfId> is unique. Before credit to the customer, FI needs to manually double check before credit.

b) Request

Parameter	**Description**
cm_user_name	Username of the CM which makes the request
cm_password	Password of the CM which makes the request
payee_participant_code	The receiver bank code

Sample of the request

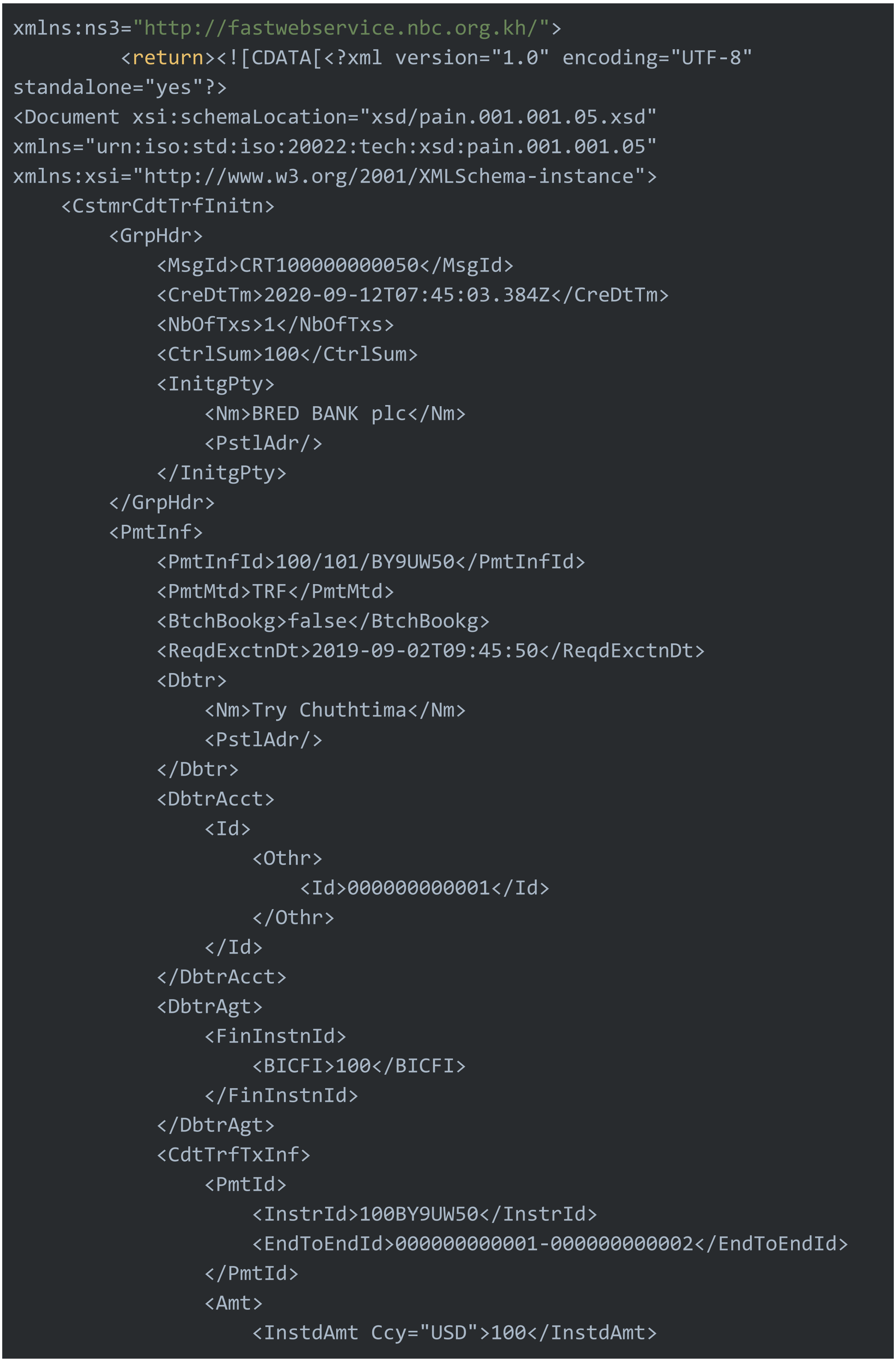
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
   <soapenv:Header/>
   <soapenv:Body>
      <fas:getIncomingTransaction>
         <cm_user_name>prince-adapter</cm_user_name>
            <cm_password>aA12345@</cm_password>
            <ccy>USD</ccy>
            <payee_participant_code>CMPLBPRIPP</payee_participant_code>
      </fas:getIncomingTransaction>
   </soapenv:Body>
</soapenv:Envelope>
```

c) Response

iso_message	pain.001.001.05 (success) or custom (fail)

Sample of the response for a successful request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:getIncomingTransactionResponse
```



```
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<Document xsi:schemaLocation="xsd/pain.001.001.05.xsd"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>CRT100000000050</MsgId>
            <CreDtTm>2020-09-12T07:45:03.384Z</CreDtTm>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>100</CtrlSum>
            <InitgPty>
                <Nm>BRED BANK plc</Nm>
                <PstlAdr/>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>100/101/BY9UW50</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>false</BtchBookg>
            <ReqdExctnDt>2019-09-02T09:45:50</ReqdExctnDt>
            <Dbtr>
                <Nm>Try Chuthtima</Nm>
                <PstlAdr/>
            </Dbtr>
            <DbtrAcct>
                <Id>
                    <Othr>
                        <Id>000000000001</Id>
                    </Othr>
                </Id>
            </DbtrAcct>
            <DbtrAgt>
                <FinInstnId>
                    <BICFI>100</BICFI>
                </FinInstnId>
            </DbtrAgt>
            <CdtTrfTxInf>
                <PmtId>
                    <InstrId>100BY9UW50</InstrId>
                    <EndToEndId>000000000001-000000000002</EndToEndId>
                </PmtId>
                <Amt>
                    <InstdAmt Ccy="USD">100</InstdAmt>
                </Amt>
                <ChrgBr>CRED</ChrgBr>
                <CdtrAgt>
                    <FinInstnId>
                        <BICFI>101</BICFI>
                    </FinInstnId>
                </CdtrAgt>
                <Cdtr>
                    <Nm>Leng Alisa</Nm>
                    <PstlAdr/>
                </Cdtr>
                <Purp/>
                <RmtInf>
                    <Ustrd>RCVD</Ustrd>
                    <Strd>
                        <RfrdDocInf>
                            <Tp>
                                <CdOrPrtry>
                                    <Cd>CINV</Cd>
                                </CdOrPrtry>
                            </Tp>
                            <Nb>50</Nb>
                            <RltdDt>2019-09-02Z</RltdDt>
                        </RfrdDocInf>
                        <AddtlRmtInf></AddtlRmtInf>
                    </Strd>
                </RmtInf>
            </CdtTrfTxInf>
        </PmtInf>
    </CstmrCdtTrfInitn>
</Document>]]></return>
      </ns3:getIncomingTransactionResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

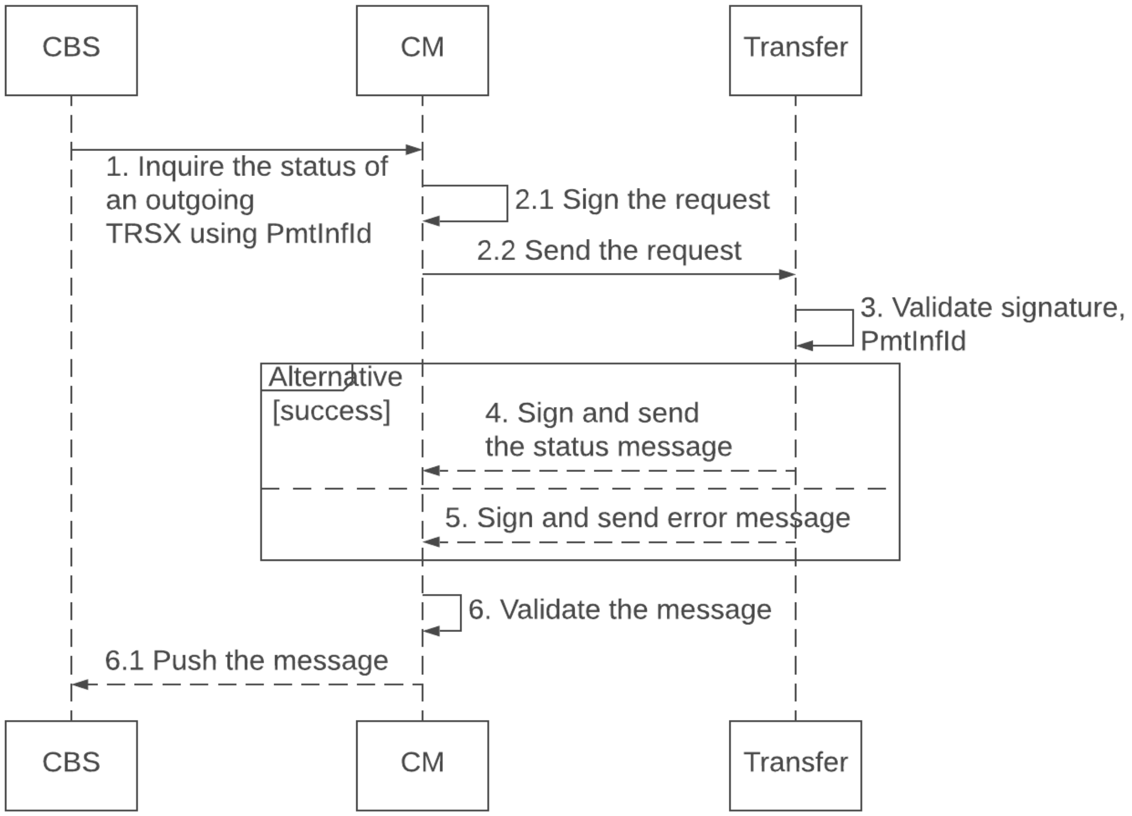
Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:getIncomingTransactionResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return>[CurrencyCode[EUR] is not found.]</return>
      </ns3:getIncomingTransactionResponse>
```

```
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.11 Sender Bank Pulls Outgoing Transfers

2.11.1 Message Flow



CBS
CM
Transfer
1. Inquire the status of an outgoing TRSX using PmtInfId
2.1 Sign the request
2.2 Send the request
3. Validate signature, PmtInfId
Alternative
[success]
4. Sign and send the status message
5. Sign and send error message
6. Validate the message
6.1 Push the message
CBS
CM
Transfer

1. CBS of the sender bank calls **[getOutgoingTransactionByPmtInfId]** via SOAP web service to retrieve the status of an outgoing transaction by PmtInfId
2. CM of the sender bank signs the request and sends to the Transfer Service
3. Transfer Service validates the signature and the PmtInfId
4. If it is valid, Transfer Service will sign and send the corresponding transaction using pain.001.001.05
5. If it is invalid, Transfer Service will construct an error message and sends back to Bank A CM
6. CM of the sender bank validates the message and sends to its CBS

2.11.2 Web Service

a) Description

The sender participant bank invokes this API to get the specific outgoing transaction status by payment information ID.

b) Request

Parameter	Description
cm_user_name	Username of the CM which makes the request
cm_password	Password of the CM which makes the request
payer_participant_code	The sender bank code
instruction_ref	Payment information id of the outgoing transaction

Sample of the request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
   <soapenv:Header/>
   <soapenv:Body>
      <fas:getOutgoingTransactionByPmtInfId>
         <!--Optional:-->
         <cm_user_name>bred-adapter</cm_user_name>
         <!--Optional:-->
         <cm_password>aA12345@</cm_password>
         <!--Optional:-->
         <payer_participant_code>BREDKHPPXX</payer_participant_code>
         <!--Optional:-->
         <instruction_ref>AAAA5225</instruction_ref>
      </fas:getOutgoingTransactionByPmtInfId>
   </soapenv:Body>
</soapenv:Envelope>
```

c) Response

iso_message	pain.001.001.05 (success) or custom

Sample of the response for a successful request

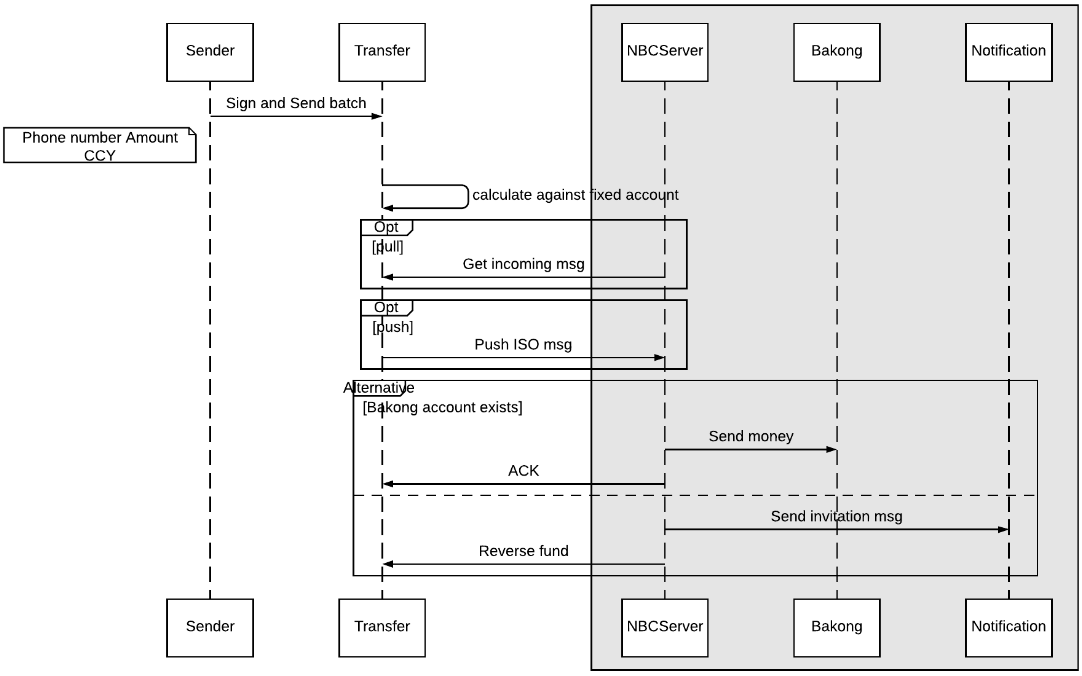
```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:getOutgoingTransactionByPmtInfIdResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<Document xsi:schemaLocation="xsd/pain.001.001.05.xsd"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <CstmrCdtTrfInitn>
        <GrpHdr>
            <MsgId>CRTBREDKHPPXX00050</MsgId>
            <CreDtTm>2020-09-30T10:00:35.382Z</CreDtTm>
            <NbOfTxs>1</NbOfTxs>
            <CtrlSum>1000</CtrlSum>
            <InitgPty>
                <Nm>BRED BANK plc</Nm>
                <PstlAdr/>
            </InitgPty>
        </GrpHdr>
        <PmtInf>
            <PmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA5225</PmtInfId>
            <PmtMtd>TRF</PmtMtd>
            <BtchBookg>false</BtchBookg>
            <ReqdExctnDt>2019-09-02T09:45:50</ReqdExctnDt>
            <Dbtr>
                <Nm>Try Chuthtima</Nm>
                <PstlAdr/>
            </Dbtr>
            <DbtrAcct>
                <Id>
                    <Othr>
                        <Id>000000000001</Id>
                    </Othr>
                </Id>
            </DbtrAcct>
            <DbtrAgt>
                <FinInstnId>
                    <BICFI>BREDKHPPXX</BICFI>
                </FinInstnId>
            </DbtrAgt>
            <CdtTrfTxInf>
                <PmtId>
                    <InstrId>100000CYKK650</InstrId>
                    <EndToEndId>000000000001-000000000002</EndToEndId>
                </PmtId>
                <Amt>
                    <InstdAmt Ccy="USD">1000</InstdAmt>
                </Amt>
                <ChrgBr>CRED</ChrgBr>
                <CdtrAgt>
                    <FinInstnId>
                        <BICFI>PRINKHPPXX</BICFI>
                    </FinInstnId>
                </CdtrAgt>
                <Cdtr>
                    <Nm>Leng Alisa</Nm>
                    <PstlAdr/>
                </Cdtr>
                <Purp>
                    <Cd>RCVD</Cd>
                </Purp>
                <RmtInf>
                    <Ustrd>RCVD</Ustrd>
                    <Strd>
                        <RfrdDocInf>
                            <Tp>
                                <CdOrPrtry>
                                    <Cd>CINV</Cd>
                                </CdOrPrtry>
                            </Tp>
                            <Nb>50</Nb>
                            <RltdDt>2019-09-02Z</RltdDt>
                        </RfrdDocInf>
                        <AddtlRmtInf></AddtlRmtInf>
                    </Strd>
                </RmtInf>
            </CdtTrfTxInf>
        </PmtInf>
    </CstmrCdtTrfInitn>
</Document>]]></return>
      </ns3:getOutgoingTransactionByPmtInfIdResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
```

```
      <ns3:getOutgoingTransactionByPmtInfIdResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return>[Can not find SenderBankId by
SenderBankSwiftCode[BREDKHPVXX]., Transaction doesn't belong to your
bank.]</return>
      </ns3:getOutgoingTransactionByPmtInfIdResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

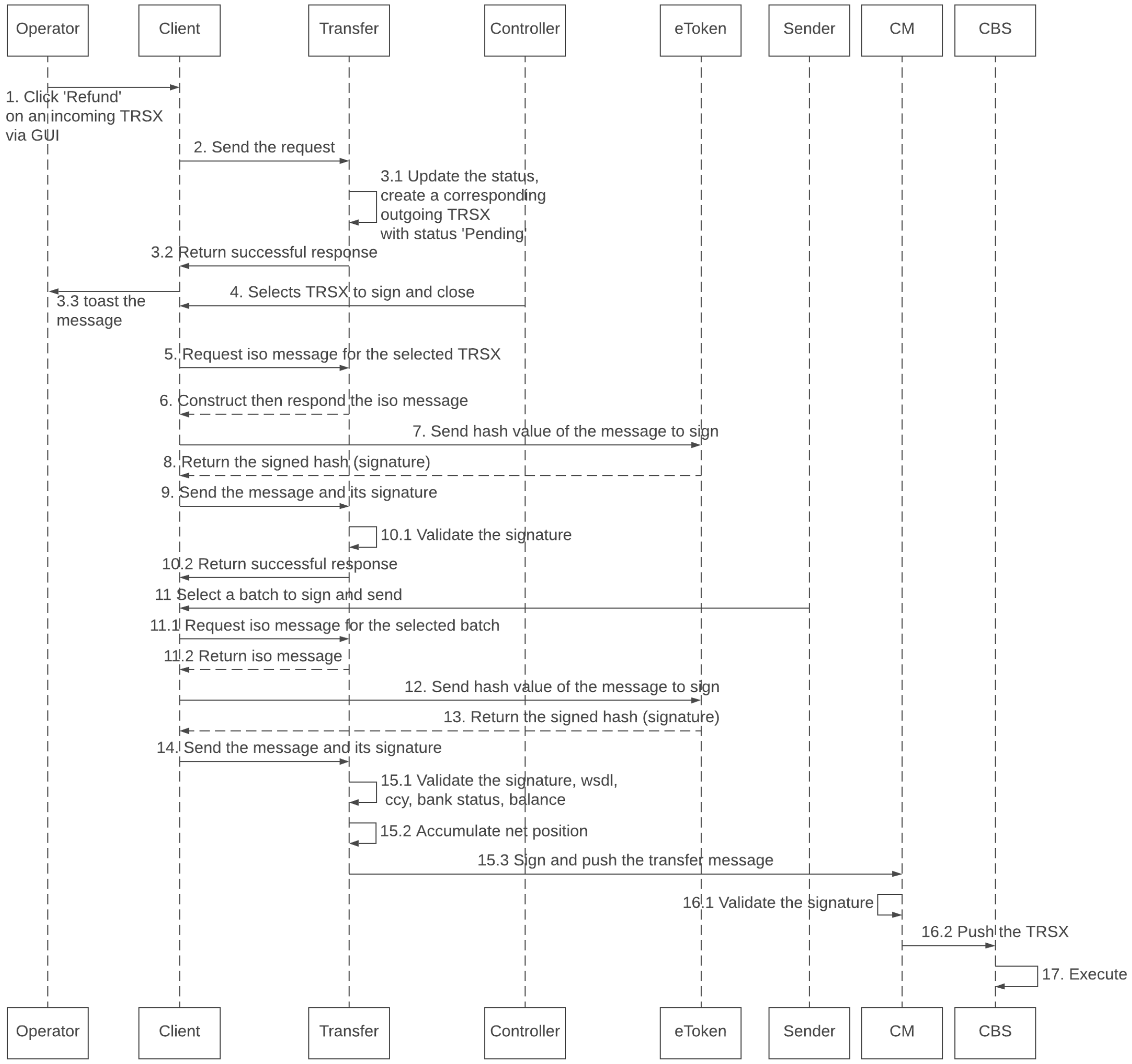
2.12 Non-account based transfer



Note: NBC implements the functionalities in the Gray Area. For every non-account based transfer, we will calculate the amount against the fixed participant bank specialized for non-account based transfer registered in FAST, operated by NBC.

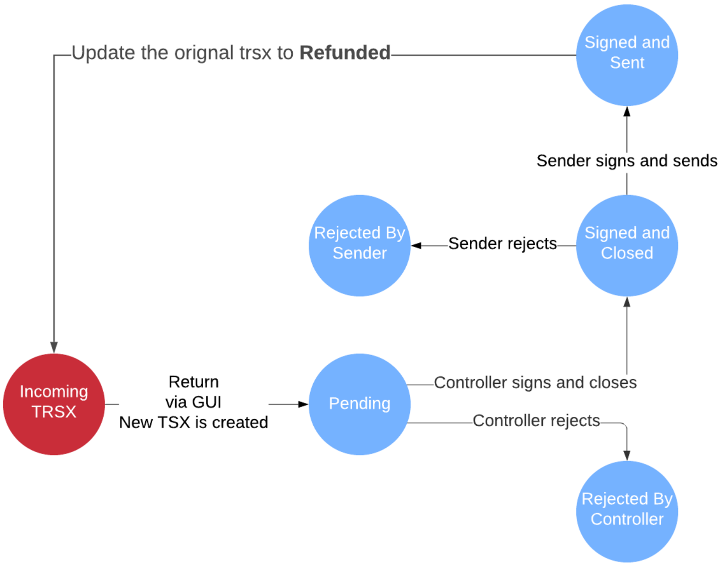
1. To simplify the diagram, the details step started from the Operator is omitted. Please refer to 2.7 Account-based Transfer (Semi STP - Manual Input, Straight-through Output - Success) for the description of the message flow (step 1 to step 20.2)
2. a. CBS (NBCServer) of the particular bank operated by NBC fetches the incoming transactions, which are not yet acknowledged, from FAST Transfer Service in the format of pain 001.001.05 by calling **[makeFullFundTransfer]** via SOAP
   b. If the STP method is chosen, CM is needed for NBC, and it will be operated in the same manner as the other FIs, in which Transfer Service signs the message pain 001.001.05 using the FAST private key and sends it to the CM. CM validates the signature on the message and sends it to its CBS using pain 001.001.05 by calling **[makeFullFundTransfer]** via SOAP

2.13 Refund (Manual via GUI)


Operator
Client
Transfer
Controller
eToken
Sender
CM
CBS
1. Click 'Refund' on an incoming TRSX via GUI
2. Send the request
3.1 Update the status, create a corresponding outgoing TRSX with status 'Pending'
3.2 Return successful response
3.3 toast the message
4. Selects TRSX to sign and close
5. Request iso message for the selected TRSX
6. Construct then respond the iso message
7. Send hash value of the message to sign
8. Return the signed hash (signature)
9. Send the message and its signature
10.1 Validate the signature
10.2 Return successful response
11 Select a batch to sign and send
11.1 Request iso message for the selected batch
11.2 Return iso message
12. Send hash value of the message to sign
13. Return the signed hash (signature)
14. Send the message and its signature
15.1 Validate the signature, wsdl, ccy, bank status, balance
15.2 Accumulate net position
15.3 Sign and push the transfer message
16.1 Validate the signature
16.2 Push the TRSX
17. Execute
Operator
Client
Transfer
Controller
eToken
Sender
CM
CBS

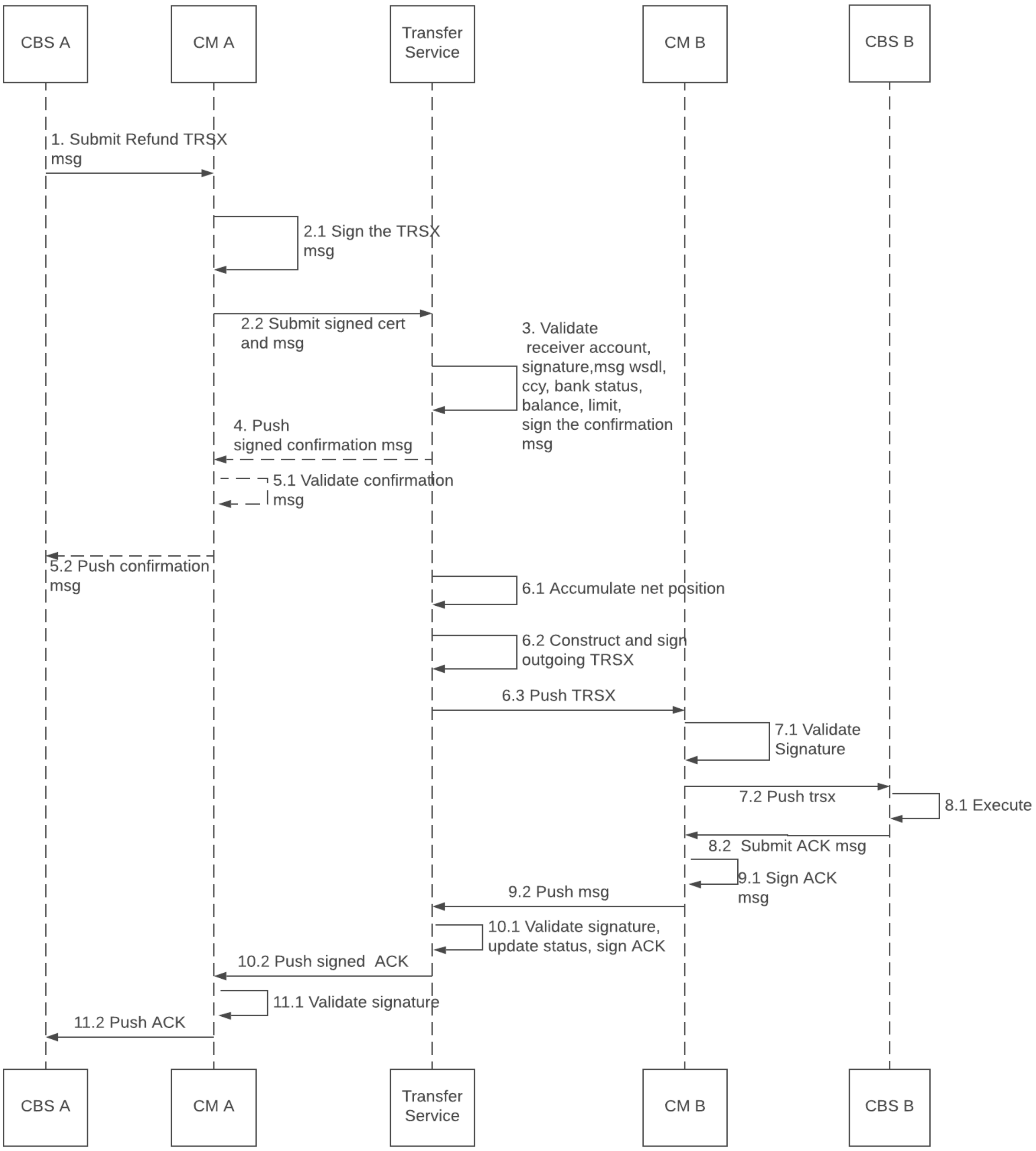
1. The operator user of Bank A goes to the *Incoming* screen and clicks ‘Refund’ on a transaction
2. The client sends the request to the Transfer Service
3. The Transfer Service creates a corresponding outgoing transaction of type refund with the status **Pending.** The transfer service returns the result to the client, and a successful message is toasted on GUI for the operator
4. Please refer to *2.7 Account-based Transfer (Semi STP - Manual Input, Straight-through Output - Success)* for the subsequent flows (step 8 onward)

   *Note: When the transaction is signed and sent, the original transaction will be updated to **Refunded**.*


Update the orignal trsx to **Refunded**
Signed and Sent
Sender signs and sends
Rejected By Sender
Sender rejects
Signed and Closed
Incoming TRSX
Return
via GUI
New TSX is created
Pending
Controller signs and closes
Controller rejects
Rejected By Controller

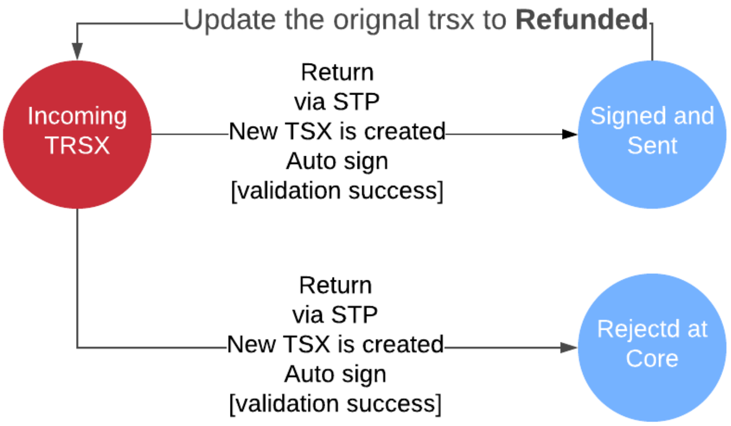
2.14 Refund (STP via API)

2.14.1 Message Flow


CBS A
CM A
Transfer Service
CM B
CBS B
1. Submit Refund TRSX msg
2.1 Sign the TRSX msg
2.2 Submit signed cert and msg
3. Validate receiver account, signature,msg wsdl, ccy, bank status, balance, limit, sign the confirmation msg
4. Push signed confirmation msg
5.1 Validate confirmation msg
5.2 Push confirmation msg
6.1 Accumulate net position
6.2 Construct and sign outgoing TRSX
6.3 Push TRSX
7.1 Validate Signature
7.2 Push trsx
8.1 Execute
8.2 Submit ACK msg
9.1 Sign ACK msg
9.2 Push msg
10.1 Validate signature, update status, sign ACK
10.2 Push signed ACK
11.1 Validate signature
11.2 Push ACK
CBS A
CM A
Transfer Service
CM B
CBS B

1. CBS of Bank A submits a refund message ISO 20022 (pain 007.001.05) by calling **[makeReverseTransaction]** via SOAP web service into Bank A CM
2. CM validates and sends to FAST, and FAST will construct, sign, and send an outgoing message *pain 001.001.05* to the receiver bank B. For the detailed flow of validation and signing. Please refer to *2.3 Account-based Transfer (Full STP - Success)*

*Note: When the transaction is signed and sent (payment status is ACSP), the original transaction will be updated to **Refunded.***


Update the orignal trsx to **Refunded**
Incoming TRSX
Return via STP New TSX is created Auto sign [validation success]
Signed and Sent
Return via STP New TSX is created Auto sign [validation success]
Rejectd at Core

2.14.2 Web Service

makeReverseTransaction (Receiver Participant to FAST)

a) Description

The receiver participant bank invokes this API to submit a refund message to reverse the transaction when the Receiver bank fails to credit its client account. Notice that only the original transaction amount is refunded.

b) Request

Parameter	**Description**
cm_user_name	Username of the CM which makes the refund
cm_password	Password of the CM which makes the refund
iso_message	pain.007.001.05
import_file_name	Name of the xml file which stores batch transaction

Sample of the request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
   <soapenv:Header/>
   <soapenv:Body>
      <fas:makeReverseTransaction>
         <cm_user_name>prince-adapter</cm_user_name>
         <cm_password>aA12345@</cm_password>
         <content_message><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.007.001.05">
  <CstmrPmtRvsl>
    <GrpHdr>
      <MsgId>REFPRINKHPPXXXX00336</MsgId>
      <CreDtTm>2019-08-30T10:46:08.000+07:00</CreDtTm>
      <NbOfTxs>1</NbOfTxs>
      <InitgPty>
        <Nm>Prince Cambodia Plc HQ-001</Nm>
      </InitgPty>
      <DbtrAgt>
        <FinInstnId>
          <BICFI>PRINKHPPXX</BICFI>
        </FinInstnId>
      </DbtrAgt>
      <CdtrAgt>
        <FinInstnId>
          <BICFI>BREDKHPPXX</BICFI>
        </FinInstnId>
      </CdtrAgt>
    </GrpHdr>
    <OrgnlGrpInf>
      <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
      <OrgnlMsgNmId>pacs.003.001.04</OrgnlMsgNmId>
      <OrgnlCreDtTm>2019-08-30T10:46:08.000+07:00</OrgnlCreDtTm>
    </OrgnlGrpInf>
    <OrgnlPmtInfAndRvsl>
      <OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1916</OrgnlPmtInfId>
      <TxInf>
        <RvslId>Reversal001</RvslId>
        <OrgnlEndToEndId>000000000001-000000000002</OrgnlEndToEndId>
        <OrgnlInstdAmt Ccy="USD">1000</OrgnlInstdAmt>
        <RvsdInstdAmt Ccy="USD">1000</RvsdInstdAmt>
        <RvslRsnInf>
          <Orgtr>
            <Nm>SYSTEM</Nm>
          </Orgtr>
          <Rsn>
            <Cd>AM05</Cd>
          </Rsn>
        </RvslRsnInf>
        <OrgnlTxRef>
          <ReqdExctnDt>2019-08-30T09:45:50</ReqdExctnDt>
          <Dbtr>
            <Nm>Try Chuthtima</Nm>
          </Dbtr>
          <DbtrAcct>
            <Id>
              <Othr>
                <Id>000000000002</Id>
              </Othr>
            </Id>
          </DbtrAcct>
          <Cdtr>
            <Nm>Leng Alisa</Nm>
          </Cdtr>
        </OrgnlTxRef>
      </TxInf>
    </OrgnlPmtInfAndRvsl>
  </CstmrPmtRvsl>
</Document>]]></content_message>
      </fas:makeReverseTransaction>
   </soapenv:Body>
</soapenv:Envelope>
```

c) Response

iso_message	pain.002.001.06

Sample of the response for a successful request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeReverseTransactionResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPPRINKHPPXX000000100</MsgId>
            <CreDtTm>2020-10-01T10:25:53.409</CreDtTm>
            <InitgPty>
                        <Nm>Prince Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>BREDKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>PRINKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-08-30T10:46:08.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1916</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>000000000001-000000000002</OrgnlEndToEndId>
                    <TxSts>ACSP</TxSts>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Try Chuthtima</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeReverseTransactionResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeReverseTransactionResponse
```

```
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPPRINKHPPXX000000110</MsgId>
            <CreDtTm>2020-10-01T10:29:17.760</CreDtTm>
            <InitgPty>
                        <Nm>Prince Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>BREDKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>PRINKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTBREDKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-08-30T10:46:08.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/AAAA1919</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>000000000001-000000000002</OrgnlEndToEndId>
                    <TxSts>RJCT</TxSts>
                    <StsRsnInf>
                        <Orgtr>
                            <id>
                                <OrgId>
                                    <AnyBIC>BREDKHPPXX<AnyBIC>
                                <OrgId>
                            </id>
                        </Orgtr>
                        <AddtlInf>REJECTED AT CORE DUE TO:
[OriginalPaymentInfoID[BREDKHPPXX/PRINKHPPXX/AAAA1919] is not
found.]</AddtlInf>
                    </StsRsnInf>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Try Chuthtima</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeReverseTransactionResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

makeReverseTransaction (FAST to Sender Participant)

a) Description

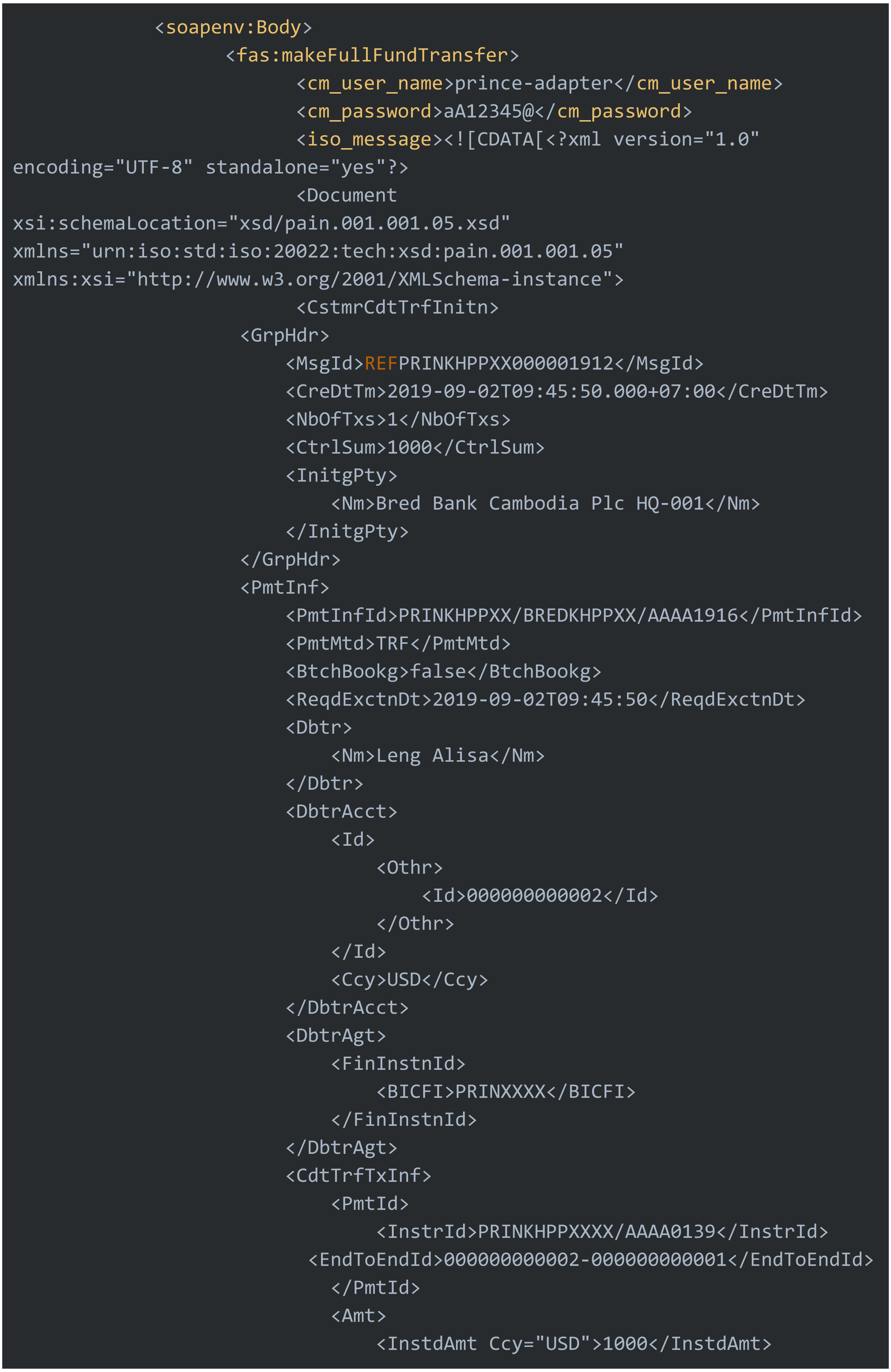
The FAST system invokes this API to submit the refund message to the sender bank CBS. In such that the message is signed and sent by the FAST Transfer Service to the sender bank CBS through the Client Module (CM).

b) Request

Parameter	**Description**
cm_user_name	Username of the FAST in the sender participant bank CBS
cm_password	Password of the FAST in the sender participant bank CBS
iso_message	pain.001.001.05
import_file_name	Name of the xml file which stores batch transaction

Sample of the request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
      <soapenv:Header/>
```



```
            <soapenv:Body>
                  <fas:makeFullFundTransfer>
                         <cm_user_name>prince-adapter</cm_user_name>
                         <cm_password>aA12345@</cm_password>
                         <iso_message><![CDATA[<?xml version="1.0"
encoding="UTF-8" standalone="yes"?>
                         <Document
xsi:schemaLocation="xsd/pain.001.001.05.xsd"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                         <CstmrCdtTrfInitn>
                    <GrpHdr>
                        <MsgId>REFPRINKHPPXX000001912</MsgId>
                        <CreDtTm>2019-09-02T09:45:50.000+07:00</CreDtTm>
                        <NbOfTxs>1</NbOfTxs>
                        <CtrlSum>1000</CtrlSum>
                        <InitgPty>
                            <Nm>Bred Bank Cambodia Plc HQ-001</Nm>
                        </InitgPty>
                    </GrpHdr>
                    <PmtInf>
                        <PmtInfId>PRINKHPPXX/BREDKHPPXX/AAAA1916</PmtInfId>
                        <PmtMtd>TRF</PmtMtd>
                        <BtchBookg>false</BtchBookg>
                        <ReqdExctnDt>2019-09-02T09:45:50</ReqdExctnDt>
                        <Dbtr>
                            <Nm>Leng Alisa</Nm>
                        </Dbtr>
                        <DbtrAcct>
                            <Id>
                                <Othr>
                                    <Id>000000000002</Id>
                                </Othr>
                            </Id>
                            <Ccy>USD</Ccy>
                        </DbtrAcct>
                        <DbtrAgt>
                            <FinInstnId>
                                <BICFI>PRINXXXX</BICFI>
                            </FinInstnId>
                        </DbtrAgt>
                        <CdtTrfTxInf>
                            <PmtId>
                                <InstrId>PRINKHPPXXXX/AAAA0139</InstrId>
                          <EndToEndId>000000000002-000000000001</EndToEndId>
                            </PmtId>
                            <Amt>
                                <InstdAmt Ccy="USD">1000</InstdAmt>
                            </Amt>
                            <ChrgBr>CRED</ChrgBr>
                            <CdtrAgt>
                                <FinInstnId>
                                    <BICFI>BREDXXXX</BICFI>
                                </FinInstnId>
                            </CdtrAgt>
                            <Cdtr>
                                <Nm>Try Chuthtima</Nm>
                            </Cdtr>
                            <CdtrAcct>
                                <Id>
                                    <Othr>
                                        <Id>000000000001</Id>
                                    </Othr>
                                </Id>
                            </CdtrAcct>
                            <Purp>
                                <Cd>REFU</Cd>
                            </Purp>
                            <RmtInf>
                                <Ustrd>RCVD</Ustrd>
                                <Strd>
                                    <RfrdDocInf>
                                        <Tp>
                                            <CdOrPrtry>
                                                <Cd>CINV</Cd>
                                            </CdOrPrtry>
                                        </Tp>
                                        <Nb>REVERSING 
PRIN10009/AAAA1916</Nb>
                                        <RltdDt>2019-09-02</RltdDt>
                                    </RfrdDocInf>
                                    <AddtlRmtInf>REVERSING 
PRIN10009/AAAA1916</AddtlRmtInf>
                                </Strd>
                            </RmtInf>
                        </CdtTrfTxInf>
                    </PmtInf>
                         </CstmrCdtTrfInitn>
                         </Document>]]></iso_message>
                   <import_file_name></import_file_name>
             </fas:makeFullFundTransfer>
      </soapenv:Body>
</soapenv:Envelope>
```

c) Response

iso_message	pain.002.001.06

Sample of the response for a successful request

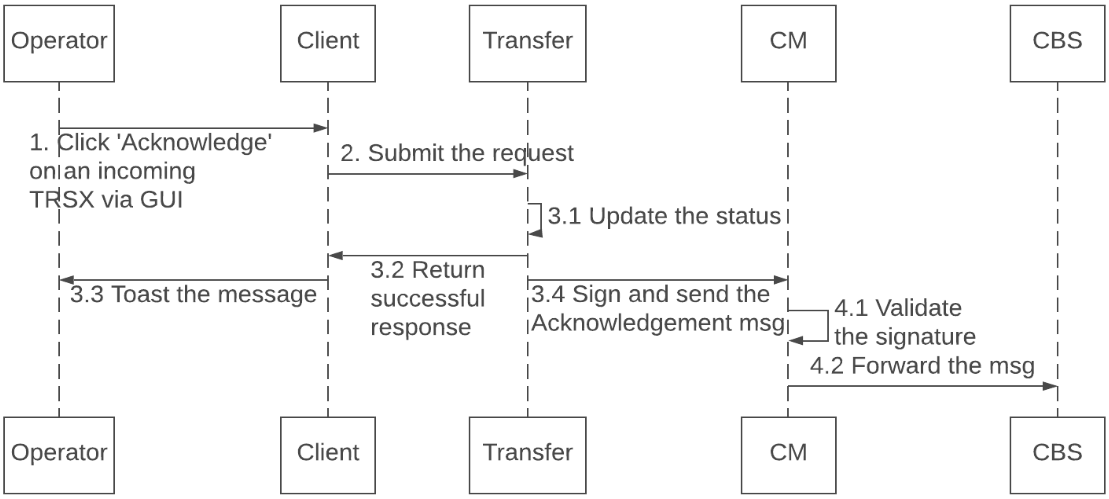
```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeFullFundTransferResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPPRINKHPPXX000000130</MsgId>
            <CreDtTm>2020-10-01T10:31:41.507</CreDtTm>
            <InitgPty>
                        <Nm>Bred Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>PRINKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>BREDKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTPRINKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-09-02T09:45:50.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>PRINKHPPXX/BREDKHPPXX/AAAA1916</OrgnlPmtInfId>
            <TxInfAndSts>
<OrgnlEndToEndId>000000000002-000000000001</OrgnlEndToEndId>
                    <TxSts>ACSP</TxSts>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Leng Alisa</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Try Chuthtima</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeFullFundTransferResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeFullFundTransferResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPPRINKHPPXX000000160</MsgId>
            <CreDtTm>2020-10-01T10:33:03.145</CreDtTm>
            <InitgPty>
                        <Nm>Bred Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>PRINKHPPXX</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>BREDKHPPXX</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTPRINKHPPXX000001912</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-09-02T09:45:50.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>PRINKHPPXX/BREDKHPPXX/AAAA1916</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>000000000002-000000000001</OrgnlEndToEndId>
                    <TxSts>RJCT</TxSts>
                    <StsRsnInf>
                        <Orgtr>
                            <id>
                                <OrgId>
                                    <AnyBIC>PRINKHPPXX<AnyBIC>
                                <OrgId>
                            </id>
                        </Orgtr>
                        <AddtlInf>REJECTED AT CORE DUE TO: [Receiver
AccountNumber[000000000001]'s name is Try Chuthtima but in iso message,
is Try Chuthtimaa]</AddtlInf>
                    </StsRsnInf>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">1000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Leng Alisa</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Try Chuthtimaa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
```

```
</Document>]]></return>
      </ns3:makeFullFundTransferResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

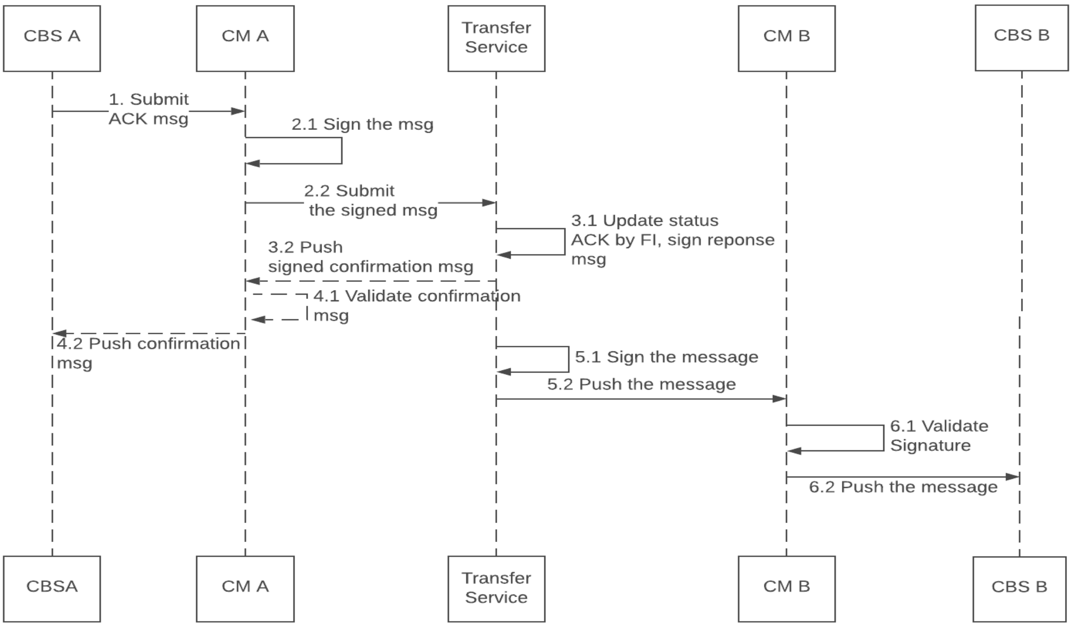
2.15 Acknowledgement (Manual via GUI)


Operator
Client
Transfer
CM
CBS
1. Click 'Acknowledge' on an incoming TRSX via GUI
2. Submit the request
3.1 Update the status
3.2 Return successful response
3.3 Toast the message
3.4 Sign and send the Acknowledgement msg
4.1 Validate the signature
4.2 Forward the msg
Operator
Client
Transfer
CM
CBS

1. An operator user of Bank A goes to ‘Incoming’ screen and clicks ‘Acknowledge’ on a transaction via GUI
2. The FAST client invokes the ACK API of Transfer Service
3. Transfer Service updates the status of the transaction to **Acknowledged by FI** and responds to the client, and a successful message is toasted on GUI for the operator. If the receiver bank using STP option, Transfer Service will construct, sign, and send Acknowledgement message *pain 002.001.06* to the CM of the receiver bank
4. CM of the receiver bank validates the message and sends it to the receiver bank CBS by calling **[makeAcknowledgment]** via SOAP

2.16 Acknowledgement (STP via API)

2.16.1 Message Flow



1. CBS of Bank A submits an acknowledge message *pain 002.001.06* by calling **[makeAcknowledgment]** via SOAP web service into Bank A CM
2. Bank A CM signs on the message using the private key of the FI and sends to FAST Transfer Service
3. Transfer Service validates the signature and updates the status of the transaction to **Acknowledged by FI**, signs and sends the response message to the Sender Bank CM
4. The sender bank CM validates the message and sends it to its CBS
5. Transfer Service signs this message pain 002.001.06 using the FAST private key and sends it to the receiver bank B CM
6. Bank B CM validates the signature on the message and sends it to CBS of Bank B by calling **[makeAcknowledgment]** via SOAP

2.16.2 Web Service

makeAcknowledgment (Receiver Participant to FAST)

a) Description

The receiver participant bank invokes this API to submit an acknowledgment message to confirm the transaction after crediting its customer.

b) Request

Parameter	Description
cm_user_name	Username of the CM which makes the acknowledgment
cm_password	Password of the CM which makes the acknowledgment
content_message	pain.002.001.06

Sample of the request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
<soapenv:Header/>
<soapenv:Body>
<fas:makeAcknowledgment>
  <cm_user_name>prince-adapter</cm_user_name>
  <cm_password>aA12345@</cm_password>
  <content_message><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
<CstmrPmtStsRpt>
<GrpHdr>
<MsgId>REPCMPLBBREPP000000250</MsgId>
<CreDtTm>2019-08-28T14:07:26.000+07:00</CreDtTm>
<InitgPty>
 <Nm>Prince Bank Cambodia Plc HQ-001</Nm>

</InitgPty>
<DbtrAgt>
 <FinInstnId>
   <BICFI>CMPLBBREPP</BICFI>
 </FinInstnId>
</DbtrAgt>
<CdtrAgt>
 <FinInstnId>
   <BICFI>CMPLBPRIPP</BICFI>
 </FinInstnId>
</CdtrAgt>
</GrpHdr>
<OrgnlGrpInfAndSts>
<OrgnlMsgId>CRTPRINKHPPXXXX000072408</OrgnlMsgId>
<OrgnlMsgNmId>pain.001.001.05</OrgnlMsgNmId>
<OrgnlCreDtTm>2019-08-28T14:07:26.000+07:00</OrgnlCreDtTm>
```

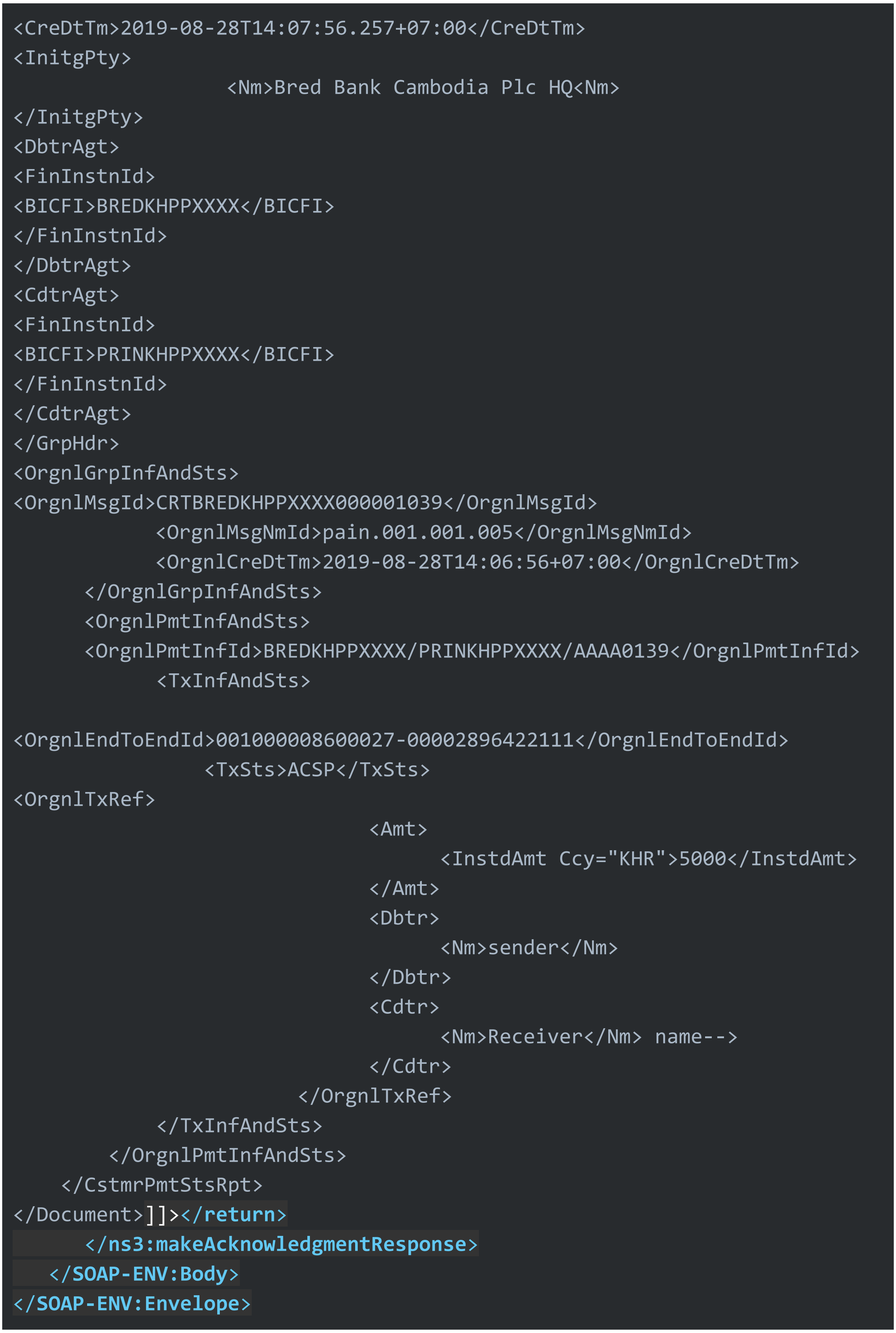
```xml
</OrgnlGrpInfAndSts>
<OrgnlPmtInfAndSts>
<OrgnlPmtInfId>BREDKHPPXX/PRINKHPPXX/CBRP3382</OrgnlPmtInfId>
<TxInfAndSts>
 <OrgnlEndToEndId>000030795-001000008300012</OrgnlEndToEndId>
 <TxSts>ACSC</TxSts>
 <OrgnlTxRef>
   <Amt>
     <InstdAmt Ccy="USD">100</InstdAmt>
   </Amt>
   <Dbtr>
     <Nm>Try Chuthtima</Nm>

   </Dbtr>
   <Cdtr>
     <Nm>Leng Alisa</Nm>

   </Cdtr>
 </OrgnlTxRef>
</TxInfAndSts>
</OrgnlPmtInfAndSts>
</CstmrPmtStsRpt>
</Document>]]></content_message>
</fas:makeAcknowledgment>
</soapenv:Body>
</soapenv:Envelope>
```

c) Response

return	pain.002.001.06

Sample of the response for a successful request

```xml
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeAcknowledgmentResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
<GrpHdr>
<MsgId>REPBREDKHPPXXXX000073405</MsgId>
```



```
<CreDtTm>2019-08-28T14:07:56.257+07:00</CreDtTm>
<InitgPty>
                  <Nm>Bred Bank Cambodia Plc HQ<Nm>
</InitgPty>
<DbtrAgt>
<FinInstnId>
<BICFI>BREDKHPPXXXX</BICFI>
</FinInstnId>
</DbtrAgt>
<CdtrAgt>
<FinInstnId>
<BICFI>PRINKHPPXXXX</BICFI>
</FinInstnId>
</CdtrAgt>
</GrpHdr>
<OrgnlGrpInfAndSts>
<OrgnlMsgId>CRTBREDKHPPXXXX000001039</OrgnlMsgId>
            <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>
            <OrgnlCreDtTm>2019-08-28T14:06:56+07:00</OrgnlCreDtTm>
      </OrgnlGrpInfAndSts>
      <OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>BREDKHPPXXXX/PRINKHPPXXXX/AAAA0139</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                <TxSts>ACSP</TxSts>
<OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="KHR">5000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>sender</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Receiver</Nm> name-->
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeAcknowledgmentResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeAcknowledgmentResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPCMPLBPRIPP000000270</MsgId>
            <CreDtTm>2020-09-12T14:13:48.001</CreDtTm>
            <InitgPty>
                        <Nm>Prince Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>CMPLBBREPP</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>CMPLBPRIPP</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTPRINKHPPXXXX000072408</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-08-28T14:07:26.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>CMPLBBREPP/CMPLBPRIPP/CBRP3382</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>000030795-001000008300012</OrgnlEndToEndId>
                    <TxSts>RJCT</TxSts>
                    <StsRsnInf>
                        <Orgtr>
                            <id>
                                <OrgId>
```

```
                                    <AnyBIC>CMPLBBREPP<AnyBIC>
                                <OrgId>
                            </id>
                        </Orgtr>

<AddtlInf>[OriginalPaymentInfoID[CMPLBBREPP/CMPLBPRIPP/CBRP3382] is not
found.]</AddtlInf>
                    </StsRsnInf>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">100</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>Try Chuthtima</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeAcknowledgmentResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

makeAcknowledgment (FAST to Sender Participant)

a) Description

The FAST system invokes this API to submit the acknowledgment message to the sender bank CBS. In such that the message is signed and sent by the FAST Transfer Service to the sender bank CBS through the sender Client Module (CM).

b) Request

Parameter	**Description**
cm_user_name	Username of the FAST in the receiver participant bank CBS
cm_password	Password of the FAST in the receiver participant bank CBS
iso_message	pain.002.001.06
import_file_name	Name of the xml file which stores batch transaction

Sample of the request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
      <soapenv:Header/>
            <soapenv:Body>
                   <fas:makeAcknowledgment>
                         <cm_user_name>prince</cm_user_name>
                         <cm_password>aA12345@</cm_password>
                         <iso_message><![CDATA[
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
<GrpHdr>
<MsgId>REPBREDKHPPXXXX000073405</MsgId>
<CreDtTm>2019-08-28T14:07:56.257+07:00</CreDtTm>
<InitgPty>
                   <Nm>Bred Bank Cambodia Plc HQ<Nm>
</InitgPty>
<DbtrAgt>
<FinInstnId>
<BICFI>BREDKHPPXXXX</BICFI>
</FinInstnId>
</DbtrAgt>
<CdtrAgt>
<FinInstnId>
<BICFI>PRINKHPPXXXX</BICFI>
</FinInstnId>
</CdtrAgt>
</GrpHdr>
<OrgnlGrpInfAndSts>
<OrgnlMsgId>CRTBREDKHPPXXXX000001039</OrgnlMsgId>
            <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>
            <OrgnlCreDtTm>2019-08-28T14:06:56+07:00</OrgnlCreDtTm>
      </OrgnlGrpInfAndSts>
      <OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>BREDKHPPXXXX/PRINKHPPXXXX/AAAA0139</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                <TxSts>ACSC</TxSts>
<OrgnlTxRef>
                               <Amt>
                                     <InstdAmt Ccy="KHR">5000</InstdAmt>
                               </Amt>
                               <Dbtr>
                                     <Nm>sender</Nm>
                               </Dbtr>
```

```
                                <Cdtr>
                                      <Nm>Receiver</Nm> name-->
                                </Cdtr>
                         </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>
]]></iso_message>
                   <import_file_name></import_file_name>
             </fas:makeAcknowledgment>
      </soapenv:Body>
</soapenv:Envelope>
```

c) Response

iso_message	pain.002.001.06

Sample of the response for a successful request

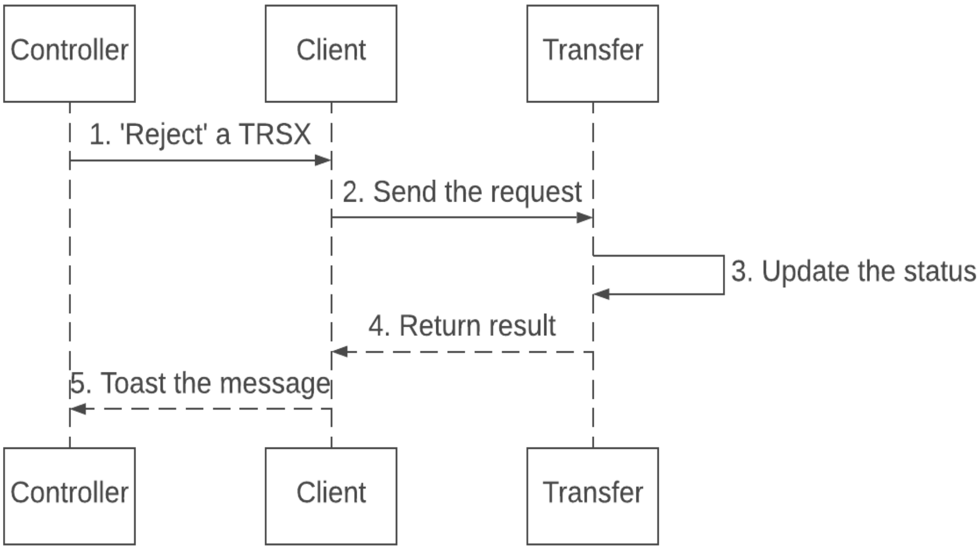
```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeAcknowledgmentResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
<GrpHdr>
<MsgId>REPBREDKHPPXXXX000073405</MsgId>
<CreDtTm>2019-08-28T14:07:56.257+07:00</CreDtTm>
<InitgPty>
                  <Nm>Bred Bank Cambodia Plc HQ<Nm>
</InitgPty>
<DbtrAgt>
<FinInstnId>
<BICFI>BREDKHPPXXXX</BICFI>
</FinInstnId>
</DbtrAgt>
<CdtrAgt>
<FinInstnId>
<BICFI>PRINKHPPXXXX</BICFI>
</FinInstnId>
</CdtrAgt>
</GrpHdr>
<OrgnlGrpInfAndSts>
<OrgnlMsgId>CRTBREDKHPPXXXX000001039</OrgnlMsgId>
            <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>
            <OrgnlCreDtTm>2019-08-28T14:06:56+07:00</OrgnlCreDtTm>
      </OrgnlGrpInfAndSts>
      <OrgnlPmtInfAndSts>
      <OrgnlPmtInfId>BREDKHPPXXXX/PRINKHPPXXXX/AAAA0139</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>001000008600027-00002896422111</OrgnlEndToEndId>
                <TxSts>ACSP</TxSts>
<OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="KHR">5000</InstdAmt>
                              </Amt>
                              <Dbtr>
                                    <Nm>sender</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Receiver</Nm> name-->
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeAcknowledgmentResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:makeAcknowledgmentResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.06">
    <CstmrPmtStsRpt>
        <GrpHdr>
            <MsgId>REPCMPLBPRIPP000000270</MsgId>
            <CreDtTm>2020-09-12T14:13:48.001</CreDtTm>
            <InitgPty>
                        <Nm>Prince Bank Cambodia Plc HQ-001<Nm>
                  </InitgPty>
                  <DbtrAgt>
                        <FinInstnId>
                              <BICFI>CMPLBBREPP</BICFI>
                        </FinInstnId>
                  </DbtrAgt>
                  <CdtrAgt>
                <FinInstnId>
                    <BICFI>CMPLBPRIPP</BICFI>
                </FinInstnId>
            </CdtrAgt>
        </GrpHdr>
        <OrgnlGrpInfAndSts>
                  <OrgnlMsgId>CRTPRINKHPPXXXX000072408</OrgnlMsgId>
                  <OrgnlMsgNmId>pain.001.001.005</OrgnlMsgNmId>

<OrgnlCreDtTm>2019-08-28T14:07:26.000+07:00</OrgnlCreDtTm>
            </OrgnlGrpInfAndSts>
            <OrgnlPmtInfAndSts>

<OrgnlPmtInfId>CMPLBBREPP/CMPLBPRIPP/CBRP3382</OrgnlPmtInfId>
            <TxInfAndSts>

<OrgnlEndToEndId>000030795-001000008300012</OrgnlEndToEndId>
                    <TxSts>RJCT</TxSts>
                    <StsRsnInf>
                        <Orgtr>
                            <id>
                                <OrgId>
                                    <AnyBIC>CMPLBBREPP<AnyBIC>
                                <OrgId>
                            </id>
                        </Orgtr>

<AddtlInf>[OriginalPaymentInfoID[CMPLBBREPP/CMPLBPRIPP/CBRP3382] is not
found.]</AddtlInf>
                    </StsRsnInf>
                        <OrgnlTxRef>
                              <Amt>
                                    <InstdAmt Ccy="USD">100</InstdAmt>
                              </Amt>
                              <Dbtr>
```

```
                                    <Nm>Try Chuthtima</Nm>
                              </Dbtr>
                              <Cdtr>
                                    <Nm>Leng Alisa</Nm>
                              </Cdtr>
                        </OrgnlTxRef>
            </TxInfAndSts>
        </OrgnlPmtInfAndSts>
    </CstmrPmtStsRpt>
</Document>]]></return>
      </ns3:makeAcknowledgmentResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

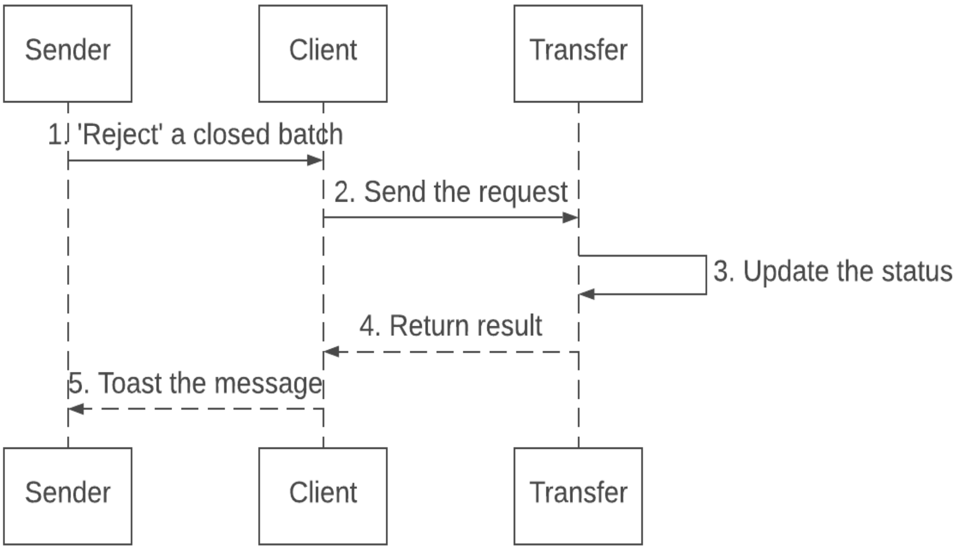
2.17 Controller Rejects Pending Transactions



1. A controller goes to the ‘Pending Transactions’ screen, clicks the ‘Reject’ button on a pending transaction. The controller has to input the reason for the rejection and confirm
2. The FAST client invokes Reject API provided by FAST Transfer Service
3. The FAST Transfer Service updates the status to **Rejected by Controller**
4. The FAST Transfer Service returns the result to the client
5. A successful message is toasted on GUI for the controller

Note: An operator can see the rejected transaction in ‘Outgoing’ screen

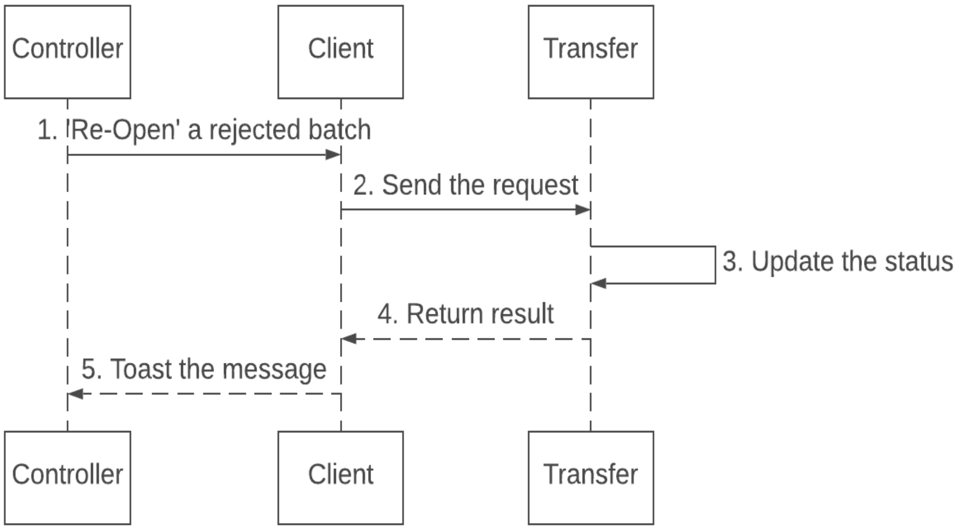
2.18 Sender Rejects Signed and Closed Batch



1. A sender goes to the 'Closed Batches' screen, clicks the 'Reject' button on a closed batch. The sender has to input the reason for the rejection
2. The FAST client invokes the Reject API provided by FAST Transfer Service
3. The FAST Transfer Service updates the status to **Rejected by Sender**
4. The FAST Transfer Service returns the result to the client
5. A successful message is toasted on GUI for the sender

   Note: The controller can see the rejected batch in 'Rejected Batches' screen

2.19 Controller Re-opens Rejected by Sender Batch(es)

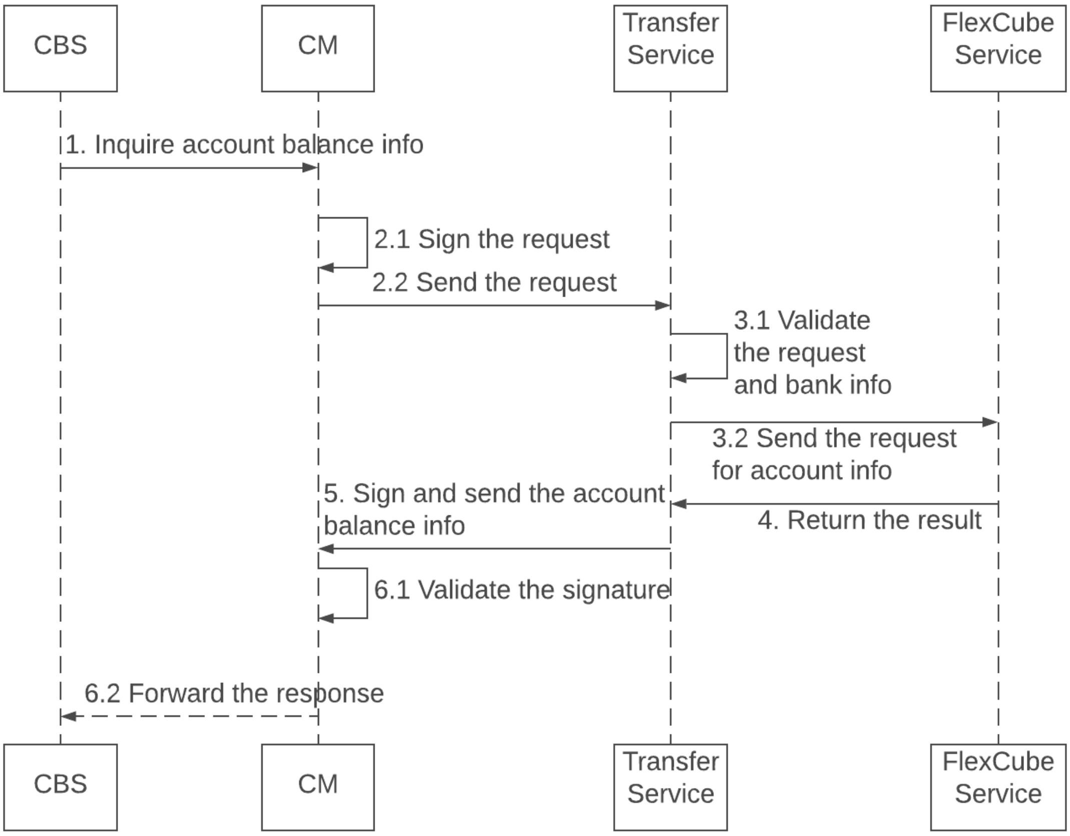


1. A controller goes to ‘Rejected Batches’ screen, clicks ‘Re-open’ button on a rejected batch
2. The FAST client invokes the Reopen API provided by FAST Transfer Service
3. FAST Transfer Service ungroups the transactions in the batch and updates the status to **Pending**
4. Transfer Service returns the result to the client
5. A successful message is toasted on GUI for the controller

    Note: An operator can see those pending transactions in their ‘Outgoing’ screen. And, the controller can either reject or close again by selecting the transactions on ‘Pending Transaction’ screen

2.20 Participant Bank Account Balance Inquiry

2.20.1 Message Flow


CBS
CM
Transfer Service
FlexCube Service
1. Inquire account balance info
2.1 Sign the request
2.2 Send the request
3.1 Validate the request and bank info
3.2 Send the request for account info
5. Sign and send the account balance info
4. Return the result
6.1 Validate the signature
6.2 Forward the response
CBS
CM
Transfer Service
FlexCube Service

1. The sender bank CBS inquires its account balance info by calling **[getAccountInquiry]** via SOAP web service
2. The sender bank CM signs the request and sends it to the Transfer Service
3. Transfer Service validates the request and the bank info. If it is valid, the Transfer Service requests the FlexCube Service for the account information of the requested bank
4. The FlexCube Service returns the result to the Transfer Service
5. Transfer Service signs and sends the response to the sender bank CM
6. The sender bank CM validates the signature and sends it to the sender bank CBS

2.20.2 Web Service

a) Description

The participant bank invokes this API to get the bank account's information, including the settlement and current accounts' available balance, net position, and the next settlement session.

b) Request

Parameter	Description
cm_user_name	Username of the CM which makes the account inquiry
cm_password	Password of the CM which makes the account inquiry
participant_code	The bank code which invokes account inquiry

Sample of the request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:fas="http://fastwebservice.nbc.org.kh/">
   <soapenv:Header/>
   <soapenv:Body>
      <fas:getAccountInquiry>
         <!--Optional:-->
         <cm_user_name>prince-adapter</cm_user_name>
         <!--Optional:-->
         <cm_password>aA12345@</cm_password>
         <!--Optional:-->
         <participant_code>CMPLBPRIPP</participant_code>
      </fas:getAccountInquiry>
   </soapenv:Body>
</soapenv:Envelope>
```

c) Response

message	Custom

Sample of the response for a successful request

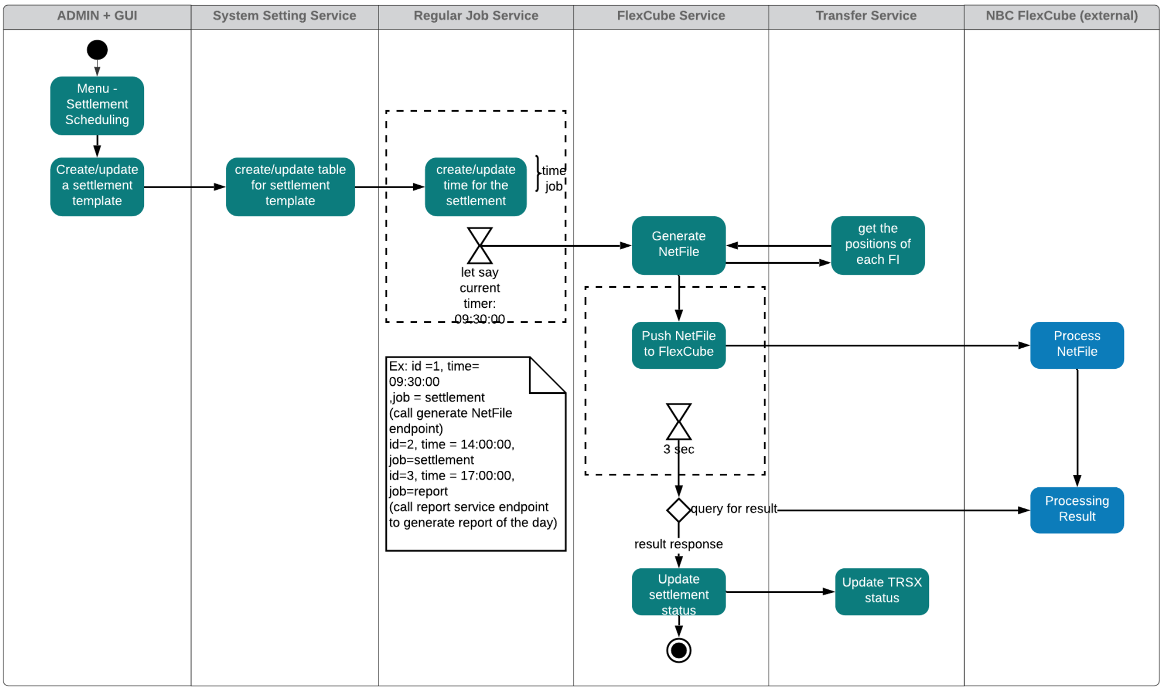
```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns3:getAccountInquiryResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return><![CDATA[<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
                  <Doc>
                        <BizDate>12 September 2020</BizDate>
                        <BizTime>16:05:28</BizTime>
                        <myBankID>101</myBankID>
                        <nextSess>2020-09-15T09:45:00</nextSess>

<nextSettleUSD>1200</nextSettleUSD>
<nextSettleKHR>400</nextSettleKHR>
<nextSettleTHB>0</nextSettleTHB>

<settleUSD>110000000</settleUSD>
<settleKHR>1100000000</settleKHR>
<settleTHB>0</settleTHB>

<curAccUSD>
                              <Brnch>
                                    <Nb>001</Nb>
                                    <Acc>001-18</Acc>
                                    <Bal>101000000</Bal>
                              </Brnch>
                        </curAccUSD>
                        <totalUSD>101000000</totalUSD>
<curAccKHR>
                              <Brnch>
                                    <Nb>001</Nb>
                                    <Acc>001-19</Acc>
                                    <Bal>1010000000</Bal>
                              </Brnch>
                        </curAccKHR>
                        <totalKHR>1010000000</totalKHR>
<curAccTHB>
                              <Brnch>
                                    <Nb>001</Nb>
                                    <Acc>001-20</Acc>
                                    <Bal>1010000000</Bal>
                              </Brnch>
                        </curAccTHB>
                        <totalTHB>1010000000</totalTHB>
                  </Doc>]]></return>
      </ns3:getAccountInquiryResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample of the response for a failure request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
```

```
      <ns3:getAccountInquiryResponse
xmlns:ns3="http://fastwebservice.nbc.org.kh/">
         <return>[Can not find SenderBankId by
SenderBankSwiftCode[CMPLBIRIPP]., BankSwiftCode[CMPLBIRIPP] doesn't
belong to your bank. ]</return>
      </ns3:getAccountInquiryResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.21 Regular Job Service and Settlement


ADMIN + GUI
System Setting Service
Regular Job Service
FlexCube Service
Transfer Service
NBC FlexCube (external)
Menu - Settlement Scheduling
Create/update a settlement template
create/update table for settlement template
create/update time for the settlement
time job
let say current timer: 09:30:00
Ex: id =1, time= 09:30:00 ,job = settlement (call generate NetFile endpoint) id=2, time = 14:00:00, job=settlement id=3, time = 17:00:00, job=report (call report service endpoint to generate report of the day)
Generate NetFile
get the positions of each FI
Push NetFile to FlexCube
Process NetFile
3 sec
query for result
Processing Result
result response
Update settlement status
Update TRSX status

Regular job service is mainly implemented for periodic task scheduling, such as report and settlement. The above diagram illustrates the procedure of settlement:

1. NBC Admin goes to ‘Settlement Scheduling’ to create or update a settlement template
2. System Setting Service updates the record and calls Regular Job Service to create or update the corresponding record in its database
3. Once the settlement time comes, Regular Job Service will call FlexCube Service to generate the NetFile for the session
4. FlexCube Service retrieves the net position from Transfer Service and submits to the NBC FlexCube System (External)
5. FlexCube Service fetches the result from the NBC FlexCube System at a pre-defined time interval (3 sec)
6. FlexCube Service updates the settlement status when the result is retrieved successfully

7. FlexCube Service calls Transfer Service to updates the status of the transactions in this settlement session to **Settled**

1.