Foreign Exchange - Currency Conversion

The Mojaloop currency conversion functionality enables foreign exchange (FX) transactions, supporting multiple approaches for currency conversion within the ecosystem. Currently, the system implements **Payer DFSP currency conversion**, where the Payer DFSP (Digital Financial Services Provider) coordinates with a foreign exchange provider (FXP) to obtain liquidity in another currency to facilitate a transfer.

Future enhancements to the currency conversion design include:

1. **Payee DFSP conversion**
   The Payee DFSP arranges for foreign exchange conversion.
2. **Reference currency conversion**
   Both the Payer and Payee DFSPs engage with FXPs to convert funds via a reference currency.
3. **Bulk conversion**
   DFSPs can procure currency liquidity from an FXP in bulk.

Role of the Foreign Exchange Provider (FXP)

A core feature of Mojaloop's currency conversion capability is its support for a competitive FX marketplace, where multiple FXPs can provide real-time exchange rate quotes. This design fosters an open and dynamic environment for foreign exchange transactions.

The currency conversion process follows a three-step workflow:

1. **Quote Request**
   The Payer DFSP requests a quote from an FXP. For example, a Zambian DFSP can obtain a conversion quote for a specific transfer.
2. **Quote Agreement**
   The Payer DFSP reviews the exchange rate and terms provided by the FXP. Once accepted, the FXP locks in the rate.
3. **Transfer Finalization**
   Upon notification from the Mojaloop scheme that the dependent transfer has been completed, the conversion process is finalized.

This streamlined approach ensures transparency and competitiveness in FX transactions, benefiting both DFSPs and end users.

Impact of Amount Type on Currency Conversion

The Payer DFSP conversion implementation supports two distinct scenarios based on the amount type specified in the transaction:

1. **Sending funds in the source (local) currency**
2. **Making a payment in target (a foreign) currency**

Sending Funds to an Account in Another Currency

In this use case, the **Payer DFSP** initiates a transfer using the amount type **SEND**, specifying the transfer amount in the payer's local currency (source currency). This method is commonly used for **P2P remittance**

transfers, where the sender transfers funds in their local currency, and the recipient receives the equivalent amount in their respective currency after conversion.

Currency Conversion Transfer (source currency)

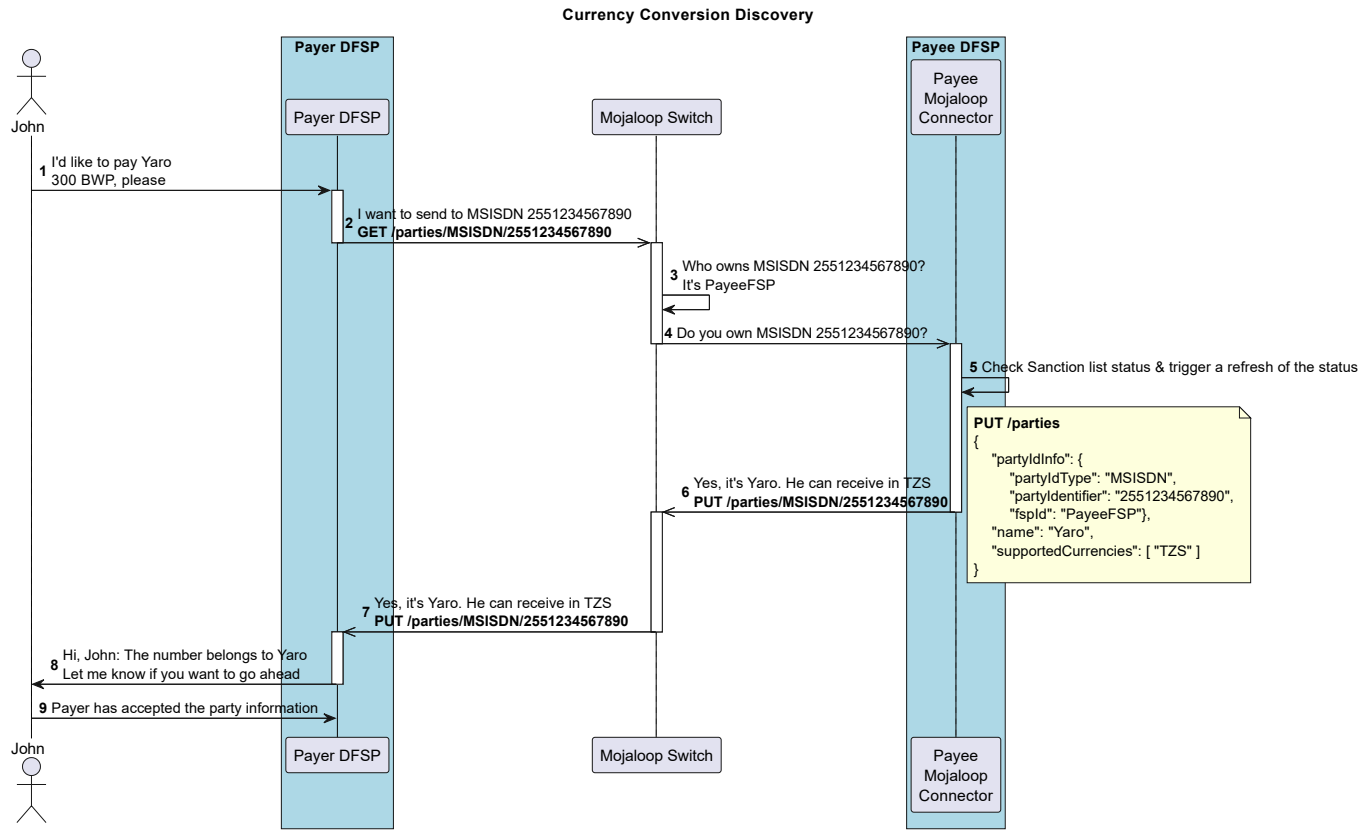
Below is a simplified sequence diagram showing the flows between the Participant organizations, the foreign exchange providers and the Mojaloop switch for a currency conversion transfer specified in source currency.

The flow is divided up into:

1. Discovery Phase
2. Agreement Phase - Currency Conversion
3. Agreement Phase
4. Payer DFSP presents terms to Payer
5. Transfer Phase

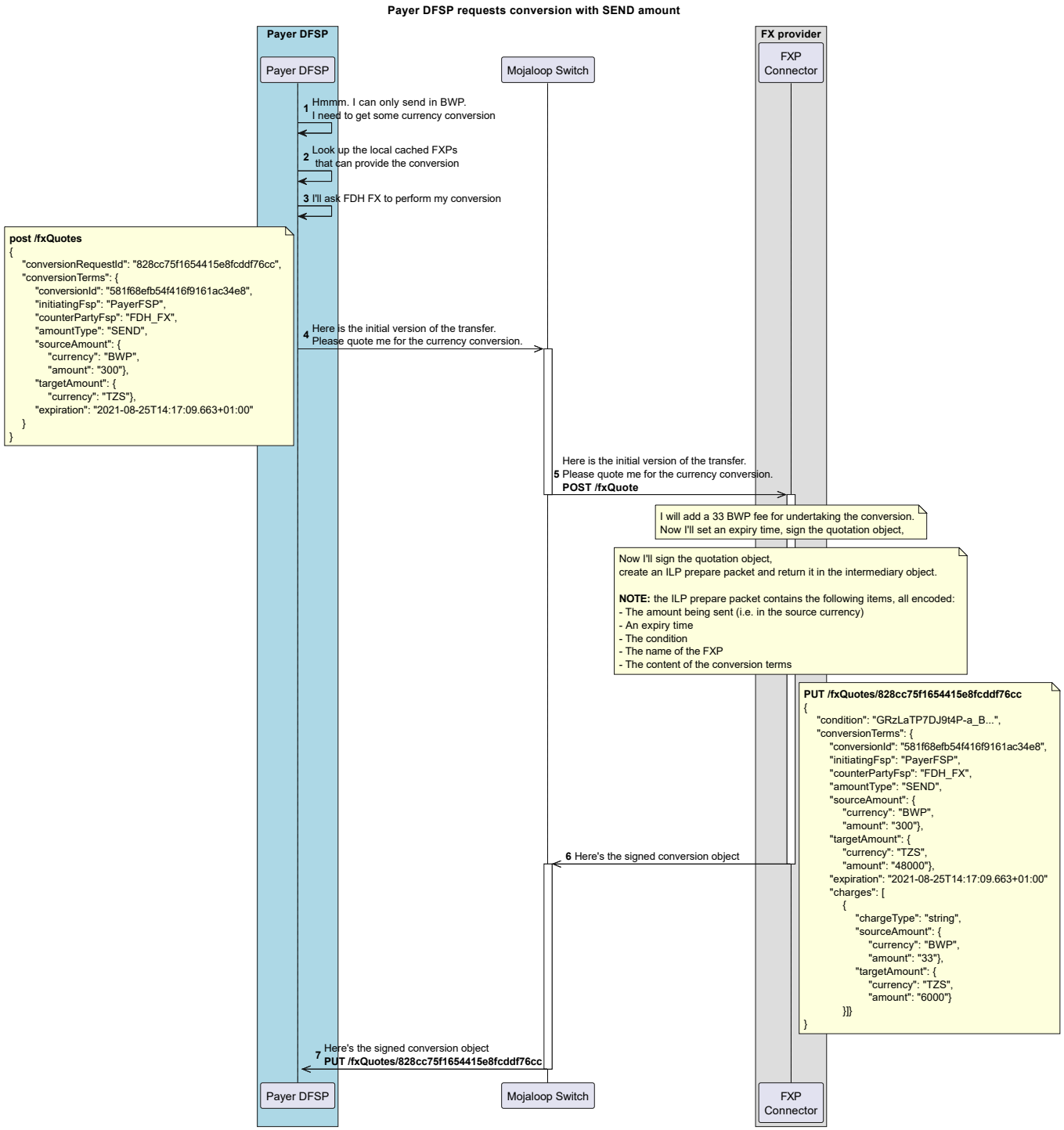
Discovery Phase

The Payer DFSP identifies the Payee DFSP organization and confirms the account validity and currency.



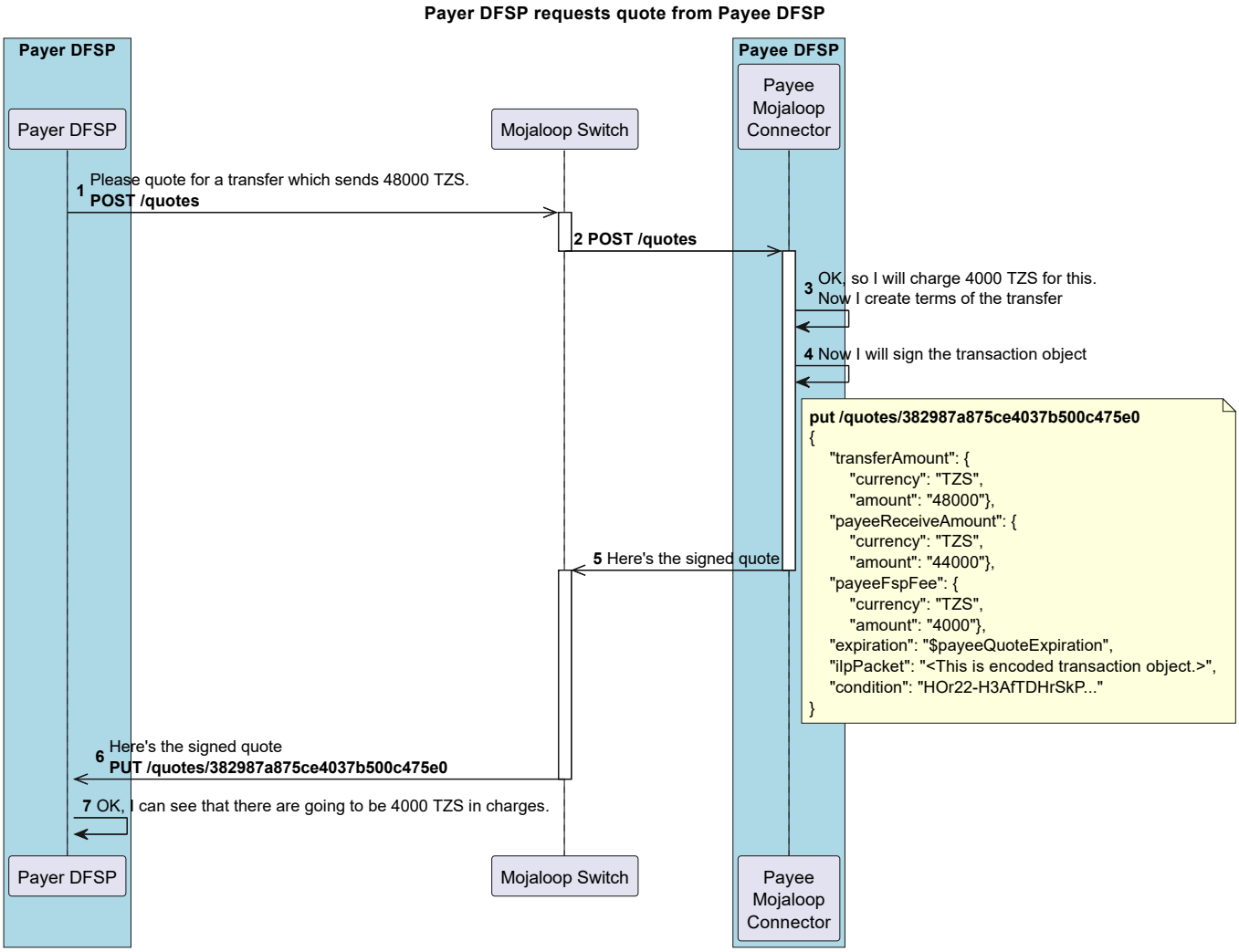
Agreement Phase - Currency Conversion

The Payer DFSP makes a request to the FXP for liquidity cover for transfer. The currency conversion terms are returned.



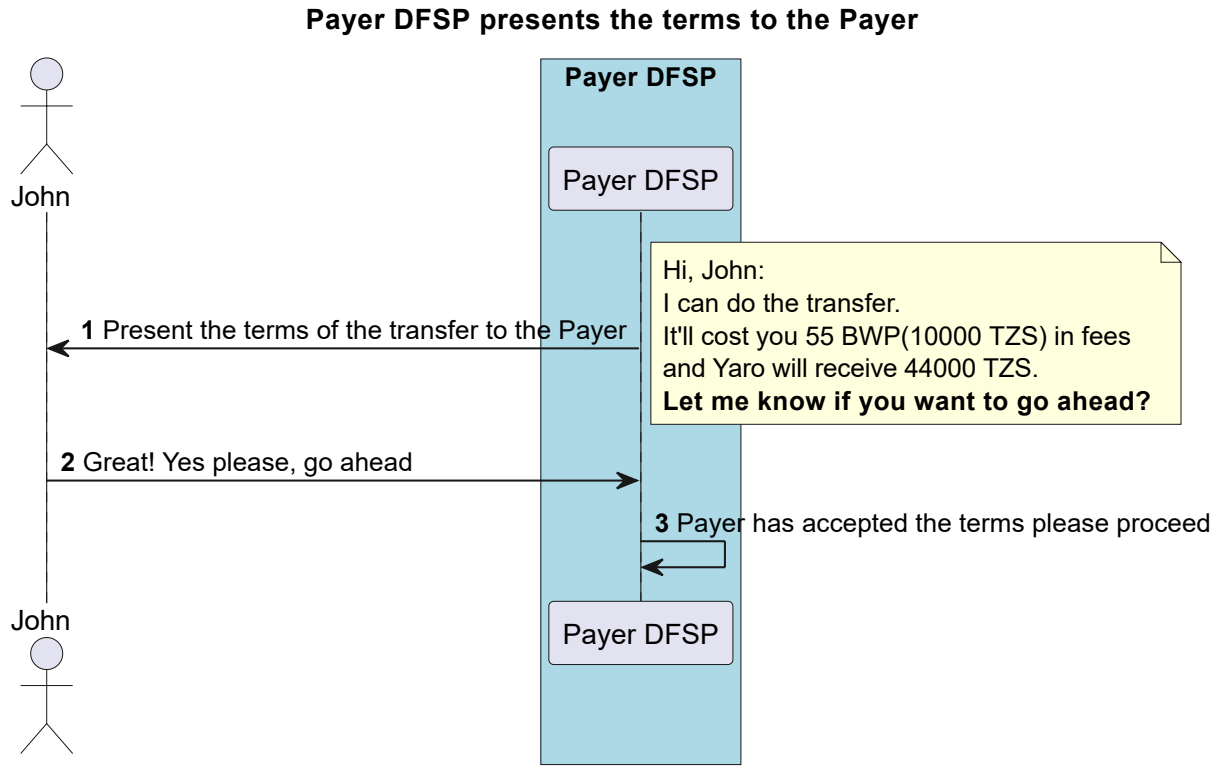
Agreement Phase

The Payer DFSP makes a request to the Payee DFSP for the transfer terms.



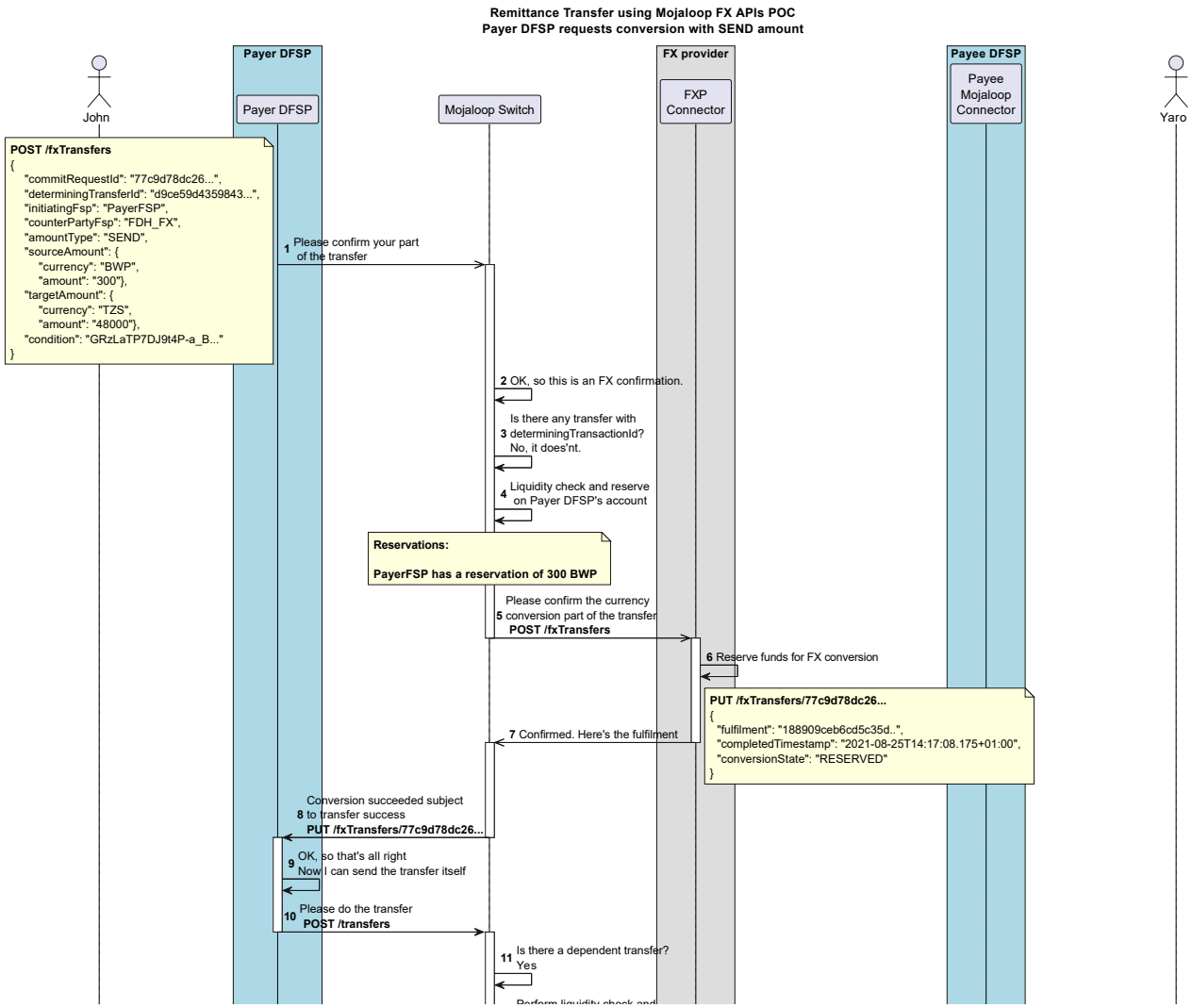
Payer DFSP presents terms to Payer

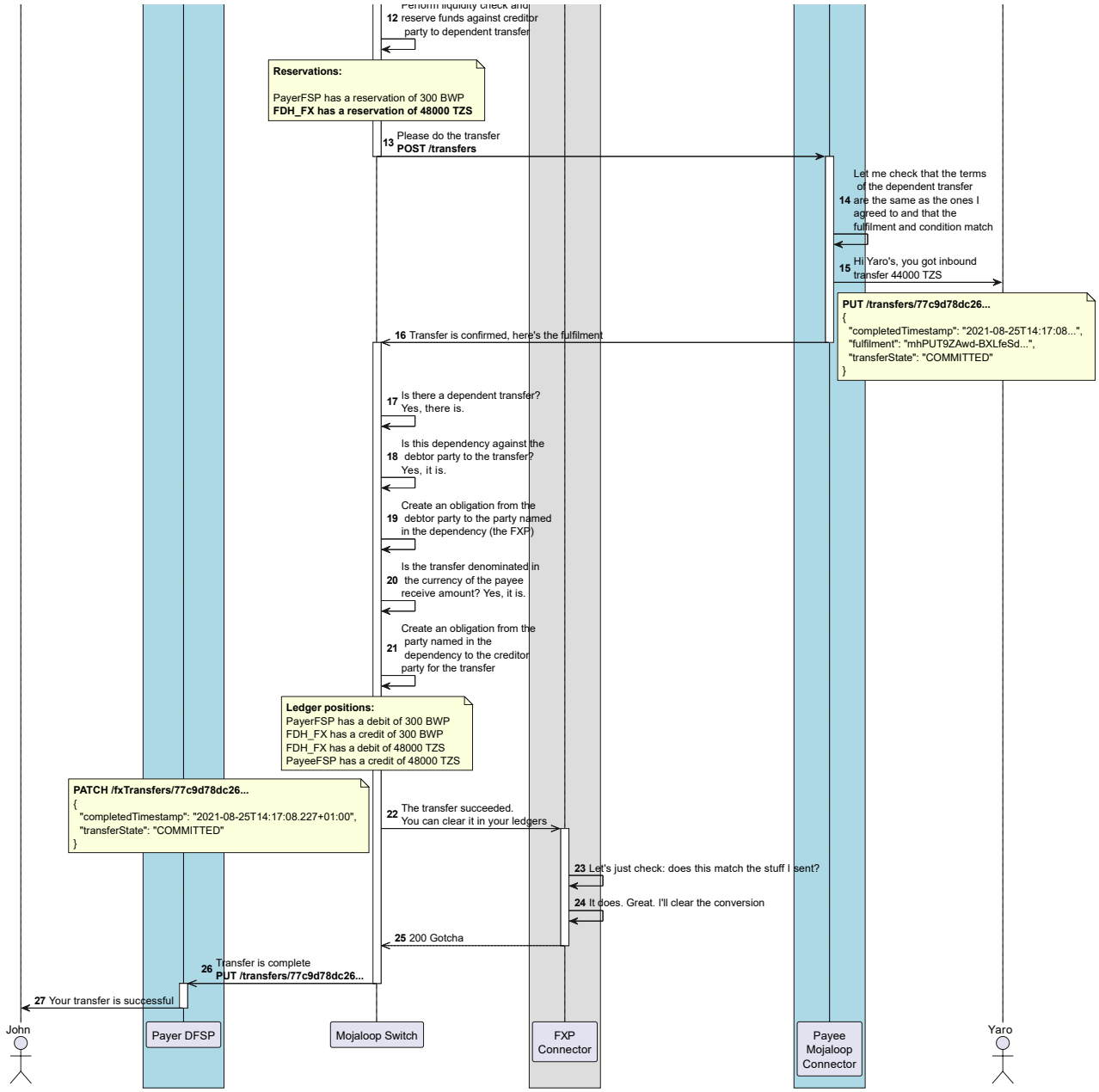
At this point the party information, the conversion terms, and the transfer terms have been provided to the Payer DFSP. The Payer DFSP presents these terms to the Payer and asks wether to proceed or not.



Transfer Phase

Now that the terms of the transfer have been agreed to, the transfer can proceed. Both the Conversion and the transfer terms are comitted together.





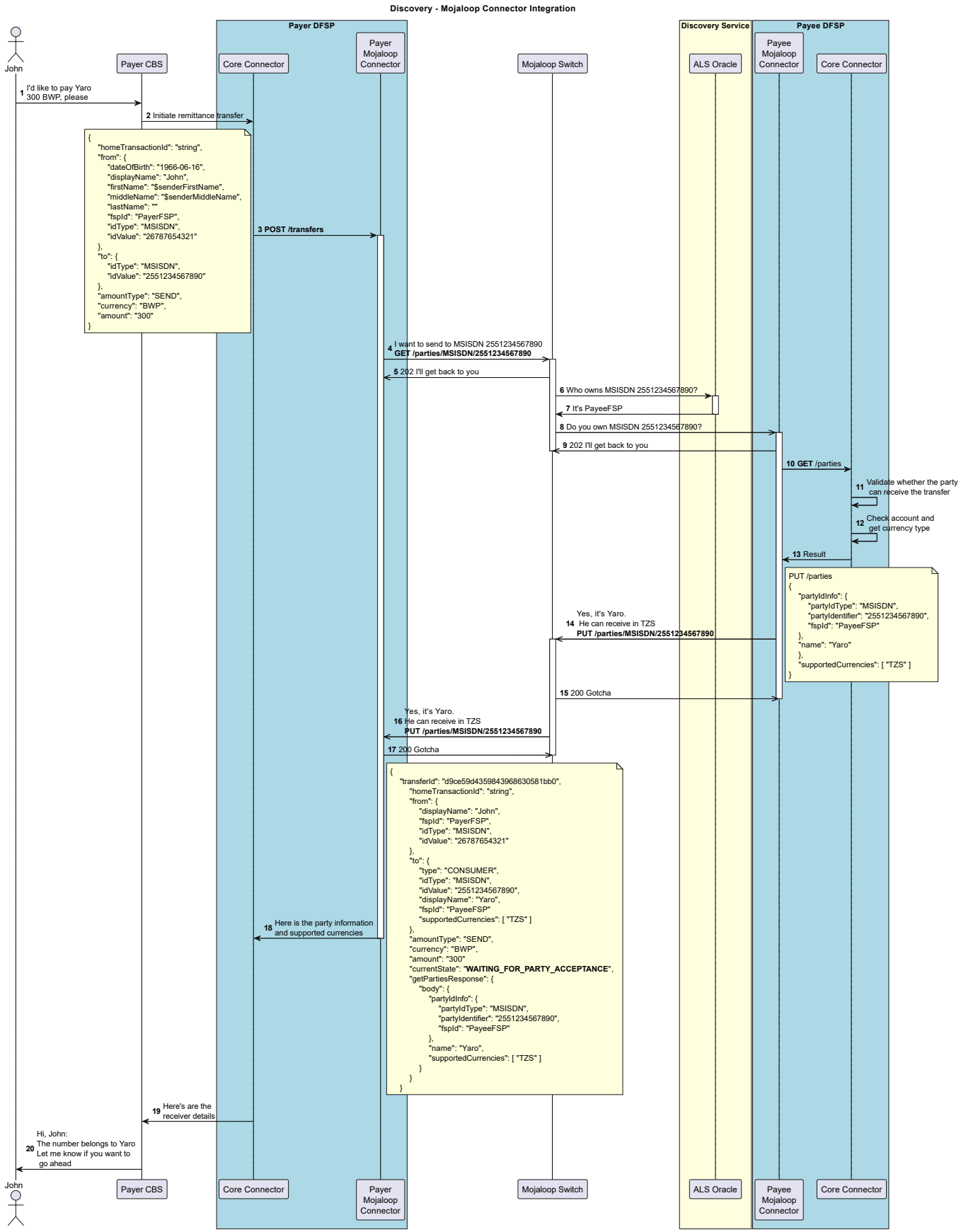
Mojaloop Connector Integration for Currency Conversion

Below is a detailed sequence diagram that shows the complete flow, and includes the **Mojaloop Connector** and integration APIs for all participant organizations. (This is a useful view if you are building integrations as a participant organization.)

Discovery phase - Mojaloop Connector

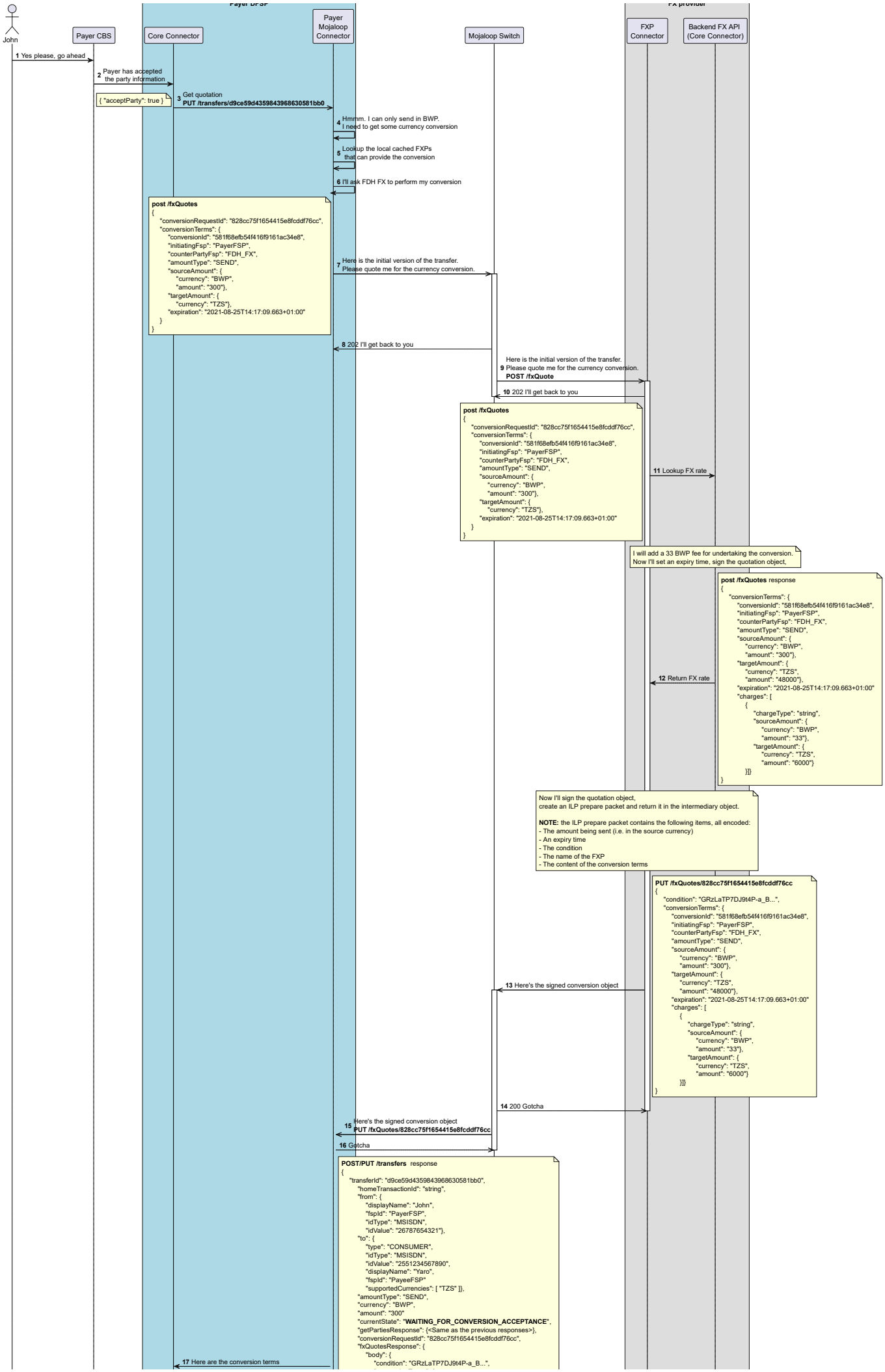
Mojaloop make use of an Oracle to identify the DFSP organization associated with the Party identifier. The Payee DFSP must respond to the GET /parties to confirm that the account exists and is active for that Party

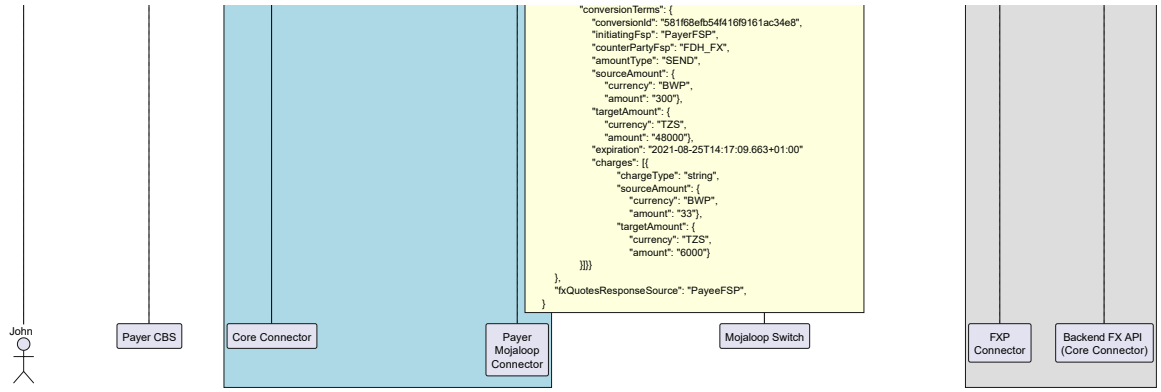
identifier. The supported currencies for that account are returned.



Agreement Phase Currency Conversion - Mojaloop Connector

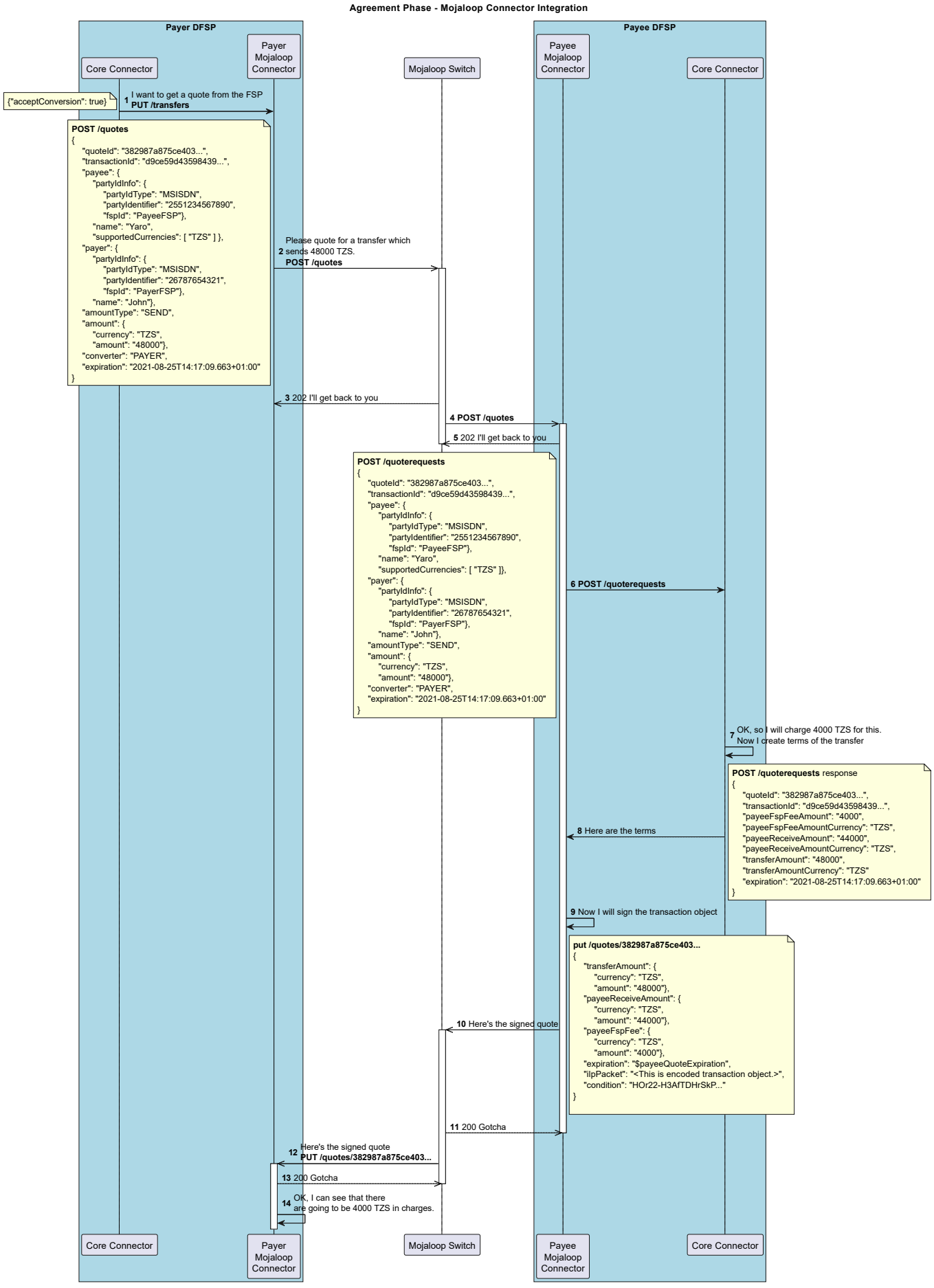
The Payer DFSP does not transact in any of the Payee DFPSs supported currencies. This triggers the requirement for currency conversion inside the Mojaloop Connector. The Payer DFSP uses it's local cache of FXPs to select and makes a request to the Foreign Exchange provider for liquidity cover and a conversion rate.


John
Payer CBS
Payer DFSP
Core Connector
Payer Mojaloop Connector
Mojaloop Switch
FX provider
FXP Connector
Backend FX API (Core Connector)
1 Yes please, go ahead
2 Payer has accepted the party information
{ "acceptParty": true }
3 Get quotation
PUT /transfers/d9ce59d43598439686305 81bb0
4 Hmmm. I can only send in BWP. I need to get some currency conversion
5 Lookup the local cached FXPs that can provide the conversion
6 I'll ask FDH FX to perform my conversion
post /fxQuotes
{
"conversionRequestId": "828cc75f1654415e8fcddf76cc",
"conversionTerms": {
"conversionId": "581f68efb54f416f9161ac34e8",
"initiatingFsp": "PayerFSP",
"counterPartyFsp": "FDH_FX",
"amountType": "SEND",
"sourceAmount": {
"currency": "BWP",
"amount": "300"},
"targetAmount": {
"currency": "TZS"},
"expiration": "2021-08-25T14:17:09.663+01:00"
}
}
7 Here is the initial version of the transfer. Please quote me for the currency conversion.
8 202 I'll get back to you
Here is the initial version of the transfer.
9 Please quote me for the currency conversion.
POST /fxQuote
10 202 I'll get back to you
post /fxQuotes
{
"conversionRequestId": "828cc75f1654415e8fcddf76cc",
"conversionTerms": {
"conversionId": "581f68efb54f416f9161ac34e8",
"initiatingFsp": "PayerFSP",
"counterPartyFsp": "FDH_FX",
"amountType": "SEND",
"sourceAmount": {
"currency": "BWP",
"amount": "300"},
"targetAmount": {
"currency": "TZS"},
"expiration": "2021-08-25T14:17:09.663+01:00"
}
}
11 Lookup FX rate
I will add a 33 BWP fee for undertaking the conversion.
Now I'll set an expiry time, sign the quotation object,
post /fxQuotes response
{
"conversionTerms": {
"conversionId": "581f68efb54f416f9161ac34e8",
"initiatingFsp": "PayerFSP",
"counterPartyFsp": "FDH_FX",
"amountType": "SEND",
"sourceAmount": {
"currency": "BWP",
"amount": "300"},
"targetAmount": {
"currency": "TZS",
"amount": "48000"},
"expiration": "2021-08-25T14:17:09.663+01:00"
"charges": [
{
"chargeType": "string",
"sourceAmount": {
"currency": "BWP",
"amount": "33"},
"targetAmount": {
"currency": "TZS",
"amount": "6000"}
}]}
}
12 Return FX rate
Now I'll sign the quotation object,
create an ILP prepare packet and return it in the intermediary object.
NOTE: the ILP prepare packet contains the following items, all encoded:
- The amount being sent (i.e. in the source currency)
- An expiry time
- The condition
- The name of the FXP
- The content of the conversion terms
PUT /fxQuotes/828cc75f1654415e8fcddf76cc
{
"condition": "GRzLaTP7DJ9t4P-a_B...",
"conversionTerms": {
"conversionId": "581f68efb54f416f9161ac34e8",
"initiatingFsp": "PayerFSP",
"counterPartyFsp": "FDH_FX",
"amountType": "SEND",
"sourceAmount": {
"currency": "BWP",
"amount": "300"},
"targetAmount": {
"currency": "TZS",
"amount": "48000"},
"expiration": "2021-08-25T14:17:09.663+01:00"
"charges": [
{
"chargeType": "string",
"sourceAmount": {
"currency": "BWP",
"amount": "33"},
"targetAmount": {
"currency": "TZS",
"amount": "6000"}
}]}
}
13 Here's the signed conversion object
14 200 Gotcha
15 Here's the signed conversion object
PUT /fxQuotes/828cc75f1654415e8fcddf76cc
16 Gotcha
POST/PUT /transfers response
{
"transferId": "d9ce59d435984396863058 1bb0",
"homeTransactionId": "string",
"from": {
"displayName": "John",
"fspId": "PayerFSP",
"idType": "MSISDN",
"idValue": "26787654321"},
"to": {
"type": "CONSUMER",
"idType": "MSISDN",
"idValue": "2551234567890",
"displayName": "Yaro",
"fspId": "PayeeFSP"
"supportedCurrencies": ["TZS"]},
"amountType": "SEND",
"currency": "BWP",
"amount": "300"
"currentState": "WAITING_FOR_CONVERSION_ACCEPTANCE",
"getPartiesResponse": {<Same as the previous responses>},
"conversionRequestId": "828cc75f1654415e8fcddf76cc",
"fxQuotesResponse": {
"body": {
"condition": "GRzLaTP7DJ9t4P-a_B...",
17 Here are the conversion terms



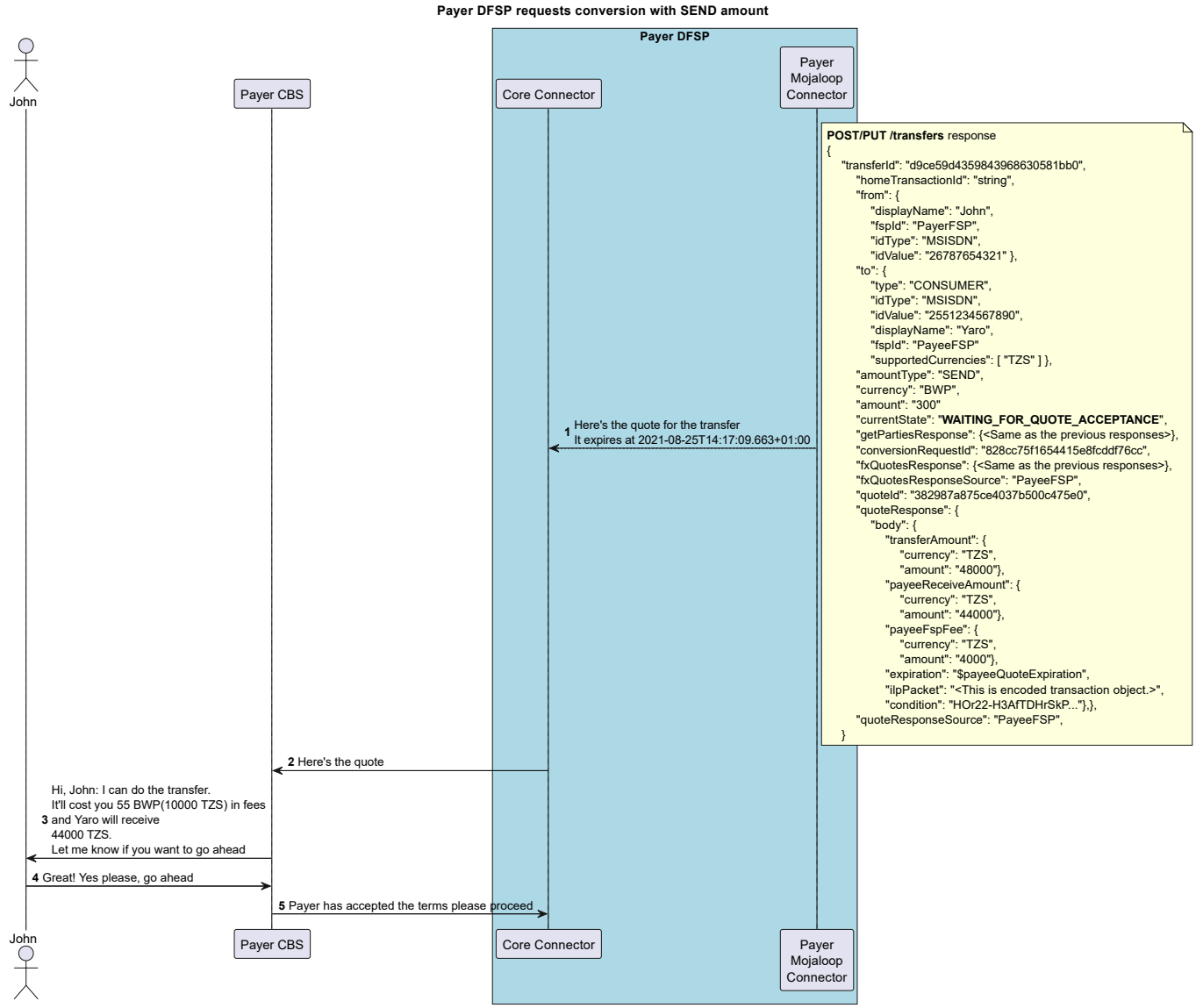
Agreement Phase - Mojaloop Connector

Liquidity in target currency has been secured. The Payer DFSP can now proceed to request an agreement of terms from the Payee DFSP. These terms are in target currency.



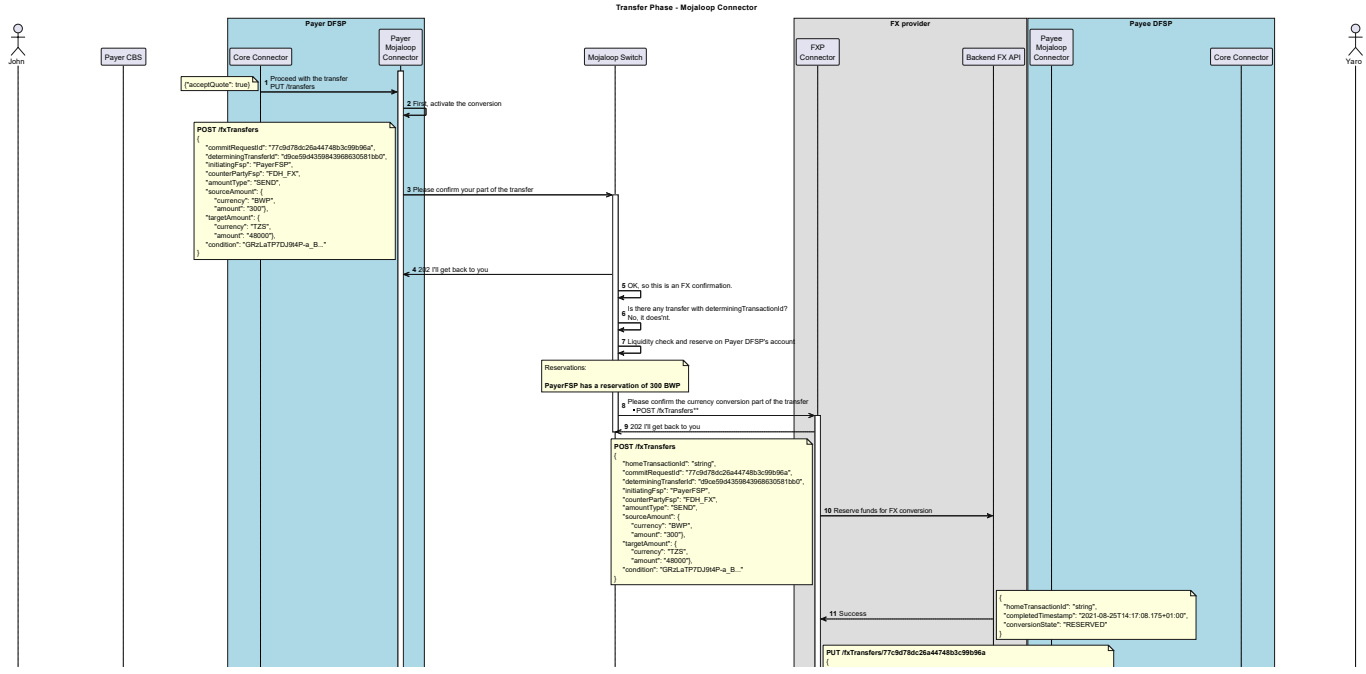
Sender Confirmation

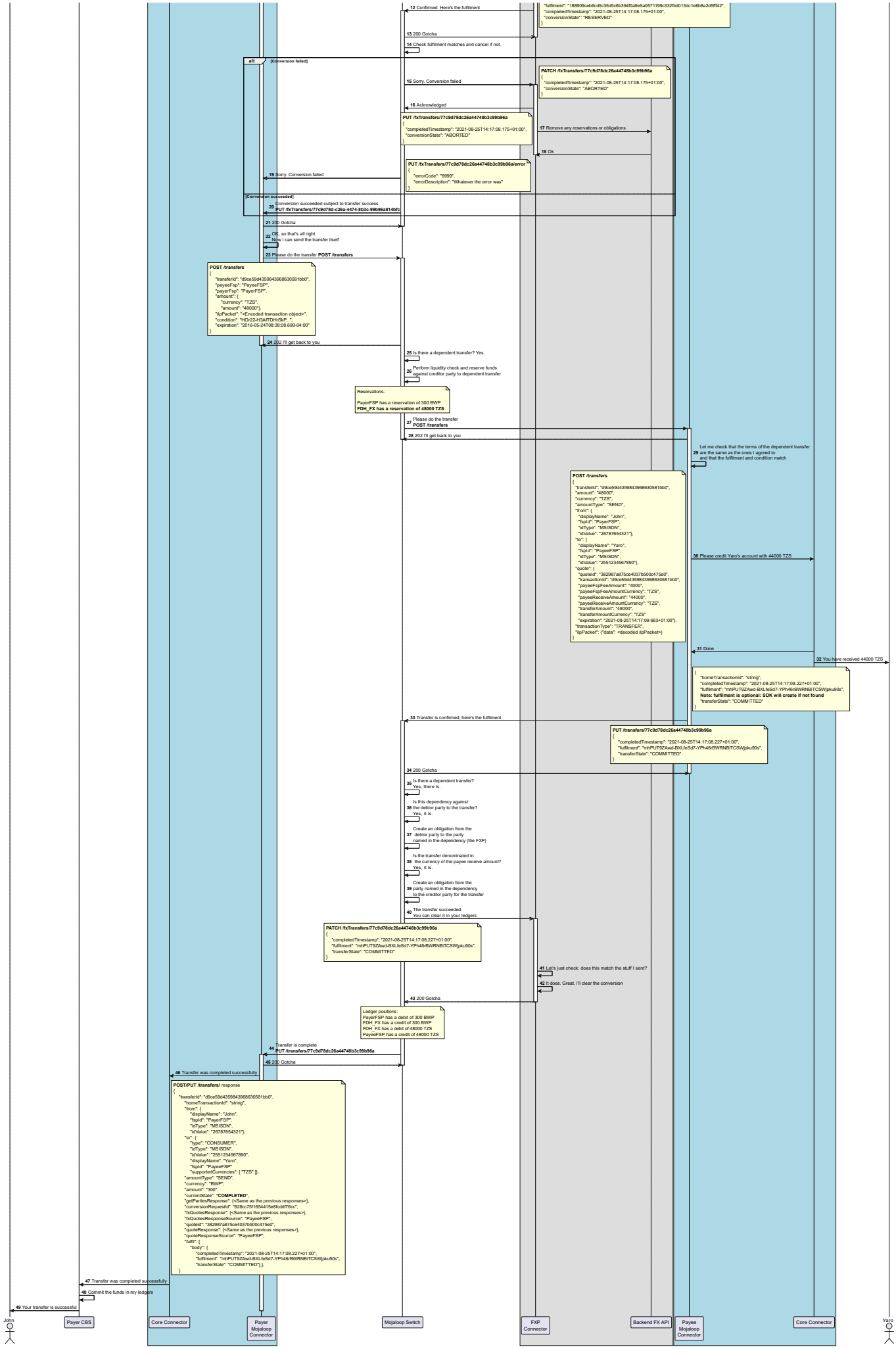
All the terms for the currency conversion and transfer have been obtained by the Payer DFSP and FXP. It is now time to collate those terms and present them to the Payer for confirmation.



Transfer Phase

The terms of the transfer have been accepted. The transfer phase can now commence.




"fulfilment": "188909ceb6cd5c35d5c6b3940fa9e5a0571199c332fbd013dc1e6b8a2d9fff42",
"completedTimestamp": "2021-08-25T14:17:08.175+01:00",
"conversionState": "RESERVED"
}
12 Confirmed. Here's the fulfilment
13 200 Gotcha
14 Check fulfilment matches and cancel if not.
alt
[Conversion failed]
PATCH /fxTransfers/77c9d78dc26a44748b3c99b96a
{
"completedTimestamp": "2021-08-25T14:17:08.175+01:00",
"conversionState": "ABORTED"
}
15 Sorry. Conversion failed
16 Acknowledged
PUT /fxTransfers/77c9d78dc26a44748b3c99b96a
{
"completedTimestamp": "2021-08-25T14:17:08.175+01:00",
"conversionState": "ABORTED"
}
17 Remove any reservations or obligations
18 Ok
PUT /fxTransfers/77c9d78dc26a44748b3c99b96a/error
{
"errorCode": "9999",
"errorDescription": "Whatever the error was"
}
19 Sorry. Conversion failed
[Conversion succeeded]
20 Conversion succeeded subject to transfer success
PUT /fxTransfers/77c9d78d-c26a-4474-8b3c-99b96a814bfc
21 200 Gotcha
22 Ok, so that's all right
Now I can send the transfer itself
23 Please do the transfer POST /transfers
POST /transfers
{
"transferId": "d9ce59d435984396863058 1bb0",
"payeeFsp": "PayeeFSP",
"payerFsp": "PayerFSP",
"amount": {
"currency": "TZS",
"amount": "48000"},
"ilpPacket": "<Encoded transaction object>",
"condition": "HOr22-H3AfTDHrSkP..",
"expiration": "2016-05-24T08:38:08.699-04:00"
}
24 202 I'll get back to you
25 Is there a dependent transfer? Yes
26 Perform liquidity check and reserve funds against creditor party to dependent transfer
Reservations:
PayerFSP has a reservation of 300 BWP
FDH_FX has a reservation of 48000 TZS
27 Please do the transfer
POST /transfers
28 202 I'll get back to you
29 Let me check that the terms of the dependent transfer are the same as the ones I agreed to and that the fulfilment and condition match
POST /transfers
{
"transferId": "d9ce59d435984396863058 1bb0",
"amount": "48000",
"currency": "TZS",
"amountType": "SEND",
"from": {
"displayName": "John",
"fspId": "PayerFSP",
"idType": "MSISDN",
"idValue": "26787654321"},
"to": {
"displayName": "Yaro",
"fspId": "PayeeFSP",
"idType": "MSISDN",
"idValue": "2551234567890"},
"quote": {
"quoteId": "382987a875ce4037b500c475e0",
"transactionId": "d9ce59d435984396863058 1bb0",
"payeeFspFeeAmount": "4000",
"payeeFspFeeAmountCurrency": "TZS",
"payeeReceiveAmount": "44000",
"payeeReceiveAmountCurrency": "TZS",
"transferAmount": "48000",
"transferAmountCurrency": "TZS"
"expiration": "2021-08-25T14:17:09.663+01:00"},
"transactionType": "TRANSFER",
"ilpPacket": {"data": <decoded ilpPacket>}
}
30 Please credit Yaro's account with 44000 TZS
31 Done
32 You have received 44000 TZS
{
"homeTransactionId": "string",
"completedTimestamp": "2021-08-25T14:17:08.227+01:00",
"fulfilment": "nhPUT9ZAwd-BXLfeSd7-YPh46rBWRNBITCSWjpku90s",
Note: fulfilment is optional: SDK will create if not found
"transferState": "COMMITTED"
}
33 Transfer is confirmed, here's the fulfilment
PUT /transfers/77c9d78dc26a44748b3c99b96a
{
"completedTimestamp": "2021-08-25T14:17:08.227+01:00",
"fulfilment": "nhPUT9ZAwd-BXLfeSd7-YPh46rBWRNBITCSWjpku90s",
"transferState": "COMMITTED"
}
34 200 Gotcha
35 Is there a dependent transfer?
Yes, there is.
36 Is this dependency against the debtor party to the transfer?
Yes, it is.
37 Create an obligation from the debtor party to the party named in the dependency (the FXP)
38 Is the transfer denominated in the currency of the payee receive amount?
Yes, it is.
39 Create an obligation from the party named in the dependency to the creditor party for the transfer
40 The transfer succeeded.
You can clear it in your ledgers
PATCH /fxTransfers/77c9d78dc26a44748b3c99b96a
{
"completedTimestamp": "2021-08-25T14:17:08.227+01:00",
"fulfilment": "nhPUT9ZAwd-BXLfeSd7-YPh46rBWRNBITCSWjpku90s",
"transferState": "COMMITTED"
}
41 Let's just check: does this match the stuff I sent?
42 It does. Great. I'll clear the conversion
43 200 Gotcha
Ledger positions:
PayerFSP has a debit of 300 BWP
FDH_FX has a credit of 300 BWP
FDH_FX has a debit of 48000 TZS
PayeeFSP has a credit of 48000 TZS
44 Transfer is complete
PUT /transfers/77c9d78dc26a44748b3c99b96a
45 200 Gotcha
46 Transfer was completed successfully
POST/PUT /transfers/ response
{
"transferId": "d9ce59d435984396863058 1bb0",
"homeTransactionId": "string",
"from": {
"displayName": "John",
"fspId": "PayerFSP",
"idType": "MSISDN",
"idValue": "26787654321"},
"to": {
"type": "CONSUMER",
"idType": "MSISDN",
"idValue": "2551234567890",
"displayName": "Yaro",
"fspId": "PayeeFSP"
"supportedCurrencies": ["TZS"]},
"amountType": "SEND",
"currency": "BWP",
"amount": "300"
"currentState": "COMPLETED",
"getPartiesResponse": {<Same as the previous responses>},
"conversionRequestId": "B28cc75f1654415e8fcddf76cc",
"fxQuotesResponse": {<Same as the previous responses>},
"fxQuotesResponseSource": "PayeeFSP",
"quoteId": "382987a875ce4037b500c475e0",
"quoteResponse": {<Same as the previous responses>},
"quoteResponseSource": "PayeeFSP",
"fulfil": {
"body": {
"completedTimestamp": "2021-08-25T14:17:08.227+01:00",
"fulfilment": "nhPUT9ZAwd-BXLfeSd7-YPh46rBWRNBITCSWjpku90s",
"transferState": "COMMITTED"},},
}
47 Transfer was completed successfully
48 Commit the funds in my ledgers
49 Your transfer is successful
John
Payer CBS
Core Connector
Payer Mojaloop Connector
Mojaloop Switch
FXP Connector
Backend FX API
Payee Mojaloop Connector
Core Connector
Yaro

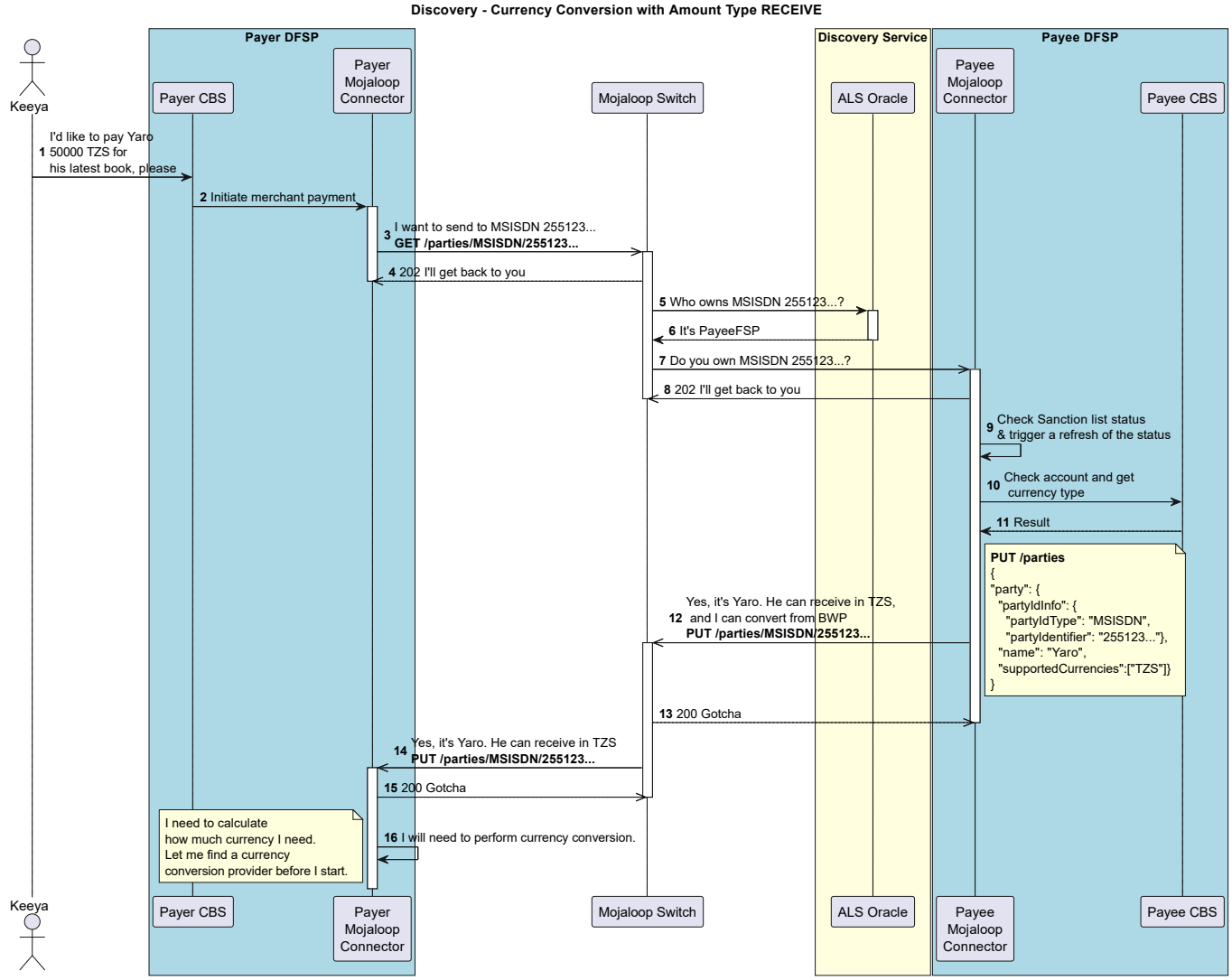
Currency Conversion Transfer (target currency)

For this use case, the Payer DFSP will specify the transfer with amount type **RECEIVE** and define the transfer amount in the **Payee's local currency** (the target currency). An secondary use case example for this is a cross boarder Merchant Payment.

Below is a detailed sequence diagram that shows the complete flow, and includes the Mojaloop connector and integration APIs for all participant organizations.

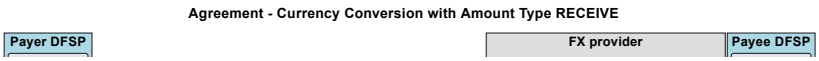
Discovery

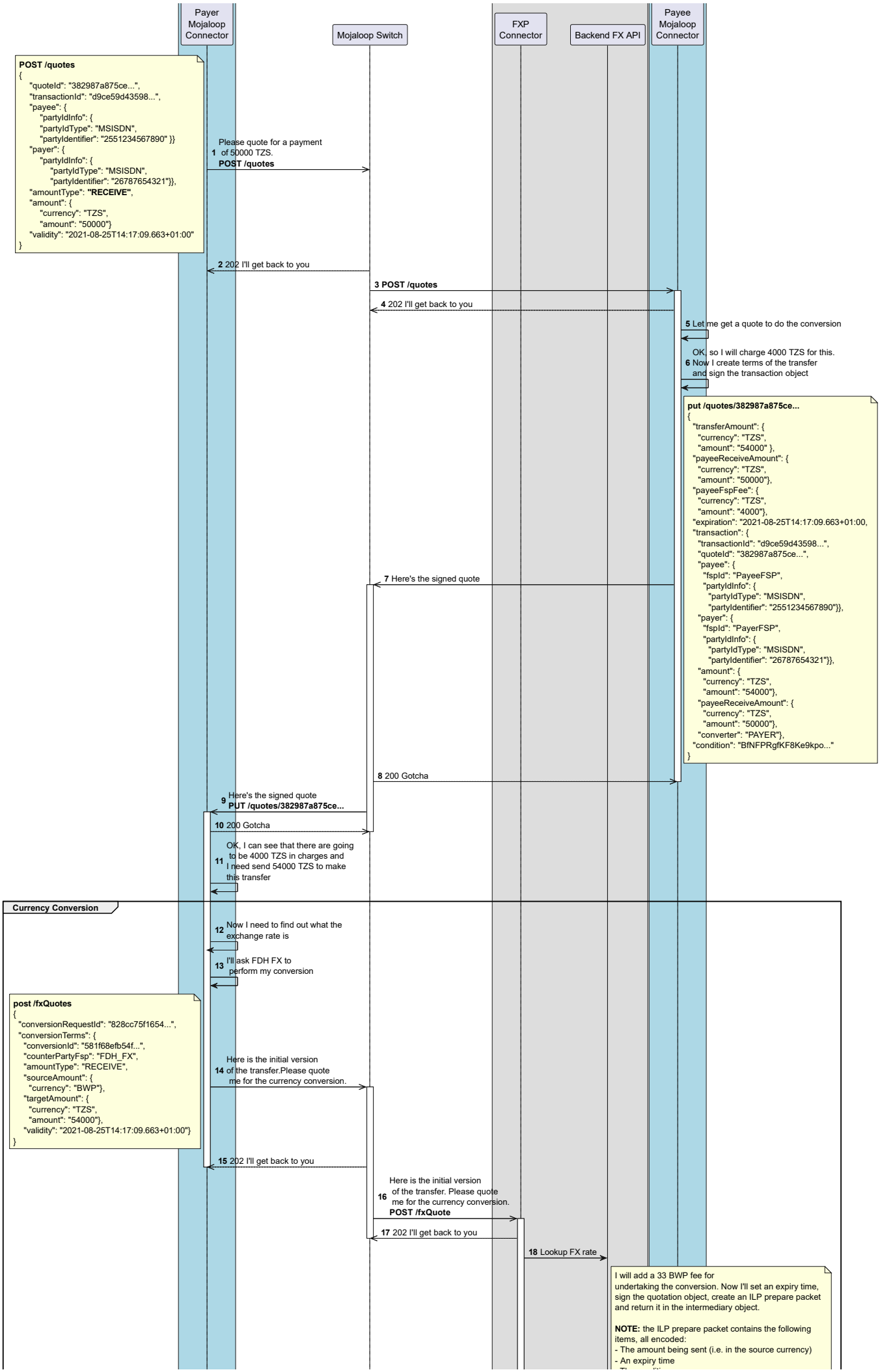
Mojaloop make use of an Oracle to identify the DFSP organization associated with the Party identifier. The Payee DFSP must respond to the GET /parties to confirm that the account exists and is active for that Party identifier. The supported currencies for that account are returned.

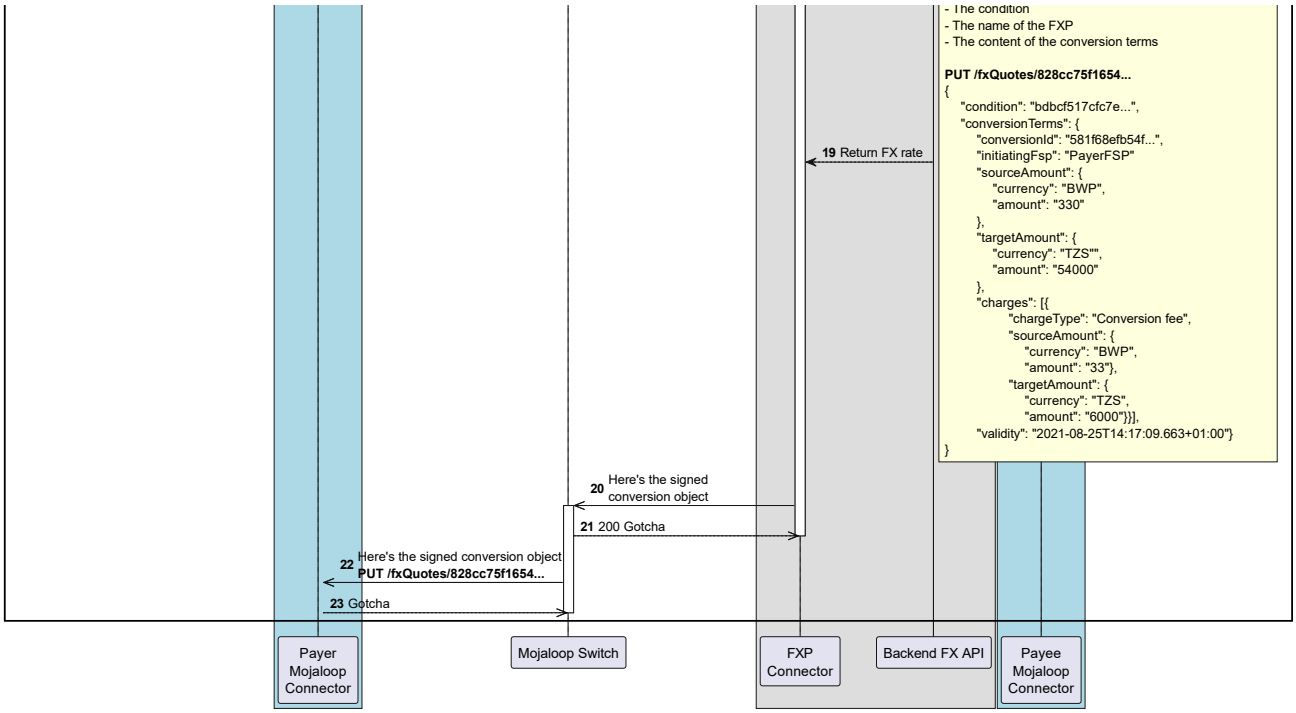


Agreement

The Payer DFSP does not support any of the Payee DFSP's currencies, requiring currency conversion within the Mojaloop Connector. Since the payment request is in the target currency, an agreement with the Payee DFSP must be established before initiating a liquidity request with the foreign exchange provider. The Payer DFSP first negotiates the transfer terms with the Payee DFSP, then uses its local cache of foreign exchange providers to select one and request liquidity coverage and a conversion rate.



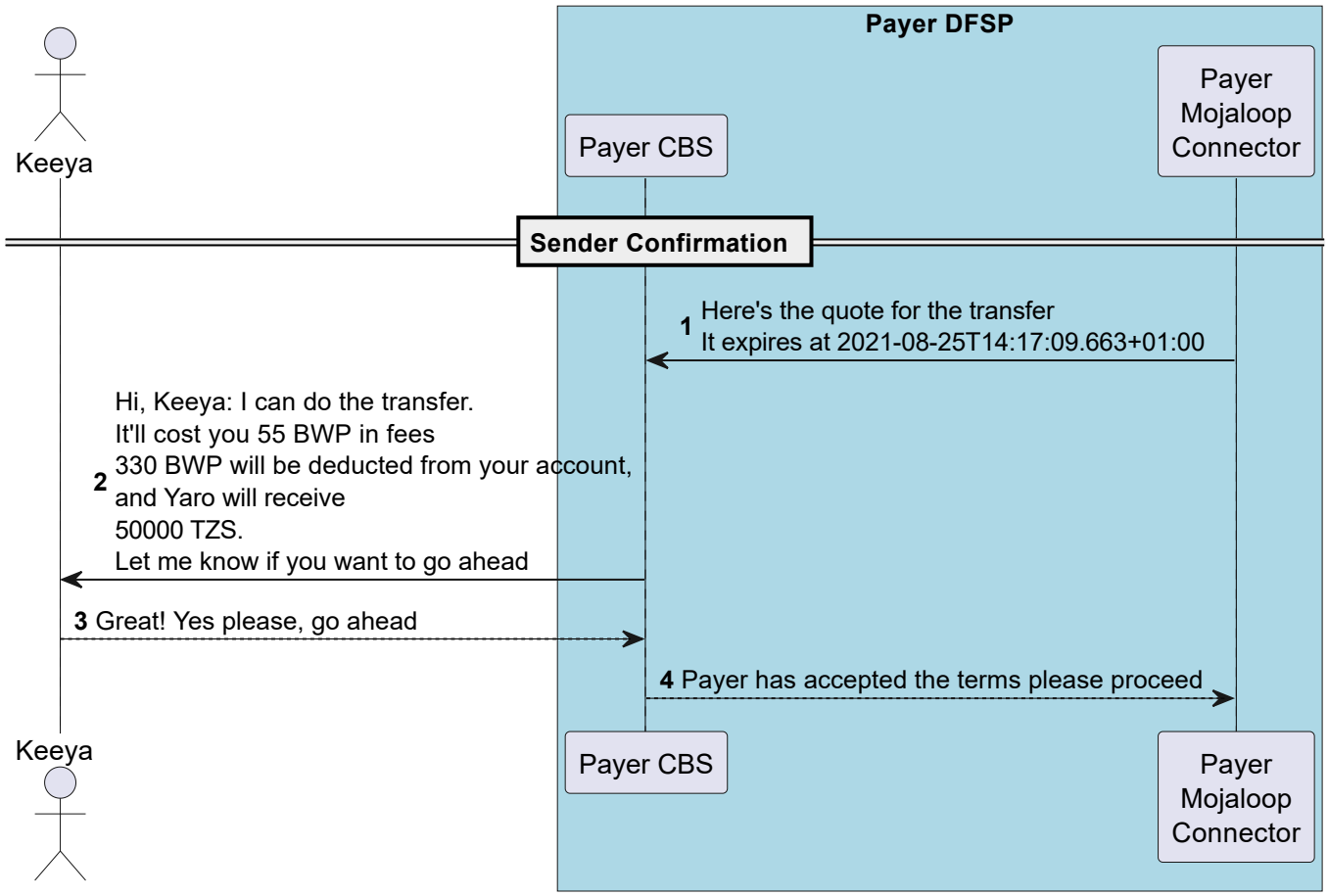

Payer Mojaloop Connector
Mojaloop Switch
FXP Connector
Backend FX API
Payee Mojaloop Connector
POST /quotes
{
"quoteId": "382987a875ce...",
"transactionId": "d9ce59d43598...",
"payee": {
"partyIdInfo": {
"partyIdType": "MSISDN",
"partyIdentifier": "2551234567890" }}
"payer": {
"partyIdInfo": {
"partyIdType": "MSISDN",
"partyIdentifier": "26787654321"}},
"amountType": "RECEIVE",
"amount": {
"currency": "TZS",
"amount": "50000"}
"validity": "2021-08-25T14:17:09.663+01:00"
}
1 Please quote for a payment of 50000 TZS.
POST /quotes
2 202 I'll get back to you
3 POST /quotes
4 202 I'll get back to you
5 Let me get a quote to do the conversion
6 OK, so I will charge 4000 TZS for this. Now I create terms of the transfer and sign the transaction object
put /quotes/382987a875ce...
{
"transferAmount": {
"currency": "TZS",
"amount": "54000" },
"payeeReceiveAmount": {
"currency": "TZS",
"amount": "50000"},
"payeeFspFee": {
"currency": "TZS",
"amount": "4000"},
"expiration": "2021-08-25T14:17:09.663+01:00,
"transaction": {
"transactionId": "d9ce59d43598...",
"quoteId": "382987a875ce...",
"payee": {
"fspId": "PayeeFSP",
"partyIdInfo": {
"partyIdType": "MSISDN",
"partyIdentifier": "2551234567890"}},
"payer": {
"fspId": "PayerFSP",
"partyIdInfo": {
"partyIdType": "MSISDN",
"partyIdentifier": "26787654321"}},
"amount": {
"currency": "TZS",
"amount": "54000"},
"payeeReceiveAmount": {
"currency": "TZS",
"amount": "50000"},
"converter": "PAYER"},
"condition": "BfNFPRgfKF8Ke9kpo..."
}
7 Here's the signed quote
8 200 Gotcha
9 Here's the signed quote
PUT /quotes/382987a875ce...
10 200 Gotcha
11 OK, I can see that there are going to be 4000 TZS in charges and I need send 54000 TZS to make this transfer
Currency Conversion
12 Now I need to find out what the exchange rate is
13 I'll ask FDH FX to perform my conversion
post /fxQuotes
{
"conversionRequestId": "828cc75f1654...",
"conversionTerms": {
"conversionId": "581f68efb54f...",
"counterPartyFsp": "FDH_FX",
"amountType": "RECEIVE",
"sourceAmount": {
"currency": "BWP"},
"targetAmount": {
"currency": "TZS",
"amount": "54000"},
"validity": "2021-08-25T14:17:09.663+01:00"}
}
14 Here is the initial version of the transfer.Please quote me for the currency conversion.
15 202 I'll get back to you
16 Here is the initial version of the transfer. Please quote me for the currency conversion.
POST /fxQuote
17 202 I'll get back to you
18 Lookup FX rate
I will add a 33 BWP fee for undertaking the conversion. Now I'll set an expiry time, sign the quotation object, create an ILP prepare packet and return it in the intermediary object.
NOTE: the ILP prepare packet contains the following items, all encoded:
- The amount being sent (i.e. in the source currency)
- An expiry time



Sender Confirmation

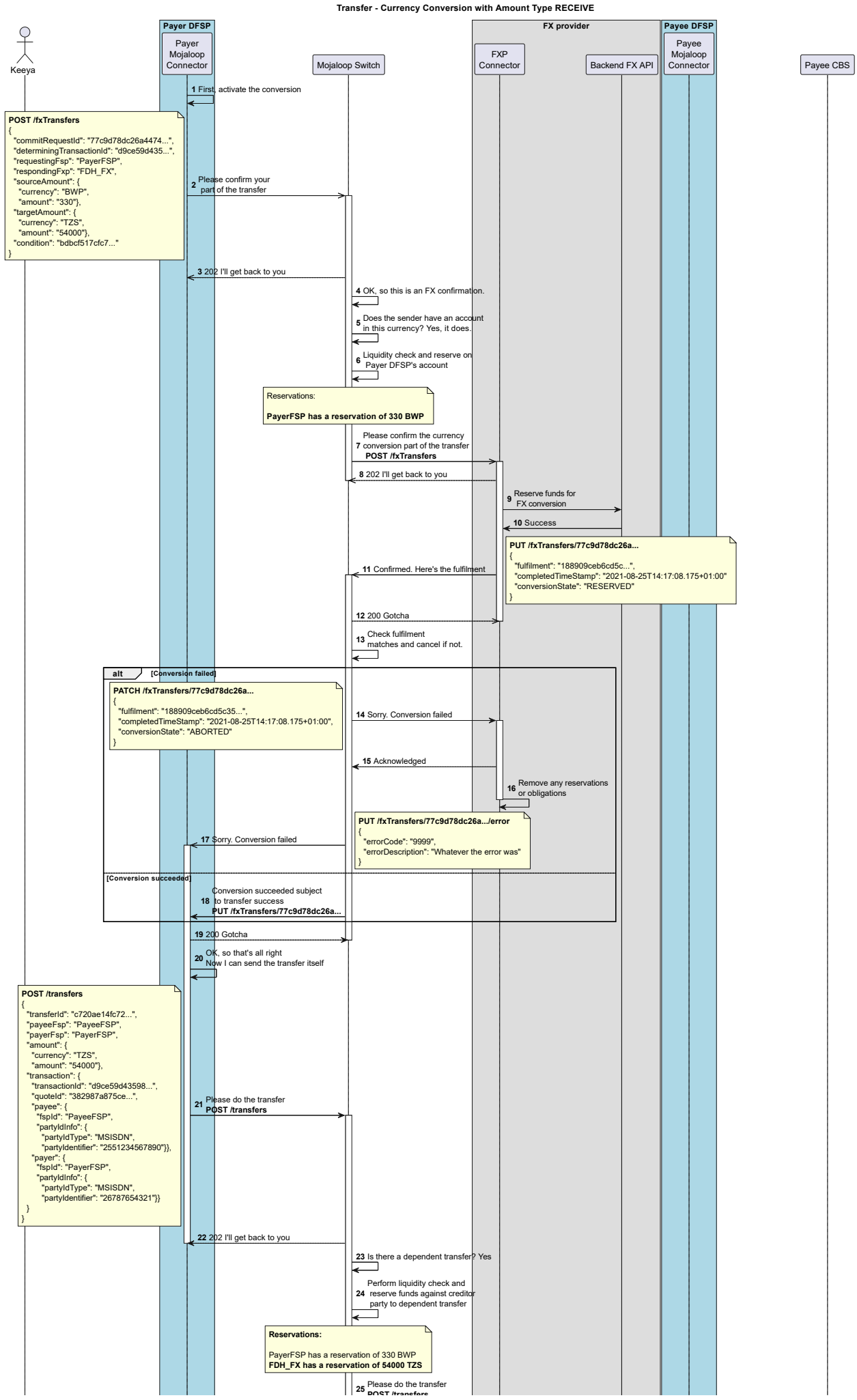
All the terms for the currency conversion and transfer have been obtained by the Payer DFSP and FXP. It is now time to collate those terms and present them to the Payer for confirmation.

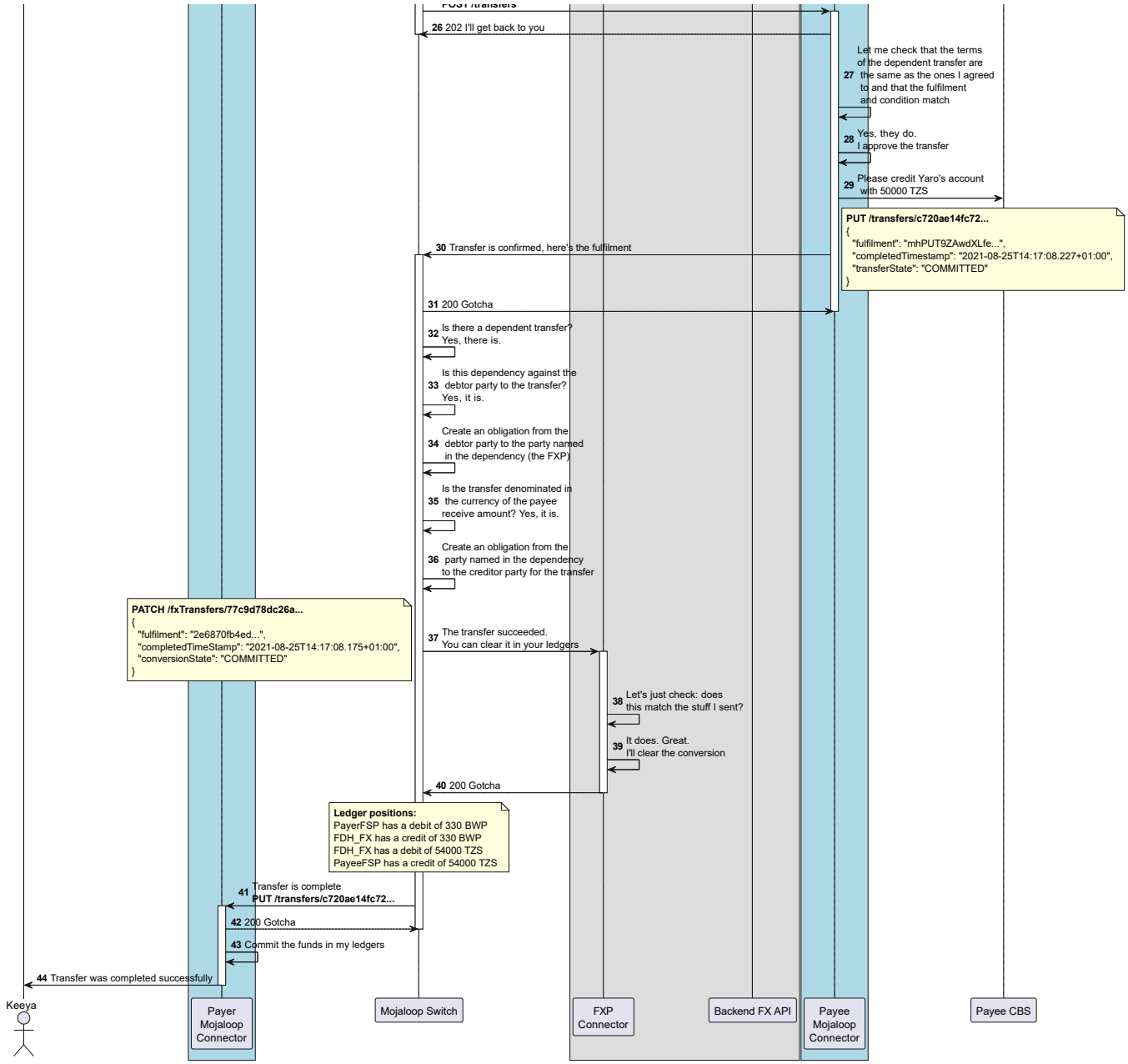
Currency Conversion with Amount Type RECEIVE



Transfer

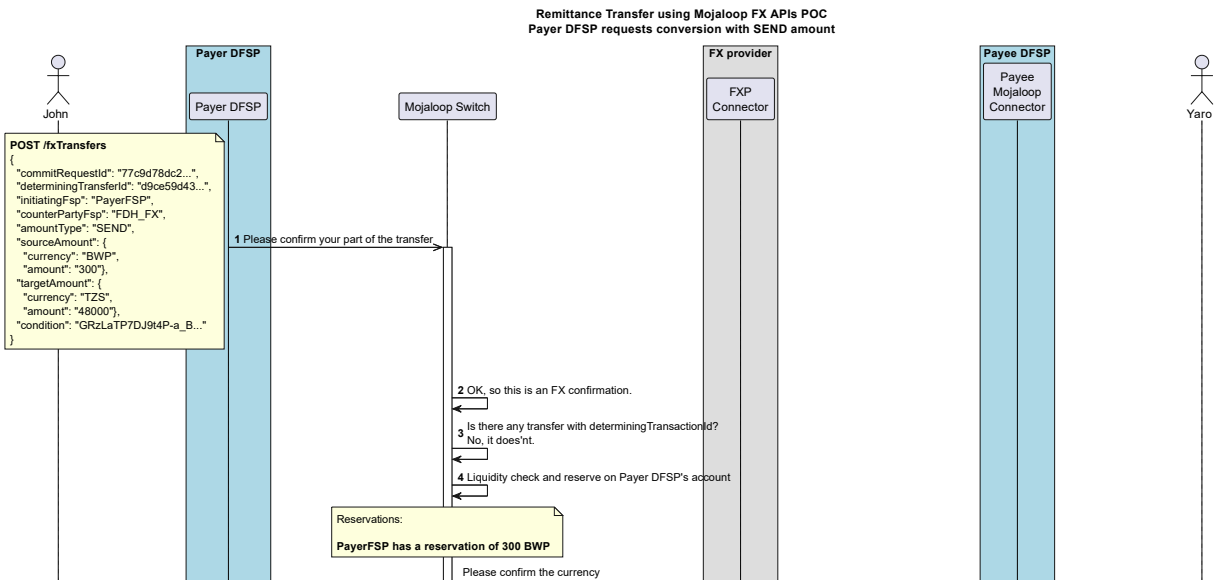
The terms of the transfer have been accepted. The transfer phase can now commence.

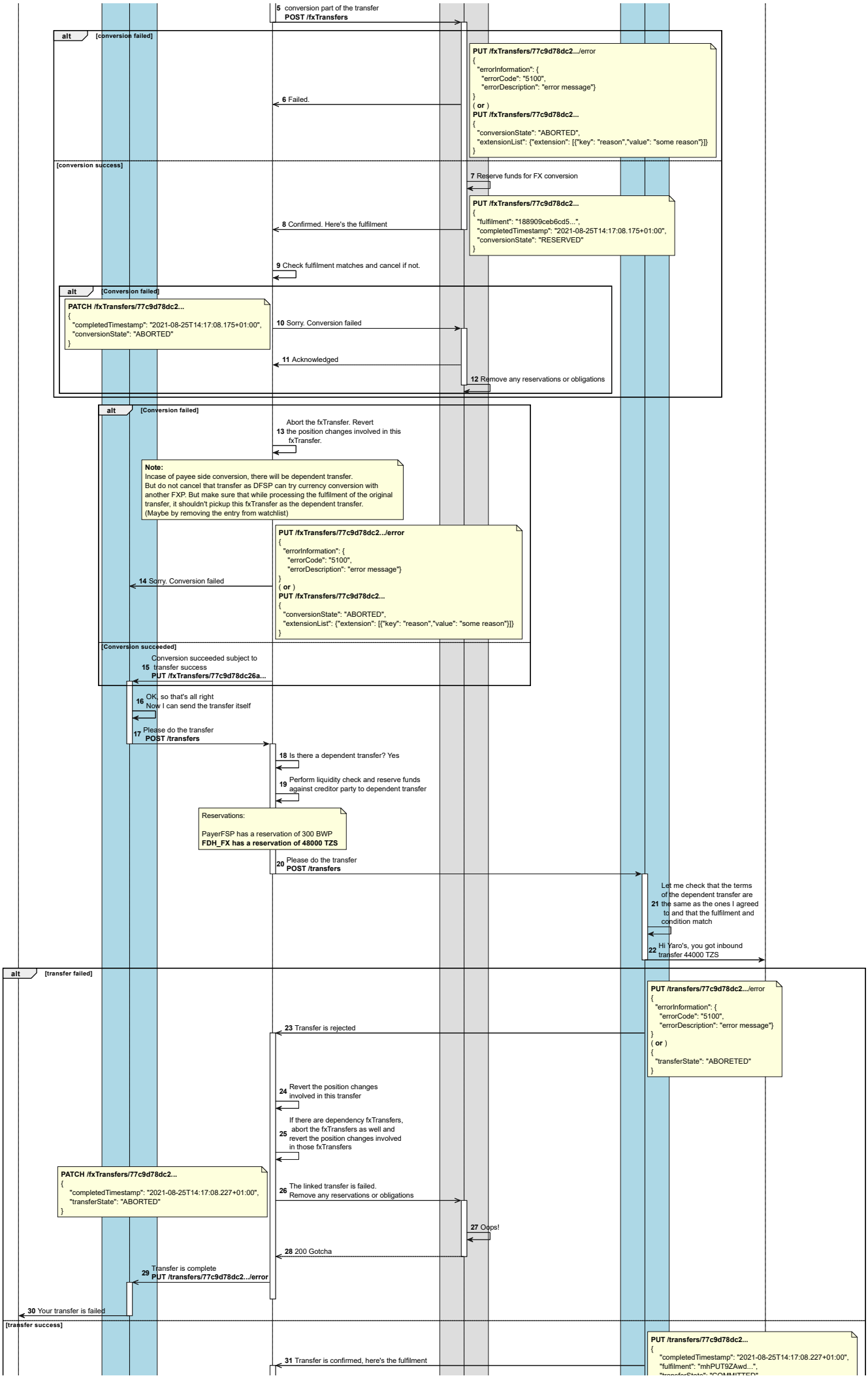


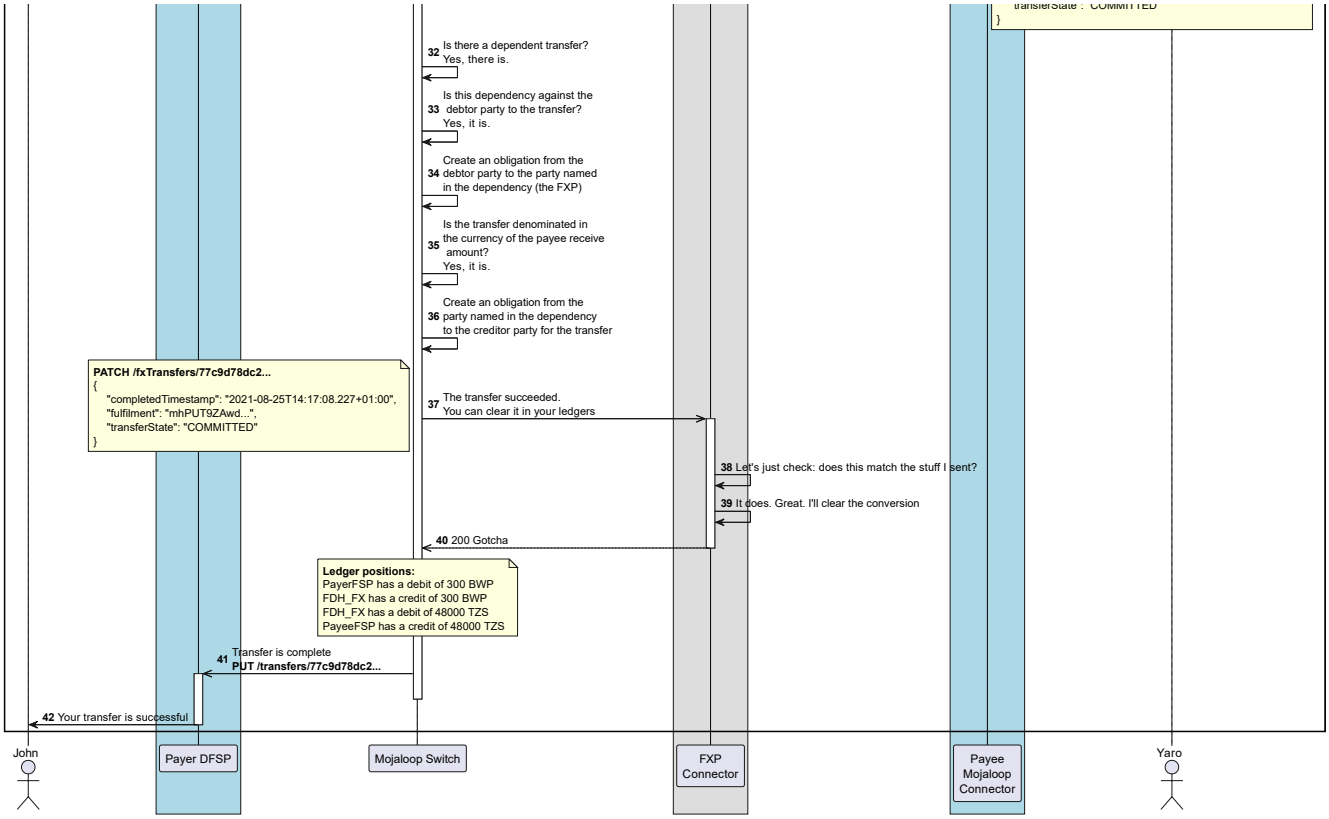


Abort flows

This sequence diagram show how the design implements abort messages during the currency conversion transfer phase.




5 conversion part of the transfer
POST /fxTransfers
alt
[conversion failed]
PUT /fxTransfers/77c9d78dc2.../error
{
"errorInformation": {
"errorCode": "5100",
"errorDescription": "error message"}
}
(or)
PUT /fxTransfers/77c9d78dc2...
{
"conversionState": "ABORTED",
"extensionList": {"extension": [{"key": "reason","value": "some reason"}]}
}
6 Failed.
[conversion success]
7 Reserve funds for FX conversion
PUT /fxTransfers/77c9d78dc2...
{
"fulfilment": "188909ceb6cd5...",
"completedTimestamp": "2021-08-25T14:17:08.175+01:00",
"conversionState": "RESERVED"
}
8 Confirmed. Here's the fulfilment
9 Check fulfilment matches and cancel if not.
alt
[Conversion failed]
PATCH /fxTransfers/77c9d78dc2...
{
"completedTimestamp": "2021-08-25T14:17:08.175+01:00",
"conversionState": "ABORTED"
}
10 Sorry. Conversion failed
11 Acknowledged
12 Remove any reservations or obligations
alt
[Conversion failed]
13 Abort the fxTransfer. Revert the position changes involved in this fxTransfer.
Note:
Incase of payee side conversion, there will be dependent transfer.
But do not cancel that transfer as DFSP can try currency conversion with another FXP. But make sure that while processing the fulfilment of the original transfer, it shouldn't pickup this fxTransfer as the dependent transfer.
(Maybe by removing the entry from watchlist)
PUT /fxTransfers/77c9d78dc2.../error
{
"errorInformation": {
"errorCode": "5100",
"errorDescription": "error message"}
}
(or)
PUT /fxTransfers/77c9d78dc2...
{
"conversionState": "ABORTED",
"extensionList": {"extension": [{"key": "reason","value": "some reason"}]}
}
14 Sorry. Conversion failed
[Conversion succeeded]
15 Conversion succeeded subject to transfer success
PUT /fxTransfers/77c9d78dc26a...
16 OK, so that's all right
Now I can send the transfer itself
17 Please do the transfer
POST /transfers
18 Is there a dependent transfer? Yes
19 Perform liquidity check and reserve funds against creditor party to dependent transfer
Reservations:
PayerFSP has a reservation of 300 BWP
FDH_FX has a reservation of 48000 TZS
20 Please do the transfer
POST /transfers
21 Let me check that the terms of the dependent transfer are the same as the ones I agreed to and that the fulfilment and condition match
22 Hi Yaro's, you got inbound transfer 44000 TZS
alt
[transfer failed]
PUT /transfers/77c9d78dc2.../error
{
"errorInformation": {
"errorCode": "5100",
"errorDescription": "error message"}
}
(or)
{
"transferState": "ABORETED"
}
23 Transfer is rejected
24 Revert the position changes involved in this transfer
25 If there are dependency fxTransfers, abort the fxTransfers as well and revert the position changes involved in those fxTransfers
PATCH /fxTransfers/77c9d78dc2...
{
"completedTimestamp": "2021-08-25T14:17:08.227+01:00",
"transferState": "ABORTED"
}
26 The linked transfer is failed.
Remove any reservations or obligations
27 Oops!
28 200 Gotcha
29 Transfer is complete
PUT /transfers/77c9d78dc2.../error
30 Your transfer is failed
[transfer success]
PUT /transfers/77c9d78dc2...
{
"completedTimestamp": "2021-08-25T14:17:08.227+01:00",
"fulfilment": "mhPUT9ZAwd...",
31 Transfer is confirmed, here's the fulfilment



Open API References

These Open API references are designed to be readable for both software an review person. The show the detailed requirements and implementations of the API design.

- FSPIOP v2.0 specification - Open Api definition.
- FSPIOP v2.0 ISO 20022 specification - Open Api definition.
- API Snippets Open Api definition
- Mojaloop Connector backend - Open Api definition
- Mojaloop Connector outbound - Open Api definition