Coil Technologies Pty Ltd

Integration of Mojaloop with ISO 8583

Proof of Concept

Renjith Palamattom
3-3-2019

Contents

Document History

Version	Date	Change Description
v0.1	03-03-2019	Initial Draft
v1.0	26-03-2019	Use case Updates
v1.1	28-03-2019	ISO 8583 - Open API mapping

Executive Summary

Coil Technologies Pty Ltd, has embarked on an initiative to integrate Interledger protocol with ISO 8583 protocol to provide seamless cross technology platform integration between legacy payment systems and cutting - edge technology trends.

As a part of this project, Coil has decided to make use of the Mojaloop system and related technologies, as a starting point to prove that such an integration between legacy and futuristic technologies would be possible. It is assumed that, it would always be better to embark on launching a new business in the EFT & Payments vertical, by integrating such systems with conventional payment systems and devices, rather than the radical replacement of an existing system, especially in a market where the payments systems are dominated and monopolized by multinational corporate establishments, using robust but dated legacy technologies.

Mojaloop

Mojaloop is an open source software for creating digital payments platforms that connect all customers, merchants, banks, and other financial providers in a country's economy. Rather than a financial product or application in itself, Mojaloop establishes a blueprint for technology that bridges all the financial products and applications in any given market.

ISO Payments Switch

A Payment Switch is Transaction processing software which receives transaction request from more than one interface (ATM, POS, mPOS, payment gateway and many more) and obtain authorization for transaction from defined hosts (Banking hosts for account authorization for debit cards or credit card hosts or prepaid card host).

A Switch is a comprehensive suite of Switching solution with capabilities to drive POS, ATM and eCommerce channels. Along with conforming to PCI/PA DSS standards, the solution is also capable of MasterCard/Visa/AMEX interfacing and processing.

The Switch is based on standard ISO 8583 and ISO JSON formats, which makes the solution ideal for a multitude of transaction channel processing including e-Commerce and mobile payments.

- o A full suite of modules which can be used to process financial transactions originating from ATM, BNA, POS, MPOS, E-Commerce and Hosts
- o Modular architecture
- o Supports Horizontal scaling
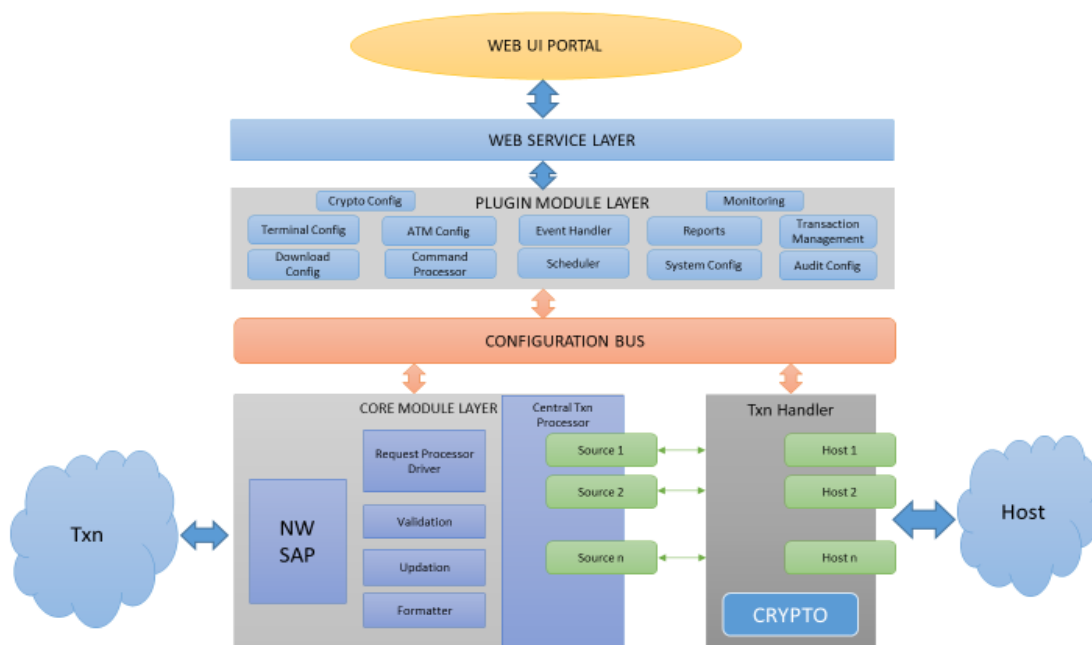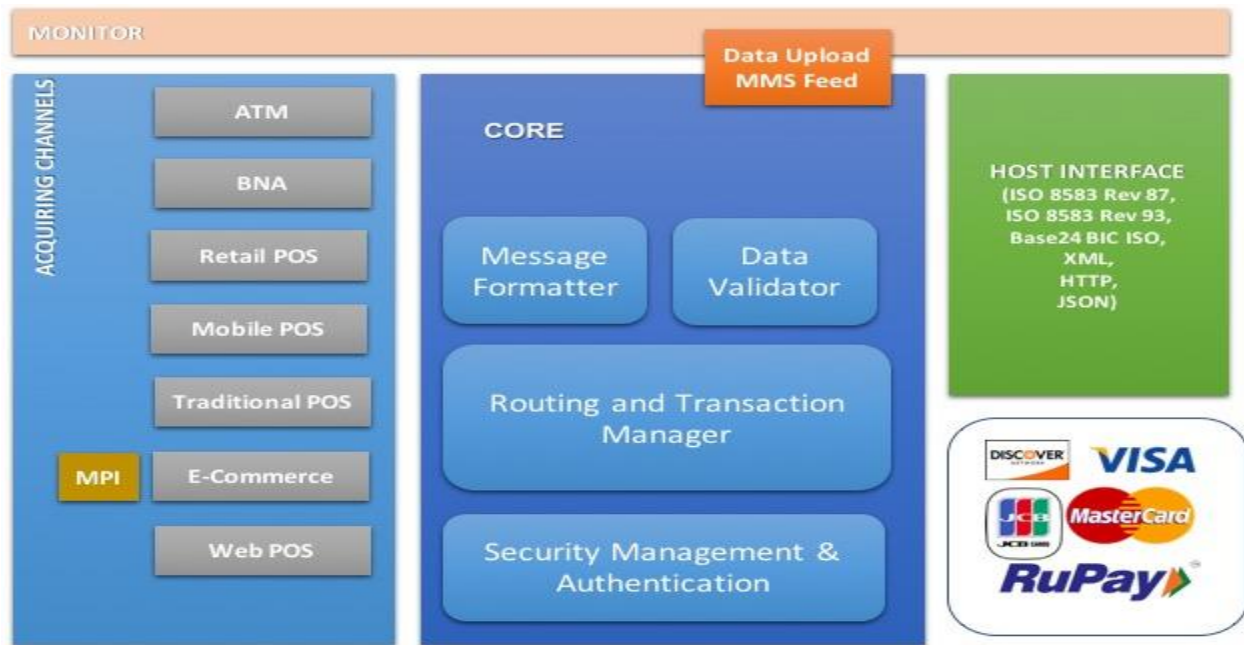
Terminal Interfaces

Terminal Driver is an application that has been developed to drive a device that accepts and processes EFT & card-based transactions, such as ATM, Pin pads, POS, etc. and could be integrated with any type of Switch/Host /Network. It has been designed in such a way that multiple instances could be implemented by separating database, which in turn reduces downtime. This also helps to handle

multiple FI's with a single implementation without dependency or sharing of individual sensitive data. Implementation of P2PE has been done to provide extra security to the card holder.

The system also has user friendly Web based Interface for Centralized management and monitor the system/ATM/POS/device's performance. The terminal driver/interface supports full download configuration flexibility with States, screens, languages, channels, marketing themes, etc. It also supports all standard Payments protocols such as ISO 8583/20022, ISO JSON, Base24 etc. as well is PCI/PA DSS, EMV compliant.
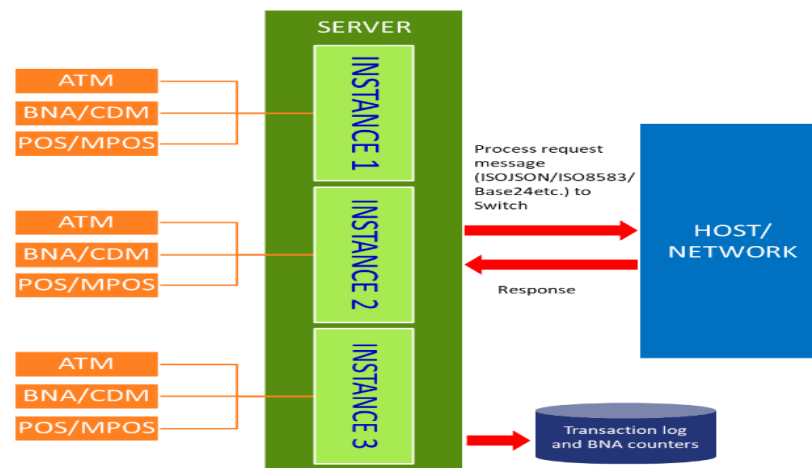
Integration Solution

The idea of the integration solution is to make use of Mojaloop system or similar systems, as an alternative to ISO based Payment switches. Similar to what Mojaloop is to SWIFT, we are proposing a solution that would be affordable to the participants and hence accelerating the process for financial inclusion.

The proposed solution would make use of the terminal application interface to send payment transactions through to Mojaloop system, by making use of an ISO-OPEN API converter/connecter plug-in or interface, similar to the Scheme Adapter in the Mojaloop system. As the Scheme adapter performs Mojaloop API to FSP API conversion, the custom plug-in/interface would function as a protocol and message translator between the ISO interface and the ML API or the Scheme Adapter.

Use Case I

ATM Initiated Cash Out

This use case involves two parties: ATM and Customer. ATM initiates a Cash-Out request from the customer account and the customer confirms the request by providing authentication (OTP) on ATM. The customer pre-generates an OTP for cash-out and uses this OTP on ATM device to initiate ATM Cash-out. The Payer FSP validates the OTP received in ATM-Initiated Cash-Out request for the validity of OTP as well as for authentication. If the customer authentication via OTP is successful; then the customer's

account will be debited at Payer FSP and ATM account maintained at Payee FSP will be credited. As a result, the customer receives cash from ATM
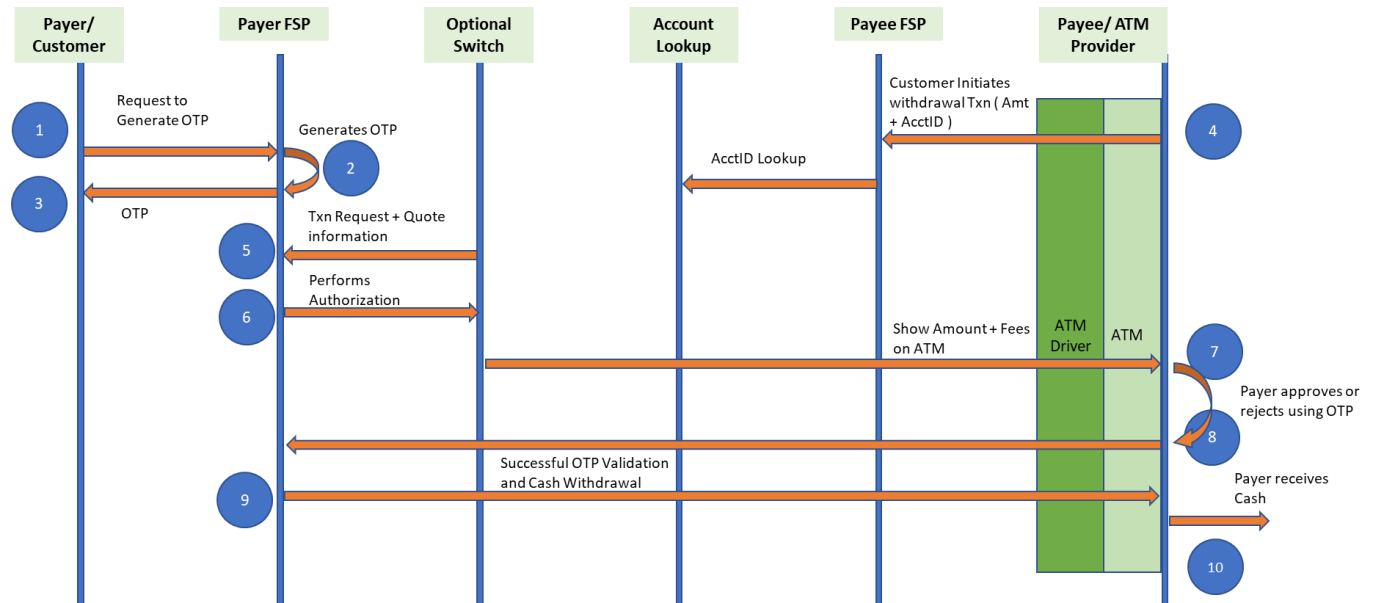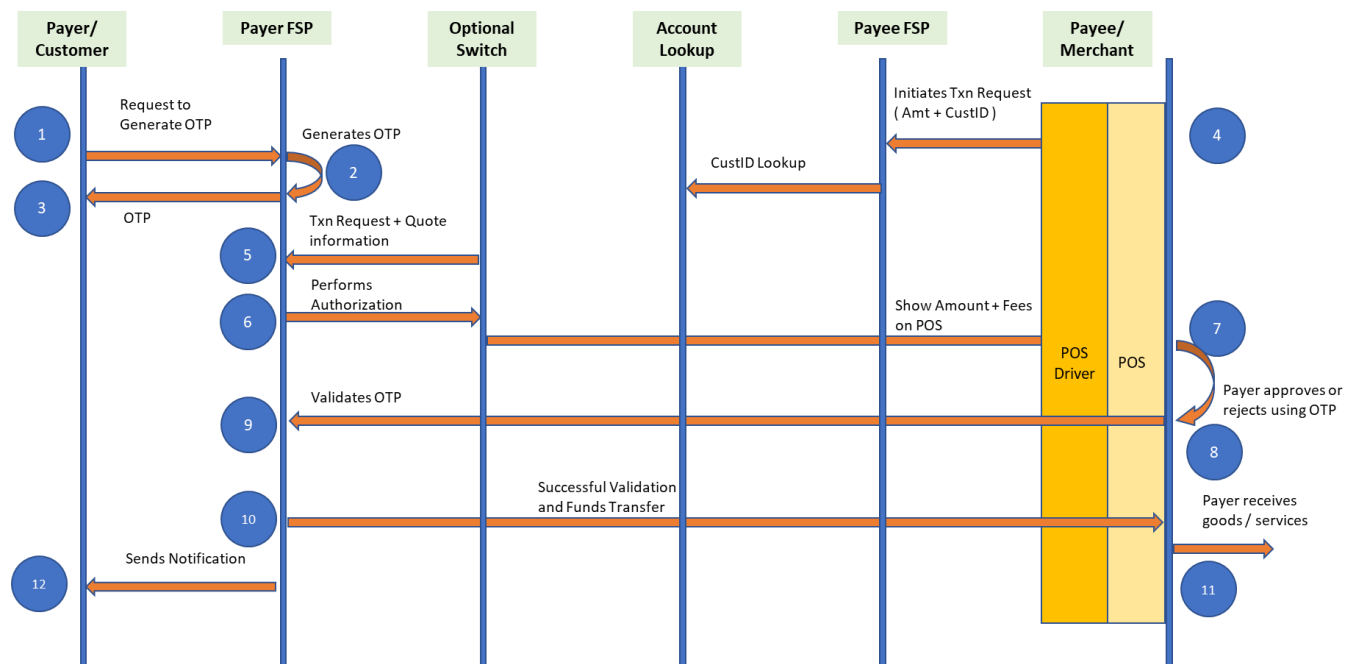

Transaction flow

Firstly, customer will have to generate OTP through mobile application for desired ATM Initiated Cash-out amount. This OTP will be sent to customer's mobile number.

- Customer generates an OTP before initiating the transaction request from ATM.
- The customer initiates withdrawal transaction on the ATM by entering their account ID and amount.
- Cash Out Request will be generated by ATM Driver in NDC protocol. This will be converted to OPEN API call and will be sent to Mojaloop.
- Mojaloop will perform Account lookup and the transaction request will be sent to Payer FSP for authentication.
- The Payer FSP validates the transaction request and also calculate the Quote for the transaction.
- The calculated Quote will be displayed on the Terminal for confirmation by the Payee.
- The Customer will authenticate the transaction by entering pre-generated OTP.
- Payer FSP will authenticate the OTP.
    - If successful, the customer's account will be debited, and the ATM account maintained by Payee FSP will be credited.
- Payer FSP will send response back to Mojaloop.
- Response will be received by ATM Driver in OPEN API. This will be converted to ISO and is sent to ATM.
- ATM will perform the actions as mentioned by the message. (Dispense and Print etc.)

Use Case II

Merchant-Initiated Merchant Payment Authorized on POS

This use case describes a merchant payment initiated by a merchant using a device such as POS, and how to authorize a transaction with an OTP or a QR code. The merchant initiates a merchant payment transaction using a POS device. This device has the capability to capture the customer's authorization on POS instead of the customer's mobile device. The authorization information captured in POS should be sent to Payer FSP to perform the authorization. The business process involves two parties, Merchant and Customer. The merchant initiates a request to pay for the customer, and the customer reviews the payment request on POS and authorizes the payment by OTP or QR code on the POS itself. The customer authentication information is sent from Payee FSP to Payer FSP for authentication by Payer FSP. If authentication is successful then transaction will be posted on Payer FSP and Payee FSP.

Transaction flow

- Customer requests for an OTP (pre-generate OTP using mobile app/CMS)
- Merchant will initiate payment for the desired amount and Customer ID through POS device.
- The request will be converted from ISO to OPEN API and will be sent to Mojaloop. From there account lookup will be done and the request will be send to Payer FSP for authorization.
- The Payer FSP validates the transaction request and also calculate the Quote for the transaction.
- The calculated Quote will be displayed on the Terminal for confirmation by the Payee.

- The Payer FSP will authorize the transaction with dynamic OTP (or QR Code) which is generated through mobile application and entered by the customer.
- If Payer FSP authorizes the transaction, funds will be sent to Merchant (Payee) FSP. If Payer FSP declines, the transaction will be aborted.
- Response will be sent back to POS driver through Mojaloop in OPEN API. It will be converted to ISO(POS) and will send the response to POS.
- Notification will be sent to Payer and Payee from respective FSP's.
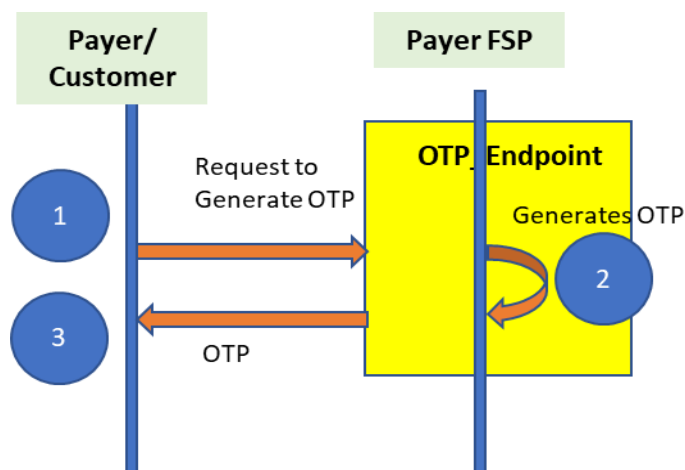
Simulator Endpoints

As a part of the terminal integration project, we will be making use of the FSP simulator, to function as the Payer FSP and Payee FSP as per the use cases.

In order to cater for this functionality, three new endpoints will be implemented on the simulator as below:

- **OTP Endpoint**
- **ATM Endpoint**
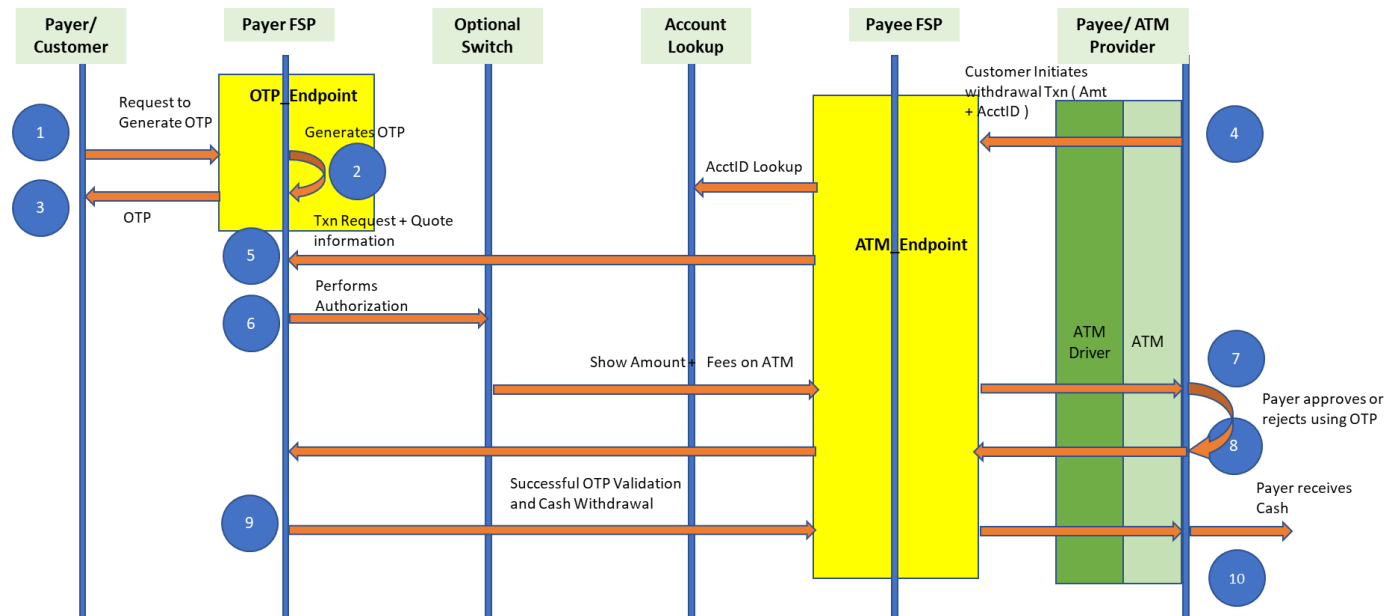- **POS Endpoint**

OTP Endpoint
The OTP Endpoint will be responsible for the handling of OTP requests from the Payer source. Upon receipt of an OTP request, the endpoint will generate the OTP and respond with the generated OTP through the same channel.

ATM Endpoint

ATM Endpoint will be responsible for the handling and processing of transactions originating from ATM devices. The transactions in ISO/NDC/DDC formats will be converted to Open API and processed through the simulator.
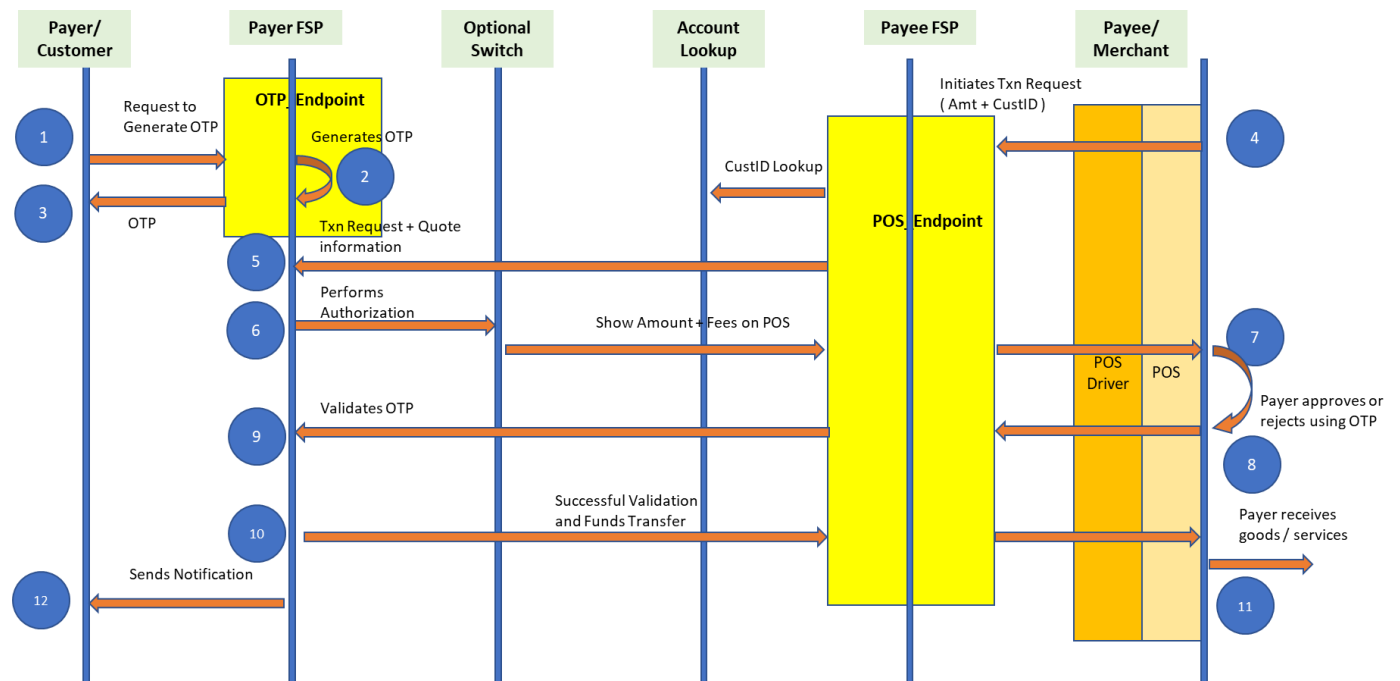

Transaction flow

Firstly, customer will have to generate OTP through mobile application for desired ATM Initiated Cash-out amount. This request will be processed by the OTP_Endpoint and the resultant OTP will be sent to customer's mobile number.

- Customer generates an OTP using OTP_Endpoint before initiating the transaction request from ATM.
- The customer initiates withdrawal transaction on the ATM by entering their account ID and amount.
- Cash Out Request will be generated by ATM Driver in NDC protocol.
- This will be sent to ATM_Endpoint where it will be converted to OPEN API call and will be sent to Mojaloop.
- Mojaloop will perform Account lookup and the transaction request will be sent to Payer FSP for authentication.
- The Payer FSP validates the transaction request and also calculate the Quote for the transaction.
- The calculated Quote will be sent back to the ATM_Endpoint which intern will be displayed on the Terminal for confirmation by the Payee.

- The Customer will authenticate the transaction by entering pre-generated OTP.
- Transaction authentication request will be sent from the ATM Driver to ATM_Endpoint for validation.
- Payer FSP will authenticate the OTP.
  - If successful, the customer's account will be debited, and the ATM account maintained by Payee FSP will be credited.
- Payer FSP will send response back to Mojaloop.
- Response will be received by ATM_Endpoint in OPEN API. This will be converted to ISO and is sent to ATM.
- ATM will perform the actions as mentioned by the message. (Dispense and Print etc.)

POS Endpoint

The POS Endpoint is implemented to handle and process transactions originating from POS and mPOS devices. Similar to the ATM Endpoint, the transactions from POS devices in the ISO format will be processed by the endpoint through the simulator.

Transaction flow

- Customer requests for an OTP (pre-generate OTP using mobile app/CMS)
- OTP generation process will be processed using OTP_Endpoint for OTP generation and will be sent to the customer.
- Merchant will initiate payment for the desired amount and Customer ID through POS device.
- The transaction request will be converted from ISO to OPEN API using POS_Endpoint and will be sent to Mojaloop. From there account lookup will be done and the request will be sent to Payer FSP for authorization.
- The Payer FSP validates the transaction request and calculate the Quote for the transaction.
- The calculated Quote will be sent to the POS_Endpoint where the request message will be converted to ISO format to be displayed on the Terminal for confirmation by the Payee.
- The Payee will enter the pre-generated OTP and the transaction request will be sent to Payer FSP after converting the ISO message to the OPEN API call at the POS_Endpoint
- The Payer FSP will authorize the transaction with dynamic OTP (or QR Code) which is generated through mobile application and entered by the customer.
- If Payer FSP authorizes the transaction, funds will be sent to Merchant (Payee) FSP. If Payer FSP declines, the transaction will be aborted.
- Response will be sent back to POS driver through Mojaloop in OPEN API and converted to ISO format at the POS_Endpoint.
- Notification will be sent to Payer and Payee from respective FSP's.

ISO 8583 to Open-API mapping

ATM Endpoint

ATM Endpoint converts the ISO message coming from ATM driver onto Open-API and sends it to Mojaloop.

ATM Endpoint Transaction mapping

API Field Name	API Format	ISO 8583 Field Name	ISO 8583 Format	Example
transactionRequestId	UUID	Field 11 - System trace audit number (STAN)	n 6	a8323bc6-c228-4df2-ae82-e5a997baf898
payee	JSON Complex type: Enum of String(1..32), String(1..4), String(1..128)	Field41 - Card acceptor terminal identification Field 42 - Card acceptor identification code	ans 8 ans 15 ans 40	"payee": { "partyIdInfo": { "partyIdType": "MSISDN", "partyIdentifier": "123456789", "fspId": "MobileMoney" }, "personalInfo": {

		Field 43 - Card acceptor name/location		"complexName": { "firstName": "Henrik", "lastName": "Karlsson" } } }
payer	JSON Complex type: Enum of String(1..32), String(1..128), String(1..128), String(1..32)	Field 102 - Mobile number/Account Number		"payer": { "personalInfo": { "complexName": { "firstName": "Mats", "lastName": "Hagman"
amount	String(1..4)	Field 4 - Amount transaction	n 12	"amount": { "amount": "99", "currency": "USD"
transactionType	Enum of String(1..32)	Field 3 - Processing code Transaction type - 01	n 6	"transactionType": { "scenario": "TRANSFER", "initiator": "PAYER", "initiatorType": "CONSUMER"
note	String(1..128)	Field 48	ans999LLVAR	
geoCode	JSON String (Latitude, Longitude)	Field 48+Field 43	ans999LLVAR	
authenticationType	Enum of String(1..32)	Field 48	ans999LLVAR	"OTP"
expiration	JSON String DateTime	Field 7 – Transmission date & time	n 10,MMDDhhmmss	Transaction date time + time delay
extensionList	Extension	Additional info	ans..999, LLLVAR	

POS Endpoint

POS endpoint converts the ISO message coming from the POS driver onto Open API message and forward it to Mojaloop.

POS Endpoint Transaction mapping

API Field Name	API Format	ISO 8583 Field Name	ISO 8583 Format

transactionRequestId	UUID	Field 11 - System trace audit number (STAN)	n 6
payee	JSON Complex type: Enum of String(1..32), String(1..4), String(1..128)	Field41 - Card acceptor terminal identification Field 42 - Card acceptor identification code Field 43 - Card acceptor name/location	ans 8 ans 15 ans 40
payer	JSON Complex type: Enum of String(1..32), String(1..128), String(1..128), String(1..32)	Field 102 - Mobile number/Account Number	"payer": { "personalInfo": { "complexName": { "firstName": "Mats", "lastName": "Hagman"
amount	String(1..4)	Field 4 - Amount transaction	n 12
transactionType	Enum of String(1..32)	Field 3 - Processing code Transaction type - 00	n 6
note	String(1..128)	Field 48	ans999LLVAR
geoCode	JSON String (Latitude, Longitude)	Field 48	ans999LLVAR
authenticationType	Enum of String(1..32)	Field 48	ans999LLVAR
expiration	JSON String DateTime	Field 7 – Transmission date & time	n 10,MMDDhhmmss
extensionList	Extension	Additional info	ans..999, LLLVAR

OTP Endpoint

This endpoint creates OTP for the card less ATM transaction and POS device transaction.