Payment Network – API Reference Guide

API Documentation

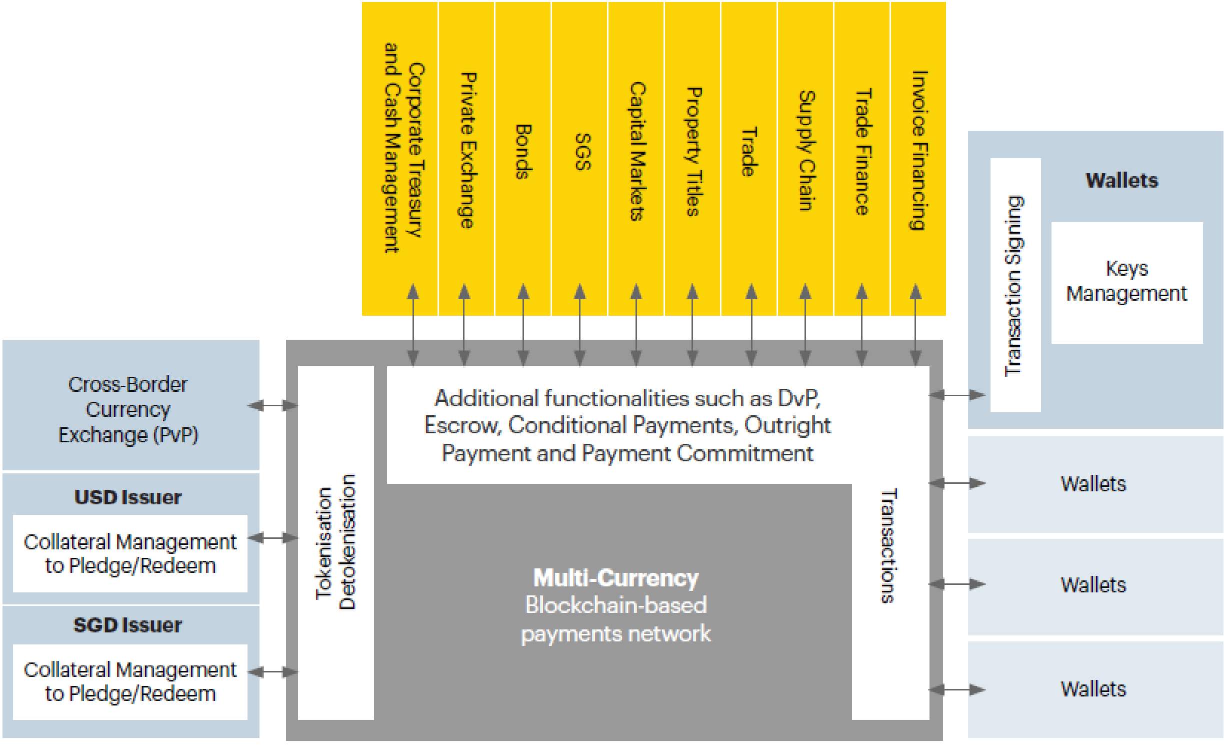
This document describes the technical details of connectivity and various APIs available in Payment Network.

Version: 0.7

Contents

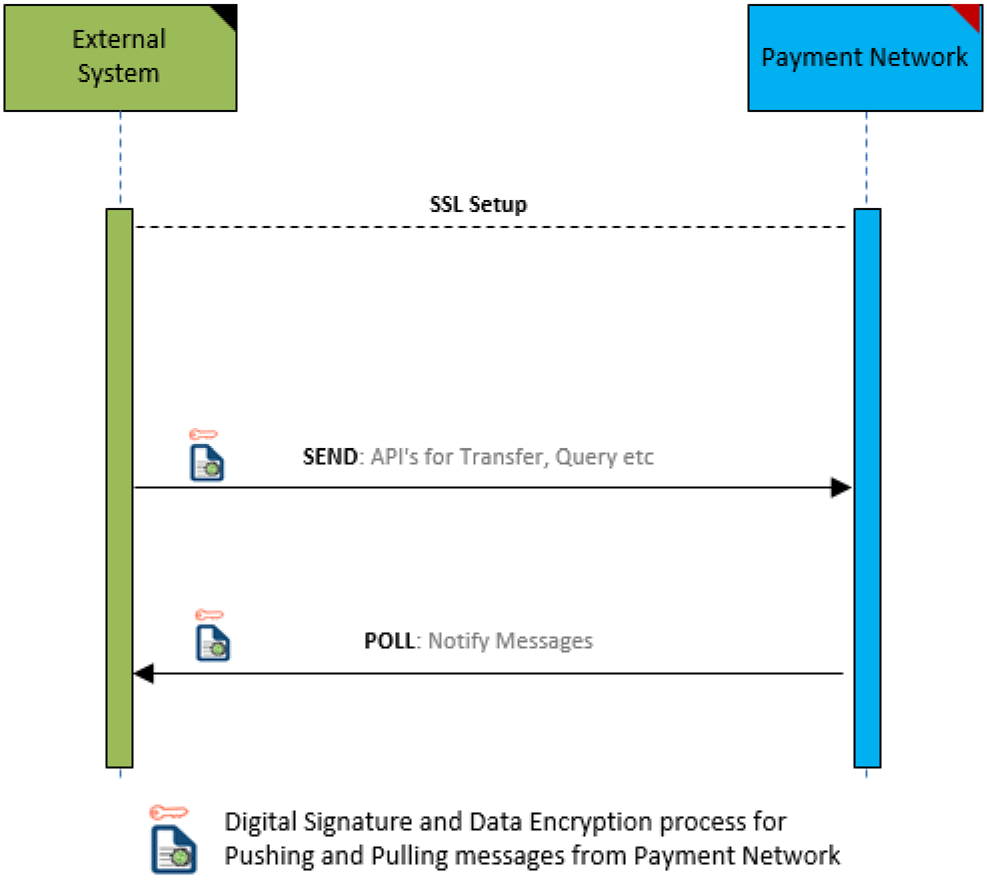
1 Overview

Phase 5 of Project Ubin aims to develop a Multi-Currency Payments Network on DLT and enable integration with other DLT networks to support payments for different use-cases. This Multi-Currency Payments Network (Ubin V) will support wholesale interbank as well as corporate payments with multiple currencies on the same network. Below is the high-level view of overall setup.



Corporate Treasury and Cash Management
Private Exchange
Bonds
SGS
Capital Markets
Property Titles
Trade
Supply Chain
Trade Finance
Invoice Financing
Wallets
Transaction Signing
Keys Management
Cross-Border Currency Exchange (PvP)
USD Issuer
Collateral Management to Pledge/Redeem
SGD Issuer
Collateral Management to Pledge/Redeem
Tokenisation Detokenisation
Additional functionalities such as DvP, Escrow, Conditional Payments, Outright Payment and Payment Commitment
Multi-Currency Blockchain-based payments network
Transactions
Wallets
Wallets
Wallets

2 Payment Network – Connectivity


External System
Payment Network
SSL Setup
SEND: API's for Transfer, Query etc
POLL: Notify Messages
Digital Signature and Data Encryption process for Pushing and Pulling messages from Payment Network

2.1 Client Authentication

Client applications have to be authenticated and authorized to send or poll messages from Gateway. Each request sent by the client to Gateway is required to have a Bearer token, generated and retrieved from the identity and access management application, in the request header. The client will be allowed to send or poll messages only if authenticated and authorized.

Below are two different ways external system will connect to payment network.

Sending Request with Gateway-library:

Gateway-library has integrated token retrieval with sending requests. Client applications using this library have to enable authentication and specify the client credentials for the library to retrieve token and send it with each request. Please refer to readme document present in the Gateway-library for detailed instruction.

Sending Request without Gateway-library:

Token Retrieval:

POST https://apilab.iinconnect.com/token

Client applications can retrieve access tokens and add it as Bearer Token to each subsequent request header sent to the gateway for authentication. Access tokens have an expiration value of 15 minutes. If clients want to make API requests 15 minutes after getting an access token, they will need to obtain a new token.

Request JSON Object:

The request body should be a JSON object consisting of the following fields:

- **grant_type** (string) – This should be set to client_credentials
- **client_id** (string) – This client ID shared during onboarding
- **client_secret** (string) – The secret key shared during onboarding

Response Status:

- 2xx – Success
- 403 – Unauthorized
- 4xx – Request Error
- 5xx – System Error

Response JSON Object:

The request body will include a JSON object consisting of the following fields:

- **token_type** (string) – This will be set to bearer
- **access_token** (string) – The access token to be used for subsequent API calls

Example JSON request:

```
HTTP Endpoint: https://apilab.iinconnect.com/token
HTTP Method: POST
Header:
      Content-Type: application/x-www-form-urlencoded
Body:
      grant_type: client_credentials
      client_id: <client_id>
      client_secret: <client_secret>
```

Example JSON response:

```
200 OK
Content-Type: application/json
{
    "token_type": "bearer",
    "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldU…"
}
```

2.2 Message Encryption and Signing

This section contains details of the steps that client applications need to perform for encryption/decryption and signing/verification of payload messages when sending and receiving messages to/from the Gateway Communication Services.

Sending Message to Gateway

A. Generate a symmetric key for each message using AES-256.
B. Encrypt the data to be transmitted:
   a. Encrypt data with the symmetric key.
   b. Encode the encrypted data using Base64.
   c. Add the encoded, encrypted data to “data” tag of payload.
C. Encrypt the secret key used:
   a. Symmetric key will be encoded using BASE64.
   b. Encrypt the encoded symmetric key with RSA public key of Coin Network.
   c. Encrypted key will be encoded with BASE64.
   d. Add the encoded, encrypted key to “secret” tag of payload.
D. Generate Hash of complete “message” object using SHA256. Use RSA private key to sign the hash of request JSON string to generate the signature value.
E. The signature value will be encoded using BASE64 and added to the “signature” tag in the payload.
F. Send the generic payload to the payment network.

Sample Java Code highlighting implementation of the steps above:

```
//Generate AES-256 Symmetric key
KeyGenerator keyGen = KeyGenerator.getInstance("AES");
keyGen.init(256);
SecretKey secretKey = keyGen.generateKey();

//Encrypt the raw body with AES secret key
Cipher cipher = Cipher.getInstance(“AES/GCM/NoPadding”);
cipher.init(Cipher.ENCRYPT_MODE, secretKey, new GCMParameterSpec(128, new byte[96]));
byte[] encryptedData = cipher.doFinal(dataToEncrypt.getBytes(“UTF-8”));

//Encode the encrypted text with Base64
String encodedEncryptedData = Base64.encodeBase64String(encryptedData);

//Encrypt the secretKey using network’s publickey as per steps below
//First encode the secret key using Base64
String encodedSecretKey = Base64.getEncoder().encodeToString(secretKey.getEncoded())

//Encrypt encoded secretKey using network’s publickey using RSA with below padding
cipher = Cipher.getInstance("RSA/ECB/OAEPWithSHA-256AndMGF1Padding");
cipher.init(Cipher.ENCRYPT_MODE, key, new OAEPParameterSpec("SHA-256", "MGF1",
MGF1ParameterSpec.SHA256, PSource.PSpecified.DEFAULT));
byte [] encryptedSecretKey = cipher.doFinal(encodedSecretKey.getBytes(“UTF-8”))

//Finally, encode the encrypted secretKey using Base64
String encodedEncryptedSecretKey = Base64.encodeBase64String(encryptedText)

//Set the encryptedBody against the ”data” property of the payload and encryptedSecret with
the “secret” property of the payload
messageBody.setData(encodedEncryptedData);
messageBody.setSecret(encodedEncryptedSecretKey);

Generating message signature
// Use RSA private key of the sender to sign the contents of the message object
Signature signature = Signature.getInstance("SHA256withRSA");
signature.initSign(key);
signature.update(message.getBytes(“UTF-8”));
byte[] signed = signature.sign();
String signature = new String(Base64.encodeBase64(signed));
```

Receiving Message from Gateway

A. Verify the signature received in the payload:
   a. Decode the “signature” value using BASE64 decoder
   b. Use the RSA public key of sender system to verify the “signature” value by comparing the hash value of the received “message” object

B. Decrypt the encrypted symmetric key from “secret” tag of generic payload with RSA private key after using BASE64 decoder. Decode the decrypted key again using Base64.

C. Decode the data from “data” tag with Base64 and then decrypt it with the decrypted AES-256 symmetric key (from step B).

D. Process the payload data.

Sample Java Code highlighting implementation of the steps above:

```
//Verify the signature of the response payload by using network’s public key
Signature signature = Signature.getInstance("SHA256withRSA");
signature.initVerify(networkPublicKey);
signature.update(payload.getBytes(“UTF-8”));
boolean isValid = signature.verify(Base64.decodeBase64(signature.getBytes()));

Payload decryption

//First decrypt secret key with receiver’s private key
byte[] decodedEncryptedSecretKey = Base64.decodeBase64(encryptedSecret);
Cipher cipher = Cipher.getInstance("RSA/ECB/OAEPWithSHA-256AndMGF1Padding")
cipher.init(Cipher.DECRYPT_MODE, privateKey, new OAEPParameterSpec("SHA-256", "MGF1",
MGF1ParameterSpec.SHA256, PSource.PSpecified.DEFAULT));
String decryptedSecretKey = new String(cipher.doFinal(decodedBytes), “UTF-8”)

//Finally, decode the decryptedKey
byte[] decodedSecretKey = Base64.getDecoder().decode(decryptedSecretKey);

//Decrypt the actual response payload
//Recreate the SecretKey object from the decoded key above
SecretKey secretKey = new SecretKeySpec(decodedSecretKey, 0, decodedKey.length, "AES");

//Decrypt the data part of the payload with AES secret key
Cipher cipher = Cipher.getInstance(“AES/GCM/NoPadding”);
cipher.init(Cipher.DECRYPT_MODE, secretKey, new GCMParameterSpec(128, new byte[96]));
byte[] decodedEncryptedData = Base64.decodeBase64(encryptedData);
String decryptedData = new String(cipher.doFinal(decodedEncryptedData), "UTF-8");
```

2.3 Send Message

POST /gateway/send/(string: system_id)

Send a message to the destination blockchain network.

Parameters:

- **system_id** (string) – System ID of message sender. This should be same as the systemId populated in the payload header.

Request JSON Object:

As per the generic message payload format defined in Coin Messages section. The message will be routed to the destination as per the fields populated in the payload header tag.

Response Status:

- 2xx – Success
- 403 – Unauthorized to send message to gateway
- 4xx – Request Error
- 5xx – System Error

Example JSON request:

```
POST /gateway/send/UBIN001
Host: network.example.com
Content-Type: application/json
Accept: application/json
Authorization: Bearer <access-token>

{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "MINT.INIT",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

2.4 Poll New Messages

GET /gateway/poll/(string: system_id)

Get new messages from blockchain networks. This call will return all pending messages for the system_id provided.

Parameters:

- **system_id** (string) – System ID of message sender.

Response JSON Array of Objects:

As per the generic message payload format defined in Coin Messages section.

Response Status:

- 2xx – Success
- 403 – Unauthorized to poll messages from gateway
- 4xx – Request Error
- 5xx – System Error

Example JSON request:

```
GET /gateway/poll/UBIN001
Host: network.example.com
Accept: application/json
Authorization: Bearer <access-token>
```

Example JSON response:

```
200 OK
Content-Type: application/json
[
  {
    "message": {
      "header": {
        "version": "1.0",
        "messageId": "RFP001010",
        "sendTimestamp": "2018-04-23T18:25:43.511Z",
        "systemId": "UBIN001",
        "messageType": "MINT.NOTIFY",
        "assetType": "CASH",
        "platform": "QUORUM",
        "network": "UBIN"
      },
      "body": {
        "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
        "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
      }
    },
    "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
  },
  {
    "message": {
      "header": {
        "version": "1.0",
        "messageId": "RFP001011",
        "sendTimestamp": "2018-04-23T18:25:43.511Z",
        "systemId": "UBIN001",
        "messageType": "TRANSFER.NOTIFY",
        "assetType": "CASH",
        "platform": "QUORUM",
        "network": "UBIN",
        "reserve": {}
      },
      "body": {
        "data": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…",
        "secret": "Kr46wD4SV5o6fpH83i9rAwpz=…"
      }
    },
    "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
  }
]
```

3 APIs for Interfacing with Payment Network

3.1 Generic Message Payload

All messages should be sent as per the generic message payload structure defined below. The message format used is JSON.

The generic payload consists of three sections:

a) **<u>Header</u>** – This contains the message metadata that is used to track the message. This fields in this section are common for all messages.
b) **<u>Body</u>** – This contains the encrypted data to be submitted to the blockchain as well as the encrypted one-time secret key to be used to decrypt the data. The data that is encrypted differs based on the message type.
c) **<u>Signature</u>** – This contains the signature of the message body that is used by the receiver to verify the authenticity and integrity of the data that has been transferred.

```
{
  "message": {
    "header": {…},
    "body": {…},
  "signature": "…"
}
```

3.1.1 Header

The message header is a JSON object consisting of the following fields. This header is intended to be used for all types of messages for various use cases.

Field	Type	Required	Description
version	String	M	API version number
messageId	String	M	Unique Message Identifier
sendTimestamp	String	M	Message Date / Timestamp in UTC format
systemId	String	M	Client / External System Identifier
messageType	String	M	Type of message
assetType	String	M	Type of asset in blockchain - CASH
platform	String	M	Protocol of the blockchain network - QUORUM
network	String	M	Destination blockchain network - UBIN

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "MINT.INIT",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {…},
  "signature": "…"
}
```

3.1.2 Body

The body consists of two fields – ‘data’ and ‘secret’. This section is used to send encrypted data to the blockchain adapter as per the message type.

Field	Type	Required	Description
data	String	M	The encrypted object of message attributes.
secret	String	M	The encrypted value of the symmetric key that is used to encrypt the data.

```
{
  "message": {
    "header": {…},
    "body": {
      "data":
"Kr46wD4SV5o6fpH83i9rAwpzR/vpMZAYlkeQphOOqI4pJdmRrSTmcq3cPdfgwWYoUQc0rAX06v00UDaYKOrMThVBS
7MRBElMRqzUZ+CNxl7TN5W/8ZlRbQ9hj/yRqvRQk1iNO3BF0bjSZSU2J4rrsXZj1XOSVuENGA5CD8r6CCJs2JVDw6a
mxo8=",
      "secret":
"f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuynVFbCt6Ksyj8KlxOLZ3gasAPRAYlSSC0+xaY1OZmB8TL9Av+QqnSjymEUqL
utmrp9WVsGJ/+bpyN8o5Xm7bupqEjZSccA8BQb3gjRfqvDSxrev1tpMfbuAJkgkJ8gSBhpWx2GG5yQ=="
    }
  },
  "signature": "…"
}
```

3.1.3 Signature

This section is used by the recipient to ensure that the message received has been sent by the correct sender system (authenticity) and that the data has not been tampered in transit (integrity).

Field	Type	Required	Description
signature	String	M	The encrypted hash of the message object used to ensure data integrity.

```
{
  "message": {
    "header": {…},
    "body": {…}
  },
  "signature":
"lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5hRHmWSarCOo5/3KC4uWPkc/aVESkwsSQ89IXDV4b6OWH4uYZt6zA
bmUrSMCkAhG/2bRfyfxVEH5GekRorphsJSFtxys+4ciay78dmi8E8R75ygDnTjunZd5jQBlsbfdxaYki9Lap8HIH9p
rob3XSF4QGo0lJyjPBGGP7EQpkIZZaHc2xWE4JPftuJj7Ro/K7WVOcb8ccnxddAL9RvI72kTPxifbxKWLyyU0oxMyp
K5EPINFve1EHEFSlNRxwpgHpc1HaJiD5ylKlX8DYKQiT2AHwn/Olr8KdMBlUMWGfGpjzSn84g=="
}
```

3.2 Generic Technical Failure Payload

In the case of technical failure, a generic technical failure payload is sent to source system or network operator for further investigation. The distinction between generic technical failure payload and a normal error payload is that the latter is part of normal business flow.

3.2.1 ERROR.NOTIFY

The following is the technical failure payload definition.

Field	Type	Required	Description
requestId	String	M	The message ID of the original message.
error	Object	M	Consists of error code and description fields.
code	String	M	Identification code for the error that has occurred
description	String	M	Description of the error that has occurred

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ERROR.NOTIFY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The "data" field in "body" contains the encrypted value of below object:

{
  "requestId": "RFP001009",
  "error":{
     "code": "000010",
     "description": "Blockchain Connection Timeout"
   }
}
```

3.3 Coin Messages

The following are the different Coin specific messages that can be sent from external systems to payment network and vice versa. The fields specified in this section are encrypted and sent as part of the 'data' tag in the 'body' section of the generic payload structure defined above.

3.3.1 Overview

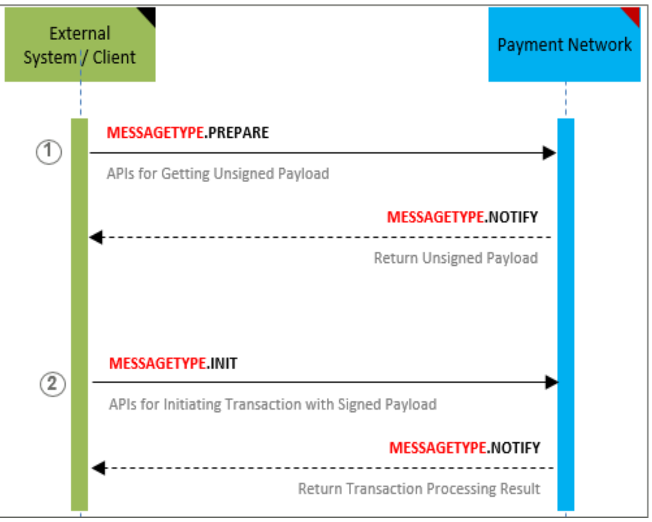
There are two models for clients to submit transactions to the Payment Network depending on which system is the custodian of the blockchain account private keys:

- **Model 1: External System as Custodian of private keys**:

Custodian of Private Key : External System / Client

When External System is the custodian of Private Key for Blockchain account then 2 step process to be followed.

- Step-1 : Send PREPARE API with required information to be called for getting the Unsigned transaction from Payment Network.
- Step-2 : External System will Sign the Unsigned transaction (received as part of PREPARE API result in Step-1) with Blockchain account Private Key. Send INIT API with Signed transaction to initiate the processing in Payment Network.

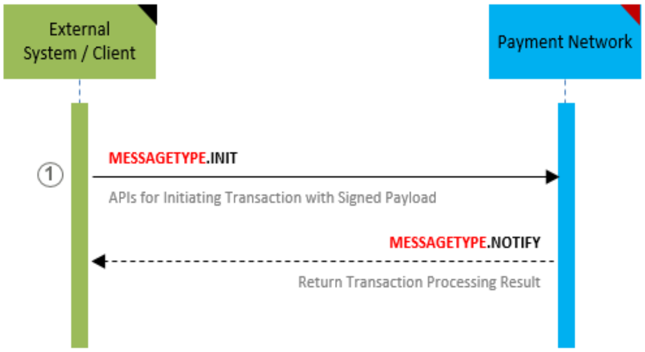


- **Model 2: Payment Network as Custodian of private keys:**

Custodian of Private Key : Payment Network

When Payment Network is the custodian of Private Key for Blockchain account for external system / Client then following step to be followed.

- Step-1 : External System / Client will send the INIT API with required information to initiate the transaction in Payment Network.


External System / Client
Payment Network
1
MESSAGETYPE.INIT
APIs for Initiating Transaction with Signed Payload
MESSAGETYPE.NOTIFY
Return Transaction Processing Result

3.3.2 REDEEM.INIT.PREPARE

To create unsigned transaction for initiating Token Redemption request. The requestor needs to sign the unsigned transaction and submit it back.

NOTE: This message would only be applicable when external systems are the custodians of the blockchain account private keys.

Field	Type	Required	Description
referenceNo	String	M	The source system reference used to identify the token redemption request
blockchainId	String	M	The blockchain address from which the tokens are to be redeemed
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens to be redeemed
value	Number	M	The number of tokens to be redeemed

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "REDEEM.INIT.PREPARE",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:

{
  "referenceNo": "QUO02020123",
  "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

3.3.3 REDEEM.INIT

This message is used to initiate a redemption of tokens from an accounts on the blockchain. When funds are transferred in the accounting system from blockchain account to client DDA account, then external systems will need to send this message to burn the same number of tokens in the blockchain.

Field	Type	Required	Description
referenceNo	String	M	The source system reference used to identify the token redemption request
blockchainId	String	M	The blockchain address from which the tokens are to be redeemed
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens to be redeemed
value	Number	M	The number of tokens to be redeemed
txData	Object	O	Only applicable for model 1
signedHex	String	O	Signed Transaction in HEX format

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "REDEEM.INIT",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below is a sample message content for model 1:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "txData": {
    "signedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
}
```

In case of model 2, the message content would be as below:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

3.3.4 REDEEM.NOTIFY

This message is used to notify the completion/failure of a token redemption request on the blockchain.

Field	Type	Required	Description
referenceNo	String	M	The reference used to identify the redemption event on the blockchain
requestId	String	O	The request message ID of the REDEEM.INIT message sent to initiate token redemption. This field is optional.
status	String	M	The status of the redemption transaction – SUCCESS, PENDING, QUEUED, REJECTED or ERROR
blockchainId	String	M	The blockchain address from which the tokens have been redeemed
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens that have been redeemed
value	Number	M	The number of tokens that have been redeemed
error	Object	O	Consists of error code and description fields (if status is ERROR)
code	String	O	Identification code for the error that has occurred
description	String	O	Description of the error that has occurred
txData	Object	O	If object is set, then to be offline signed. Only applicable for model 1.
unsignedHex	String	O	Unsigned Transaction to be signed by client (HEX format)

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "REDEEM.NOTIFY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below sample response is sent after REDEEM.INIT.PREPARE in model 1:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "requestId": "RFP001011",
  "status": "PENDING",
  "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "txData": {
    "unsignedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
}
```

Following sample response is sent after REDEEM.INIT for both models:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "requestId": "RFP001010",
  "status": "PENDING",
  "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

Below is an example of an error message:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "requestId": "RFP001010",
  "status": "ERROR",
  "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "error":{
    "code": "000010",
    "description": "Invalid address"
  }
}
```

3.3.5 TRANSFER.PREPARE

To create unsigned transaction for initiating Transfer request between two accounts. The requestor needs to sign the unsigned transaction and submit it back.

NOTE: This message would only be applicable when external systems are the custodians of the blockchain account private keys.

Field	Type	Required	Description
referenceNo	String	M	The source system reference used to identify the token transfer request
senderId	String	M	The blockchain address from which the tokens are to be transferred
receiverId	String	M	The blockchain address to which the tokens are to be transferred
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens to be minted
value	Number	M	The number of tokens to be minted

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "TRANSFER.PREPARE",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:

{
  "referenceNo": "QUO02020123",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

3.3.6 TRANSFER.INIT

This message is used to initiate a transfer of tokens between two accounts on the blockchain.

Field	Type	Required	Description
referenceNo	String	M	The source system reference used to identify the token transfer request
senderId	String	M	The blockchain address from which the tokens are to be transferred
receiverId	String	M	The blockchain address to which the tokens are to be transferred
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens to be minted
value	Number	M	The number of tokens to be minted
txData	Object	O	Only applicable for model 1.
signedHex	String	O	Signed Transaction in HEX format

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001010",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "TRANSFER.INIT",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below is a sample message content for model 1:

```
The “data” field in “body” contains the encrypted value of below object:

{
  "referenceNo": "QUO02020123",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "txData": {
    "signedHex": "0xf00asdfasdfasdf00asdfasdf"
  }
}
```

In case of model 2, the message content would be as below:

```
The “data” field in “body” contains the encrypted value of below object:

{
  "referenceNo": "QUO02020123",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

3.3.7 TRANSFER.NOTIFY

This message is used to notify the client systems regarding a token transfer event. The message is sent by the Payment Network when the token transfer is committed on the blockchain.

Field	Type	Required	Description
referenceNo	String	M	The reference used to identify the transfer event on the blockchain
requestId	String	O	The request message ID of the TRANSFER.INIT message sent to initiate token transfer. This field is optional.
status	String	M	The status of the transfer transaction – SUCCESS, QUEUED or ERROR
senderId	String	M	The blockchain address from which the tokens have been transferred
receiverId	String	M	The blockchain address to which the tokens have been transferred
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens that have been transferred
value	Number	M	The number of tokens that have been transferred
error	Object	O	Consists of error code and description fields (if status is ERROR)
code	String	O	Identification code for the error that has occurred
description	String	O	Description of the error that has occurred
txData	Object	O	If object is set, then to be offline signed. Only applicable for model 1.
unsignedHex	String	O	Unsigned Transaction to be signed by client (HEX format)

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "TRANSFER.NOTIFY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below is a sample response sent after TRANSFER.PREPARE in model 1:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "txData": {
    "unsignedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
}
```

Following response is sent after TRANSFER.INIT for both models:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

Below is an example of an error message:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "QUO02020123",
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "error":{
    "code": "000010",
    "description": "Invalid address"
  }
}
```

3.3.8 BALANCE.ENQUIRY

This message is used to get the current blockchain coin balance. This can be used to retrieve balance of the specified blockchain addresses of client system or to perform client level reconciliation.

Field	Type	Required	Description
blockchainIds	Array	O	An array of blockchain addresses of which balance is retrieved; If no addresses are specified, balance of all client's addresses will be retrieved.
currency	String	M	The currency of the token whose balance is to be retrieved

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "BALANCE.ENQUIRY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:
{
  "blockchainIds": ["0xabcdefghijklmnopqrstuvwxyz012345678900",
"0xabcdefghijklmnopqrstuvwxyz012345678901"],
  "currency": "USD"
}
```

Below is an example of client reconciliation with no blockchain addresses specified:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "currency": "USD"
}
```

3.3.9 BALANCE.NOTIFY

This message is used to provide the current blockchain coin balance for the address and currency specified in the corresponding request message.

Field	Type	Required	Description
requestId	String	M	The request message ID of the BALANCE.ENQUIRY message sent to query balance.
balanceList	Array	M	Request deatils
status	String	M	The status of the balance request for each blockchainId – SUCCESS or ERROR
blockchainId	String	M	The blockchain address whose balance is retrieved
amount	Object	M	
currency	String	M	The currency of the tokens whose balance is to be retrieved
value	Number	M	The token balance for the specified address
error	Object	O	Consists of error code and description (if status is ERROR)
code	String	O	Identification code for the error that has occurred
description	String	O	Description of the error that has occurred

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "BALANCE.NOTIFY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "1DuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "balanceList": [
    {
      "status": "SUCCESS",
      "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
      "amount": {
        "currency": "USD",
        "value": 28887.88
      }
    },
    {
      "status": "SUCCESS",
      "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678901",
      "amount": {
        "currency": "USD",
        "value": 200.0
      }
    }
  ]
}
```

Below is an example of an error message:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "balanceList": [
    {
      "status": "SUCCESS",
      "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
      "amount": {
        "currency": "USD",
        "value": 28887.88
      }
    },
    {
      "status": "ERROR",
      "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678910",
      "amount": {
        "currency": "USD",
        "value": null
      },
      "error": {
        "code": "0004",
        "description": "Failed to retrieve balance"
      }
    }
  ]
}
```

3.3.10 TRANSACTION.ENQUIRY

This message is used to get the transaction history. This can be used to retrieve the transaction history of the specified blockchain addresses of client system or to perform client level reconciliation.

Field	Type	Required	Description
blockchainIds	Array	O	An array of blockchain addresses of which transaction history is retrieved. If no addresses are specified, transaction history of all client's addresses will be retrieved.
currency	String	M	The currency of the token whose transaction history is to be retrieved
startTimestamp	Number	O	Unix timestamp (in seconds) of start of interval for which transaction history is required. This field is optional.
endTimestamp	Number	O	Unix timestamp (in seconds) of end of interval for which transaction history is required. This field is optional.

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "TRANSACTION.ENQUIRY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The "data" field in "body" contains the encrypted value of below object:
{
  "blockchainIds": ["0xabcdefghijklmnopqrstuvwxyz012345678900",
"0xabcdefghijklmnopqrstuvwxyz012345678901"],
  "currency": "USD",
  "startTimestamp": 1564730422,
  "endTimestamp": 1564735167
}
```

3.3.11 TRANSACTION.NOTIFY

This message is used to provide the transaction history for the address and currency specified in the corresponding request message.

Field	Type	Required	Description
requestId	String	M	The request message ID of the TRANSACTION.ENQUIRY message sent to query balance.
currency	String	M	The currency of the tokens whose transaction history is to be retrieved
startTimestamp	Number	O	Unix timestamp (in seconds) of start of interval.
endTimestamp	Number	O	Unix timestamp (in seconds) of end of interval.
transactionList	Array	M	
status	String	M	The status of the transaction query request for each blockchain address– SUCCESS or ERROR
blockchainId	String	M	The blockchain address whose transaction history is retrieved
transactions	Array	M	List of transactions as per the fields specified in the request message for the blockchain address. Empty array if no matching transactions are found.
transactionRef	String	O	The reference used to identify the transaction on the blockchain
sourceRef	String	O	The source system reference used to identify the transaction
senderId	String	O	The blockchain address from which the tokens were transferred
receiverId	String	O	The blockchain address to which the tokens were transferred
value	Number	O	The number of tokens that were transferred
creationTimestamp	String	O	Transaction creation date / timestamp in UTC format
updateTimestamp	String	O	Transaction last update date / timestamp in UTC format
transactionStatus	String	O	Transaction status in blockchain
type	String	O	Type of transaction – MINT, TRANSFER and REDEEM
error	Object	O	Consists of error code and description (if status is ERROR)
code	String	O	Identification code for the error that has occurred
description	String	O	Description of the error that has occurred

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "TRANSACTION.NOTIFY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

The “data” field in “body” contains the encrypted value of below object:

```
{
  "requestId": "RFP001011",
  "currency": "USD",
  "startTimestamp": 1564730422,
  "endTimestamp": 1564735167,
  "transactionList": [
    {
      "status": "SUCCESS",
      "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
      "transactions": [
        {
          "transactionRef": "005DE8A82F000000",
          "sourceRef": "005DE8A82F000000",
          "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
          "receiverId": "0xabcdefghijklmnopqrstuvwxyz012345678901",
          "value": 20.0,
          "creationTimestamp": "2019-12-05T06:48:15.807Z",
          "updateTimestamp": "2019-12-05T06:48:15.807Z",
          "transactionStatus": "SUCCESS",
          "type": "TRANSFER"
        }
      ]
    },
    {
      "status": "SUCCESS",
      "blockchainId": "0xabcdefghijklmnopqrstuvwxyz012345678901",
      "transactions": [
        {
          "transactionRef": "005DE8A82F000000",
          "sourceRef": "005DE8A82F000000",
          "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
          "receiverId": "0xabcdefghijklmnopqrstuvwxyz012345678901",
          "value": 20.0,
          "creationTimestamp": "2019-12-05T06:48:15.807Z",
          "updateTimestamp": "2019-12-05T06:48:15.807Z",
          "transactionStatus": "SUCCESS",
          "type": "TRANSFER"
        }
      ]
    }
  ]
}
```

Below is an example of an error message:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "currency": "USD",
  "startTimestamp": 1564730422,
  "endTimestamp": 1564735167,
  "transactionList": [
    {
      "status": "ERROR",
      "blockchainId": "0xsddaffffffffffffffffffffffffffffffffff",
      "transactions":null,
      "error": {
        "code": "0003",
        "description": "Blockchain address not allowed"
      }
    }
  ]
}
```

3.3.12 ESCROW.INIT.PREPARE

This message is used to create an unsigned transaction to initiate a new Escrow transaction between the sender and receiver addresses. The requestor needs to sign the unsigned transaction using the sender address' private key and submit it back to the payment network.

NOTE: This message would only be applicable when external systems are the custodians of the blockchain account private keys.

Field	**Type**	**Required**	**Description**
referenceNo	String	M	The source system reference number used to identify the transaction
senderId	String	M	The blockchain address from which tokens are to be transferred
receiverId	String	M	The blockchain address to which tokens are to be transferred
agentId	String	M	The blockchain address of the escrow agent for this transaction. NOTE: This should always be included as one of the signers.
signers	Array	M	Array of blockchain addresses authorized to sign token release from the escrow account
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens that are to be transferred
value	Number	M	The number of tokens that are to be transferred

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ESCROW.INIT.PREPARE",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

The “data” field in “body” contains the encrypted value of below object:

```
{
  "referenceNo": "CLIENTREF001",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
  "signers": [
         "0xabcdefghijklmnopqrstuvwxyz012345678900",
         "0xzyxwvutsrqponmlkjihgfedcba987654321000",
         "0x9aae0514d65594a368effef94e0de04209091740"
  ],
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

3.3.13 ESCROW.INIT

This message is used to initiate a new Escrow transaction between the sender and receiver addresses. Based on this message, the specified amount will be locked in an Escrow account.

Field	Type	Required	Description
referenceNo	String	M	The source system reference number used to identify the transaction
senderId	String	M	The blockchain address from which tokens are to be transferred
receiverId	String	M	The blockchain address to which tokens are to be transferred
agentId	String	M	The blockchain address of the escrow agent for this transaction. NOTE: This should always be included as one of the signers.
signers	Array	M	Array of blockchain addresses authorized to sign token release from the escrow account
amount	Object	M	Consists of currency and value fields
currency	String	M	The currency of the tokens that are to be transferred
value	Number	M	The number of tokens that are to be transferred
txData	Object	O	Only applicable for model 1.
signedHex	String	O	Signed Transaction in HEX format

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ESCROW.INIT",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below is a sample message content for model 1:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "CLIENTREF001",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
  "signers": [
         "0xabcdefghijklmnopqrstuvwxyz012345678900",
         "0xzyxwvutsrqponmlkjihgfedcba987654321000",
         "0x9aae0514d65594a368effef94e0de04209091740"
  ],
  "amount": {
    "currency": "USD",
    "value": 40003
  },
  "txData": {
    "signedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
}
```

In case of model 2, the message content would be as below:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "referenceNo": "CLIENTREF001",
  "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
  "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
  "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
  "signers": [
         "0xabcdefghijklmnopqrstuvwxyz012345678900",
         "0xzyxwvutsrqponmlkjihgfedcba987654321000",
         "0x9aae0514d65594a368effef94e0de04209091740"
  ],
  "amount": {
    "currency": "USD",
    "value": 40003
  }
}
```

3.3.14 ESCROW.SIGN.PREPARE

This message is used to create an unsigned transaction to specify an action for an Escrow transaction. The requestor needs to sign the unsigned transaction using the respective signer's private key and submit it back to the payment network.

NOTE: This message would only be applicable when external systems are the custodians of the blockchain account private keys.

Field	Type	Required	Description
escrowId	String	M	The reference number used to identify the escrow transaction
signerId	String	M	The blockchain address to be used to sign this escrow transaction with one of the below action types.
actionType	String	M	This can be one of the following three values – RELEASE, REVERT or DISPUTE - RELEASE: This is used to release an Escrow transaction that has previously been initiated. Based on this, participants will indicate that they agree to proceed with this escrow transaction. The transaction amount locked in an Escrow account will be transferred to the receiver address ONLY IF the majority of the signers have sent this message. - REVERT: This is used to revert an Escrow transaction that has previously been initiated. Based on this message, participants will indicate that they agree to revert/cancel this escrow transaction. The transaction amount locked in an Escrow account will be transferred back to the sender address ONLY IF the majority of the signers have sent this message. - DISPUTE: This is used to mark an escrow transaction as disputed. Based on this message, the participants can indicate that they do not agree with the escrow conditions. The escrow agent can then review and settle the dispute offline and release or revert the locked funds by sending the respective message as defined earlier.

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ESCROW.SIGN.PREPARE",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:
{
  "escrowId": "2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824",
  "signerId": "0xzyxwvutsrqponmlkjihgfedcba987654321000"",
  "actionType": "RELEASE"
}
```

3.3.15 ESCROW.SIGN

This message is used to sign an Escrow transaction with a particular action. Based on this message, participants will indicate that they agree to release/revert/dispute this escrow transaction.

Field	Type	Required	Description
escrowId	String	M	The reference number used to identify the escrow transaction
signerId	String	M	The blockchain address to be used to sign this escrow transaction with one of the below action types.
actionType	String	M	This can be one of the following three values – RELEASE, REVERT or DISPUTE - RELEASE: This is used to release an Escrow transaction that has previously been initiated. Based on this, participants will indicate that they agree to proceed with this escrow transaction. The transaction amount locked in an Escrow account will be transferred to the receiver address ONLY IF the majority of the signers have sent this message. - REVERT: This is used to revert an Escrow transaction that has previously been initiated. Based on this message, participants will indicate that they agree to revert/cancel this escrow transaction. The transaction amount locked in an Escrow account will be transferred back to the sender address ONLY IF the majority of the signers have sent this message. - DISPUTE: This is used to mark an escrow transaction as disputed. Based on this message, the participants can indicate that they do not agree with the escrow conditions. The escrow agent can then review and settle the dispute offline and release or revert the locked funds by sending the respective message as defined earlier.
txData	Object	O	Only applicable for model 1.
signedHex	String	O	Signed Transaction in HEX format

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ESCROW.SIGN",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below is a sample message content for model 1:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "escrowId": "2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824",
  "signerId": "0xzyxwvutsrqponmlkjihgfedcba987654321000"",
  "actionType": "RELEASE",
  "txData": {
    "signedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
}
```

In case of model 2, the message content would be as below:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "escrowId": "2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824",
  "signerId": "0xzyxwvutsrqponmlkjihgfedcba987654321000"",
  "actionType": "RELEASE"
}
```

3.3.16 ESCROW.ENQUIRY

This message is to enquire regarding the status of an Escrow transaction that has previously been initiated/released/reverted/disputed.

Field	Type	Required	Description
escrowId	String	M	The reference number used to identify the escrow transaction

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001011",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ESCROW.ENQUIRY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:
{
  "escrowId": "2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824"
}
```

3.3.17 ESCROW.NOTIFY

This message is used to notify the external systems regarding the status and details of an Escrow transaction based on the escrow ID provided in the corresponding request message. This message is sent for each escrow action message sent by the participants.

Field	**Type**	**Required**	**Description**
requestId	String	M	The request message ID of the ESCROW.INIT, ESCROW.SIGN or ESCROW.ENQUIRY message sent to the blockchain
status	String	M	The status of this transaction query request – SUCCESS, ERROR or QUEUED
escrow	Object	M	Start tag for escrow details. Empty if no matching escrows are found.
escrowId	String	O	The reference number used to identify the escrow transaction
referenceNo	String	O	The source system reference number provided by the sender system
senderId	String	O	The blockchain address from which tokens are to be transferred
receiverId	String	O	The blockchain address to which tokens are to be transferred
agentId	String	O	The blockchain address of the escrow agent for this transaction
signers	Array	O	Array of blockchain addresses authorized to sign token release/revert from the escrow account
amount	Object	O	Consists of currency and value fields
currency	String	O	The currency of the tokens that are to be transferred
value	Number	O	The number of tokens that are to be transferred
signStatus	Object	O	Start tag for object for signing status
release	Array	O	Array of blockchain addresses that have signed to release escrow funds to receiver.
revert	Array	O	Array of blockchain addresses that have signed to revert escrow funds back to sender
dispute	Array	O	Array of blockchain addresses that have signed to mark the transaction as disputed
escrowStatus	String	O	Status of the escrow transaction (INITIATED, RELEASED, REVERTED, DISPUTED, ERROR)
isDisputed	Boolean	O	Flag to indicate whether the transaction was disputed in its lifecycle
error	Object	O	Consists of error code and description fields (if escrowStatus is ERROR)
code	String	O	Identification code for the error that has occurred
description	String	O	Description of the error that has occurred
txData	Object	O	If object is set, then to be offline signed. Only applicable for model 1.
unsignedHex	String	O	Unsigned Transaction to be signed by client (HEX format)

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001012",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "messageType": "ESCROW.NOTIFY",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below is a sample response sent after ESCROW.INIT.PREPARE in model 1:

```
The “data” field in “body” contains the encrypted value of below object:

{
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "escrow": {
    "referenceNo": "CLIENTREF001",
    "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
    "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
    "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
    "amount": {
      "currency": "USD",
      "value": 40003
    },
    "signStatus": {},
    "isDisputed": false
  },
  "txData": {
    "unsignedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
}
```

Below is a sample response sent after ESCROW.SIGN.PREPARE in model 1:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "escrow": {
    "escrowId": "ABCDEFGHIJB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824",
    "referenceNo": "CLIENTREF001",
    "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
    "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
    "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
    "signers": [
          "0xabcdefghijklmnopqrstuvwxyz012345678900",
          "0xzyxwvutsrqponmlkjihgfedcba987654321000",
          "0x9aae0514d65594a368effef94e0de04209091740"
    ],
    "amount": {
      "currency": "USD",
      "value": 40003
    },
    "signStatus": {
      "release": [],
      "revert": [],
      "dispute": [],
    },
    "escrowStatus": "INITIATED”,
    "isDisputed": false
  },
  "txData": {
    "unsignedHex": "0xf00asdfasdfasdf00asdfasdasdff"
  }
```

Following response is sent after ESCROW.INIT for both models:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "escrow": {
    "escrowId": "ABCDEFGHIJB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824",
    "referenceNo": "CLIENTREF001",
    "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
    "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
    "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
    "signers": [
          "0xabcdefghijklmnopqrstuvwxyz012345678900",
          "0xzyxwvutsrqponmlkjihgfedcba987654321000",
          "0x9aae0514d65594a368effef94e0de04209091740"
    ],
    "amount": {
      "currency": "USD",
      "value": 40003
    },
    "signStatus": {
      "release": [],
      "revert": [],
      "dispute": []
    },
    "escrowStatus": "INITIATED",
    "isDisputed": false
  }
}
```

Following response is sent after ESCROW.SIGN and ESCROW.ENQUIRY for both models:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "status": "SUCCESS",
  "escrow": {
    "escrowId": "ABCDEFGHIJB0A30E26E83B2AC5B9E29E1B161E5C1FA7425E73043362938B9824",
    "referenceNo": "CLIENTREF001",
    "senderId": "0xabcdefghijklmnopqrstuvwxyz012345678900",
    "receiverId": "0xzyxwvutsrqponmlkjihgfedcba987654321000",
    "agentId": "0x9aae0514d65594a368effef94e0de04209091740",
    "signers": [
         "0xabcdefghijklmnopqrstuvwxyz012345678900",
         "0xzyxwvutsrqponmlkjihgfedcba987654321000",
         "0x9aae0514d65594a368effef94e0de04209091740"
    ],
    "amount": {
      "currency": "USD",
      "value": 40003
    },
    "signStatus": {
      "release": [
         "0xabcdefghijklmnopqrstuvwxyz012345678900",
         "0x9aae0514d65594a368effef94e0de04209091740"
      ],
      "revert": ["0xzyxwvutsrqponmlkjihgfedcba987654321000"],
      "dispute": ["0xzyxwvutsrqponmlkjihgfedcba987654321000"],
    },
    "escrowStatus": "RELEASED”,
    "isDisputed": true
  }
}
```

Below is an example of an error message:

```
The “data” field in “body” contains the encrypted value of below object:
{
  "requestId": "RFP001011",
  "status": "ERROR",
  "escrow": {
    "escrowId": "667F07885AB45AE46410C4220DCB7A8F690F1DA83CC521F521B22BF3BA689A4D",
    "escrowStatus": "ERROR"
  },
  "error": {
    "code": "0004",
    "description": "Escrow Not Found"
  }
}
```

3.3.18 ONBOARDING.ENQUIRY

3.3.18.1 ACCOUNT.VERIFY

This message is used to verify if an account currency combination exists.

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001012",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

```
The “data” field in “body” contains the encrypted value of below object:
{
    "accounts": [
      {
        "blockchainId": "0xca843569e3427144cead5e4d5999a3d0ccf92b8e",
        "currency": "USD"
      }
     ],
     "enquiryType": "ACCOUNT.VERIFY",
     "messageType": "ONBOARDING.ENQUIRY"
}
```

3.3.19 ONBOARDING.NOTIFY

This message is used to notify the status of requested Onboarding Enquiry. If Enquiry is of ACCOUNT.VERIFY, response will indicate if each account currency combination is valid.

```
{
  "message": {
    "header": {
      "version": "1.0",
      "messageId": "RFP001012",
      "sendTimestamp": "2018-04-23T18:25:43.511Z",
      "systemId": "UBIN001",
      "assetType": "CASH",
      "platform": "QUORUM",
      "network": "UBIN"
    },
    "body": {
      "data": "Kr46wD4SV5o6fpH83i9rAwpz=…",
      "secret": "f2nEOpxOgFnR9ubQ3KBPZ3heXyWDuy==…"
    }
  },
  "signature": "lDuvJo2SsOkdWNi2dJ0dE5v/6TmSx2sXDqoG5Olr8KdMBlUMWGfGpjzSn84g=="
}
```

Below response is received if enquiryType = ACCOUNT.VERIFY

```
The “data” field in “body” contains the encrypted value of below object:
[
    {
        "blockchainId": "0xca843569e3427144cead5e4d5999a3d0ccf92b8e",
        "currency": "USD",
        "valid": true
    }
]
```