|  |
| --- |
| MB ReWrite: High Level Design Template |
| EPIC 3- Fund Transfer |
|  |

# Version history

# Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Reviewer** | **Approver** | **Comments** |
| **0.1** | 16 June 2022 | Ovais | Chandramouli | Prasad Hebbar | Draft Version |
| **0.2** | 28 July | Ovais | Pradeep |  |  |

Table of Contents

# Introduction

## Scope of the document

This document outlines the high-level functional design of beneficiary registration, TPT functionality. It highlights/refers the high-level flows / Use cases in registration process, design of components, along with the rationale for the same. It serves as an input to the low-level design documents that would further elaborate on the application design.

## Intended Audience

* + - *Backend Development Team*
    - *Architecture Team*
    - *Quality Team*

## System overview

This system provides a back end api’s to the beneficiary management for TPT includes IMPS, NEFT, RTGS, UPI, IFT others. The system also integrates with admin module, SOA, OBP and can provide a wrapper for these integrations.

The system will be developed using Golang, Google PubSub, Aerospike and will leverage the GCP cloud service for hosting and deployment.

# System Design

## Application Design

This section will capture Process flow, Information flow having sequence diagram for different use cases, component design details, key design considerations taken and API catalogue

### Process Flow

#### Fund Transfer

Graphical user interface, application

Description automatically generated

Fund Transfer process details

|  |  |
| --- | --- |
| Serial | Description |
| 1 | Fund Transfer request is accepted from the UI in encrypted format Data object |
| 2 | Following validations are performed,  MPIN  Account Status  Customer relationship with the account  in case of IMPS, check for participants member & blacklisting of IFSC |
| 2a | In case account information is missing then enquiry call is made to the OBP and results are stored in the local cache |
| 3 | Transaction details are stored in the local cache |
| 4 | Limits HOLD call is made, (for non Self transfer only) |
| 4a | In case of non-OTT, beneficiary call is validated |
| 5 | Fund Transfer call is being requested to the OBP for NEFT/ IMPS / RTGS / IFT / Self |
| 6 | In case of error, RELEASE call is requested\*.  In case of successful, CONFIRM call will be requested by a background scheduler. |
| 7 | Transaction details are updated in the local cache |

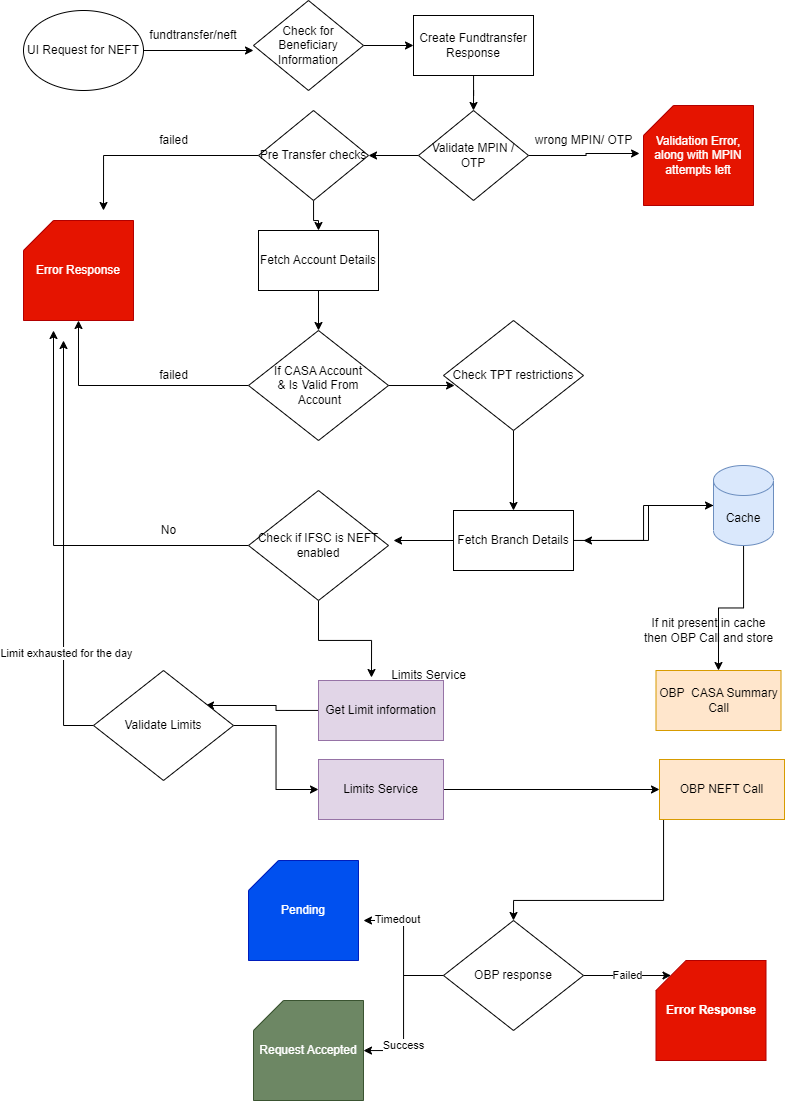
\*In case of timeout from OBP, relevant information is passed to the UI but the limit system will be updated by the background scheduler.

Information Flow

* *This section will capture flow of information between all application components required to meet the requirement*
* *This will capture flow for both positive and negative scenarios*

***Sequence Diagram:***

NEFT Fund transfer



## OBP API Discovery:

## <https://confluence.hdfcbank.com/display/MBR/NEFT+-+API+Discovery>

## Components Design

## The system will comprise of the following main micro services which will together function,

* Beneficiary System
* Fund Transfer
* TPT Manager
* Transaction Manger

These systems will interact with the following systems,

* Common Services (Bene & Limits, User)
* OBP (FC, Axiom, E2FA, Axiom others)

## Key Design Considerations

* + For Common services, it will REST call
  + OBP – REST and SOAP

## API Catalogue

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **API** | **OBP / CS** | **Description** |
| 1 | /validateHDFCaccount |  | PartnerAccountInquiryRest/InquirePartnerAccountDetails |
| 2 | /addpayee |  | Beneficiary system |
| 3 | /updatepayee |  | Beneficiary system |
| 4 | /deletepayee |  | Beneficiary system |
| 5 | /favourites |  | Beneficiary system |
| 6 | /getpayees |  | Beneficiary system |
|  |  |  |  |
|  | tptmanager |  |  |
| 7 | /getbanks |  | AeroSpike |
| 8 | /searchbranches | OBP | FinancialInstitutionInquiry |
| 9 | /searchifsc | OBP | FinancialInstitutionInquiry |
| 10 | /limitsinfo |  | limitsSystem |
| 11 |  |  |  |
|  | fundtransfer |  |  |
| 12 | /imps | OBP | IMPS/ProxyServices/PS\_IMPSTransaction |
| 13 | /neft | OBP | NEFTPayment |
| 14 | /neftsi | OBP | PaymentStandingInstructions |
| 15 | /selfsi | OBP | StandingInstructionsForOwnAccounts |
| 16 | /iftsi | OBP | StandingInstructions |
| 17 | /rtgs | OBP | RTGSPayment |
| 18 | /ift | OBP | FundsTransferMiscRestWrapper/miscFundTransfer |
| 19 | /self | OBP | FundsTransferOwnAccount |
| 20 | /tptcharges |  | config / admin module |
|  |  |  |  |
|  | customer |  |  |
| 21 | /getbalanebycustid | OBP | casaSummarySpi |
|  | Transactionmanager |  |  |
| 22 | /transactionhistory | OBP | FundsExchangeInquiryApplicationServiceSpi |

Utility Apis,

Get Limits Info

It determines the logged in customer’s eligibility to use the fund transfer flow. The limitsinfo API fetches TPT registration information from the common services user status api, for a user to be considered to a TPT registered, the user should have all the 3 roles **TPT, MBC TPT and MBB TPT** in the **RS** app of the user status response. The TPT registration information will be however processed by the admin module. User also has the option to enable / disable the TPT from the profile dashboard, this information will be accessed by the limits API and shared in the tptEnabled field.

Request Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** | **Remarks** |
| 1 | Hashed customer Id | string | Y |  |
| 2 | Device Platform | string | Y |  |

Response Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **FieldName** | **Remarks** | **Type** |
| 1 | **overallLimit** | Over all limit | float64 |
| 2 | **tptLimit** | TPT limit | float64 |
| 3 | **minTptLimit** | Minimum TPT limit | float64 |
| 4 | **maxTptLimit** | Maximum TPT limit | float64 |
| 5 | **oneTimeTransferLimit** | One time transfer limit | float64 |
| 6 | **mbChannelLimit** | MB channel limit | float64 |
| 7 | **tptEnabled** | If user is TPT enabled | bool |
| 8 | **tptRegistered** | If user is TPT registered | bool |
| 9 | **availableOttLimit** | Available One time transfer limit | float64 |
| 10 | **availableChannelLimit** | Available channel limit | float64 |
| 11 | **availableTptLimit** | Available TPT limit | float64 |
| 12 | **deviceLimit** | Device specific limit | float64 |

Sample response:

**{**

**"body":{ "overallLimit": 3000,**

**"tptLimit": 2000,**

**"minTptLimit": 1,**

**"maxTptLimit": 4000,**

**"oneTimeTransferLimit": 100,**

**"mbChannelLimit": 300, "tptEnabled": true, "tptRegistered": true, "availableOttLimit": 340,**

**"availableChannelLimit": 200,**

**"availableTptLimit": 500,**

**"deviceLimit": 200**

**},**

**"isEncrypted": true**

**}**

**Common Services API used:**

1. **/bene-limits-service/user/limits/get**
2. **/bene-limits-service/user/available-limits/get**
3. **/user-status-service/get**

Get TPT charges.

Available payment options including the display name & fees/charges (admin driven), benetype validation from FC (NEFT / RTGS), IMPS blacklisted/participation (master table populated by admin), NEFT limit availability according to the time of the day (admin driven).

Request Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** | **Remarks** |
| 1 | IFSC Code | string | Y |  |
| 2 | Hashed customer Id | string | Y |  |

Response Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **FieldName** | **Remarks** | **Type** |
| 1 | imps | IMPS charges | array of object |
| 2 | rtgs | RTGS charges | array of object |
| 3 | neft | NEFT charges | array of object |
| 4 | ift | IFT charges | array of object |
| 5 | isimpsBlacklisted | If imps blacklisted IFSC | bool |
| 6 | isimpsActive | If imps enabled bank | bool |

Sample response:

Sample response: { "body": { "imps": [ { "amountFrom": 1, "amountTo": 1000, "charges": 3.5, "gst": 0.18, "duration": "INSTANT", "displayName": "IMPS", "maxAmountAllowed: "400000" }, { "amountFrom": 1001, "amountTo": 100000, "charges": 5, "gst": 0.18, "duration": "INSTANT", "displayName": "IMPS", "maxAmountAllowed: "400000" }, { "amountFrom": 100001, "amountTo": 500001, "charges": 15, "gst": 0.18, "duration": "INSTANT", "displayName": "IMPS", "maxAmountAllowed: "400000" } ], "rtgs": [ { "amountFrom": 200000, "amountTo": 5000000, "charges": 0, "gst": 0.18, "duration": "INSTANT", "displayName": "RTGS", "maxAmountAllowed: "400000" } ], "neft": [ { "amountFrom": 1, "amountTo": 1000000, "charges": 0, "gst": 0.18, "duration": "INSTANT", "displayName": "NEFT", "maxAmountAllowed: "400000" } ], "ift": [ { "amountFrom": 1, "amountTo": 5000000, "charges": 0, "gst": 0.18, "duration": "INSTANT", "displayName": "IFT", "maxAmountAllowed: "400000" } ] isimpsBlacklisted: false, isimpsActive: true }, "isEncrypted": true }

Get Customer Balances By CustID

Customer can opt only selected account to Transfer Funds from, the relationship and account status determines the condition , the allowed relationship can be of any of the SOW, JOO, JOF and the prohibited statutes are 1,3,5,8,11. The status and relationship are fetched from the casaSummary API. The NRE, NRO flag is determined from product type and account status and relationship field are also present in the response of the API. The results are cached in the local DB which can be leveraged for the validations in IFT Fund Transfer.

Request Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** | **Remarks** |
| 1 | Hashed customer Id | string | Y |  |

Response Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **FieldName** | **Remarks** | **Type** |
| 1 | **overallLimit** | Over all limit | float64 |
| 2 | **tptLimit** | TPT limit | float64 |
| 3 | **minTptLimit** | Minimum TPT limit | float64 |
| 4 | **maxTptLimit** | Maximum TPT limit | float64 |
| 5 | **oneTimeTransferLimit** | One time transfer limit | float64 |
| 6 | **mbChannelLimit** | MB channel limit | float64 |
| 7 | **tptEnabled** | If user is TPT enabled | bool |
| 8 | **tptRegistered** | If user is TPT registered | bool |
| 9 | **availableOttLimit** | Available One time transfer limit | float64 |
| 10 | **availableChannelLimit** | Available channel limit | float64 |
| 11 | **availableTptLimit** | Available TPT limit | float64 |
| 12 | **deviceLimit** | Device specific limit | float64 |

Sample response:

**{**

**"body":{ "overallLimit": 3000,**

**"tptLimit": 2000,**

**"minTptLimit": 1,**

**"maxTptLimit": 4000,**

**"oneTimeTransferLimit": 100,**

**"mbChannelLimit": 300, "tptEnabled": true, "tptRegistered": true, "availableOttLimit": 340,**

**"availableChannelLimit": 200,**

**"availableTptLimit": 500,**

**"deviceLimit": 200**

**},**

**"isEncrypted": true**

**}**

**Common Services API used:**

1. **/bene-limits-service/user/limits/get**
2. **/bene-limits-service/user/available-limits/get**
3. **/user-status-service/get**

# Data Design

## Data Models

Fund Transfer

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** |
| 1 | TransactionID | string | Y |
| 2 | From Account No | string | Y |
| 3 | Hash User Id | string | Y |
| 4 | To Account No | string | Y |
| 5 | To IFSC | string | Y |
| 6 | To Name | string | Y |
| 7 | Amount | float | Y |
| 8 | BeneficiaryID |  |  |
| 9 | Remarks | string | N |
| 10 | SenderCommunicationModeValue | string | N |
| 11 | SenderCommunicationMode | string | Y |
| 12 | OTT | string | N |
| 13 | Mpin | string | N |
| 14 | OtpRefNo | string | N |
| 15 | OTPToken | string | N |

FundTransferSuccessResponse

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** |
|  | ToAccountNumber | string |
|  | PaidToName | string |
|  | FromAccountNumber | string |
|  | HDFCTransactionID | string |
|  | ReferenceNumber | string |
|  | Amount | float |
|  | TransferType | string |
|  | ValidationMethod | string |
|  | Status | string |
|  | ErrorMessage | string |
|  | TransactionTime | string |
|  | SLIToken | string |
|  | MpinAttemptsLeft | string |

Schedule Transfer

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** |
| 1 | ScheduleID | string | Y |
| 2 | **IsRecurring** | bool | Y |
| 3 | **StartDate** | datetime | Y |
| 4 | **EndDate** | datetime | N |
| 5 | **Frequency** | int | N |
| 6 | **Interval** | int | N |

TPT Option

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** |
|  | Option | string |
|  | Charges | float |

BanksList

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** |
| 1 | name | string | Y |
| 2 | ifscstart | string | Y |
| 3 | icon | string | Y |
| 4 | genericifscCode | string | Y |

Branches

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** |
| 1 | id | string |
| 2 | ifscCode | string |
| 3 | name | string |
| 4 | address | string |
| 5 | bankId | string |

Transaction Details

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Type** | **Mandatory** |
|  | hdfc\_ref\_no | string |  |
|  | transfer\_method | string | Y |
|  | created\_at | datetime | Y |
|  | updated\_at | datetime | Y |
|  | ext\_ref\_no | string |  |
|  | error | string | Y |
|  | status | string | Y |
|  | hashed\_user\_id | string |  |
|  | from\_acc\_no | string | Y |
|  | to\_ifsc | string |  |
|  | to\_acc\_no | string | Y |
|  | amount | float | Y |
|  | limits\_status | string |  |

## Data Access Mechanism

There will be a high level technical design consideration which will cover framework for data access mechanism i.e. ORM implementation details etc. In this section, any customization specific to module (for which this document is prepared) will be covered.

## Data Retention Policies

* + *This Section will capture policies related to data purging/data archiving*
  + *This will also capture specific tables for which data will be purged/archived*

## Data Migration

* + *This section will capture the data migration strategy*
  + *We know that data migration is big activity so the strategy may get capture in separate document. If it is being captured in separate document, we need to give the reference of same document.*

# Interfaces

This section will capture the high level details of other Interfaces which are interacting with our application in scope.

# State and Session Management

There will be separate high level technical design document which will cover framework for state and session management. In this section, any customization specific to module (for which this document is prepared) will be covered.

# Caching

There will be separate high level technical design document which will cover framework for caching. In this section, any customization specific to module (for which this document is prepared) will be covered.

# Non Functional Requirements

* *This section will capture non-functional aspects of system*
* *This section will capture Security aspects and Performance aspects of the system*

## Performance Aspects

This section will capture the performance aspects of the application in scope

# References

This section will list down the documents which has been refereed while creating this document

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Document Name** | **Version** | **Date** |
|  |  |  |  |
|  |  |  |  |

# Appendices

[Appendix 1. Comments Matrix 12](#_bookmark23)

# Appendix 1. Comments Matrix

Since there is no formal tool finalized currently for logging comments on various document deliverables, Comments Matrix is being put here to ensure comments are being tracked and resolved. Once a formal tool is identified, this process might change for future deliverables.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **IRD Review Comments** | **Date Raised** | **Vendor Remarks** |
| **1** |  |  |  |
| **2** |  |  |  |