

RAPIPAY

RBL NODAL SERVICE

Introduction	3
High-Level Flow	3
Components Architecture	3
Deployment Diagram	4
Database Architecture	8
Low-Level Flow	10
RBL API's for Registrations	10
Nodal Beneficiary registration	10
Request	11
Responses	13
Request	14
Responses	14
RBL API's for Settlement	17
Request	Error! Bookmark not defined.
Responses	18
Micro-services	22
API Signatures	22
Performance Plan	28
Flows need a performance test	28
Security Plan	28
Migration Plan	29

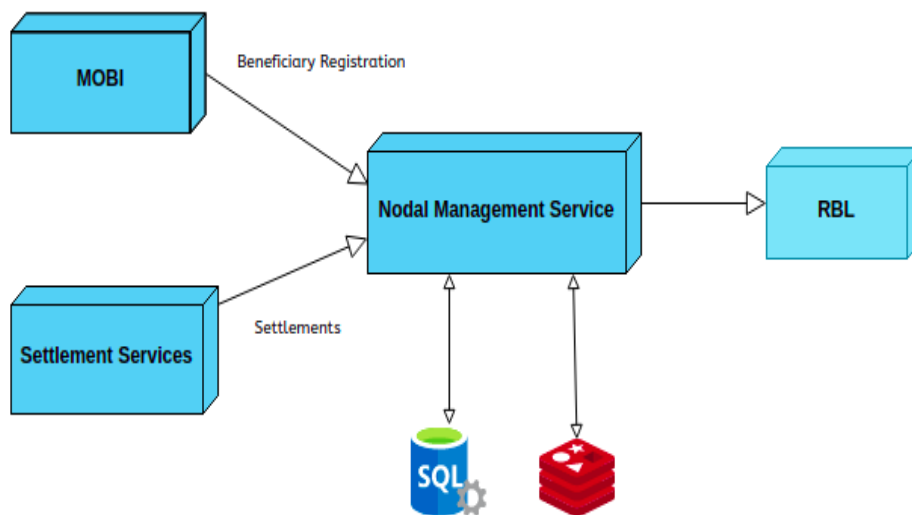
Introduction

We will be using the Nodal Management Service to access all services exposed by the RBL Bank that will be required in the functioning of Rapipay POS Service.

We will need to contact the RBL Service for mainly two purposes : one will be for beneficiary addition and updation and other for the money transfer from our nodal account to the beneficiary added.

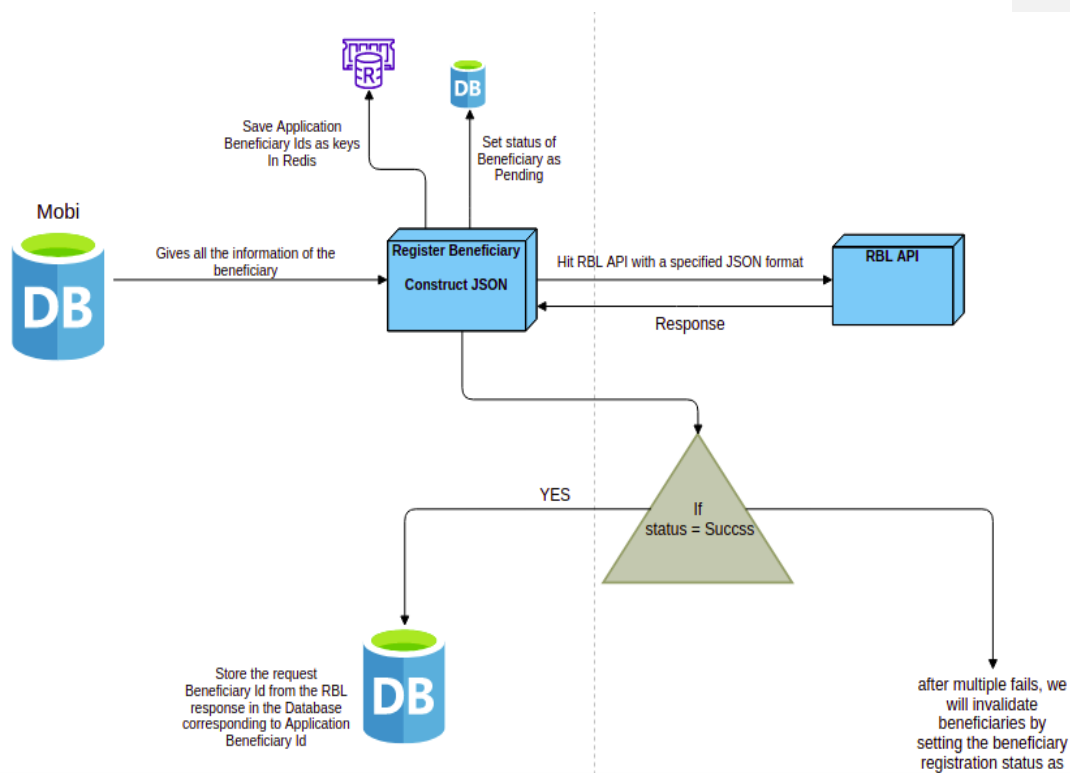
High-Level Flow

1.Components Architecture

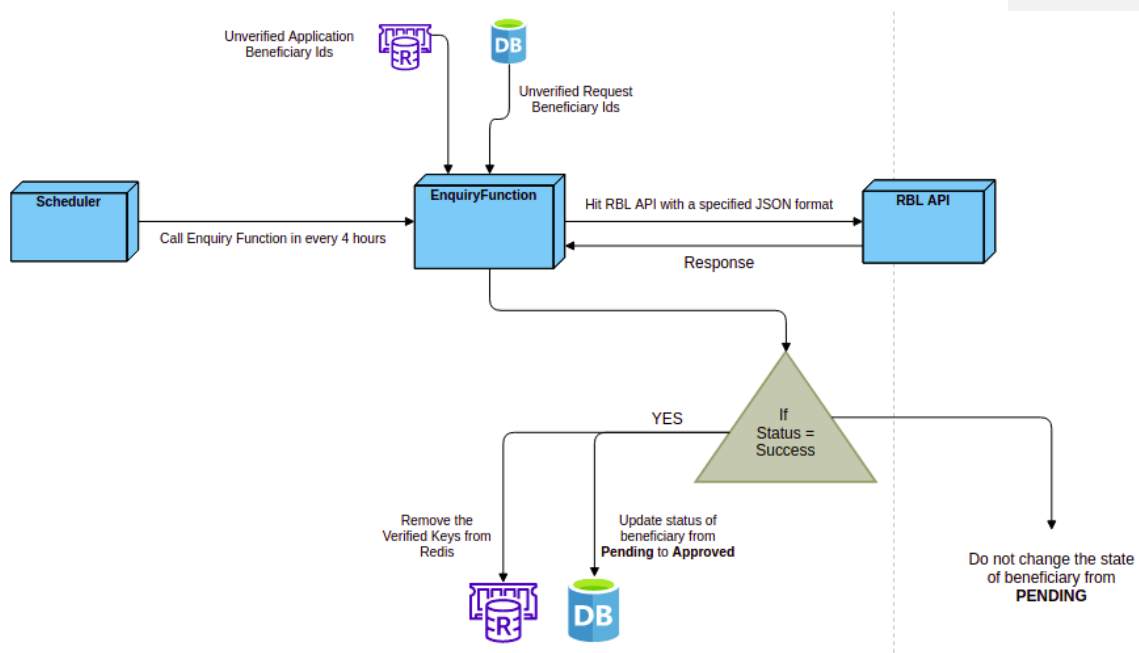


2. Deployment Diagram

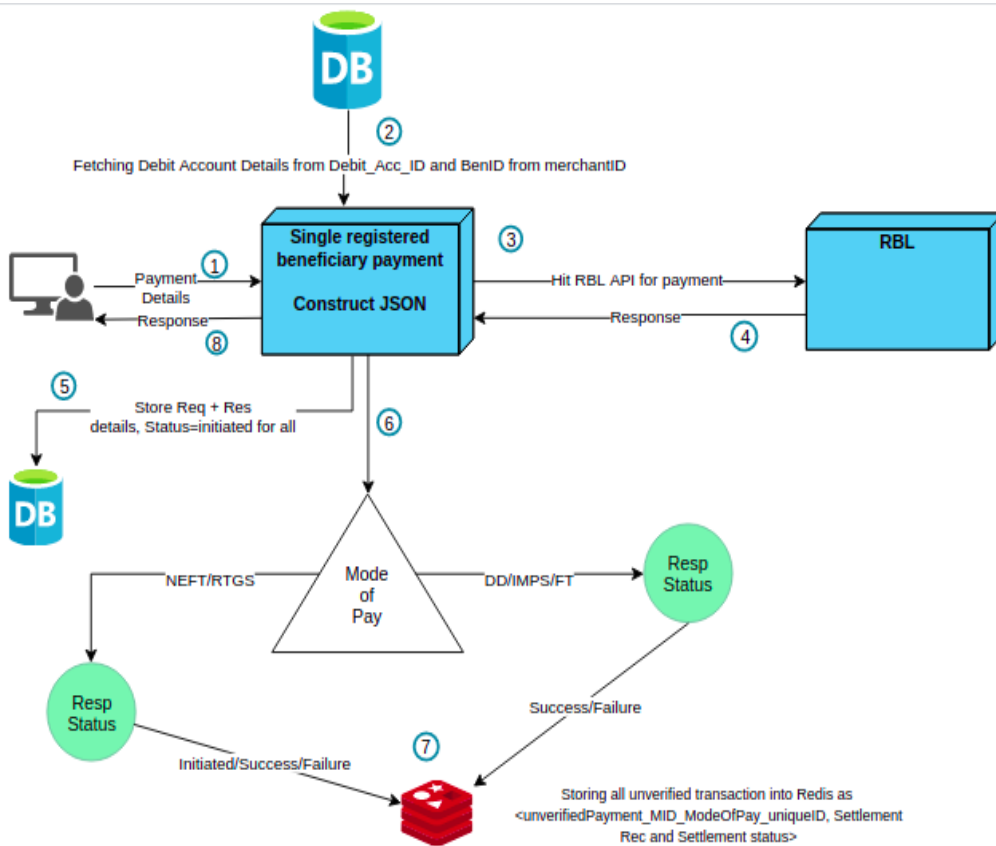
For Beneficiary Registration



For Beneficiary Enquiry



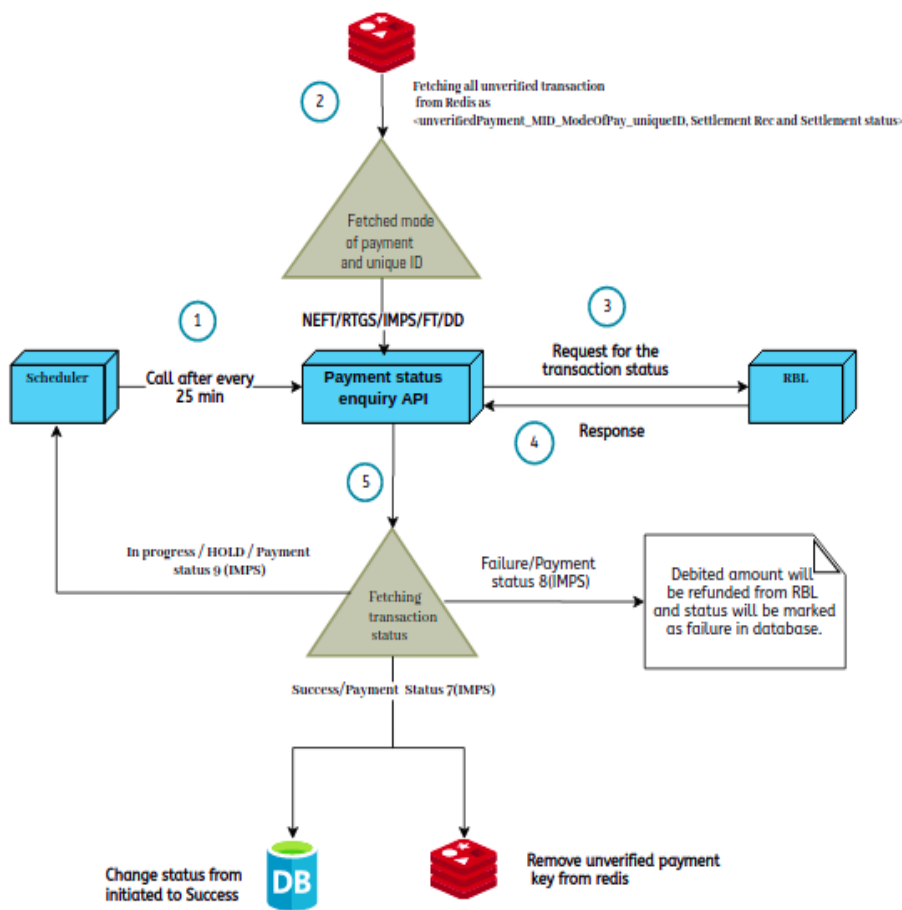
For Single Beneficiary payment



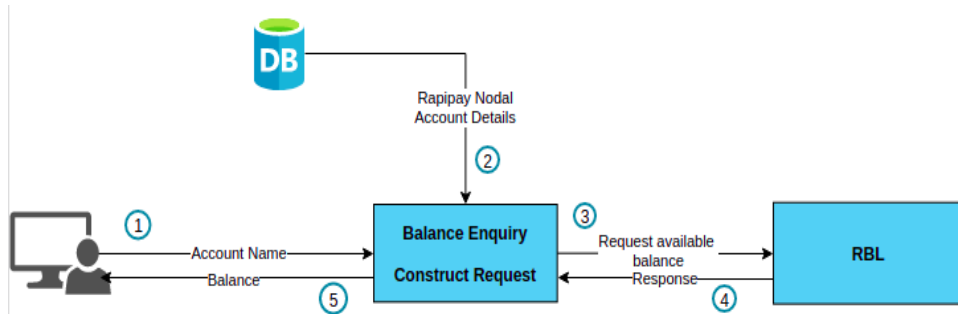
Commented [1]: 1. BenId from merchantId in Point 2?
2. what tables/columns will be updated in Point 5?
3. similarly in Point 7.

Commented [2]: 1. through database queries we will fetch the beneficiary id of a particular merchant
2. All the columns corresponding to the request and response sent and received from RBL in Settlement Record and Settlement Status Tables.
3. Same in the case of sql

For Single Beneficiary payment status enquiry



For Nodal Account Balance Enquiry

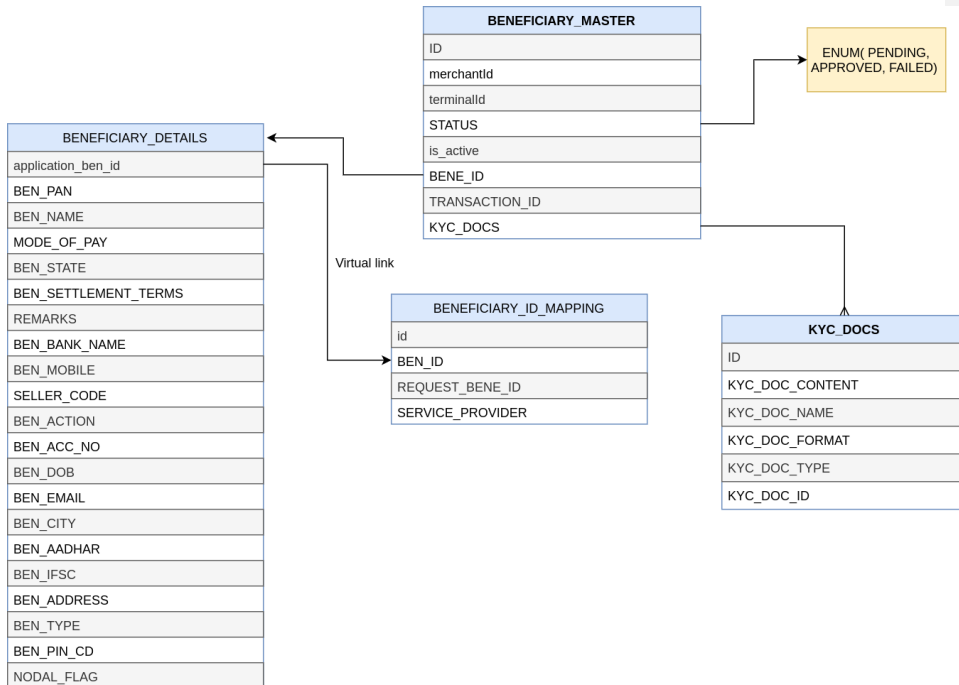


Commented [3]: Update balance in the db as well

Commented [4]: Sure.

3. Database Architecture

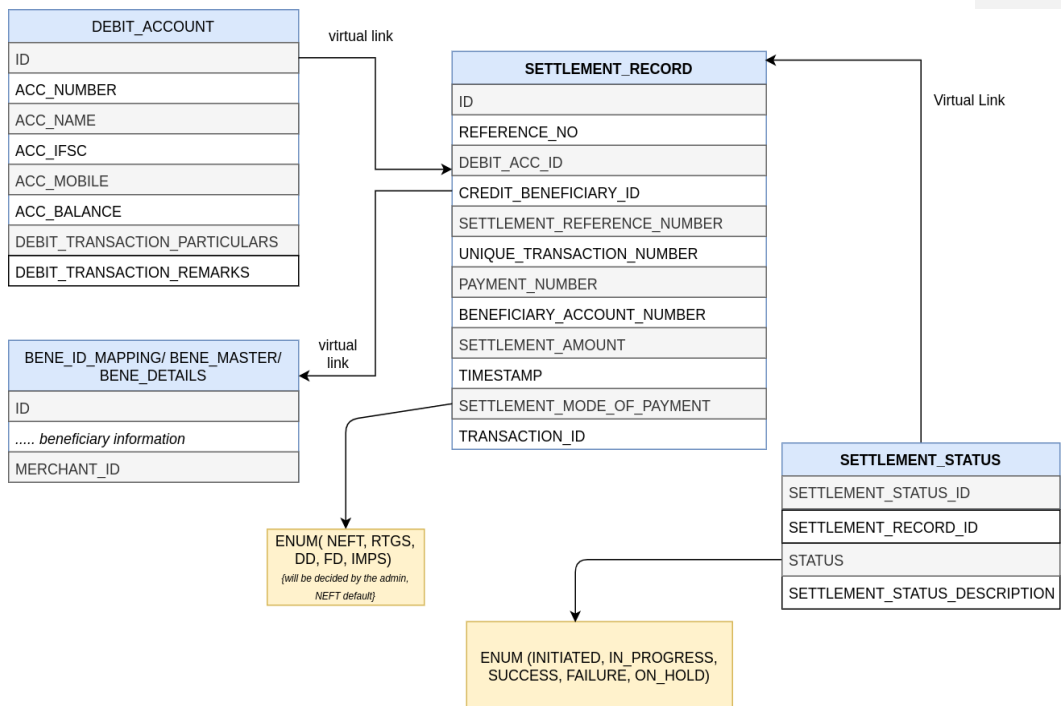
BENEFICIARY REGISTRATION/ENQUIRY DATABASE STRUCTURE



BEN_PARTICULARS - string that will get reflected in the bene accounts statement.
Currently by default we are setting it as

Seller_id - Generated by the merchantID

NODAL SETTLEMENTS DATABASE DESIGN



Low-Level Flow

1. RBL API's for Registrations

1. Nodal Beneficiary registration

Use Case : This API will be used for registering the merchant (that we have on-boarded using Rapipay MOBI Services) as a beneficiary on our RBL Nodal Account. Once

registered as a beneficiary we will be able to transfer funds between the Merchants Account and Rapipay Nodal Account.

URL for RBL Beneficiary registration api:

<https://apideveloper.rblbank.com/test/sb/rbl/api/v1.5/na-beneficiary/registration>

NOTE :-

- In below Code the **green highlighted** variables values are not mandatory .
- The variable's description is mentioned in comments and highlighted in **yellow** .
- The variables functionality which is not clear is highlighted in **red**.

1) Request

```
{
  "Beneficiary_Nodal_Account_Registration_Req": {
    "Header": {
      "TranID": 1,
      "Corp ID": "RAPIPAY",
      "Maker ID": "",
      "Checker ID": "",
      "Approver ID": ""
    },
    "Body": {
      "Ben_IFSC": "CBIN0R10001",
      "Ben_Acct_No": 1256905,
      "Ben_Name": "KRALIYET WELLNESS OPC PRIVATE LIMITED",
      "Ben_Address": "N5-10 IRC VILLAGE NAYAPALLI",
      "Ben_State": "Orissa",
      "Ben_City": "Bhubaneswar",
      "Ben_PinCd": 751015,
      "Ben_DOB": "1989-04-11",
      "Ben_BankName": "Kotak Mahindra Bank",
      "Ben_BankCd": 176,
      "Ben_BranchCd": 110,
      "Ben_Email": "satyajeeetb2@gmail.com",
      "Ben_Mobile": 9658065598,
      "Ben_TxnParticular": "abc",
      "Ben_PartInBank": "abc",
      "Issue_BranchCd": 61,
      "Ben_PAN": "AAGCK0479A",
      "Ben_UID": ""
    }
  }
}
```

Commented [5]: This is a unique ID assigned to each corporate for identification, once customer onboarding process is completed. Bank will provide this ID

Commented [6]: This ID is mandatory only if customers have opted for validation option; Bank will provide this ID in case validation is opted by customers. Otherwise, no need to pass this value.

Commented [7]: This ID is mandatory only if customers have opted for validation option; Bank will provide this ID in case validation is opted by customers. Otherwise, no need to pass this value.

Commented [8]: same as that of Checker_ID

Commented [9]: Beneficiary's details are given below

Commented [10]: Value is Mandatory for DD and When Ben Action 0 and 1

Commented [11]: Beneficiary Transaction Particulars. This value will get reflect in narration of beneficiary account statement.

Commented [12]: These are the additional remarks, which will not reflect in the statement. But will get saved at Bank end.

```

"VirtualAccNum": "VAABC09E346581",
"seller Code": 1,
"Mode_of_Pay": {
  "NEFT": {
    "YN": "Y",
    "Limit": {
      "Daily": 5000,
      "Weekly": 35000,
      "Monthly": 105000
    }
  },
  "RTGS": {
    "YN": "Y",
    "Limit": {
      "Daily": "",
      "Weekly": "",
      "Monthly": ""
    }
  },
  "DD": {
    "YN": "Y",
    "Limit": {
      "Daily": "",
      "Weekly": "",
      "Monthly": ""
    }
  },
  "FT": {
    "YN": "Y",
    "Limit": {
      "Daily": "",
      "Weekly": "",
      "Monthly": ""
    }
  },
  "IMPS": {
    "YN": "Y",
    "Limit": {
      "Daily": "",
      "Weekly": "",
      "Monthly": ""
    }
  }
},
"Bene_Type": "Private/Public Ltd Co/One Person
Company",
"Ben_SettlementTerm": "",
"Ben_CommercialTerm": "",

```

Commented [13]: Unique SellerID assigned to every beneficiary

Commented [14]: Settlement terms in between Beneficiary and corporate

is not required

Commented [15]: Commercial Terms in between Beneficiary and Corporate

is not required

```

        "KYC_Document": [{
            "KYC_Doc_Id": 55555,
            "KYC_Doc_Name": "PAN Card",
            "KYC_Doc_Type": "POI",
            "KYC_Doc_Format": "PDF",
            "KYC_Doc_Content": "abc"
        }],
        "Remarks": "Sending Request for Register new
Beneficiary",
        "Ben_Action": 0,
        "Nodal_Flag": "N",
        "Ben_ID": ""
    },
    "Signature": {
        "Signature": "12345"
    }
}

```

Commented [16]: This value should always be passed as "N". It gets used for some internal validation.

Commented [17]: Currently any value can be passed in this field .

Encrypted key for authenticating the request. Currently not required.

2) Responses

```

{
    "getListofAccountsfromCorpIDReq": {
        "Header": {
            "TranID": 1,
            "Corp_ID": "CorpID",
            "Maker_ID": "",
            "Checker_ID": "",
            "Approver_ID": ""
        },
        "Body": {
            "Status": "Success",
            "Ben_Id": "BENCORP10064 ",
            "Corp_Name": "RED CHILLI",
            "Ben_Name": "Beneficiary name",
            "Remarks": "NODAL",
            "Ref_No": "BEN1489045502008",
            "Txn_Time": "2017-03-09 13:15:08.259481"
        },
        "Signature": {

```

Commented [18]: not getting List

Commented [19]: Unique reference number provided for each reqes

```

        "Signature": "Signature001"
    }
}

```

2) Nodal Account Beneficiary Enquiry

Use Case : This API gives the detailed information for the given Beneficiary id in the request. Users can check the status of the registered beneficiary. Also it gives the rejection reason, remarks etc. in case of rejected beneficiary. This API will be triggered through a scheduler running every four hours to update the status automatically as well on trigger.

1. Request

```

{
  "Ben_Temp_Acc_Enq_Req": {
    "Header": {
      "Transit": "152276816",
      "Corp_ID": "RAPIPAY",
      "Maker_ID": "",
      "Checker_ID": "",
      "Approver_ID": ""
    },
    "Body": {
      "Ben_ID": " BENNODAL250340",
      "Date": ""
    },
    "Signature": {
      "Signature": "Sign"
    }
  }
}

```

Commented [20]: Transaction ID generated by consumer of service which is unique

Commented [21]: Date when beneficiary is registered

2. Responses

```

{
  "Ben_Temp_Acc_Enq_Res": {
    "Header": {
      "TransId": "152276816",
      "Corp_ID": " RAPIPAY ",
      "Maker_ID": "",
      "Checker_ID": "",
      "Approver_ID": ""
    },
    "Body": {
      "Status": "Success",
      "EnquiryTime": "2018-04-05 11:52:27.514917",
      "Details": [
        {
          "BEN_ID": "BENNODAL250340",
          "BEN_IFSC": "CBIN0R10001",
          "BEN_NAME": "Akash T",
          "BEN_DOB": "2017-04-30",
          "BEN_ACCNO": "501042034",
          "BEN_ACCSTATUS": "D",
          "BEN_BANKNAME": "RblBank-Mumbai",
          "BEN_BANKCODE": "176",
          "BEN_BRANCHCODE": "0075",
          "BEN_ADDRESS": "A/P-Mumbai,Goregaon West.Techniplex",
          "BEN_STATE": "Daman and Diu",
          "BEN_CITY": "Daman",
          "BEN_PINCD": "416310",
          "BEN_TYPE": "Sole Proprietor",
          "BEN_MOBILE": "9810100100",
          "BEN_EMAIL": "abc@rblbank.com",
          ""
        }
      ],
      "DD": {
        "YN": "Y",
        "Limit": [
          {
            "DAILY": 0
          },
          {
            "WEEKLY": 0
          },
          {
            "MONTHLY": 0
          }
        ]
      }
    }
  }
}

```

Commented [22]: Transaction Id that will be entered by the customer and it will be unique for each request.

Commented [23]: The field will show the time on which enquiry has been initiated.

```

    ]
  },
  "FT": {
    "YN": "Y",
    "Limit": [
      {
        "DAILY": 111
      },
      {
        "WEEKLY": 1111
      },
      {
        "MONTHLY": 11111
      }
    ]
  },
  "IMPS": {
    "YN": "Y",
    "Limit": [
      {
        "DAILY": 101
      },
      {
        "WEEKLY": 10011
      },
      {
        "MONTHLY": 1100111
      }
    ]
  },
  "TXNTIME": "2016-04-05 11:51:43.728889",
  "NODAL_FLAG": "No",
  "APP_REJ_STATUS": "Pending",
  "APP_REJ_REMARK": "",
  "CHKR_REMARKS": "",
  "KYC_Document": [
    {
      "KYC_Doc_Id": "Creating6",
      "KYC_Doc_Name": "License issued by Registering
Authority",
      "KYC_Doc_Type": "POI",
      "KYC_Doc_Content": "",
      "KYC_Doc_Format": "PDF"
    }
  ]
}

```

Commented [24]: Timestamp

Commented [25]: Always NO

Commented [26]: The field will show the remarks entered by final authority who has approved or rejected the beneficiary ID

Commented [27]: NA


```

    ]
  },
  "Signature": {
    "Signature": "Sign"
  }
}
}

```

2. RBL API's for Settlement

1. Single Beneficiary Settlement API

USE CASE : After the reconciliation process is completed, Rapipay admin will trigger the flow that performs the settlement for a single merchant as well as upload a file that can handle bulk settlements

URL for RBL Single Beneficiary Payment API:

https://apideveloper.rblbank.com/test/sb/rbl/api/v1/payment_bid/pay

Commented [28]: (first reconcile from switch ledger files - transaction files(TRANSACTION table) -> admin view will have reconciliation process done and give a generated sum of settlement → this will acts as the output file for settlement(T+1).
Rapipay Admin -> will settle - Bulk upload + single payment and trigger the RBL flow

Request

```

{
  "Single_Payment_Corp_Req": {
    "Header": {
      "TranID": "TranID",
      "Corp_ID": "CorpID",
      "Maker_ID": "",
      "Checker_ID": "",
      "Approver_ID": ""
    },
    "Body": {
      "Amount": "1",
      "Debit_Acct_No": "123456789",
      "Debit_Acct_Name": "Rahul",
      "Debit_IFSC": "RAHUL22123",

```

```

        "Debit_Mobile": "9791979797",
        "Debit_TrnParticulars": "TrParticulars",
        "Debit_PartTrnRmks": "PartTrnRmks",
        "Mode_of_Pay": "NEFT",
        "Remarks": "Remarks",
        "Ben_ID": "BENID"
    },
    "Signature": {
        "Signature": "Signature"
    }
}

```

Commented [29]: Which field of which table Ben_IrD?

Commented [30]: This is a typo the field is "Ben_ID", and this is fetched from the BeneficiaryIdMapping tables "REQUEST_BENE_ID"

Responses

```

    {"Single_Payment_Corp_Resp": {
"Header": {
    "TranID": "TranID",
    "Corp_ID": "CorpID",
    "Maker_ID": "",
    "Checker_ID": "",
    "Approver_ID": "",
    "Status": "Initiated",
    "Error_Cde": {},
    "Error_Desc": {}
},
"Body": {
    "RefNo": "SP123456789123",
    "UTRNo": "RATNN12345678912",
    "PONum": "000098765432",
    "Ben_Acct_No": "987654321",
    "Amount": "1",
    "Txn_Time": "2001-01-22 18:56:47.840921",
    "Ben_ID": "BENID"
},
"Signature": {"Signature": "Signature"}
}}

```

Commented [31]: What Is PONum, validations on it?

Commented [32]: Please refer to RBL's documentation. This field is stored as it is in the database on our end.

Commented [33]: Is this beneficiary Id sent in request?

Commented [34]: yes

2. Single Beneficiary Settlement API Status Check

USE CASE : To check the status of a payment. Trigger points - update regularly on an update - give live updates to admin

URL for RBL Single Beneficiary settlement Status API:

https://apideveloper.rblbank.com/test/sb/rbl/api/v1/payment_bid/payment/query

Request

```
{
  "get_Single_Payment_Status_Corp_Req": {
    "Header": {
      "TranID": "2018051800021",
      "Corp_ID": "ZSA01",
      "Maker_ID": "",
      "Checker_ID": "",
      "Approver_ID": ""
    },
    "Body": {
      "UTRNo": "RATNN17361382321"
    },
    "Signature": {
      "Signature": "Signature"
    }
  }
}
```

```
}  
}
```

Responses

```
{ "get_Single_Payment_Status_Corp_Res": {  
  "Header": {  
    "TranID": "2018051800021",  
    "Corp_ID": "ZSA01",  
    "Maker_ID": "",  
    "Checker_ID": "",  
    "Approver_ID": "",  
    "Status": "SUCCESS",  
    "Error_Cde": "",  
    "Error_Desc": ""  
  },  
  "Body": {  
    "ORIGTRANSACTION": "789889970",  
    "AMOUNT": "1.00",  
    "REFNO": "SPZSA01789889970",  
    "UTRNO": "RATNN17361382321",  
    "PONUM": "000027007310",  
    "BEN_ACCT_NO": "1000110010002463",
```

Commented [35]: Significance of this field

Commented [36]: Please refer RBL's document. This is part of RBL's request and response structure

Commented [37]: Significance of this field?

Commented [38]: Please refer RBL's document. This is part of RBL's request and response structure

```

    "BENIFSC": "CBIN0R10001",

    "TXNSTATUS": "Success",

    "STATUSDESC": "",

    "BEN_CONF_RECEIVED": "N",

    "TXNTIME": "2017-12-27 11:17:38.62"

  },

  "Signature": {"Signature": "Signature"}

}}

```

Commented [39]: Significance of this field?

Commented [40]: Please refer RBL's document. This is part of RBL's request and response structure

3. Nodal Account balance Inquiry

USE CASE : Rapipay admin can check available balance present in Rapipay Nodal RBL bank account.

URL of RBL for balance enquiry: **POST:**

<https://apideveloper.rblbank.com/test/qa/rbl/v1/accounts/balance/query>

Request

```

{
  "getAccountBalanceReq": {
    "Header": {
      "TranID": "1234678901234567",
      "Corp_ID": "RAPIPAY",
      "Approver_ID": "A001"
    },
    "Body": {
      "AcctId": "1256905"
    },
    "Signature": {
      "Signature": "12345"
    }
  }
}

```

Response:

```
{
  "getAccountBalanceRes": {
    "Header": {
      "TranID": "1234678901234567",
      "Corp_ID": "RAPIPAY",
      "Approver_ID": "A001",
      "Status": "Success",
      "Error_Cde": "",
      "Error_Desc": ""
    },
    "Body": {
      "BalAmt": {
        "amountValue": "",
        "currencyCode": {}
      }
    },
    "Signature": {
      "Signature": "12345"
    }
  }
}
```

Micro-services

- RBL Nodal Management Service

g

API Signatures

RESTFUL API's

1. Beneficiary Registration

- URL

POST: <http://10.10.20.83:8085/nodal-beneficiary/register>

- Sample Request Body

```
{
  "mid": "999",
  "name": "Tester 1",
  "mobileNo": "9818455922",
  "email": "Tester@gmail.com",
  "dobOnDoc": "1998-11-17",
  "docAddress": "C-115/Green Park",
  "accountNumber": "438624961052",
  "bankName": "HDFC bank",
  "ifscCode": "HDFC0065",
  "state": "KARNATAKA",
  "businessCity": "Mysore",
  "pinCode": "110017",
  "tid": "17",
  "panNumber": "BYQPP5817D",
  "contentList": [{
    "kycDocName": "PAN Card",
    "kycDocType": "POI",
    "kycDocFormat": "JPEG",
    "kycDocContent": "NA"
  }]
}
```

- Sample Response Body

```
{
  "ResponseCode": 200,
  "ResponseMessage": "Success",
  "ResponseBody": {
    "status": "Success",
    "benId": "BENRAPIPAY21041611593743718581",
    "corpName": "Rapipay",
    "benName": "Shatadru Sharma",
    "remarks": "ePOS",
    "refNo": "BENRAPIPAY21041611593743718581",
    "txnTime": "2021-04-16 11:59:39.463645"
  }
}
```

2. Beneficiary Enquiry

- URL

DEV - GET: `http://10.10.20.83:8085/nodal-beneficiary/enquire`

- Sample Request Parameters

```
{
  "merchantId": "123",
  "terminalId": "123abc"
}
```

- Sample Response Body

```
{
  "code": 200,
  "message": "Success",
  "data": "PENDING"
}
```

3. Single Beneficiary Payment

- URL
- POST: `http://10.10.20.83:8085/nodal-payment/single-beneficiary-payment`
- Sample Request Body


```
{
  "merchantId": "999",
  "terminalId": "87",
  "modeOfPay": "NEFT", (Admin Rapipay)
  "amount": "1",
  "debitAccountId": "123",
  "remarks": "Remarks"
}
```

- Commented [41]: Validations on mid, tid
- Commented [42]: covered in the discussion on 6th May
- Commented [43]: Scope of modeOfPay-NEFT, RTGSetc
- Commented [44]: covered in the discussion on 6th May
- Commented [45]: Validations on amount
- Commented [46]: covered in the discussion on 6th May
Refer <https://rapipayhq.atlassian.net/browse/POS-349> for MOM

- Sample Response Body

```
{
  "ResponseCode": 200,
  "ResponseMessage": "Success",
  "ResponseBody": {
    "status": "Initiated",
    "tranId": "1593743718581",
    "settlementRefNo": "3344533",
    "benId": "eerr",
    "amount": "1",
    "settlementRecordId": "78",
    "merchantId": "999",
    "terminalId": "87",
    "modeOfPay": "NEFT", (Admin Rapipay)
    "debitAccountId": "123"
  }
}
```

- Commented [47]: Validations and persistence of these fields
- Commented [48]: covered in the discussion on 6th May
- Commented [49]: Validations on mid, tid
- Commented [50]: covered in the discussion on 6th May
- Commented [51]: Scope of modeOfPay-NEFT, RTGSetc
- Commented [52]: covered in the discussion on 6th May

4. Single Beneficiary Payment Enquire

- URL
- DEV - POST:<http://10.10.20.83:8085/nodal-beneficiary/paymentEnquire>

- Sample Request Parameters

```
1801
```

- Sample Response Body

```
{
  "ResponseCode": "RES_200",
  "ResponseMessage": "Success",
  "ResponseBody": {
    "settlementStatusId": 1802,
    "settlementRecordId": 1801,
    "settlementStatusEnum": "IN_PROGRESS",
    "settlementStatusDescription": "",
    "transactionId": "1d3d57ea96",
    "referenceNumber": "NA",
    "debitAccountId": 1,
    "creditBeneficiaryId": 1784,
    "settlementReferenceNumber": "SPRAPIPAY1d3d57ea96",
    "uniqueTransactionNumber": "RATNN21126150867",
    "paymentNumber": "000130658475",
    "beneficiaryAccountNumber": "438624961052",
    "settlementAmount": 1,
    "timeStamp": "2021-05-06 19:13:55.157241",
    "settlementModeOfPayment": "NEFT"
  }
}
```

5. Balance Enquiry API

- URL
- GET: <http://10.10.20.83:8085/nodal-payment/balance-enquire>
- Sample Request Params

```
{
  "accountName": "RFPL RBL Noda1",
  "debitAccountId": "123"
}
```

- Sample Response Body

```
{
  "ResponseCode": 200,
  "ResponseMessage": "Success",
  "ResponseBody": {
    "accountName": "RFPL RBL Noda1",
    "accountId": "4",
    "amount": "1"
  }
}
```



KAFKA ENDPOINTS

1. Beneficiary Registration Consumer

Topic : beneficiary.registration

Expected JsonPayload mapped by POJO:

```
{
  "mid": "123",
  "name": "Asutoshi Padhi s",
  "mobileNo": "8452066043",
  "email": "atrijsharma2@gmail.com",
  "dobOnDoc": "1999-08-04",
  "docAddress": "17TH CROSS,NO.18,KAY KAY TOERS,9TH MAIN,7TH
SECTOR,HSR LAYOUT,KARNATAKA,560102 BANGALORE",
  "accountNumber": "3090020279979008",
  "bankName": "HDFC bank",
  "ifscCode": "HDFC0065",
  "state": "KARNATAKA",
  "businessCity": "Mysore",
  "pinCode": "110067",
  "tId": "abc123",
  "panNumber": "BYQPP5816C",
  "contentList": [{
    "kycDocName": "atrijdoc",
    "kycDocType": "POI",
    "kycDocFormat": "JPEG",
    "kycDocContent": "NA"
  }]
}
```

Expected POJO :

```
public class BeneficiaryRegistrationPayload implements Serializable
{
    private String mid;
    private String name;
    private String mobileNo;
    private String email;
    private String dobOnDoc;
    private String docAddress;
    private String accountNumber;
    private String bankName;
    private String ifscCode;
    private String state;
    private String businessCity;
    private String pinCode;
```

```
private String tid;  
private String panNumber;  
private List<KYCDocuments> contentList;  
  
}
```

ERROR CODES

The following document contains the list of all RBL error codes as well as a thorough list of all the NodalManagementService Error Codes and messages.

<https://docs.google.com/spreadsheets/d/19e95ueKkNR1eMFhV85W3U9GD4SbiJfTbDBrzhA9WIC8/edit?usp=sharing>

Single beneficiary payment error codes mapping with NodalManagementService Error Codes and messages.

<https://docs.google.com/spreadsheets/d/1xQdcyXdNStz9Qv26vcDuC98thsCXXTZkfEmSk8CIF7Y/edit?usp=sharing>

Performance Plan

Flows need a performance test

Please refer to the following kafka configuration document which contains details regarding the test bed as well as configuration parameters.

https://docs.google.com/spreadsheets/d/1XY7J_Nd9p7gzluYztkZVE1HdxB9ZZih8Kb_JXPuP-II/edit?usp=sharing

Expected throughput - existing base line

Actual throughput - To be updated later

Security Plan

Nodejs will act as the proxy layer to filter all the incoming requests. No requests should come directly to these endpoints. All requests will come through the proxy layer which will handle malicious requests as well

Migration Plan

No migration required.