UNIFIED PAYMENTS INTERFACE

API and Technical Specification Document

Version 2.0

DOCUMENT RELEASE NOTICE

Name	Version No.	Date	Description
Unified Payments Interface API and Technology Specifications	1.0	5-May-2015	Provides API and technical specifications of UPI.
Unified Payments Interface API and Technology Specifications	1.1	15-Oct-2015	1. Added TxnId in UPI URL 2. Updated ReqPay, RespPay message with DEBIT, CREDIT and REVERSAL Message 3. Type in Section 5.2, 5.3 4. Updated Meta APIs with Txn Id in Section 5.6 5. Updated Section 5.6.3 List Keys with UIDAI and UPI keys 6. Updated ListAccount API 7. Added Mobile Banking Registration API in Section 5.6 8. Added Transaction Confirmation API in section 5.6 9. Added Annexure A for Common Library Specifications
Unified Payments Interface API and Technology Specifications	1.2	10-Feb-2016	1. Added customer reference 2. Update List Account and List Keys API
Unified Payments Interface API and Technology Specifications	1.2.1	16-Mar-2016	1. Changes in ListAccount Manage 2. Verified Address API, Mobile Registration
Unified Payments Interface API and Technology Specifications	1.2.2	10-Jun-2016	1. Changes in Ref tag of RespPay/ ReqTxnConfirmation 2. Changes in chkTxnStatus API 3. Attribute definition of ListAccount api dtype element updated.
Unified Payments Interface API and Technology	1.2.3	20-Jun-2016	1. Balance Enquiry Data format example added 2. Device tag Capability, type examples added

Specifications			
UPI API Specification	2.0 v1	14-Jul-2017	1. UPI-Mandate 2. Aadhaar Biometric 3. Online Refund 4. Web Collect 5. ATM PIN Validation in Issuer Page 6. ATM PIN Changes (CL 1.5) 7. API Field Updates 8. USSD Updates 9. Fields for FIR.
UPI API Specification	Updated 2.0 v2	01-Sept-2017	1. API Fields updates 2. Updated few desriptions in sec 5.9 and 5.13
UPI API Specification	Updated 2.0 v3	07-Sept-2017	1. Included the Preapproved Mandate flow under scenarios
UPI API Specification	Updated 2.0 v4	08-Sept-2017	1. Included the Fortnightly in Recurrence pattern in UPIMandate
UPI API Specification	Updated 2.0 v5	11-Sept-2017	1. Added Sequential flow for Preapproved Mandate flow 2. Updated Sequential flow for Online Refund 3. Added Sequential financial Flow for Preapproved Mandate Flow
UPI API Specification	Updated2.0 v34	21-03-2018	1. Added purpose& merchantType field in all financial and mandate api's 2. Added type="ListPspKeys" & pspOrgId field in the ReqListKeys api 3. Added min & max version in ReqListAccPvd & ReqListPsp api's 4. Added subtype=PAY|COLLECT|REFUND|REVERSAL|MANDATE|DEBIT|CREDIT in the ReqChkTxn api 5. Added type=BalChk &amount tag in the ReqBalEnq and data value will be Y|N for BalChk api 6. Included InitiatedBy & blockFund in all the mandate api's 7. Note will be added for revoke mandate cred block formation
UPI API Specification	Updated 2.0 v35	12-04-2018	1. Added umn in check transaction API and also added result="REVOKED" in the RespChktxn API. 2. Added new cred in ReqRegMob & ReqOtp api's and related comments
UPI API Specification	Updated 2.0 v37	23-04-2018	1. Minor tag level changes 2. Versioning comment has been changed 3. Online Refund related tag changes in ReqPay, RespPay, ReqChkTxn & RespChkTxn api
UPI API SPecification	Updated 2.0 V40	27-07-2018	1. New Account types are included (SOD|UOD) 2. Merchant block is added in Mandate api's 3. New purpose types are added

UPI API SPecification	Updated 2.0 V41	23-08-2018	1. Added "refCategory" field in txn block for financial and mandate legs 2. Added 4 fields (pageSize, pageRecStart, pageRecEnd, pageSeqNum & pageTotal) in ListKeys and List Vae api's for pagination purpose 3. Added "merchantGenre" & "onBoardingType" fields inside merchant block 4. Added "ATMREDIRECT" type in OTP flow 5. Added featured supported tag in Response Validate Address

Contents

1. Glossary

Sender / Payer	Person/Entity who pays money. Payer's account is debited as part of payment transaction.
Receiver / Payee	Person/Entity who receives money. Payee's accountis credited as part of payment transaction.
Customer	An individual person or an entity having an account and wishes to pay and/or receive money.
Payment Account (or 'Account')	Any bank account or any other payment accounts (PPI, Wallets, Mobile Money, etc.) offered by a regulated entity where money can be held, debited from, and credited to.
Payments Service Provider (PSP)	Bank, Payment Bank, PPI, or any other RBI regulated entity that is allowed to acquire customers and provide payment (credit/debit) services to individuals and entities.
NPCI	National Payments Corporation of India.
RBI	Reserve Bank of India.
UIDAI	Unique Identification Authority of India which issues digital identity (called Aadhaar number) to residents of India and offers online authentication service.
IMPS	Immediate Payment System, a product of NPCI, offering an instant, 24X7, inter-bank electronic fund transfer service through mobile phone.
AEPS	Aadhaar Enabled Payment System. A system allowing Aadhaar biometric authentication based transactions, from a bank account that is linked with the Aadhaar number.
APB	Aadhaar Payment Bridge. A system allowing remittances to Aadhaar number, without providing any account details.
2-FA	Two factor authentication.
USSD	Unstructured Supplementary Services Data
UPI	Unified Payments Interface
API	Application Programming Interface
AUA	Authentication User Agency

FRM	Fraudulent Risk Management
Aeba	Aadhaar Enabled Bank Account
Mbeba	Mobile Banking Enabled Bank Account
MTO	Money Transfer Operator
MTSS	Money transfer Service Scheme
RDA	Rupee Drawing Arrangement
FIR	Foreign Inward Remittance

2. Introduction

Over the decades, India has made steady progress in the field of electronic payments. The innovations in the digital payments motivated the organizations to consolidate and integrate multiple systems with varying service levels, into a nation-wide, uniform, and standard business process for all retail payment systems.

The consolidated system should facilitate an affordable payment mechanism to benefit the common men across the country and help financial inclusion.

National Payments Corporation of Indiaunderstood the importance of such payment product and introduced **Unified Payments Interface** (**UPI**).

Unified Payments Interface (UPI) is a highly innovative, flexible product, and can be integrated easily with any bank in a standardized way in minimal time.

This document provides details of payments'architecture, which is directly connected to achieving the goals of universal electronic payments, a less cash society, and financial inclusion; using the latest technology trends laid down by RBI in RBI Payment System Vision Document (2012-15).

The RBI Payments System Vision document emphasises the mission and vision clearly:

MISSION

To ensure payment and settlement systems in the country are safe, efficient, interoperable, authorised, accessible, inclusive and compliant with international standards.

VISION

To proactively encourage electronic payment systems for ushering in a less-cash society in India.

The Mission statement indicates RBI's renewed commitment towards providing a safe, efficient, accessible, inclusive, interoperable, and authorised payment and settlement systems for the country. Payments system is driven by customer's demands for convenience, ease of use and access that will impel the necessary convergence of innovative e-payment products and capabilities. Regulation will channelize innovation, and competition to meet these demands will be consistent with international standards and best practises. Payments System also identifies the challenges very clearly:

1. Currently the number of non-cash transactions per person stands at just 6 per year.
2. A fraction of the 10 million plus retailers in India have card payment acceptance infrastructure – presently this number stands just at 1.1 million.
3. Of about six lakh villages in India, the total number of villages with banking services stands less than one lakh by end March 2011.Nearly 145 million households are excluded from banking. Over the last few years, there are significant improvements in terms of coverage; and with Direct Benefits Transfer (DBT) and Jan Dhan Yojana (PMJDY), number of households with bank account has improved. .

NPCI was set up in April 2009 with the core objective to consolidate and integrate] multiple systems with varying service levels into nation-wide, uniform, and standard business process for all retail payment systems. The other objective was to facilitate an affordable payment mechanism to have financial inclusion across the country.

In this regards NPCI has taken up new initiative of implementing "**Unified Payments Interface**" to simplify and provide a single interface across all systems.

Key aspects of this initiative are:

- SIMPLICITY- Paying and receiving payments should be as easy as making a call on mobile phone. With UPI system,anyone who has an account can send and/or receive money from their mobile phone with just an identifier unacquainted of bank/account details. The customer has to select "pay to" or "collect from" a "payment address" (such as Aadhaar number, Mobile number, Debit/Credit Card, virtual payment address, etc.) with a single click.
- INNOVATION- System is simple and layered so that innovations on both payee and payer side can happen with no change to core interface. This unified layer allows application providers to take advantage of enhancements in mobile devices and payment channels, provide integrated payments on new consumer devices, provide innovative user interface features, take advantage of newer authentication services, etc.
- ADOPTION– System is designed for scalability and mass adoption. This allows interoperability across payment channels, devices, and institutions for inclusive participation. Similarly, it allows full interoperability among multiple identifiers such as Aadhaar number, mobile number, and virtual payment address.
- SECURITY- System provides end to end resilient security and data protection. Considering self-service mobile applications, data capture is secured by encryption. Similarly, system allows a mechanism to pay and collect using valid virtual addresses without having to

reveal any bank/account details. System provides convenience by offering 1-click 2-factor authentication, risk scoring, protectionfrom phishing, etc.

- COST- Considering the fact that about 150 million smartphone users exist today and that number is expected to grow to 500 million in the next 5 years. The solution leverages the growing use of mobile phones as acquiring devices and uses virtual addresses instead of physical cards, thus reducing cost on both acquiring and issuing infrastructure.

The term "**Payment System Players**" (**PSP**) is used in this document to collectively define all RBI regulated entities under Payments and Settlement Act of 2007. These include banks, payments banks, PPIs, and other regulated entities.

The term "**Virtual Payment Address**" is used to depict an identifier that is uniquely mapped to an individual account using a translation service. In addition to Aadhaar number and Mobile number as global identifiers (mapped by NPCI), PSPs can offer any number of virtual addresses to customers to enable making and receiving payments.

Objective of UPI is **to offer an architecture and a set of standard APIs to facilitate the next generation online immediate payments, leveraging trends such as increased smartphone adoption, Indian language interfaces, and universal access to Internet and data.**

Following are some of the key features of the Unified Payments Interface.

1. The UPI is expected to further propel easy instant payments via mobile, web, and other applications.
2. The payments can be either sender (payer) or receiver (payee) initiated and are carried out in a secure, convenient, and integrated fashion.
3. This design provides an ecosystem driven scalable architecture and a set of APIs taking full advantage of mass adoption of smartphone.
4. Capabilities include virtual payment addresses, 1-click 2-factor authentication, Aadhaar integration, and use of payer's smartphone for secure credential capture.
5. It allows banks and other players to innovate and offer a superior customer experience to make electronic payments convenient and secure.
6. Supports the growth of e-commerce, while simultaneously meeting the target of financial inclusion.

3. UPI Architecture

This section of document covers the core features, high level architecture, key concepts, overall value proposition, a list of possible use cases and real world usage examples are provided to better understand the proposal. All technical details of the interface are covered in subsequent chapters.

3.1. Overview

1. INTEROPERABILITY
   A. Interoperability across payment channels, devices, and institutions for inclusive participation
   B. Allows full interoperability between multiple identifiers such as Aadhaar number, mobile number, and new virtual payment addresses
   C. Allows money to be transferred instantly across bank accounts / wallets in entire system
2. PUSH & PULL PAYMENTS
   A. Payments can be initiated by either sender (payer) or receiver (payee)
   B. Pay request: Theinitiating customer pushes funds to the intended beneficiary
   C. Collect request: The customer pulls funds from the intended remitter using virtual address
3. SINGLE CLICK 2FA
   A. UPI follows one click 2 factor authentication
   B. When a transfer is initiated using a smart phone, the device fingerprint (IMEI number for the device or any technical details unique to the device) isitself the first factor of authentication
   C. Second factor is a PIN number which has to be keyed in
4. IDENTIFIER
   A. Ability to integrate accounts/wallets with different banks
   B. Enables user to carry out all the payment transactions across multiple accounts and thus provides a single interface for all payments

3.2. Core Features

UPI provides the following core features via a set of APIs.

1. Ability to use personal mobile as the primary device for all payments, including person to person, person to entity, and entity to person.
2. Ability to use personal mobile to "**PAY**" someone (push) as well as "**COLLECT**" from someone (pull).

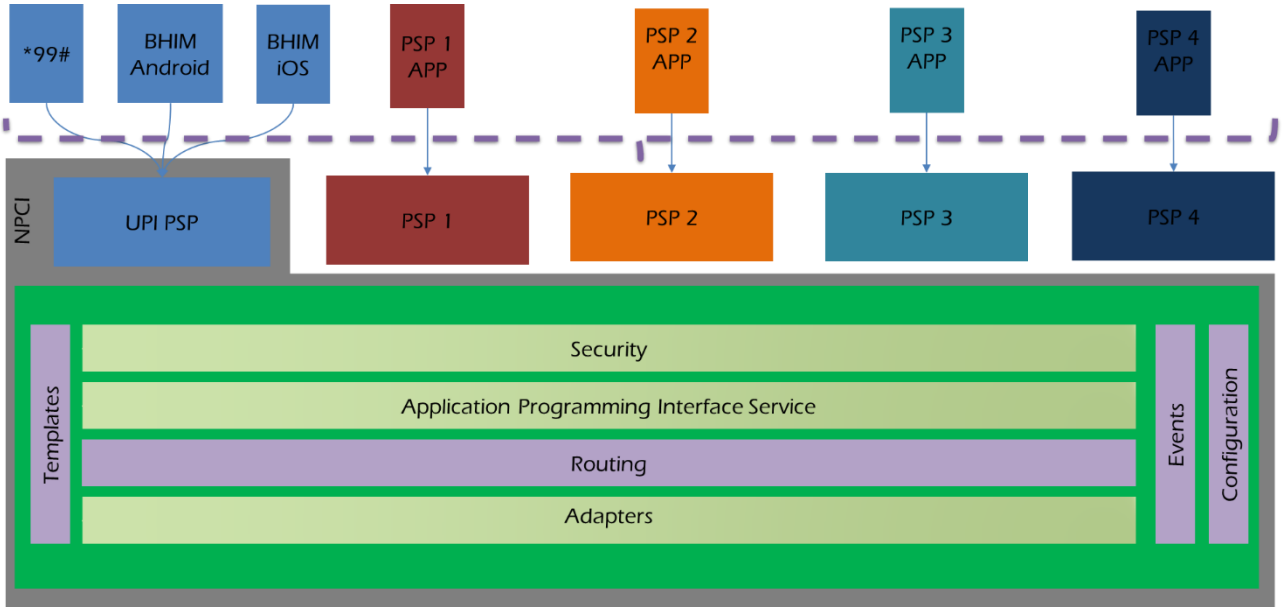
3. Ability to use Aadhaar number, mobile number, card number, and account number in a unified way. In addition, ability to pay and collect using "**Virtual Payment Addresses**" that are "aliases" to
4. Make payments by providing an address without having to ever provide account details or credentials on 3rd party applications or websites.
5. Ability to send "collect" requests to others (person to person or entity to person) with "pay by" date allows customers to pay at a later date without having to block the money in the account.
6. Ability to pre-authorize multiple recurring payments similar to ECS (utilities, school fees, subscriptions, etc.) with a one-time secure authentication and rule based access.
7. Ability of all PSPs to use a standard set of APIs for any-to-any push and pull payments.
8. Ability to use PSP provided mobile applications,which allow payments from any account, using any number of virtual addresses by providing credentials such as passwords, PINs, or biometrics.
9. Ability to use a fully interoperable system across all PSPs without having silos and closed systems.
10. Ability to make payments using 1-click 2-factor authentication just by using a personal phone and without any acquiring devices or physical tokens.

3.3. Authorization

Today, authentication and authorization are part of the same transaction flow and inline. But, in newer systems such as AEPS, use of third party authentication is followed where authorization is done within the banking system. Adopting 3rd party authentication and cardless payment scheme allows banks to reduce the overall issuance cost while still keeping authorization and account management within its control.

3.4. Architecture

The below diagram shows the overall architecture of UPI allowing USSD, smartphone, Internet banking, and other channel integration onto a common layer at NPCI. This common layer orchestrates these transactions and ensures settlement across accounts using systems such as IMPS, AEPS, NFS, e-comm etc. Usage of existing systems ensure reliability of payment transactions across various channels and also takes full advantage of all the investments so far.



Facilitates online real-time payments through the three payment APIs and a set of supporting APIs. All APIs are asynchronous in nature, meaning once the request is sent and acknowledgement received, the processing of response can be sent backseparately via corresponding response API.

The overall transaction processing is done through two external-facing interfaces, each of them having clearly defined responsibilities through application programming interface (API).

- Payment Service Provider (PSP) Interfaces has set of sub-interfaces which is used to communicate between UPI and PSP.
  - Routing and Processing
    - Routes and processes payment request (ReqPay) and request ispersisted in cache and DB. This will act as interface for all components of the system.
  - Resolver
    - Resolves the virtual address and sends the ReqAuthDetails message to, and receives RespAuthDetails from the corresponding PSP
  - Debitor
    - Interacts with the PSPs through ReqPay / RespPay (Debit) messages for debit.
  - Creditor
- Handles all functionalities regarding Credit.Internal System Interfaces
  - Provides interfaces to communicate directly to the underlying NPCI systems such as AEPS, andFRM.
    a. **AEPS:** Aadhaar Enabled Payment System is a payment service empowering a bank customer to use Aadhaar as his/her identity to access

his/her respective Aadhaar enabled bank account and perform basic banking operations like balance enquiry, cash deposit, cash withdrawal, remittances through a Business Correspondent, etc

b. **FRM:** It is used at network level.Designed and implemented as a Real-time Fraud Risk Monitoring and Management solution (FRM). This solution is envisaged as a value added service offered by NPCI to participanting members as a real-time monitoring tool for fraud detection and prevention

Across all application layers, REST API isused to design the integration interfaces makingthe system simple for a web-centric approach. This permits self-service for application developers and app users, provides API access to valuable enterprise resources, encourages collaboration among internal and external resources, and increases the value of current customers by offering existing services via new platforms and devices.

3.5. Core Domain Entities

Every payment request has the following core elements:

1. Payer and payee account and institution details for routing and authorization
2. Authentication credentials (, UPI-PIN, Biometrics, CVV, etc. as required for debit, can be bank provided or 3rd party provided such as UIDAI)
3. Transaction amount
4. Transaction reference
5. Timestamp
6. Metadata attributes such as location, product code, mobile number, device details, etc. as required.

Out of the above, items 1 and 2 are critical to be abstracted so that single architecture can handle current and futuristic scenarios of "any payment address" using "any trusted authentication scheme". Following sections describe these concepts in detail.

3.5.1. Payment Address

Every payment transaction must have source (payer) account details (for debit) and destination (payee) account details (for credit). At the end, before the transaction can be completed, these must be resolved to an actual account number/ID.

"**Payment Address**" is an abstract form to represent a handle that uniquely identifies account details in a"normalized" notation. In this architecture, all payment addresses are denoted as "**account@provider**" form. Address translation may happen at provider/gateway level or at NPCI level. Address should only contain a-z, A-Z, 0-9, . (dot), - (hyphen).

Examples of normalized (fully qualified) payment addresses are:

- IFSC code and account number combination, resolved directly by NPCI, is represented as **account-no@ifsc-code.ifsc.npci** (e.g. 12345@HDFC0000001.ifsc.npci)
- Aadhaar number, resolved directly by NPCI using existing Aadhaar to bank mapper,is represented as **aadhaar-no@aadhaar.npci** (e.g. 234567890123@aadhaar.npci)
- Mobile number, resolved directly by NPCI using proposed mobile to accountmapper, is represented as **mobile-no@mobile.npci** (e.g. 9800011111@mobile.npci)**(It is for future use)**
- Aadhaar and IIN number, resolved directly by NPCI, is represented as **Aadhaar-no@IIN-no.iin.npci** (e.g. 200012955794@607152.iin.npci)
- RuPay card number, resolved directly by NPCI, is represented as **card-no@rupay.npci** (e.g. 1234123412341234@rupay.npci)**(It is for future use)**
- When bank itself is the PSP, any account identifier, resolved directly by bank as the PSP, is represented as **account-id@bank-psp-code** (e.g. 12345678@icici)
- A PPI provider issued card number, resolved directly by PPI provider, is represented as **ppi-card-no@ppi-psp-code** (e.g. 00001234678 9@myppi)**(It is for future use)**
- A user id provided by PSP, resolved directly by that PSP, is represented as **userid@psp-code** (e.g. joeuser@mypsp)
- A one time or time/amount limited tokens issued by a PSP, resolved directly by that PSP, is represented as **token@psp-code** (e.g. ot123456@mypsp)

Provider is expected to map the payment address to actual account details at appropriate time. Providers who provide "virtual addresses" should expose the address translation API (see later sections for API details) for converting their virtual addresses to an address that can be used by NPCI. Unlike current systems with fixed length account numbers and provider numbers (BIN, IFSC, etc.), payment addresses are strings of sufficient length to ensure it accommodates future possibilities.

3.6. Authentication

Traditionally, payment account provider themselves provide the authentication scheme. Account management (KYC, opening account, managing transactions, etc.) was tightly coupled with internal authentication schemes. Authentication schemes separately evolved, as new payment channels evolved. While numeric or alpha-numeric PIN/Passwords is the dominant authentication factor, different PINs were issued for different channels (Internet PIN, ATM PIN, Mobile PIN, etc.). In addition, OTP based authentication is used these days to offer 2-FA authentication schemes.

Account management including KYC should be loosely coupled with authentication. Aadhaar authentication provides trusted external authentication scheme and is already used today within the payment systems. Micro-ATMs (handhelds with biometric sensors) used by BCs take advantage of Aadhaar authentication via NPCI to conduct payment transactions.

Digital Signatures, including Aadhaar enabled e-sign, can also play an important role to identify the authenticity of the request and bring out new ways of issuing e-Cheques, ECS mandates, and other payment instruments.

UPI enables multiple authentication schemes (account provider as well as trusted 3rd party like UIDAI's Aadhaar authentication), without tightly coupling with account provisioning and management. This allows future one or multi-factor authentication scheme(s) to be plugged into the architecture, as long as account providers allow such trusted external authentications. In UPI one of the authentications is performed by the PSP, while the other is performed within the domain of the account provider. In addition, strong mobile binding and finger printing allows mobile as an authentication factor to be used within the system.

3.7. Aadhaar

3.7.1 Aadhaar System

Unique Identification Authority of India (UIDAI) has issued over 80 crore Aadhaar numbers to Indian residents. It has become an accepted form of identity across the country for various government and non-government agencies. It has been approved as an identity document by various regulators including RBI, SEBI, etc for KYC. Aadhaar provides an online authentication service for electronic verification of identity which is being used in the banking sector

3.7.2 Aadhaar Authentication

Aadhaar authentication is the process wherein Aadhaar number, along with other attributes, including biometrics, are submitted online via an API to the UIDAI system for its verification on the basis of information or documents available with it. Authentication module handles online resident authentication from various Authentication User Agencies (AUA).

3.7.3 NPCI Central Mapper

NPCI maintains an association between customer's Aadhaar number and Bank identifier. This central repository can be used to route payment instructions based on Aadhaar number.

The Aadhaar Payments Bridge System (APBS) uses NPCI's central mapper as a part of National Automated Clearing House (NACH) to enableGovernment user-departments to electronically transfer subsidies and direct benefit transfers to individuals on the basis of their Aadhaar number. APB system enables payments to be credited to end beneficiaries' Aadhaar-enabled accounts (AEA) on the basis of Aadhaar number being unique identifier. Hence the Aadhaar number becomes a payment address.

UPI, IMPS, and National Unified USSD Platform (NUUP) can take advantage of Central Mapper for fetching and routing their payments. Hence having such a common repository can create a great process value add, for overall payment ecosystem and as a consequence to the end customer.

3.7.4 Aadhaar Payment Bridge System (APBS)

The APBS facilitates the processing of payments from the Government departments received via the sponsor banks (assigned bank), and subsequently routing of the payments to the beneficiary's bank. The beneficiary's bank has the Aadhaar number mapping to the beneficiary's bank account number to credit the amount in the end beneficiary's account.

4. Sample Use Cases

This chapter provides a set of examples of usage of this unified interface. All examples fall into two categories - "Direct Pay" to push money and "Collect Pay" to pull money from one account to another.

Purpose is to illustrate a set of real life use cases and not enumerate all possible usages. It is expected that PSPs and user ecosystem will innovate and find more interesting usage scenarios for this simple and unified payments interface.

4.1 Sending money to relative

A migrant worker, Ram, living in Mumbai having an account with State Bank of India, using his low cost Android phone, can send money to his wife, Laxmi, in a village via her Aadhaar number with single click.

Here is how it works:

1. Ram gets an account created in SBI using paperless Aadhaar e-KYC option. He also provided his mobile phone during application.
2. His wife, Laxmi, has also opened an account in Bank of India using Aadhaar e-KYC.
3. If he has not obtained an MPIN, he can use *99# (NPCI USSD service accessible across country) on his phone to set first time MPIN using his RuPay card and expiry.
4. He downloads SBI mobile application and uses MPIN to set his profile up.
5. SBI mobile application is now integrated with unified payments interface at NPCI and offers convenient features to send money, collect money, and manage integrated address book.
6. He adds his wife's Aadhaar number to his address book. No other information such as IFSC code, etc. are required to be stored for his wife.
7. In the mobile application, witha single click on his address book entry of his wife, he enters an amount and hits send. SBI application allows him toremember the Aadhaar number for future use.

Behind the scene, whenever money is sent, SBI application does the following:

1. Validates user and debits his account.
2. Uses Unified Payments Interface and initiates a "Pay" transaction with "payee" address to be simply "Aadhaar number" of Laxmi.
3. NPCI Unified Payments Interface layer looks up the Aadhaar mapper and translates the destination address to bank identification number and routes the transaction to destination bank via AEPS.
4. Destination bank uses their system to credit the amount to the Aadhaar linked account and sends confirmation back to NPCI.
5. NPCI confirms the credit back to SBI application.

6. SBI application pushes a notification to the mobile device confirming credit.

4.2 Collecting money from friend

Two friends Ram and Shyam go out for dinner and Ram pays the bill. They agree to split the bill in half. Ram wants to collect half of the bill from Shyam and uses his android mobile phone to do so and requests Shyam to pay in a week's time.

Here is how it works:

1. Ram logs on to his Punjab National Bank (PNB) mobile app.
2. Ram initiates collect request by providing Shyam's address which, in this case is sham.444@icici
3. Ram enters the amount to be paid by Shyam.
4. Shyam gets a message on his phone stating that there is a collect request from Ram for a given amount. Shyam's PSP also shows Ram's full name as in the Aadhaar system which was verified during Ram's on boarding.
5. Shyam is in a meeting, so he snoozes the request and decides to attend it later. Since the request had specified that it can be paid within a week, Shyam's mobile application allows such snooze and reminder features.
6. His mobile application reminds him after the snooze period.
7. He accepts the collect request, provides biometric credential using his biometric enabled smartphone, and authorizes the payment.
8. Ram receives the confirmation of payment.

This is how it works behind the scenes:

1. PNB sends the collect request to NPCI with Ram's details and Shyam's address.
2. Since the payer address (shyam.444@icici) is a "virtual payment address", NPCI invokes the PSP (in this case ICICI) authorization and address translation API.
3. NPCI routes the request to ICICI.
4. ICICI takes the requests and resolves Shyam's address.
5. ICICI sends the request to Shyam's mobile.
6. Shyam accepts the message, provides credentials, and ICICI debits the money from his account.

7. ICICI confirms the debit back to NPCI.
8. On receiving the debit confirmation, based on the Ram's details, NPCI processes the credit request to PNB
9. PNB credits Ram's account and responds to NPCI.
10. PNB pushes a notification to Ram's mobile number confirming the credit.

4.3 Buying on an e-commerce site

Sita is browsing myCartDeal for a deal on furniture. She finds a good offer on a leather sofa that costs Rs.40000/-. She logs in to myCartDeal and places the order.

Since it is a custom made furniture, myCartDeal allows her to pay 70% as advance during order and remaining 30% on delivery. During checkout, she chooses "Collect Pay" option and provides her virtual address provided by her PSP, Yes Bank, to make advance payment

Here is how it works:

1. Sita enters her virtual address on the myCartDeal site during checkout process.
2. Since it is a custom made furniture, myCartDeal wants to collect only 70% as advance.
3. They initiate the first "collect" request with Rs.28000/- as amount during checkout.
4. They send the collect request along with order number to NPCI via their PSP.
5. NPCI routes the request based on Sita's virtual address (sita.1234@yesbank) to her PSP which happens to be Yes Bank.
6. Yes Bank application sends a notification to Sita's mobile.
7. Sita accepts the collect request by providing her credentials.
8. Yes Bank debits the specified amount (Rs.28000/-) within the collect request from her account and confirms the debit back to NPCI.
9. NPCI notifies myCartDeal's PSP about the successful payment and myCartDeal confirms the order.
10. Once the furniture is ready, myCartDeal creates a new collect request with remaining amount (Rs.12000/-) with a "pay by" date and sends it to Sita's PSP.
11. Sita snoozes the request and leaves it in her mobile application's inbox since it needs to be paid only after delivery.
12. Once the furniture is delivered, Sita clicks on her inbox item (second pending collect request) and authorizes the payment for Rs.12000/-.

4.4 Buying railway ticket on IRCTC application

Abdul wants to buy train ticket from Mumbai to Delhi. He logs into IRCTC and enter the travel details. IRCTC initiates the collect request via its PSP using the virtual payment address which was part of Abdul's profile, collects money from him and issues ticket.

Here is how it works:

1. Abdul logs into his IRCTC account and provides the travel details.
2. Abdul has already provided his payment address to IRCTC as part of the profile.
3. He had used his PSP application to create a new virtual address "abdul2014.irctc@mypsp".
4. His PSP allows a feature to limit specific addresses only for collect from a specific merchant with a maximum amount limit!
5. Since itis just a virtual address (merchant bound and amount limited), no one else can use it to collect money from him.
6. This address is also bound (within Abdul's mobile app) to a default bank account.
7. With a single click buy (without entering any card or other details and no redirections on web pages), IRCTC initiates collect pay to NPCI via their PSP.
8. NPCI sends the payment address to the PSP ("mypsp" in this case) where Abdul is registered with.
9. The PSP translates Abdul's Payment address and sends notification to his mobile to capture credentials.
10. Abdul enters his bank authentication credentials on his mobile device and does a single click authorization.
11. His PSP responds to NPCI with the actual account details which was bound to the virtual address along with encrypted authentication credentials.
12. NPCI sends the debit request to Abdul's bank that was sent back in response.
13. On successful response, NPCI sends credit request to IRCTCs bank account (which was part of collect request)
14. On successful response both IRCTC's PSP and Abdul are notified on the sameand ticket is issued.

4.5 Using for bill payments and insurance premium collections

Collect pay mechanism has enabled Sita's phone company and insurance company to send her the bill/premium collection request in an automated fashion to her virtual address registered with her bank's mobile application. Interestingly, with the unified interface having the ability to specify the "pay by" date, these companies can send these bills several days ahead of time to Sita and allow her to pay any time within the request expiry period. Her mobile phone smartly sets reminders based on request metadata and allows her to pay these on time all via a simple 1-click interface on her smartphone.

When ECS like auto authorizations are used, collect pay mechanism can be further simplified by providing a time limited (say, for 12 months) and amount limited (say, less than a particular amount) electronic mandate with PSP. In such cases, customers can be provided with the convenience of one time authorization instead of authorizing every time.

4.6 Collecting money for Monthly Phone Bill

Corporate (BSNL) wants to collect monthly phone bill amount from Ram (Payer PSP - SBI). So corporate initiates a collect request to Ram using the Corporate PSP (Payee PSP - ICICI).

Behind the scenes:

Before the collect request is being initiated by corporate PSP (BSNL-Payee) to Ram (Payer). Ram/Corporate PSP has to create an UPI-Mandate and provide RAM'S "UMN" to corporate.

Here is how it works:

1. Corporate (Payee PSP – ICICI) initiates the collect Request to UPI for the bill amount.
2. UPI forwards the request to Ram (Payer PSP – SBI).
3. Payer PSP sends the response auth with mandate cred block to UPI.
4. UPI forwards the same XML to Remitter bank to debit Ram's account.
5. With the mandate cred block details, remitter bank debits the customer account and sends response to UPI.
6. UPI, then initiates credit request to Beneficiary bank and the corporate account is credited
7. Beneficiary bank sends the credit response back to UPI.
8. UPI sends the transaction confirmation message to Ram (Payer PSP - SBI).

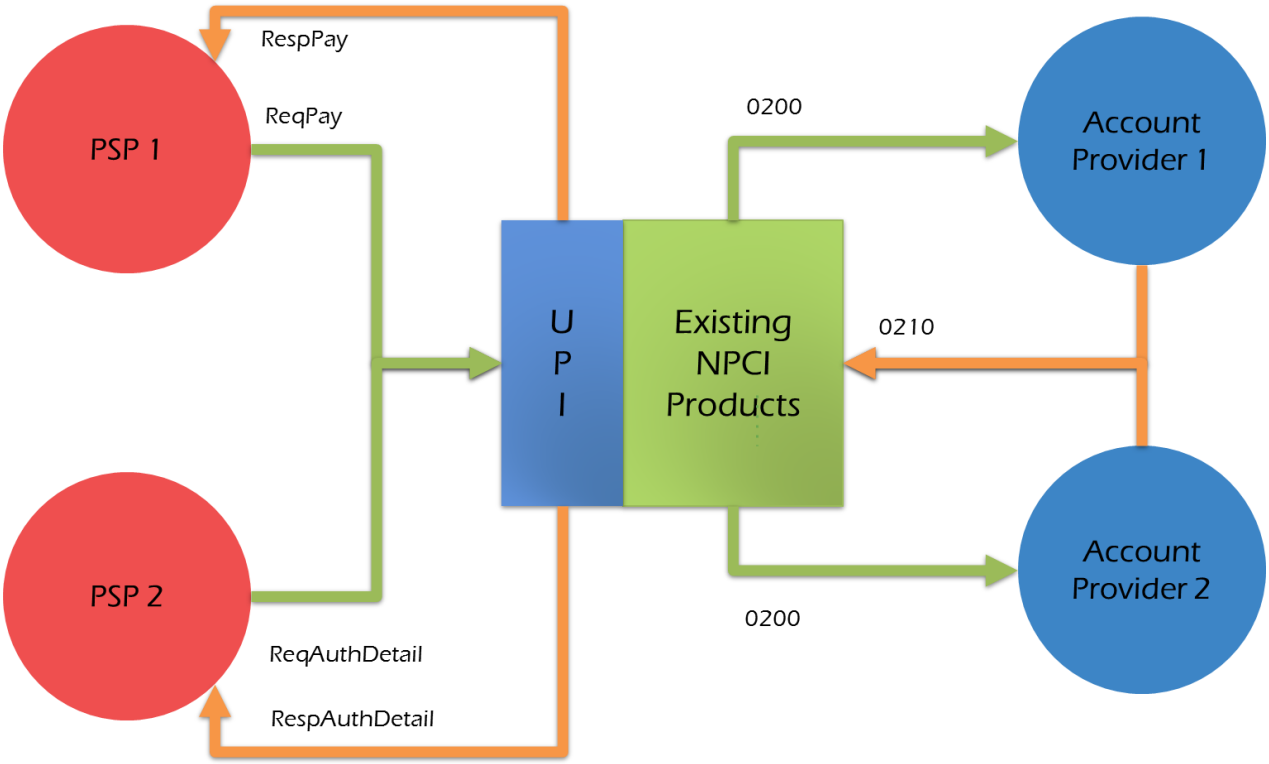
5. Design

This chapter provides the high level technical specifications for various types of payments that can be done through the UPI, and the corresponding high level flows.

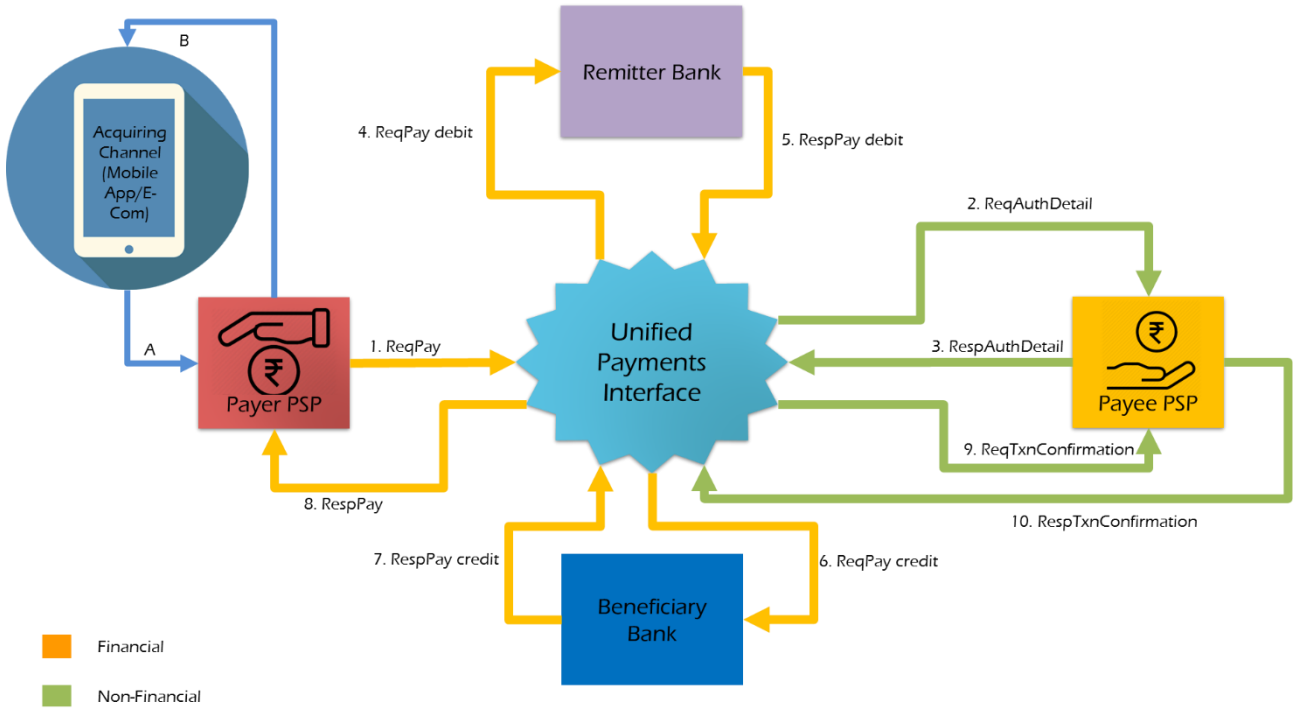
5.1 UPI - Message Flow

Diagram below depicts a general scenario where PSP1 is doing a "Pay" or "Collect" to PSP2 address and initiating account under PSP1 is mapped to Account provider 1 and PSP2's address is mapped to Account Provider 2.

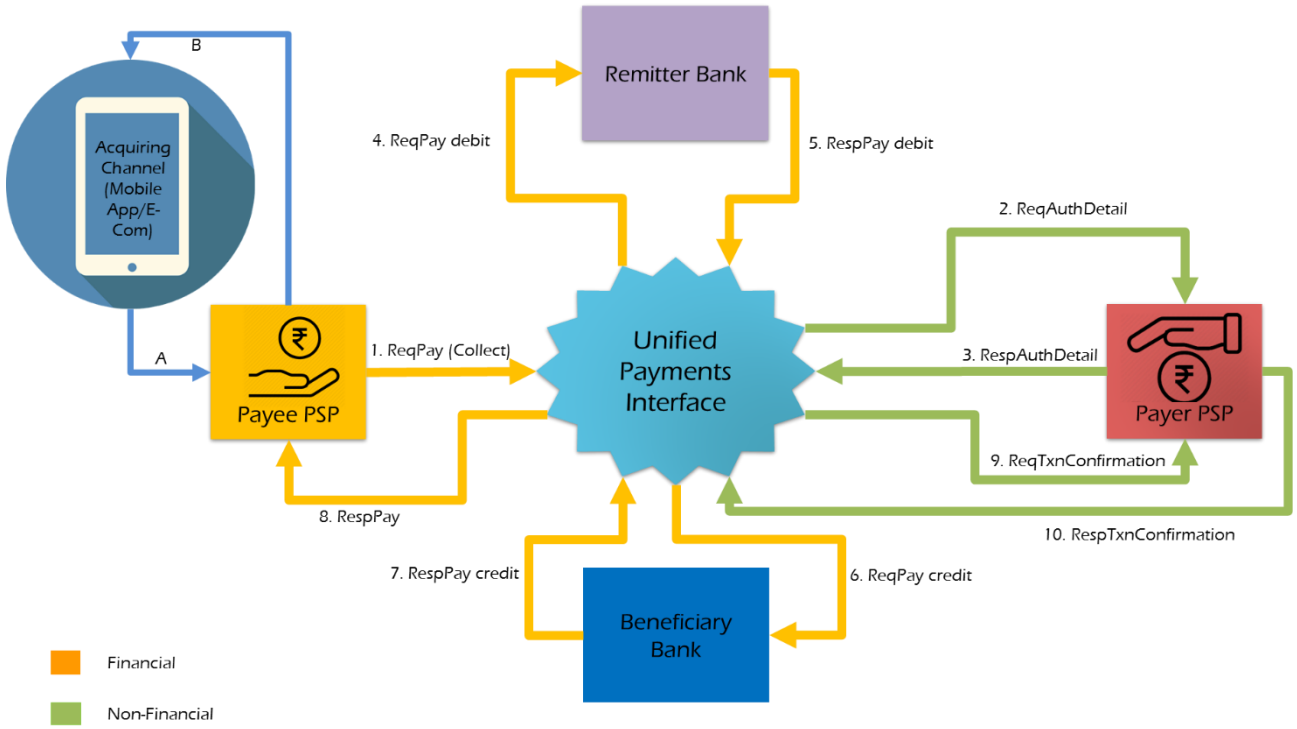
All Unified interface APIs are called using XML over HTTPS whereas all APIs behind the existing systems at NPCI are done over ISO 8583 Messages (0200/0210).



5.1.1 Pay Flow



5.1.2 Collect Flow



5.2 APIs at a Glance

All APIs are asynchronous in nature meaning once the request is sent, response is sent back separately via corresponding response API. This allows same APIs to be used for instant payment as well as delayed payments. This also allows APIs to scale without having to wait in a blocking mode. Callers are expected to call the API with a unique transaction ID for which response is sent via a response API exposed by the caller.

All APIs are expected to work in asynchronous mode. This allows the response to API call to return to the caller immediately after queuing the request. All request-response correlation must be done via the transaction ID set by the originating point. Exactly same set of APIs are exposed by NPCI and PSPs.

All APIs must be exposed via HTTPS using XML input and output (as defined in next chapter). When calling APIs via a synchronous protocol like HTTP, listening server should push the message into a queue and send an acknowledgement response.

5.3 Payment API

This API is the primary API that the PSPs will initiate to NPCI. Single API will be used for both Direct Pay and Collect Pay transaction processing. The PSPs maintain the PSP specific payment addresses which can be resolved to any of the common global addresses (Aadhaar number, Mobile number, Account + Provider ID) that NPCI can understand.

In the Direct Pay request to NPCI, the Sender PSP will provide the complete details of the sender and payment address of the Receiver. NPCI will fetch the Receiver details from the Receiver PSP. Once NPCI has complete details to process the financial transaction, the debit and credit will be processed through the online products like IMPS, AEPS etc.

In the Collect Pay request to NPCI, the Receiver PSP will provide the complete details of the Receiver and payment address of the Sender. NPCI will fetch the Sender details from the Sender PSP. Once NPCI has complete details to process the financial transaction, the debit and credit will be processed through the online products like IMPS, AEPS etc.

5.4 Authorization & Address Translation API

This API is used to authorize a payment and translate PSP specific payment addresses to any of the common global addresses (Aadhaar number, Mobile number, Account + Provider ID) that NPCI can understand. PSPs may offer one or more virtual addresses (multi use or one time use with time and/or amount limited) to customers. This allows customers

to simply provide such virtual (tokenized) address to others (individuals, entities, etc.) without having to reveal actual account details.

"ReqAuthDetails" API is called to translate PSP address and obtain appropriate authorization details. "RespAuthDetails" API is the response call back interface to return the required details. After processing the ReqAuthDetails API, PSP should send response to the authorization by calling the "RespAuthDetails" API and send to NPCI.

5.5 Security Considerations

For data security, the following classes of information are defined:

1. **Sensitive Data** - Data such as PIN, passwords, biometrics, etc. are not to be stored and should be transported in encrypted form.
2. **Private Data** - Data such as account number. This information may be stored by the PSP, but only in encrypted form.
3. **Non-Sensitive data** - Name, transaction history (amount, timestamp, response code, location, etc.) can be stored in unencrypted form.

5.5.1 Identity & Account Validation

The following identity data needs to be validated in the messages to ensure trust in the system. In case the data has not been validated, it must be indicated:

Identity Data	Validated By	When	How
Mobile Number	PSP	Customer Registration	Using OTP
	Issuer	Account Registration	During first transaction
Aadhaar Number or PAN number	PSP	Customer Registration	Aadhaar authentication or PAN card verification
Customer Name	PSP	Customer Registration	Aadhaar e-KYC or Bank or PAN card verification or any other KYC verification

Identity Data	Validated By	When	How
Account Details - Number, Account Ownership,	PSP using the issuer credentials (captured via common library)	Every time a payment account is added	During first transaction

5.5.2 Protecting Account Details

- PSP is mandated to use a secure protocol when transmitting sensitive data such as account details from the device to the PSP server.
- PSPs is mandated to safeguard account information within PSP system as per regulatory and the payment card industry (when storing card details) compliance standards.

5.5.3 Protecting Authentication Credentials

- Trusted common library for credential (UPI-PIN/ATMPIN/Biometrics etc) capture is provided by NPCI. This library needs to be integrated with PSP application. Please refer Annexure A for Common library specifications.
- Authentication credentials are captured and encrypted within the common library. PSP should not capture issuer specific authentication credentials outside the common library.
- The encrypted credentials are base64 encoded by the common library and given back to PSP application for subsequent transports through UPI.
- PSP should not log or store encrypted credentials within any permanent storage.

5.5.4 Protecting against Phishing

Following techniques may be used to protect against phishing:

- Payer's PSP application should mandatorily show verified payee's name to the payer during a collect request.
- Payee's PSP application should mandatorily send verified payee's name to NPCI as part of the collect request.

- In the case of payee being a whitelisted entity, payer's PSP should show the whitelisting information (Name, logo, URL, etc.) which is available within the collect request. This whitelisting information is populated from NPCI's central rating system.
- PSP should ensure that their applications have anti phissing protection. PSP should also have adequate awareness programs for their customers.

Whenever a collect payment request comes, payer's PSP application should show the KYC information of the requester, whitelisting information from the central system, and transaction reference number (sales order number, transaction note, etc.) to help payer make the decision to accept or reject the request.

Message Security, Trust, and Non-Repudiability

- Every message within the unified system must be digitally signed.
- Every message has unique transaction ID (that spans across the organizations for same transaction) and unique message ID for every request-response pair.
- All APIs must be done over a secure channel (HTTPS).
- Auditing transaction (no sensitive data) data as per the regulatory requirements.

5.6 Direct Pay (Sender/Payer initiated)

In this flow, the payer initiates a payment transaction, while specifying the recipient. There are 2 sub-flows – when the sender is an individual, or a system (presumably a company).

5.6.1 Person Initiated

The sender uses an application to send money to a receiver by providing sender credentials and receiver/beneficiary "address". For ex. to pay a friend via a mobile banking application.

5.6.2 System Initiated

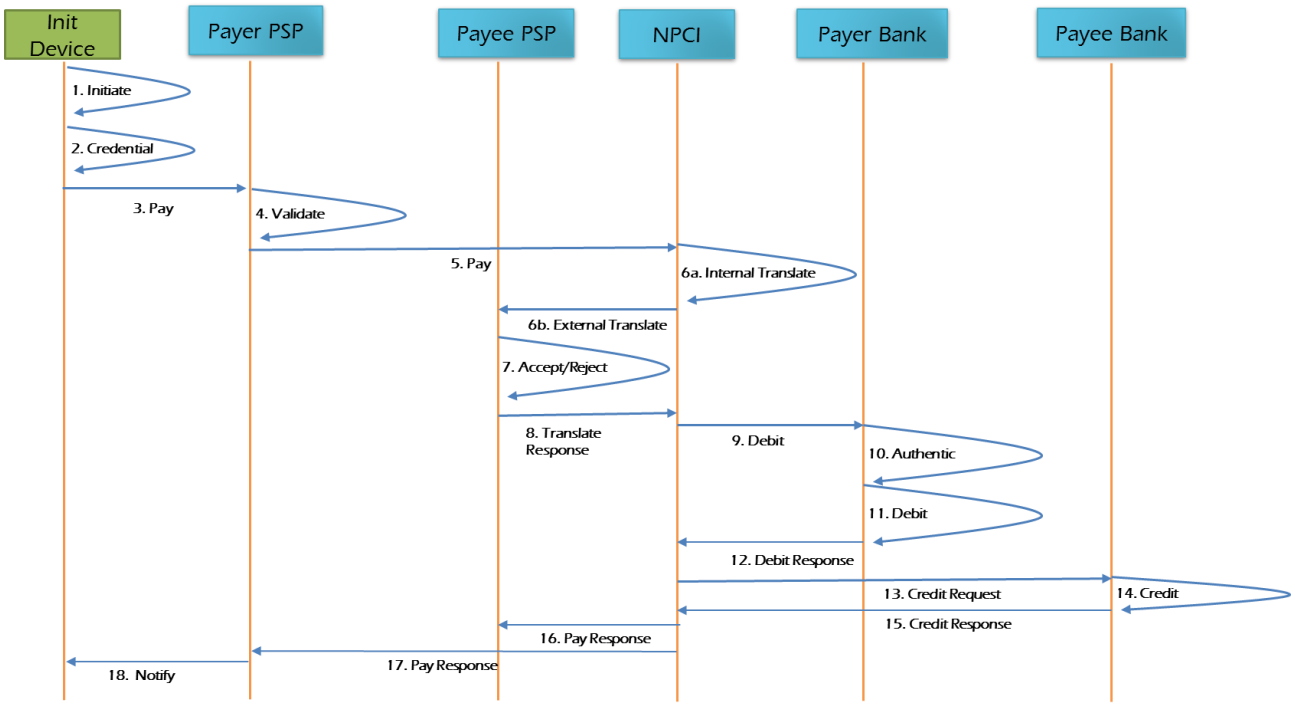
The sender system initiates a payment, using a digitally signed request. For ex. The system generates a daily commission payment to agents.

5.6.3 Transaction Flow

1. Payer initiates transaction through his PSP application in his Device.
2. Payer provides authentication credentials in his Device.
3. The Payer Device initiates the Pay request to Payer PSP system.

4. Payer PSP validates the Payer details and validates the first factor authentication.
5. Payer PSP sends the pay request to NPCI.
6. NPCI resolves the Payee Address in the following two ways
    a. If the Address has global identifiers (Mobile # and Aadhaar #) then the Payee Address is resolved by NPCI central Mapper.
    b. If the Address has virtual address offered by Payee's PSP, then NPCI will send the request to Payee's PSP for address translation.
7. In case of 6a, the Payee PSP accepts or rejects the request based on the rules set at its end.
8. In case of 6b, on accepting the Pay request, Payee PSP populates the Payee detailsand responds to NPCI.
9. NPCI sends the debit request to the debit account provider.
10. Account provider authenticates the Payer based on the credential provided.
11. Account provider debits the Payer account.
12. Account provider sends Debit response to NPCI.
13. NPCI sends the Credit request to the credit account provider.
14. Account provider credits the account based on the Payee details.
15. Account provider sends Credit response to NPCI.
16. NPCI sends Pay response to Payee PSP.
17. NPCI sends pay response to Payer PSP.
18. Payer PSP notifies payer.

The following diagram illustrates the above flow.



5.6.4 Multiple Pay Scenario

Payer can initiate a single pay transaction to one or more payee's account.

SUCCESS

1. Customer initiates ReqPay for Rs100 to UPI via payer PSP to two payee's paying Rs50 each.
2. Enters both payees' addresses.
3. Customer authenticates the transaction using UPIPIN.
4. UPI sends ReqAuthDetail to Payee's PSP for address resolution.
5. Payee's PSP responds back to UPI with the resolved address.
6. After that, UPI sends ReqPay debit to remitter bank and respond with RespPay Debit.
7. Remitter bank will debit Rs 100 from the customer account.
8. Then UPI initiates ReqPay credit to both beneficiary banks to credit each payee.
9. Both the Beneficiary banks respond to UPI as RespPay credit with SUCCESS message
10. Both payees are credited with Rs50 each.
11. UPI sends final RespPay message to payer PSP.
12. UPI sends ReqTxnConfirmation message to Payee's PSP.
13. Payee's PSP responds back to UPI.

FAILURE

1. Customer initiates ReqPay for Rs100 to UPI via payer PSP to multiple Payee's paying Rs50 each.
2. Enters both payees' addressesCustomer authorises the transaction using UPIPIN.
3. UPI sends ReqAuthDetail to Payee's PSP for address resolution for both the payees.
4. Payee's PSP responds back to UPI with the resolved address.
5. After that, UPI sends ReqPay debit to remitter bank and respond with RespPay Debit.
6. Remitter bank will debit Rs100 from the customer account.
7. Then UPI initiates two ReqPay credit to beneficiary banks to credit both payees.
8. Beneficiary bank credits one payee's account and responds with SUCCESS message and another beneficiary bank responds with FAILURE message due to some issue with one of the beneficiary bank.
9. Then UPI initiates debit reversal to remitter bank as PARTIAL debit for the FAILURE payee.
10. Remitter bank credits back the customer with PARTIAL amount of Rs50.
11. UPI sends final RespPay message to payer PSP.
12. UPI sends ReqTxnConfirmation message to Payees PSP.
13. Payee's PSP responds back to UPI.

Partial scenario is where Payer pays for multiple Payee but amount is credited for only one Payee, hence the above scenario becomes partial.

5.6.5 Failure Scenarios

This section explains how various failure scenarios are handled during the PAY transaction. The transaction flow mentioned above will be considered while describing the failure scenarios.

Failure at step 18 - PSP unable to notify the Payer:

In this scenario, when the PSP is not able to notify the end customer on the status of the transaction, a mechanism has to be put in place by the PSP to notify the customer at a later stage. This can be achieved by PSP reinitiating the notification message to customer or by providing the customer an option to check the status of the transaction through his application, or by providing a list of all transactions (with status) in the application.

Failure at step 16/17 - Response from NPCI does not reach Payee/Payer PSP:

In this scenario, when the response sent by NPCI does not reach Payer/Payee PSP, the PSPs should have a mechanism to initiate a Check Status API to know the status of the transaction. The PSP can only initiate the Check Status API to NPCI after a time period of Transaction expiry time (see expireAfter Attribute) + 90 seconds.

Failure at step 15 - Response from Payee bank does not reach NPCI:

In this scenario, when the response sent by Payee bank does not reach NPCI, this transaction will be considered as declined/failedand declined. Response will be sent to Payee and Payer PSPs. NPCI initiates the reversal message for such transactions.

Failure at step 15 - Declined Response from Payee bank to NPCI:

In this scenario, when the Payee bank responds with a declined response to NPCI, NPCI will send the reversal request to Payer bank and respond to Payee and Payer PSPs with declined response.

Failure at step 13 - Payee bank is not available to NPCI:

In this scenario, when the Payee bank is not available to NPCI, NPCI will send the reversal request to Payer bank and respond to Payee and Payer PSPs with declined response.

Failure at step 12 - Declined Response from Payer bank to NPCI:

In this scenario, when the Payer bank responds with a declined response to NPCI, NPCI will respond to Payee and Payer PSPs with declined response. No credit request will be initiated to Payee bank.

Failure at step 12 - Response from Payer bank does not reach NPCI:

In this scenario, when the response sent by Payer bank does not reach NPCI, NPCI will timeout the transaction. NPCI will respond to Payee/Payer PSP's with timeout response.

Failure at step 9 - Payer bank is not available to NPCI:

In this scenario, when the Payer bank is not available to NPCI, NPCI will respond to Payee and Payer PSPs with declined response.

Failure at step 8 - Declined Response from Payee PSP to NPCI:

In this scenario, when the Payee PSP responds with a declined response to NPCI, NPCI will respond to Payer PSP with declined response.

Failure at step 8 - Response from Payee PSP does not reach NPCI:

In this scenario, when the response sent by Payee PSP does not reach NPCI, NPCI will wait for the response till the timeout period. Payee PSP may have a mechanism to re send the

response within the timeout period. If NPCI do not receive response within the timeout period, NPCI will timeout the transaction and respond to Payer PSP's with a timeout response.

Failure at step 6 - Payee PSP is not available to NPCI:

In this scenario, when the Payee PSP is not available to NPCI, NPCI will respond to Payer PSP with declined response.

Failure at step 5 - NPCI is not available to Payer PSP:

In this scenario, when NPCI is not available to Payer PSP, Payer PSP will have a mechanism to re initiate the Pay request to NPCI.

For a failed/declined preapproved transaction remitter Bank/PSP should reverse the debit on receiving the declined response from NPCI

5.7 Collect Pay (Receiver/Payee Initiated)

The UPI allows payment requests to be initiated by the recipient. Common use cases for this include personal payments, such as expense sharing; merchant payments; billing, etc.

5.7.1 Remote Collect

1. Payee/Receiver (person or entity) triggers the request without capturing sender credentials
    a. Uses a USSD or Smartphone to do push authorization on sender phone
    b. Eliminates any credential entry on external apps
    c. Allows single click one or two factor (mobile + PIN, mobile + biometrics, etc.) on a "trusted application" (bank/NPCI app, etc.)
    d. Sender's phone becomes secure terminal for credential entry,
2. Examples
    a. Kirana store employee uses his/her phone app to "collect" by entering customer's mobile number
    b. Car service agency application "collecting" payment via mobile number for home deliveryof the car.
    c. Magazine subscription application requesting authorization for subscription renewal

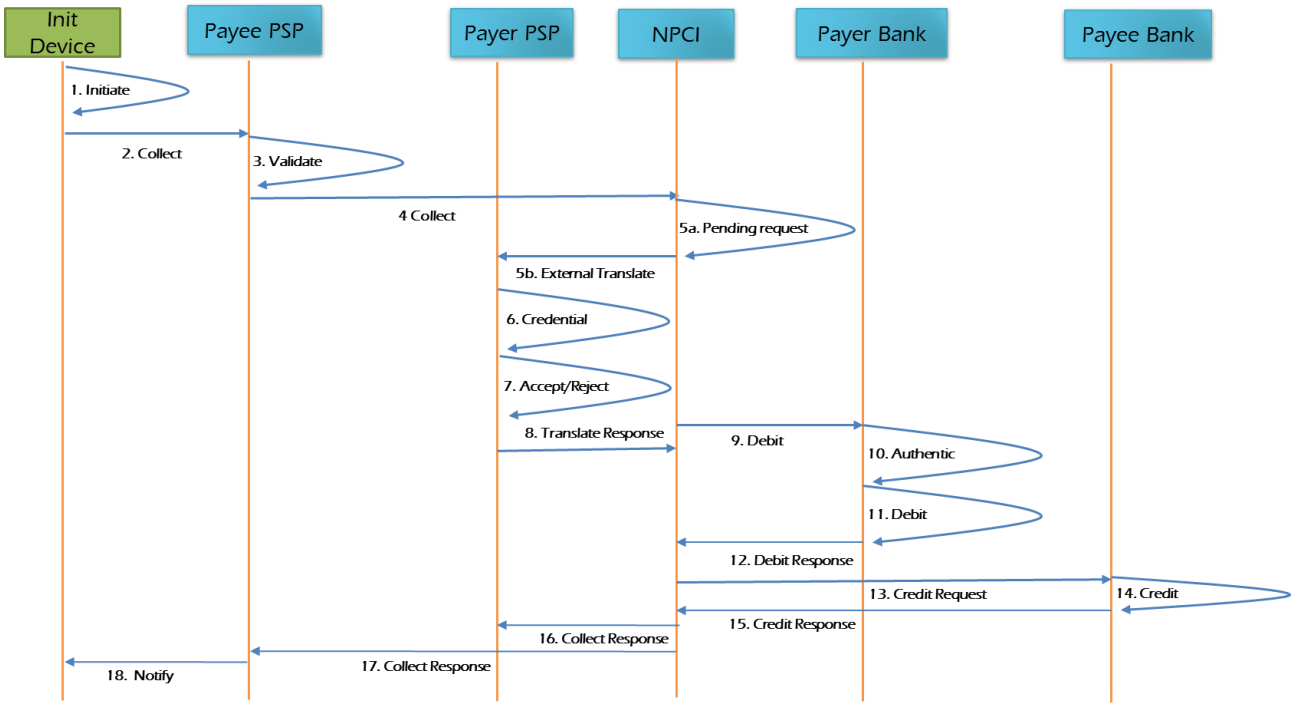
5.7.2 Local Collect (Proximity Payments)

The merchant charges a customer at the point of sale. The merchant system captures the payer's payment address, and sends a request to pay the bill amount. The request is approved by the payee using a smart phone application. Local exchange of encrypted credential is not currently supported by UPI.

5.7.2.1 Transaction Flow

1. Payee initiates transaction through his PSP application in his Device.
2. The Payee Device initiates the Collect request to Payee PSP system.
3. Payee PSP validates the Payee details and validates the first factor authentication.
4. Payee PSP sends the Collect request to NPCI.
5. NPCI resolves the Payer Address in the following two ways
    a. If the Address has global identifiers (Mobile # & Aadhaar #) then PSP requests NPCI for pending messages via API against a given mobile number or Aadhaar number.
    b. If the Address is virtual address offered by Payer's PSP, then NPCI will send the request to Payer's PSP for address translation.
6. In case of 5b, the Payer PSP accepts or rejects the request based on the rules set at its end.
7. In case of 5b, on accepting the Collect request, Payer PSP initiates a request to Payer device to enter his authentication credentials. Payer provides authentication credentials in his Device.
8. In case of 5b, The Payer PSP populates the Payer details and responds to NPCI.
9. NPCI sends the debit request to the debit account provider.
10. Account provider authenticates the Payer based on the credential provided.
11. Account provider debits the Payer's account.
12. Account provider sends Debit response to NPCI.
13. NPCI sends the Credit request to the credit account provider.
14. Account provider credits the account based on the Payee details.
15. Account provider sends Credit response to NPCI.
16. NPCI sends Pay response to Payer PSP.
17. NPCI sends pay response to Payee PSP.
18. Payee PSP notifies payee.

The following diagram illustrates the above flow.



5.7.2.2 Failure Scenarios

This section explains how the various failure scenarios are handled during the Collect transaction. The transaction flow mentioned above will be considered while describing the failure scenarios.

Failure at step 18 - PSP unable to notify the Payer:

In this scenario, when the PSP is not able to notify the end customer on the status of the transaction, a mechanism has to be put in place by the PSP to notify the customer at a later stage. This can be achieved by PSP reinitiating the notification message to customer or by providing the customer an option to check the status of the transaction through his application, or by providing a list of all transactions (with status) in the application.

Failure at step 16/17 - Response from NPCI does not reach Payee/Payer PSP:

In this scenario, when the response sent by NPCI does not reach Payer/Payee PSP, the PSPs should have a mechanism to initiate a Check Status API to know the status of the transaction. The PSP can only initiate the Check Status API to NPCI after a time period of Transaction expiry time (see expireAfter Attribute) + 90 seconds.

Failure at step 15 - Response from Payee bank does not reach NPCI:

In this scenario, when the response sent by Payee bank does not reach NPCI, this transaction will be considered Deemed acceptance and Deemed acceptance Response will

be sent to Payee and Payer PSPs. NPCI initiates maximum Three Advice messages to Payee bank to know the status of the transaction. Once the actual status is known by NPCI, message with actual response will be sent to Payee and Payer PSPs. PSPs should be able to handle multiple responses for the same transaction in this case.

Failure at step 15 - Declined Response from Payee bank to NPCI:

In this scenario, when the Payee bank responds with a declined response to NPCI, NPCI will send the reversal request to Payer bank and respond to Payee and Payer PSPs with declined response.

Failure at step 13 - Payee bank is not available to NPCI:

In this scenario, when the Payee bank is not available to NPCI, NPCI will send the reversal request to Payer bank and respond to Payee and Payer PSPs with declined response.

Failure at step 12 - Declined Response from Payer bank to NPCI:

In this scenario, when the Payer bank responds with a declined response to NPCI, NPCI will respond to Payee and Payer PSPs with declined response. No credit request will be initiated to Payee bank.

Failure at step 12 - Response from Payer bank does not reach NPCI:

In this scenario, when the response sent by Payer bank does not reach NPCI, NPCI will timeout the transaction and send reversal message to Payer bank. NPCI will respond to Payee and Payer PSPs with timeout response.

Failure at step 9 - Payer bank is not available to NPCI:

In this scenario, when the Payer bank is not available to NPCI, NPCI will respond to Payee and Payer PSPs with declined response.

Failure at step 8 - Declined Response from Payee PSP to NPCI:

In this scenario, when the Payee PSP responds with a declined response to NPCI, NPCI will respond to Payer PSP with declined response.

Failure at step 8 - Response from Payer PSP does not reach NPCI:

In this scenario, when the response sent by Payer PSP does not reach NPCI, NPCI will wait for the response till the timeout period. Payer PSP may have a mechanism to resend the response within the timeout period. If NPCI does not receive response within the timeout period, NPCI will timeout the transaction and respond to Payee PSP with a timeout response.

Failure at step 5 - Payer PSP is not available to NPCI:

In this scenario, when the Payer PSP is not available to NPCI, NPCI will respond to Payee PSP with declined response.

Failure at step 4 - NPCI is not available to Payee PSP:

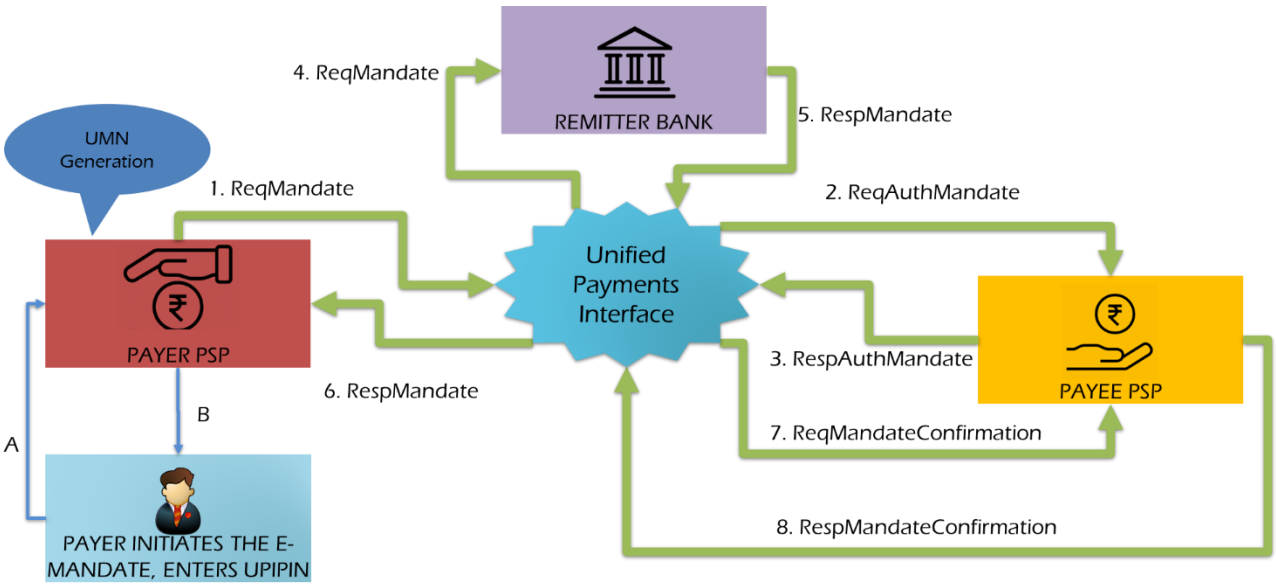
In this scenario, when NPCI is not available to Payee PSP, Payee PSP may have a mechanism to re-initiate the Collect request to NPCI.

5.8 UPI-Mandate

The objective of the UPI-Mandate is to replace the paper flow in the Mandate Flow, allowing the customer/corporate to issue/revoke in a real time manner, while the collection process remains the same as the existing collect process in UPI.

5.8.1 Scenarios

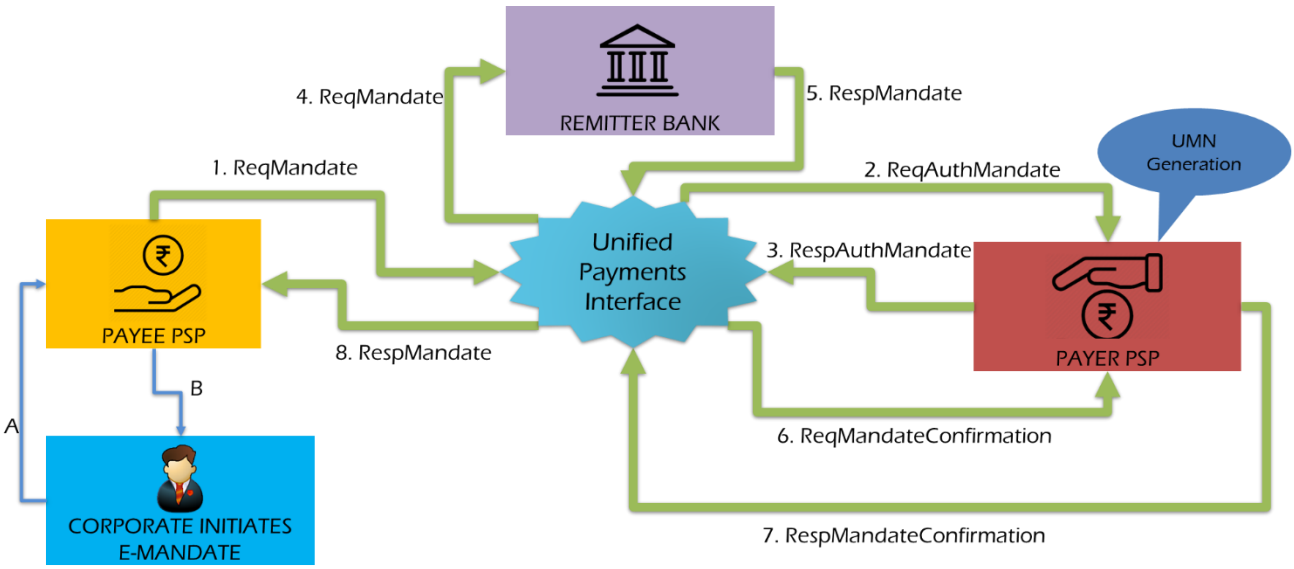
5.8.1.1 Scenario 1- Payer Initiated Mandate



A & B are the communication between the customer and Payer PSP.

1. Payer (customer) creates the mandate on PSP app against the payee VPA by filling in the mandate attributes (amount, one time or multi time, frequency, etc.)
    - ❖ Payer provides credentials (PIN/Biometrics)
    - ❖ PSP creates UUID based UMN (Unique Mandate Number). Payer PSP sends ReqMandate to UPI switch
    - ❖ Payer has the option (shareToPayee) to share UPI-Mandate details with payee which is applicable only for Single Occurance UPI-Mandate only. Payee PSP will get the ReqAuthMandate and Payee PSP will respond it. But Payee PSP should not send notificationto the customer if shareToPayee="N"
2. UPI switch sends ReqAuthMandate to Payee PSP/Corporate PSP for address resolution.
3. Payee PSP resolves the address and responds to UPI with a RespAuthMandate. (If Aadhaar based auth, UPI sends authentication request to UIDAI)
4. UPI sends ReqMandate to Remitter Bank for signed mandate.
5. Remitter validates request, PIN (if PIN based auth), etc. and if valid "digitally signs" the mandate XML and returns the entire digitally signed mandate block within the RespMandate to UPI
    - ❖ Digital signature ensures that mandate is non-tamperable and authenticity is verifiable.
    - ❖ Remitter may or may not store it (not necessary to store since entire mandate which is digitally signed by the remitter bank can be validated when debit request comes back).
    - ❖ Remitter should sign the mandate block which is similar to the standard XML digital signature method. (NPCI is advise to use a separate certificate for mandate block digital signing)
    - ❖ Remitter should sign the mandate block prior to signing the RespMandate XML.
6. UPI Switch sends RespMandate to Payer PSP
7. Also UPI Switch sends the ReqMandateConfirmation message to Payee PSP without the digitally signed mandate block.
8. Payee PSP sends the RespMandateConfirmation to UPI.Payer PSP stores it under the user as a VPA (umn@psp). Payer PSP app UI should show mandates under separate section/tab and not mix with regular VPAs, to avoid confusion.
    - ❖ Payer can see the valid mandates, revoke them, etc. at PSP level

5.8.1.2 Scenario 2 - Payee Initiated Mandate

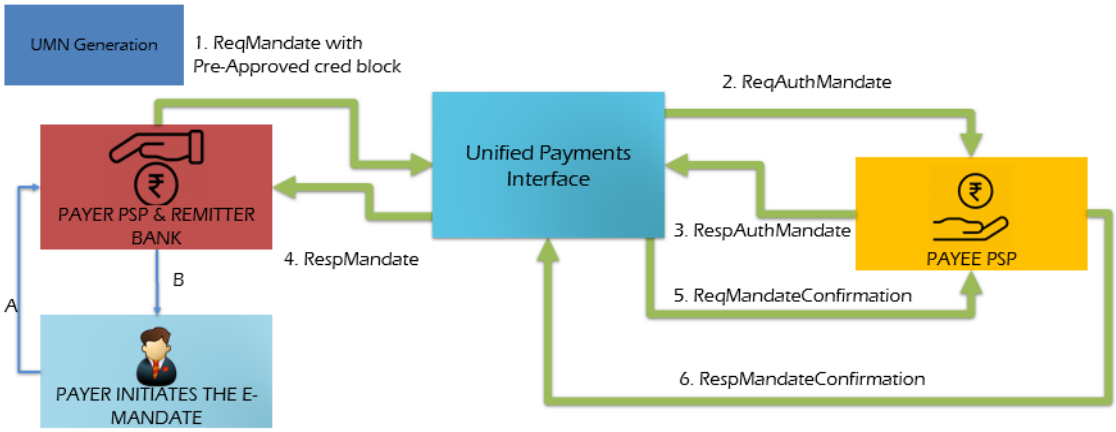


A & B are the communication between the Corporate and Payee PSP.

1. Payer (customer) provides his VPA to the payee (merchant, entity, individuals)
   - ❖ Payee collects input amount etc. as required from payer and forms the mandate request and sends via payee PSP.Payee PSP sends the ReqMandate request to UPI
2. UPI switch sends the ReqAuthMandate to the payer PSP based on the handle in the VPA
3. Payer views the request on his mobile and authorizes the mandate by providing PIN/biometrics. Payer PSP sends RespAuthMandate to UPI.(If Aadhaar auth, UPI sends authentication request to UIDAI)
4. UPI forwards ReqMandate to Remitter bank for validating credentials.
5. Remitter bank validates request, PIN (if PIN based auth), etc. and signed "digitally signs" the mandate block prior to signing the RespMandate XML. Remitter bank returns the entire digitally signed mandate block within the RespMandate to UPI.
6. UPI responds to Payee PSP with RespMandate without the digital signed mandate block.
7. Also UPI sends the ReqMandateConfirmation with the digital signed block to the Payer PSP.Payer PSP stores it under the user as a VPA (umn@psp). PSP app UI should show mandates under separate section/tab and not mix with regular VPAs to avoid confusion.
   - ❖ Payer can see the valid mandates, revoke them, etc. at PSP level
   - ❖ Revoke option is not applicable for loan mandate payments.Current version of UPI-Mandate does not support loan repayment options.

8. Payer PSP sends RespMandateConfirmation to UPI.

5.8.1.3 Scenario 3 - Payer Initiated PreApproved Mandate



A & B are the communication between the customer and Payer PSP.

1. Payer (customer) creates the mandate on PSP app against the payee VPA by filling in the mandate attributes (amount, one time or multi time, frequency, etc.)
   - ❖ PSP creates UUID based UMN (Unique Mandate Number). Payer PSP sends ReqMandate with Pre-Approved cred block to UPI switch if Payer PSP and Remitter Bank are same.
2. UPI switch sends ReqAuthMandate to Payee PSP/Corporate PSP for address resolution.
3. Payee PSP resolves the address and responds to UPI with a RespAuthMandate.
4. UPI Switch sends RespMandate to Payer PSP.
5. Also UPI Switch sends the ReqMandateConfirmation message to Payee PSP.
6. Payee PSP sends the RespMandateConfirmation to UPI. Payer PSP stores it under the user as a VPA (umn@psp). Payer PSP app UI should show mandates under separate section/tab and not mix with regular VPAs, to avoid confusion.
   - ❖ Payer can see the valid mandates, revoke them, etc. at PSP level

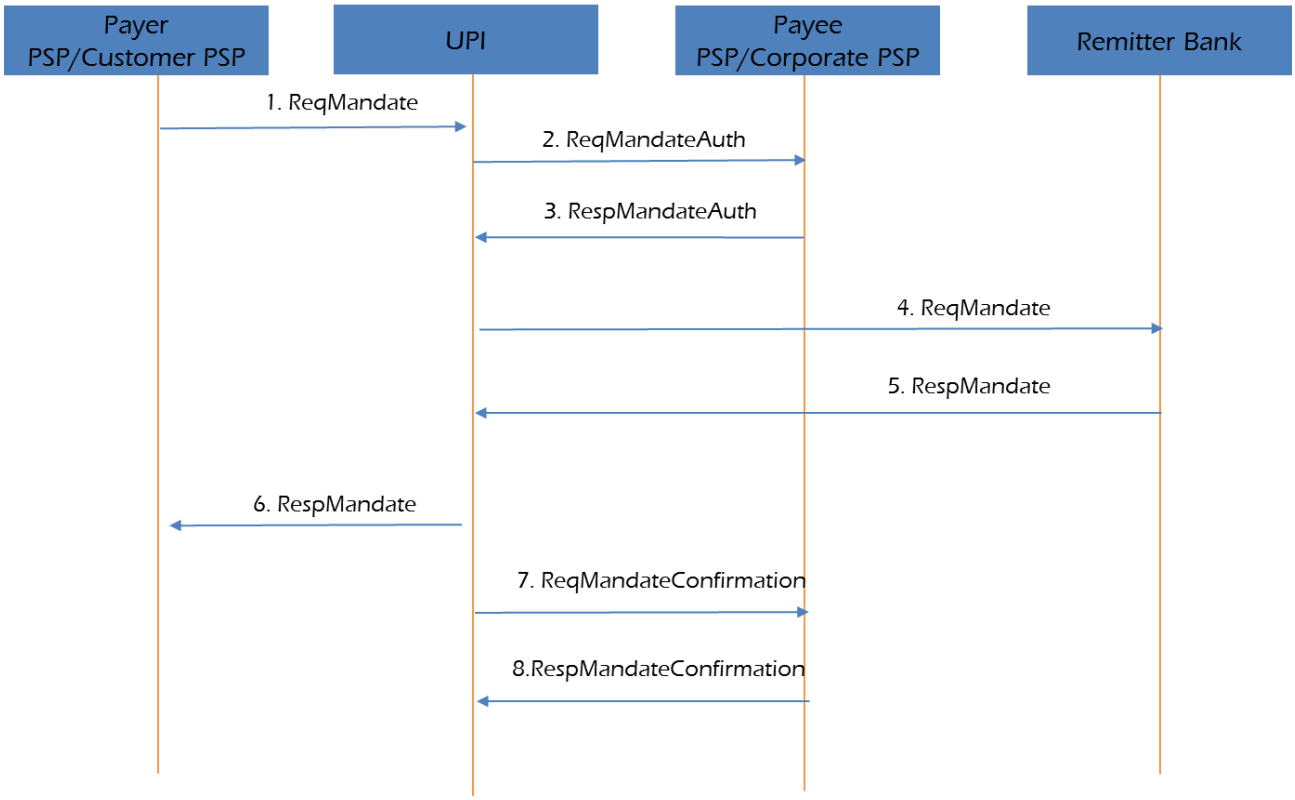
Note:

1. Revoked mandate cannot be reused

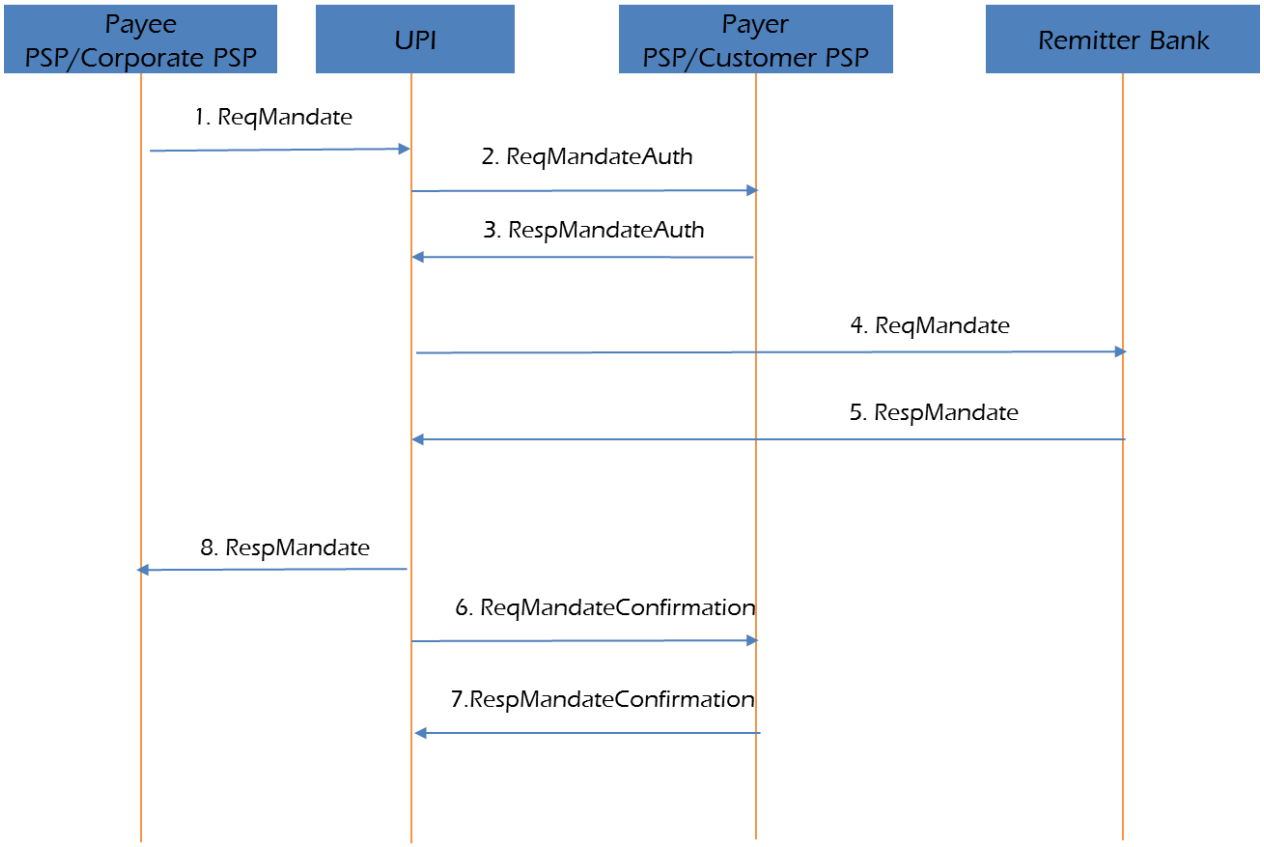
2. In case of payer psp fails to send ACK or RespMandate for the final leg and Remitter fails to send ACK or RespMandate, UPI will forward the failure message in the ReqMandateConfirmation leg to Remitter bank. Remitter bank should unblock the customer account and send the RespMandateConfirmation back to UPI

5.8.2 UPI-Mandate Sequential Flow

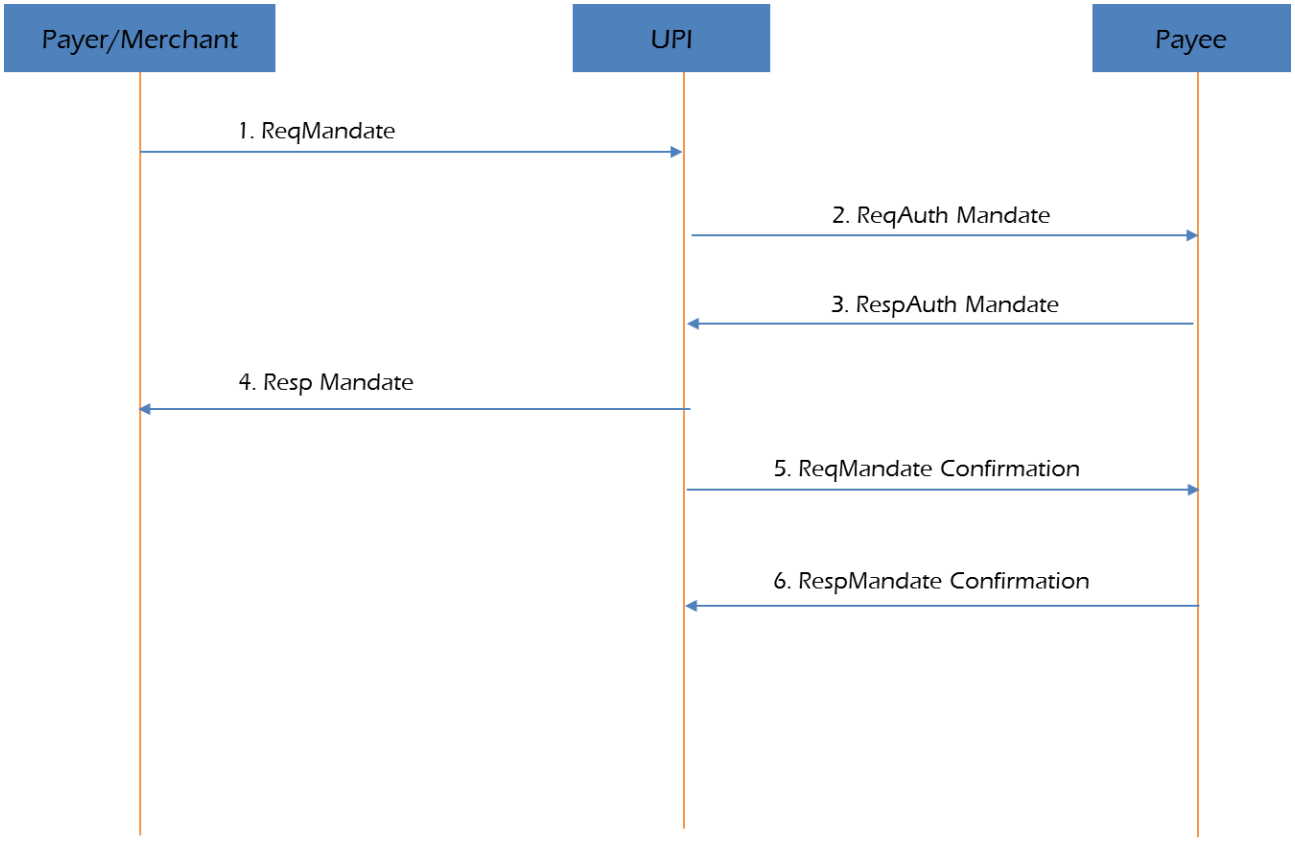
5.8.2.1 Flow 1- Payer Initiated Mandate



5.8.2.2 Flow 2- Payee Initiated Mandate

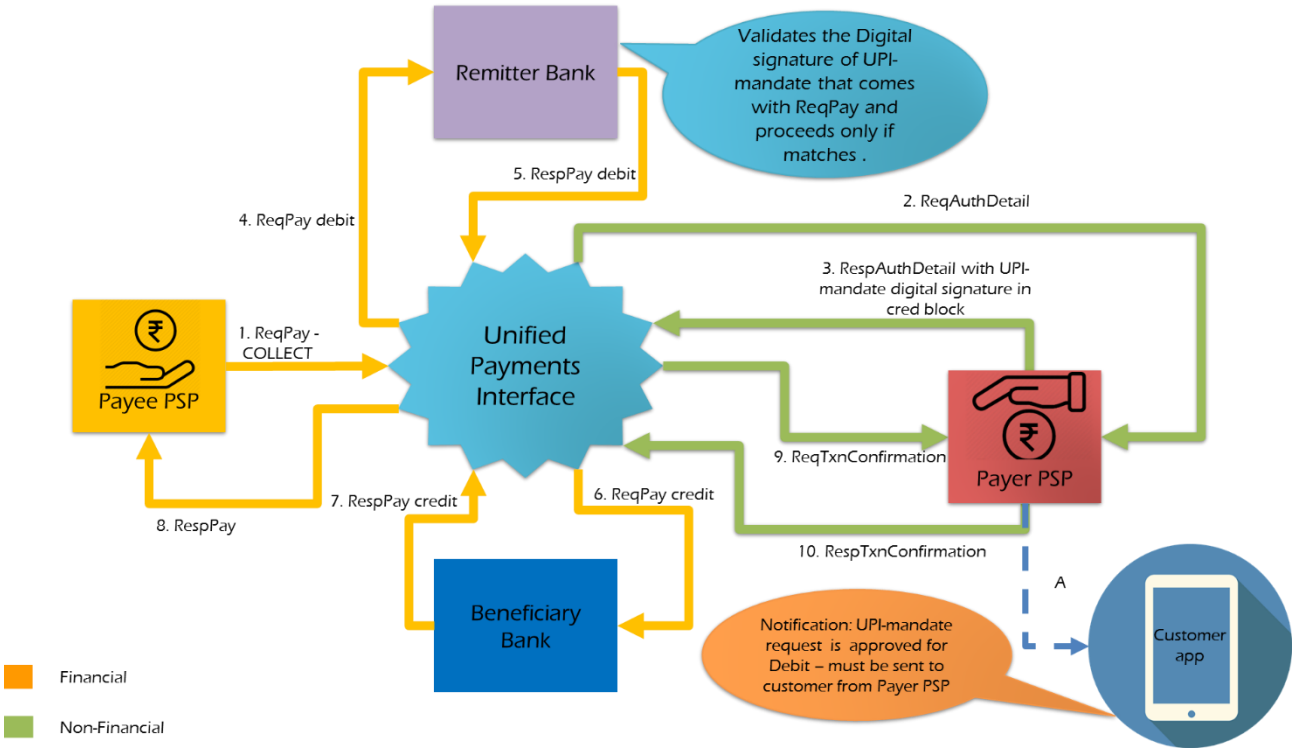


5.8.2.3 Flow 3- Payer Initiated PreApproved Mandate



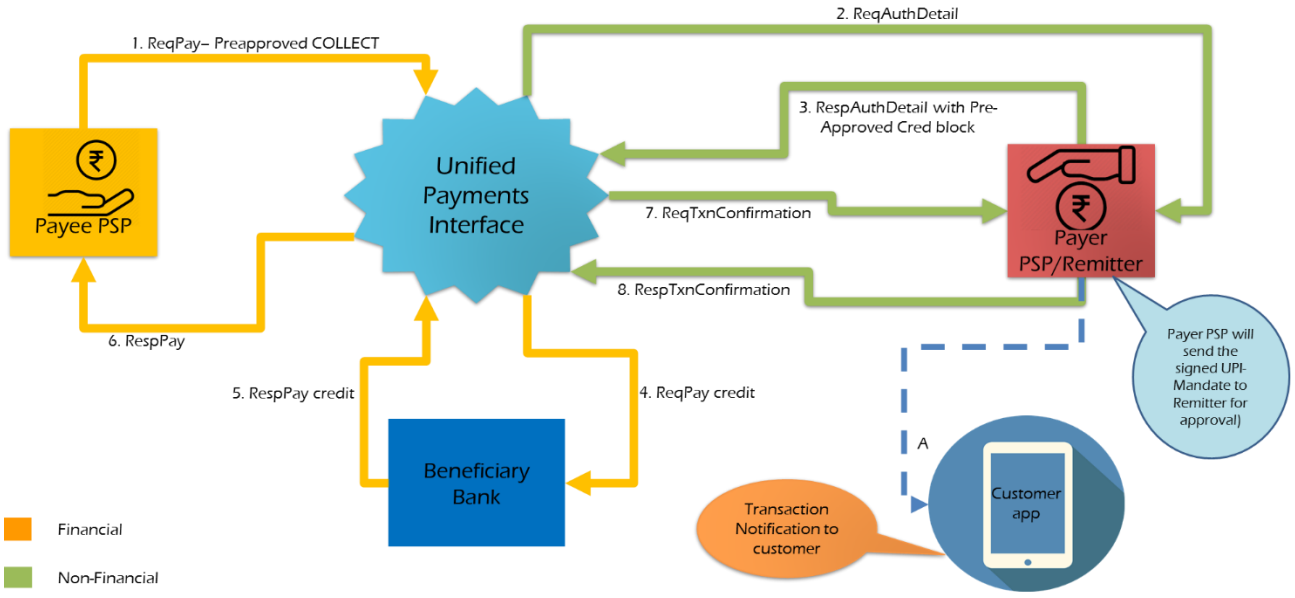
5.8.3 Financial Flow

5.8.3.1 Financial Flow UPI-Mandate



Note: For API detail, kindly refer sec 6.4

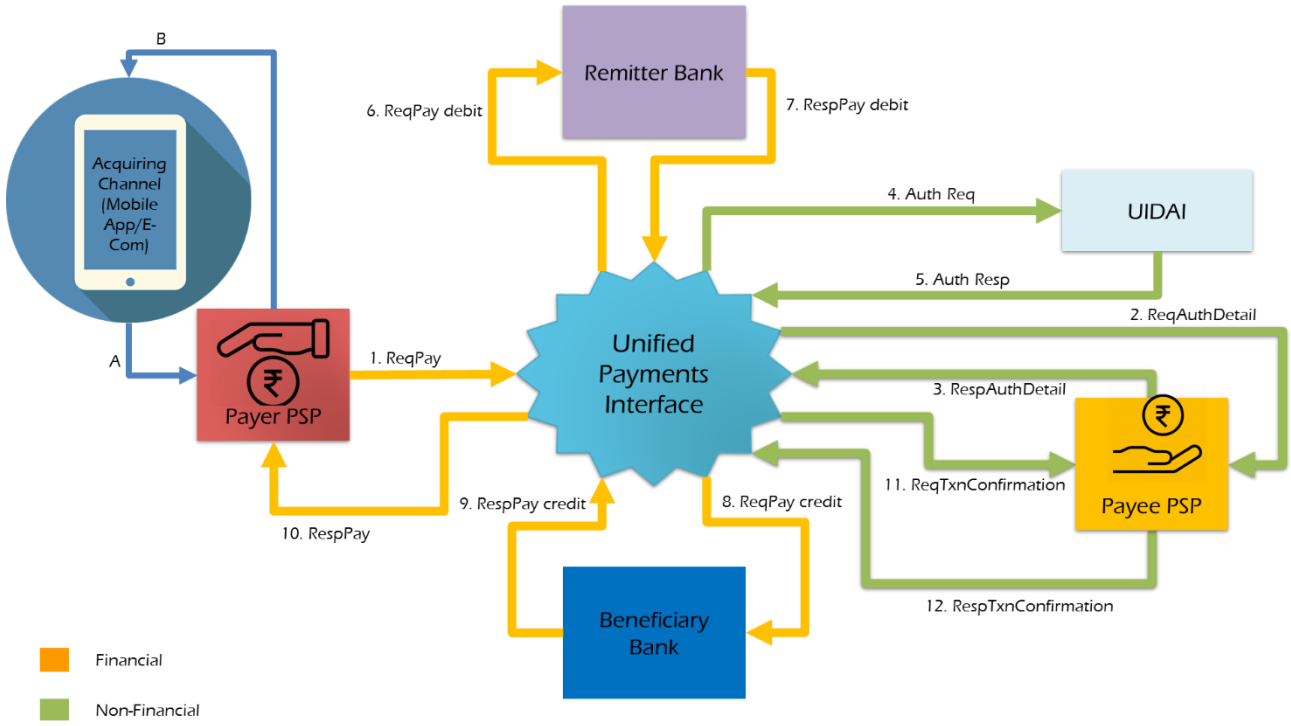
5.8.3.2 Financial FlowUPI-MandatePre-Approved



Note: For API detail, kindly refer sec 6.4

MAJOR PSP LEVEL CHANGES – UPI-MANDATE
1. Payer/ Payee PSP have to include the mandate API's (ReqMandate, RespMandate, ReqAuthMandate, RespAuthMandate, ReqMandateConfirmation and RespMandateConfirmation).
2. Payer/Payee can initiate UPI-Mandate request
3. Payee/corporate PSP can only initiate the financial UPI-Mandate (collect) request.
4. Payer PSP can only generate UMN
5. Payer PSP should add the digital signed xml as a part of cred block which was provided by remitter bank at the time of mandate creation. Remitter bank should verify this digital signed block before initiating the debit.

5.9 Aadhaar Biometric



A & B are communications between mobile and Payer PSP

1. Customer initiates ReqPay to UPI via payer PSP. Customer enters his Biometric as authentication parameter.
2. UPI sends ReqAuthDetail to Payee PSP for address resolution.
3. Payee PSP send RespAuthDetail to UPI with the resolved address.
4. UPI sends AuthReq to UIDAI to authenticate the customer's biometric
5. UIDAI send AuthResp successfully to UPI.

6. After that, UPI sends ReqPay debit to remitter bank along with the authentication code provided by UIDAI.
7. Remitter bank will debit the customer account based on the UIDAI authentication and sends the success RespPay debit.
8. UPI sends ReqPay credit to beneficiary bank to credit payee's account
9. Beneficiary bank send RespPay credit to UPI.
10. UPI sends final RespPay message to payer PSP
11. UPI sends ReqTxnConfirmation message to Payee PSP.
12. Payee PSP send RespTxnConfirmation to UPI.

5.9.1 Credential Flow - Biometric

- PSP APP will initiate account listing call to UPI and issuer responds with Aadhaar details
- The Account listing response has Aadhaar enabled parameter called AEBA, This flag will be passed to Common Library (CL), in turn CL initiates a call to UIDAI registered device service to discover the biometric devices attached to the mobile device.
- Capturing option which will be shown to the customer, with two options as IRIS or FINGERPRINT.
- If the customer selects Biometric and if the mobile is capable of capturing both IRIS/Fingerprint, then CL provides an UI to customer to select which option.
- Common Library will invoke UIDAI RD service of the device using String capture (String pidOptions) interface of Aadhaar sdk to capture the IRIS/Fingerprint.

5.9.2 Aadhaar Authentication Request

- PSP App will send the received response from CL to PSP server
- PSP server will form ReqPay request.
- PSP will form the credblock like

```
<Cred type="AADHAAR" subType="AADHAAR-BIO-FP | AADHAAR-BIO-IRIS | AADHAAR-OTP">
                    <Meta lk="" ac="" sa="" uid="" ver=""/>
<!– The above"ver" attribute value will be the Aadhaar API version –>
<Data code="" ki=""> base-64 encoded authentication data</Data>
<!– # Will have  Aadhar BIO response of CL –>
</Cred>
```

- PSP will add lk and ac elements to the respond XML tag and send the request UPI.
- UPI will send ReqAuthXml to UIDAI for Authentication
- UIDAI validates the captured IRIS/Finger print and responds back with a 40 digit Authentication response code & validation result (y/n) to UPI

- UIDAI sends RespAuthXMl with the authenticated response to UPI.
- If the Aadhaar authentication is success, then UPI will send the debit and credit request to process the transaction.
- If Aadhaar auth fails, UPI sends the final RespPay with declined response with specifc error code.

Note: Please refer UIDAI document for more information about the fields and elements mentioned inside the credblock.

5.9.3 Aadhaar Integration

- Biometric authentication using UIDAI central repository to perform financial and non-financial transactions
- New Credential sub types will be added BIO-FP | BIO-IRIS
- Account listing response has Aadhaar enabled parameter(AEBA) which will be deciding parameter to show the UPI PIN /Biometric authentication page to user
- Aadhaar Registered Device (RD) Service will be used to connect and capture biometric data securely

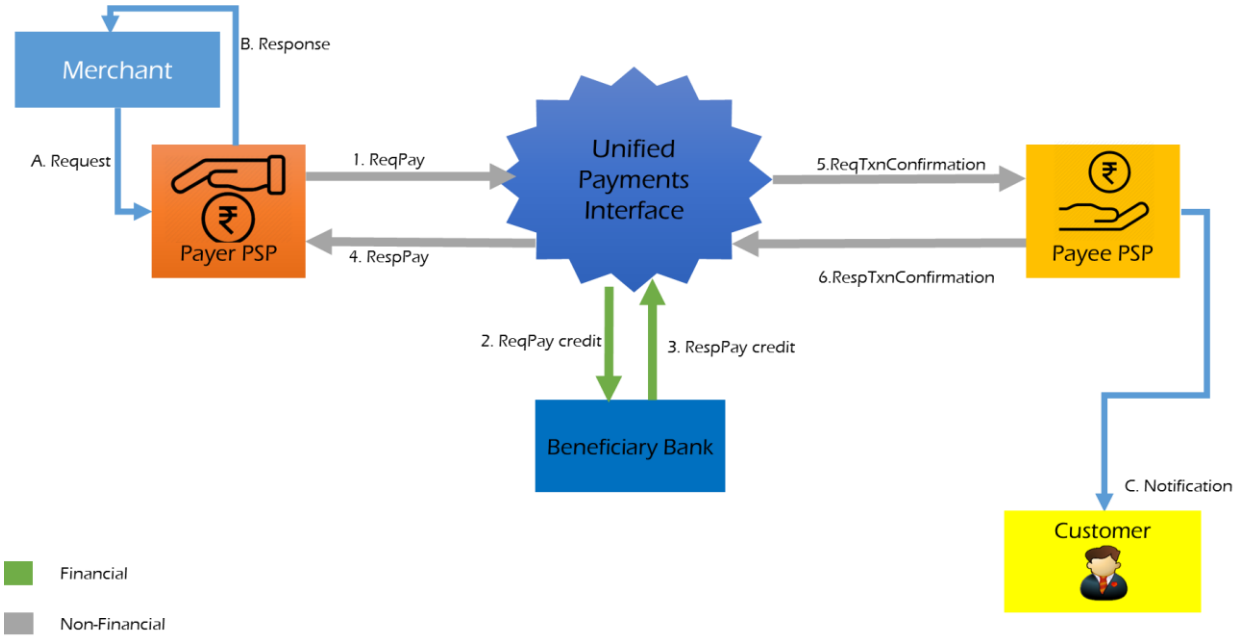
Note: For more references, kindly refer UIDAI document: https://uidai.gov.in/images/resource/aadhaar_registered_devices_2_0_09112016.pdf

MAJOR PSP LEVEL CHANGES – AADHAAR
1. PSP has to integrate with the new CL and call CL with respect to the response list account
2. If aeba flag=Y, PSP have to call the CL with the new options of IRIS/Finger Print
3. If it is aeba=N, PSP needs to follow the existing flow.
4. All the Aadhaar flow transaction are authorized by UIDAI.
5. Remitter bank should have the capability to debit the customer's account by considering the UIDAI response

5.10 Online Refund

At present the refund is done by the merchant in off-line mode only. The below sections explains how the online refund can be done in UPI

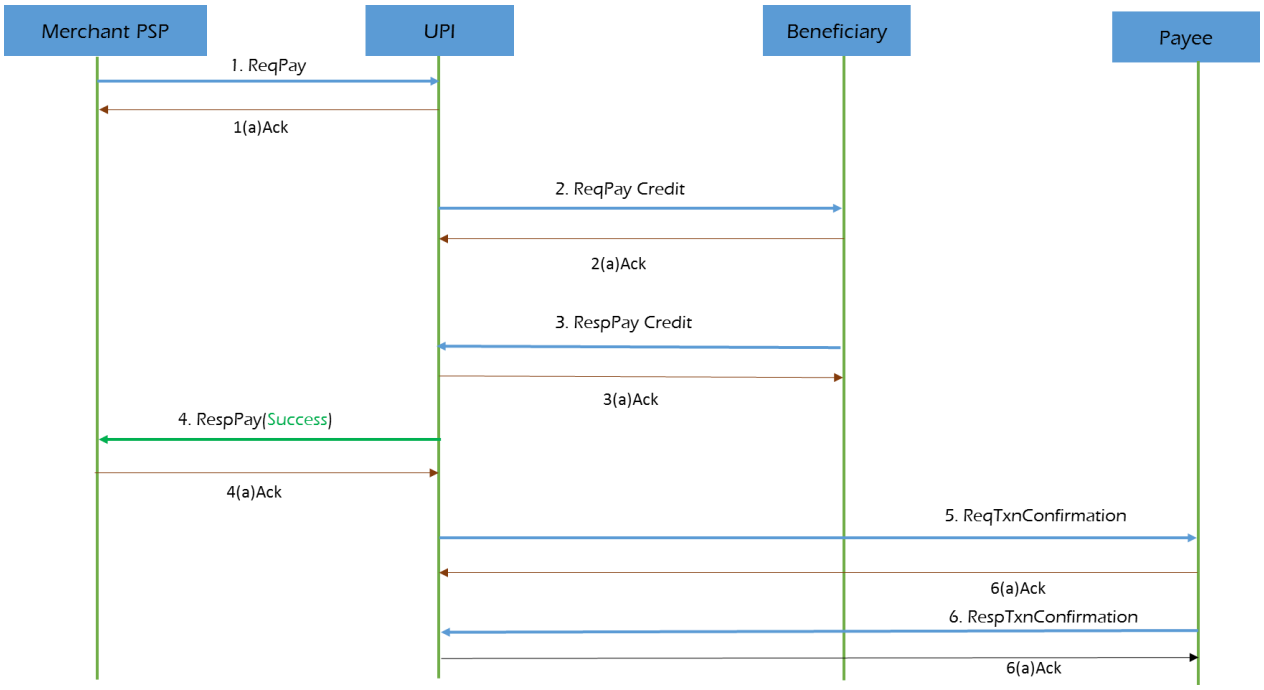
5.10.1 Merchant Initiated Refund



A & B are the communication between the Merchant and Payer PSP.

1. Payer PSP initiates Pre-Approved Refund (Pay Transaction) to UPI with type as 'REFUND'.
    a. The Merchant can initiate the REFUND transaction with VPA & Account No+ IFSC as well as Global Address along with OrgTxnId, OrgRrn, and OrgTxnDate & Requested amount.
    b. All the necessary validations will be done by the initiator PSP.
2. UPI will send the credit request to customer bank.
3. Customer bank will send the RespPayCredit to UPI.
4. UPI will send the Transaction Confirmation to the Payee PSP (Customer PSP) only when Payer PSP initiated with VPA.
5. In turn Payee PSP should confirm the customer with a notification for the credit back.

5.10.2 Sequential Flow



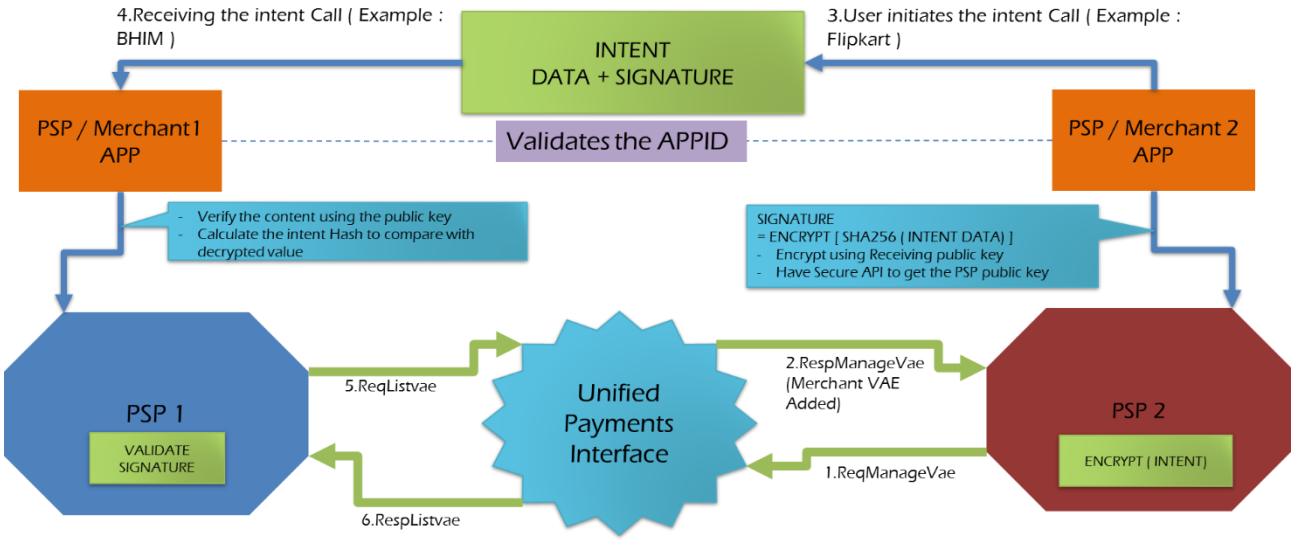
MAJOR PSP LEVEL CHANGES – ONLINE REFUND
1. PSP will initiate REFUND request with type=REFUND and OrgTxnId.
2. PSP should verify the orginal transaction details before raising refund
3. Refunds will happen only to the account no + IFSC used for transaction irrespective of VPA and current underlying account.

5.11 Signed Intent / QR

Objective of intent / QR Code based payments is to incorporate simplicity, security and seamlessness in UPI transactions. These methods makethe payment integration easier for merchants providing scope for new use cases. Signing of intent/QR provide an additional layer of security, simplify transaction completion and bring sanity across ecosystem for intent based payments.

Intent / QR Code payment method allows the user to complete the transaction, invoking the PSP application by means of Android/iOS intent, NFC, BLE & UHF. The invoked application prompts the user to enter UPI PIN to complete the transaction.

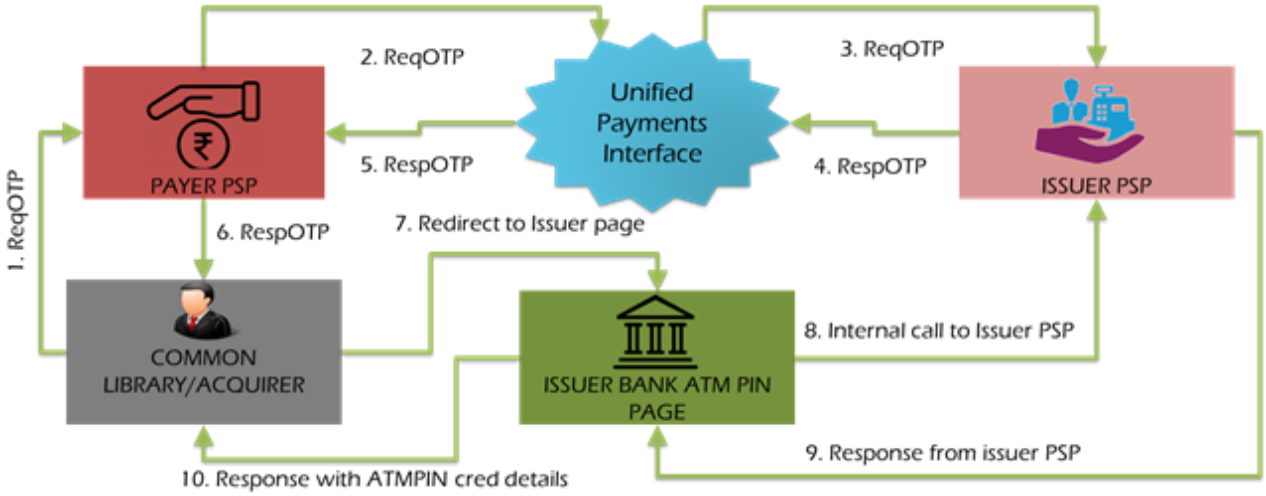
5.11.1 Functional Architecture



1. UPI plays central role to provide certificate registry for all PSP and Merchants.
2. PSP will add the public key for all their merchants using Mange VAE API which will have new block to capture public key of the merchant.
3. PSP will use the List VAE to retrive the public key of the merchants
4. PSP already has all the other PSP keys which we can be retrived using the List Keys API.
5. The Merchant/PSP will use the private key to sign the hashed conent of the intent (SHA256 with RSA) and that will send to the other PSP via intent communication.
6. The receiving PSP will use the public key to verify the content integrity.
7. This will help to secure the payment data flows between PSP's APP's using intent /QRcommunication.

5.12 ATMPIN Validation in Issuer Page

5.12.1 Functional Architecture



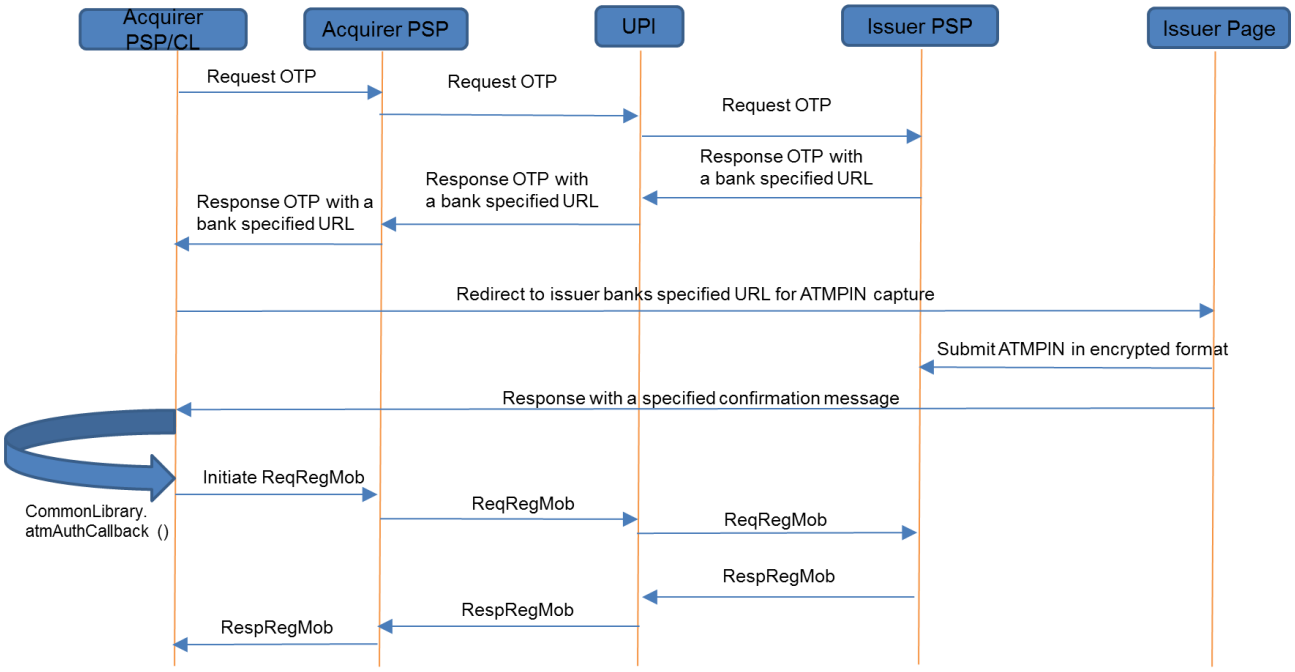
1. In the mobile registeration process, last 6 digit of the card number and expiry date are captured in acquirer PSP app.
    a. After the submittion of card details, the pre-requistic process of OTP validation happen.
2. Acquirer PSP initiates ReqOTP along with card details to UPI.
3. UPI forwards the same request to issuer bank.
4. Issuer bankstores the txn id of ReqOTP, account details, mobile number,card details and will generate a token with all these details for verification later when ATM PIN is received thorugh the bank page.
    a. RespOTP will be returned with a **bank specified URL** (here after called BankURL) to the initiator PSP via NPCI (new attribute called "securePinUrl" is added in the RespOTP to capture the URL from Issuer).
    b. Issuer bank sends the OTP message to customer registered mobile number
5. UPI sends the RespOTP to acquirer PSP.
6. Acquirer App will call the Common Library(CL) with the bank specified URL in the specified format. CL will auto-populate the OTP.
7. CL will then call the Bank URL to re-direct to issuer page for ATM PIN capture. Bank URL will contain all the required details including the token to identifiy the transaction. The Bank URL should have self sufficient information to understand it triggered from a secured source which means banks can specify more parameters in the url to validate the uniqueness of the page data
8. Issuer bank page will get the customer's ATM PIN and validates with the Issuer PSP. PSP will use the details stored during the ReqOTP for ATM PIN verification.Token received from bank page will help to find the corresponding ReqOTP message.

9. Issuer PSPwill generate a "Response ID" and store against the Token generated in RespOTP. "Response ID" will be used to link the ReqRegMob with this transactions. Issuer PSPsends the response to issuer bank page with the "Response ID" and status of ATM PIN verification.
10. Issuer bank page respond with the signed authenticated "Response ID" to CL.
    a. With all the credentials, PSP triggers ReqRegMob with OTP, MPIN and ATMPIN cred block (response from issuer bank) which are encrypted with NPCI key to UPI.
    b. Issuer will authenticate the customer using the OTP and values present in the ATMPIN cred block in the ReqRegMob.
    c. CL will through an error if the status from the issuer bank page is failure.

NOTE:

1. SSL communication will be used between App/CL and bank user interface.
2. Issuer Bank will sent the Bank ATM validation URL as part of response OTP
3. ATM PIN will be entered only in issuer page, Issuer will validate the ATM PIN for the given user Card details.
4. "Response Id" generated by issuer will be used for authentication in the ReqRegMob

5.12.2 Sequential Flow



5.12.3 ATM PIN Callback

- NPCI will expose JS interface (via WebView) called "CommonLibrary.atmAuthCallback"
- Bank to call this method once ATM PIN entered by the user has been validated
- Argument to this method should contain a JSON string as follows

  {"responseId": "bankUniqueResponseId",

  "status:00|XY|ZM|XL",

  "version": "1.0|2.0"}

  - responseId - will be a unique Id present only there if the validation is success
- Version code will be "1.0|2.0". This is used for extensibility.
- Response ID parameter will identify the result of the authentication attempt. The result of the auth itself is not explicitly provided to CL.
- Response ID & version will be sent from CL => Acq => NPCI => Issuer
- Issuer must maintain the correlation between Txn ID & Response ID. This must be validated while receiving ReqRegMob.

MAJOR PSP LEVEL CHANGES – ATM PIN
1. In mobile registration process, after the OTP response, CL re-direct via PSP to issuer page using URL which has self explained the transaction detail
2. Issuer bank validate and send the authenticated id as a ATM PIN cred block
3. With all the credentials, CL triggers ReqRegMob with OTP, UPIPIN and ATMPIN block response from issuer bank as a cred block which are encrypted with NPCI key to UPI .

6. Detail API Specifications

6.1 API Protocol

All APIs are exposed as stateless service over HTTPS. PSP should ensure idempodent behaviour for all APIs. Usage of open data format in XML and widely used protocol such as HTTP allows easy adoption by the members.

API input data should be sent to the following URL as XML document using Content-Type "application/xml" or "text/xml".

```
https://<host>/upi/<api>/<ver>/urn:txnid:<txnId>
```

host– API server address (Actual production server address will be provided to members at the time of rollout and all API clients should ensure that actual URL is configurable).

upi– static value denoting the root of all API URL paths under the Unified Payments Interface.

api– name of the API URL endpoint.

ver– version of the API. Multiple versions of the same API may be available for supporting gradual migration. As of this specification, default version is "1.0|2.0".

txnId– Transaction id which will be used for load balancing purpose at UPI end

All APIs have same ack response as given below:

```
<upi:Ack xmlns:upi="" api="" reqMsgId="" errCode="" ts=""/>
```

Ack – root element name of the acknowledgement message.

api– name of the API for which acknowledgement is given out.

reqMsgId - message ID of the input for which the acknowledgement is given out.

err - this denotes any error in receiving the original request message.

ts - timestamp of the acknowledgement sent by the receiver.

The below are the list of Financial APIs defined in the UPI system.

S.No	Financial API Names	API Description
1	ReqPay	API is used for both Direct Pay and Collect Pay transaction initiation by the PSPs and processing the transaction through one of the following channels IMPS,AEPS etc.
2	RespPay	API is used for sending back the response of transaction (Direct and Collect Pay) initiated through ReqPay Api to the PSPs
3	ReqAuthDetails	API is used to authorize a payment and translate PSP specific payment addresses to any of the common global addresses (Aadhaar number, Mobile number, Account + Provider ID) that NPCI can understand.This API is called to translate PSP address and obtain appropriate authorization details

S.No	Financial API Names	API Description
1	ReqPay	API is used for both Direct Pay and Collect Pay transaction initiation by the PSPs and processing the transaction through one of the following channels IMPS,AEPS etc.
4	RespAuthDetails	API is the response call back interface to return back details. After processing theReqAuthDetails API, PSP should send response to NPCI the authorization by calling the "RespAuthDetails" API .

The below are the list of Non-Financial APIs (META-API's) defined in the UPI system.

S.No	Non-FinancialAPI Names	API Description
1	List PSP	This API allows the PSPs to request the list of all registered PSPs for local caching. This data should be used for validating payment address before initiating the transaction.
2	List Account Providers	This API allows PSP to get list of all account providers who are connected via unified interface. PSPs should maintain the list and check for registered account providers before registering a customer account within their application.
3	List Keys	This API allows the PSPs to request and cache the list of public keys of account providers and other entities in the UPI eco system. Trusted and certified libraries will be used by PSPs for credential capture and PKI public key encryption at capture time.
4	List Account	API allows PSPs to find the list of accounts linked to the mobile by an account provider.
5	List Verified Address Entries	API allows PSPs to request and cache the List of Verified Address Entries to protect customers from attempts to spoof well known merchants such as LIC, Indian Railways, ecommerce players, telecom players, bill payment entities, etc.
6	Manage Verified Address Entries	API is a mechanism, where the PSPs can manage, and access the common collection of verified address entries. NPCI, with the help of PSPs, will define a process to manage these entries.
7	Validate Address	This API will be used by the PSPs when their customer wants to add a beneficiary within PSP application (for sending & collecting money).

S.No	Non-FinancialAPI Names	API Description
8	Set Credentials	This API is required for providing a unified channel for setting and changing UPIPIN across various account providers
9	Reg Mob	This API allows customer to register for mobile banking
10	Check Txn Status	This API allows the PSPs to request the transaction status. The PSPs must request for status only after the specified timeout period.
11	OTP-Request	This API allows the PSPs to request an OTP for a particular customer from an issuer.
12	Balance-Enquiry	This API Allows PSP to enquirebalance of a user.
13	HeartBeat Messages	This API is a mechanism for UPI system monitoring. (monitoring connection with PSPs and sending EOD to PSPs)
14	Request Pending Messages	This API allows PSP to request pending messages against a given mobile number or Aadhaar number.
15	Request Txn Confirmation	This API provides transaction status confirmation from UPI to PSP. At the end of every transaction, this API will be initiated to second PSP for status confirmation.
16	ReqMandate	This API allows the corporate/customer to create a mandate request via UPI.
17	RespMandate	API will be used for sending back the response of mandate to the initiated PSPs.
18	ReqAuthMandate	API is used to authorize a payment and translate PSP specific payment addresses to any of the common global addresses (Aadhaar number, Mobile number, Account + IFSC) that NPCI can understand.This API is called to translate PSP address and obtain appropriate authorization details
19	RespAuthMandate	"RespAuthMandate" API is the response call back interface to return back details. After processing the ReqAuthMandate API, PSP should send response to NPCI with authorization by calling the "RespAuthMandate" API.
20	ReqMandateConfirmation	This API provides the response for the confirmation message received from UPI.
21	RespMandateConfirmation	This API provides the confirmation message from the PSP to UPI.

6.2 Financial APIs

6.2.1 ReqPay

Complete (not all elements/attributes are required for all transactions) XML input message structure for ReqPay API is given below.

```
<upi:ReqPay xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Meta>
<Tag name="PAYREQSTART" value=""/>
<Tag name="PAYREQEND" value=""/>
</Meta>
<Txn id="" note="" custRef="" refId="" refUrl=""ts="" refCategory="00|01|02|03|04|05|06|07|08|09"
        type="PAY|COLLECT|DEBIT|CREDIT|REVERSAL|REFUND" orgTxnId="" orgRrn="" orgTxnDate=""
purpose="00|01|02|03|04|05|06|07|08|09|10"<! – The psp should populate the purpose field which
is used to fill the purpose of the txn– >
        initiationMode="00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15" subType=""
orgRespCode=""><!—the subType field is only applicable for ReqPay_debit/credit/reversal – >
        <RiskScores>
        <Score provider="sp" type="TXNRISK" value=""/>
        <Score provider="npci" type="TXNRISK" value=""/>
</RiskScores>
<Rules>
<Rule name="EXPIREAFTER" value="1 miniute to max 64800 minitues"/>
        <!—If EXPIREAFTER is not provided default value will be taken as 30 minutes – >
<Rule name="MINAMOUNT" value=""/>
</Rules>
</Txn>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">

<Merchant >
        <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
        merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR"/>
        <Name brand="" legal="" franchise=""/>
        <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
<Institution type="MTO|BANK" route="MTSS|RDA">
<Name value="" acNum=""/>
<Purpose code="" note=""/>
<Originator name="" type="INDIVIDUAL|COMPANY" refNo="">
        <Address location="" city="" country="" geocode=""/>
</Originator>
<Beneficiary name=""/>
      </Institution>
<Info>
<Identity id= "" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
```

```
</Info>
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/..."/>
</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail                                                                    name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>

</Ac>
<Ac addrType ="CARD">
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="CARDNUM" value=""/>
</Ac>
<Creds>
<Cred type="AADHAAR" subType="AADHAAR-BIO-FP|AADHAAR-BIO-IRIS|AADHAAR-BIO-OTP">
                    <Meta Ik="" ac="" sa="" uid="" ver=""/>
                        <Datacode="" Ki=""> base-64 encoded/ encrypted authentication data </Data>
                    <! – If it is Aadhaar authentication issued below is the format
                        UIDAI Aadhaar Response Code (if y = 000) | UIDAI Aadhaar response authentication
                        40-digit code– >
</Cred>
<Cred type="UPI-Mandate" subType="DS">
                    <Data> base-64 encoded digitally signed UPI-Mandate</Data>
                    <! – This cred block is applicable only for the UPI-mandate txn – >
</Cred>
<Cred type="OTP" subType="SMS|EMAIL|HOTP|TOTP">
<Datacode="" ki="">
base-64 encoded/encrypted authentication data
</Data>
<Cred type="PIN" subType="MPIN">
                        <Datacode="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="CARD" subType="CVV1|CVV2|EMV">
<Datacode="" ki=""> base-64 encoded/encrypted authentication data</Data>
```

```
</Cred>

        <Cred type="PREAPPROVED" subType="NA">
                <Data> base-64 encoded</Data>
                <!– #data includes respCode and approvalRef
                In the format "respCode|approvalNum"
                – >
</Cred>
</Creds>
<Amount value="" curr="INR">
<Split name="PURCHASE|CASHBACK" value=""/> <!—It is for future use for multiple payer option – >
</Amount>
</Payer>
<Payees>
<Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">

<Merchant >
        <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
        merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR"/>
        <Name brand="" legal="" franchise=""/>
        <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>

<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value="+91.99999.99999"/>
<Tag name="GEOCODE" value="12.9667,77.5667"/>
<Tag name="LOCATION" value="Sarjapur Road, Bangalore, KA, IN" /> <! —It is mandatory for
Merchant for payee– >
<Tag name="IP" value="123.456.123.123"/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value="123456789"/>
<Tag name="OS" value="Android 4.4"/>
<Tag name="APP" value="CC 1.0"/>
<Tag name="CAPABILITY" value="011001"/>
<Tag name="TELECOM" value="Airtel/Vodafone/.."/>
</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail                                                                    name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET| SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Amount value="" curr="INR">
```

<Split name="PURCHASE|CASHBACK" value=""/>
</Amount>
</Payee>
</Payees>
</upi:ReqPay>

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
1.1	API Name	<upi>	1..1			Y	
1.1.1	API Schema namespace	xmlns	1..1	Alphanumeric	Min Length : 1 Max Length : 255	Y	
2.1	Header for the message	<Head>	1..1	Alphabetic	Fixed value	Y	
2.1.1	Version of the API	ver	1..1	Numeric	Min Length : 1 Max Length : 6	Y	019_Head_Version
2.1.2	Time of request from the creator of the message	ts	1..1	ISODateTime	Min Length : 1 Max Length : 255	Y	020_Head_ts
2.1.3	Organization id that created the message	orgId	1..1	Numeric	Min Length : 1 Max Length : 20	Y	
2.1.4	Message identifier- used to correlate between request and response	msgId	1..1	Alphanumeric	Length =35	Y	021_Head_MsgId
3.1	Meta data primarily for analytics purposes	<Meta>	0..1	Alphabetic	Fixed value	N	
3.2	Meta data primarily for analytics purposes	<Meta.Tag>	0..1	Alphabetic	Fixed value	N	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
3.2.1	Name of the property	name	1..n	Code	Min Length : 1 Max Length : 20	Y	
3.2.2	Value of the property	value	1..n	ISODateTime	Min Length : 1 Max Length : 255	Y	
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1	Alphabetic	Fixed Value	Y	
4.1.1	Unique Identifier of the transaction across all entities, created by the originator	id	1..1	Alphanumeric	Length =35	Y	022_Txn_UUID
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1	Alphanumeric	Min Length : 1 Max Length : 50	Y	
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1	Alphanumeric	Min Length : 1 Max Length : 35	Y	
4.1.4	URL for the transaction	refUrl	1..1	Alphanumeric	Min Length : 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 35		
4.1.5	Transaction origination time by the creator of the message	ts	1..1	ISODateTime	Min Length : 1 Max Length :255	Y	020_Head_ts
4.1.6	Type of the Transaction	type	1..1	Code	Min Length : 1 Max Length : 20	Y	001_ReqPay_Pay 002_ReqPay_Collect 003_ReqPay_Debit 004_ReqPay_Credit 005_ReqPay_DebitReversal 006_ReqPay_CreditReversal y
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1	Alphanumeric	Length =35	Y	023_Txn_orgTxnId
4.1.8	Customer reference number for the initiated transaction	custRef	1..1	Numeric	Length =12	Y	
4.1.9	Subtype of transaction	subType	0..1	Code	Min Length : 1 Max Length : 20	N	030_Txn_SubType
4.1.10	Initiation mode	InitiationMode	1..1	Code	Min Length : 1 Max Length : 3	Y	031_Txn_Initiation mode

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
4.1.11	OrgRespCode of the transaction	OrgRespCode	0..1	Alphanumeric	Minlength:1 Max length: 20	N	039_ReqPay_OrgRespCode
4.1.12	Purpose of the txn	purpose	1..1	Code	Fixed Value	Y	045_ReqPay_Txn_purpose
4.1.13	Original RRN	orgRrn	0..1	Numeric	Length =12	N	
4.1.14	Original Date of the txn	orgTxnDate	0..1	ISODateTime	Min Length : 1 Max Length :255	N	
4.1.15	Reference category	refCategory	1..n	Code	Fixed Value	Y	052_ReqPay_Txn_refCategory
4.2	Risk Score related to the transaction and the entities	<Txn.RiskScores>	0..1	Alphabetic	Fixed value	N	
4.3	Risk Score related to the transaction and the entities	<Txn.RiskScores.Score>	0..n	Alphabetic	Fixed value	N	
4.3.1	Entity providing the risk score	provider	1..1	Code	Min Length : 1 Max Length : 20	Y	
4.3.2	Type of risk	type	1..1	Code	Min Length : 1 Max Length : 99	Y	
4.3.3	Value of risk evaluation ranging from 0 (No Risk) to 100 (Maximum Risk)	value	1..1	Integer	Min Length : 1 Max Length : 5	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
4.4	Rules that govern the payment	<Txn.Rules>	0..1	Alphabetic	Fixed value	N	
4.5	Rule for the transaction	<Txn.Rules.Rule>	0..n	Alphabetic	Fixed value	N	
4.5.1	Name of the property	name	1..n	Code	Min Length: 1 Max Length: 20	Y	
4.5.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length: 255	Y	
5.1	Details related to the Payer	<Payer>	1..1	Alphabetic	Fixed value	Y	
5.1.1	Address of the Payer	addr	1..1	Alphanumeric	Min Length: 1 Max Length: 255	Y	
5.1.2	Name of the Payer	name	1..1	Alphanumeric	Min Length: 1 Max Length: 99	Y	
5.1.3	Unique identifier for each transaction inside a file including payer and payee	seqNum	1..1	Numeric	Min Length: 1 Max Length: 3	Y	
5.1.4	Type of the Payer	type	1..1	Code	Fixed value	Y	029_Payer/ Payee_Type
5.1.5	Merchant Classification Code - MCC	code	1..1	Numeric	Length =4	Y	024_Txn_code

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
5.16	Merchant block	<payer.Merchant>	0..1	Alphabetic	Fixed value	N	037_ReqPay_Payer/Payee_MerchantTag
5.17	Identifier	<Payer.Merchant.identifier>	0..1	Alphabetic	Fixed value	N	
5.17.1	Subcode	subCode	0..1	Code	Length :4	N	
5.17.2	Merchant identifier	mid	0..1	Alphanumeric	Min Length : 1 Max Length : 20	N	
5.17.3	Store id	sid	0..1	Alphanumeric	Min Length : 1 Max Length : 20	N	
5.17.4	Terminal identifier	tid	0..1	Alphanumeric	Min Length : 1 Max Length : 20	N	
5.17.5	Merchant type	merchantType	1..n	Alphabetic	Fixed value	N	
5.17.6	Merchant Genre	merchantGenre	0..1	Alphabetic	Fixed value	N	
5.17.7	Merchant onboardingType	onBoardingType	0..1	Alphabetic	Fixed value	N	
5.18	Name	<Payer.Merchant.name>	0..1	Alphabetic	Min Length : 1 Max Length : 99	N	
5.18.1	Brand	brand	1..n	Alphanumeric	Min Length : 1 Max Length : 99	Y	
5.18.2	Legal	legal	0..1	Alphanumeric	Min Length : 1 Max	N	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 99		
5.18.3	Franchise	franchise	0..1	Alphanumeric	Min Length : 1 Max Length : 99	N	
5.19	Ownership	<Payer.Merchant.Ownership>	0..1	Alphabetic	Fixed Value	N	
5.19.1	Type	type	0..1	Code	Fixed Value	N	038_ReqPay_Merchant Tag_Ownership_Type
5.20	Institution	<Institution>	1..n	Alphabetic	Fixed value	N	042_ReqPay_Initiation mode
5.20.1	Type	type	1..n	Code	Fixed value	Y	043_ReqPay_Institution_type
5.20.2	Route	route	1..n	Code	Fixed Value	Y	044_ReqPay_Institution_route
5.21	Name	<name>	1..n	Alphabetic	Fixed value	Y	
5.21.1	Value	value	1..n	Alphanumeric	Min Length : 1 Max Length : 100	Y	
5.21.2	acNum	acNum	1..n	Alphanumeric	Min Length : 1 Max Length : 30	Y	
5.22	purpose	<purpose>	1..n	Alphabetic	Fixed Value	Y	
5.22.1	code	code	1..n	Code	Min Length : 1 Max Length : 50	Y	
5.22.2	note	note	1..n	Alphanumeric	Min Length : 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 50		
5.23.	Originator	<Originator>	1..n	Alphabetic	Fixed value	Y	
5.23.1	Name	Name	1..n	Alphanumeric	Min Length : 1 Max Length : 50	Y	
5.23.2.	Type	Type	1..n	Code	Fixed value	Y	
5.23.3	refNo	refNo	1..n	Alphanumeric	Min Length : 1 Max Length : 35	Y	
5.24	address	<address>	1..n	Alphabetic	Fixed value	Y	
5.24.1	Location	location	1..n	Alphanumeric	Min Length :1 Max Length :40	Y	
5.24.2	City	city	1..n	Alphanumeric	Min Length :1 Max Length :100	Y	
5.24.3	Country	country	1..n	Alphanumeric	Min Length :1 Max Length :100	Y	
5.24.4	Geocode	geocode	1..n	Alphanumeric	nn.nnnn,nn.nnnn	Y	
5.25	Beneficiary	<Beneficiary>	1..n	Alphabetic	Fixed value	Y	
5.25.1	name	name	1..n	Alphabetic	Min Length : 1 Max Length : 50	Y	
5.4	Information related	<Payer.Info>	1..1	Alphabetic	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	to the Payer						
5.5	Payer Identity Is mandatory for "pay" and optional for "collect"	<Payer.Info.Identity>	1..1	Alphabetic	Min Length : 1 Max Length : 20	Y	
5.5.1	Id of the identifier	id	1..1	Alphanumeric	Min Length : 1 Max Length : 99	Y	
5.5.2	Type of the identifier	type	1..1	Code	Fixed value	Y	
5.5.3	Name as per the identifier	verifiedName	1..1	Alphanumeric	Min Length : 1 Max Length : 99	Y	
5.6	Rating of the payer	<Payer.Info.Rating>	0..1	Alphabetic	Fixed value	N	
5.6.1	verifiedAddress	verifiedAddress	0..1	Code	Boolean TRUE/FALSE	N	026_Payer/Payee_InfoRating
5.7	Details of Device from which the transaction was initiated	<Payer.Device>	1..1	Alphabetic	Fixed value	Y	
5.8	Device Tag	<Payer.Device.Tag>	1..n	Alphabetic	Fixed value	Y	
5.8.1	Name of the property	name	1..n	code(MOBILE,GEOCODE,LOCATION,IP,TYPE,ID,OS,APP,CAPABILITY,TELECOM	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
				OPERATOR)			
5.8.2	Value of the property	value	1..n	Alphanumeric	Min Length : 1 Max Length : 20	Y	034_ReqPay_DeviceDetails_Values 035_ReqPay_DeviceDetails_type 036_ReqPay_DeviceDetails_OS
5.9	Only one entity is allowed for a payer	<Payer. Ac>	1..1	Alphabetic	Fixed value	Y	
5.9.1	Type of the address	addrType	1..1	Code	Min Length : 1 Max Length : 20	Y	046_ReqPay_Ac_addrType
5.10	Details related to Payer Address	<Payer. Ac.Detail>	1..n	Alphabetic	Min Length : 1 Max Length : 255	Y	
5.10.1	Name of the property	name	1..n	Code	Fixed value	Y	047_ReqPay_Ac_name_Aadhaar 048_ReqPay_Ac_name_Account 049_ReqPay_Ac_name_Mobile 050_ReqPay_Ac_name_Card
5.10.2	Value of the property	value	1..n	Alphanumeric	Min Length : 1 Max Length : 20	Y	
5.11	Information related to Payer	<Payer. Creds>	1..1	Alphabetic	Min Length : 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	Credentials				Length : 20		
5.12	Credentials are used to authenticate the request	<Payer. Creds.Cred>	1..1	Alphabetic	Min Length : 1 Max Length : 20	Y	040_ReqPay_Credblock 041_RespAuthDetail UPI-mandate_CollectCredblock 007_ReqPay_PreApproved 025_Response_ApprovalNum
5.12.1	Type of financial instrument used for authentication	type	1..1	Code	Fixed value	Y	
5.12.2	subType	subType	1..1	Code	Fixed value	Y	040_ReqPay_Credblock
5.13	base-64 encoded/ encrypted authentication data	<Payer. Creds.Cred.Data>	1..1	Alphabetic	Fixed value	Y	
5.13.1	Data Code	Data. Code	1..1	Code	Fixed value	Y	
5.13.2	Key Index	Ki	1..1	Code	Fixed Value	Y	
5.13.3	Meta tag for Aadhaar transaction	<Meta>	1..1	Alphabetic	Fixed value	Y	
5.13.3.1	License Key assigned to the AUA	lk	1..1	AlphaNumeric	Max Length :64	Y	
5.13.3.2	A unique code for AUA	ac	1..1	AlphaNumeric	Max Length :10	Y	
5.13.3.3	A unique sub_AUA code	sa	1..1	AlphaNumeric	Max Length :10	Y	
5.13.3.4	Aadhaar number of the person being	uid	1..1	AlphaNumeric	Max Length : 12	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	authenticated						
5.13.3.5	Version of the API	ver	1..1	Numeric	Min Length: 1 Max Length: 6	Y	
5.14	Information related to the amounts in the transaction	<Payer.Amount>	1..1	Alphabetic	Fixed value	Y	
5.14.1	Transaction amount	value	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
5.14.2	Currency of the transaction	curr	1..1	Text	Min Length: 1 Max Length: 3	Y	
5.15	Details of transaction amount	<Payer.Amount.Split>	0..1	Alphabetic	Fixed value	N	
5.15.1	Name of the property	name	1..n	Code	Min Length: 1 Max Length: 20	Y	
5.15.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length: 99	Y	
6.1	Details related to the Payees	<Payees>	1..1	Alphabetic	Fixed value	Y	
6.2	Details related to the Payee	<Payee>	1..1	Alphabetic	Fixed value	Y	
6.2.1	Address of the Payee	addr	1..1	Alphanumeric	Min Length: 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 255		
6.2.2	Name of the Payee	name	1..1	AlphaNumeric	Min Length : 1 Max Length : 99	Y	
6.2.3	Unique identifier for each transaction inside a file including Payee and payee	seqNum	1..1	Numeric	Min Length : 1 Max Length : 3	Y	
6.2.4	Type of the Payee	type	1..1	Code	Fixed Value	Y	029_Payer/Payee_Type
6.4	Payee Identity	<Payee.Info.Identity>	1..1	Alphabetic	Fixed value	Y	
6.4.1	Type of the identifier	type	1..1	Code	Fixed value	Y	
6.4.2	Name as per the identifier	verifiedName	1..1	Alphanumeric	Min Length : 1 Max Length : 99	Y	
6.4.3	Id of the identifier	id	1..1	Alphanumeric	Min Length : 1 Max Length : 99	Y	
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1	Alphabetic	Fixed value	N	
6.5.1	verifiedAddress	verifiedAddress	0..1	Code	Boolean TRUE/ FALSE	N	026_Payer/Payee_InfoRating
6.6	Details of Device from which the transaction was initiated	<Payee.Device>	1..1	Alphabetic	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
6.7	Device Tag	<Payee.Device.Tag>	1..n	Alphabetic	Fixed value	Y	
6.7.1	Name of the property	name	1..n	Code	Fixed value	Y	
6.7.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length: 99	Y	
6.8	Only one entity is allowed for a Payee	<Payee.Ac>	1..1	Alphabetic	Fixed value	Y	
6.8.1	Type of the address	addrType	1..1	Code	Min Length: 1 Max Length: 20	Y	
6.9	Details related to Payee Address	<Payee.Ac.Detail>	1..n	Alphabetic	Fixed value	Y	
6.9.1	Name of the property	name	1..n	Code	Fixed value	Y	
6.9.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length: 99	Y	
6.10	Information related to the amounts in the transaction	<Payee.Amount>	1..1	Alphabetic	Fixed value	Y	
6.10.1	Transaction amount	value	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
6.10.2	Currency of the transaction	curr	1..1	Text	Min Length: 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 3		
6.11	Details of transaction amount	<Payee.Amount.Split>	0..1	Alphabetic	Fixed value	N	
6.11.1	Name of the property	name	1..n	Code	Fixed value	Y	
6.11.2	Value of the property	value	1..n	Alphanumeric	Min Length : 1 Max Length : 99	Y	

6.2.2 RespPay

Complete XML structure for response API (RespPay) is given below.

```
<upi:RespPay xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" custRef="" refUrl=""ts=""
purpose="00|01|02|03|04|05|06|07|08|09|10"type="PAY|COLLECT|DEBIT|CREDIT|REVERSAL|REF
UND"subType="" initiationMode=""orgTxnId="" orgRrn="" orgTxnDate=""
refCategory="00|01|02|03|04|05|06|07|08|09">
<RiskScores>
<Score provider="sp" type="TXNRISK" value=""/>
<Score provider="npci" type="TXNRISK" value=""/>
</RiskScores>
<Resp reqMsgId="" result="SUCCESS|FAILURE|PARTIAL|DEEMED" errCode="" actn="">
<Ref type="PAYER" seqNum="" addr="" regName="" acNum="" IFSC="" code="" accType="
SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"
settAmount="" orgAmount="" settCurrency="" approvalNum="" respCode="" reversalRespCode=""/>
<Ref type="PAYEE" seqNum="" addr="" regName="" acNum ="" IFSC=""code="" accType ="
SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD" settAmount=""
orgAmount="" settCurrency="" approvalNum="" respCode=""reversalRespCode="" />
</Resp>
</upi:RespPay>
```

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
1.1	API Name	<RespPay>	1..1			Y	
1.1.1	API Schema namespace	Xmlns	1..1	Alphanumeric	Min Length: 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length: 255		
2.1	Header for the message	<Head>	1..1	Alphabetic	Fixed value	Y	
2.1.1	Version of the API	Ver	1..1	Numeric	Min Length: 1 Max Length : 6	Y	019_Head_Version
2.1.2	Time of request from the creator of the message	Ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
2.1.3	Organization id that created the message	orgId	1..1	Numeric	Min Length: 1 Max Length : 20	Y	
2.1.4	Message identifier- used to correlate between request and response	msgId	1..1	Alphanumeric	Length= 35	Y	021_Head_MsgId
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1	Alphabetic	Fixed value	Y	
4.1.1	Unique Identifier of the transaction across all entities created by the originator	id	1..1	Alphanumeric	Length= 35	Y	022_Txn_UUID
4.1.2	Description of the transaction (which will be printed	note	1..1	Alphanumeric	Min Length: 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	on Pass book)				Length : 50		
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.4	URL for the transaction	refUrl	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.5	Transaction origination time by the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
4.1.6	Type of the Transaction	type	1..1	Code	Min Length: 1 Max Length : 20	Y	016_RespPay_Pay 017_RespPay_Collect 018_RespPay_Reversal
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1	Alphanumeric	Length= 35	Y	023_Txn_ orgTxnId
4.1.9	Subtype of transaction	subType	0..1	Code	Min Length: 1 Max Length: 20	N	030_Txn_SubType
4.1.10	Initiation mode	initiation mode	1..1	Code	Min Length: 1 Max Length: 3	Y	031_Txn_Initiation mode
11.1	Response	<Resp>	1..1	Alphabetic	Fixed value	Y	
11.1.1	Request Message identifier	reqMsgId	1..1	Alphanumeric	Length= 35	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
11.1.2	Result of the transaction	result	1..1	Code	Minlength:1 Max length:20	Y	
11.1.3	Error code if failed	errCode	1..1	Alphanumeric	Minlength:1 Max length:20	Y	027_Response_ErrCode
11.1.4	Authentication code	actn	1..n	Numeric	Minlength:1 Max length:40	Y	033_RespPay_ActCode
11.2	Response Reference	<Ref>	1..n	Alphabetic	Fixed value	Y	
11.2.1	Reference type	type	1..1	Code	Fixed value	Y	016_RespPay_Pay 017_RespPay_Collect
11.2.2	Sequence Number	seqNum	1..1	Numeric	Minlength:1 Max length:3	Y	
11.2.3	Payment address	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
11.2.4	Settlement Amount	settAmount	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
11.2.5	Settlement Currency	settCurrency	1..1	Text	Min Length: 1 Max Length : 3	Y	
11.2.6	Approval Reference Number	approvalNum	1..1	Alphanumeric	Length= 6	Y	025_Response_ApprovalNum
11.2.7	Response code	respCode	1..1	Alphanumeric	Min Length: 1 Max Length : 20	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
11.2.8	Registered name with bank	regName	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
11.2.9	Original amount	orgAmount	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
11.2.10	Reversal Response Code	reversalRespCode	1..1	Alphanumeric	Min Length: 1 Max Length : 20	Y	028_Response_Reversal
11.2.11	Account number	acNum	1..1	Alphanumeric	Min Length: 1 Max Length : 30	Y	
5.1.5	Merchant Classification Code - MCC	code	1..1	Numeric	Length= 4 digit	Y	024_Txn_code
11.2.12	IFSC code	IFSC	1..n	Alphanumeric	Length :11	Y	032_RespPay_RefTag_IFSC
11.2.13	Account type	ACTYPE	1..n	Code	Fixed Value	Y	048_ReqPay_Ac_name_Account

6.2.3 ReqAuthDetails

Input message XML for ReqAuthDetails API.

```
<upi:ReqAuthDetails xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="NPCI" msgId=""/>
        <Txn id="" note="" refId="" custRef="" refUrl="" ts=""
type="PAY|COLLECT|DEBIT|CREDIT|REVERSAL|REFUND"
initiationMode=""purpose="00|01|02|03|04|05|06|07|08|09|10"
refCategory="00|01|02|03|04|05|06|07|08|09" >
```

```
<RiskScores>
<Score provider="sp" type="TXNRISK" value=""/>
<Score provider="NPCI" type="TXNRISK" value=""/>
</RiskScores>
<Rules>
<Rule name="EXPIREAFTER" value="1 miniute to max 64800 minitues"/>
<Rule name="MINAMOUNT" value=""/>
</Rules>

</Txn>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">

<Merchant >
        <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
        merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR"/>
        <Name brand="" legal="" franchise=""/>
        <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
<Info>
<Identity id=""type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail                                                                    name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET| SOD|UOD "/>
<Detail name="ACNUM" value=""/>
</Ac>

<Amount value="" curr="INR">
<Split name="PURCHASE|CASHBACK" value=""/>
</Amount>
</Payer>
<Payees>
<Payee seqNum="" addr="" name="" type="PERSON|ENTITY" code="">
  <Info>
<Identity id=""type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
        <Ac addrType="ACCOUNT">
        <Detail name="IFSC" value=""/>
        <Detail                                                                    name="ACTYPE"
        value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|
        SOD|UOD "/>
        <Detail name="ACNUM" value=""/>
        </Ac>

        <Amount value="" curr="INR">
        <Split name="PURCHASE|CASHBACK" value=""/>
        </Amount>
```

</Payee>
</Payees>
</upi:ReqAuthDetails>

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
1.1	API Name	<ReqAuthDetails>	1..1			Y	
1.1.1	API Schema namespace	xmlns	1..1	Alphanumeric	Min Length: 1 Max Length: 255	Y	
2.1	Header for the message	<Head>	1..1	Alphabetic	Fixed value	Y	
2.1.1	Version of the API	Ver	1..1	Numeric	Min Length: 1 Max Length : 6	Y	019_Head_Version
2.1.2	Time of request from the creator of the message	Ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
2.1.3	Organization id that created the message	orgId	1..1	Numeric	Min Length: 1 Max Length : 20	Y	
2.1.4	Message identifier- used to correlate between request and response	msgId	1..1	Alphanumeric	Length=35	Y	021_Head_MsgId
4.1	Transaction information, Carried throughout the	<Txn>	1..1	Alphabetic	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	system, visible to all parties						
4.1.1	Unique Identifier of the transaction across all entities, created by the originator	id	1..1	Alphanumeric	Length=35	Y	022_Txn_UUID
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1	Alphanumeric	Min Length: 1 Max Length : 50	Y	
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.4	URL for the transaction	refUrl	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.5	Transaction origination time by the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
4.1.6	Type of the Transaction	type	1..1	Code	Min Length: 1 Max	Y	008_ReqAuth_Pay 009_ReqAuth_Collect

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 20		
4.1.7	Original transaction ID when reversal/ Refund has to be done	orgTxnId	1..1	Alphanumeric	Length=35	Y	023_Txn_orgTxnId
4.1.8	Customer reference number for the initiated transaction	custRef	1..1	Numeric	Length=12	Y	
4.1.10	Initiation mode	Initiation mode	1..1	Code	Min Length: 1 Max Length: 3	Y	031_Txn_Initiation mode
4.2	Risk Score related to the transaction and the entities	<Txn.RiskScores>	0..1	Alphabetic	Fixed value	N	
4.3	Risk Score related to the transaction and the entities	<Txn.RiskScores.Score>	0..n	Alphabetic	Fixed value	N	
4.3.1	Entity providing the risk score	provider	1..1	Code	Min Length: 1 Max Length : 20	Y	
4.3.2	Type of risk	type	1..1	Code	Min Length: 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 99		
4.3.3	Value of risk evaluation ranging from 0 (No Risk) to 100 (Maximum Risk)	value	1..1	Integer	Min Length: 1 Max Length : 5	Y	
4.4	Rules that govern the payment	<Txn.Rules>	0..1	Alphabetic	Fixed value	N	
4.5	Rule for the transaction	<Txn.Rules.Rule>	0..n	Alphabetic	Fixed value	N	
4.5.1	Name of the property	name	1..n	Code	Min Length: 1 Max Length : 255	Y	
4.5.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.1	Details related to the Payer	<Payer>	1..1	Alphabetic	Fixed value	Y	
5.1.1	Address of the Payer	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
5.1.2	Name of the Payer	name	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.1.3	Unique identifier for each	seqNum	1..1	Numeric	Min Length: 1	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	transaction inside a file including payer and payee				Max Length : 3		
5.1.4	Type of the Payer	type	1..1	Code	Fixed value	Y	029_Payer/Payee_Type
5.1.5	Merchant Classification Code - MCC	code	1..1	Numeric	Length :4	Y	024_Txn_code
5.4	Information related to the Payer	<Payer.Info>	1..1	Alphabetic	Fixed value	Y	
5.5	Payer Identity is mandatory for "pay" and optional for "collect"	<Payer.Info.Identity>	1..1	Alphabetic	Fixed value	Y	
5.5.1	Id of the identifier	id	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.5.2	Type of the identifier	type	1..1	Code	Fixed value	Y	
5.5.3	Name as per the identifier	verifiedName	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.6	Rating of the payer	<Payer.Info.Rating>	0..1	Alphabetic	Fixed value	N	
5.6.1	verifiedAddress	verifiedAddress	0..1	Code	Boolean TRUE/ FALSE	N	026_Payer/Payee_InfoRating

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
5.9	Only one entity is allowed for a payer	<Payer.Ac>	1..1	Alphabetic	Fixed value	Y	
5.9.1	Type of the address	addrType	1..1	Code	Min Length: 1 Max Length : 20	Y	046_ReqPay_Ac_addrType
5.10	Details related to Payer Address	<Payer.Ac.Detail>	1..n	Alphabetic	Fixed value	Y	
5.10.1	Name of the property	name	1..n	Code	Fixed value	Y	047_ReqPay_Ac_name_Aadhaar 048_ReqPay_Ac_name_Account 049_ReqPay_Ac_name_Mobile 050_ReqPay_Ac_name_Card
5.10.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 20	Y	
5.14	Information related to the amounts in the transaction	<Payer.Amount>	1..1	Alphabetic	Fixed value	Y	
5.14.1	Transaction amount	value	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
5.14.2	Currency of the transaction	curr	1..1	Text	Min Length: 1 Max	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 3		
5.15	Details of transaction amount	<Payer.Amount.Split>	0..1	Alphabetic	Fixed value	N	
5.15.1	Name of the property	name	1..n	Code	Fixed value	Y	
5.15.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.1	Details related to the Payees	<Payees>	1..1	Alphabetic	Fixed value	Y	
6.2	Details related to the Payee	<Payee>	1..1	Alphabetic	Fixed value	Y	
6.2.1	Address of the Payee	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
6.2.2	Name of the Payee	name	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.2.3	Unique identifier for each transaction inside a file including Payee and payee	seqNum	1..1	Alphanumeric	Min Length: 1 Max Length : 3	Y	
6.2.4	Type of the Payee	type	1..1	Numeric	Fixed value	Y	029_Payer/Payee_Type
6.2.5	Merchant Classification	code	1..1	Numeric	Length=4	Y	024_Txn_code

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	Code - MCC						
6.3	Information related to the Payee	<Payee.Info>	1..1	Alphabetic	Fixed value	Y	
6.4	Payee Identity	<Payee.Info.Identity>	1..1	Alphabetic	Fixed value	Y	
6.4.1	Type of the identifier	type	1..1	Code	Fixed value	Y	
6.4.2	Name as per the identifier	verifiedName	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.4.3	Id of the identifier	id	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1	Alphabetic	Fixed value	N	
6.5.1	verifiedAddress	verifiedAddress	0..1	Code	Boolean TRUE/ FALSE	N	026_Payer/Payee_InfoRating
6.8	Only one entity is allowed for a Payee	<Payee.Ac>	1..1	Alphabetic	Fixed value	Y	
6.8.1	Type of the address	addrType	1..1	Code	Min Length: 1 Max Length : 20	Y	
6.9	Details related to Payee Address	<Payee.Ac.Detail>	1..n	Alphabetic	Fixed value	Y	
6.9.1	Name of the property	name	1..n	Code	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
6.9.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.10	Information related to the amounts in the transaction	<Payee.Amount>	1..1	Alphabetic	Fixed value	Y	
6.10.1	Transaction amount	value	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
6.10.2	Currency of the transaction	curr	1..1	Text	Min Length: 1 Max Length : 3	Y	
6.11	Details of transaction amount	<Payee.Amount.Split>	0..1	Alphabetic	Fixed value	N	
6.11.1	Name of the property	name	1..n	Code	Fixed value	Y	
6.11.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	

6.2.4 RespAuthDetails

Following is the XML data format for RespAuthDetails API.

```
<upi:RespAuthDetails xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
        <Txn id="" note="" refId="" custRef="" refUrl="" ts="" type="PAY|COLLECT|DEBIT|CREDIT|REVERSAL|REFUND"
```

refCategory="00|01|02|03|04|05|06|07|08|09" initiationMode=""
purpose="00|01|02|03|04|05|06|07|08|09|10" >
<RiskScores>
<Score provider="sp" type="TXNRISK" value=""/>
<Score provider="NPCI" type="TXNRISK" value=""/>
</RiskScores>
<Rules>
<Rule name="EXPIREAFTER" value="1 miniute to max 64800 minitues"/>
<!—If EXPIREAFTER is not provided default value will be taken as 30 minutes – >
<Rule name="MINAMOUNT" value=""/>
</Rules>
</Txn>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">

<Merchant >
<Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR"/>
<Name brand="" legal="" franchise=""/>
<Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/..."/>
</Device>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET| SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Creds>
< Cred type="AADHAAR" subType="AADHAAR-BIO-FP|AADHAAR-BIO-IRIS|AADHAAR-BIO-OTP">
<Meta lk="" ac="" sa="" uid="" ver=""/>
<Datacode="" ki=""> base-64 encoded/encrypted authentication data</Data>

```
</Cred>
<Cred type="PIN" subType="MPIN">
                    <Datacode="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>

        <Cred type="PREAPPROVED" subType="NA">
                    <Data> base-64 encoded</Data>
                    <!– #data includes respCode and approvalRef
                    In the format "respCode|approvalNum"
                    – >
<Cred type="UPI-Mandate" subType="DS">
                    <Data> base-64 encoded digitally signed UPI-Mandate</Data>
                    <! – This cred block is applicable only for the UPI-mandate txn – >
</Cred>
</Creds>
<Amount value="" curr="INR">
<Split name="PURCHASE|CASHBACK" value=""/>
</Amount>
</Payer>
<Payees>
<Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">

<Merchant >
        <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
        merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR"/>
        <Name brand="" legal="" franchise=""/>
        <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>

<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" /> <! —It is mandatory for Merchant for payee– >
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/..."/>
</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
```

<Detail name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Amount value="" curr="INR">
<Split name="PURCHASE|CASHBACK" value=""/>
</Amount>
</Payee>
</Payees>
</upi:RespAuthDetails>

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
1.1	API Name	<RespAuthDetails>	1..1			Y	
1.1.1	API Schema namespace	xmlns	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
2.1	Header for the message	<Head>	1..1	Alphabetic	Fixed value	Y	
2.1.1	Version of the API	ver	1..1	Numeric	Min Length: 1 Max Length : 6	Y	019_Head_Version
2.1.2	Time of request from the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
2.1.3	Organization id that created the message	orgId	1..1	Numeric	Min Length: 1 Max Length : 20	Y	
2.1.4	Message identifier- used to correlate between request and response	msgId	1..1	Alphanumeric	Length= 35	Y	021_Head_MsgId

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1	Alphabetic	Fixed value	Y	
4.1.1	Unique Identifier of the transaction across all entities, created by the originator	id	1..1	Alphanumeric	Length=35	Y	022_Txn_UUID
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1	Alphanumeric	Min Length: 1 Max Length : 50	Y	
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.4	URL for the transaction	refUrl	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.5	Transaction origination time by the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
4.1.6	Type of the	type	1..1	Code	Min Length: 1	Y	010_RespAuth_Pay 011_RespAuth_Collect

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	Transaction				Max Length : 20		
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1	Alphanumeric	Length=35	Y	023_Txn_ orgTxnId
4.1.8	Customer reference number for the initiated transaction	custRef	1..1	Numeric	Length=12	Y	
4.1.10	Initiation mode	Initiation mode	1..1	Code	Min Length: 1 Max Length: 3	Y	031_Txn_Initiation mode
4.2	Risk Score related to the transaction and the entities	<Txn.RiskScores>	0..1	Alphabetic	Fixed value	N	
4.3	Risk Score related to the transaction and the entities	<Txn.RiskScores.Score>	0..n	Alphabetic	Fixed value	N	
4.3.1	Entity providing the risk score	provider	1..1	Code	Min Length: 1 Max Length : 20	Y	
4.3.2	Type of risk	type	1..1	Code	Min Length: 1 Max Length : 99	Y	
4.3.3	Value of risk	value	1..1	Integer	Min Length:	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	evaluation ranging from 0 (No Risk) to 100 (Maximum Risk)				1 Max Length : 5		
4.4	Rules that govern the payment	<Txn.Rules>	0..1	Alphabetic	Fixed value	N	
4.5	Rule for the transaction	<Txn.Rules.Rule>	0..n	Alphabetic	Fixed value	N	
4.5.1	Name of the property	name	1..n	Code	Min Length: 1 Max Length : 20	Y	
4.5.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 255	Y	
5.1	Details related to the Payer	<Payer>	1..1	Alphabetic	Fixed value	Y	
5.1.1	Address of the Payer	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
5.1.2	Name of the Payer	name	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.1.3	Unique identifier for each transaction inside a file including	seqNum	1..1	Numeric	Min Length: 1 Max Length : 3	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	payer and payee						
5.1.4	Type of the Payer	type	1..1	Code	Fixed value	Y	029_Payer/Payee_Type
5.1.5	Merchant Classification Code – MCC	Code	1..1	Numeric	Length=4	Y	024_Txn_code
5.4	Information related to the Payer	<Payer.Info>	1..1	Alphabetic	Fixed value	Y	
5.5	Payer Identity Is mandatory for "pay" and optional for "collect"	<Payer.Info.Identity>	1..1	Alphabetic	Fixed value	Y	
5.5.1	Id of the identifier	id	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.5.2	Type of the identifier	type	1..1	Code	Fixed value	Y	
5.5.3	Name as per the identifier	verifiedName	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
5.6	Rating of the payer	<Payer.Info.Rating>	0..1	Alphabetic	Fixed value	N	
5.6.1	verifiedAddress	verifiedAddress	0..1	Code	Boolean TRUE/FALSE	N	026_Payer/Payee_InfoRating
5.9	Only one entity is allowed for a payer	<Payer.Ac>	1..1	Alphabetic	Fixed value	Y	
5.9.1	Type of the address	addrType	1..1	Code	Min Length: 1 Max	Y	046_ReqPay_Ac_addrType

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 20		
5.10	Details related to Payer Address	<Payer.Ac.Detail>	1..n	Alphabetic	Fixed value	Y	
5.10.1	Name of the property	name	1..n	Code	Fixed value	Y	047_ReqPay_Ac_name_Aadhaar 048_ReqPay_Ac_name_Account 049_ReqPay_Ac_name_Mobile 050_ReqPay_Ac_name_Card
5.10.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 20	Y	
5.11	Information related to Payer Credentials	<Payer.Creds>	1..1	Alphabetic	Fixed value	Y	
5.12	Credentials are used to authenticate the request	<Payer.Creds.Cred>	1..1	Alphabetic	Fixed value	Y	040_ReqPay_Credblock 041_RespAuthDetail UPI-mandate_CollectCredblock 007_ReqPay_PreApproved 025_Response_ApprovalNum
5.12.1	Type of financial instrument used for authentication	type	1..1	Code	Fixed value	Y	
5.12.2	subType	subType	1..1	Code	Fixed value	Y	040_ReqPay_Credblock
5.13	base-64 encoded/ encrypted authentication data	<Payer.Creds.Cred.Data>	1..1	Alphabetic	Fixed value	Y	
5.13.1	Data Code	Data. Code	1..1	Code	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
5.13.2	Key Index	Ki	1..1	Code	Fixed Value	Y	
5.14.2	Currency of the transaction	curr	1..1	Text	Min Length: 1 Max Length : 3	Y	
5.15	Details of transaction amount	<Payer.Amount.Split>	0..1	Alphabetic	Fixed value	N	
5.15.1	Name of the property	name	1..n	Code	Min Length: 1 Max Length : 20	Y	
5.15.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.1	Details related to the Payees	<Payees>	1..1	Alphabetic	Fixed value	Y	
6.2	Details related to the Payee	<Payee>	1..1	Alphabetic	Fixed value	Y	
6.2.1	Address of the Payee	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.2.2	Name of the Payee	name	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.2.3	Unique identifier for each transaction inside a file including	seqNum	1..1	Numeric	Min Length: 1 Max Length : 3	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	Payee and payee						
6.2.4	Type of the Payee	type	1..1	Code	Fixed value	Y	029_Payer/Payee_Type
6.2.5	Merchant Classification Code – MCC	code	1..1	Code	Length=4	Y	024_Txn_code
6.3	Information related to the Payee	<Payee.Info>	1..1	Alphabetic	Fixed value	Y	
6.4	Payee Identity	<Payee.Info.Identity>	1..1	Alphabetic	Fixed value	Y	
6.4.1	Type of the identifier	type	1..1	Code	Fixed value	Y	
6.4.2	Name as per the identifier	verifiedName	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.4.3	Id of the identifier	id	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1	Alphabetic	Fixed value	N	
6.5.1	verifiedAddress	verifiedAddress	0..1	Code	Boolean TRUE/FALSE	N	026_Payer/Payee_InfoRating
6.8	Only one entity is allowed for a Payee	<Payee.Ac>	1..1	Alphabetic	Fixed value	Y	
6.8.1	Type of the address	addrType	1..1	Code	Min Length: 1 Max Length : 20	Y	
6.9	Details related to Payee Address	<Payee.Ac.Detail>	1..n	Alphabetic	Fixed value	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
6.9.1	Name of the property	name	1..n	Code	Fixed value	Y	
6.9.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	
6.10	Information related to the amounts in the transaction	<Payee.Amount>	1..1	Alphabetic	Fixed value	Y	
6.10.1	Transaction amount	value	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
6.10.2	Currency of the transaction	curr	1..1	Text	Min Length: 1 Max Length : 3	Y	
6.11	Details of transaction amount	<Payee.Amount.Split>	0..1	Alphabetic	Fixed value	N	
6.11.1	Name of the property	name	1..n	Code	Fixed value	Y	
6.11.2	Value of the property	value	1..n	Alphanumeric	Min Length: 1 Max Length : 99	Y	

6.3 Meta APIs

In addition to transactional APIs described above, a set of Meta APIs are required to ensure the entire system can function in an automated fashion. These Meta APIs allow PSPs to validate accounts during customer on boarding, validate addresses for sending and

collecting money, provide phishing protection using whitelisting APIs, etc. Following are the list of Meta APIs proposed as part of this unified interface.

6.3.1 List PSP

NPCI will maintain the list of all registered PSPs and their details. This API allows the PSPs to request the list of all registered PSPs for local caching. This data should be used for validating payment address before initiating the transaction. Ss

ReqListPsp: Request PSP list

```
<upi:ReqListPsp xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="ListPsp" />
</upi:ReqListPsp>
```

RespListPsp: Response for PSP list

```
<upi:RespListPsp xmlns:upi="http://npci.org/upi/schema/">
          <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
              <Txn id="" note="" refId="" refUrl="" ts="" type="ListPsp" />
            <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
          <PspList>
<Psp name="HDFC" codes="hdfcgold,hdfcsliver" active="Y/N" url="" spocName="" spocEmail="" spocPhone="" lastModifedTs="" >
<VersionSupported>
<Version no="1.0" description="UPI 1.0 BASE VERSION" mandatory="TRUE"/>
<Version no="2.0" description="UPI 2.0: ALL TAG LEVEL CHANGES" mandatory="TRUE"/> <! – If mandatory="TRUE", then psp should be live in this root version before going live with next or any child version (2.1, 2.2, ...) – >
<Version no="2.1" description="MANDATE"/>
<Version no="2.2 description="REFUND"/>
<Version no="2.3 description="AADHAAR"/>
</VersionSupported>
 </Psp>

<Psp name="ICICI" codes="icici,iciciwallet" active="Y/N" url="" spocName="" spocEmail="" spocPhone="" lastModifedTs="" >
<VersionSupported>
<Version no="1.0" description="UPI 1.0 BASE VERSION" mandatory="TRUE"/>
```

<Version no="2.0" description="UPI 2.0: ALL TAG LEVEL CHANGES" mandatory="TRUE"/> <! – If mandatory="TRUE", then psp should be live in this root version before going live with next or any child version (2.1, 2.2, ...) – >
<Version no="2.1" description="MANDATE"/>
<Version no="2.2 description="REFUND"/>
<Version no="2.3 description="AADHAAR"/>
</VersionSupported>
</Psp>
</PspList>
</upi:RespListPsp>

Tag Num	Message Item	<XML Tag>	Occurrence
21.1	PSP List	<PspList>	1..1
21.2	Details related to registered PSP	<PspList.Psp>	1..1
21.2.1	Name of the PSP	Name	1..1
21.2.2	Codes defined for the PSP	Codes	1..n
21.2.3	Status of the PSP if it is active or not	Active	1..1
21.2.4	URL link provided by PSP	url	0..n
21.2.5	Name of the SPOC	spocName	0..n
21.2.6	E-mail of the SPOC	spocEmail	0..n
21.2.7	Phone Number of the SPOC	spocPhone	0..n
21.2.8	Last Modified date of the PSP information in the UPI system	lastModifiedTs	1..1
21.2.9	Version supported	<VersionSupported>	1..1
21.2.10	Details of versioning	<Version>	1..n
21.2.11	Version Number	no	1..n
21.2.12	Version descriptions	description	1..n
21.2.13	Description of mandatory flag	mandatory	1..n

6.3.2 List Account Providers

NPCI will maintain the list of all account providers who are connected via unified interface. PSPs should maintain the list and check for registered account providers before registering a customer account within their application.

In addition to the exiting, List Account Provider API will provide the issuer capability to verify ATM PIN for all the registered PSP with UPI. A new attribute "regMobFormat" has been introduced for the same.

ReqListAccPvd: Request for Account Providers list

```
<upi:ReqListAccPvd xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="ListAccPvd" />

</upi:ReqListAccPvd>
```

RespListAccPvd: Response for Account providers list

```
<upi:RespListAccPvd xmlns:upi="http://npci.org/upi/schema/">
          <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="ListAccPvd" />
           <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
         <AccPvdList>
<AccPvd name="HDFC" iin="901345" ifsc="" active="Y/N" url="" spocName=""      spocEmail=""
spocPhone=""  prods="AEPS,IMPS,CARD,NFS"  lastModifedTs=""  mobRegFormat="FORMAT1|FORMAT2|
ATM_REDIRECT" >
<! –ATMREDIRECT will be populated for the remitter banks who want the ATMPIN to be captured in their
bank page – >
<VersionSupported>
<Version no="1.0" description="UPI 1.0 BASE VERSION" mandatory="TRUE"/>
<Version no="2.0" description="UPI 2.0: ALL TAG LEVEL CHANGES" mandatory="TRUE"/> <! – If
mandatory="TRUE", then psp should be live in this root version before going live with next or any child
version (2.1, 2.2, ...). Head and URL version be always root version – >
<Version no="2.1" description="MANDATE"/>
<Version no="2.2 description="REFUND"/>
<Version no="2.3 description="AADHAAR"/>
</VersionSupported>
</AccPvd>
<AccPvd name="ICICI" iin="901346" ifsc="" active="Y/N" url="" spocName=""      spocEmail=""
spocPhone="" prods="AEPS,IMPS,CARD,NFS" lastModifedTs="" mobRegFormat ="FORMAT1|FORMAT2" >
<VersionSupported>
<Version no="1.0" description="UPI 1.0 BASE VERSION" mandatory="TRUE"/>
<Version no="2.0" description="UPI 2.0: ALL TAG LEVEL CHANGES" mandatory="TRUE"/> <! – If
mandatory="TRUE", then psp should be live in this root version before going live with next or any child
version (2.1, 2.2, ...) – >
<Version no="2.1" description="MANDATE"/>
<Version no="2.2 description="REFUND"/>
```

<Version no="2.3 description="AADHAAR"/>
</VersionSupported>
</AccPvd>
</AccPvdList>
</upi:RespListAccPvd>

Tag Num	Message Item	<XML Tag>	Occurrence
22.1	Account providers List	<AccPvdList>	1..1
22.2	Details of registered Account providers List	<AccPvdList.AccPvd>	1..1
22.2.1	Name of the Account Provider	name	1..1
22.2.2	IIN of Account provider	iin	1..n
22.2.3	IFSC	Ifsc	1..n
22.2.4	Status of the account provider if it is active or not	active	1..1
22.2.5	URL link provided by account provider	url	0..n
22.2.7	Name of the SPOC	spocName	0..n
22.2.8	E-mail of the SPOC	spocEmail	0..n
22.2.9	Phone Number of the SPOC	spocPhone	0..n
22.2.10	List of NPCI products for which account provider is live	prods	0..n
22.2.11	Last Modified date of the account provider information in the UPI system	lastModifiedTs	1..1
22.2.12	Register format of the account provider information in the UPI system	regMobFormat	1..1
22.2.13	Version supported	<VersionSupported>	1..1
22.2.14	Details of versioning	<Version>	1..n
22.2.15	Version Number	No	1..n
22.2.16	Version descriptions	Description	1..n
22.2.17	Description of mandatory flag	Mandatory	1..n

6.3.3 List Keys

NPCI maintains the list of all public keys for encryption. This API allows the PSPs to request and cache the list of public keys of NPCI and UIDAI. Trusted and certified libraries will be used by PSPs for credential capture and PKI public key encryption at capture time. These libraries will be provided by NPCI.

ReqListKeys: Request list of Key's

<upi:ReqListKeys xmlns:upi="http://npci.org/upi/schema/">

```
<Head ver="1.0|2.0" ts="" orgId="" msgId="" pageSize="5000"/> <! — the default page size will be 1000, if psp wants to change they can change the required page value between min="1000" to max="10000" – >/>
<Txn id="" note="" refId="" refUrl="" ts="" type="ListKeys/GetToken/ListPspKeys"pspOrgId=""/>
<! – If type="ListPspKeys", the field "pspOrgId" is used to get the public signed intent key of the respective psp involved in signed intent call. It is an optional field. If pspOrgId is not populated, UPI will provide all the psp signed intent keys. Psp should fire ReqIsit keys once in a day->
<Creds>
        <Cred type="challenge" subType="initial/reset/rotate">
                <data code="" ki=""></data>
        </Cred>
</Creds>
</upi:ReqListKeys>
```

Tag Num	Message Item	<XML Tag>	Occurrence
2.1.5	Page size	<Head.pageSize>	1..1

RespListKeys: Response for List of Key's

```
<upi: RespListKeys xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId="" pageSeqNum="1" pageRecStart="1" pageRecEnd="1000" pageTotal="10" />
<! — for e.g. if records are 10,000 & pageTotal="2" for, then psp receives 2 RespListKeys from UPI – >
<Txn id="" note="" refId="" refUrl="" ts="" type="ListKeys/GetToken" />
    <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
  <keyList>
      <key code="NPCI" type="PKI" owner="" ki="yyyymmdd">
                <keyValue>base64 encoded certificate</keyValue>
        </key>
      <key code="NPCI" type="CLF" owner="" ki="yyyymmdd">
                <keyValue>Token|Encrypted/base64 encoded certificate</keyValue>
        </key>
<key code="700001" type="CLF" owner="" ki="yyyymmdd">
                <keyValue>Token|Encrypted/base64 encoded certificate</keyValue>
        </key>
<key code="700002" type="CLF" owner="" ki="yyyymmdd">
                <keyValue>Token|Encrypted/base64 encoded certificate</keyValue>
        </key>

    </keyList>
</upi:RespListKeys>
```

Note: Page size, pageRecStart, pageRecEnd & pageTotal only applicable for ver 2.0 and above

Tag Num	Message Item	<XML Tag>	Occurrence
23.1	List of Public Keys of Account providers	<KeyList>	1..1
23.2	Details related to Public Keys	<KeyList.Key>	1..1
23.2.1	Account provider code	code	1..1
23.2.2	Owner of the Key	owner	1..1
23.2.3	Type of the Key	type	1..1
23.2.4	Key Index Date	ki	1..1
23.3	Base64 encoded certificate	< KeyList.Key.KeyValue>	1..1
2.1.6	Page record start count	<Head.pageRecStart>	1..1
2.1.7	Page record end count	<Head.pageRecEnd>	1.1
2.1.8	Total no.of.pages	<Head.pageTotal>	1..1
2.1.9	pageSeqNum	<Head.pageSeqNum>	1..1

6.3.4 List Verified Address Entries

NPCI offers a mechanism to protect customers from attempts to spoof well known merchants such as LIC, Indian Railways, e-commerce players, telecom players, bill payment entities, etc.

ReqListVae:Request list of Verified Address Entries

```
<upi:ReqListVae xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId="" pageSize="1000"/> <! — The default page size will be 1000, if psp wants to change they can change the required page value between min="" to max="" – >
        <Txn id="" note="" refId="" refUrl="" ts="" type="ListVae" /></upi:ReqListVae>
```

RespListVae:Response for List of Verified Address Entries

```
<upi:RespListVae xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId="" pageSeqNum="1" pageRecStart="1" pageRecEnd="1000" pageTotal="10" /> <! — for e.g. if records are 10,000 & pagesize="1000", then psp receives 10 RespListVae from UPI – >
        <Txn id="" note="" refId="" refUrl="" ts="" type="ListVae" />
    <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
      <VaeList>
        <Vae name="LIC" addr="lic@hdfc" logo="image" url="">
        <key code="NPCI" type="PKI" ki="yyyymmdd">
        <keyValue>base64 encoded certificate</keyValue>
        </key>
        </Vae>
        <Vae name="IRCTC" addr="irctc@icici" logo="image" url="">
```

```
<key code="NPCI" type="PKI" ki="yyyymmdd">
<keyValue>base64 encoded certificate</keyValue>
</key>
</Vae>

</VaeList>
</upi:RespListVae>
```

Tag Num	Message Item	<XML Tag>	Occurrence
26.1	List of Verified Address Entries	<VaeList>	1..1
26.2	Details Related to list of Verified Address Entries	<VaeList.Vae>	1..1
26.2.1	Name of the Merchant	name	1..1
26.2.2	Payment Address of the Merchant	addr	1..1
26.2.3	Logo of the Merchant	logo	1..n
26.2.4	URL Link provided by Merchant	url	1..n

6.3.5 List Account

PSPs to find the list of accounts linked to the mobile or Adhaar by a particular account provider. If the destination bank name is not known details of account provider will be fetched from central mapper.

As part of ATM PIN introduction, the issuer bank has to respond with new cred block with subtype as ATM PIN, its type and length, where PIN length can be 4 or 6 digits. This info will be used to capture ATM PIN in the common library.

ReqListAccount: Request for Account List

```
<upi:ReqListAccount xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
    <Txn id="" note="" refId="" refUrl="" ts="" type="ListAccount" />
    <Link type="MOBILE|AADHAAR" value=""/>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="" aadhaarConsent=" Y|N">
  <Device>
      <Tag name="MOBILE" value=""/>
      <Tag name="GEOCODE" value=""/>
      <Tag name="LOCATION" value="" />
      <Tag name="IP" value=""/>
      <Tag name="TYPE" value=""/>
      <Tag name="ID" value=""/>
```

```
        <Tag name="OS" value=""/>
        <Tag name="APP" value=""/>
        <Tag name="CAPABILITY" value=""/>
        <Tag name="TELECOM" value="Airtel/Vodafone"/>
        </Device>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
</Ac>
</Payer>
</upi:ReqListAccount>
```

Tag Num	Message Item	<XML Tag>	Occurrence
24.1	Linked account list	<Link>	1..1
24.1.2	Account linkage to Mobile/Aadhaar	type	1..1
24.1.3	Mobile or Aadhaar Number	value	1..1
24.1.4	Aadhaar consent	Boolean value	1..1
24.1.5	TELECOM Operator	Value	1..n

RespListAccount: Response for Account List

```
<upi:RespListAccount xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
      <Txn id="" note="" refId="" refUrl="" ts="" type="ListAccount"/>
        <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
       <AccountList>
            <Account
            accType="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"mbeba="" accRefNumber="" <! Mbeba flag is used to mention the UPI PIN availabilitiy>>
            maskedAccnumber="" ifsc="HDFC0000101" mmid="9056014" name="" aeba="Y/N" aadhaarNo='1234 5678 9012"–'><!—if user consent for Aadhaar (aadhaarConsent) is "Y" and aeba="Y" then bank should send the Aadhaar no. Masked accout.no should be masked with capital letter"X" –!>
            <CredsAllowed type="PIN" subType="ATMPIN" dType="" dLength=""/>
            </Account>
            <Account
            accType="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"mbeba=""accRefNumber=""
            maskedAccnumber="" ifsc="HDFC0000103" mmid="9056114" name="" aeba="Y/N">
            <CredsAllowed type="PIN" subType=" MPIN" dType="" dLength=""/>
            <CredsAllowed type="PIN" subType="ATMPIN" dType="" dLength=""/>
            <CredsAllowed type="OTP" subType="SMS" dType="" dLength=""/>
            </Account>

        </AccountList>
</upi:RespListAccount>
```

Tag Num	Message Item	<XML Tag>	Occurrence
25.1	Account List	<AccountList>	1..1

Tag Num	Message Item	<XML Tag>	Occurrence
25.2	Details Related to Account	<AccountList.Account>	1..n
25.2.1	Masked Account Number	maskedAccNumber	1..1
25.2.2	IFSC code of the Account	ifsc	1..1
25.2.3	MMID linked to Mobile	mmid	1..1
25.2.4	Name of the Account Holder	name	1..1
25.2.5	Aadhaar Enabled Bank Account or not	aeba	1..1
25.2.6	Aadhaar No	12 digits value	0..1
25.2.7	Account reference number provided by Bank	accRefNumber	1..1
25.2.8	Mobile banking enabled bank account or not	mbeba	1..1
25.2.9	Account Type	accType	1..1
25.3.1	Details related to credentials supported for an account	<AccountList.Account.CredsAllowed>	1..1
25.3.2	Creds allowed	<type>	1..1
25.3.3	Creds allowed	<subType>	1..1
25.3.4	CredsAllowed format alphanumeric/numeric	dType	1..1
25.3.5	Allowed length of the credential.	dLength	1..1

6.3.6 Manage Verified Address Entries

NPCI offers a mechanism to protect customers from attempts to spoof well known merchants such as LIC, Indian Railways, e-commerce players, telecom players, bill payment entities, etc. This mechanism is an API, where the PSPs can manage, and access the common collection of verified address entries. NPCI, with the help of PSPs, will define a process to manage these entries.

ReqManageVae:Request Manage for Verified Address Entries

```
<upi:ReqManageVae xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="ManageVae" />
<VaeList>
<Vae op="ADD|UPDATE|REMOVE" seqNum="1" name="LIC" addr="lic@hdfc" logo="image" url="">
<key code="NPCI" type="PKI" ki="yyyymmdd">
<keyValue>base64 encoded certificate</keyValue>
</key>
</Vae>
```

<Vae op="ADD|UPDATE|REMOVE" seqNum="2" name="IRCTC" addr="irctc@icici" logo="image" url="">
<key code="NPCI" type="PKI" ki="yyyymmdd">
<keyValue>base64 encoded certificate</keyValue>
</key>
</Vae>

</VaeList>
</upi:ReqManageVae>

RespManageVae: Response Manage for Verified Address Entries

<upi:RespManageVae xmlns:upi="http://npci.org/upi/schema/">
<Headver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="ManageVae" />
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode="">
<Ref op="" seqNum="1" addr="" result="SUCCESS|FAILURE" respCode=""/>
<Ref op="" seqNum="2" addr="" result="SUCCESS|FAILURE" respCode=""/></Resp>
</upi:RespManageVae>

All the attributes available in this API is same as the above API. Please refer 4.6.4

Tag Num	Message Item	<XML Tag>	Occurrence
27.1	Option to Update or Remove	op	1..1

6.3.7 Validate Address

This API will be used by the PSPs when their customer wants to add a beneficiary within PSP application (for sending & collecting money).

ReqValAdd: Validate Address Request

<upi:ReqValAdd xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="ValAdd"/>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>

```
        <Tag name="LOCATION" value="" />
        <Tag name="IP" value=""/>
        <Tag name="TYPE" value=""/>
        <Tag name="ID" value=""/>
        <Tag name="OS" value=""/>
        <Tag name="APP" value=""/>
        <Tag name="CAPABILITY" value=""/>
        <Tag name="TELECOM" value="Airtel/Vodafone/.."/>
        </Device>

</Payer>
<Payee seqNum="" addr=""/>
</upi:ReqValAdd>
```

RespValAdd: Validate Address Response

```
<upi:RespValAdd xmlns:upi="http://npci.org/upi/schema/">
            <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
          <Txn id="" note="" refId="" refUrl="" ts="" type="ValAdd"/>
        <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode="" maskName="" code="" type="" IFSC=""
        accType="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|
        SOD|UOD"      IIN="" pType="UPIMANDATE"> <! –only when payee psp address is umn@handle,
        pType=UPIMANDATE – >
      <Merchant >
          <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
          merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR" />
          <Name brand="" legal="" franchise=""/>
          <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
<FeatureSupported value="01|02|03|04|05|06|07|08|09" />
<! – In case mandate functionality is supported by the customer VPA, then psp should send RespValAdd
          with feature supported tag 01-MANDATE, otherwise "FeatureSupport" tag itself should not be
          present. 02 to 09 for future purpose – >
</Resp>
</upi:RespValAdd>
```

Note:

Feature supported value tag is only applicable for mandate

Tag Num	Message Item	<XML Tag>	Occurrence
28.1	Mask Name of the Beneficiary	maskName	1..1
28.2	If it is a UPI mandate, this field is mandatory. It will return as UPIMANDATE	pType	0..n
28.3	Feature supported tag	<FeatureSupported>	0..n
28.3.1	Value of the feature supported tag	value	0..n

6.3.8 Set Credentials

This API is required for providing a unified channel for setting and changing UPIPIN across various account providers. This is critical to ensure customers can easily change UPIPIN via their mobile or by going to a biometric terminal at a BC. Currently this API is restricted to NPCI and banks to be used via USSD or bank mobile/BC application.

ReqSetCre: Set credential Request

```
<upi:ReqSetCre xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="SetCre" />
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/..."/>
</Device>

<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail                                                                    name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
<Creds>

<Cred type="PIN" subType=" MPIN">
<Data> base-64 encoded/encrypted authentication data</Data>
</Cred>
</Creds>
<NewCred>
<Cred type="PIN" subType=" MPIN">
<Data> base-64 encoded/encrypted authentication data</Data>
</Cred>
</NewCred>
        </Payer>
</upi:ReqSetCre>
```

Tag Num	Message Item	<XML Tag>	Occurrence
29.1	New credentials for Authentication	<NewCred>	1..1
29.1.1	Type of Credentials used to authenticate the request	type	1..1
29.1.2	Type of financial instrument used for authentication	subType	1..1
29.2	Base64 encoded authentication	<Data>	1..1

RespSetCre: Response for Set Credential

<upi:RespSetCre xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="SetCre"/>
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
</upi:RespSetCre>

6.3.9 Mobile Banking Registration

This API allows the customer to set new UPIPIN for the first time.PSP will send the "FORMAT1"or "FORMAT2"to the remitter banks based on their readiness. Cred block with subtype "ATMPIN" is allowed only for FORMAT2.

ReqRegMob: Request for Mobile registration

<upi:ReqRegMob xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="ReqRegMob"/>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/.."/>
</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>

<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
</Payer>

<RegDetails type="FORMAT1|FORMAT2|ATM_REDIRECT">

<! Either of the below block will come based on card details being captured at APP or CL – >

<! – The below block will be used if the payer psp is getting the card details in the APP itself – >
<Detail name="MOBILE" value=""/>
<Detail name="CARDDIGITS" value=""/> <last 6 digit of card no>
<Detail name="EXPDATE" value=""/># MMYY format

<Creds>
<! – The below block will be used if the payer psp is getting the card details in the CL page – >
<Cred type="CARD" subType="CARDDETAILS">
<Data code="" ki=""> base-64 encoded/encrypted authentication data </Data>
</Cred> <! Consists of MOBILE, CARD DIGITS, EXPDATE!>
<! This cred block is used when the payer psp has upgraded to new CL version, which supports capture of card detail in CL itself– >

<Cred type="OTP" subType="SMS|EMAIL|HOTP|TOTP">
<Datacode="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="PIN" subType=" MPIN">
<Datacode="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="PIN" subType="ATMPIN">
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
</Creds>
</RegDetails>

<! –
The formation of the cred blocks will depend on the remitter bank supported format as given below
FORMAT1 – Cred type = "OTP" (subType="SMS") and "PIN" (subType ="MPIN")
FORMAT2 – Cred type = "OTP" (subType="SMS"), "PIN" (subType ="MPIN") and "PIN" (subType ="ATMPIN")
ATM_REDIRECT – Cred type = "OTP" (subType="SMS"), "PIN" (subType ="MPIN") and "PIN" (subType ="ATMPIN") ATMPIN block will contain the value passed by remitter bank page – >
</upi:ReqRegMob>

RespRegMob: Response for Mobile Registration

<upi:RespRegMob xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="ReqRegMob"/>
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
</upi:RespRegMob>

6.3.10 Check Txn Status

This API allows the PSPs to request for the status of the transaction. The PSPs must request for status only after the specified timeout period.

ReqChkTxn: Request for check Txn Status

<upi:ReqChkTxn xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" refCategory="00|01|02|03|04|05|06|07|08|09" ts="" type="ChkTxn|BackOffice" umn="" orgMsgId="" orgRrn="" orgTxnId="" subType="DEBIT|CREDIT|PAY|COLLECT|REFUND|REVERSAL|MANDATE" orgTxnDate="' 'initiationMode="" purpose="00|01|02|03|04|05|06|07|08|09|10" />
<! — If type=BackOffice, the response will be given from the BackOffice system including the dispute status. If type=BackOffice, "orgTxnId, orgRrn & orgTxnDate" should be of the original txn (not refund txn) – >
</upi:ReqChkTxn>

Note:
1. If UPI sends the ReqChkTxn, "subType=DEBIT|CREDIT" to bank.
2. If bank sends to UPI, then "subType=PAY|COLLECT|REFUND|REVERSAL|MANDATE
3. If subType=Mandate, then umn is mandatory and type=ChkTxn
4. The ReqChkTxn can be initiated only after the transaction settlement cycle is over when type=BackOffice

RespChkTxn: Response for check Txn Status

If type="ChkTxn", then the below RespChktxn will be provided by UPI

<upi:RespChkTxn xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts=""orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" refCategory="00|01|02|03|04|05|06|07|08|09" ts="" type="ChkTxn" orgMsgId="" orgTxnId="" orgTxnDate="'initiationMode="" purpose="00|01|02|03|04|05|06|07|08|09|10" umn="" subType="PAY|COLLECT|DEBIT|CREDIT|REFUND|REVERSAL|MANDATE" />
<Resp reqMsgId=""result="SUCCESS|FAILURE|PARTIAL|DEEMED|PENDING|REVOKED"errCode="">
<Ref type="PAYER" seqNum="" addr="" regName="" settAmount="" orgAmount="" settCurrency="" acNum ="" approvalNum=""IFSC="" code="" accType="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD" respCode="" reversalRespCode=""/>
<Ref type="PAYEE" seqNum="" addr="" settAmount="" orgAmount="" settCurrency=""acNum ="" regName="" IFSC="" code="" accType="

```
        SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|
        SOD|UOD"approvalNum="" respCode="" reversalRespCode=""/>

        </Resp>
</upi:RespChkTxn>
```

If type="BackOffice", then the below RespChktxn will be provided by UPI

```
<upi:RespChkTxn xmlns:upi="http://npci.org/upi/schema/">
  <Head ver="2.0" ts="" orgId="" msgId=""/>
  <Txn id="" note="" refId="" refUrl="" ts="" type="BackOffice" orgMsgId="" orgRrn = "" orgTxnId=""
subType="PAY|COLLECT|REFUND|REVERSAL" orgTxnDate='' initiationMode=""
purpose="00|01|02|03|...." refCategory="00|01|02|03|04|05|06|07|08|09"
/>
        <Resp reqMsgId=""result="SUCCESS|FAILURE|DEEMED|PARTIAL" txnRespCode="">
                <Ref type="PAYER" seqNum="" addr="" settAmount="" settCurrency="" IFSC="" acNum =""
        approvalNum="" code=""/>
                <Ref type="PAYEE" seqNum="" addr="" settAmount="" settCurrency="" IFSC="" acNum =""
        approvalNum="" code="" />
                <Ref type="CREDITADJUSTMENT" seqNum="" settAmount="" settCurrency=""
                disputeRespCode="" adjustmentDate= "" adjustmentFlag= "" adjustmentRaisedBank=""/>
                 <Ref type="ONLINEREFUND" seqNum="" settAmount="" settCurrency=""
                disputeRespCode="" adjustmentDate= "" adjustmentFlag= "" adjustmentRaisedBank=""/>
                 <Ref type="CHARGEBACK" seqNum="" settAmount="" settCurrency=""
                disputeRespCode="" adjustmentDate= "" adjustmentFlag= "" adjustmentRaisedBank=""/>
                 <Ref type="PREARBITRATION" seqNum="" settAmount="" settCurrency=""
                disputeRespCode="" adjustmentDate= "" adjustmentFlag= "" adjustmentRaisedBank=""/>
                 <Ref type="ARBITRATION" seqNum="" settAmount="" settCurrency=""
                disputeRespCode="" adjustmentDate= "" adjustmentFlag= "" adjustmentRaisedBank=""/>
        </Resp>
</upi:RespChkTxn>
```

Tag Num	Message Item	<XML Tag>	Occurrence
30.1	orgTxnDate format as like the ts format:YYYY-MM-DDThh:mm:ss+GMT	OrgTxnDate	1..1
30.2	Transaction response code as per the back office system	txnRespcode	1..1
30.3	Dispute response code as per back office system	disputeRespCode	1..1
30.4	Adjustment date as per back office system	adjustmentDate	1..1
30.5	Adjustment flag as per back office system	adjustmentFlag	1..1
30.6	Adjustment raised bank as per back office system	adjustmentRaisedBank	1..1

6.3.11 OTP-Request

This API allows the PSPs to request for an OTP for a particular customer

ReqOtp: Request for OTP

```
<upi:ReqOtp xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="Otp" />
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/.."/>

</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail                                                                    name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
</Payer>

<! In case remitter bank is supporting ATM_REDIRECT, the payer psp should populate the card details in the ReqOtp API itself. With these card details, the remitter bank should form the ATMPIN redirect URL and provide in the RespOtp API for authentication purpose – >

<! Either of the below RegDetails will come based on card details being captured at APP or CL if the remitter bank is in ATM_REDIRECT – >

<! – The below block will be used if the payer psp is getting the card details in the APP itself – >
<RegDetails type="ATM_REDIRECT">
    <Detail name="MOBILE" value=""/>
    <Detail name="CARDDIGITS" value=""/> <!—Last 6 digit of card no – >
    <Detail name="EXPDATE" value=""/>
```

```
</RegDetails>

<! – The below block will be used if the payer psp is getting the card details in the CL page – >
<RegDetails type="ATM_REDIRECT">
    <Creds>
        <Cred type="CARD" subType="CARDDETAILS">
        <Data code="" ki=""> base-64 encoded/encrypted authentication data </Data>
        </Cred>    <! Consists of MOBILE, CARD DIGITS, EXPDATE!>
    </Creds>
<! This cred block is used when the payer psp has upgraded to new CL version, which supports capture of
card detail in CL itself– >
</RegDetails>

</upi:ReqOtp>
```

Note:

2. **If card details are captured in psp page, then RegDetails should** contains a plain text
3. **If card details are captured in CL page, then RegDetails should** contains a **cred block**

RespOtp: Response for OTP

```
<upi:RespOtp xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="Otp" />
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode="" securePinUrl=""/>

    < !–The sample securePinUrl will be like below
    https://www.mybank.com/verify/atmpin?sessionParams=xyzabc",
    (The parameters will not be in url, for illustrative purpose been specified like
    above ) – >

</upi:RespOtp>
```

Tag Num	Message Item	<XML Tag>	Occurrence
31.1	url redirecting to the Issuer PSP page from CL	securePinUrl	1..1

6.3.12 Balance-Enquiry

This API Allows PSP to enquirye balanceof a user.

ReqBalEnq: Request for Balance Enquiry

```
<upi:ReqBalEnq xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
```

<Txn id="" note="" refId="" refUrl="" ts="" type="BalEnq|BalChk"><! – If type has balance check then amount tag will be mandatory – >
<RiskScores>
<Score provider="sp" type="TXNRISK" value=""/>
<Score provider="NPCI" type="TXNRISK" value=""/>
</RiskScores>
</Txn>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/.."/>

</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE"
value="SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
<Ac addrType ="CARD">
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="CARDNUM" value=""/>
</Ac>
<Creds>
< Cred type="AADHAAR" subType="AADHAAR-BIO-FP|AADHAAR-BIO-IRIS|AADHAAR-BIO-OTP">
<Meta lk="" ac="" sa="" uid="" ver=""/>
<Data> base-64 encoded/encrypted authentication data</Data></Cred>
<Cred type="OTP" subType="SMS|EMAIL|HOTP|TOTP">
<Data> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="PIN" subType=" MPIN">
<Data> base-64 encoded/encrypted authentication data</Data></Cred>
<Cred type="CARD" subType="CVV1|CVV2|EMV">

<Data> base-64 encoded/encrypted authentication data</Data>
</Cred>
</Creds>
<Amount value="" curr="INR"> <! – Against the amount is specified the issuer has to verify and confirm in the response by Y|N – >

<Split name="PURCHASE|CASHBACK" value=""/> <! – Split Name is used for future purpose which is used for multiple payer concepts – >
</Amount>
</Payer>
</upi:ReqBalEnq>

RespBalEnq: Response for Balance Enquiry

<upi:RespBalEnq xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" refId="" refUrl="" ts="" type="BalEnq|BalChk">
<RiskScores>
<Score provider="sp" type="TXNRISK" value=""/>
<Score provider="NPCI" type="TXNRISK" value=""/>
</RiskScores>
</Txn>
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Bal>
<Data>base-64 encoded/encrypted data</Data>
<! – if type="BalEnq" – >

<Data> Y|N </Data><!—if type=BalChk then the RespBalEnq should send result=SUCCESS then the value should be "Y" and if the result=FAILURE then the data value should be "N" – > <! – if type="BalChk" – >

</Bal>
</Payer>
</upi:RespBalEnq>

Note: For balance enquiry format, refer annexure document

Tag Num	Message Item	<XML Tag>	Occurrence
32.1	Data For Balance enquiry	<Bal>	1..1
32.2	Base 64 encoded authentication	<Bal.Data>	1..1

6.3.13 HeartBeat Messages

This API is a mechanism for UPI system monitoring (monitoring connection with PSPs and sending EOD to PSPs).

ReqHbt: Request for HeartBeat Request

```
<upi:ReqHbt xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
                    <Txn id="" note="" refId="" refUrl="" ts="" type="Hbt" />
<HbtMsg type="EOD|ALIVE" value="DATE|NA"/></upi:ReqHbt>
```

RespHbt: Response for HeartBeat Request

```
<upi:RespHbt xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts=""orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="Hbt" />
        <Resp reqMsgId=""result="SUCCESS|FAILURE" errCode=""/>
</upi:RespHbt>
```

Tag Num	Message Item	<XML Tag>	Occurrence
33.1	Defines heartbeat messages	<HbtMsg>	1..1
33.1.1	Defines message type	< HbtMsg.type>	1..1
33.1.2	Details related to type	< HbtMsg.value>	1..1

6.3.14 Request Pending Messages

This API allows PSP to request pending messages against a given mobile number or Aadhaar number.

ReqPendingMsg: Request for pending messages

```
<upi:ReqPendingMsg xmlns:upi="http://npci.org/upi/schema/">
          <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="PendingMsg" />
          <ReqMsg type="MOBILE|AADHAAR" value="" addr="" />
</upi: ReqPendingMsg>
```

Tag Num	Message Item	<XML Tag>	Occurrence
33.1	Defines Request Pending messages	<ReqMsg>	1..1

34.1.1	Defines message type	type	1..1
34.1.2	Details PSP address	addr	1..1
34.1.3	Details of PSP value	Value	1..1

RespPendingMsg: Response for pending messages

```
<upi:RespPendingMsg xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1. ts="" orgId="" msgId=" "/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="PendingMsg" />
        <Resp reqMsgId=" " result="SUCCESS/FAILURE" errCode=""/>
        <RespMsg>
                <PenTxn id=" " note="" refId="" refUrl="" ts="" type="COLLECT" orgTxnId=""/>
                <PenTxn id=" " note="" refId="" refUrl="" ts="" type="COLLECT" orgTxnId=""/>
        </RespMsg>
</upi:RespPendingMsg>
```

6.3.15 Transaction Confirmation

This API will be used to inform the status of the transaction to PSP's.

ReqTxnConfirmation: Request

```
<upi:ReqTxnConfirmation xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" note="" refId="" refUrl="" ts="" type="TxnConfirmation" orgTxnId="" initiationMode=""
custRef="" purpose="00|01|02|03|04|05|06|07|08|09|10"
refCategory="00|01|02|03|04|05|06|07|08|09"/>
<TxnConfirmation note="" orgStatus="SUCCESS/FAILURE/PENDING"orgErrCode="" type=""actn="">
        <Ref type="PAYER|PAYEE" seqNum="" addr="" regName=""settAmount="" orgAmount=""
        settCurrency="" approvalNum=""acNum="" IFSC="" code="" accType="
        SAVINGS|CURRENT|DEFAULT|NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"
        respCode="" reversalRespCode=""/>
</TxnConfirmation>
</upi:ReqTxnConfirmation>
```

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
1.1	API Name	<upi>	1..1			Y	
1.1.1	API Schema namespace	xmlns	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
2.1	Header for the message	<Head>	1..1	Alphabetic	Fixed value	Y	
2.1.1	Version of the API	ver	1..1	Numeric	Min Length: 1 Max Length : 6s	Y	019_Head_Version
2.1.2	Time of request from the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
2.1.3	Organization id that created the message	orgId	1..1	Numeric	Min Length: 1 Max Length : 20	Y	
2.1.4	Message identifier- used to correlate between request and response	msgId	1..1	Alphanumeric	Length= 35	Y	021_Head_MsgId
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1	Alphabetic	Fixed value	Y	
4.1.1	Unique Identifier of the transaction across all entities, created by the originator	id	1..1	Alphanumeric	Length= 35	Y	022_Txn_UUID
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1	Alphanumeric	Min Length: 1 Max Length : 50	Y	
4.1.3	Consumer reference number to	refId	1..1	Alphanumeric	Min Length: 1	Y	

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
	identify (like Loan number, etc.)				Max Length : 35		
4.1.4	URL for the transaction	refUrl	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.5	Transaction origination time by the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1	Alphanumeric	Length= 35	Y	023_Txn_ orgTxnId
4.1.8	Customer reference number for the initiated transaction	custRef	1..1	Numeric	Length= 12	Y	
4.1.10	Initiation mode	Initiation mode	1..1	Code	Min Length: 1 Max Length: 3	Y	031_Txn_Initiation mode
4.1.13	Transaction Type	Type	1..1	Code	Min Length: 1 Max Length : 20	Y	The value of the tag should be "TxnConfirmation" always.
13.1	Transaction Confirmation	<TxnConfirmation>	1..1	Alphabetic	Fixed value	Y	
13.1.1	Description of the transaction(which will be printed on Pass book)	note	1..1	Alphanumeric	Min Length: 1 Max Length : 50	Y	
13.1.2	Type of the Transaction	type	1..1	Code	Min Length: 1 Max	Y	012_ReqTxn_Pay 013_ReqTxn_Collect

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
					Length : 20		
13.1.3	Orignal transaction error code	orgErrorCode	0..1	Code	Min Length: 1 Max Length : 20	N	
13.1.4	Orignal transaction status	orgStatus	1..1	Code	Min Length: 1 Max Length : 20	Y	
11.1.4	Authentication code	actn	1..n	Numeric	Minlength:1 Max length:40	Y	033_RespPay_ActCode
6.1	Details related to the Payer /Payee	<Payer> /<Payee>	1..1	Alphabetic	Fixed value	Y	
6.1.1	Address of the Payer	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
11.2.10	Reversal Response Code	reversalRespCode	0..n	Code	Min Length: 1 Max Length : 20	N	028_Response_Reversal
11.2.1	Ref type	type	1..1	Code		Y	016_RespPay_Pay 017_RespPay_Collect 018_RespPay_Reversal
11.2.2	Sequence Number	seqNum	1..1	Numeric	Length=4	Y	
11.2.3	Payment address	addr	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
11.2.4	Settlement Amount	settAmount	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value

Tag Num	Message Item	<XML Tag>	Occurrence	Datatype	Length	Mandatory	Rules
11.2.5	Settlement Currency	settCurrency	1..1	Text	Min Length: 1 Max Length : 3	Y	
11.2.6	Approval Reference Number	approvalNum	1..1	Alphanumeric	Length= 6	Y	025_Response_ApprovalNum
11.2.7	Response code	respCode	1..1	Alphanumeric	Min Length: 1 Max Length : 20	Y	
11.2.8	Registered name with bank	regName	1..1	Alphanumeric	Min Length: 1 Max Length : 99	Y	
11.2.9	Original amount	orgAmount	1..1	Numeric	minInclusive: 0 totalDigits: 15	Y	051_ReqPay_Amount_Value
11.2.11	Account number	acNum	1..1	Alphanumeric	Min Length: 1 Max Length : 30	Y	
5.1.5	Merchant Classification Code - MCC	code	1..1	Numeric	Length= 4 digit	Y	024_Txn_code
11.2.12	IFSC code	IFSC	1..n	Alphanumeric	Length :11	Y	032_RespPay_RefTag_IFSC
11.2.13	Account type	accType	1..n	Code	Fixed Value	Y	048_ReqPay_Ac_name_Account

RespTxnConfirmationStatus: Response

```
<upi:RespTxnConfirmation xmlns:upi="http://npci.org/upi/schema/">
        <Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
        <Txn id="" custRef="123456789012"note="" refId="" refUrl="" ts="" type="TxnConfirmation"
refCategory="00|01|02|03|04|05|06|07|08|09" orgTxnId=""initiationMode=""
        purpose="00|01|02|03|04|05|06|07|08|09|10" />
        <Resp reqMsgId="" result="SUCCESS/Failure" errCode=""/>
</upi:RespTxnConfirmation>
```

Tag Num	Message Item	<XML Tag>	Occurrence	DATATYPE	LENGTH	Mandatory	Rules
1.1	API Name	<upi>	1..1			Y	
1.1.1	API Schema namespace	xmlns	1..1	Alphanumeric	Min Length: 1 Max Length : 255	Y	
2.1	Header for the message	<Head>	1..1	Alphabetic	Fixed value	Y	
2.1.1	Version of the API	ver	1..1	Numeric	Min Length: 1 Max Length : 6	Y	019_Head_Version
2.1.2	Time of request from the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
2.1.3	Organization id that created the message	orgId	1..1	Numeric	Min Length: 1 Max Length : 20	Y	
2.1.4	Message identifier-used to correlate between request and response	msgId	1..1	Alphanumeric	Length=35	Y	021_Head_MsgId
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1	Alphabetic	Fixed value	Y	
4.1.1	Unique Identifier of the transaction across all entities, created by the originator	id	1..1	Alphanumeric	Length=35	Y	022_Txn_UUID

Tag Num	Message Item	<XML Tag>	Occurrence	DATATYPE	LENGTH	Mandatory	Rules
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1	Alphanumeric	Min Length: 1 Max Length : 50	Y	
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.4	URL for the transaction	refUrl	1..1	Alphanumeric	Min Length: 1 Max Length : 35	Y	
4.1.5	Transaction origination time by the creator of the message	ts	1..1	ISODateTime	Min Length: 1 Max Length : 255	Y	020_Head_ts
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1	Alphanumeric	Length=35	Y	023_Txn_ orgTxnId
4.1.8	Customer reference number for the initiated transaction	custRef	1..1	Numeric	Length=12	Y	
4.1.13	Transaciton Type	Type	1..1	Code	Min Length: 1 Max Length : 20	Y	TxnConfirmation
4.1.10	Initiation mode	Initiation mode	1..1	Code	Min Length: 1 Max Length: 3	Y	031_Txn_Initiation mode
11.1	Response	<Resp>	1..1	Alphabetic	Fixed value	Y	
11.1.1	Request Message identifier	reqMsgId	1..1	Alphanumeric	Length=35	Y	
11.1.2	Result of the transaction	result	1..1	Code	Min Length: 1 Max Length : 20	Y	

6.4 UPI-Mandate APIs

6.4.1 Request Mandate

<upi:ReqMandate xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Meta>
<Tag name="PAYREQSTART" value=""/>
<Tag name="PAYREQEND" value=""/>
</Meta>
<Txn id="" note="" custRef="" refId="" refUrl="" refCategory="00|01|02|03|04|05|06|07|08|09" ts="" type="CREATE|REVOKE|UPDATE"initiationMode="" initiatedBy="PAYER|PAYEE" purpose="00|01|02|03|04|05|06|07|08|09|10" orgTxnId="" />
<!— "Txn.note" is used to describe the scheme/plan reference number if any – >
<!—"Txn.refId" is used to populate the consumer reference number if any – >
<! – " orgTxnId" is only for REVOKE/UPDATE – >
<Rules>
<Rule name="EXPIREAFTER" value="1 miniute to max 64800 minitues"/>
<!—If EXPIREAFTER is not provided default value will be taken as 30 minutes – >
</Rules>

</Txn>

<Mandate name="" txnId="" umn="" ts="" revokeable="Y|N" shareToPayee="Y|N" type="" blockFund="Y|N">
<!—the field "blockFund"is used for initimating remitter bank to block the necessary fund against customer account– >
<!—the field "name" should describe the purpose of UPI-Mandate – ><!—the field "type" is an optional added for future use- >

<!–umn will be created by customer/Payer PSP & UMN length to be 32digit in UUID Logic(the UMN should be random, non-guessable and active UMN should be unique. E.g format XYZa977ccabb11e7abc4cec278b6b50a@mypsp). The total length of UMN address should be 70digit– >
<!—txnId will be same as Txn element id attribute →
<!– shareToPayee="Y|N" will be N only for single occurance – >
<Validity start="ddMMYYYY" end="ddMMYYYY"/>
<Amount value="" rule="MAX|EXACT"/>
<Recurrence pattern="ONETIME|DAILY|WEEKLY|FORTNIGHTLY| MONTHLY| BIMONTHLY|QUARTERLY|HALFYEARLY|YEARLY|ASPRESENTED">
<Rule value="" type="BEFORE|ON|AFTER"/>
<!–Example : ONETIME, DAILY, AND ASPRESENTED should not have any rules →
<!–Example : if the pattern selected as WEEKLY then the value will be from (1-Monday to 7- Sunday) FORTNIGHTLY(1-15 days), MONTHLY|BIMONTHLY|QUARTERLY|HALFYEARLY|YEARLY| (1-30/31 days)-If 30/31 is not available for any given month then last day of the month will be considered →
</Recurrence>

```
</Mandate>

<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Info>
<Identity id ="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value=""/>
<Tag name="GEOCODE" value=""/>
<Tag name="LOCATION" value="" />
<Tag name="IP" value=""/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value=""/>
<Tag name="OS" value=""/>
<Tag name="APP" value=""/>
<Tag name="CAPABILITY" value=""/>
<Tag name="TELECOM" value="Airtel/Vodafone/.."/>
</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
<Ac addrType ="CARD">
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="CARDNUM" value=""/>
</Ac>
<Creds>
< Cred type="AADHAAR" subType="AADHAAR-BIO-FP|AADHAAR-BIO-IRIS|AADHAAR-BIO-OTP">
        <Meta lk="" ac="" sa="" uid="" ver=""/>
        <Data code="" type="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="OTP" subType="SMS|EMAIL|HOTP|TOTP">
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
<Cred type="PIN" subType=" MPIN">
```

```
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
        <Cred type="PREAPPROVED" subType="NA">
                <Data> base-64 encoded</Data>
                <!– #data includes respCode and approvalRef
                In the format "respCode|approvalNum"
                – >
</Cred>
<Cred type="CARD" subType="CVV1|CVV2|EMV">
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
</Creds>
</Payer>
<Payees>
<Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
    <Merchant >
        <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
        merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR" />
        <Name brand="" legal="" franchise=""/>
        <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
<Info>
<Identity type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Device>
<Tag name="MOBILE" value="+91.99999.99999"/>
<Tag name="GEOCODE" value="12.9667,77.5667"/>
<Tag name="LOCATION" value="Sarjapur Road, Bangalore, KA, IN" />
<Tag name="IP" value="123.456.123.123"/>
<Tag name="TYPE" value=""/>
<Tag name="ID" value="123456789"/>
<Tag name="OS" value="Android 4.4"/>
<Tag name="APP" value="CC 1.0"/>
<Tag name="CAPABILITY" value="011001"/>
<Tag name="TELECOM" value="Airtel/Vodafone/.."/>
</Device>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
</Payee>
</Payees>
</upi:ReqMandate>
```

NOTE:

1. When request mandate is initiated by the customer, then few tags are not required (payee<info>, payee<device> and payee<Ac>)
2. When request mandate is initiated by the corporate PSP, then few tags are not required (payer<info>, payer<device>, payer <cred>, payer<Ac>)
3. When ReqMandate is triggered from UPI to remitter bank, then few tags are not required (payee<device>)
4. Field recurrence pattern if"QUARTERLY", we should consider 3 months completely.
   Eg:-
   - January-March (starts with January 1st with '1' ends with last numbered date of March as '90' or '91' (leap year))
   - April-June (starts with April 1st with '1' ends with last numbered date of June as '91')
   - July-September (starts with July 1st with '1' ends with last numbered date of September as '92')
5. Block Fund: Mandate functionality introduces a new feature called blocking of funds which is relevant for scenarios like IPO and mutual funds. When an authorized create mandate request comes to remitter bank where 'blockfund' tag is set, then remitter bank needs to block the specified amount in customers account. This functionality will only be allowed for one-time mandates.
   Now when the mandate executes, one of the below cases can occur:
   a. Payee initiates mandate collect, debiting the entire amount which was blocked by remitter bank.
   b. Payee initiates mandate collect, debiting partial amount from the blocked funds. When partial funds are debited, remitter bank needs to unblock the remaining amount.
   c. If payee wants to release the funds before the expiry of mandate by calling revoke mandate on that umn.
   d. If mandate is not executed by payee then with expiry of the mandate, remitter bank should release the funds.
6. Revoke Mandate: It can be initiated by both the parties, Now when the revoke occurs any one of the below cases can occur,
   a. Payer psp initiates revoke mandate will go to the customer for authorization with his/her UPI pin. So the payer psp should form the UPI pin cred block in the ReqMandate api.
   b. Payee psp initiates revoke mandate should not go to the customer for authorization. So the payer psp should take the mandate signed cred block and form the mandate cred in the RespAuthMandate api.

Tag Num	Message Item	<XML Tag>	Occurrence

33.1	Defines to create mandate	<Mandate>	1..1
33.1.1	Defines unified mandate number	umn	1..1
33.1.2	Details time stamp	Ts	1..1
33.1.3	Transaction Id	txnId	1.1
33.1.4	Share to Payee flag	shareToPayee	0.n
33.1.5	Mandate Initiation Channel	type	1.1
33.1.6	Defines mandate name	name	1..1
33.1.7	Defines mandate revokeabililty	revokeable	1..1
33.2	Defines mandate validity	<Mandate.Validity>	1.1
33.2.1	Defines start time of validity	start	1.1
33.2.2	Defines end time of validity	end	1.1
33.3	Defines mandate amount	<Mandate.Amount>	1.1
33.3.1	Defines amount value	value	1.1
33.3.2	Defines amount rule	rule	1.1
33.4	Defines mandate recurrence	<Mandate.Recurrence>	1.1
33.4.1	Defines recurrence pattern	Pattern	1.1
33.5	Defines mandate rule	<Mandate.Rule>	1.1
33.5.1	Defines rule value	value	1.1
33.5.2	Defines rule type	type	1.1

6.4.2 Response Mandate

<upi:RespMandate xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>

<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode="" actn="">
<Ref type="PAYER|PAYEE" regName="" addr="" approvalNum="" acNum="" IFSC="" code="" respCode="" />
</Resp>

<Txn id="" note="" custRef="" refId="" refUrl="" ts="" refCategory="00|01|02|03|04|05|06|07|08|09" type="CREATE|REVOKE|UPDATE" initiationMode="" initiatedBy="PAYER|PAYEE" purpose="00|01|02|03|04|05|06|07|08|09|10" orgTxnId="" />
<Mandate name="" txnId="" umn="" ts="" revokeable="Y|N" shareToPayee="Y|N" type="" blockFund="Y|N"><Validity start="ddMMYYYY" end="ddMMYYYY"/>
<Amount value="" rule="MAX|EXACT"/>

<Recurrence
pattern="ONETIME|DAILY|WEEKLY|BIMONTHLY|MONTHLY|QUARTERLY|HALFYEARLY|YEARLY|ASPRESENTED|FORTNIGHTLY">
<Rule value="" type="BEFORE|ON|AFTER"/>
</Recurrence>
</Mandate>
<Signature id="MANDATE">
<!–Digital Signature of the issuer -!>
</Signature>

</upi:RespMandate>

NOTE

1. The same response mandate which is received from the remitter Bank is forwarded to the customer in case of customer created mandate.
2. If the corporate initiates the mandate request, UPI compresses the final response which will not contain the digital signed XML block.

Tag Num	Message Item	<XML Tag>	Occurrence
34.1	Defines digital signed xml	<signature>	1..1

6.4.3 ReqAuthMandate

<upi:ReqAuthMandate xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" custRef="" refId="" refUrl="" ts="" type="CREATE|REVOKE|UPDATE"initiationMode=""
initiatedBy="PAYER|PAYEE" purpose="00|01|02|03|04|05|06|07|08|09|10"
refCategory="00|01|02|03|04|05|06|07|08|09" orgTxnId="" />

<Mandate name="" txnId="" umn="" ts="" revokeable="Y|N" shareToPayee="Y|N" type=""
blockFund="Y|N">
<Validity start="ddMMYYYY" end="ddMMYYYY"/>
<Amount value="" rule="MAX|EXACT"/>
<Recurrence
pattern="ONETIME|DAILY|WEEKLY|BIMONTHLY|MONTHLY|QUARTERLY|HALFYEARLY|YEARLY|ASPRESENTED|FORTNIGHTLY">
<Rule value="" type="BEFORE|ON|AFTER"/>
</Recurrence>
</Mandate>

<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Info>

<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
<Ac addrType ="CARD">
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="CARDNUM" value=""/>
</Ac>
</Payer>
<Payees>
<Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Merchant >
<Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE" merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR" />
<Name brand="" legal="" franchise=""/>
<Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
</Payee>
</Payees>
</upi:ReqAuthMandate>

6.4.4 RespAuthMandate

<upi:RespAuthMandate xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" custRef="" refId="" refUrl="" ts="" type="CREATE|REVOKE|UPDATE"initiationMode="" initiatedBy="PAYER|PAYEE" purpose="00|01|02|03|04|05|06|07|08|09|10" refCategory="00|01|02|03|04|05|06|07|08|09" orgTxnId="" />
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>

<Mandate name="" txnId="" umn="" ts="" revokeable="Y|N" shareToPayee="Y|N" type="" blockFund="Y|N">>

<Validity start="ddMMYYYY" end="ddMMYYYY"/>
<Amount value="" rule="MAX|EXACT"/>
<Recurrence
pattern="ONETIME|DAILY|WEEKLY|BIMONTHLY|MONTHLY|QUARTERLY|HALFYEARLY|YEARLY|ASPRESENTED|FORTNIGHTLY">
<Rule value="" type="BEFORE|ON|AFTER"/>
</Recurrence>
</Mandate>

<Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
<Ac addrType ="CARD">
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="CARDNUM" value=""/>
</Ac>
<Creds>
< Cred type="AADHAAR" subType="AADHAAR-BIO-FP|AADHAAR-BIO-IRIS|AADHAAR-BIO-OTP">
<Meta lk="" ac="" sa="" uid="" ver=""/>
<Data code="" type="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="OTP" subType="SMS|EMAIL|HOTP|TOTP">
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
<Cred type="PIN" subType=" MPIN">
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>
<Cred type="CARD" subType="CVV1|CVV2|EMV">
<Data code="" ki=""> base-64 encoded/encrypted authentication data</Data>
</Cred>

```
        <Cred type="PREAPPROVED" subType="NA">
                <Data> base-64 encoded</Data>
                <!– #data includes respCode and approvalRef
                In the format "respCode|approvalNum"
                – >
</Cred>
</Creds>
</Payer>
<Payees>
<Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
    <Merchant >
        <Identifier subCode="" mid ="""" sid ="" tid="" merchantType="SMALL|LARGE"
        merchantGenre="OFFLINE|ONLINE" onBoardingType="BANK|AGGREGATOR" />
        <Name brand="" legal="" franchise=""/>
        <Ownership type="PROPRIETARY|PARTNERSHIP|PRIVATE|PUBLIC|OTHERS"/>
</Merchant>
<Info>
<Identity id="" type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
<Rating VerifiedAddress="TRUE|FALSE"/>
</Info>
<Ac addrType ="AADHAAR">
<Detail name="IIN" value=""/>
<Detail name="UIDNUM" value=""/>
</Ac>
<Ac addrType="ACCOUNT">
<Detail name="IFSC" value=""/>
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT
NRE|NRO|CREDIT|PPIWALLET|BANKWALLET|SOD|UOD"/>
<Detail name="ACNUM" value=""/>
</Ac>
<Ac addrType ="MOBILE">
<Detail name="MMID" value=""/>
<Detail name="MOBNUM" value=""/>
</Ac>
<Ac addrType ="CARD">
<Detail name="ACTYPE" value="SAVINGS|CURRENT|DEFAULT"/>
<Detail name="CARDNUM" value=""/>
</Ac>
</Payee>
</Payees>
</upi:RespAuthMandate>
```

6.4.5 ReqMandateConfirmation

<upi:ReqMandateConfirmation xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" custRef="" refId="" refUrl="" ts="" type="CREATE|REVOKE|UPDATE"initiationMode="" initiatedBy="PAYER|PAYEE" purpose="00|01|02|03|04|05|06|07|08|09|10" refCategory="00|01|02|03|04|05|06|07|08|09" orgTxnId="" />

<TxnConfirmation note="" orgStatus="SUCCESS/FAILURE/PENDING"orgErrCode="" type="" actn="">
<Ref type="PAYER|PAYEE" regName="" addr="" approvalNum="" acNum="" IFSC="" code="" respCode="" />
</TxnConfirmation>
<Mandate name="" txnId="" umn="" ts="" revokeable="Y|N" shareToPayee="Y|N" type="" blockFund="Y|N">
<Validity start="ddMMYYYY" end="ddMMYYYY"/>
<Amount value="" rule="MAX|EXACT"/>
<Recurrence pattern="ONETIME|DAILY|WEEKLY|BIMONTHLY|MONTHLY|QUARTERLY|HALFYEARLY|YEARLY|ASPRESENTED">
<Rule value="" type="BEFORE|ON|AFTER"/>
</Recurrence>
</Mandate>
<Signature id="MANDATE">
<!–Digital Signature of the issuer -!>
</Signature>
</upi:ReqMandateConfirmation>

NOTE:

1. If the ReqMandateConfirmation is send from UPI to Payee PSP, UPI haves to compress the final confirmation response which will not contain the digital signed XMIL block.

6.4.6 RespMandateConfirmation

<upi:RespMandateConfirmation xmlns:upi="http://npci.org/upi/schema/">
<Head ver="1.0|2.0" ts="" orgId="" msgId=""/>
<Txn id="" note="" custRef="" refId="" refUrl="" ts="" type="CREATE|REVOKE|UPDATE"initiationMode="" initiatedBy="PAYER|PAYEE" purpose="00|01|02|03|04|05|06|07|08|09|10" refCategory="00|01|02|03|04|05|06|07|08|09" orgTxnId="" />
<Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
</upi:RespMandateConfirmation>

7. Annotated Examples

Recollect example scenarios of usage of the proposed APIs in the earlier chapter. This section provides sample filled XMLs for the most common two scenarios.

7.1 Scenario 1 – Direct Pay

Ram wants to send money to his wife Laxmi. Ram has a mobile enabled account with AXIS, and Laxmi has an Aadhaar enabled bank account with Bank of India. He uses an application on his mobile phone to initiate a transaction. He selects his wife as the recipient, and enters his UPIPIN to authenticate himself, and approve the transaction.

AXIS, his PSP, sends the following message to NPCI.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:ReqPay xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="AXIc2ed455b797e4add8392110cfc528acc" orgId="400000"
ts="2018-02-17T13:39:54.939+05:30" ver="2.0"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="PAY"/>
<Payer addr="ram@axis" code="0000" name="ram" seqNum="1" type="PERSON">
<Info>
<Identity id="058010100083492" type="ACCOUNT" verifiedName="Ram"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Device>
<Tag name="MOBILE" value="918143308193"/>
<Tag name="GEOCODE" value="72.9918372,19.1737834"/>
<Tag name="ID" value="911489204188596"/>
<Tag name="OS" value="Android5.1"/>
<Tag name="IP" value="10.193.72.15"/>
<Tag name="APP" value="com.upi.axispay"/>
<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100000000"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac>
<Creds>
```

<Cred subType="MPIN" type="PIN">
<Datacode="NPCI"
ki="20150822">2.0|Nb4B9+IzNMdHBrQREtpvH/5EWjmU0UCc0G7tXhmcevpZT7sQZj51pGXeukKKLVjnI
Q+f/rtmmAqZgze7Q033VRvXBIBJ0aNoBRjQZbDegyLerUMUmTms4izb66Em5kdO4adHiOxr53t7ija1ygi/
meEWCRFWBoxQ8WTofC8Wcn+vB/fKcBI7g7kMY1hISHupKuvt34UNydfghjH4F0yUSg1zgcKPdCZ19KnK
At3uUG21dc1ojUUDcpzGazYB4bS7aXd4pzz0Nt0zltBGftJrbIG5DoB9h05Hw6K1voyjAanwweiSnJkzJR4W
4LBfBH3NINGzkO0oyMZOA/jkQKw==</Data>
</Cred>
</Creds>
<Amount curr="INR" value="2.00"/>
</Payer>
<Payees>
<Payee addr="laxmi@boi" seqNum="1" type="PERSON">
<Amount curr="INR" value="2.00"/>
</Payee>
</Payees>
</ns2:ReqPay>

NPCI notices that the payee account details are not available, and sends a translation request to the payee's service provider (Laxmi's PSP is BOI in this example).

<? xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqAuthDetails xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:44+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfscXLm20"/>
<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157" refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="PAY" custRef="804813039157" initiationMode="00">
<RiskScores/>
</Txn>
<Payees>
<Payee addr="raja25@boi" seqNum="1" type="PERSON">
<Amount value="2.00" curr="INR"/>
</Payee>
</Payees>
<Payer addr="ram@axis" name="RAM" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity id="2345678765" type="ACCOUNT" verifiedName="RAM" id="058010100083492"/>
<Rating verifiedAddress="TRUE">
</Rating>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083000"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac>
<Amount value="2.00" curr="INR"/>

```
</Payer>
</ns2:ReqAuthDetails>
```

The service provider translates the payee address, and sends it back to NPCI. In this case, Laxmi has an Aadhaar enabled bank account, which is identified by her Aadhaar number.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:RespAuthDetails xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="BOIa4097f0d7c684ca4a6e2eddc965968b1" orgId="410005"
ts="2018-02-17T13:39:56.040+05:30" ver="2.0"/>
<Resp reqMsgId="1GRDpegBbA5wfscXLm20" result="SUCCESS"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="PAY">
<RiskScores/>
</Txn>
<Payer addr="ram@axis" code="0000" name="ram" seqNum="1" type="PERSON">
<Info>
<Identity id="058010100083492" type="ACCOUNT" verifiedName="Ram"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083000"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac>
<Amount curr="INR" value="2.00"/>
</Payer>
<Payees>
<Payee addr="laxmi@boi" code="0000" name="Laxmi"
seqNum="1" type="PERSON">
<Info>
<Identity id="910010050136217" type="ACCOUNT" verifiedName="Laxmi "/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="910010050136000"/>
<Detail name="IFSC" value="BKID0000004"/>
</Ac>
<Amount curr="INR" value="2.00"/>
</Payee>
</Payees>
</ns2:RespAuthDetails>
```

NPCI can send the ReqPay_Debit to Remitter bank to debit the issuer account

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqPay xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:45+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfsd3Xhmt"/>
<Meta/>
<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157" refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="DEBIT" custRef="804813039157" initiationMode="00" subType="PAY">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Payer addr="ram@axis" name="Ram" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Ram" id="058010100083492"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Device>
<Tag name="MOBILE" value="918143308193"/>
<Tag name="GEOCODE" value="72.9918372,19.1737834"/>
<Tag name="ID" value="911489204188596"/>
<Tag name="OS" value="Android5.1"/>
<Tag name="IP" value="10.193.72.15"/>
<Tag name="APP" value="com.upi.axispay"/>
<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device><Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083000"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac>
<Creds>
<Cred type="PIN" subType="MPIN"><Data code="400005" ki="20180110">KuypAXXecOqCusBmukRRt3j2o00QZwCB5UcS6Gdtoz/rgVFDanGsyVKyqk+WWARhNuoNR2gnJJkFEoWGt7f6T/toUJ1dUmr26PAAHo5XIfdIY6TXbGQVi6JhmUyk4I8J1FI9779RbqXmpUavBHtyuirkSTAhaaf73l/fPVco7PzSpSDZoa0GXcILJJhVQpi5uh0I5QeLYHMPH+etTSQEguOxY/EhadzD0o+I2DWN7PX99NOZVQ9GEDpTShMnX77CsCFOmUfoPV8Rupy6A31Ywax+3h2/TvRKCVaUkQ7YkQ7NQo5mbvmQjZofd7KY59BHIeEHMYQQg5SLg7XcBImbuQ==</Data>
</Cred>
</Creds>
<Amount value="2.00" curr="INR"/>
</Payer>
<Payees>
<Payee addr="laxmi@boi" name="Laxmi" seqNum="1" type="PERSON" code="0000">

<Info>
<Identity type="ACCOUNT" verifiedName="Laxmi" id="910010050136217"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="910010050136217"/>
<Detail name="IFSC" value="BKID0000004"/>
</Ac>
<Amount value="2.00" curr="INR"/>
</Payee>
</Payees>
</ns2:ReqPay>

Then the remitter remits the customer account and sends successful RespPay_Debit to UPI

<ns2:RespPay xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="AXIfc2b109349844cd8a16355ac52440e39" orgId="400000"
ts="2018-02-17T13:39:58.262+05:30" ver="2.0"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" subType="PAY"
ts="2018-02-17T13:39:54.944+05:30" type="DEBIT">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Resp reqMsgId="1GRDpegBbA5wfsd3Xhmt" result="SUCCESS">
<Ref addr="ram@axis" approvalNum="169353" respCode="00"
seqNum="1" settAmount="2.00" settCurrency="INR" type="PAYER" regName="Ram"/>
</Resp>
</ns2:RespPay>

Now UPI sends ReqPay_credit to beneficiary bank to credit the Payee's account

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqPay xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:47+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfsdhGcUe"/>
<Meta/>
<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="CREDIT" custRef="804813039157"
initiationMode="00" subType="PAY">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>

```
</RiskScores>
</Txn>
<Payer addr="ram@axis" name="Ram" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Ram" id="058010100083492"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Device>
<Tag name="MOBILE" value="918143308193"/>
<Tag name="GEOCODE" value="72.9918372,19.1737834"/>
<Tag name="ID" value="911489204188596"/>
<Tag name="OS" value="Android5.1"/>
<Tag name="IP" value="10.193.72.15"/>
<Tag name="APP" value="com.upi.axispay"/>
<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083492"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac><Amount value="2.00" curr="INR"/>
</Payer>
<Payees>
<Payee addr="laxmi@boi" name="Laxmi" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Laxmi" id="910010050136217"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="910010050136000"/>
<Detail name="IFSC" value="BKID0000004"/>
</Ac>
<Amount value="2.00" curr="INR"/>
</Payee>
</Payees>
</ns2:ReqPay>
```

Benficary bank credits the customer account and sends RespPAy_Crediut with success response to UPI

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:RespPay xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="BOId17432ea58ab42b8b8bd52e9f4f19013" orgId="410005"
```

ts="2018-02-17T13:39:59.126+05:30" ver="2.0"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" subType="PAY"
ts="2018-02-17T13:39:54.944+05:30" type="CREDIT">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Resp reqMsgId="1GRDpegBbA5wfsdhGcUe" result="SUCCESS">
<Ref addr="laxmi@boi" approvalNum="959826" respCode="00"
seqNum="1" settAmount="2.00" settCurrency="INR" type="PAYEE" regName="Laxmi"/>
</Resp>
</ns2:RespPay>

This is the confirmation sent to AXIS.

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RespPay xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:48+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfsdnypyf"/>
<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="PAY" custRef="804813039157"
initiationMode="00">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Resp reqMsgId="AXIc2ed455b797e4add8392110cfc528acc" result="SUCCESS" actn="">
<Ref type="PAYER" seqNum="1" addr="ram@axis" settAmount="2.00" settCurrency="INR"
approvalNum="169353" respCode="00" regName="Ram" orgAmount="2.00" acNum="058010100083000"
IFSC="AXIS0000058" code="0000"/>
<Ref type="PAYEE" seqNum="1" addr="laxmi@boi" settAmount="2.00" settCurrency="INR"
approvalNum="959826" respCode="00" regName="Laxmi" orgAmount="2.00"/>
</Resp>
</ns2:RespPay>

NPCI sends the final confirmation to Payee psp

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqTxnConfirmation xmlns:ns2="http://npci.org/upi/schema/"
xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:49+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfsdq7R5v"/>
<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="TxnConfirmation"
orgTxnId="AXIb1fbc9cea1f34049904e083034723d49" custRef="804813039157" initiationMode="00"/>
<TxnConfirmation note="testpay" orgStatus="SUCCESS" type="PAY" actn="">

<Ref type="PAYEE" seqNum="1" addr="laxmi@boi" settAmount="2.00" settCurrency="INR" approvalNum="959826" respCode="00" regName="Laxmi" orgAmount="2.00"/>
</TxnConfirmation>
</ns2:ReqTxnConfirmation>

Payee psp sends successful confirmation api back to NPCI

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RespTxnConfirmation xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="BOI3bd5a4cc46084ad295cc98b586952ead" orgId="410005"
ts="2018-02-17T13:39:59.903+05:30" ver="2.0"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay"
orgTxnId="AXIb1fbc9cea1f34049904e083034723d49" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="TxnConfirmation"/>
<RespreqMsgId="1GRDpegBbA5wfsdq7R5v" result="SUCCESS"/>
</ns2:RespTxnConfirmation>

< Scenario 2 – Collect Pay>

Two friends Ram and Shyam go out for dinner and Ram pays the bill. They agree to split the bill in half. Ram is going to collect half of the bill from John and will use his android mobile phone to do so and requests Shyam to pay in a week's time. Ram has an account with Axis, and Shyam with BOI. Ram uses his mobile phone, and initiates a request to get money from Shyam.

His service provider (AXIS) , sends the following message to NPCI.

<?xml version="1.0" encoding="UTF-8"?>
<ns2:ReqPay xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="AXI4e11325e968340fba48ebe70cb6be409" orgId="400000"
ts="2018-02-23T14:52:32.422+05:30" ver="2.0"/>
<Txn custRef="805414040578" id="AXIfcd764aeab2a4bd195b25d652c1887f7"
initiationMode="00" note="collect" refId="805414040578"
refUrl="http://axis.com/upi" ts="2018-02-23T14:52:32.427+05:30" type="COLLECT">
<Rules>
<Rule name="EXPIREAFTER" value="1440"/>
<Rule name="MINAMOUNT" value="1.00"/>
</Rules>
</Txn>
<Payer addr="shyam@boi" code="0000" name="shyam" seqNum="1" type="PERSON">
<Amount curr="INR" value="2.00"/>
</Payer>
<Payees>
<Payee addr="ram@axis" code="0000"

name="RAM" seqNum="1" type="PERSON">
<Info>
<Identity id="058010100083492" type="ACCOUNT" verifiedName="Ram"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Device>
<Tag name="MOBILE" value="918143308193"/>
<Tag name="GEOCODE" value="72.991948,19.174975"/>
<Tag name="ID" value="911489204188596"/>
<Tag name="OS" value="Android5.1"/>
<Tag name="IP" value="10.33.237.58"/>
<Tag name="APP" value="com.upi.axispay"/>
<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083000"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac>
<Amount curr="INR" value="2.00"/>
</Payee>
</Payees>
</ns2:ReqPay>

NPCI notices that the payer account details are not available, and sends a translation request to the payer's service provider (BOI).

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqAuthDetails xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-23T15:04:04+05:30" orgId="NPCI" msgId="1GRDpegBbA5wuiCOKaMo"/>
<Txn id="AXIfcd764aeab2a4bd195b25d652c1887f7" note="collect" refId="805414040578" refUrl="http://axis.com/upi" ts="2018-02-23T14:52:32.427+05:30" type="COLLECT" custRef="805414040578" initiationMode="00">
<RiskScores/>
<Rules>
<Rule name="EXPIREAFTER" value="1440"/>
<Rule name="MINAMOUNT" value="1.00"/>
</Rules>
</Txn>
<Payees>
<Payee addr="ram@axis" name="RAM" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Ram" id="058010100083492"/>

<Rating verifiedAddress="TRUE">
</Rating>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083400"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac><Amount value="2.00" curr="INR"/>
</Payee>
</Payees>
<Payer addr="shyam@boi" seqNum="1">
<Amount value="2.00" curr="INR"/>
</Payer>
</ns2:ReqAuthDetails>

The service provider translates the payer address, and sends it back to NPCI. Shyam also authenticates with biometrics.

<?xml version="1.0" encoding="UTF-8"?>
<ns2:RespAuthDetails xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="BOIbc4a5c78392e4d89af82589fb5f8d461" orgId="410005"
ts="2018-02-23T14:53:37.721+05:30" ver="2.0"/>
<RespreqMsgId="1GRDpegBbA5wuiCOKaMo" result="SUCCESS"/>
<Txn custRef="805414040578" id="AXIfcd764aeab2a4bd195b25d652c1887f7"
initiationMode="00" note="collect" refId="805414040578"
refUrl="http://axis.com/upi" ts="2018-02-23T14:52:32.427+05:30" type="COLLECT">
<RiskScores/>
<Rules>
<Rule name="EXPIREAFTER" value="1440"/>
<Rule name="MINAMOUNT" value="1.00"/>
</Rules>
</Txn>
<Payer addr="shyam@axis" code="0000" name="shyam"
seqNum="1" type="PERSON">
<Info>
<Identity id="910010050136217" type="ACCOUNT" verifiedName="Shyam"/>
</Info>
<Device>
<Tag name="MOBILE" value="919701425053"/>
<Tag name="GEOCODE" value="19.0911 ,72.9208"/>
<Tag name="ID" value="356823072981728"/>
<Tag name="OS" value="Android5.1.1"/>
<Tag name="IP" value="10.133.126.45"/>
<Tag name="APP" value="com.upi.axispay"/>

<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="910010050136000"/>
<Detail name="IFSC" value="BKID0000004"/>
</Ac>
<Creds>
<Cred subType="MPIN" type="PIN">
<Data code="NPCI" ki="20150822">2.0|GneSwLbOfn/b+Q6AbmMxAIONVm7FAS9QtbGXXXgzxJPcX2wMWRxOI1GbDu9O9zpafqHV7m5NFViZ1dpF4Ddf8vGiJQKgLxmY0Wc5JDuoCA5dA/CMb8Xfyp5/qwd1Q+PKu5/jASeKR8AWg6bdEH0EptFeIvc73z/Cpo2CjVBR8kqMe/xMNma/jgMcQ0jrYGcK08X9jpGrY+aBBnEWbnuOn/jYixcwjWkaz6Lq3/3MIHKIz/ao4r0sAoIamSrtb3UQOnAZy5/gvrs0Bs3A/vy4v4XFtUXulqfmiuW46NiaCHiF/gD0HiIZC/v2yQk8DfbhqWzaWO4Q/fW23F0IswtwIw==</Data>
</Cred>
</Creds>
<Amount curr="INR" value="2.00"/>
</Payer>
<Payees>
<Payee addr="ram@axis" code="0000"
name="RAM" seqNum="1" type="PERSON">
<Info>
<Identity id="058010100083492" type="ACCOUNT" verifiedName="Ram"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083000"/>
<Detail name="IFSC" value="AXIS0000058"/>
</Ac>
<Amount curr="INR" value="2.00"/>
</Payee>
</Payees>
</ns2:RespAuthDetails>

NPCI sends the ReqPay_Debit to Remitter bank to debit the issuer account

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqPay xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:45+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfsd3Xhmt"/>
<Meta/>

<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157" refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="DEBIT" custRef="804813039157" initiationMode="00" subType="COLLECT">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Payer addr="shyam@boi" name="shyam" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="SHYAM" id="058010100083492"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Device>
<Tag name="MOBILE" value="918143308193"/>
<Tag name="GEOCODE" value="72.9918372,19.1737834"/>
<Tag name="ID" value="911489204188596"/>
<Tag name="OS" value="Android5.1"/>
<Tag name="IP" value="10.193.72.15"/>
<Tag name="APP" value="com.upi.axispay"/>
<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083000"/>
<Detail name="IFSC" value="BKID0000058"/>
</Ac>
<Creds>
<Cred type="PIN" subType="MPIN"><Data code="400005" ki="20180110">KuypAXXecOqCusBmukRRt3j2o00QZwCB5UcS6Gdtoz/rgVFDanGsyVKyqk+WWARhNuo
NR2gnJJkFEoWGt7f6T/toUJ1dUmr26PAAHo5XIfdIY6TXbGQVi6JhmUyk4I8J1FI9779RbqXmpUavBHtyuir
kSTAhaaf73l/fPVco7PzSpSDZoa0GXcILJJhVQpi5uh0I5QeLYHMPH+etTSQEguOxY/EhadzD0o+I2DWN7P
X99NOZVQ9GEDpTShMnX77CsCFOmUfoPV8Rupy6A31YwaxiioohjhTvRKCVaUkQ7YkQ7NQo5mbvmQj
Zofd7KY59BHleEHMYQQg5SLg7XcBImbuQ==</Data>
</Cred>
</Creds>
<Amount value="2.00" curr="INR"/>
</Payer>
<Payees>
<Payee addr="ram@axis" name="RAM" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Ram" id="910010050136217"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>

<Detail name="ACNUM" value="910010050136000"/>
<Detail name="IFSC" value="AXIS0000004"/>
</Ac>
<Amount value="2.00" curr="INR"/>
</Payee>
</Payees>
</ns2:ReqPay>

Then the remitter remits the customer account and sends successful RespPay_Debit to UPI

<ns2:RespPay xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="BOIfc2b109349844cd8a16355ac52440e39" orgId="410005"
ts="2018-02-17T13:39:58.262+05:30" ver="2.0"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" subType="COLLECT"
ts="2018-02-17T13:39:54.944+05:30" type="DEBIT">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Resp reqMsgId="1GRDpegBbA5wfsd3Xhmt" result="SUCCESS">
<Ref addr="shyam@boi" approvalNum="169353" respCode="00"
seqNum="1" settAmount="2.00" settCurrency="INR" type="PAYER" regName="Shyam"/>
</Resp>
</ns2:RespPay>

Now UPI sends ReqPay_credit to beneficiary bank to credit the Payee's account

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqPay xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-17T13:50:47+05:30" orgId="NPCI" msgId="1GRDpegBbA5wfsdhGcUe"/>
<Meta/>
<Txn id="AXIb1fbc9cea1f34049904e083034723d49" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" ts="2018-02-17T13:39:54.944+05:30" type="CREDIT" custRef="804813039157"
initiationMode="00" subType="COLLECT">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Payer addr="shyam@boi" name="Shyam" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Shyam " id="058010100083492"/>
<Rating verifiedAddress="TRUE"/>

```
</Info>
<Device>
<Tag name="MOBILE" value="918143308193"/>
<Tag name="GEOCODE" value="72.9918372,19.1737834"/>
<Tag name="ID" value="911489204188596"/>
<Tag name="OS" value="Android5.1"/>
<Tag name="IP" value="10.193.72.15"/>
<Tag name="APP" value="com.upi.axispay"/>
<Tag name="TYPE" value="MOB"/>
<Tag name="CAPABILITY" value="011001"/>
</Device>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="058010100083492"/>
<Detail name="IFSC" value="BKID0000058"/>
</Ac>
<Amount value="2.00" curr="INR"/>
</Payer>
<Payees>
<Payee addr="ram@axis" name="RAM" seqNum="1" type="PERSON" code="0000">
<Info>
<Identity type="ACCOUNT" verifiedName="Ram" id="910010050136217"/>
<Rating verifiedAddress="TRUE"/>
</Info>
<Ac addrType="ACCOUNT">
<Detail name="ACTYPE" value="SAVINGS"/>
<Detail name="ACNUM" value="910010050136000"/>
<Detail name="IFSC" value="AXIS0000004"/>
</Ac>
<Amount value="2.00" curr="INR"/>
</Payee>
</Payees>
</ns2:ReqPay>
```

Benficary bank credits the customer account and sends RespPAy_Credit with successful response to UPI

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:RespPay xmlns:ns2="http://npci.org/upi/schema/">
<Head msgId="AXId17432ea58ab42b8b8bd52e9f4f19013" orgId="400000"
ts="2018-02-17T13:39:59.126+05:30" ver="2.0"/>
<Txn custRef="804813039157" id="AXIb1fbc9cea1f34049904e083034723d49"
initiationMode="00" note="testpay" refId="804813039157"
refUrl="http://axis.com/upi" subType="COLLECT"
ts="2018-02-17T13:39:54.944+05:30" type="CREDIT">
```

<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Resp reqMsgId="1GRDpegBbA5wfsdhGcUe" result="SUCCESS">
<Ref addr="ram@axis" approvalNum="959826" respCode="00"
seqNum="1" settAmount="2.00" settCurrency="INR" type="PAYEE" regName="Ram"/>
</Resp>
</ns2:RespPay>

NPCI sends the RespPay to Payee psp (AXIS)

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RespPay xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/"><Head ver="2.0" ts="2018-02-23T15:05:12+05:30" orgId="NPCI" msgId="1GRDpegBbA5wuiKfsAxv"/><Txn id="AXIfcd764aeab2a4bd195b25d652c1887f7" note="collect" refId="805414040578" refUrl="http://axis.com/upi" ts="2018-02-23T14:52:32.427+05:30" type="COLLECT" custRef="805414040578" initiationMode="00">
<RiskScores>
<Score provider="NPCI" type="TXNRISK" value="00995"/>
</RiskScores>
</Txn>
<Resp reqMsgId="AXI4e11325e968340fba48ebe70cb6be409" result="SUCCESS">
<Ref type="PAYEE" seqNum="1" addr="ram@axis" settAmount="2.00" settCurrency="INR" approvalNum="770977" respCode="00" regName="ram" orgAmount="2.00" acNum="058010100083000" IFSC="AXIS0000058" code="0000"/>
<Ref type="PAYER" seqNum="1" addr=shyam@boi" settAmount="2.00" settCurrency="INR" approvalNum="645899" respCode="00" regName="shyam" orgAmount="2.00" acNum="910010050136200" IFSC="BKID0000004" code="0000"/>
</Resp>
</ns2:RespPay>

NPCI sends the Final confirmation API to Payer psp

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ReqTxnConfirmation xmlns:ns2="http://npci.org/upi/schema/" xmlns:ns3="http://npci.org/cm/schema/">
<Head ver="2.0" ts="2018-02-23T15:05:12+05:30" orgId="NPCI" msgId="1GRDpegBbA5wuiKfsAxw"/>
<Txn id="AXIfcd764aeab2a4bd195b25d652c1887f7" note="collect" refId="805414040578" refUrl="http://axis.com/upi" ts="2018-02-23T14:52:32.427+05:30" type="TxnConfirmation" orgTxnId="AXIfcd764aeab2a4bd195b25d652c1887f7" custRef="805414040578" initiationMode="00"/>
<TxnConfirmation note="collect" orgStatus="SUCCESS" type="COLLECT" actn="">
<Ref type="PAYER" seqNum="1" addr="shyam@boi" settAmount="2.00" settCurrency="INR" approvalNum="645899" respCode="00" regName="Shyam" orgAmount="2.00" acNum="910010050136200" IFSC="BKID0000004" code="0000"/>
</TxnConfirmation>
</ns2:ReqTxnConfirmation>

Payer psp sends the confirmation response with successful API back to NPCI

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Head msgId="BOI780850b2c617429abd2db1dc10b65dee" orgId="410005"
ts="2018-02-23T14:53:41.111+05:30" ver="2.0"/>
<Txn custRef="805414040578" id="AXIfcd764aeab2a4bd195b25d652c1887f7"
initiationMode="00" note="collect"
orgTxnId="AXIfcd764aeab2a4bd195b25d652c1887f7" refId="805414040578"
refUrl="http://axis.com/upi" ts="2018-02-23T14:52:32.427+05:30" type="TxnConfirmation"/>
<RespreqMsgId="1GRDpegBbA5wuiKfsAxw" result="SUCCESS"/>
</ns2:RespTxnConfirmation>

8. Appendix – Rules

Rule Id	Tag num	Condition	Value	Action
001_ReqPay_Pay	4.1.6	if(type=PAY)	PAY	If it is a PAY txn below tags are mandatory Under Payer 1. Info Tag 2. Device tag 3. Account tag 4. Amount tag. 5. Cred tag Under Payee 1. Payee tag. 2. Amount tag.
002_ReqPay_Collect	4.1.6	if(type=COLLECT)	COLLECT	If it is a COLLECT txn below tags are mandatory. Under payeetag 1. Info Tag 2. Device tag details 3. Account tag 4. Amount tag. Under Payer tag 1. Amount tag.

Rule Id	Tag num	Condition	Value	Action
003_ReqPay_Debit	4.1.6	if(type=DEBIT)	DEBIT	If type is DEBIT below tags are mandatory. Under Payer tag 1. Info Tag 2. Device tag 3. Account tag 4. Amount tag. 5. Cred tag Under Payee tag 1. Info tag 2. Account tag 3. Amount tag 4. Device tag (applicable only in case of COLLECT)
004_ReqPay_Credit	4.1.6	if(type=CREDIT)	CREDIT	If type is CREDIT below tags are mandatory. Under Payer tag 1. Info Tag 2. Device tag 3. Account tag 4. Amount tag Under Payee tag 1. Info tag 2. Device tag (applicable only in case if COLLECT) 3. Account tag 4. Amount tag
005_ReqPay_DebitReversal	4.1.6	if(type=REVERSAL)	REVERSAL	This leg will be sent for reversal from UPI to Remitter If type is REVERSAL below tags are mandatory. Under Payer tag 1. Info Tag 2. Account tag 3. Amount tag
006_ReqPay_CreditReversal	4.1.6	if(type=REVERSAL)	REVERSAL	This leg will be sent for reversal from UPI to Beneficiary If type is REVERSAL below tags are mandatory. Under Payee tag 1. Info Tag 2. Account tag 3. Amount tag

Rule Id	Tag num	Condition	Value	Action
007_ReqPay_PreApproved	5.12	if(type=PAY&&Cred.type="PREAPPROVED") /(type=COLLECT&&Cred.type="PREAPPROVED")	PREAPPROVED	1. If txn type is PAY and PREAPPROVEDthen the following cred block should be present in ReqPay 2. If txn type is COLLECT and PREAPPROVED then the following cred block should be present in RespAuthDetails <Cred type="PREAPPROVED subType="NA"">. Format: respCode|approvalNum Example –00|972345 "|" - to be used as delimiter
008_ReqAuth_Pay	4.1.6	if(type=PAY)	PAY	If type is PAY below tags are mandatory in ReqAuthDetails Under Payer tag 1. Info Tag 2. Account tag 3. Amount tag Under Payee tag 1. Amount tag
009_ReqAuth_Collect	4.1.6	if(type=COLLECT)	COLLECT	If type is COLLECT below tags are mandatory in ReqAuthDetails Under Payer tag 1. Amount tag Under Payee tag 1. Info Tag 2. Account tag 3. Amount tag
010_RespAuth_Pay	4.1.6	if(type=PAY)	PAY	If type is PAY below tags are mandatory in RespAuthDetails Under Payer tag 1. Info Tag 2. Account tag 3. Amount tag Under Payee tag 1. Info Tag 2. Account tag 3. Amount tag

Rule Id	Tag num	Condition	Value	Action
011_RespAuth_Collect	4.1.6	**if(type=COLLECT)**	COLLECT	If type is COLLECT below tags are mandatory in RespAuthDetails Under Payer tag 1. Info Tag 2. Account tag 3. Amount tag 4. Device tag 5. Cred tag Under Payee tag 1. Info Tag 2. Account tag 3. Amount tag
012_ReqTxn_Pay	13.2	**if(type=PAY)**	PAY	Ref tag of payee details will be present in the ReqTxnConfirmation.
013_ReqTxn_Collect	13.2	**if(type= COLLECT)**	COLLECT	Ref tag of payer details will be present in the ReqTxnConfirmation
016_RespPay_Pay	4.1.6, 11.2.1	**if(type=DEBIT)**	DEBIT	Ref tag of payer details will only be sent in the RespPay
017_RespPay_Collect	4.1.6, 11.2.1	**if(type=CREDIT)**	CREDIT	Ref tag of payee details will only be sent in the RespPay

Rule Id	Tag num	Condition	Value	Action
018_RespPay_Reversal	4.1.6, 11.2.1	if(type=REVERSAL)	REVERSAL	Ref tag of Payer details will be sent in debit reversal Ref tag of Payee details will be sent in credit reversal
019_Head_Version	2.1.1	General	Numeric	Default is '1.0'or'2.0'
020_Head_ts	2.1.2, 4.1.5	General	ISO Date time format	The string format should be: YYYY-MM-DDTHH:mm:ss.sssZ , where: YYYY-MM-DD – is the date: year-month-day. The character "T" is used as the delimiter. HH:mm:ss: sss – is the time: hours, minutes, seconds and milliseconds. The 'Z' part denotes the time zone in the format +-hh:mm HH/hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) sss= three digit of milli second (000 through 999) +/- hh:mm = followed by time zone difference from GMT in hours and minutes.This is Mandatory
021_Head_MsgId	2.1.4	General	Alphanumeric	Message ID is unique for particular API leg. It should be always 35 Digits. First 3 digit should be bank Participation code assigned by NPCI followed by 32 digit generated by UUID logic

Rule Id	Tag num	Condition	Value	Action
022_Txn_UUID	4.1.1	General	Alphanumeric	Transaction ID is unique for the any transaction. It should be always 35 Digits. First 3 digit should be bank Participation code assigned by NPCI followed by 32 digit generated by UUID logic
023_Txn_ orgTxnId	4.1.7	if(type=REVERSAL)	Alphanumeric	Mandatory , used only if REVERSAL/Refund happens
024_Txn_code	5.1.5, 6.2.5	General	PERSON=0000 ENTITY=XXXX	"XXXX" is MCC(Merchant Category Code) of the Merchant
025_Response_ApprovalNum	5.12, 11.2.6	if(Result=SUCCESS)	Alphanumeric	6 digits must be Alphanumeric. If result is success, Approval number is mandatory
026_Payer/Payee_InfoRating	5.6.1, 6.5.1	General	Numeric	TRUE|FALSE
027_Response_ErrCode	11.1.3	General	Alphanumeric	only if FAILURE
028_Response_Reversal	11.2.10	if(type=REVERSAL)	Numeric	Mandatory only if FAILURE
029_Payer/Payee_Type	5.1.4, 6.2.4	General	PERSON/ ENTITY	Either PERSON/ENTITY
030_Txn_SubType	4.1.9	If(type=DEBIT/CREDIT/REVERSAL/ REFUND)	DEBIT/ CREDIT/ REVERSAL /REFUND	PAY/COLLECT
031_Txn_Initiation mode	4.1.10	If(type=PAY|COLLECT|DEBIT|CREDIT|REVERSAL|REFUND|ChkTxn|TxnConfirmation) In mandate, if (type=CREATE|UPDATE|REVOKE)	01/02/03/ 04/05/06/ 07//08/09 /10/11/12 /13/14/00	00=Default 01=QR Code 02=Secure QR Code 03=Bharat QR Code 04=Intent 05=Secure Intent 06=NFC(Near Field Communication) 07=BLE (Bluetooth) 08=UHF(Ultra High Frequency) 09=Aadhaar 10=SDK (Software Development Kit) 11=UPI-Mandate 12= FIR (Foreign Inward Remittance) 13= QR Mandate 14= BBPS 15-18 for future purpose

Rule Id	Tag num	Condition	Value	Action
032_RespPay_RefTag_IFSC	11.2.12	if(Response.result=SUCCESS)	IFSC	IFSC code of the respective bank branch and should be 11 digits
033_RespPay_ActCode	11.1.4	If (UIDAIAuth=FAILURE)	Tag value =" XXX|ZZZ" Authentication code	Tag XXX" – Will be populated by NPCI from "err" tag of UIDAI AuthRes ZZZ – Will be populated by NPCI if present in "actn" of UIDAI AuthRes. Else only XXX will be present Please refer the UPI error code document for UIDAI response codes
034_ReqPay_DeviceDetails_Values	5.8.2	if(DEVICE.Tag occurs)	Device Values	MOBILE:91nnnnnnnnn GEOCODE:nn.nnnn,nn.nnnn LOCATION:Area with city, state and Country Code 01-23- Terminal Address 24-36- Terminal City 37-38- Terminal State Code 39-40- Terminal Country Code IP:Valid IP address format(v4,v6) TYPE:Min Length – 1 , Max Length – 20 (Refer Rule_035) ID:Min Length – 1 , Max Length – 35 OS:Min Length – 1 , Max Length – 20 APP:Min Length – 1 , Max Length – 20 CAPABILITY:Min Length – 1 , Max Length – 99 (refer to DE-61) e.g: "52000002000100040006392929292 ". For more details, refer annexure document TELECOM OPERATOR:Min Length-1,Max Length-99 (It is mandatory for USSD)
035_ReqPay_DeviceDetails_type	5.8.2	If(Device.tag.name="Type")	Device type	Initiating Channel 1. MOB(Mobile) 2. INET(Internet) 3. USDC/USDB(USSD) 4. POS(Point of Sale)

Rule Id	Tag num	Condition	Value	Action
036_ReqPay_DeviceDetails_OS	5.8.2	If(Device.tag.name="OS")	Device OS	OS of the initiating Device 1. iOS 2. Android
037_ReqPay_Payer/Payee_MerchantTag	5.16	If(Payer.type=ENTITY)	Payer/Payee Merchant block	If the merchant comes through an aggregator then the merchant block element will contain the merchant details as follows 1. **Identifier.subCode=**MCC code of the merchant 2. **Identifier.mid**="Merchant id" 3. **Identifier.sid=**"Store id" 4. **Identifier.tid=**"Terminal id" 5. **Name.brand =** Brand any of the merchant 6. **Name.Legal=**Legal Name of the merchant 7. **Name.Franchise=**Franchise agent name 8. **Ownership.type=** See rule 038 9. merchantType="SMALL|LARGE" 10. merchantGenre="OFFLINE|ONLINE" 11. onBoardingType="BANK|AGGREGATOR"
038_ReqPay_MerchantTag_Ownership_Type	5.19.1	If(Payer.type=ENTITY)	Payer/Payee Merchant tag_owner_type	Type of Ownership: PROPRIETARY PARTNERSHIP PRIVATE PUBLIC OTHERS
039_ReqPay_OrgRespCode	4.1.11	If(txn.type=REVERSAL)	Txn tag_Reversal	Possible only if Reversal/Refund scenario occurs

Rule Id	Tag num	Condition	Value	Action
040_ReqPay_Credblock	5.12, 5.12.2	If(Txn.payment= Aadhaar)	Cred block	Cred type="AADHAAR" subType="AADHAAR-BIO-FP|AADHAAR-BIO-IRIS|AADHAAR-BIO-OTP
041_RespAuthDetail UPI-mandate_CollectCredblock	5.12	If(Txn.type=Collect)	Cred block	This cred block will come in ReqPay (Debit)and RespAuthDetails for UPI Mandate transactions. <Cred type="UPI-Mandate" subType="DS">.
042_ReqPay_Initiation mode	5.20	If(initiation mode="12")	Payers institution block	This institution block should contain all the mandatory fileds mentioned in the ReqPay table as per tag no:5.20. This XML block will be applicable to ReqPay & ReqAuthDetails.
043_ReqPay_Institution_type	5.20.1	If(type="MTO|BANK")	Payers Institution type	Only these two modes of payment type is admissible. 1.MTO- Money Transfer Operator 2.BANK
044_ReqPay_Institution_route	5.20.2	If(route="MTSS|RDA")	Payers Institution route	Only these two modes of payment route is admissible. 1.MTSS-Money transfer service scheme 2.RDA- Rupee Drawing Arrangement
045_ReqPay_Txn_purpose	4.1.12	If(txn_type=PAY|COLLECT|REFUND|REVERSAL|DEBIT|CREDIT) For mandate txn also	00|01|02|03|04|05|06|07|08|09|10	The purpose field is specially used for SEBI txn 00- DEFAULT 01-SEBI 02- AMC 03- Travel 04- Hospitality 05- Hospital 06- Telecom

Rule Id	Tag num	Condition	Value	Action
				07- Insurance 08- Education 09- Gifting 10- Others
046_ReqPay_Ac_addrType	5.9.1	If(addrType=AADHAAR|ACCOUNT|MOBILE|CARD)	Account values	1.If addrType= AADHAAR is applicable for Aadhaar txn's 2. If addrType=ACCOUNT is applicable for account + IFSC txn's 3. If addrType=MOBILE is applicable for mobile banking txn's 4. If addrType=CARD is applicable for card payments.
047_ReqPay_Ac_name_Aadhaar		If(addrType=AADHAAR)	Aadhaar values	If addrType=AADHAAR, then two below details are mandatory IIN= It should be 6 digit numeric UIDNUM= It should be 12 digit numeric assigned by UIDAI
048_ReqPay_Ac_name_Account		If(addrType=ACCOUNT)	Account values	If addrType=ACCOUNT, then three below details are mandatory IFSC= It should be 11 digit alphanumeric ACTYPE= It should be a fixed value SAVINGS|DEFAULT|CURRENT|NRE|NRO|PPIWALLET|BANKWALLET|CREDIT|SOD|UOD ACNUM= it should be max 30 digits
049_ReqPay_Ac_name_Mobile		If(addrType=MOBILE)	Mobile values	If addrType=MOBILE, then two below details are mandatory MOBNUM= It should contains 10 digit numeric with prefix +91. (total 12 digit) MMID=It should contains 7 digits numeric.

Rule Id	Tag num	Condition	Value	Action
050_ReqPay_Ac_name_Card		If(addrType=CARD)	Card values	If addrType=CARD, then the below values are mandatory ACTYPE= It should be a fixed value SAVINGS|DEFAULT|CURRENT CARDNUM=It should be Max-16 digits Numeric
051_ReqPay_Amount_Value		If (amount, orgAmount, settamount)	Amount value	The amount value should be numeric. It should be populate in below format. 2 digit should come after the decimal. E.g.(Amount Value="100.00")
052_ReqPay_Txn_refCategory		If(txn_type=PAY|COLLECT|REFUND|REVERSAL|DEBIT|CREDIT) For mandate txn also	00|01|02|03|04|05|06|07|08|09	If refUrl is present, then refCatergory is mandatory. The refCategory field is used to identify the category of the transaction 00- NULL 01-Advertisement 02- Invoice Others for future use

9. References

1. "RBI Payment System Vision document", RBI, 2012-15, http://rbi.org.in/scripts/PublicationVisionDocuments.aspx?ID=664
2. "Committee on Comprehensive Financial Services for Small Businesses and Low Income Households", RBI, January 2014, http://www.rbi.org.in/Scripts/PublicationReportDetails.aspx?UrlPage=&ID=727
3. "Report of the Technical Committee on Mobile Banking", RBI, February 2014, http://www.rbi.org.in/Scripts/PublicationReportDetails.aspx?UrlPage=&ID=760#8
4. "Report on Enabling PKI in Payment System Applications", RBI, April 2014, http://www.rbi.org.in/Scripts/PublicationReportDetails.aspx?UrlPage=&ID=765

5. "Pradhan Mantri Jan-Dhan Yojana", Ministry of Finance, August 2014, http://www.pmjdy.gov.in/financial_literacy.aspx
6. "Report of the Task Force on an Aadhaar-Enabled Unified Payment Infrastructure", Finance Ministry, February 2012, http://finmin.nic.in/reports/Report_Task_Force_Aadhaar_PaymentInfra.pdf
7. "Role of Biometric Technology in Aadhaar Authentication", UIDAI, March 2012, http://uidai.gov.in/images/role_of_biometric_technology_in_aadhaar_authentication_020412.pdf
8. "Micro-ATM Standards", IBA, March 2013, http://www.iba.org.in/upload/MicroATM_Standards_v1.5.1_Clean.pdf
9. "Immediate Payment System (IMPS)", NPCI, http://www.npci.org.in/imps_product.aspx
10. "Aadhaar Authentication", UIDAI, http://uidai.gov.in/auth
11. "Aadhaar e-KYC API Specification", UIDAI, http://uidai.gov.in/images/aadhaar_kyc_api_1_0_final.pdf
12. "Aadhaar Enabled Payment Systems (AEPS)", NPCI, http://www.npci.org.in/AEPSOverview.aspx
13. "Aadhaar Payment Bridge (APB)", NPCI, http://www.npci.org.in/apbs.aspx
14. "RuPay", NPCI, http://www.npci.org.in/RuPayBackground.aspx
15. "National Payment Corporation of India", NPCI, http://www.npci.org.in/home.aspx