UNIFIED PAYMENT INTERFACE

API and Technology Specifications

Specifications – Version 1.0

Contents

Glossary

Sender / Payer	Person/Entity who pays the money. Account of payer is debited as part of the payment transaction.
Receiver / Payee	Person/Entity who receives the money. Account of payee is credited as part of the payment transaction.
Customer	An individual person or an entity that has an account and wishes to pay or receive money.
Payment Account (or just Account)	Any bank account or any other payment accounts (PPI, Wallets, Mobile Money, etc.) offered by a regulated entity where money can be held, money can be debited from, and can be credited to.
Payment System Player (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 or entities.
NPCI	National Payment 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, interbank 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 Aadhaar number.
APB	Aadhaar Payment Bridge. A system allowing remittances to be made to an Aadhaar number without providing any other bank or account details.
2-FA	Two factor authentication.
USSD	Unstructured Supplementary Services Data
UPI	Unified Payments Interface
API	Application Programming Interface
AUA	Authentication User Agency

1. Introduction

Over decades, India has made steady progress in the field of electronic payments. The innovations in payments have leveraged major technological innovations in each era. However, given the scale and diversity of our country, much remains to be done and we cannot rest on our laurels.

This Unified Payment Interface (UPI) document provides a payments architecture that is directly linked to achieving the goals of universal electronic payments, a less cash society, and financial inclusion, using the latest technology trends, laid down in the RBI Payment System Vision Document (2012-15).

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

Mission Statement

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 systems will be driven by customer demands of convenience, ease of use and access that will impel the necessary convergence in innovative e-payment products and capabilities. Regulation will channelize innovation and competition to meet these demands consistent with international standards and best practises.

It 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 at just 1.1 million.
3. Of about six lakh villages in India, the total number of villages with banking services stands at less than one lakh villages as at end March 2011 and nearly 145 million households are excluded from banking. Over the last few years, significant improvements have come in terms of coverage and with Direct Benefits Transfer (DBT) and Jan Dhan Yojana (PMJDY), number of households having bank account has also gone up.

NPCI was set up in April 2009 with the core objective to consolidate and integrate the 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 Payment 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 swiping a phone book entry and making a call on mobile phone. With UPI system everyone who has an account can send and receive money from their mobile phone with just an identifier without having any other bank/account details. All they need to do is to "*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 between multiple identifiers such as Aadhaar number, mobile number, and new virtual payment addresses.
- **Security** - System provides end to end strong security and data protection. Considering self-service mobile applications, data capture is strongly encrypted at capture. Similarly, solution allows a mechanism to pay and collect using true virtual addresses without having to reveal any bank/account details. While providing convenience, solution offers 1-click 2-factor authentication, risk scoring, protection from 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 presence 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 can be *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 so that they can use the virtual address for making and receiving payments.

1.1 Objectives

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

Following are some of the key feautures of the Unified Payment Interface.

1. The UPI is expected to further propel easy instant payments via mobile, web, and other applications.
2. The payments can be both sender (payer) and 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.

2. UPI Architecture

This chapter covers the UPI architecture. After introducing the core features, high level architecture, key concepts, and 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.

2.1 Core Features

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

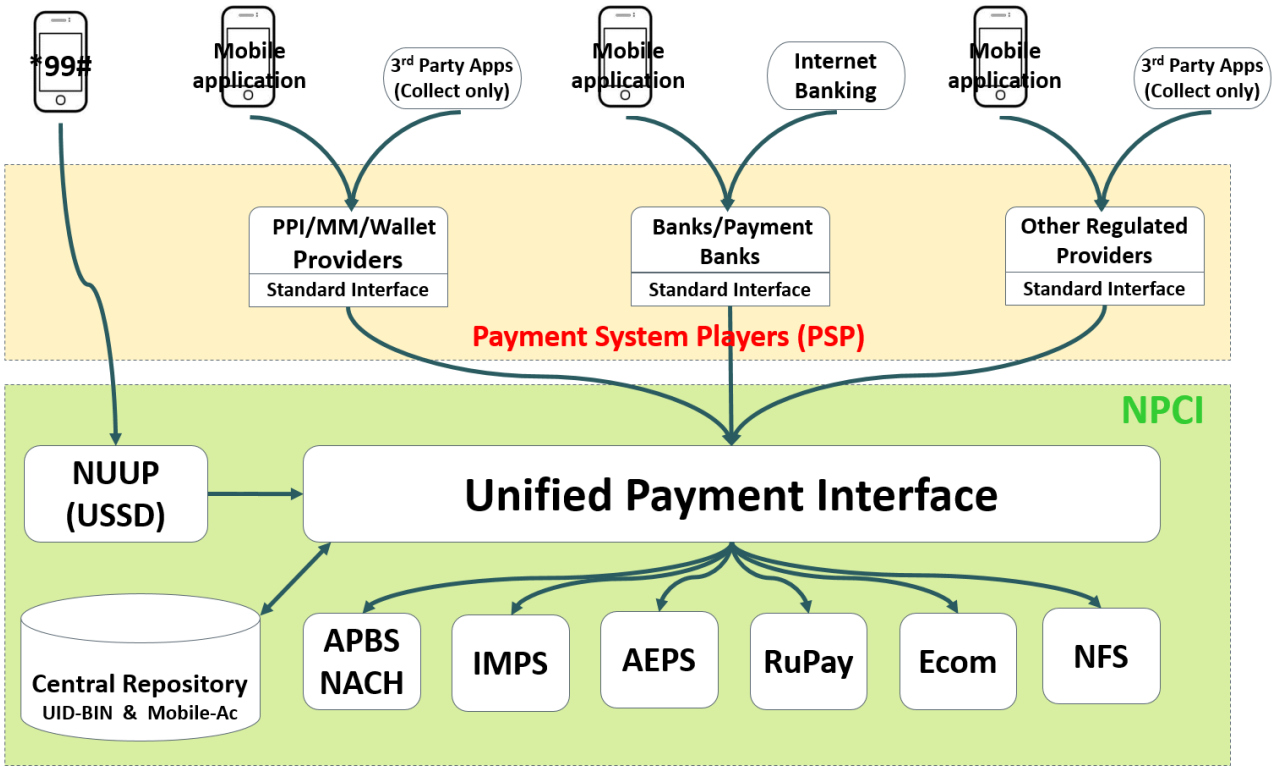
1. Ability to use personal mobile as the primary device for all payments including person to person, person to entity, and entity to person.
2. Ability to use personal mobile to "*pay*" someone (push) as well as "*collect*" from someone (pull).
3. Ability to use Aadhaar number, mobile number, card number, and account number in a unified way. In addition, ability to pay and collect using "*virtual payment addresses*" that are "*aliases*" to accounts that may be payee/amount/time limited providing further security features.
4. Make payments only by providing an address without having to ever provide account details or credentials on 3rd party applications or websites.
5. Ability for sending “collect” requests to others (person to person or entity to person) with "pay by" date to allow customer 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 for all PSPs to use a standard set of APIs for any-to-any push and pull payments.
8. Ability to have PSP provided mobile applications that allow paying from any account using any number of virtual addresses using 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 without any acquiring devices or physical tokens.

2.2 Architecture

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

As illustrated in the diagram, 3rd party API integration (merchant sites, etc.) can "*collect*" payment from "*an address*" avoiding the need to share account details or credentials on 3rd party applications or websites. Within this solution, payment authentication and authorization are always done using personal phone. Since this layer offers a unified interface, any-to-any (Aadhaar number, mobile, account, virtual addresses) payments to be done using standard set of APIs.



2.3 Concepts

Every payment has the following core elements:

1. Payer and payee account and institution details for routing and authorization
2. Authentication credentials (password, 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. Other 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.

2.3.1 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 identify an 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 account mapper, is represented as **`mobile-no@mobile.npci`** `(e.g. 9800011111@mobile.npci)`
- RuPay card number, resolved directly by NPCI, is represented as **`card-no@rupay.npci`** `(e.g. 1234123412341234@rupay.npci)`
- 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. 000012346789@myppi)`

- A user id provided by PSP, resolved directly by that PSP, is represented as **`user-id@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.

2.3.2 Authentication

Traditionally, payment account provider themselves provided 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 schemes to be plugged into the architecture as long as account providers allow such trusted external authentications. In UPI one of the authentication 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.

2.3.3 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 was still 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.

2.4 Supporting Infrastructure

2.4.1 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.

2.4.1.1 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 data or documents available with it. Authentication module handles online resident authentication from various Authentication User Agencies (AUA).

2.4.1.2 Aadhaar e-KYC

The Aadhaar e-KYC service provides a convenient mechanism for agencies to offer an electronic, paperless KYC experience to Aadhaar holders. The e-KYC service provides simplicity to the resident, while providing cost-savings from processing paper documents and eliminating the risk of forged documents to the service agencies. This service is offered via an Application Programming Interface (API) that allows organizations to integrate Aadhaar e-KYC within their applications.

Aadhaar e-KYC service is now approved by the RBI as a valid KYC process. PSPs can easily integrate these services within their application to provide low cost, paperless on-boarding of their customers.

2.4.1.3 Aadhaar Enabled Payment System (AEPS)

Aadhaar Enabled Payments System (AEPS) enables banks to route the financial transactions through a switching and clearing agency to empower the resident to use Aadhaar as his identity to authenticate and subsequently operate his respective Aadhaar enabled account and perform basic financial transactions.

MicroATMs allow customers to perform basic financial transactions (Deposit, Withdrawal, Funds Transfer, Balance Enquiry and Mini Statement) using the Aadhaar number and their fingerprint as identity proof (along with a Bank Identification Number for inter-bank transactions). The cash-in / cash-out functions of the microATMs are performed by an agent of the bank. This would not only offer convenience to the resident but would also reduce credit and operational risks for the banking system apart from reducing transaction costs.

The interoperable Aadhaar-enabled payments architecture is an overlay on the existing payment architecture, where authentication information is routed to UIDAI.

2.4.2 NPCI Central Mapper

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

The Aadhaar Payments Bridge System (APBS) uses NPCI central mapper as a part of National Automated Clearing House (NACH) to enable Government 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.

Similarly, central mapper allows anyone to send/receive money from a mobile number without knowing the destination account details. This is achieved by mapping mobile number to one or more account.

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.

2.4.2.1 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 beneficiaries 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.

Currently the mapper has about 160 million Aadhaar to bank mappings in its database. As part of large scale adoption of Direct Benefits Transfer (DBT) across all subsidy systems, it is expected that mapping database will have about 200-250 million Aadhaar mappings within next 12-18 months.

3. 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 payment interface.

3.1.1 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 payment 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. On the mobile application, using a single click on his address book entry of his wife, he enters an amount and click send. SBI application allows him to remember the amount for future use.

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

1. Validates user and debit his account.
2. Uses unified payment interface and initiates a "Pay" transaction with "payee" address to be simply "Aadhaar number" of Laxmi.
3. NPCI unified payment 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 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.

3.1.2 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 shyam.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 through IMPS system.
9. PNB credits Ram's account and responds to NPCI.
10. PNB pushes a notification to Ram's mobile number confirming the credit.

3.1.3 Buying on an ecommerce site

Sita is browsing myCartDeal for a deal on furniture. She finds a good for 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 happened to be Yes Bank.
6. Yes Bank application sends a notification to Sita's mobile application.
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 send 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

3.1.4 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.
    a. He had used his PSP application to create a new virtual address "abdul2014.irctc@mypsp".
    b. His PSP allows a feature to limit specific addresses only for collect from a specific merchant with a maximum amount limit!
    c. Since this is just a virtual address (merchant bound and amount limited), no one else can use it to collect money from him!
    d. This address is also bound (within Abdul's mobile app) to a default bank account.
3. 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.
4. NPCI sends the payment address to the PSP ("mypsp" in this case) where Abdul is registered with.
5. The PSP translates Abdul's Payment address and sends notification to his mobile to capture credentials.
6. Abdul enters his bank authentication credentials on his mobile device and does a single click authorization.
7. His PSP responds to NPCI with the actual account details which was bound to the virtual address along with encrypted authentication credentials.
8. NPCI sends the debit request to Abdul's bank that was sent back in response.
9. On successful response, NPCI sends credit request to IRCTCs bank account (which was part of collect request).
10. On successful response both IRCTC's PSP and Abdul are notified on the

3.1.5 Using a taxi services

Jaspreet has an account with a wallet provider myWallet (PSP). He regularly books MeLa cab. As part of his profile with MeLa booking application, he has provided his payment address "jasprt007@myWallet". He uses myWallet mobile application and authorizes the cab company payee address (MeLa@bank1) to auto charge him within Rs.1500. Now, every time he travels, he simply walks out of the cab and MeLa can charge Jaspreet automatically within the set limit.

Jaspreet gets notified on every charge and can anytime decide to pause or deactivate the automatic authorization. Both Jaspreet and MeLa can be on separate PSP networks and still transact conveniently.

3.1.6 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, above 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. High Level Specifications

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.

4.1 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).

4.1.1 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.

4.1.2 System Initiated

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

4.1.3 Transaction Flow

1. Payer initiates transaction through his PSP application at his Device.
2. Payer provides authentication credentials at 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 #, Aadhaar # or Account #)

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 6b, the Payee PSP accepts or rejects the request based on the rules set at his end.
8. In case of 6b, on accepting the Pay request, Payee PSP populates the Payee 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 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.

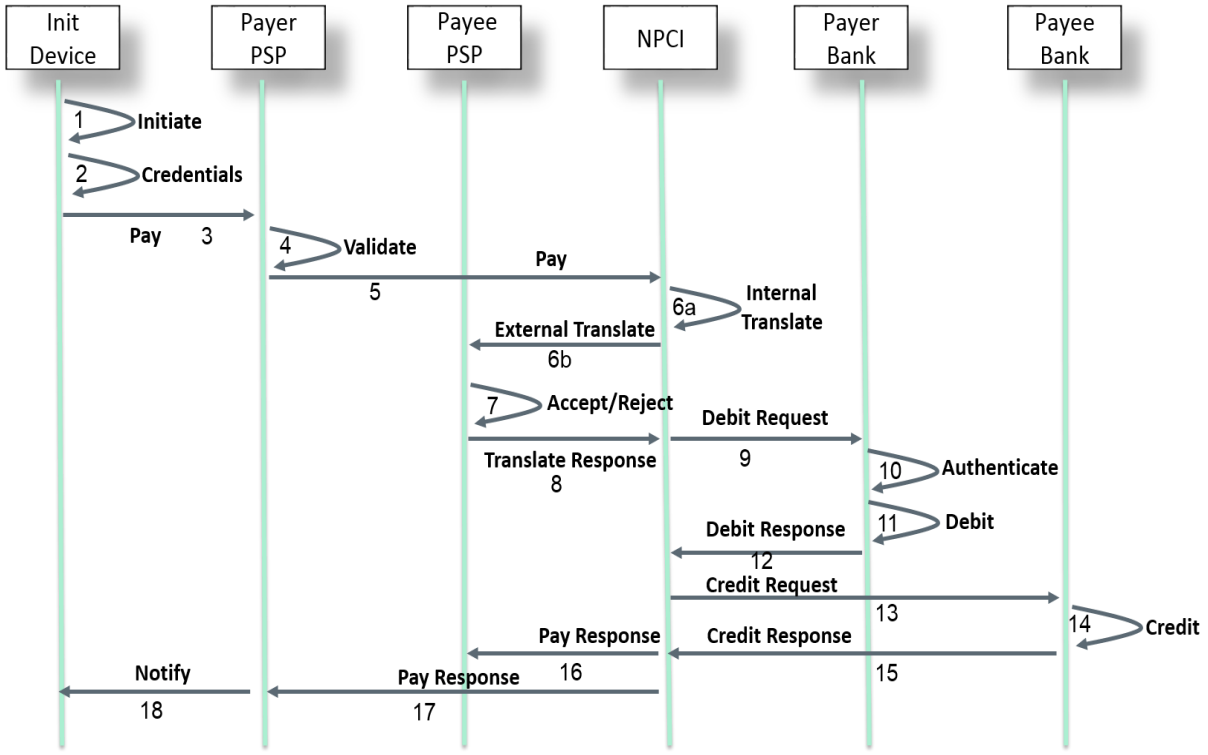


Figure4.1

4.1.4 Failure Scenarios

This section explains how the 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 do not reach Payee/Payer PSP:

In this scenario, when the response sent by NPCI do 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 do not reach NPCI:

In this scenario, when the response sent by Payee bank do not reach NPCI, this transaction will be considered as Deemed acceptance and Deemed acceptance Response will be sent to Payee/Payer PSP's. 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/Payer PSP's. 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/Payer PSP's 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/Payer PSP's 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/Payer PSP's with declined response. No credit request will be initiated to Payee bank.

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

In this scenario, when the response sent by Payer bank do not reach NPCI, NPCI will timeout the transaction and send reversal message to Payer bank. 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/Payer PSP's 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 do not reach NPCI:

In this scenario, when the response sent by Payee PSP do 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 may have a mechanism to re initiate the Pay request to NPCI.

4.2 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.

4.2.1 Remote Collect

1. Payee/Receiver (persons or entities) 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, wallet
2. Examples
    a. Kirana store person uses his/her phone app to "collect" by entering customer mobile number
    b. Car service agency application "collecting" payment via mobile number without car owner having to go to collect car
    c. Magazine subscription application requesting authorization for subscription renewal

4.2.2 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 in UPI.

4.2.3 Transaction Flow

1. Payee initiates transaction through his PSP application at 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 # or Account #) then PSP to request NPCI for pending messages via API against a given mobile number or Aadhaar number.
    b. If the Address has 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 his 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 at 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 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 payer.

The following diagram illustrates the above flow.

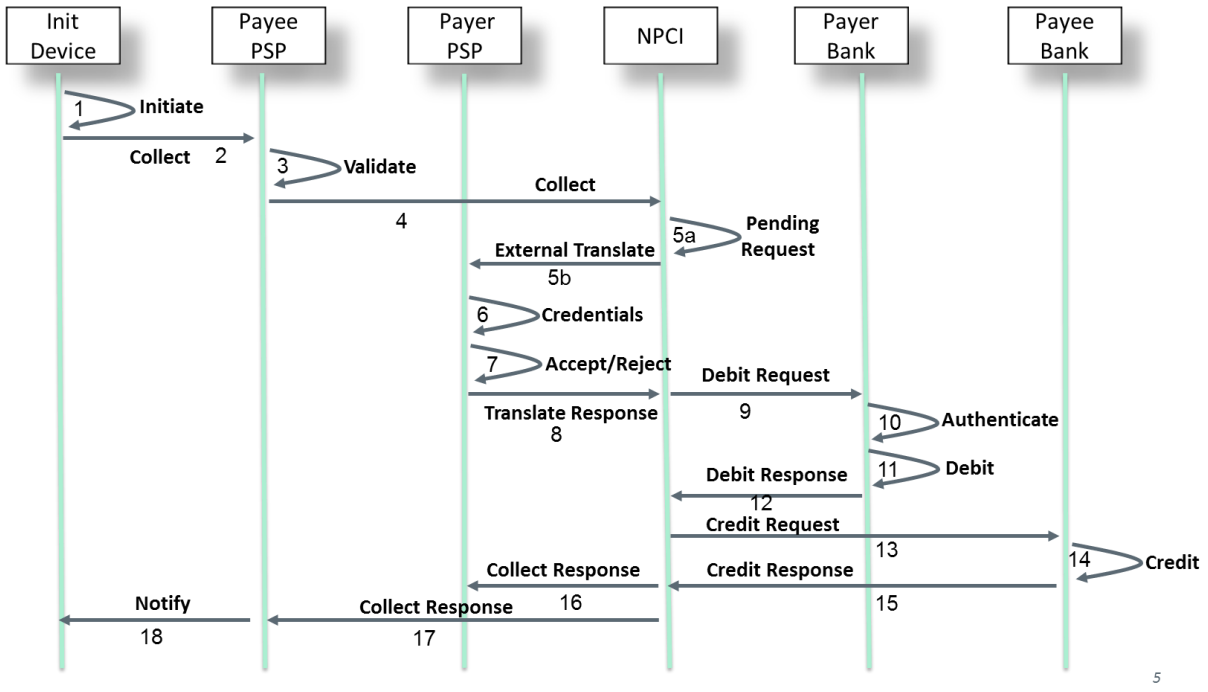


Figure 4.2

4.2.4 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 do not reach Payee/Payer PSP:

In this scenario, when the response sent by NPCI do 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 do not reach NPCI:

In this scenario, when the response sent by Payee bank do not reach NPCI, this transaction will be considered as Deemed acceptance and Deemed acceptance Response will be sent to Payee/Payer PSP's. 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/Payer PSP's. 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/Payer PSP's 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/Payer PSP's 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/Payer PSP's with declined response. No credit request will be initiated to Payee bank.

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

In this scenario, when the response sent by Payer bank do not reach NPCI, NPCI will timeout the transaction and send reversal message to Payer bank. 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/Payer PSP's 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 do not reach NPCI:

In this scenario, when the response sent by Payer PSP do not reach NPCI, NPCI will wait for the response till the timeout period. Payer 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 Payee PSP's 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 Pay request to NPCI.

4.3 APIs at a Glance

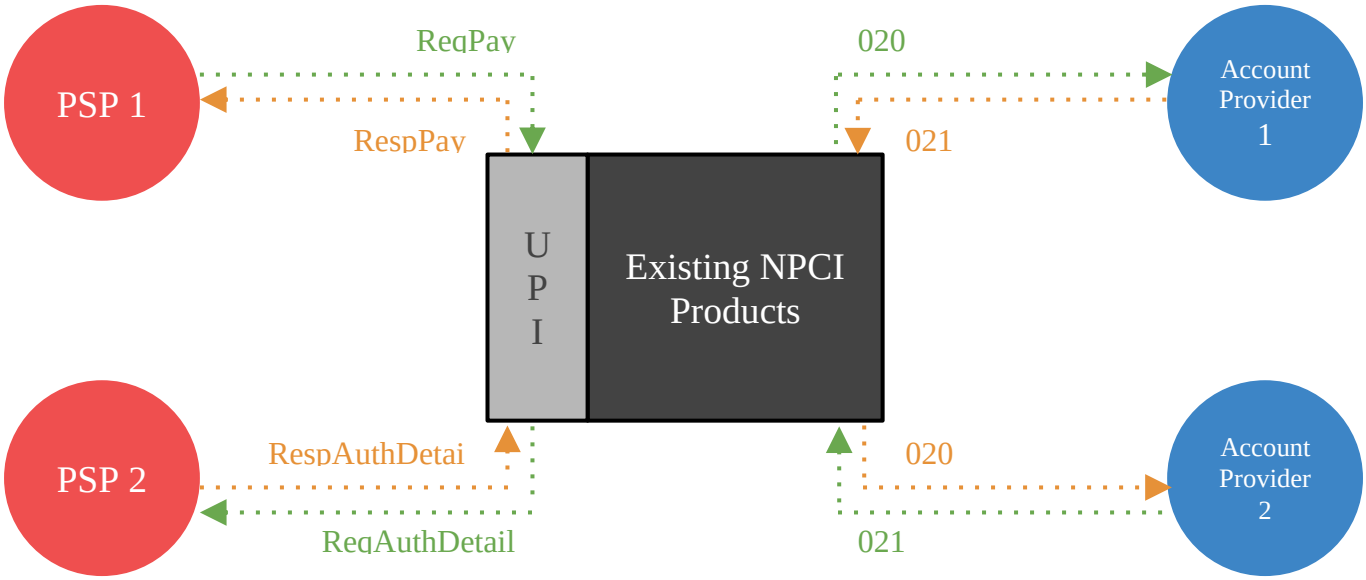
All APIs are asynchronous in nature meaning once the request is sent, response is sent back separately via corresponding response API. This allows same APIs to be used for instant payment as well as delayed payments. This also allows APIs to scale without having to wait in a blocking mode. Callers are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

4.3.1 Unified Interface - Message Flow

Diagram below depicts a general scenario of 4-party flow 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 done using XML over HTTPS whereas all APIs behind the existing systems at NPCI are done over ISO 8583 Messages (0200/0210).

4.3.2 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 the 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 the complete details to process the financial transaction, the debit and credit will be processed through the online products like IMPS, AEPS etc.

4.3.3 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 back the details. After processing the API, PSP should send response to the authorization by calling the "RespAuthDetails" API at NPCI.

4.4 Security Considerations

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

1. **Sensitive Data** - Data such as PIN, passwords, biometrics, etc. These are not to be stored and should only 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.) that can be stored in unencrypted form.

4.4.1 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 so 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
Account Details - Number, Account Ownership,	PSP using the issuer credentials (captured via common library)	Every time a payment account is added	During first transaction

4.4.2 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.

4.4.3 Protecting Authentication Credentials

- Trusted common library for credential (MPIN/Password/PIN/Biometrics etc) capture is provided by NPCI. This library needs to be integrated with PSP application.
- 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.

4.4.4 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.

4.4.5 Message Security, Trust, and Non-Repudiability

- Every messages 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. Detail API Specifications

5.1 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>
```

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 Payment 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".

All APIs have same ack response as given below:

```
<upi:Ack xmlns:upi="" api="" reqMsgId="" err="" 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 - the timestamp at which the receiver sends the acknowledgement.

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

Sno#	API NAME	API Description
1	**ReqPay**	API will be used for both Direct Pay and Collect Pay transaction initiation by the PSP's and processing the transaction through one of the following channels IMPS,AEPS etc.
2	**RespPay**	API will be used for sending back the response of transaction (Direct and Collect Pay) initiated through ReqPay Api to the PSP's
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.API is called to translate PSP address and obtain appropriate authorization details
4	**RespAuthDetails**	"RespAuthDetails" API is the response call back interface to return back the details. After processing the API, PSP should send response to the authorization by calling the "RespAuthDetails" API at NPCI.
5	**List PSP**	This API allows the PSPs to request for the list of all registered PSPs for local caching. This data should be used for validating payment address before initiating the transaction.
6	**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.
7	**List Keys**	This API allows the PSPs to request for 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.
8	**List Account**	API allows PSPs to find the list of accounts linked to the mobile by an account provider.
9	**List Verified Address Entries**	API allows PSPs to request a and cache the List of Verified Address Entries to protect customers from attempts to spoof well known merchants such as LIC, Indian Railways, e-commerce players, telecom players, bill payment entities, etc.
10	**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.
11	**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).
12	**Set Credentials**	This API is required for providing a unified channel for setting and changing MPIN across various account providers

13	**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.
14	**OTP-Request**	This API allows the PSPs to request for an OTP for a particular customer from an issuer.
15	**Balance-Enquiry**	This API Allows PSP to Request for Balance enquiry for a user.
16	**HeartBeat Messages**	This API is a mechanism for UPI system monitoring.(monitoring connection with PSPs and sending EOD to PSPs)
17	**Request Pending Messages**	This API allows PSP to request pending messages against a given mobile number or Aadhaar number.

5.2 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" ts="" orgId="" msgId=""/>
    <Meta>
        <Tag name="PAYREQSTART" value=""/>
        <Tag name="PAYREQEND" value=""/>
    </Meta>
    <Txn id="" note="" refId="" refUrl="" ts="" type="PAY|COLLECT|REFUND" orgTxnId="">
        <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="">
        <Info>
            <Identity 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=""/>
        </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"/>
            <Detail name="CARDNUM" value=""/>
        </Ac>
        <Creds>
            <Cred type="AADHAAR" subtype="IIR|FMR|FIR|OTP">
                <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">
            <Split name="PURCHASE|CASHBACK" value=""/>
        </Amount>
        <PreApproved respCode="" approvalRef=""/>
    </Payer>
    <Payees>
        <Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
            <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"/>
            </Device>
            <Ac addrType ="AADHAAR">
                <Detail name="IIN" value=""/>
                <Detail name="UIDNUM" value=""/>
            </Ac>
            <Amount value="" curr="INR">
                <Split name="PURCHASE|CASHBACK" value=""/>
            </Amount>
```

```
        </Payee>
    </Payees>
</upi:ReqPay>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<upi>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message	ts	1..1
2.1.3	Organization id that created the message	orgId	1..1
2.1.4	Message identifier-used to correlate between request and response	msgId	1..1
3.1	Meta data primarily for analytics purposes	<Meta>	0..1
3.2	Meta data primarily for analytics purposes	<Meta.Tag>	0..1
3.2.1	Name of the property	name	1..n
3.2.2	Value of the property	value	1..n
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1
4.1.1	Unique Identifier of the transaction across all entities, created by the originator	id	1..1
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1
4.1.4	URL for the transaction	refUrl	1..1
4.1.5	Transaction origination time by the creator of the message	ts	1..1
4.1.6	Type of the Transaction	type	1..1
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1
4.2	Risk Score related to the transaction and the entities	<Txn.RiskScores>	0..1
4.3	Risk Score related to the transaction and the entities	<Txn.RiskScores.Score>	0..1
4.3.1	Entity providing the risk score	provider	1..1
4.3.2	Type of risk	type	1..1
4.3.3	Value of risk evaluation ranging from 0 (No Risk) to 100 (Maximum Risk)	value	1..1
4.4	Rules that govern the payment	<Txn.Rules>	0..1
4.5	Rule for the transaction	<Txn.Rules.Rule>	0..n
4.5.1	Name of the property	name	1..n
4.5.2	Value of the property	value	1..n
5.1	Details related to the Payer	<Payer>	1..1
5.1.1	Address of the Payer	addr	1..1

Index	Message Item	<XML Tag>	Occurrence
5.1.2	Name of the Payer	name	1..1
5.1.3	Unique identifier for each transaction inside a file including payer and payee	seqNum	1..1
5.1.4	Type of the Payer	type	1..1
5.1.5	Merchant Classification Code -MCC	code	1..1
5.2	Information related to the Payer	<Payer.Info>	1..1
5.3	Payer Identity Is mandatory for "collect" and optional for "pay"	<Payer.Info.Identity>	1..1
5.3.1	Type of the identifier	type	1..1
5.3.2	Name as per the identifier	verifiedName	1..1
5.4	Rating of the payer	<Payer.Info.Rating>	0..1
5.4.1	Payer is whitelisted or not	whiteListed	1..1
5.5	Details of Device from which the transaction was initiated	<Payer.Device>	1..1
5.6	Device Tag	<Payer.Device.Tag>	1..n
5.6.1	Name of the property	name	1..n
5.6.2	Value of the property	value	1..n
5.7	Only one entity is allowed for a payer	<Payer.Ac>	1..1
5.7.1	Type of the address	addrType	1..1
5.8	Details related to Payer Address	<Payer.Ac.Detail>	1..n
5.8.1	Name of the property	name	1..n
5.8.2	Value of the property	value	1..n
5.9	Information related to Payer Credentials	<Payer.Creds>	1..1
5.10	Credentials are used to authenticate the request	<Payer.Creds.Cred>	1..1
5.10.1	Type of financial instrument used for authentication	type	1..1
5.11	base-64 encoded/encrypted authentication data	<Payer.Creds.Cred.Data>	1..1
5.12	Information related to the amounts in the transaction	<Payer.Amount>	1..1
5.12.1	Transaction amount	value	1..1
5.12.2	Currency of the transaction	curr	1..1
5.13	Details of transaction amount	<Payer.Amount.Split>	0..1
5.13.1	Name of the property	name	1..n
5.13.2	Value of the property	value	1..n
5.14	Information if the debit is already authorized	<Payer.PreApproved>	0..1
5.14.1	Response Code	respCode	1..1
5.14.2	Approval Reference	approvalRef	1..1
6.1	Details related to the Payees	<Payees>	1..1
6.2	Details related to the Payee	<Payee>	1..1
6.2.1	Address of the Payee	addr	1..1
6.2.2	Name of the Payee	name	1..1
6.2.3	Unique identifier for each transaction inside a file including Payee and payee	seqNum	1..1
6.2.4	Type of the Payee	type	1..1

Index	Message Item	<XML Tag>	Occurrence
6.2.5	Merchant Classification Code -MCC	code	1..1
6.3	Information related to the Payee	<Payee.Info>	1..1
6.4	Payee Identity	<Payee.Info.Identity>	1..1
6.4.1	Type of the identifier	type	1..1
6.4.2	Name as per the identifier	verifiedName	1..1
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1
6.5.1	Payee is whitelisted or not	whiteListed	1..1
6.6	Details of Device from which the transaction was initiated	<Payee.Device>	1..1
6.7	Device Tag	<Payee.Device.Tag>	1..n
6.7.1	Name of the property	name	1..n
6.7.2	Value of the property	value	1..n
6.8	Only one entity is allowed for a Payee	<Payee.Ac>	1..1
6.8.1	Type of the address	addrType	1..1
6.9	Details related to Payee Address	<Payee.Ac.Detail>	1..n
6.9.1	Name of the property	name	1..n
6.9.2	Value of the property	value	1..n
6.10	Information related to the amounts in the transaction	<Payee.Amount>	1..1
6.10.1	Transaction amount	value	1..1
6.10.2	Currency of the transaction	curr	1..1
6.11	Details of transaction amount	<Payee.Amount.Split>	0..1
6.11.1	Name of the property	name	1..n
6.11.2	Value of the property	value	1..n

5.3 RespPay

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

```
<upi:RespPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="" orgId="" msgId=""/>
    <Txn id="" note="" refId="" refUrl="" ts="" type="PAY|COLLECT|REFUND" orgTxnId="">
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="npci" type="TXNRISK" value=""/>
        </RiskScores>
    <Resp reqMsgId="" result="SUCCESS|FAILURE|PARTIAL|DEEMED" errCode="">
        <Ref type="PAYER" seqNum="" addr="" settAmount="" settCurrency=""
             approvalNum="" respCode=""/>
        <Ref type="PAYEE" seqNum="" addr="" settAmount="" settCurrency=""
             approvalNum="" respCode=""/>
    </Resp>
</upi:RespPay>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<RespPay>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message	ts	1..1
2.1.3	Organization id that created the message	orgId	1..1
2.1.4	Message identifier-used to correlate between request and response	msgId	1..1
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1
4.1.1	Unique Identifier of the transaction across all entities created by the originator	id	1..1
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1
4.1.3	Consumer reference number to identify (like Loan number, etc.)	refId	1..1
4.1.4	URL for the transaction	refUrl	1..1
4.1.5	Transaction origination time by the creator of the message	ts	1..1
4.1.6	Type of the Transaction	type	1..1
4.1.7	Original transaction ID when reversal/Refund has to be done	orgTxnId	1..1
11.1	Response	<Resp>	1..1
11.1.1	Request Message identifier	reqMsgId	1..1
11.1.2	Result of the transaction	result	1..1
11.1.3	Error code if failed	errCode	1..1
11.2	Response Reference	<Ref>	1..n
11.2.1	Customer type	type	1..1
11.2.2	Sequence Number	seqNum	1..1
11.2.3	Payment address	addr	1..1
11.2.4	Settlement Amount	settAmount	1..1
11.2.5	Settlement Currency	settCurrency	1..1
11.2.6	Approval Reference Number	approvalNum	1..1
11.2.7	Response code	respCode	1..1

5.4 ReqAuthDetails

Input message XML for ReqAuthDetails API.

```
<upi:ReqAuthDetails xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="" orgId="NPCI" msgId=""/>
    <Txn id="" note="" ref="" ts="" type="PAY">
        <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 type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
            <Rating VerifiedAddress="TRUE|FALSE"/>
        </Info>
        <Amount value="" curr="INR">
            <Split name="PURCHASE|CASHBACK" value=""/>
        </Amount>
    </Payer>

    <Payees>
        <Payee seqNum="" addr="" name="">
            <Info>
                <Identity type="PAN|AADHAAR|ACCOUNT" verifiedName=""/>
                <Rating VerifiedAddress="TRUE|FALSE"/>
            </Info>
              <Amount value="" curr="INR">
                  <Split name="PURCHASE|CASHBACK" value=""/>
              </Amount>
        </Payee>
    </Payees>
</upi:ReqAuthDetails>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ReqAuthDetails>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message	ts	1..1
2.1.3	Organization id that created the message	orgId	1..1
2.1.4	Message identifier-used to correlate between request and response	msgId	1..1
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1
4.1.1	Unique Identifier of the transaction across all entities created by the originator	id	1..1
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1
4.1.3	Consumer reference number to identify (like Loan number, etc.)	ref	1..1
4.1.4	Transaction origination time by the creator of the message	ts	1..1
4.1.5	Type of the Transaction	type	1..1
4.2	Risk Score related to the transaction and the entities	<Txn.RiskScores>	0..1
4.3	Risk Score related to the transaction and the	<Txn.RiskScores.Score>	0..1

Index	Message Item	<XML Tag>	Occurrence
	entities		
4.3.1	Entity providing the risk score	provider	1..1
4.3.2	Type of risk	type	1..1
4.3.3	Value of risk evaluation ranging from 0 (No Risk) to 100 (Maximum Risk)	value	1..1
4.4	Rules that govern the payment	<Txn.Rules>	0..1
4.5	Rule for the transaction	<Txn.Rules.Rule>	0..n
4.5.1	Name of the property	name	1..n
4.5.2	Value of the property	value	1..n
5.1	Details related to the Payer	<Payer>	1..1
5.1.1	Address of the Payer	addr	1..1
5.1.2	Name of the Payer	name	1..1
5.1.3	Unique identifier for each transaction inside a file including payer and payee	seqNum	1..1
5.1.4	Type of the Payer	type	1..1
5.1.5	Merchant Classification Code -MCC	code	1..1
5.2	Information related to the Payer	<Payer.Info>	1..1
5.3	Payer Identity is mandatory for “collect” and optional for “pay”	<Payer.Info.Identity>	1..1
5.3.1	Type of the identifier	type	1..1
5.3.2	Name as per the identifier	verifiedName	1..1
5.4	Rating of the payer	<Payer.Info.Rating>	0..1
5.4.1	Payer is whitelisted or not	whiteListed	1..1
5.12	Information related to the amounts in the transaction	<Payer.Amount>	1..1
5.12.1	Transaction amount	value	1..1
5.12.2	Currency of the transaction	curr	1..1
5.13	Details of transaction amount	<Payer.Amount.Split>	0..1
5.13.1	Name of the property	name	1..n
5.13.2	Value of the property	value	1..n
6.1	Details related to the Payees	<Payees>	1..1
6.2	Details related to the Payee	<Payee>	1..1
6.2.1	Address of the Payee	addr	1..1
6.2.2	Name of the Payee	name	1..1
6.2.3	Unique identifier for each transaction inside a file including Payee and payee	seqNum	1..1
6.2.4	Type of the Payee	type	1..1
6.2.5	Merchant Classification Code -MCC	code	1..1
6.3	Information related to the Payee	<Payee.Info>	1..1
6.4	Payee Identity	<Payee.Info.Identity>	1..1
6.4.1	Type of the identifier	type	1..1
6.4.2	Name as per the identifier	verifiedName	1..1
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1

Index	Message Item	<XML Tag>	Occurrence
6.5.1	Payee is whitelisted or not	whiteListed	1..1
6.8	Only one entity is allowed for a Payee	<Payee.Ac>	1..1
6.8.1	Type of the address	addrType	1..1
6.9	Details related to Payee Address	<Payee.Ac.Detail>	1..n
6.9.1	Name of the property	name	1..n
6.9.2	Value of the property	value	1..n
6.10	Information related to the amounts in the transaction	<Payee.Amount>	1..1
6.10.1	Transaction amount	value	1..1
6.10.2	Currency of the transaction	curr	1..1
6.11	Details of transaction amount	<Payee.Amount.Split>	0..1
6.11.1	Name of the property	name	1..n
6.11.2	Value of the property	value	1..n

5.5 RespAuthDetails

Following is the XML data format for RespAuthDetails API.

```
<upi:RespAuthDetails xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="" orgId="" msgId=""/>
    <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
    <Txn id="" note="" ref="" ts="" type="PAY">
        <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 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">
        </Device>
        <Ac addrType ="AADHAAR">
            <Detail name="IIN" value=""/>
            <Detail name="UIDNUM" value=""/>
        </Ac>
        <Creds>
            <Cred type="AADHAAR" subtype="IIR|FMR|FIR|OTP">
                <Data> base-64 encoded/encrypted authentication data</Data>
```

```
            </Cred>
        </Creds>
        <Amount value="" curr="INR">
            <Split name="PURCHASE|CASHBACK" value=""/>
        </Amount>
        <PreAuth respCode="" approvalRef=""/>
    </Payer>
    <Payees>
        <Payee addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
            <Info>
                <Identity type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
                <Rating VerifiedAddress="TRUE|FALSE"/>
            </Info>
            <Ac addrType ="AADHAAR">
                <Detail name="IIN" value=""/>
                <Detail name="UIDNUM" value=""/>
            </Ac>
            <Amount value="" curr="INR">
                <Split name="PURCHASE|CASHBACK" value=""/>
            </Amount>
        </Payee>
    </Payees>
</upi:RespAuthDetails>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<RespAuthDetails>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message	ts	1..1
2.1.3	Organization id that created the message	orgId	1..1
2.1.4	Message identifier-used to correlate between request and response	msgId	1..1
3.1	Meta data primarily for analytics purposes	<Meta>	0..1
3.2	Meta data primarily for analytics purposes	<Meta.Tag>	0..1
3.2.1	Name of the property	Name	1..n
3.2.2	Value of the property	value	1..n
11.1	Response	<Resp>	1..1
11.1.1	Request Message identifier	reqMsgId	1..1
11.1.2	Result of the transaction	result	1..1
11.1.3	Error code if failed	errCode	1..1
4.1	Transaction information, Carried throughout the system, visible to all parties	<Txn>	1..1
4.1.1	Unique Identifier of the transaction across all entities created by the originator	id	1..1
4.1.2	Description of the transaction(which will be printed on Pass book)	note	1..1
4.1.3	Consumer reference number to identify (like Loan number, etc.)	ref	1..1
4.1.4	Transaction origination time by the creator of the	ts	1..1

Index	Message Item	<XML Tag>	Occurrence
	message		
4.1.5	Type of the Transaction	type	1..1
4.2	Risk Score related to the transaction and the entities	<Txn.RiskScores>	0..1
4.3	Risk Score related to the transaction and the entities	<Txn.RiskScores.Score>	0..1
4.3.1	Entity providing the risk score	provider	1..1
4.3.2	Type of risk	type	1..1
4.3.3	Value of risk evaluation ranging from 0 (No Risk) to 100 (Maximum Risk)	value	1..1
5.1	Details related to the Payer	<Payer>	1..1
5.1.1	Address of the Payer	addr	1..1
5.1.2	Name of the Payer	name	1..1
5.1.3	Unique identifier for each transaction inside a file including payer and payee	seqNum	1..1
5.1.4	Type of the Payer	type	1..1
5.1.5	Merchant Classification Code -MCC	code	1..1
5.2	Information related to the Payer	<Payer.Info>	1..1
5.3	Payer Identity is mandatory for “collect” and optional for “pay”	<Payer.Info.Identity>	1..1
5.3.1	Type of the identifier	type	1..1
5.3.2	Name as per the identifier	verifiedName	1..1
5.4	Rating of the payer	<Payer.Info.Rating>	0..1
5.4.1	Payer is whitelisted or not	whiteListed	1..1
5.5	Details of Device from which the transaction was initiated	<Payer.Device>	1..1
5.6	Device Tag	<Payer.Device.Tag>	1..n
5.6.1	Name of the property	name	1..n
5.6.2	Value of the property	value	1..n
5.7	Only one entity is allowed for a payer	<Payer.Ac>	1..1
5.7.1	Type of the address	addrType	1..1
5.8	Details related to Payer Address	<Payer.Ac.Detail>	1..n
5.8.1	Name of the property	name	1..n
5.8.2	Value of the property	value	1..n
5.9	Information related to Payer Credentials	<Payer.Creds>	1..1
5.10	Credentials are used to authenticate the request	<Payer.Creds.Cred>	1..1
5.10.1	Type of financial instrument used for authentication	type	1..1
5.11	base-64 encoded/encrypted authentication data	<Payer.Creds.Cred.Data>	1..1
5.12	Information related to the amounts in the transaction	<Payer.Amount>	1..1
5.12.1	Transaction amount	value	1..1
5.12.2	Currency of the transaction	curr	1..1

Index	Message Item	<XML Tag>	Occurrence
5.13	Details of transaction amount	<Payer.Amount.Split>	0..1
5.13.1	Name of the property	name	1..n
5.13.2	Value of the property	value	1..n
5.14	Information if the debit is already authorized	<Payer.PreApproved>	0..1
5.14.1	Response Code	respCode	1..1
5.14.2	Approval Reference	approvalRef	1..1
6.1	Details related to the Payees	<Payees>	1..1
6.2	Details related to the Payee	<Payee>	1..1
6.2.1	Address of the Payee	addr	1..1
6.2.2	Name of the Payee	name	1..1
6.2.3	Unique identifier for each transaction inside a file including Payee and payee	seqNum	1..1
6.2.4	Type of the Payee	type	1..1
6.2.5	Merchant Classification Code -MCC	code	1..1
6.3	Information related to the Payee	<Payee.Info>	1..1
6.4	Payee Identity	<Payee.Info.Identity>	1..1
6.4.1	Type of the identifier	type	1..1
6.4.2	Name as per the identifier	verifiedName	1..1
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1
6.5.1	Payee is whitelisted or not	whiteListed	1..1
6.6	Details of Device from which the transaction was initiated	<Payee.Device>	1..1
6.7	Device Tag	<Payee.Device.Tag>	1..n
6.7.1	Name of the property	name	1..n
6.7.2	Value of the property	value	1..n
6.8	Only one entity is allowed for a Payee	<Payee.Ac>	1..1
6.8.1	Type of the address	addrType	1..1
6.9	Details related to Payee Address	<Payee.Ac.Detail>	1..n
6.9.1	Name of the property	name	1..n
6.9.2	Value of the property	value	1..n
6.10	Information related to the amounts in the transaction	<Payee.Amount>	1..1
6.10.1	Transaction amount	value	1..1
6.10.2	Currency of the transaction	curr	1..1
6.11	Details of transaction amount	<Payee.Amount.Split>	0..1
6.11.1	Name of the property	name	1..n
6.11.2	Value of the property	value	1..n

5.6 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.

5.6.1 List PSP

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

ReqListPsp: Request PSP list

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

RespListPsp: Response for PSP list

```
<upi:RespListPsp xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
      <PspList>
            <Psp name="HDFC" codes="hdfcgold,hdfcsliver" active="Y/N" url=""
            spocName="" spocEmail="" spocPhone="" lastModifedTs=""/>
            <Psp name="ICICI" codes="icici,iciciwallet" active="Y/N" url=""
            spocName="" spocEmail="" spocPhone="" lastModifedTs=""/>
      </PspList>
</upi:RespListPsp>
```

Index	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

5.6.2 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.

ReqListAccPvd: Request for Account Providers list

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

RespListAccPvd: Response for Account providers list

```
<upi:RespListAccPvd xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
      <AccPvdList>
            <AccPvd name="HDFC" iin="901345" active="Y/N" url="" spocName=""
      spocEmail="" spocPhone="" prods="AEPS,IMPS,CARD,NFS" lastModifedTs=""/>
            <AccPvd name="ICICI" iin="901346" active="Y/N" url="" spocName=""
      spocEmail="" spocPhone="" prods="AEPS,IMPS,CARD,NFS" lastModifedTs=""/>
      </AccPvdList>
</upi:RespListAccPvd>
```

Index	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	Status of the account provider if it is active or not	active	1..1
22.2.4	URL link provided by account provider	url	0..n
22.2.5	Name of the SPOC	spocName	0..n
22.2.6	E-mail of the SPOC	spocEmail	0..n
22.2.7	Phone Number of the SPOC	spocPhone	0..n
22.2.8	List of NPCI products for which account provider is live	prods	0..n
22.2.9	Last Modified date of the account provider information in the UPI system	lastModifiedTs	1..1

5.6.3 List Keys

NPCI maintains the list of all public keys for encryption. This API allows the PSPs to request for 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. These libraries can be provided by NPCI.

ReqListKeys: Request list of Key's

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

RespListKeys: Response for List of Key's

```
<upi:RespListKeys xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
      <keyList>
            <key code="HDFC" owner="NPCI | PSP | Bank |UIDAI" type="PKI"
ki="yyyymmdd">
                  <keyValue>base64 encoded certificate</keyValue>
            </key>
      </keyList>
</upi:RespListKeys>
```

Index	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

5.6.4 List Account

PSPs to find the list of accounts linked to the mobile by an account provider has to call this

ReqListAccount: Request for Account List

```
<upi:ReqListAccount xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
      <Link type="MOBILE|AADHAAR" value=""/>
```

```
</upi:ReqListAccount>
```

Index	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

RespListAccount: Response for Account List

```
<upi:RespListAccount xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
      <AccountList>
            <Account number="0101307583747" ifsc="HDFC0000101" mmid="9056014" name=""
aeba="Y/N"/>
            <Account number="0103307890765" ifsc="HDFC0000103" mmid="9056114" name=""
aeba="Y/N"/>
      </AccountList>
</upi:RespListAccount>
```

Index	Message Item	<XML Tag>	Occurrence
25.1	Account List	<AccountList>	1..1
25.2	Details Related to Account	<AccountList.Account>	1..n
25.2.1	Account Number	number	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

5.6.5 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" ts="" orgId="" msgId=""/>
</upi:ReqListVae>
```

RespList:Response for List of Verified Address Entries

```
<upi:RespListVae xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode="">
```

```
            <VaeList>
                  <Vae name="LIC" addr="lic@hdfc" logo="image" url=""/>
                  <Vae name="IRCTC" addr="irctc@icici" logo="image" url=""/>
            </VaeList>
      </Resp>
</upi:RespListVae>
```

Index	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

5.6.6 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" ts="" orgId="" msgId=""/>
            <VaeList>
                  <Vae op="UPDATE|REMOVE" name="LIC" addr="lic@hdfc" logo="image"
            url=””/>
                  <Vae op="UPDATE|REMOVE" name="IRCTC" addr=”irctc@icici”
            logo="image” url=""/>
            </VaeList>
</upi:ReqManageVae >
```

RespManageVae: Response Manage for Verified Address Entries

```
<upi:RespManageVae xmlns:upi="http://npci.org/upi/schema/">
      <Headver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
</upi:RespManageVae >

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

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

5.6.7 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" ts="" orgId="" msgId=""/>
    <Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
        <Info>
            <Identity type="PAN|AADHAAR|ACCOUNT" verifiedName="" />
            <Rating VerifiedAddress="TRUE|FALSE"/>
        </Info>
    </Payer>
    <Payee seqNum="" addr="" name=""/>
</upi:ReqValAdd>
```

RespValAdd: Validate Address Response

```
<upi:RespValAdd xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode="" maskName="">
</upi:RespValAdd>
```

Index	Message Item	<XML Tag>	Occurrence
28.1	Mask Name of the Beneficiary	maskName	1..1

5.6.8 Set Credentials

This API is required for providing a unified channel for setting and changing MPIN across various account providers. This is critical to ensure customers can easily set and change MPIN 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" ts="" orgId="" msgId=""/>
   <Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
       <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>
        <Creds>
            <Cred type="AADHAAR" subtype="IIR|FMR|FIR|OTP">
                <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>
        </Creds>
            <NewCred type="PIN" subtype="MPIN">
               <Data> base-64 encoded/encrypted authentication data</Data>
            </NewCred>
</upi:ReqSetCre>
```

Index	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" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
</upi:RespSetCre>
```

5.6.9 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" ts="" orgId="" msgId=""/>
   <Txn id="" note="" ref="" ts="" type=""/>
</upi:ReqChkTxn>
```

RespChkTxn: Response for check Txn Status

```
<upi:RespChkTxn xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts=""orgId="" msgId=""/>
      <Txnid="" note="" ref="" ts="" type=""/>
      <Resp reqMsgId=""result="SUCCESS|FAILURE|PARTIAL|DEEMED|PENDING" errCode=""/>
</upi:RespChkTxn>
```

5.6.10 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" ts="" orgId="" msgId=""/>
   <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=""/>
        </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>
   </Payer>
</upi:ReqOtp>
```

RespOtp: Response for OTP

```
<upi:RespOtp xmlns:upi="http://npci.org/upi/schema/">
      <Head ver="1.0" ts="" orgId="" msgId=""/>
      <Resp reqMsgId="" result="SUCCESS|FAILURE" errCode=""/>
</upi:RespOtp>
```

5.6.11 Balance-Enquiry

This API Allows PSP to Request for Balance enquiry for a user.

ReqBalEnq: Request for Balance Enquiry

```
<upi:ReqBalEnq xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="" orgId="" msgId=""/>
    <Txn id="" note="" ref="" ts="" type="">
        <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 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=""/>
        </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"/>
            <Detail name="CARDNUM" value=""/>
        </Ac>
        <Creds>
            <Cred type="AADHAAR" subtype="IIR|FMR|FIR|OTP">
                <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>
```

```
      </Payer>
</upi:ReqBalEnq>
```

RespBalEnq: Response for Balance Enquiry

```
<upi:RespBalEnq xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="" orgId="" msgId=""/>
    <Txn id="" note="" ref="" ts="" type="">
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="NPCI" type="TXNRISK" value=""/>
        </RiskScores>
    </Txn>
    <Payer addr="" name="" seqNum="" type="PERSON|ENTITY" code="">
             <Bal>
                    <Data> base-64 encoded/encrypted data</Data>
             </Bal>
    </Payer>
  </upi:RespBalEnq>
```

Index	Message Item	<XML Tag>	Occurrence
30.1	Data For Balance enquiry	<Bal>	1..1
30.2	Base 64 encoded authentication	<Bal.Data>	1..1

5.6.12 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" ts="" orgId="" msgId=""/>
      <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" ts=""orgId="" msgId=""/>
      <Resp reqMsgId=""result="SUCCESS|FAILURE" errCode=""/>
</upi:RespHbt>
```

Index	Message Item	<XML Tag>	Occurrence
31.1	Defines heartbeat messages	<HbtMsg>	1..1
31.1.1	Defines message type	< HbtMsg.type>	1..1
31.1.2	Details related to type	< HbtMsg.value>	1..1

5.6.13 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" ts="" orgId="" msgId=""/>
      <ReqMsg type="MOBILE|AADHAAR" value="" addr="" />
</upi:ReqHbt>
```

Index	Message Item	<XML Tag>	Occurrence
32.1	Defines Request Pending messages	<ReqMsg>	1..1
32.1.1	Defines message type	type	1..1
32.1.2	Details PSP address	addr	1..1

5.7 Elements and Attributes Definition

1.1 Element: Root

Definition: XML root element representing each API (ReqPay, RespPay, ReqAuthDetails, RespAuthDetails)

Presence: [1..1]

1.1.1 Attribute: xmlns

Presence: [1..1]

Definition: API Schema Namespace.

Data Type: Alphanumeric

Format: Min Length: 1

Max Length: 255

2.1 Element: <Head>

Presence: [1..1]

2.1.1 Attribute: ver

Presence: [1..1]

Definition: Version of the API

This is the API version. NPCI may host multiple versions for supporting gradual migration. As of this specification, default production version is "1.0".

Data Type: Alphanumeric

Format: Min Length: 1

Max Length: 6

2.1.2 Attribute: ts

Presence: [1..1]

Definition: Time of request from the creator of the message.

API request time stamp. Since timestamp plays a critical role, it is highly recommended that devices are time synchronized with a time server.

Data Type: ISODateTime

Format: Max Length: 25

YYYY-MM-DDThh:mm:ssZ+/-hh:mm

(eg 1997-07-16T19:20:30+05:30)

where;

YYYY = four-digit year

MM = two-digit month (01=January, etc.)

DD = two-digit day of month (01 through 31)

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)

Z +/- hh:mm = time zone designator (Z) followed by time zone difference from GMT in hours and minutes. THIS IS OPTIONAL. If not provided, it is assumed to be IST (+5.30).

2.1.3 Attribute: orgId

Presence: [1..1]

Definition: Organization id that created the message

Each organization will be identified with a unique ID. The member has to

request NPCI with a required organisation ID. Based on availability NPCI will register and assign the same.

Data Type: Alphanumeric

Format: Min Length: 1
Max Length: 20

2.1.4 Attribute: msgId

Presence: [1..1]

Definition: Message identifier-used to correlate between the request and response. The unique identifier created by the originator of the message and will be used to correlate the response with the original request.

Data Type: Alphanumeric

Format: Min Length: 1
Max Length: 35

3.1 Element: <Meta>

Presence: [0..1]

Definition: The data provided in the Meta element will be used for MIS and analysis purpose.

3.2 Element: <Meta.Tag>

Presence: [0..1]

Definition: The tag is defined in name value pairs to accommodate the MIS related parameters. The tag itself is optional and if the tag is present it is mandatory to have the two attributes with two codes mentioned below

3.2.1 Attribute: name

Presence: [1..n]

Definition: The name attribute will have the values as defined in the code table

Data Type: Code

Format: Min Length: 1
Max Length: 20

Code	Definition
PAYREQUESTSTART	The time at which the transaction was initiated in the device/medium
PAYREQUESTEND	The time at which the transaction was send out from

	the device/medium

3.2.2 Attribute: value

Presence: [1..n]
Definition: The data provided will have the details of transaction initiated time and end time in the device/medium
Data Type: ISODateTime
Format: Min Length: 1
Max Length: 255

4.1 Element: <Txn>

Presence: [1..1]
Definition: This element contains the Transaction details and is visible to all parties involved in the transaction processing. This element is populated by the originator of the transaction and the same must be passed across all the entities.

4.1.1 Attribute: id

Presence: [1..1]
Definition: Unique Identifier for the transaction across all entities.
This will be created by the originator. This will be used to identify each transaction uniquely across all the entities. PSP should use UUID scheme to ensure globally unique identifiers are used.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 35

4.1.2 Attribute: note

Presence: [1..1]
Definition: Description of the transaction which is in free text format (which will be printed on Pass book).
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 50

4.1.3 Attribute: refId

Presence: [1..1]
Definition: External reference number to identify the payment like Loan number, invoice number, etc.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 35

4.1.4 Attribute: refUrl

Presence: [1..1]
Definition: URL for the transaction
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 35

4.1.5 Attribute: ts

Presence: [1..1]
Definition: Transaction origination time by the creator of the transaction.
This same value to be passed across all the entities
Data Type: ISODateTime
Format: Min Length: 1
Max Length: 25

4.1.6 Attribute: type

Presence: [1..1]
Definition: This attribute describes the type of the transaction
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PAY	When a push transaction is initiated
COLLECT	When a pull transaction is initiated
REFUND	Refund transaction for original transaction

4.1.7 Attribute: orgTxnId

Presence: [1..1]
Definition: Original transaction ID when reversal/Refund has to be done
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 35

4.2 Element: <Txn.RiskScores>

Presence: [0..1]
Definition: This element defines the risk evaluation associated with the transaction and the interested parties in the transaction.

4.3 Element: <Txn.RiskScores.Score>

Presence: [0..n]

4.3.1 Attribute: provider

Presence: [1..n]
Definition: Entity providing the risk score.
This is the entity which evaluates the risk associated with the transaction.
Data Type: Code
Format: Min Length: 1
Max Length: 20

4.3.2 Attribute: type

Presence: [1..n]
Definition: This attribute describes the type of risk
Data Type: Code
Format: Min Length: 1
Max Length: 20

4.3.3 Attribute: value

Presence: [1..n]
Definition: Value of risk score ranging from 0 (No Risk) to 100 (Maximum Risk)
Data Type: Integer
Format: Min Length: 1
Max Length: 5

4.4 Element: <Txn.Rules>

Presence: [0..1]
Definition: This element defines the rules that govern the transaction

4.5 Element: <Txn.Rules.Rule>

Presence: [0..n]

4.5.1 Attribute: name

Presence: [1..n]
Definition: The name attribute will have the values as defined in the code table.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
EXPIREAFTER	The time at which the request should expire mainly in collect scenario. The value should be in minutes. It can be 1 minutes to max 64,800 minutes
MINAMOUNT	The minimum Amount that can be accepted mainly in collect scenario. In this case the requested amount and the paid amount would be different

4.5.2 Attribute: value

Presence: [1..n]
Definition: The values will be as defined for respective codes
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

5.1 Element: <Payer>

Presence: [1..1]

Definition: This element contains the complete details of the Payer.

5.1.1 Attribute: addr

Presence: [1..1]
Definition: Address of the Payer
Alias name with which the payer can be identified by his registered entity
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

5.1.2 Attribute: name

Presence: [1..1]
Definition: Name of the Payer
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

5.1.3 Attribute: seqNum

Presence: [1..1]
Definition: This attribute is the unique sub-identifier if there are multiple instructions in a single transaction.
Data Type: Numeric
Format: Min Length: 1
Max Length: 3
This should be defaulted to '1' for payer

5.1.4 Attribute: type

Presence: [1..1]
Definition: This attribute defines the type of the Payer
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PERSON	When the payer is a Person
ENTITY	When the payer is a Merchant/Entity

5.1.5 Attribute: code

Presence: [1..1]
Definition: Merchant Category Code –MCC
It is a 4 digit code describing a merchant's type of business. The value should be present as per the MCC code given in ISO 18245.
Data Type: Numeric
Format: Min Length: 1
Max Length: 4

5.2 Element: <Payer.Info>

Presence: [1..1]
Definition: This element contains Information related to the Payer

5.3 Element: <Payer.Info.Identity>

Presence: [1..1]
Definition: This element contains identity details of the Payer.

5.3.1 Attribute: type

Presence: [1..1]
Definition: Type of the identifier, this element contains the details of the identity that is used during the verification of the Payer.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PAN	PAN card number
UIDAI	Aadhaar Number
BANK	Bank Account Number

5.3.2 Attribute: verifiedName

Presence: [1..1]
Definition: This attribute provides the payer name as registered with the identifying authority as mentioned in 5.3.1

Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

5.4 Element: <Payer.Info.Rating>

Presence: [1..1]
Definition: This element contains the rating of the payer

5.4.1 Attribute: whiteListed

Presence: [1..1]
Definition: This attribute describes if the payer is whitelisted or not as per NPCI
Data Type: Code
Format: Min Length: 1
Max Length: 5

Code	Definition
TRUE	If the Payer is Whitelisted
FALSE	If the payer is not Whitelisted

5.5 Element: <Payer.Device>

Presence: [1..1]
Definition: This element contains the details of the device from which the transaction was initiated

5.6 Element: <Payer.Device.Tag>

Presence: [1..n]
Definition: This tag captures the device details in name value pair

5.6.1 Attribute: name

Presence: [1..n]
Definition: The name attribute will have the values as defined in the code table
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
MOBILE	Mobile Number of the payer
GEOCODE	Latitude and Longitude of the device
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	IP address of the device
TYPE	Type of the device
ID	Terminal ID of the device
OS	OS version of the device
APP	Application of the device
CAPABILITY	Terminal Capability (DE 61 of RuPay spec)

5.6.2 Attribute: value

Presence: [1..n]
Definition: The values will be as defined for respective codes
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

Code	Format	Example
MOBILE	91nnnnnnnnnn	919999999999
GEOCODE	nn.nnnn,nn.nnnn	12.9667,77.5667
LOCATION	Location, City, State Code, India Code	Sarjapur Road, Bangalore, KA, IN
IP	Valid IP address format(v4,v6)	123.456.123.123
TYPE	Min Length - 1 , Max Length - 20	Mobile
ID	Min Length - 1 , Max Length - 35	123456789

OS	Min Length - 1 , Max Length - 20	Android 4.4
APP	Min Length - 1 , Max Length - 20	CC 1.0
CAPABILITY	Min Length - 1 , Max Length - 999	011001

5.7 Element: <Payer.Ac>

Presence: [1..1]
Definition: This element contains the financial address details of the Payer

5.7.1 Attribute: addrType

Presence: [1..1]
Definition: This attribute describes the type of the financial address
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
AADHAAR	If the customer account will be identified using Aadhaar by the payer bank
ACCOUNT	If the customer account and IFSC is provided for identifying by the payer bank
MOBILE	If the customer account will be identified using Mobile by the payer bank
CARD	If the customer account will be identified using card by the payer bank

5.8 Element: <Payer.Ac.Detail>

Presence: [1..n]
Definition: This element contains the details related to Payer's financial address.

5.8.1 Attribute: name

Presence: [1..n]
Definition: The name attribute will have the values as defined in the code table. Only one of the payment details corresponding to the code given in 5.7.1 should

be provided.

Data Type: Code

Format: Min Length: 1
Max Length: 20

Code (addrType)	Code (name)	Definition
AADHAAR	IIN, UIDNUM	Provide for Aadhaar based payments
ACCOUNT	IFSC, ACTYPE, ACNUM	Provide for Account based payments
MOBILE	MMID, MOBNUM	Provide for Mobile based payments
CARD	ACTYPE, CARDNUM	Provide for Cards based payments

5.8.2 Attribute: value

Presence: [1..n]

Definition: The values will be as defined for respective codes.

Data Type: Alphanumeric

Format: Min Length: 1
Max Length: 255

5.9 Element: <Payer.Creds>

Presence: [1..1]

Definition: This element contains the information related to Payer Credentials.

5.10 Element: <Payer.Creds.Cred>

Presence: [1..1]

Definition: This element contains the Credential used to authenticate the request.

5.10.1 Attribute: type

Presence: [1..1]

Definition: The values will be as defined for respective codes. Only one of the payment credentials corresponding to the code given in 5.7.1 should be provided.

Data Type: Code

Format: Min Length: 1

Max Length: 20

5.10.2 Attribute: subtype

Presence: [1..1]
Definition: The values will be as defined for respective codes.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Type - Code	Subtype - Code
AADHAAR	IIR,FMR,FIR,OTP
OTP	SMS,EMAIL,HOTP,TOTP
PIN	MPIN
CARD	CVV1,CVV2,EMV

5.11 Element: <Payer.Creds.Cred.Data>

Presence: [1..1]
Definition: This element contains base-64 encoded/encrypted authentication data.

5.12 Element: <Payer.Amount>

Presence: [1..1]
Definition: This element contains the information related to the amounts in the transaction.

5.12.1 Attribute: value

Presence: [1..1]
Definition: The amount of transaction as per the currency given 5.12.2.
Data Type: Numeric
Format: fractionDigits: 5
minInclusive: 0
totalDigits: 18

5.12.2 Attribute: curr

Presence: [1..1]

Definition: This attribute describes the currency of the transaction.
Data Type: Text
Format: Min Length: 1
Max Length: 3

5.13 Element: <Payer.Amount.Split>

Presence: [0..1]
Definition: This element contains the detailed split of the amounts in the transaction.

5.13.1 Attribute: name

Presence: [0..n]
Definition: The name attribute will have the values as defined in the code table.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PURCHASE	Purchase amount
CASHBACK	Cash Back amount value if any
PARAMOUNT	If the transaction is done in partial

5.13.2 Attribute: value

Presence: [0..n]
Definition: The amount split as mentioned in 5.12.1
The currency of the amount mentioned here will be same as 5.12.2
Data Type: Numeric
Format: Min Length: 1
Max Length: 18

5.14 Element: <Payer.PreApproved>

Presence: [0..1]
Definition: This element contains information if the debit is already approved

5.14.1 Attribute: respCode

Presence: [1..1]
Definition: The response code of the Approval
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 3

5.14.2 Attribute: approvalRef

Presence: [1..1]
Definition: Approval Reference number of the debit done
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 6

6.1 Element: <Payees>

Presence: [1..1]
Definition: This element contains the complete details of the Payees.

6.2 Element: <Payee>

Presence: [1..n]
Definition: This element contains the complete details of the each Payee if there are multiple payees.

6.2.1 Attribute: addr

Presence: [1..1]
Definition: Address of the Payee
Alias name with which the payee can be identified by his registered entity
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

6.2.2 Attribute: name

Presence: [1..1]
Definition: Name of the Payee.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

6.2.3 Attribute: seqNum

Presence: [1..1]
Definition: This attribute is the unique sub-identifier if there are multiple instructions in a single transaction.
Data Type: Numeric
Format: Min Length: 1
Max Length: 3
This should be defaulted to '1' for payee

6.2.4 Attribute: type

Presence: [1..1]
Definition: This attribute defines the type of the Payee.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PERSON	When the payee is a Person
ENTITY	When the payee is a Merchant/Entity

6.2.5 Attribute: code

Presence: [1..1]
Definition: Merchant Category Code –MCC
It is a 4 digit code describing a merchant's type of business. The value should be present as per the MCC code given in ISO 18245.
Data Type: Numeric
Format: Min Length: 1
Max Length: 4

6.3 Element: <Payee.Info>

Presence: [1..1]
Definition: This element contains Information related to the Payee

6.4 Element: <Payee.Info.Identity>

Presence: [1..1]
Definition: This element contains identity details of the Payee.

6.4.1 Attribute: type

Presence: [1..1]
Definition: Type of the identifier, this element contains the details of the identity that is used during the verification of the Payee.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PAN	PAN card number
UIDAI	Aadhaar Number
BANK	Bank Account Number

6.4.2 Attribute: verifiedName

Presence: [1..1]
Definition: This attribute provides the payee name as registered with the identifying authority as mentioned in 5.3.1
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

6.5 Element: <Payee.Info.Rating>

Presence: [1..1]
Definition: This element contains the rating of the payee.

6.5.1 Attribute: whiteListed

Presence: [1..1]
Definition: This attribute describes if the payee is whitelisted or not as per NPCI.
Data Type: Code
Format: Min Length: 1
Max Length: 5

Code	Definition
TRUE	If the Payee is Whitelisted

FALSE	If the payee is not Whitelisted

6.6 Element: <Payee.Device>

Presence: [1..1]
Definition: This element contains the details of the device from which the transaction was initiated.

6.7 Element: <Payee.Device.Tag>

Presence: [1..n]
Definition: This tag captures the device details in name value pair.

6.7.1 Attribute: name

Presence: [1..n]
Definition: The name attribute will have the values as defined in the code table.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
MOBILE	Mobile Number of the payee
GEOCODE	Latitude and Longitude of the device
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	IP address of the device
TYPE	Type of the device
ID	Terminal ID of the device
OS	OS version of the device
APP	Application of the device
CAPABILITY	Terminal Capability (DE 61 of Rupay spec)

6.7.2 Attribute: value

Presence: [1..n]
Definition: The values will be as defined for respective codes.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

Code	Format	Example
MOBILE	91nnnnnnnnnn	919999999999
GEOCODE	nn.nnnn,nn.nnnn	12.9667,77.5667
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	Sarjapur Road, Bangalore, KA, IN
IP	Valid IP address format(v4,v6)	123.456.123.123
TYPE	Min Length - 1 , Max Length - 20	Mobile
ID	Min Length - 1 , Max Length - 35	123456789
OS	Min Length - 1 , Max Length - 20	Android 4.4
APP	Min Length - 1 , Max Length - 20	CC 1.0
CAPABILITY	Min Length - 1 , Max Length - 999	011001

6.8 Element: <Payee.Ac>

Presence: [1..1]
Definition: This element contains the financial address details of the Payee.

6.8.1 Attribute: addrType

Presence: [1..1]
Definition: This attribute describes the type of the financial address.
Data Type: Code
Format: Min Length: 1

Max Length: 20

Code	Definition
AADHAAR	If the customer account will be identified using Aadhaar by the payee bank
ACCOUNT	If the customer account and IFSC is provided for identifying by the payee bank
MOBILE	If the customer account will be identified using Mobile by the payee bank
CARD	If the customer account will be identified using card by the payee bank

6.9 Element: <Payee.Ac.Detail>

Presence: [1..n]
Definition: This element contains the details related to Payee's financial address.

6.9.1 Attribute: name

Presence: [1..n]
Definition: The name attribute will have the values as defined in the code table. Only one of the payment details corresponding to the code given in 5.7.1 should be provided.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code (addrType)	Code (name)	Definition
AADHAAR	IIN, UIDNUM	Provide for Aadhaar based payments
ACCOUNT	IFSC, ACTYPE, ACNUM	Provide for Account based payments
MOBILE	MMID, MOBNUM	Provide for Mobile based payments
CARD	ACTYPE, CARDNUM	Provide for Cards based payments

6.9.2 Attribute: value

Presence: [1..n]
Definition: The values will be as defined for respective codes.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

6.10 Element: <Payee.Amount>

Presence: [1..1]
Definition: This element contains the information related to the amounts in the transaction.

6.10.1 Attribute: value

Presence: [1..1]
Definition: The amount of transaction as per the currency given 5.12.2.
Data Type: Numeric
Format: fractionDigits: 5
minInclusive: 0
totalDigits: 18

6.10.2 Attribute: curr

Presence: [1..1]
Definition: This attribute describes the currency of the transaction.
Data Type: Text
Format: Min Length: 1
Max Length: 3

6.11 Element: <Payee.Amount.Split>

Presence: [0..1]
Definition: This element contains the detailed split of the amounts in the transaction.

6.11.1 Attribute: name

Presence: [0..n]
Definition: The name attribute will have the values as defined in the code table.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
PURCHASE	Purchase amount
CASHBACK	Cash Back amount value if any
PARAMOUNT	If the transaction is done in partial

6.11.2 Attribute: value

Presence: [0..n]
Definition: The amount split as mentioned in 5.12.1.
The currency of the amount mentioned here will be same as 5.12.2.
Data Type: Numeric
Format: Min Length: 1
Max Length: 18

11.1 Element: <Resp>

Presence: [1..1]
Definition: This element contains the response details of the transaction.

11.1.1 Attribute: reqMsgID

Presence: [1..1]
Definition: This attribute contains the message identifier of the original request. This is used to match the request and the response.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 35

11.1.2 Attribute: result

Presence: [1..1]
Definition: This attribute contains the final result of the transaction.
Data Type: Code
Format: Min Length: 1
Max Length: 20
Code: SUCCESS, FAILURE, PARTIAL, DEEMED

11.1.3 Attribute: errCode

Presence: [0..n]

Definition: The error code for the result given above in 11.1.2, the error code defines the exact reason for the failure.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 3

11.2 Element: <Ref>

Presence: [1..n]
Definition: This element contains the reference details for every account holder's (Payer and Payees) within the transaction.

11.2.1 Attribute: type

Presence: [1..1]
Definition: This attribute contains the type of the account holder about whom the details are provided. The name attribute will have the values as defined in the code table below.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 20

Code	Definition
PAYER	The account holder is PAYER
PAYEE	The account holder is PAYEE

11.2.2 Attribute: seqNum

Presence: [1..1]
Definition: This attribute contains the sequence number for the payee/payer record.
Data Type: Numeric
Format: Min Length: 1
Max Length: 3

11.2.3 Attribute: addr

Presence: [1..1]

Definition: Address of the Payee.
Alias name with which the payee can be identified by his registered entity.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

11.2.4 Attribute: settAmount

Presence: [1..1]
Definition: This attribute contains the final settlement amount.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 18

11.2.5 Attribute: settCurrency

Presence: [1..1]
Definition: This attribute contains the final settlement currency.
Data Type: Text
Format: Min Length: 1
Max Length: 3

11.2.6 Attribute: approvalNum

Presence: [1..1]
Definition: The attribute contains the approval reference number generated by the authorising system.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 6

11.2.7 Attribute: respCode

Presence: [1..1]
Definition: The attribute contains the appropriate response code defining the result.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 3

21.1 Element: <PspList>

Presence: [1..1]
Definition: The Element contains the list of PSP details.

21.2 Element: <PspList.Psp>

Presence: [1..1]
Definition: The Element contains the details of the Payment System Player.

21.2.1 Attribute: name

Presence: [1..1]
Definition: The attribute contains the name of the PSP.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

21.2.2 Attribute: codes

Presence: [1..n]
Definition: The attribute contains the payment codes defined for the PSP.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 55

21.2.3 Attribute: active

Presence: [1..1]
Definition: The attribute contains the status of the PSP, either Active or Not. This can be defined by one character either Y or N
Data Type: Alphanumeric
Format: Max Length: 1

Code	Definition
Y	If the status of the PSP is Active
N	If the status of the PSP is Inactive

21.2.4 Attribute: url

Presence: [0..n]
Definition: The attribute contains the URL link provided by the PSP.

Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

21.2.5 Attribute: spocName

Presence: [0..n]
Definition: The attribute contains the name of the SPOC for the PSPs.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

21.2.6 Attribute: spocEmail

Presence: [0..n]
Definition: The attribute contains the E-mail of the SPOC for the PSPs.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

21.2.7 Attribute: spocPhone

Presence: [0..n]
Definition: The attribute contains the contact details i.e., phone number of the SPOC for the PSPs.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 13

21.2.8 Attribute: lastModifiedTs

Presence: [1..1]
Definition: The attribute contains the last modified date of the PSP information in the system.
Data Type: ISODateTime
Format: Max Length: 25

22.1 Element: <AccPvdList>

Presence: [1..1]
Definition: The Element contains the Account Providers List.

22.2 Element: <AccPvdList.AccPvd>

Presence: [1..1]
Definition: The Element contains the details of the registered account providers list.

22.2.1 Attribute: name

Presence: [1..1]
Definition: The attribute contains the name of the account provider.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

22.2.2 Attribute: iin

Presence: [1..n]
Definition: The attribute contains the IIN Number of the Account providers.
Data Type: Numeric
Format: Min Length: 1
Max Length: 6

22.2.3 Attribute: active

Presence: [1..1]
Definition: The attribute contains the status of the Account Provider, either Active or Not. This can be defined by one character either Y or N
Data Type: AlphaNumeric
Format: Max Length : 1

Code	Definition
Y	If the status of the Account provider is Active
N	If the status of the Account Provider is Inactive

22.2.4 Attribute: url

Presence: [0..n]

Definition: The attribute contains the URL Link provided by the Account provider.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

22.2.5 Attribute: spocName

Presence: [0..n]
Definition: The attribute contains the name of the SPOC for the Account provider.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

22.2.6 Attribute: spocEmail

Presence: [0..n]
Definition: The attribute contains the E-mail of the SPOC for the Account provider.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

22.2.7 Attribute: spocPhone

Presence: [0..n]
Definition: The attribute contains the contact details i.e., phone number of the SPOC for the Account provider.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 13

22.2.8 Attribute: prods

Presence: [0..n]
Definition: The attribute contains the list of NPCI products for which account provider is Live.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

21.2.9 Attribute: lastModifiedTs

Presence: [1..1]
Definition: The attribute contains the last modified date of the Account Provider information in the system.
Data Type: ISODateTime
Format: Max Length: 25

23.1 Element: <KeyList>

Presence: [1..1]
Definition: The Element contains the list of the public keys of the Account providers list

23.2 Element: <KeyList.Key>

Presence: [1..1]
Definition: The Element contains the details related to public keys.

23.2.1 Attribute: code

Presence: [1..1]
Definition: The attribute contains the account provider code.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

23.2.2 Attribute: owner

Presence: [1..1]
Definition: The attribute contains the details of the Owner of the Key.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

23.2.3 Attribute: type

Presence: [1..1]
Definition: The attribute contains the type of key.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 10

23.2.4 Attribute: ki

Presence: [1..1]
Definition: The attribute contains the key index Date.
Data Type: Numeric
Format: Min Length: 1
Max Length: 8

(yyyymmdd)

23.3 Element: <KeyList.Key.KeyValue>

Presence: [1..1]
Definition: The element contains the Base64 encoded certificate.

24.1 Element: <Link>

Presence: [1..1]
Definition: The element contains the Linked account list.

24.1.2 Attribute: type

Presence: [1..1]
Definition: The attribute contains the account linkage to Mobile or the Aadhaar Number
Data Type: Code
Format: Min Length: 1
Max Length: 10

Code	Definition
MOBILE	If the list has to be looked up by Mobile number
AADHAAR	If the list has to be looked up by Aadhaar number

24.1.3 Attribute: value

Presence: [1..1]
Definition: The attribute contains the Mobile or the Aadhaar Number
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 15

25.1 Element: <AccountList>

Presence: [1..1]
Definition: The element contains the accounts list.

25.2 Element: <AccountList.Account>

Presence: [1..n]
Definition: The element contains the details related to accounts.

25.2.1 Attribute: number

Presence: [1..1]
Definition: The attribute contains the Account Number.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 35

25.2.2 Attribute: ifsc

Presence: [1..1]
Definition: The attribute contains the IFSC code of the Account.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 11

25.2.3 Attribute: mmid

Presence: [1..1]
Definition: The attribute contains the MMID linked to mobile.
Data Type: Numeric
Format: Min Length: 1
Max Length: 7

25.2.4 Attribute: name

Presence: [1..1]
Definition: The attribute contains the name of the account holder.
Data Type: Alphanumeric

Format: Min Length: 1
Max Length: 99

25.2.5 Attribute: aeba

Presence: [1..1]
Definition: The attribute contains the information if the account is an Aadhaar Enabled Bank Account or Not
Data Type: Alphanumeric
Format: Max Length: 1

Code	Definition
Y	If the account is linked to Aadhaar
N	If the account is not linked to Aadhaar

26.1 Element: <VaeList>

Presence: [1..1]
Definition: The element contains the list of Verified Address Entries.

26.2 Element: <VaeList.Vae>

Presence: [1..1]
Definition: The element contains the details related to list of verified address entries.

26.2.1 Attribute: name

Presence: [1..1]
Definition: The attribute contains the name of the Merchant.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

26.2.2 Attribute: addr

Presence: [1..1]
Definition: The attribute contains the payment address of the Merchant.
Data Type: Alphanumeric
Format: Min Length: 1

Max Length: 255

26.2.3 Attribute: logo

Presence: [1..n]
Definition: The attribute contains the logo Image of the Merchant.
Data Type: Image
Format: Min Length: 1
Max Length: 255

26.2.4 Attribute: url

Presence: [1..n]
Definition: The attribute contains the URL link provided by Merchant.
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

27.1 Attribute: op

Presence: [1..1]
Definition: The attribute contains the option to update or Remove.
Data Type: Code
Format: Min Length: 1
Max Length: 10

Code	Definition
UPDATE	To create or update an address details
REMOVE	To remove an address details

28.1 Attribute: maskName

Presence: [1..1]
Definition: The attribute contains the Mask name of the beneficiary .
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 99

29.1 Element: <NewCred>

Presence: [1..1]
Definition: The element contains the new credentials for Authentication.

29.1.1 Attribute: type

Presence: [1..1]
Definition: The attribute contains the credential used to Authenticate the request.
Data Type: Code
Format: Min Length: 1
Max Length: 10

Code	Definition
PIN	PIN is the type of the credential used

29.1.2 Attribute: subtype

Presence: [1..1]
Definition: The attribute contains the sub type of financial instrument used for Authentication.
Data Type: Code
Format: Min Length: 1
Max Length: 10

Code	Definition
MPIN	MPIN is the subtype of the credential used

29.2 Element: <Data>

Presence: [1..1]
Definition: The element contains the base64 encoded authentication.

30.1 Element: <Bal>

Presence: [1..1]
Definition: The element contains the data for balance Enquiry.

30.2 Element: <Bal.Data>

Presence: [1..1]
Definition: The element contains the base64 encoded authentication.

31.1 Element: <HbtMsg>

Presence: [1..1]
Definition: The element contains the details of the Heart Beat Message.

31.1.1 Attribute: type

Presence: [1..1]
Definition: The attribute contains the type of the Heart beat Request.
Data Type: Code
Format: Min Length: 1
Max Length: 10

Code	Definition
EOD	Type of message is EndOfDay
ALIVE	Type of message is Alive

31.1.2 Attribute: value

Presence: [1..1]
Definition: The attribute contains the value related to Heartbeat request.
Data Type: Code
Format: Min Length: 1
Max Length: 10

Code	Definition
Date(yyyymmdd)	Date is mentioned when the message type is EOD.

32.1 Element: <ReqMsg>

Presence: [1..1]
Definition: The element contains the details of the Heart Beat Message.

32.1.1 Attribute: type

Presence: [1..1]
Definition: Type of the identifier for which pending message list is requested.
Data Type: Code
Format: Min Length: 1
Max Length: 20

Code	Definition
MOBILE	Mobile number
AADHAAR	Aadhaar Number

32.1.2 Attribute: addr

Presence: [1..1]
Definition: Details PSP address
Data Type: Alphanumeric
Format: Min Length: 1
Max Length: 255

5.8 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.

5.8.1.1 Scenario 1 - Direct Pay

Ram wants to send money to his wife Laxmi. Ram has a mobile enabled account with SBI, 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 MPIN to authenticate himself, and approve the transaction.

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

```
<upi:ReqPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-16T14:15:43+05:30" orgId="sbi" msgId="1"/>
```

```
    <Meta>
        <Tag name="PAYREQSTART" value="2015-01-16T14:15:35+05:30"/>
        <Tag name="PAYREQEND" value="2015-01-16T14:15:42+05:30"/>
    </Meta>
    <Txn id="8ENSVVR4QOS7X1UGPY7JGUV444PL9T2C3QM"
         note="Sending money for your use"
         ts="2015-01-16T14:15:42+05:30" type="PAY">
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="npci" type="TXNRISK" value=""/>
        </RiskScores>
    </Txn>
    <Payer addr="ram@sbi" name="Ram" seqNum="1" type="PERSON">
        <Info>
            <Identity type="UIDAI" verifiedName="Ram" />
        </Info>
        <Device>
                <Tag name="MOBILE" value="+91.12345.67890"/>
                <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="ID" value="123456789"/>
                <Tag name="OS" value="Android 4.4"/>
                <Tag name="APP" value="CC 1.0"/>
                <Tag name="CAPABILITY" value="011001"/>
            </Device>
        <Ac addrType="MOBILE">
            <Detail name="MMID" value="SBIN0012024"/>
            <Detail name="MOBNUM" value="+91.12345.67890"/>
        </Ac>
        <Creds>
            <Cred type="PIN" subtype="MPIN">
                <Data>…</Data>
            </Cred>
        </Creds>
        <Amount value="5000" curr="INR"/>
    </Payer>
    <Payees>
        <Payee addr="laxmi1987@boi" name="Laxmi" seqNum="2" type="PERSON">
            <Amount value="5000" curr="INR"/>
        </Payee>
    </Payees>
</upi: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).

```
<upi:ReqAuthDetails xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-16T14:15:44+05:30" orgId="NPCI" msgId="1"/>
    <Txn id="8ENSVVR4QOS7X1UGPY7JGUV444PL9T2C3QM"
         note="Sending money for your use"
         ts="2015-01-16T14:15:42+05:30" type="PAY">
        <RiskScores>
            <Score provider="sbi" type="TXNRISK" value="0"/>
            <Score provider="NPCI" type="TXNRISK" value="0"/>
        </RiskScores>
    </Txn>
    <Payer addr="ram@sbi" name="Ram" seqNum="1" type="PERSON">
        <Info>
            <Identity type="UIDAI" verifiedName="Ram" />
        </Info>
    </Payer>
    <Payees>
        <Payee addr="laxmi1987@boi" name="Laxmi" seqNum="2" type="PERSON">
            <Amount value="5000" curr="INR"/>
        </Payee>
    </Payees>
</upi: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.

```
<upi:RespAuthDetails xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-16T14:15:45+05:30" orgId="boi" msgId="1"/>
    <Resp reqMsgId="1" result="SUCCESS" />
    <Payees>
        <Payee addr="laxmi1987@boi" name="Laxmi" seqNum="2" type="PERSON">
            <Info>
                <Identity type="UIDAI" verifiedName="Laxmi" />
            </Info>
            <Ac addrType="AADHAAR">
                <Detail name="IIN" value="508505"/>
                <Detail name="UIDNUM" value="123456789012"/>
            </Ac>
            <Amount value="5000" curr="INR"/>
        </Payee>
    </Payees>
</upi:RespAuthDetails>
```

NPCI can now complete the transaction, and sends a response to the 2 service providers, indicating that the transaction was successful. This is the response sent to SBI, who initiated the transaction.

```
<upi:RespPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-16T14:15:47+05:30" orgId="npci" msgId="2"/>
    <Txn id="8ENSVVR4QOS7X1UGPY7JGUV444PL9T2C3QM"
         note="Sending money for your use"
         ts="2015-01-16T14:15:42+05:30" type="PAY">
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="npci" type="TXNRISK" value=""/>
        </RiskScores>
    <!-- Txn is echoed back from the original transaction request -->
    <Resp reqMsgId="1" result="SUCCESS" approvalNum="3MKBVB">
    <!-- For the requester, reqMsgId is the msgId of the message used to initiate
         the transaction, else it is blank (or not present) -->
    <!-- For the requester, all settlement information is available, so there
         will be 1 Ref per successful payer, and payee -->
         <Ref type="PAYER" seqNum="1" addr="ram@sbi"
             settAmount="5000" approvalNum="AWHWU9" />
         <Ref type="PAYEE" seqNum="2" addr="laxmi1987@boi"
             settAmount="5000" approvalNum="ESOP61" />
    </Resp>
</upi:RespPay>
```

This is the confirmation sent to BOI.

```
<upi:RespPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-16T14:15:47+05:30" orgId="npci" msgId="3"/>
    <Txn id="8ENSVVR4QOS7X1UGPY7JGUV444PL9T2C3QM"
         note="Sending money for your use"
         ts="2015-01-16T14:15:42+05:30" type="PAY">
    <!-- Txn is echoed back from the original transaction request -->
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="npci" type="TXNRISK" value=""/>
        </RiskScores>
    <Resp result="SUCCESS" approvalNum="3MKBVB">
    <!-- For the requester, reqMsgId is the msgId of the message used to
         initiate the transaction, else it is blank (or not present) -->
         <Ref type="PAYEE" seqNum="2" addr="laxmi1987@boi"
             settAmount="5000" approvalNum="ESOP61" />
    </Resp>
</upi:RespPay>
```

5.8.1.2 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 Punjab National Bank, and Shyam with ICICI. Ram uses his mobile phone, and initiates a request to get money from Shyam.

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

```
<upi:ReqPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-17T20:23:03+05:30" orgId="pnb" msgId="1"/>
    <Meta>
        <Tag name="PAYREQSTART" value="2015-01-17T20:22:58+05:30"/>
        <Tag name="PAYREQEND" value="2015-01-17T20:23:02+05:30"/>
    </Meta>
    <Txn id="7KGEYCTNLBOECLO70F9ZGY5FOTQRKDKZ5RL"
         note="Your portion of the dinner bill"
         ts="2015-01-17T20:23:02+05:30" type="COLLECT">
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="npci" type="TXNRISK" value=""/>
        </RiskScores>
        <Rules>
            <Rule name="EXPIREAFTER" value="10080"/>
            <!-- Payment request will expire in 7 days (7*24*60 minutes) -->
        </Rules>
    </Txn>
    <Payees>
        <Payee addr="ram@pnb" name="Ram" seqNum="1" type="PERSON">
            <Info>
                <Identity type="UIDAI" verifiedName="Ram" />
            </Info>
            <Device>
                <Tag name="MOBILE" value="+91.12345.67890"/>
                <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="ID" value="123456789"/>
                <Tag name="OS" value="Android 4.4"/>
                <Tag name="APP" value="CC 1.0"/>
                <Tag name="CAPABILITY" value="011001"/>
            </Device>
            <Ac addrType="MOBILE">
                <Detail name="MMID" value="PNBN0012024"/>
                <Detail name="MOBNUM" value="+91.12345.67890"/>
            </Ac>
            <Amount value="200" curr="INR"/>
        </Payee>
    </Payees>
    <Payer addr="shyam.444@icici" name="Shyam" seqNum="2" type="PERSON">
        <Amount value="200" curr="INR"/>
```

```
    </Payer>
</upi:ReqPay>
```

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

```
<upi:ReqAuthDetails xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-17T20:23:04+05:30" orgId="NPCI" msgId="1"/>
    <Txn id="7KGEYCTNLBOECLO70F9ZGY5FOTQRKDKZ5RL"
         note="Your portion of the dinner bill"
         ts="2015-01-17T20:23:02+05:30" type="COLLECT">
        <RiskScores>
            <Score provider="pnb" type="TXNRISK" value="0"/>
            <Score provider="NPCI" type="TXNRISK" value="5"/>
        </RiskScores>
        <Rules>
            <Rule name="EXPIREAFTER" value="10080"/>
            <!-- Payment request will expire in 7 days (7*24*60 minutes) -->
        </Rules>
    </Txn>
    <Payees>
        <Payee addr="ram@pnb" name="Ram" seqNum="1" type="PERSON">
            <Info>
                <Identity type="UIDAI" verifiedName="Ram" />
            </Info>
        </Payee>
    </Payees>
    <Payer addr="shyam.444@icici" name="Shyam" seqNum="2" type="PERSON">
        <Amount value="200" curr="INR"/>
    </Payer>
</upi:ReqAuthDetails>
```

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

```
<upi:RespAuthDetails xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-17T20:23:35+05:30" orgId="icici" msgId="1"/>
    <Resp reqMsgId="1" result="SUCCESS" />
    <Payer addr="shyam.444@icici" name="Shyam" seqNum="2" type="PERSON">
        <Info>
            <Identity type="UIDAI" verifiedName="Shyam" />
        </Info>
        <Ac addrType="AADHAAR">
            <Detail name="IIN" value="508534"/>
            <Detail name="UIDNUM" value="123456789012"/>
        </Ac>
```

```
        <Creds>
            <Cred type="AADHAAR" subtype="IIR">
                <Data>…</Data>
            </Cred>
        </Creds>
        <Amount value="200" curr="INR"/>
        <Device>
            <Tag name="MOBILE" value="+91.67890.12345"/>
            <Tag name="LOCATION" value="Sarjapur Road, Bangalore, KA, IN" />
            <Tag name="ID" value="123456789"/>
            <Tag name="OS" value="Android 4.4"/>
            <Tag name="APP" value="CC 1.0"/>
            <Tag name="CAPABILITY" value="011001"/>
        </Device>
    </Payer>
</upi:RespAuthDetails>
```

NPCI can now complete the transaction, and sends a response to the 2 service providers, indicating that the transaction was successful. This is the response sent to PNB, who initiated the transaction.

```
<upi:RespPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-17T20:23:37+05:30" orgId="npci" msgId="2"/>
    <Txn id="7KGEYCTNLBOECLO70F9ZGY5FOTQRKDKZ5RL"
         note="Your portion of the dinner bill"
         ts="2015-01-17T20:23:02+05:30" type="COLLECT">
        <RiskScores>
            <Score provider="sp" type="TXNRISK" value=""/>
            <Score provider="npci" type="TXNRISK" value=""/>
        </RiskScores>
    <!-- Txn is echoed back from the original transaction request -->
    <Resp reqMsgId="1" result="SUCCESS" approvalNum="XTROX1">
    <!-- For the requester, reqMsgId is the msgId of the message used to initiate
         the transaction, else it is blank (or not present) -->
    <!-- For the requester, all settlement information is available, so there
         will be 1 Ref per successful payer, and payee -->
         <Ref type="PAYEE" seqNum="1" addr="ram@pnb"
             settAmount="200" approvalNum="T0VKVN" />
         <Ref type="PAYER" seqNum="2" addr="shyam.444@icici"
             settAmount="200" approvalNum="LZEQ8L" />
    </Resp>
</upi:RespPay>
```

This is the confirmation sent to icici.

```
<upi:RespPay xmlns:upi="http://npci.org/upi/schema/">
    <Head ver="1.0" ts="2015-01-17T20:23:37+05:30" orgId="npci" msgId="3"/>
    <Txn id="7KGEYCTNLBOECLO70F9ZGY5FOTQRKDKZ5RL"
```

```
        note="Your portion of the dinner bill"
        ts="2015-01-17T20:23:02+05:30" type="COLLECT">
       <RiskScores>
          <Score provider="sp" type="TXNRISK" value=""/>
          <Score provider="npci" type="TXNRISK" value=""/>
       </RiskScores>
    <!-- Txn is echoed back from the original transaction request -->
    <Resp result="SUCCESS" approvalNum="XTROX1">
    <!-- For the requester, reqMsgId is the msgId of the message used to
        initiate the transaction, else it is blank (or not present) -->
        <Ref type="PAYER" seqNum="2" addr="shyam.444@icici"
            settAmount="200" approvalNum="LZEQ8L" />
    </Resp>
</upi:RespPay>
```

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