UNIFIED PAYMENT INTERFACE

API and Technology Specifications

Specifications – Version 1.0 (DRAFT)

(Please send all feedback on this draft specifications to upisupport@npci.org.in)

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 who 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.

1. Introduction

Over decades, India has made slow but steady progress in the field of electronic payments. The innovations in payments have leveraged major technological innovations in each era. However, given the scale of our country, and that so many are unbanked, we cannot rest on our laurels.

> This Unified Payment Interface 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 0.6 million.

3. Of the 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.

It was against this background, 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 drivers are:

- **Simplicity** - Paying and receiving payments should be as easy as swiping a phone book entry and making a call on mobile phone. Everyone who has an account should be able to 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, RuPay Card, virtual payment address, etc.) with a single click.
- **Innovation** - Solution should be minimal, functional, and layerable so that innovations on both payee and payer side can evolve without having to change the whole interface. This unified layer should allow application providers to take advantage of enhancements in mobile devices, provide integrated payments on new consumer devices, provide innovative user interface features, take advantage of newer authentication services, etc.
- **Adoption** - Solution should be scalable to a billion users and large scale adoption. This should allow gradual adoption across smartphone and feature phone users and provide full interoperability across all payment players, phones, and use cases. People using smartphone should be able to send money to others who are not yet using any mobile application and vice versa. Similarly, it should allow full interoperability between multiple identifiers such as Aadhaar number, mobile number, and new virtual payment addresses.
- **Security** - Solution should provide end to end strong security and data protection. Considering self-service mobile applications, data capture must be strongly encrypted at capture. Similarly, solution should allow a mechanism to pay and collect using true virtual addresses without having to reveal any bank/account details. While providing convenient, solution should offer 1-click 2-factor authentication, protection from phishing, risk scoring, 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, solution should offer a mechanism to take full advantage of that. Use of mobile phone as the authentication (credential capture) device, use of virtual payment addresses, and use of 3rd party portable authentication schemes such as Aadhaar should allow both acquiring side and issuing side cost to be driven down. This allows banks and other payment players to focus on core business and allow half a billion phones to be the primary payment device in conjunction with other 3rd party authentication.

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.

Virtual payment addresses provide innovative mechanisms for customers to create addresses with attached rules for limiting amount, time (e.g., one time use addresses), and payees.

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 aspects of the Unified Payment Interface.

1. The Unified Payment Interface 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. Virtual payment addresses, 1-click 2-factor authentication, Aadhaar integration, use of payer's smartphone for secure credential capture, etc. are some of the core features.
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.
7. Proposed architecture is well within the regulatory framework of the mobile and ecommerce transactions having 2 factors of authentication (2FA).

1.2 Industry Trends

This section looks at some of the largest trends in the world and India in particular that has impact in the way financial transactions are conducted. It is important to design new systems so that it is fully aligned to take advantage of these sweeping trends.

1.2.1 Mobile Adoption

One of the most transformational technologies that completely changed the face of India is the massive adoption of mobile phones. Desktop usage, especially in India, was very low and nearly stagnated without mass scale adoption. From nearly no phone access, Indians went straight into using mobile phones in a massive way. A combination of regulatory and open market approach allowed many companies to compete and provide best value to end customers. Low cost phones, affordable tariffs, and a massive distribution network to handle pre-paid plans and recharges allowed an explosion of user base to a billion people.

Desktops, quite like landlines, never penetrated into daily lives of masses due to their price, complexity, interaction model, and its inability to be mobile. Whereas smart phones, with its affordable prices, simplicity, easier touch based interaction, and mobility has caught the imagination of masses. Smartphone adoption in India is exploding at a rapid pace and is expected to become the de-facto computing device for millions of people.

Considering the fact that smartphones are available at sub $100 levels, it is expected that smartphone penetration, currently about 150 million, will reach 500 million within next 4-5 years. It is expected that biometric (Iris in particular due to extreme low cost, ease of integration with mobile camera, and high authentication accuracy) enabled smartphones will also be commonplace in near future.

1.2.2 Ubiquitous Connectivity

Mobile connectivity revolutionized Indian communication landscape from a very limited landline penetration to nearly a billion mobile connections across the country. Mobile networks are now used for all kinds of communication in payment, entertainment, and other verticals and use of SMS as an effective two way communication scheme. While traditional broadband and Internet use continued to grow very slowly over last several years, in 2012, Indian mobile Internet usage exceeded all other channels.

With nearly all of India now covered by mobile networks, Telcos continue to aggressively expand their 2G and 3G coverage across country, and with really cheap tariffs, being online and connected is now taken for granted. Several applications that are being built now take advantage of this pervasive connectivity and allow users to access massive amount of information online and share with other users.

1.2.3 Aadhaar & Digital Identity

Unique Identification Authority of India (UIDAI) has the vision of empowering every resident of India with a unique identity and providing a digital platform to authenticate anytime anywhere. The purpose of the UIDAI is to issue a Unique Identification number (Aadhaar number) to every Indian resident that is (a) robust enough to eliminate duplicate and fake identities, and (b) can be verified and authenticated in an easy, electronic, cost-effective way.

> The Aadhaar system is built on a sound strategy and a strong technology backbone and has evolved into a vital digital identity infrastructure. It is built purely as an "***Identity Platform***" that other applications, Government and private, can take advantage of using a set of open APIs. It reached the kind of scale that was never achieved in any biometric identity system in the world. Currently with more than 730 million Aadhaar holders, it is expected to cross the 1 billion mark in 2015.

1.2.4 NPCI and Payment Backbone

The RBI, after setting up of the Board for Payment and Settlement Systems in 2005, released a vision document incorporating a proposal to set up an umbrella institution for all the Retail Payment System in the country. The core objective was 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 benefit the common man across the country and help financial inclusion.

Reserve Bank of India also issued authorisation to NPCI to take over the operations of National Financial Switch (NFS) from the Institute of Development and Research in Banking Technology (IDRBT) on “as is where” basis on October 15, 2009. NPCI took over the NFS operations from IDRBT in December 2009. Over the last five years the NFS footprint has had an exponential growth with transaction volumes increasing four fold. In the last 5 years, NPCI went on to launch and operationalize several innovative products such as RuPay, IMPS, APB, and AEPS on the existing platform.

IMPS facilitate customers to use mobile instruments as a channel for accessing their bank accounts and put high interbank fund transfers in a secured manner with immediate confirmation features.

Immediate Payment Service (IMPS), launched on 22nd November 2010, is now available to the Indian public from over 65 banks. This was an important big step towards peer to peer immediate payments in India. IMPS offers an instant, 24X7, interbank immediate electronic fund transfer service through multiple channels such as Internet banking, mobile banking, etc.

Aadhaar Payment Bridge (APB) is a batch processing system which allows Government and non-Government entities to send money to an Aadhaar number, and Aadhaar Enabled Payment Systems (AEPS) is an online system which allows immediate payment from one Aadhaar linked account to another. These were critical steps towards Aadhaar based payments. AEPS allows Aadhaar holders to transfer funds to anywhere in India across banks just by using their Aadhaar number and biometrics (card-less and PIN-less). Several 1000's of Micro-ATMs (handheld machines with biometric sensors operated by BCs) use AEPS system to provide cash withdrawal and fund transfer facilities in urban and rural parts of India.

These products, introduced based on the recent market trends and technology developments, are built as silos and offers very limited interoperability between the payment instruments like Card, Mobile number, and Aadhaar number. Current schemes do not offer any mechanism to use "*virtual payment addresses*" that can be used for various electronic transactions in an interoperable way across all banks and regulated players. Also there is no unified layer that makes mobile applications (banking, wallet, etc.) to seamlessly integrate with these systems using a standard set of APIs.

1.2.5 Regulatory Support

Over the years RBI has taken several steps towards the vision of less cash economy and universal banking. Reserve Bank has taken several policy initiatives to address this situation to facilitate access for the common man to the banking system. In the Annual Policy Statement of the Reserve Bank for the year 2005-06, it was stated that the Reserve Bank will implement policies to encourage banks, which provide extensive services while disincentivising those, which are not responsive to the banking needs of the community, including the underprivileged. In the broader perspective, Reserve Bank aims at 'connecting' people with the banking system.

With the objective of ensuring greater financial inclusion and increasing outreach of the banking sector, Reserve Bank, in January 2006 permitted banks to use intermediaries as Business Facilitators (BF) or Business Correspondents (BC) for providing financial and banking services. The BCs were allowed to conduct banking business as agents of the banks at places other than the bank premises.

RBI has approved and supported the use of common Aadhaar based biometric authentication, use of standardized Micro-ATM device, token less (using Aadhaar and biometrics) banking, and use of such devices and interoperable transactions across BC network. This has allowed lakhs of Aadhaar holders to open account via Aadhaar e-KYC on a BC/Agent location, transact via biometric enabled terminals, and receive subsidies just using Aadhaar number.

RBI, within its "*Report of the Technical Committee on Mobile Banking*" has clearly stated the need for aggressively adopting mobile banking, allowing common mobile application and USSD/SMS channels to make adoption easier, and allow credential setting via common electronic means including using biometric terminals.

> *"The developments in mobile telephony, as also the mobile phone density in the country, with over 870 million subscribers, presents a unique opportunity to leverage the mobile platform to meet the objectives and challenges of financial inclusion. By harnessing the potential of mobile technology, large sections of the un-banked and under-banked society can be empowered to become inclusive through the use of electronic banking services."*
>
> - Report of the Technical Committee on Mobile Banking, RBI, Feb 2014

RBI's initiative that allows new payment players such as payments banks, small banks, and wallets/PPIs is primarily focussed on universal financial inclusion and enable ease of payments. This is a significant step towards universal banking and will further boost ability for millions of people to be included in the formal system.

1.2.6 Jan Dhan Yojana

Objective of "*Pradhan Mantri Jan-Dhan Yojana*" (PMJDY) is ensuring access to various financial services like availability of basic savings bank account, access to need based credit, remittances facility, insurance and pension to the excluded sections i.e. weaker sections & low income groups. This deep penetration at affordable cost is possible only with effective use of technology.

PMJDY is a National Mission on Financial Inclusion encompassing an integrated approach to bring about comprehensive financial inclusion of all the households in the country. The plan envisages universal access to banking facilities with at least one basic banking account for every household, financial literacy, access to credit, insurance and pension facility. The plan also envisages channelling all Government benefits (from Centre / State / Local Body) to the beneficiaries' accounts and pushing the Direct Benefits Transfer (DBT) scheme of the Union Government.

The technological issues like poor connectivity, on-line transactions will be addressed. Mobile transactions through telecom operators and their established centres as Cash Out Points are also planned to be used for Financial Inclusion under the Scheme. Also an effort is being made to reach out to the youth of this country to participate in this Mission Mode Programme.

1.2.7 Other Innovations

Over the last few years, new companies and systems have emerged on the payments landscape, each of which has brought in newer technologies, and improvements on the payments experience, including easier access to the payments networks. In the below section, we highlight a few of them.

Square

Square introduced a simple piece of hardware that could turn a mobile phone into a payments device that can accept credit cards, thus allowing any one to accept card payments securely. This has now been replicated by many companies, and has expanded the number of people who can accept card payments. Square's innovation cantered around the use of a secure hardware, that could be use the compute and communication capabilities of a smartphone to enable anyone to accept a card payment.

Stripe

Stripe introduced simple APIs that allow any company with a web presence to securely accept electronic payments in as little as 10 lines of code, and a simple signup process. Stripe's innovation centres on the ease of use of acceptance of the payment information bypassing the merchant systems - thus improving security for the customer.

Apple Pay

Apple has recently launched an electronic wallet, which allows the user to make payments from existing cards, based on a 2 factor biometric authentication. This is based on the biometric sensor installed in the phone, and a local secure element. The transaction is secure, and provides additional privacy to the user.

Apple's innovation includes the use of local biometrics, built in security, the use of cryptography, and virtual card numbers to ensure that user data cannot be compromised, while ensuring that payment can be done with a remarkable ease of use.

Virtual Currencies

Over the past few years, bitcoin has created a system with a virtual currency that allows users to perform transactions, which cannot be repudiated, in the absence of a centralized trusted ledger. Based on this, and exchanges which allow the currency to be exchanged with real currencies, an entire bitcoin economy has taken off. In this economy, the cost of transactions is very low, while the amount of security is high.

Bitcoin's innovation has cantered around the creation of a distributed ledger in the absence of a centralized trusted party, the use of proof-of-work as an incentive to maintain the ledger, and the use of a language to enable innovation in the use of payments, and the creation of smart contracts.

Bitcoin has sparked off the creation of many virtual currencies, each with different characteristics. Other virtual currencies such as Litecoin and Dogecoin are also coming up in the worldwide market.

Ripple, Stellar and other Networks

The creation of virtual currencies, and exchanges has resulted in the creation of additional networks, which provide hooks into the existing financial system, allowing the exchange of currencies, and movement of money across financial institutions, thus further strengthening the use case for virtual currencies.

Ripple, and Stellar enhance the usability of virtual currencies and real currencies, including the ability to connect institutions and exchange value.

Card Tokenization

The card number has not been a secret, and is visible to many entities in the payments chain, along with other payment credentials. The security for these systems has been controlled through audit and certification, along with instructions related to how various data has to be handled. In the event of a compromise, the institution have to replace the card - which is an expensive process. However, the use of a virtualized card number reduces the replacement cost to almost nothing. In fact, if the users always use a virtualized card number, the system becomes that much more secure, because there is no value to stealing a number that is bound to change shortly.

Use of Smartphones as an authentication factor

Smartphones have become ubiquitous, more capable, and always stay with their owner. It is easy to see how they can be trusted by the relying party to become an authentication factor. For instance, the act of sending an OTP over the phone, and its use to access a secure system is an accepted form of security. The phone essentially becomes a 'what you have' credential. This is further extended through other mechanisms such as HOTP, and TOTP where the OTP does not have to be sent at all - but can be generated on the phone in a manner which the relying party can accept. This has been used in various systems - such as email systems, enterprise security systems, and payment systems.

Biometric Enabled Smartphones

Since smart phones have become an extension of the human identity, it has become important to secure them (and transactions) with an additional form of security. This has been achieved through the use of passwords, or biometrics to unlock the phone and its applications. Applications can use biometrics in the phone to capture credentials and use them in payment processes. While currently available popular phones use fingerprint technology, with the availability of cheaper Iris devices having much higher match accuracy, it is expected that Iris enabled smart phones will be available in near future.

1.3 The Opportunity

We are at the cusp of a revolution – technology is continuing to enter people's lives, making it easier. Users expect that their interactions with banks will keep pace technology trends. These expectations are beginning to take hold in India as well, and we see many banks offering these services to customers. With new players entering the payment landscape, it is to be expected that they will innovate to differentiate their services and to improve customer experience.

It is very important that NPCI takes leadership role in ensuring that the participants continue to innovate while staying interoperable with existing systems. Interoperability allows the market to grow, provides customers with true any-to-any open payments, and helps to significantly reduce cash payments.

Ecommerce, both on the web and the mobile, offers a potential area for exponential growth in electronic payments. Payments is a large issue for the players in this segment and they will continue to look for technology that will improve the payments process.

> The proliferation of smart phones, the availability of an online verifiable identity, universal access to banking, and the introduction of biometric sensors in phones provide India a unique opportunity to take the lead in electronic payments and provide customers with enhanced security and unparalleled ease of use.

Towards this, NPCI shall provide a technology platform which is:

- Unified – hiding the complexity of dealing with disparate systems – both internal and external to NPCI.
- Expandable – to allow for innovations in newer forms of identity, authentication, and banking
- Adaptable to the current way of life-
  - Smart phones as an integral part of people's identity
  - Aadhaar as a form of online verifiable identity - authenticated by a third party
  - Allow customers to enter credentials on their own device – even when the merchant requests funds.
  - E Commerce.
- Real Time – Allows banks to provide real time experience for interactive transactions.
- Secure – Allows for traceability through the entire transaction chain
- Monitorable - Allows for NPCI to monitor the system centrally.

NPCI shall take the lead in creating an easy to use system, which will allow banks to provide superior customer experience without compromising on security and staying within security and regulatory framework.

1.4 Document Scope

The purpose of this document is to clearly state the strategy for building a Unified Payment Interface and its technology and API details. In addition to technology and API details, this document also provides detailed examples, use cases, and flows. This document describes a set of APIs to do immediate money transfer in a unified way irrespective of source, destination, and authentication. These APIs provide a federated, multi-provider, mobile based instant payment mechanism on top of NPCI platform.

2. Unified Payment Interface

This chapter introduces the Unified Payment Interface and its 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

Unified Payment Interface provide the following core features via a single payment API and a set of supporting APIs.

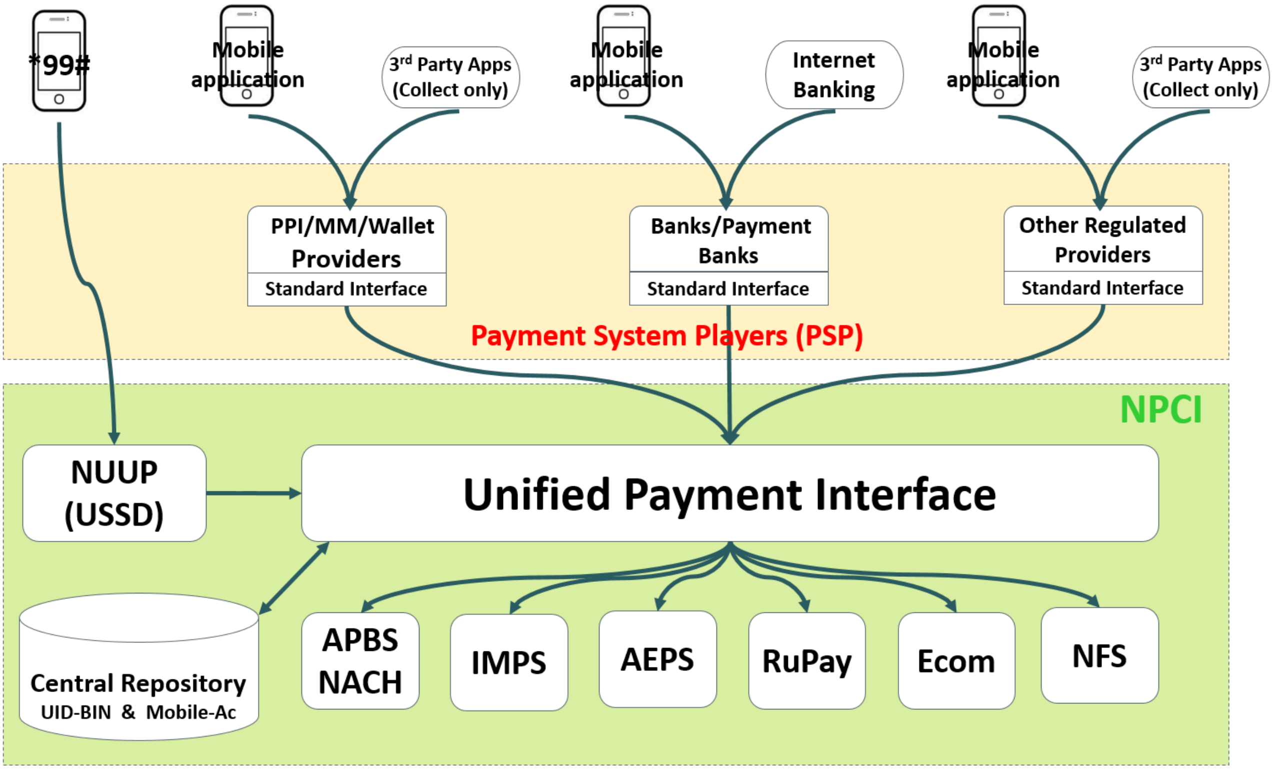
1. Ability to use personal mobile as the primary device for all payments including person to person, person to entity, and entity to person.
2. Ability to use personal mobile to "*pay*" someone (push) as well as "*collect*" from someone (pull).
3. Ability to use Aadhaar number, mobile number, card number, 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 with others without having 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 payment requests to be "snoozed" and paid later before expiry date without having to block the money in the account until customer is ready to pay.
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 payment system players 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 (on phone).

9. Ability to use a fully interoperable system across all payment system players without having silos and closed systems.
10. Ability to make payments using 1-click 2-factor authentication all using just a personal phone without having any acquiring devices or having any 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 uses existing systems such as IMPS, AEPS, etc. to orchestrate these transactions and ensure settlement across accounts. Usage of existing systems ensure reliability of payment transactions across various channels and also takes full advantage of all the investments so far. This unified layer offers next generation peer-to-peer immediate payment just by using personal phone.

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, 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.

Virtual addresses offered by the provider need not be of permanent nature. For example, a provider may offer "one time use" addresses or "amount/time limited" addresses to customers. In addition, innovative usage of virtual addresses such as "limit to specific payees" (e.g., a virtual address that is whitelisted only for transactions from IRCTC) can help increase security without sacrificing convenience. PSPs can allow their customers to create any number of virtual payment addresses and allow attaching various authorization rules to them.

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

Authentication is typically done at the account provider domain. 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 heavily these days to offer 2-FA authentication schemes. One authentication is required to be performed by the Payment Service Provider - for instance, the use of the correct mobile phone, while the other is performed within the domain of the account provider.

Traditionally, payment account provider themselves provided the authentication scheme. Account management (KYC, opening account, managing transactions, etc.) were tightly coupled with internal authentication schemes. But, conceptually, account management including KYC etc. should be loosely coupled with authentication. Aadhaar authentication is one such trusted external authentication schemes used today within the payment systems. Micro-ATMs (handhelds with biometric sensors) used by BCs take advantage of Aadhaar authentication via NPCI which, in turn, is trusted by banks to conduct payment transactions.

Digital Signatures, especially proposed Aadhaar enabled DSCs, 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.

In this unified architecture, objective is to enable 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.

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 using token less payment scheme allows banks to reduce the overall issuance (card, PIN, etc.) cost while still keeping authorization and account management within its control.

2.4 Value Proposition

> Unified interface provides significant advantage from current systems to take mobile payments to next level. Its value lies in using customer's mobile phone as the primary device for all authentication and authorization for both "*Direct Pay*" (push) and "*Collect Pay*" (pull) transactions.

The proposed Unified Payment Interface provide the following values.

1. **Simplifying Authentication** - India is the only country in the world to offer trusted 3rd party biometric authentication as a utility service. With universal coverage of Aadhaar expected in 2015, PSPs can take advantage of this utility to provide secure, convenient authentication service to a billion people without having the need to do card/PIN issuance lifecycle. Similarly NPCI offered centralized MPIN management options via USSD can allow banking customers with registered mobile to easily set and change MPIN without having any explicit issuance mechanisms.

2. **Simplifying Issuance Infrastructure** - Usage of virtual addresses and payment addresses in conjunction with mobile as the "what you have" factor helps banks to create token-less infrastructure reducing the costs.

3. **Simplifying Acquiring Infrastructure** - Use of mobile as the primary device for payment authorization can completely transform the issuance infrastructure to be easy, low cost, and universal. Considering the fact that India has nearly a billion phones and 150 million smartphones (expected to be at 500 million in next 4-5 years), massive scale can be achieved if effective use of mobile is made compared to creating costly physical acquiring infrastructure.

4. **Flexibility for PSPs -** Payment system players (RBI regulated entities such as banks, payment banks, PPIs, and their technology service providers) can offer superior mobile experience to their customers. In addition, this unified interface still allows a fully on-us scheme if both payer and payee are on their network.

5. **Flexibility for Users -** Customers get the ability to make payments securely to their friends, relatives, pay to merchants, pay bills, etc. all using their mobile phones without having to share any account details or credentials with others. In addition, innovations such as reminders, using multiple accounts via single mobile applications, using special purpose virtual addresses, etc. allow users to enjoy superior experience.

6. **Enabling 1-click 2-FA Transactions** - This proposal allows all transactions to be at least 2-FA using mobile and any other factor (Password, PIN, and biometrics). Since mobile number is bound to the device, explicit SMS based OTP need not be used every time which makes authorization simpler. When biometric sensor integrated mobiles start becoming available, payments can be done with no data entry making electronic payments extremely convenient, but still providing full 2-FA security.

7. **Stimulating Innovation** - This interface provide a very simple API that is minimalistic, fully functional, and allowing innovations in various aspects such as user interface, convenience features, authentication schemes, and mobile devices to be brought in without having to change the core API structure.

8. **Embracing Mobile Adoption** - This interface truly embraces mobile and low cost smartphone adoption in India allowing phones to be the primary device for all payments and integrating mobile numbers by allowing paying to/from a mobile number.

9. **Embracing Aadhaar Adoption** - Universal digital identity is fast becoming a reality with Aadhaar adoption crossing 730 million. With Aadhaar e-KYC allowing paperless, anytime anywhere e-KYC services, Aadhaar now a payment destination using APB, usage of Aadhaar authentication as a trusted 3rd party authentication, large scale electronic payments can be achieved unlike ever before.

10. **Creating National Interoperability** - With introduction of new payment service players such as payment banks, PPIs, and others, it is necessary that India adopt an interoperable mobile payment strategy to allow customers to send and receive from any other customer within the PSP or across PSPs in a seamless fashion. Proactively creating this unified interoperable interface allows all players to innovate and provide superior customer experience and still provide a secure, standard based, interoperable payment scheme.

2.5 Supporting Infrastructure

2.5.1 Aadhaar System

One of the key considerations is to keep the Aadhaar system purely focused on identity and nothing else. The Aadhaar system only collects minimal data just enough to provide unique identity, issue the Aadhaar number after biometric de- duplication, manage lifecycle changes of that identity record, and provide a secure Application Programming Interface (API) for verifying the identity (online authentication) for various applications requiring identity verification. Designing the Aadhaar system as pure identity platform allows clear separation of duties and leaves usage of identity to other partners, and their various applications which may be built on top of the Aadhaar platform.

2.5.1.1 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).

Combination of Aadhaar number and biometrics deliver online authentication without needing a token (such as a smartcard). During biometric authentication, agency collects the Aadhaar number along with one or more biometric impressions (e.g., one or more fingerprints, or iris impression alone, or iris impression along with fingerprints) which then encrypted and sent to Aadhaar authentication server for authenticating the resident.

2.5.1.2 Aadhaar e-KYC

Verification of the Proof of Identity (PoI) and Proof of Address (PoA) is a key requirement for access to financial products (payment products, bank accounts, insurance products, market products, etc.), SIM cards for mobile telephony, and access to various Central, State, and Local Government services. Today, customers provide physical PoI and PoA documents. Aadhaar is already a valid PoI and PoA document for various services in the financial, telecom, and Government domains.

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 become authentication and e-KYC user agencies (AUA/KUA) by signing up with UIDAI and can easily integrate these services within their application to provide low cost, paperless, and convenient KYC and authentication services to their customers.

2.5.1.3 Aadhaar Enabled Account (AEA)

In order to facilitate disbursements, remittances or any financial transaction using Aadhaar as the financial address, a resident is required to link their Aadhaar number with his/her bank account number. Customers have the option of either linking their existing bank account or opening a new bank account.

2.5.1.4 Aadhaar Payment Bridge (APB)

The Aadhaar Payments Bridge (APB) offers a simplified payment mechanism to Government user departments to electronically transfer subsidies and benefit payments 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.

The Aadhaar Payments Bridge will facilitate the processing of payments file from the Government departments received via the sponsor banks (assigned bank), and subsequently routing of the payments file 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. Aadhaar Payments Bridge (APB) is a payments service offered by National Payments Corporation of India and the process for on-boarding of banks has also been defined by NPCI.

Currently APB system has about 120 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 APB mapping database will have about 200-250 million Aadhaar mappings within next 12-18 months.

2.5.1.5 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.

A vital building block in this endeavour is developing a standard platform that will become cost effective with scale and provide real time authentication, even in remote areas. For this, standards for on-line, interoperable devices termed microATMs were finalized by a committee consisting of members from RBI, Indian Banks Association (IBA), Banks, Institute for Development and Research in Banking Technology (IDRBT), and UIDAI. A Proof of Concept was done in Jharkhand in partnership with Bank of India, Union Bank of India and ICICI Bank for these microATM-based transactions in early 2011. The pilot project for payments started in December 2011 in Jharkhand.

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.

2.5.2 NPCI Central Mapper

Aadhaar based payments are currently being processed using NACH application. For this purpose idea of mapping Aadhaar with Bank was first conceived and was institutionalized by NPCI. Aadhaar is predominantly being used for transferring all types of government benefits. However recently Government also mandated that benefits can be transferred using Account Numbers as well.

Further considering the other financial revolution and reengineering which is currently going on in our country like Unified API, IMPS, USSD platform, NPCI Central Mapper can be used 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.

2.5.2.1 Aadhaar as the Payment Address

NPCI has collaborated with Unique Identification Authority of India (UIDAI) to create a centralised Aadhaar mapper. The Aadhaar mapper, at present acts as an enabler for payment owing to the Aadhaar number mapping to the Account number as the financial address. NPCI has already build capabilities such as the e-KYC and Aadhaar Payment Bridge (APB) around this enablement.

2.5.2.2 Mobile as the Payment Address

NPCI is enhancing the central mapper to also have mobile to account mapping. This allows anyone to send/receive money from a mobile number without knowing the destination account details. Customers, via USSD, can manage multiple mobile to account mapping and conduct transactions via USSD. This feature also allows smartphone users to seamlessly interoperate with feature phone users. Unified Payment Interface allows PSPs to take full advantage of this mapping and allow their users to send/receive money just providing a destination mobile number.

2.6 Example Usage Scenarios

This section 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.

2.6.1 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.

2.6.2 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.

2.6.3 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 pending collect request) and authorizes the payment for Rs.12000/-.

2.6.4 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 same and ticket is issued.

2.6.5 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.

2.6.6 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.

2.7 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.

2.7.1 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 Device	PSP & NPCI (via common library)	Customer Registration & during transaction	SMS based OTP initially against the registered mobile and using HOTP/TOTP for implicit verification during every transaction
Aadhaar Number or PAN number	PSP	Customer Registration	Aadhaar e-KYC / Authentication or PAN card verification
Customer Name	PSP	Customer Registration	Aadhaar e-KYC / Demographic Authentication, matching with PAN card verification
Account Details - Number, Account Ownership,	PSP	Every time a payment account is added	Ideally via an API offered by account providers or via a small value (e.g. Rs.1/-) transaction

2.7.2 Protecting Account Details

- Protecting during capture
- Verifying the account details with account provider (bank, PPI, etc. - new API may be needed from banks, or Re-1 transaction may be done to validate)
- PSPs storing the data should be always in encrypted form

2.7.3 Protecting Authentication Credentials

- Authentication credentials encrypted during capture using the public key of the authentication provider
- Never in saved from capture till use

- Never logged anywhere before it reaches provider
- "Trusted" common library for credential (MPIN/Password/PIN/Biometrics) capture. This library needs to bind customer mobile using HOTP/TOTP which is verified as part of transaction.

2.7.4 Protecting against Phishing

3 core techniques may be used to protect against phishing:

- Individual (nonentities) pay/collect transactions can be against pre-created and verified address (quite like in the case of NEFT).
  - Allow direct/collect against ONLY whitelisted within the payer's pre-listed entries. Payer must add the payee explicitly into this list (quite life NEFT settings). During this, address verification can be done.
- For individuals
  - PSP application should mandatorily share Aadhaar number and verified name which is part of customer information block which can be shown by the second PSP to their customer.
- For entities
  - PSP application should mandatorily share PAN number and verified name which is part of customer information block which can be shown by the second PSP to their customer.
  - Whitelist entities (popular ones) and blacklist/rating at central database (NPCI) and show "verified symbol".

Whenever a collect payment request comes, 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.

2.7.5 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 for appropriate number of years

3. High Level Specifications

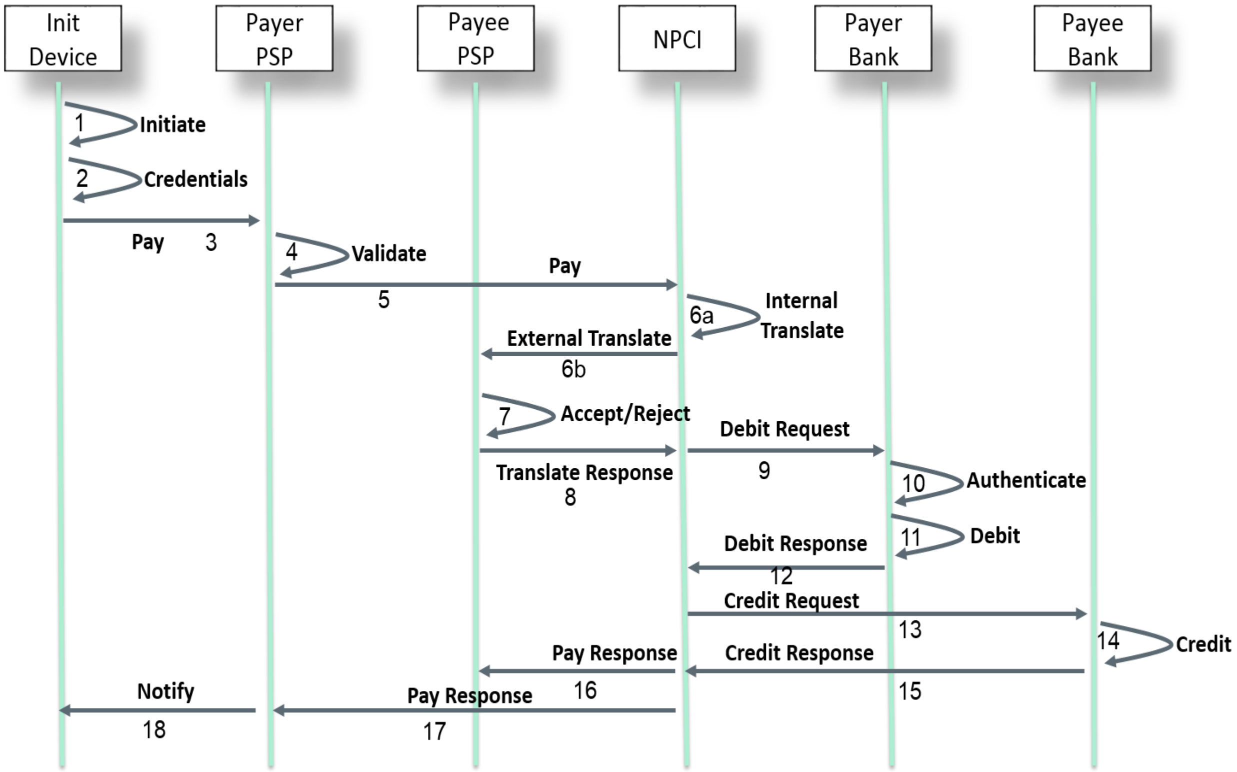
3.1 Direct Pay (Sender/Payer initiated)

3.1.1 Person Initiated

1. Sender uses an application to send money to a receiver by providing sender credentials and receiver/beneficiary "address"
2. E.g., paying a friend via a mobile banking application

3.1.2 System Initiated

1. "Sender system" (a software application) is initiating payment.
    a. An electronic mandate or digitally signed request is used
2. E.g. Automatic daily commission payments to agents from the payroll application



3.1.3 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.

3.1.4 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.

3.2 Collect Pay (Receiver/Payee Initiated)

3.2.1 Local Collect

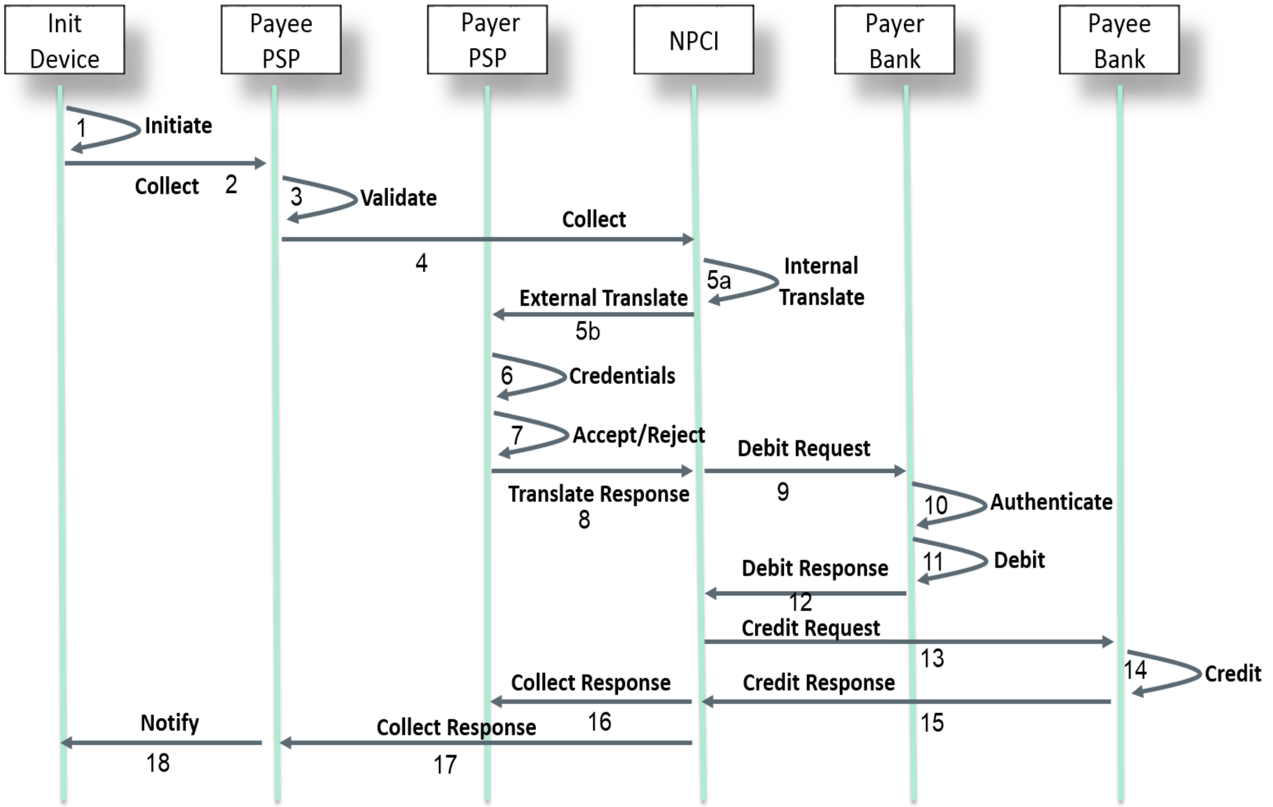
1. Paying a company using PoS/mobile application
2. Payer's smartphone captures the payee payment address, transaction reference, from PoS (SMS, QRCode, etc.)
    a. Allows payee to authorize and capture payee's payment information including credentials (eliminates any credential entry on external apps)
    b. Payee's smartphone transfers the data securely to PoS which then carries out the transaction

3.2.2 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



3.2.3 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

the Payer Address is resolved by NPCI central Mapper.
   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.

3.2.4 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.

3.3 APIs at a Glance

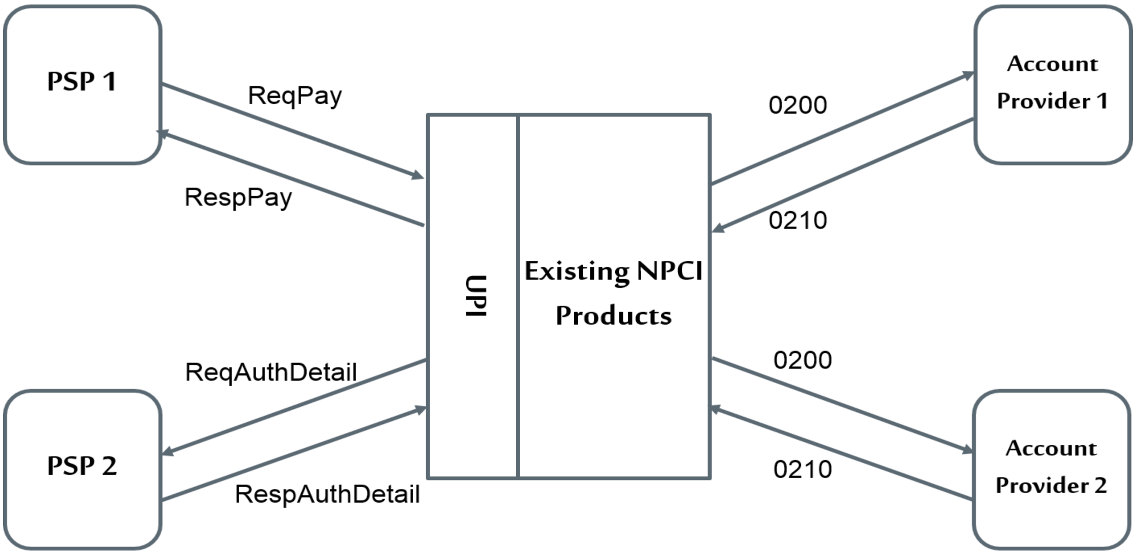
All APIs are asynchronous in nature meaning once the request is sent, response is sent back separately via corresponding response API. This allows same APIs to be used for instant payment as well as delayed payments. This also allows APIs to scale without having to wait in a blocking mode. Callers are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

3.3.1 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).

3.3.2 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.

3.3.3 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.

3.3.4 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.

3.3.4.1 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">
    </Txn>
    <Payer addr="ram@sbi" name="Ram" seqNum="1" type="PERSON">
        <Info>
            <Identity type="UIDAI" verifiedName="Ram" id="111122223333"/>
        </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" id="111122223333"/>
        </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" id="123456789012"/>
            </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">
    <!-- 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 -->
    <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>
```

3.3.4.2Scenario 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">
        <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" id="111122223333"/>
            </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" id="111122223333"/>
            </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" id="123456789012"/>
        </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">
    <!-- 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">
    <!-- 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>
```

3.3.5 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.

3.3.5.1 listPSPs

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 during before initiating the transaction.

3.3.5.2 listAccountProviders

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.

3.3.5.3 verifyAccount

This API is used by the PSPs during the on boarding process. PSPs should use this API to make sure that the Account details provided by their customers are genuine and customers have the credentials to operate the account.

3.3.5.4 validateAddress

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

3.3.5.5 listKeys

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 for encryption of credential data. Trusted and certified libraries will be used by PSPs for credential capture and encryption at capture time. These libraries can be provided by NPCI or respective banks.

3.3.5.6 setCredentials

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.

3.3.5.7 createWLentry, updateWLentry, getWLentries

NPCI will offer a mechanism to whitelist of entity addresses for protection against phishing. This list is a common collection accessible to PSPs via APIs so that popular entities such as LIC, Indian Railways, ecommerce players, telecom players, bill payment entities, etc. can be whitelisted. NPCI, with the help of PSPs, will define a process to manage these entries.

3.3.5.8 checkTxnStatus

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.

3.3.5.9 Central Mapper APIs

- search - provided ONLY to authorized users/systems for grievance/dispute handling purposes
- create/update - APIs takes one or many records from the account providers or authenticated customer sessions and create/update mapper entries

4. Detail API Specifications

4.1 API Protocol

All APIs are exposed as stateless service over HTTPS. 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 this URL as XML document using Content-Type “application/xml” or “text/xml”.

Following is the URL format for all APIs under the unified interface:

```
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.

4.2 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="" ref="" ts="" type="PAY">
        <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|UIDAI|BANK" verifiedName="" id=""/>
            <Rating whiteListed="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="IFSC">
            <Detail name="ACCOUNT" 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 ="RUPAY">
            <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="PIN|MPIN|PIN">
                <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|UIDAI|BANK" verifiedName="" id=""/>
                <Rating whiteListed="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

Index	Message Item	<XML Tag>	Occurrence
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.)	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 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
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	<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
5.3.3	ID of the identifier	id	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

Index	Message Item	<XML Tag>	Occurrence
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
5.1	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
5.10.2	Authentication Subtype	subtype	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
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
6.4.3	ID of the identifier	id	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

Index	Message Item	<XML Tag>	Occurrence
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
6.1	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

4.3 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="" ref="" ts="" type="PAY"/>
    <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	ref	1..1

Index	Message Item	<XML Tag>	Occurrence
	Loan number, etc.)		
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
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

4.4 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|UIDAI|BANK" verifiedName="" id=""/>
            <Rating whiteListed="TRUE|FALSE"/>
        </Info>
        <Amount value="" curr="INR">
            <Split name="PURCHASE|CASHBACK" value=""/>
        </Amount>
    </Payer>

    <Payees>
        <Payee seqNum="" addr="" name="">
            <Info>
                <Identity type="PAN|UIDAI|BANK" verifiedName=""/>
                <Rating whiteListed="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 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
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

Index	Message Item	<XML Tag>	Occurrence
5.3	Payer Identity	<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
5.3.3	ID of the identifier	id	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
6.4.3	ID of the identifier	id	1..1
6.5	Rating of the Payee	<Payee.Info.Rating>	0..1
6.5.1	Payee is whitelisted or not	whiteListed	1..1
6.8	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

4.5 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|UIDAI|BANK" verifiedName="" id=""/>
            <Rating whiteListed="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|UIDAI|BANK" verifiedName="" id=""/>
                <Rating whiteListed="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 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 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

Index	Message Item	<XML Tag>	Occurrence
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	<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
5.3.3	ID of the identifier	id	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
5.10.2	Authentication Subtype	subtype	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
6.2.5	Merchant Classification Code -MCC	code	1..1

Index	Message Item	<XML Tag>	Occurrence
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
6.4.3	ID of the identifier	id	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

4.6 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.

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: ref

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: 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

4.1.5 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

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

5.3.3 Attribute: id

Presence: [1..1]

Definition: This attribute contains the ID/number as maintained by the identifying authority as mentioned in 5.3.1. It will be PAN number, Aadhaar Number & Bank Account Number

Data Type: Alphanumeric

Format: Min Length: 1
Max Length: 35

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

Code	Definition
	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
IFSC	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
RUPAY	If the customer account will be identified using RUPAY 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
IFSC	IFSC, ACTYPE, ACNUM	Provide for Account based payments
MOBILE	MMID, MOBNUM	Provide for Mobile based payments
RUPAY	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	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

6.4.3 Attribute: id

Presence: [1..1]

Definition: This attribute contains the ID/number as maintained by the identifying authority as mentioned in 5.3.1. It will be PAN number, Aadhaar Number & Bank Account Number.

Data Type: Alphanumeric

Format: Min Length: 1
Max Length: 35

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

Code	Definition
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
IFSC	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
RUPAY	If the customer account will be identified using RUPAY 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
IFSC	IFSC, ACTYPE, ACNUM	Provide for Account based payments
MOBILE	MMID, MOBNUM	Provide for Mobile based payments
RUPAY	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

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