Designing UPI – System Design

Designing a Unified Payments Interface (UPI) system involves creating an architecture that enables real-time inter-bank transactions through a seamless and secure platform. Key components include user interfaces, a central UPI switch managed by the National Payments Corporation of India (NPCI), backend systems of participating banks, and third-party service providers.

The system supports core functionalities such as user registration and authentication, payment initiation and authorization, inter-bank transaction processing, and real-time settlement.
Security measures like encryption and multi-factor authentication ensure compliance and protect against fraud, making UPI a reliable and efficient payment solution.
The UPI is a real-time payment system that facilitates inter-bank transactions by instantly transferring funds between two bank accounts on a mobile platform.

Important Topics for UPI system design

1. Functional Requirements for UPI System Design
2. Non-Functional Requirements for UPI System Design
3. Capacity Estimation for UPI System Design
4. High-Level Design(HLD) for UPI System Design
5. Low-Level Diagram(LLD) for UPI System Design
6. Microservices Used in UPI System Design
7. Scalability for UPI System Design

Functional Requirements for UPI System Design

Below are the functional requirements for UPI system design:
User Registration and Authentication- Users should be able to register and create a UPI account. Provide secure authentication mechanisms (e.g., PIN, biometric authentication).
Bank Account Linking- Users should be able to link multiple bank accounts to their UPI profile. Facilitate the management of linked bank accounts (e.g., add, remove, and update account details).
Payment Address Management- Users should be able to create and manage Virtual Payment Addresses (VPAs).
Money Transfer- Support person-to-person (P2P) transfers. Support person-to-merchant (P2M) payments. Enable scheduled and recurring payments.
Transaction History- Provide users with a detailed transaction history, including status and timestamps.
Notifications- Send real-time notifications for all transactions (e.g., payment success, failure).
Payment Requests- Allow users to request payments from other UPI users.
QR Code Payments- Generate and scan QR codes for quick and easy payments.
Bill Payments- Support utility bill payments through the UPI interface.

Non-Functional Requirements for UPI System Design

Below are the non-functional requirements for UPI system design:
Scalability– The system should handle a large number of users and high transaction volumes. Support horizontal scaling to manage increasing load efficiently.
Performance- Ensure low latency for transaction processing and user interactions. Provide real-time transaction processing capabilities.
Reliability– Ensure high availability with minimal downtime. Implement failover mechanisms to handle system failures gracefully.
Security– Implement robust security measures to protect user data and financial transactions. Ensure compliance with financial regulations and standards (e.g., PCI-DSS).
Consistency– Ensure consistency in transactions to avoid discrepancies and double-spending. Implement mechanisms to ensure eventual consistency in a distributed system.
Compliance- Adhere to regulatory requirements and guidelines set by financial authorities. Ensure data privacy and protection in compliance with relevant laws.
Usability- Design a user-friendly interface that is easy to navigate. Ensure accessibility for users with different levels of technical expertise.

Capacity Estimation for UPI System Design

Below is the capacity estimation for UPI system design:
Traffic Estimate
Users: 100 million users
Daily Active Users (DAU): 10 million users
Transactions per User per Day: 5 transactions
Total Transactions per Day: 50 million transactions
Storage Estimate
Average Transaction Size: 500 bytes
Daily Data: 50 million * 500 bytes = 25 GB
Monthly Data: 25 GB * 30 = 750 GB
Yearly Data: 750 GB * 12 = 9 TB
Bandwidth Estimate
Request Size: 1 KB
Response Size: 1 KB
Total Requests per Day: 50 million
Total Data per Day: (50 million * 2 KB) / (1024 * 1024) = ~95 GB
Memory Estimate
Active Users Cached: 1 million
Data per User: 1 KB
Total Memory: 1 million * 1 KB = 1 GB

High-Level Design(HLD) for UPI System Design

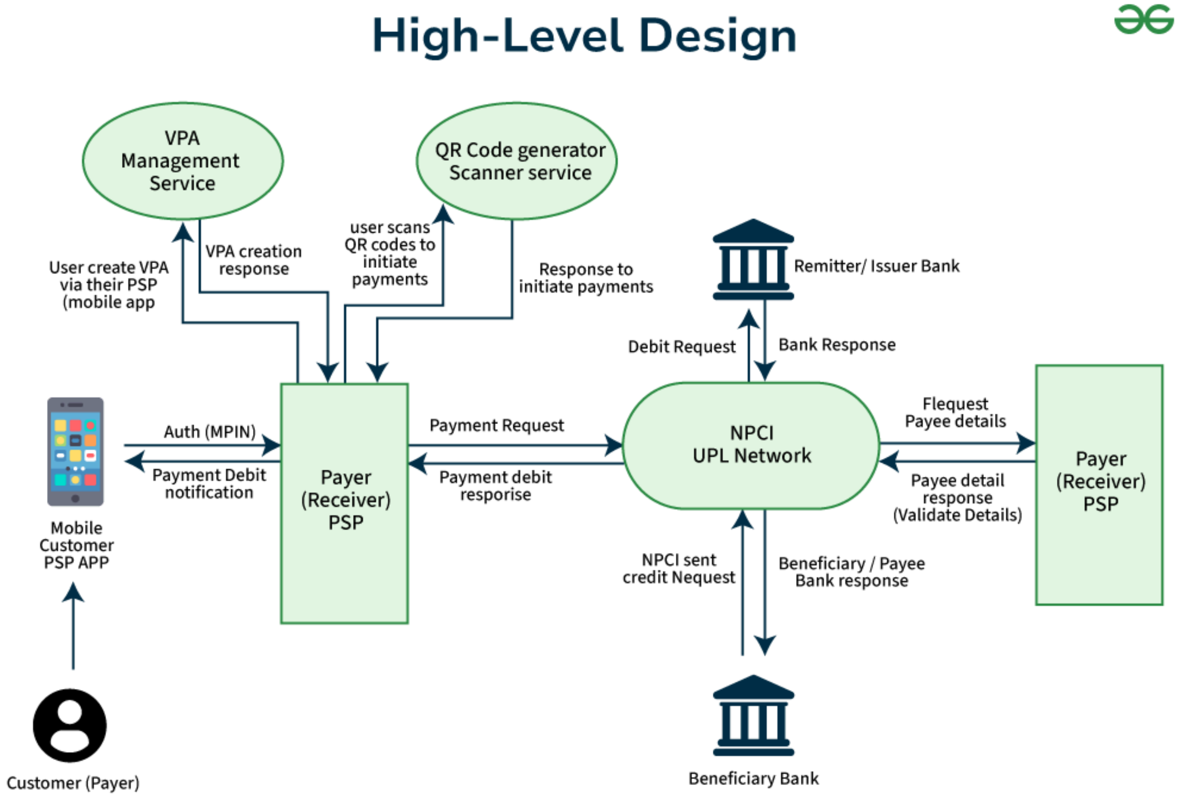
A high-level diagram for a UPI system provides an overview of the main components and their interactions. At the top layer, users interact with UPI through mobile applications or web interfaces provided by banks or third-party service providers (e.g., Google Pay, PhonePe).

These interfaces connect to the UPI service, a centralized system managed by the National Payments Corporation of India (NPCI).

The UPI service acts as a switch that handles transaction requests, routing them to the appropriate bank's backend systems. Each participating bank has its own infrastructure to manage account verification, balance checks, and transaction processing.
The UPI switch facilitates real-time communication and settlement between the payer's and payee's banks. Additionally, the system incorporates security mechanisms, including encryption and authentication, to ensure secure transactions.
APIs allow third-party apps to integrate UPI functionalities, extending the system's usability and accessibility. This high-level structure ensures efficient, secure, and seamless inter-bank transactions.

High-Level-Design

The diagram illustrates the flow of a Unified Payments Interface (UPI) transaction, which is a real-time payment system developed by the **National Payments Corporation of India (NPCI)**. It shows the interaction between various entities involved in a UPI transaction. Here's a detailed explanation:



1. Entities

Customer (Payer): The individual initiating the payment.

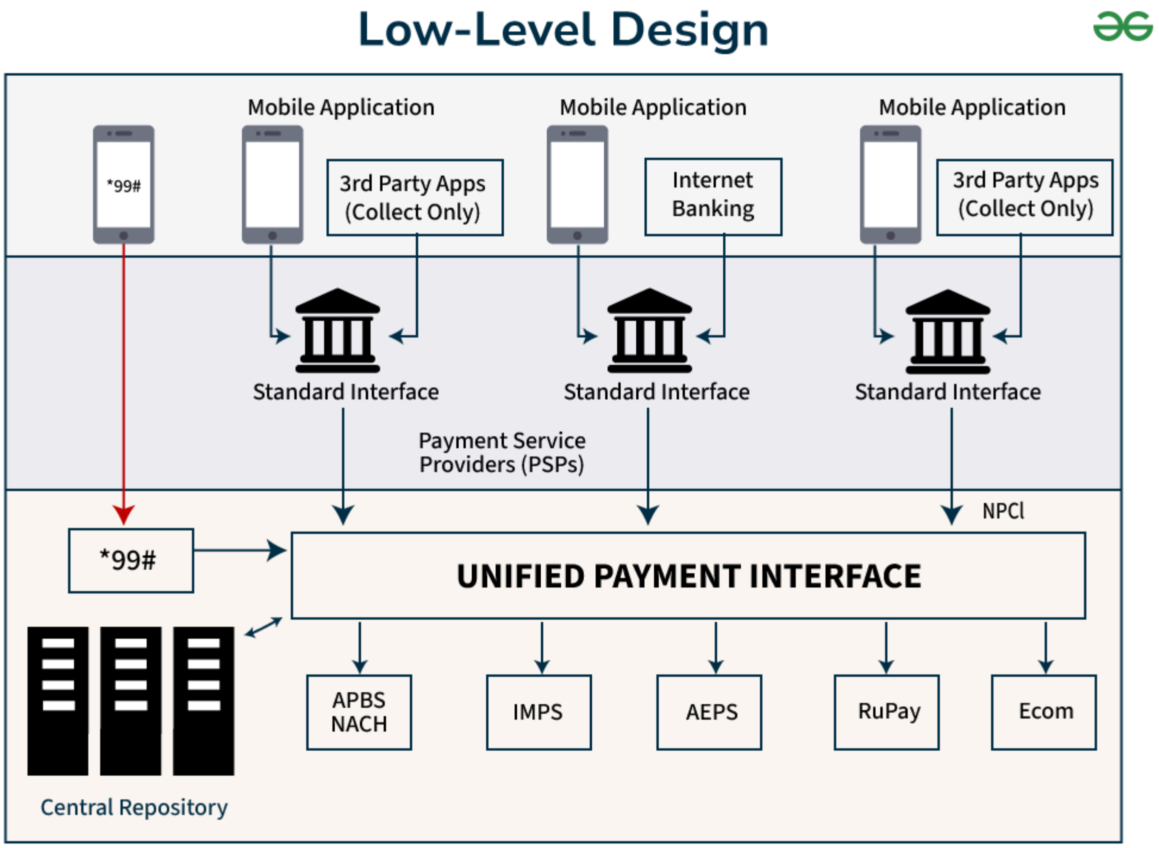
Payer (Receiver) PSP: The Payment Service Provider (PSP) of the payer, which facilitates the transaction.
Remitter/Issuer Bank: The bank of the payer.
Beneficiary Bank: The bank of the payee.
Payee (Receiver) PSP: The Payment Service Provider of the payee.
NPCI UPI Network: The central network that processes UPI transactions.
VPA Management Service: Service for managing Virtual Payment Addresses (VPAs).
QR Code Generator/Scanner Service: Service for generating and scanning QR codes for payments.

2. Flow of the Transaction
VPA Creation: The customer creates a Virtual Payment Address (VPA) via their PSP's mobile app using the VPA Management Service.
The VPA creation request is sent to the VPA Management Service, and a response is received.
Initiating Payment: The customer can initiate a payment by scanning a QR code using the QR Code Generator/Scanner Service. The QR code is scanned, and a response is received to initiate the payment.
Payment Authorization: The customer uses their mobile PSP app to authorize the payment using MPIN (Mobile Personal Identification Number). The payer PSP sends a payment request to the NPCI UPI Network
Processing Payment: The NPCI UPI Network sends a debit request to the Remitter/Issuer Bank.
The Remitter/Issuer Bank responds to the NPCI UPI Network with the bank response. The NPCI UPI Network sends a credit request to the Beneficiary Bank. The Beneficiary Bank responds to the NPCI UPI Network with the bank response.
Payee Details Validation: The NPCI UPI Network requests payee details from the Payee PSP.
The Payee PSP validates the details and responds to the NPCI UPI Network.
Completion of Transaction: The NPCI UPI Network sends a payment debit response to the Payer PSP.
The Payer PSP notifies the customer of the payment debit.

In Conclusion, The customer initiates a payment using their mobile app by either creating a VPA or scanning a QR code. The payer PSP sends a payment request to the NPCI UPI Network. The NPCI UPI Network processes the payment by interacting with the payer's and payee's banks. The transaction is completed, and the customer is notified of the payment status.

Low-Level Diagram(LLD) for UPI System Design

A low-level diagram for a UPI system illustrates the detailed interaction between its core components. At the user level, we have mobile apps and web interfaces that allow users to register, authenticate, and initiate transactions. These interfaces communicate with the UPI service through APIs, which are crucial for handling requests and responses.



The UPI service, managed by the National Payments Corporation of India (NPCI), acts as a central switch that routes transaction requests to the appropriate bank systems. Each participating bank has backend systems that verify user account details, check balances, and process transactions.

The UPI switch ensures real-time transaction processing and settlement by securely relaying information between the payer's and payee's banks. Additionally, the system includes security modules for encryption and fraud detection, ensuring the integrity and confidentiality of transactions.

Third-party service providers integrate with the UPI service via APIs to offer UPI functionalities within their apps, expanding the system's reach and usability.

Low-Level-Design

The diagram illustrates the architecture and flow of the Unified Payments Interface (UPI) system, which is a real-time payment system developed by the National Payments Corporation of India (NPCI). It shows how different entities interact within the UPI ecosystem to facilitate transactions. Here's a detailed explanation:

1. Entities and Components

Mobile Application: USSD (*99#): A service that allows users to perform banking transactions using a basic mobile phone without internet access.

3rd Party Apps (Collect Only): Third-party applications that can initiate payment collection requests.
Internet Banking: Traditional internet banking services accessed via mobile applications.

Banks: Standard Interface: The interface provided by banks to interact with the UPI system.

Payment Service Providers (PSPs): Entities that provide payment services to customers and merchants, facilitating transactions through the UPI system.

Unified Payment Interface (UPI): The central platform that processes UPI transactions, developed and managed by NPCI.

Central Repository: A centralized database that stores transaction data and other relevant information.

NPCI: National Payments Corporation of India, the organization that manages the UPI system.

2. Other Payment Systems:
APBS/NACH: Aadhaar Payment Bridge System/National Automated Clearing House.
IMPS: Immediate Payment Service.
AEPS: Aadhaar Enabled Payment System.
RuPay: A domestic card payment network.
Ecom: E-commerce transactions.

3. Flow of Transactions:
USSD (*99#): Users can initiate transactions using the USSD code *99# on their mobile phones. The request is sent to the Central Repository and then to the UPI system for processing.

Mobile Applications: Users can initiate transactions using various mobile applications, including third-party apps and internet banking apps. These applications interact with the banks' standard interfaces to process transactions.

Banks: Banks provide a standard interface to interact with the UPI system. They act as intermediaries between the users and the UPI system, facilitating the transactions.

Payment Service Providers (PSPs): PSPs facilitate the transactions by interacting with the UPI system and the banks. They ensure the smooth processing of transactions initiated by users.

Unified Payment Interface (UPI): The UPI system processes the transactions by interacting with the banks and other payment systems. It ensures real-time processing and settlement of transactions.

Central Repository: The Central Repository stores transaction data and other relevant information.
It interacts with the UPI system to provide necessary data for transaction processing.

Other Payment Systems: The UPI system can interact with other payment systems like APBS/NACH, IMPS, AEPS, RuPay, and Ecom to facilitate various types of transactions.

The diagram shows how users can initiate transactions using different methods (USSD, mobile apps, internet banking). Banks provide a standard interface to interact with the UPI system. Payment Service Providers (PSPs) facilitate the transactions. The UPI system processes the transactions in real-time, interacting with banks and other payment systems. The Central Repository stores transaction data and interacts with the UPI system.

Microservices Used in UPI System Design

Below are some microservices used in UPI System Design:

1. User Service
    a. POST /register: Register a new user.
    b. POST /login: Authenticate a user.
2. Bank Service
    a. POST /link-account: Link a bank account.
    b. GET /balance: Get account balance.
3. Transaction Service
    a. POST /transfer: Transfer money.
    b. GET /history: Get transaction history.
4. Notification Service
    a. POST /notify: Send notification.
    b. Scalability for UPI System Design
5. Horizontal Scaling:
    a. Microservices Architecture: Each service can be scaled independently based on demand.
    b. Load Balancing: Distribute incoming requests across multiple instances.
6. Caching
    a. Redis: Use Redis to cache user sessions, frequently accessed data, and transaction states.
7. Database Sharding
    a. Shard by User ID: Distribute user data across multiple database instances to reduce load.
8. Asynchronous Processing

    a. Message Queues: Use message queues (e.g., RabbitMQ, Kafka) to handle asynchronous tasks like notifications and transaction processing.

9. Fault Tolerance
    a. Circuit Breaker Pattern: Implement circuit breakers to prevent cascading failures.
    b. Retry Mechanism: Retry failed operations to handle transient failures.

Conclusion

Designing a UPI system requires careful consideration of both functional and non-functional requirements. By leveraging a microservices architecture, implementing caching, and ensuring fault tolerance, the system can efficiently handle a large number of users and transactions. Scalability is achieved through horizontal scaling, load balancing, and database sharding, ensuring the system remains performant and reliable even under high load.