

FinTech Payments : Foundation of Fintech

Anjan Mahapatra

February 2026

1 Payments: Foundation of FinTech

1.1 What Are Payments?

Payments refer to the financial systems and processes that enable the transfer of monetary value from one party (the *payer*) to another party (the *payee*) using digital or physical payment rails. These systems are designed to ensure:

- **Speed:** Low-latency authorization and settlement
- **Security:** Protection against fraud, misuse, and data breaches
- **Compliance:** Adherence to financial regulations and standards
- **Reconciliation:** Accurate recording and settlement of transactions

Formally, a payment transaction can be modeled as a value transfer function:

$$T : (Payer, Payee, Amount, Time) \rightarrow Settlement$$

where settlement represents the final and irrevocable transfer of funds.

—

1.2 Payment Ecosystem Players

A payment system operates as a multi-party ecosystem involving the following entities:

—

1.3 Types of Payments

1.3.1 Card Payments

Card payments involve credit or debit cards and are widely used for both online and offline transactions.

Player	Role
Customer (Payer)	Initiates the payment transaction using a card, bank account, or wallet
Merchant (Payee)	Receives the payment in exchange for goods or services
Payment Gateway	Acts as the technical interface that securely transmits payment data
Acquiring Bank	Merchant's bank that processes incoming payment requests
Payment Network	Routes transaction messages between acquirer and issuer
Issuing Bank	Customer's bank that authorizes or declines the transaction
Regulator	Defines rules, compliance requirements, and operational limits

Table 1: Payment Ecosystem Participants

Modes

- Card-Not-Present (CNP): Online and in-app payments
- Point-of-Sale (POS): Physical merchant terminals

Transaction Flow

Authorization → Clearing → Settlement

Key Risks

- Chargebacks due to disputes
 - Card-not-present fraud
 - High Merchant Discount Rate (MDR)
-

1.3.2 Bank Transfers

Bank transfers enable direct movement of funds between bank accounts.

- **NEFT**: Batch-based settlement
 - **RTGS**: Real-time gross settlement for high-value transactions
 - **IMPS**: Instant retail transfers
-

1.3.3 Real-Time Payments

Real-time payment systems allow instant fund transfers with immediate confirmation.

- 24×7 availability
 - Immediate settlement
 - High scalability requirements
-

1.3.4 Wallets

Digital wallets store monetary value electronically and are typically classified as prepaid instruments.

- Closed wallets
 - Semi-closed wallets
 - Open wallets
-

1.4 Payment Architecture (High-Level)

A generic high-level payment architecture is represented as follows:

User → App → Gateway → Acquirer → Network → Issuer

Issuer → Network → Acquirer → Gateway → App → User

The reverse path represents authorization or decline messages.

1.5 Key Payment Metrics

Payment systems are evaluated using the following quantitative metrics:

Success Rate

$$\text{Success Rate} = \frac{\text{Number of Successful Transactions}}{\text{Total Initiated Transactions}}$$

Latency

$$\text{Latency} = T_{\text{response}} - T_{\text{request}}$$

Drop-off Rate

$$Drop-off = 1 - \frac{Completed\ Transactions}{Initiated\ Transactions}$$

Cost per Transaction

$$Cost = Fixed\ Cost + Variable\ Cost$$

Chargeback Ratio

$$Chargeback\ Ratio = \frac{Number\ of\ Chargebacks}{Total\ Transactions}$$

Refund SLA Defined as the maximum allowed time for refund completion after initiation.

1.6 Payment Product Trade-offs

Product decisions in payments involve inherent trade-offs, as summarized below:

Decision	Trade-off
Strong fraud checks	Reduced transaction conversion rate
Faster checkout flow	Increased fraud exposure
Multiple payment gateways	Higher operational and reconciliation complexity

Table 2: Key Payment Trade-offs

1.7 Leading Payment Products

Modern payment platforms provide developer-friendly APIs, risk management, and settlement orchestration.

- **Stripe:** API-first global payment infrastructure
- **Razorpay:** End-to-end payment and banking solutions