# SDE: System Design and Engineering

## Lecture – 5

## Introduction to

## Payment and Fintech

## From Zero to Google: Architecting the Invisible Infrastructure

*by*

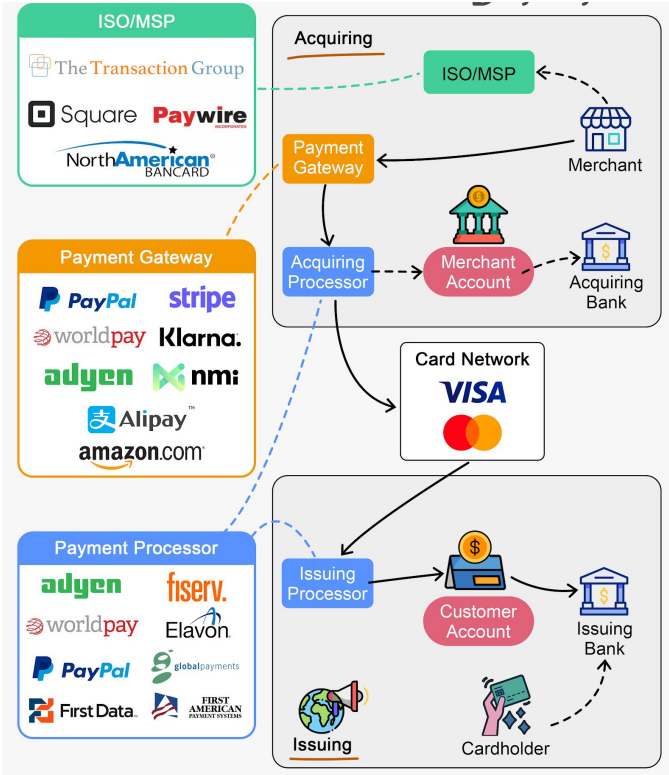
**Aatiz Ghimire**

# Sections

- Introduction to Payments & Fintech Ecosystem
- Payment Systems & Protocols
- Digital Wallets & Mobile Payments
- Payment Security & Reconciliation (Next Class)
- System Design for Scalable Payment Platforms (Next Class)

# Overview of the Global Payment Ecosystem

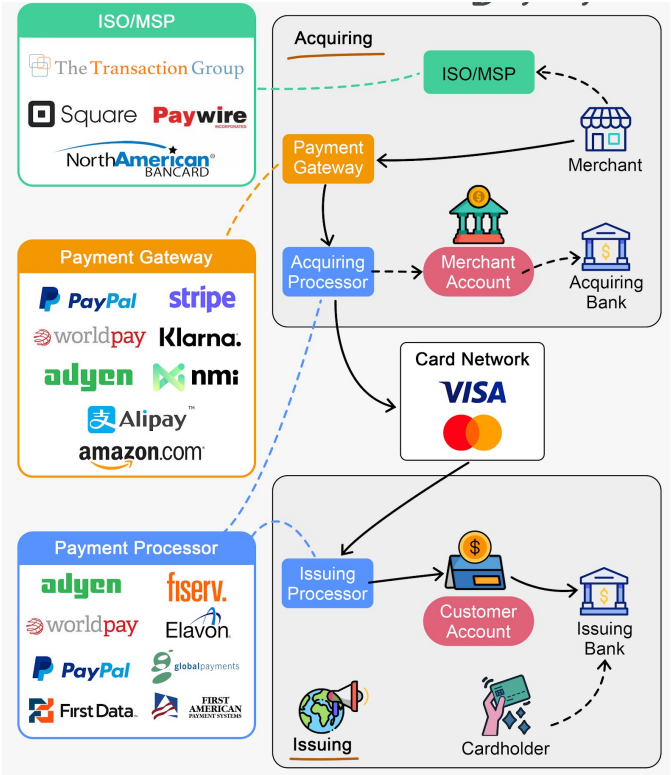
- Multiple stakeholders work together to enable secure, fast transactions.
- Includes **cardholders, merchants, banks, processors, gateways, and networks**.
- Many fintech companies (e.g., PayPal, Stripe, Square) start in one niche, then expand across the value chain
- Nepal-specific examples: **Fonepay, connectIPS, eSewa, Khalti**.



# Overview of the Global Payment Ecosystem

***Step1 :Account & Merchant Onboarding***

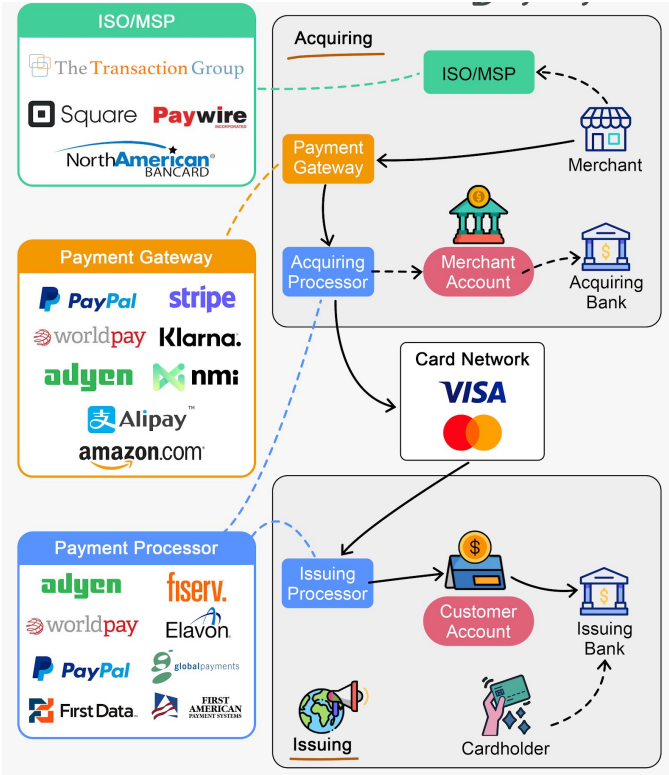
- **Cardholder:** Opens an account with an **issuing bank** and receives a debit/credit card. (e.g., Nabil Bank, Global IME Bank).
- **Merchant:** Registers with an **ISO (Independent Sales Organization)** or **MSP (Member Service Provider)** for in-store or online sales.(e.g., Fonepay, connectIPS).
- **Processor Partnering:** Merchant account opened with acquiring processor.



# Overview of the Global Payment Ecosystem

*Step 2-5: Acquiring Process*

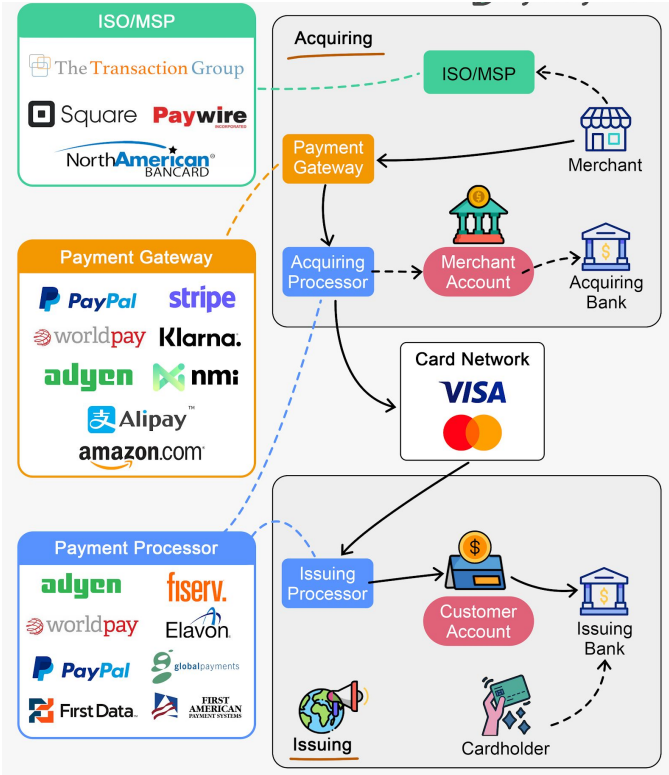
- **Payment Gateway:** Collects payment data (global: Stripe, PayPal; Nepal: Fonepay Gateway, NCHL).
- **Acquiring Processor:** Requests authorization from card network or local payment switch.
- **Networks:**
  - *Global:* VISA, Mastercard, UnionPay.
  - *Nepal:* Fonepay Network (for QR, interbank), NCHL (for bank transfers).
- **Settlement:** Funds transferred to merchant's account after processing delay.



# Nepal's Payment Ecosystem – Key Players

*Step 6–8: Issuing Process*

- **Issuing Processor:** Acts on behalf of the **issuing bank** to verify the transaction.
- **Card Network:** Facilitates secure data exchange between issuer and acquirer.
- **Issuing Bank:** Validates the account, checks available funds/credit, and approves or declines the transaction.
- *Nepal Example:* A FonePay QR payment → Fonepay switch → customer's bank (issuer) verifies & approves → settlement back to merchant's bank.



# Industry Examples

**Key Components:**

- **Card Networks:** VISA, Mastercard, UnionPay (via local banks).
- **National Payment Switch:** Nepal Clearing House Ltd. (NCHL) – runs connectIPS, NCHL-IPS, NCHL-ECC.
- **QR Payments:** Fonepay QR (interoperable across banks, wallets).
- **Digital Wallets:** eSewa, Khalti, IME Pay, Prabhu Pay.
- **Interbank Transfers:** connectIPS, NCHL-IPS.
- **Real-Time Payments:** Fonepay QR, wallet-to-wallet instant transfers.

# Industry Examples

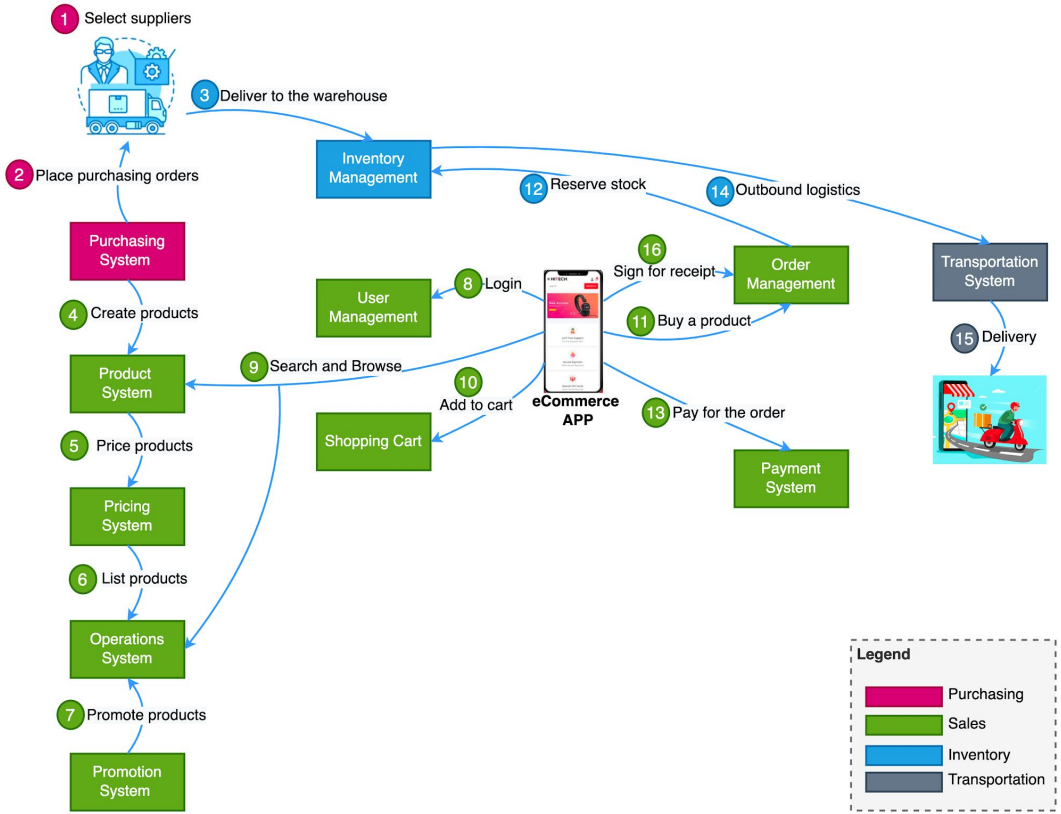
**Global:**

- PayPal – online payments, merchant services, P2P transfers.
- Stripe – developer-first gateway and APIs.
- Square (Block) – POS + lending + merchant services.

**Nepal:**

- eSewa – first digital wallet in Nepal, online & offline payments.
- Khalti – mobile wallet with bill payment, fund transfer, merchant QR.
- connectIPS – interbank fund transfer, government payments, large transaction handling.
- Fonepay – nationwide QR network integrated with 60+ banks.

# E-commerce Payment



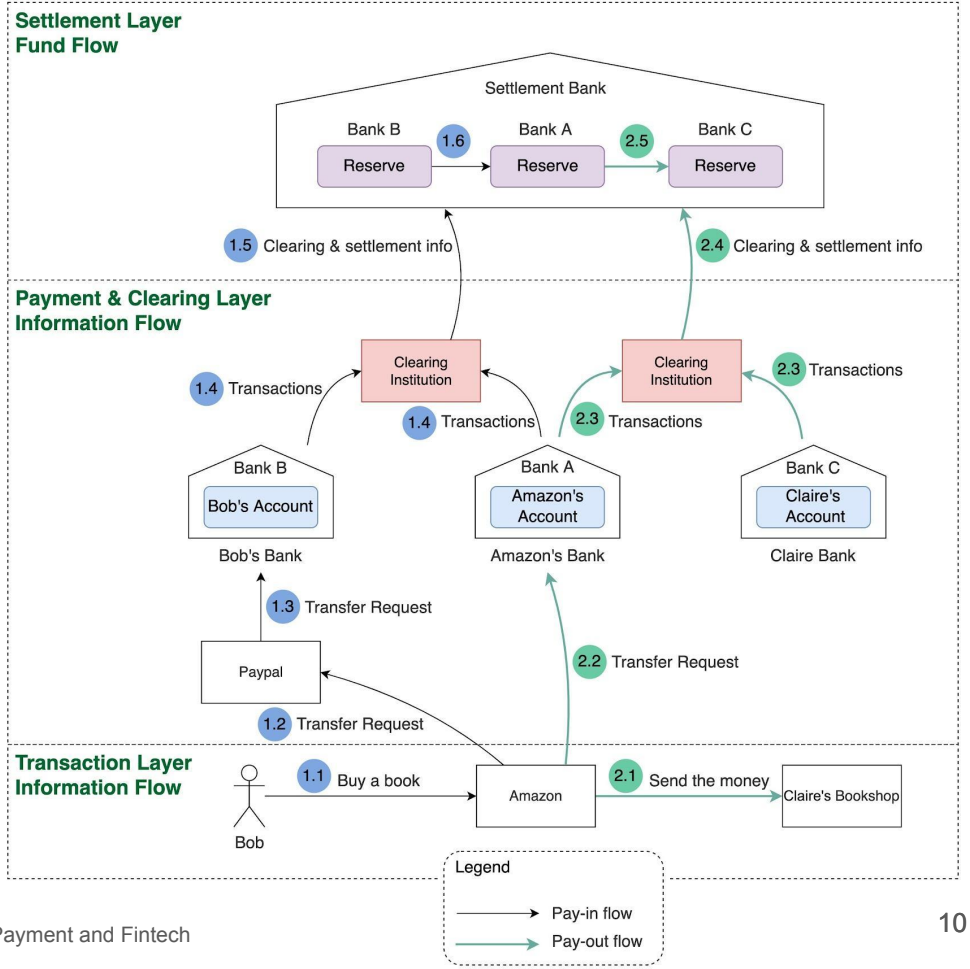
# Money Movement

**Clearing vs. Settlement**

- **Clearing:** Calculating *who* should pay *whom* and *how much* (netting transactions).
- **Settlement:** Actual money movement between **reserve accounts** in a settlement bank.

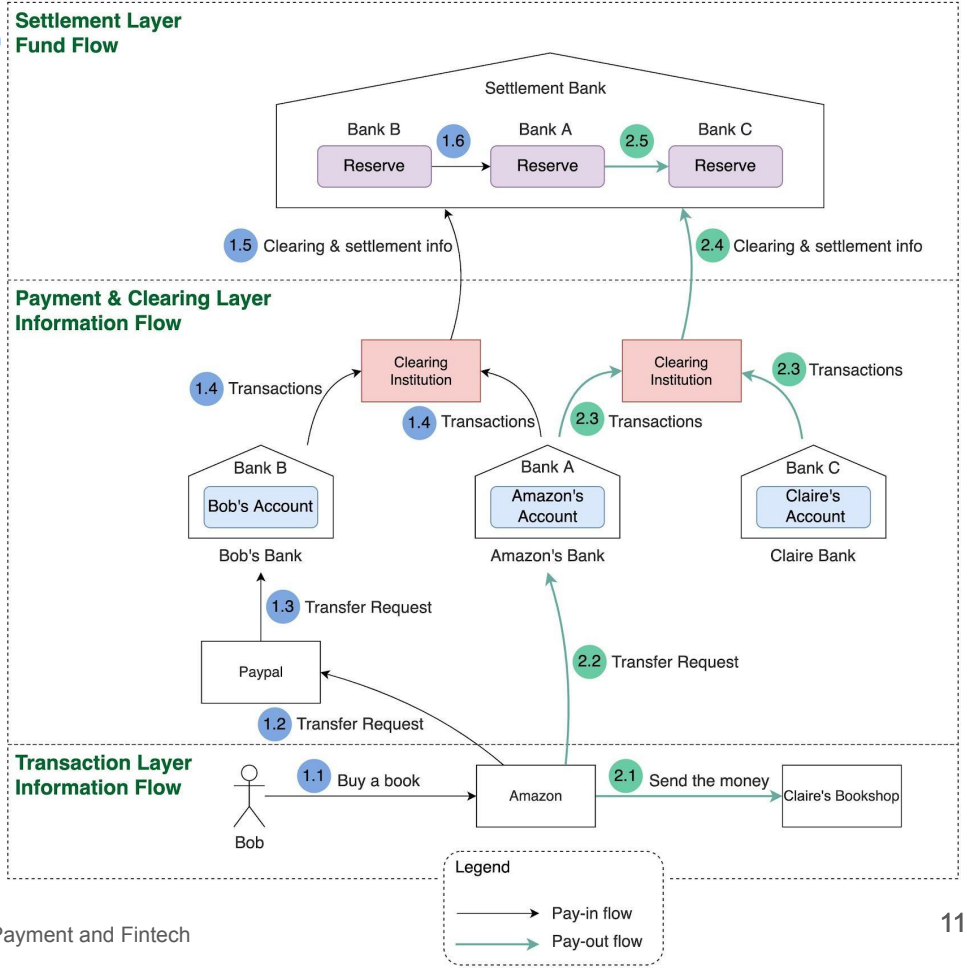
**Bob Buys a Book on Amazon (Scenario):**

- Bob buys an **SDI book** from Claire's shop on Amazon.
- Uses **PayPal** with his debit card token stored in PayPal.



# Pay-In Flow (Bob → Amazon)

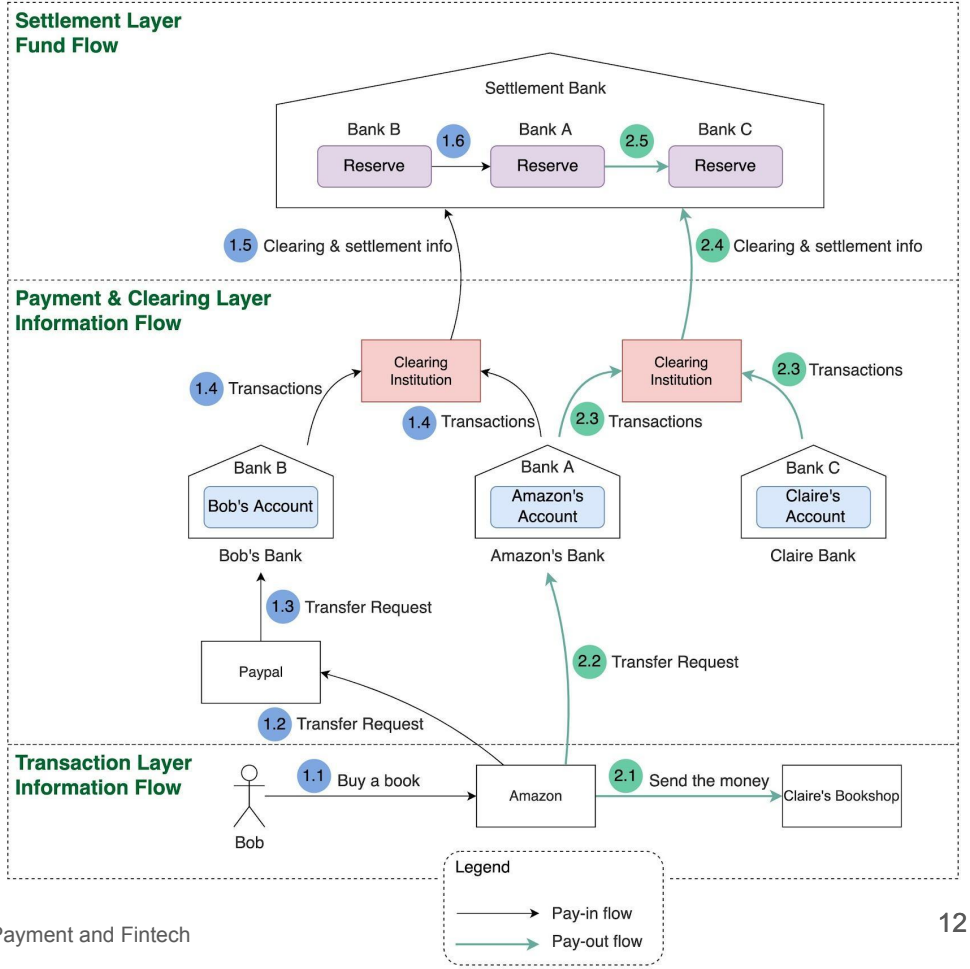
1. Bob places order → Amazon sends **money transfer request** to PayPal.
2. PayPal instructs Bob's bank (**Bank B**) to transfer money to Amazon's bank (**Bank A**).
3. **Clearing:**
   - Bank A owes Bank B $100, Bank B owes Bank A $500.
   - Netting → Bank B pays Bank A **$400**.
4. **Settlement:**
   - Clearing institution sends net settlement info to **settlement bank**.
   - Funds move between **reserve accounts** of Bank A & Bank B.



# Pay-Out Flow (Amazon→Claire)

1. Amazon notifies Claire she will be paid soon.
2. Amazon requests transfer from Bank A → Bank C (Claire's bank).
3. **Clearing:**

   Both banks send records to clearing institution for netting.
4. **Settlement:**
   Money moves from Bank A's reserve → Bank C's reserve at settlement bank.



# Three Layers of Payment Processing

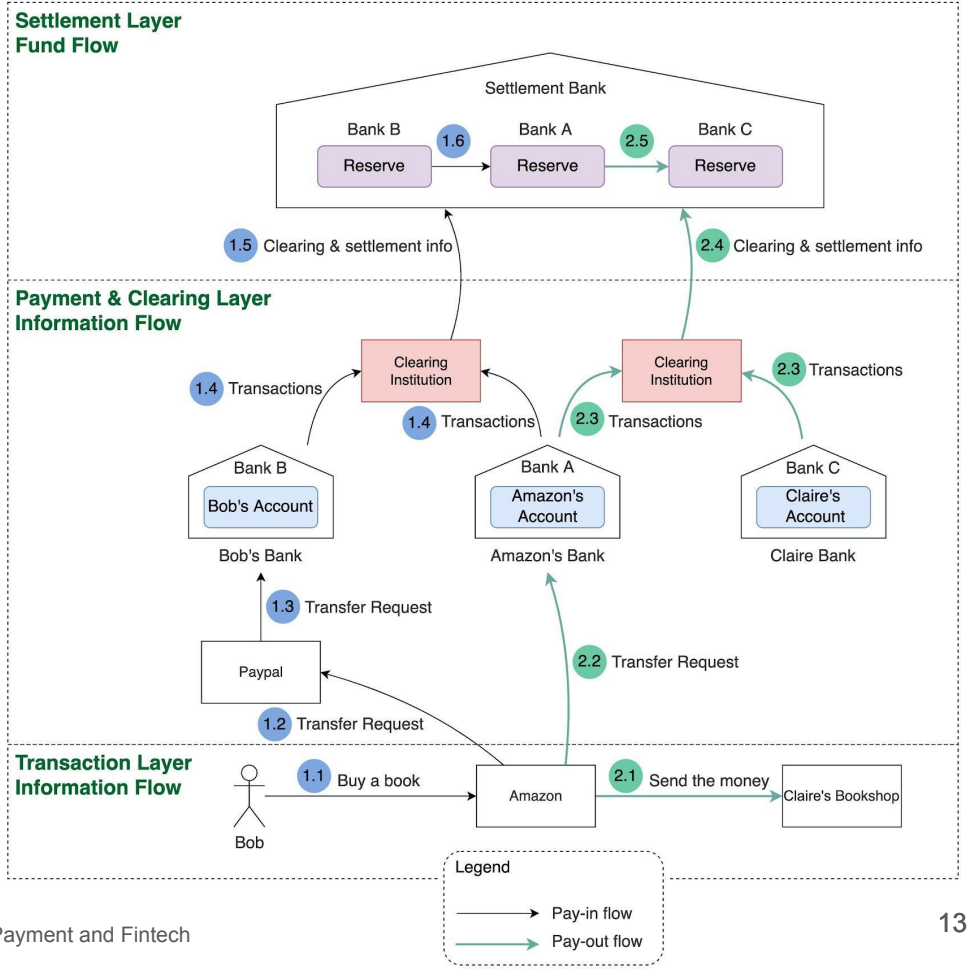
**1. Transaction Layer** – Where the purchase happens (Bob buys online).

**2. Payment & Clearing Layer** – Payment instructions + transaction netting.

**3. Settlement Layer** – Actual money movement between reserve accounts.

💡 **Info Flow** = Layers 1 & 2 (no real money moved yet)

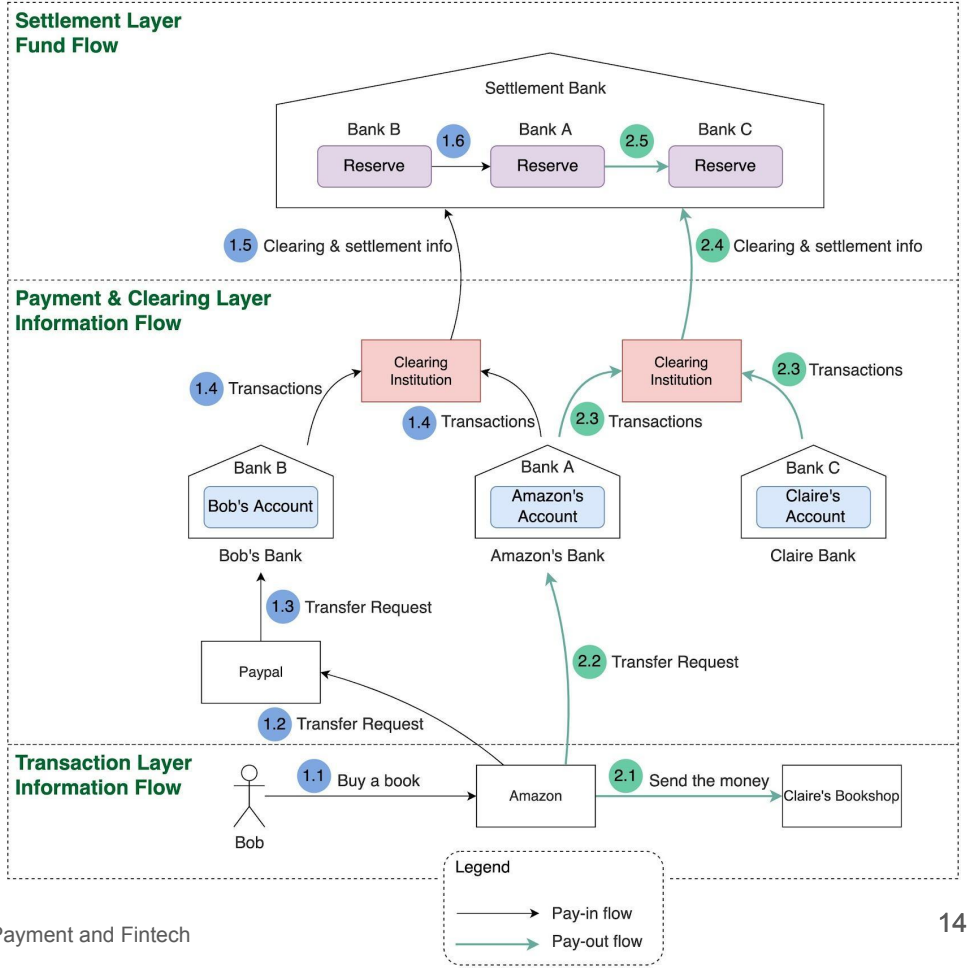
💡 **Fund Flow** = Layer 3 (real money transfer)



# International Payments Twist

**Scenario:** Bob buys a book from the **Indian market**:

- Bob pays in **USD**.
- Seller receives **INR**.
- Requires **foreign exchange settlement** between U.S. and Indian banking systems.
- Adds layers for **FX conversion**, compliance, and cross-border clearing.



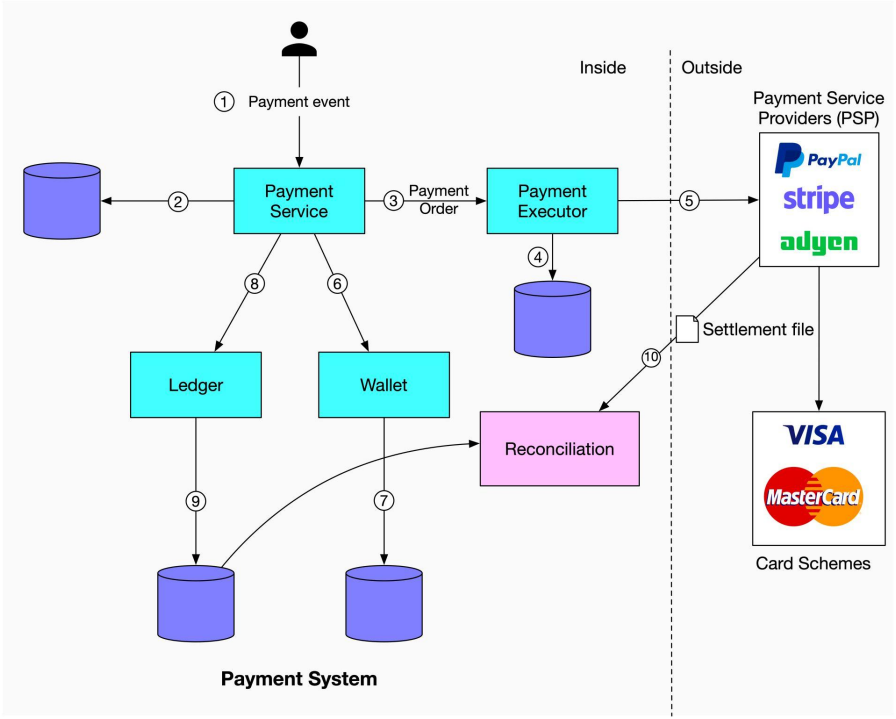
# Payment Systems & Protocols

**Architecture & Flow of a Typical Payment System**

**Key Idea:** What happens behind the scenes when you click **Buy** on Amazon or any e-commerce site.

**Core Functions:**

- Capture payment events
- Process orders via executors
- Interact with external payment providers (PSPs)
- Update seller balances & ledgers
- Reconcile through nightly settlements



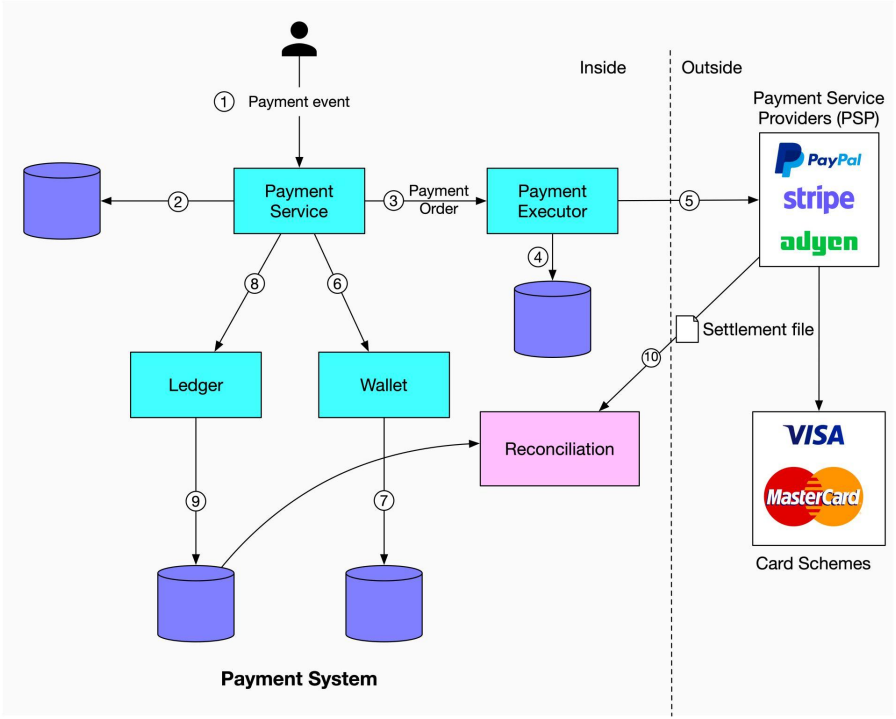
# Architecture & Flow of a Typical Payment System

**Step 1: Payment Event Flow:**

1. User clicks **Buy** → Payment event generated.
2. **Payment Service** stores the event in the database.
3. A single event may contain **multiple payment orders** (e.g., buying from multiple sellers in one checkout).

**Step 2: Payment Execution Flow:**

1. Payment Service calls **Payment Executor** for each order.
2. Payment Executor stores the **payment order** in its database.
3. Executor calls **external PSP** (e.g., Stripe, PayPal, VISA gateway) to process credit card payment.



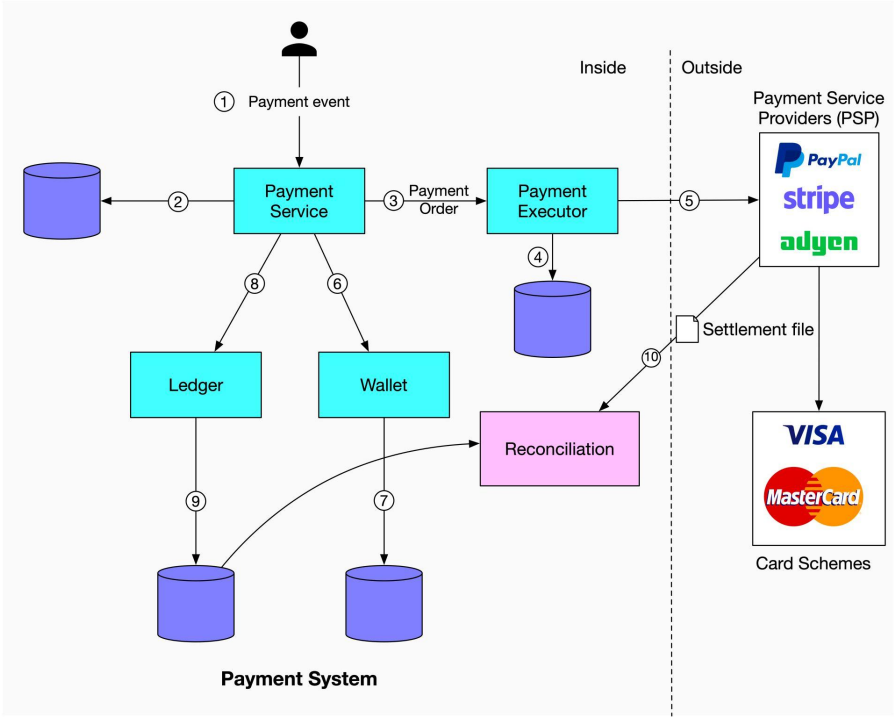
# Architecture & Flow of a Typical Payment System

**Step 3: Wallet Update Flow:**

1. After successful PSP execution, Payment Service updates the **wallet** for the seller.
2. Wallet Service updates **seller balance** in the database.

**Step 4: Ledger Update Flow:**

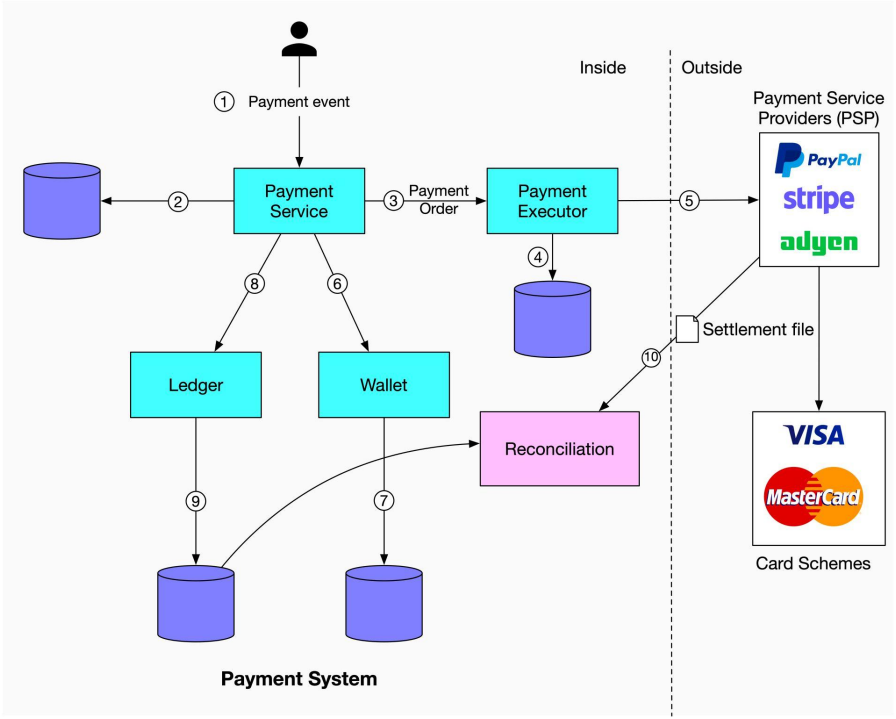
1. After wallet update, Payment Service calls **Ledger Service**.
2. Ledger Service appends new ledger entries to its database.
3. Ledger is the **source of truth** for financial records & audits.



# Architecture & Flow of a Typical Payment System

**Step 5: Settlement Flow:**

1. At day's end, PSPs or banks send **settlement files** to clients.
2. Files contain:
   - Bank account balance
   - All daily transactions
3. Used for **reconciliation** and **fund transfer finalization**



# Payment System Components

**Main Services:**

- **Payment Service:** Manages events & order creation.
- **Payment Executor:** Handles payment processing via PSPs.
- **Wallet Service:** Updates seller balances.
- **Ledger Service:** Maintains transaction history.
- **PSP:** External payment processors & networks.

**Nepal Adaptation:**

- PSP: eSewa, Khalti, connectIPS, Fonepay Gateway
- Wallet: Merchant wallet within the app or bank-linked account
- Ledger: Maintained internally by PSP or merchant system
- Settlement: Managed through **NCHL (Nepal Clearing House)** for interbank transactions.

# How VISA Works When Swiping a Credit Card

**Role of Card Networks (VISA, Mastercard & American Express)**

- Act as **intermediaries** for clearing & settling funds.
- Connect **card acquiring banks** (merchant side) and **card issuing banks** (customer side).
- Without card networks, banks would have to **settle directly with each other**, creating huge inefficiency.

**Two Main Flows in Card Payments**

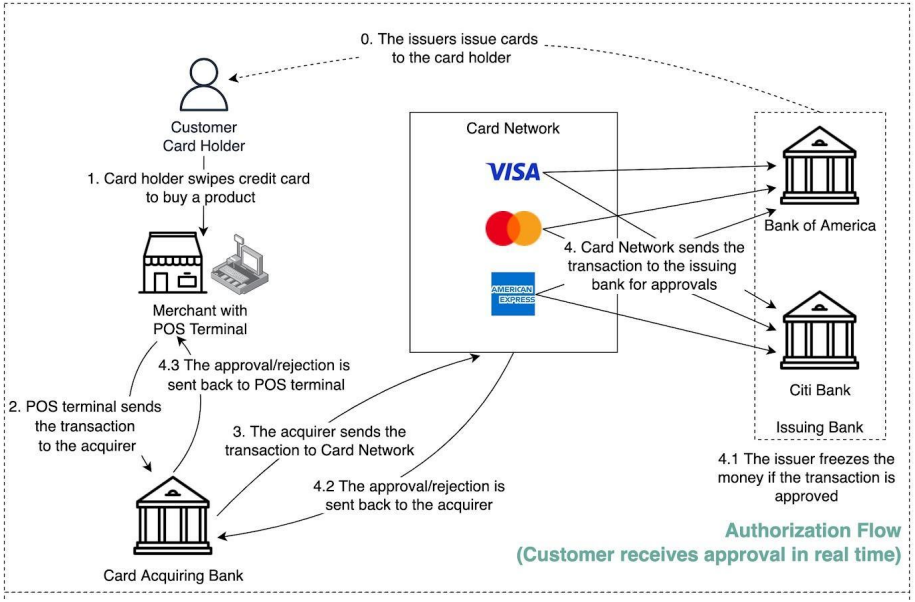
**1. Authorization Flow** – Transaction approval at the point of sale.
**2. Capture & Settlement Flow** – Actual transfer of funds to merchant.

# Authorization Flow

**Step-by-Step:**

0. **Issuing Bank** – Issues credit card to customer.
1. **Cardholder** – Swipes/taps card at merchant's POS terminal.
2. **POS Terminal** – Sends transaction to acquiring bank.
3. **Acquiring Bank** – Forwards transaction to **Card Network** (VISA/Mastercard/Amex).
4. **Card Network** – Routes to issuing bank for approval.

4.1–4.3 **Issuing Bank** – Freezes amount if approved → Sends approval/rejection back to acquirer → Returned to POS terminal.



# Capture & Settlement Flow

**Step-by-Step:**

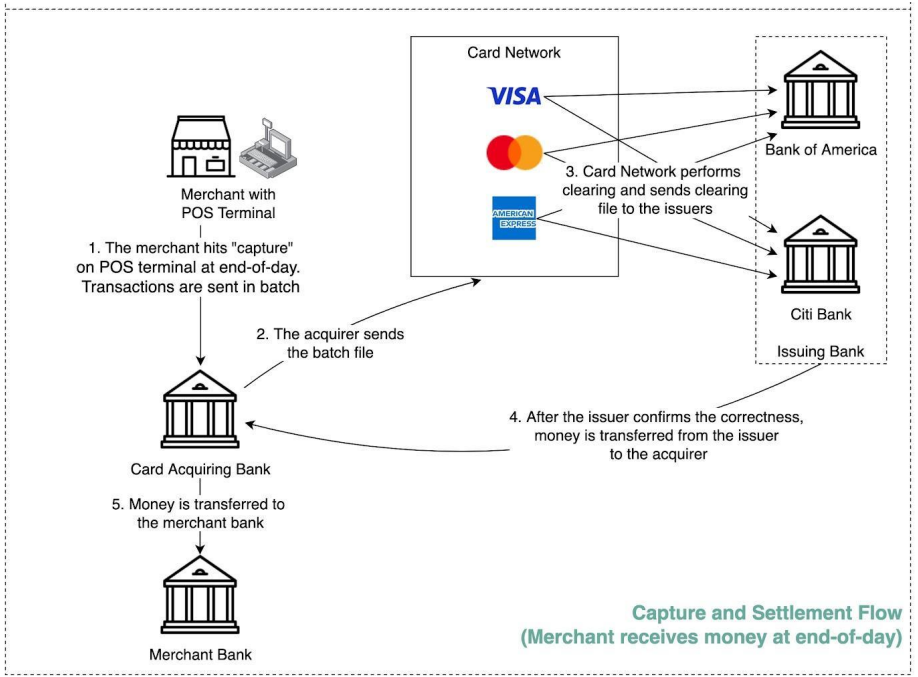
1–2 **Merchant** – At day's end, hits "Capture" → Transactions sent in batch to acquiring bank.

3. **Acquiring Bank** – Sends batch file to card network.

4. **Card Network** – Performs **clearing** across all acquirers → Sends clearing files to issuing banks.

5. **Issuing Banks** – Confirm clearing files & transfer funds to acquiring banks.

6. **Acquiring Bank** – Transfers funds to merchant's bank account.



# Clearing in Card Payments

- Netting mutual transactions to **reduce the total number of settlements**.
- Example:
  - Bank A owes Bank B $1M, Bank B owes Bank A $0.4M → Net payment = Bank A pays Bank B $0.6M.
- Handled centrally by card network to improve efficiency.

# Nepal Context

- **Global Card Networks in Nepal:** VISA, Mastercard, UnionPay.
- **Acquiring Banks:** Nabil Bank, Global IME Bank, Himalayan Bank, etc.
- **POS Terminal Providers:** Partner acquiring banks or third-party PSPs (e.g., SCT, Fonepay merchant services).
- **Settlement:**
    - Domestic card transactions – Settled via Nepal Clearing House (NCHL).
    - International card transactions – Settled via respective card network in foreign currency.

# SWIFT Payment Messaging System

- **SWIFT** = *Society for Worldwide Interbank Financial Telecommunication*
- Secure, standardized messaging network connecting **11,000+ financial institutions** in **200+ countries**.
- **Headquarters:** Belgium, operated by member banks.
- Handles **millions of payment messages daily**.
- SWIFT sends **messages**, not money — it facilitates cross-border **payment instructions**.

# SWIFT's Role in Cross-Border Payments

- Provides a **standardized format** for international payment instructions (e.g., MT and ISO 20022 messages).
- Ensures **confidentiality, integrity, and authentication** of messages.
- Acts as a **neutral intermediary** between banks.

**Message Flow Overview**

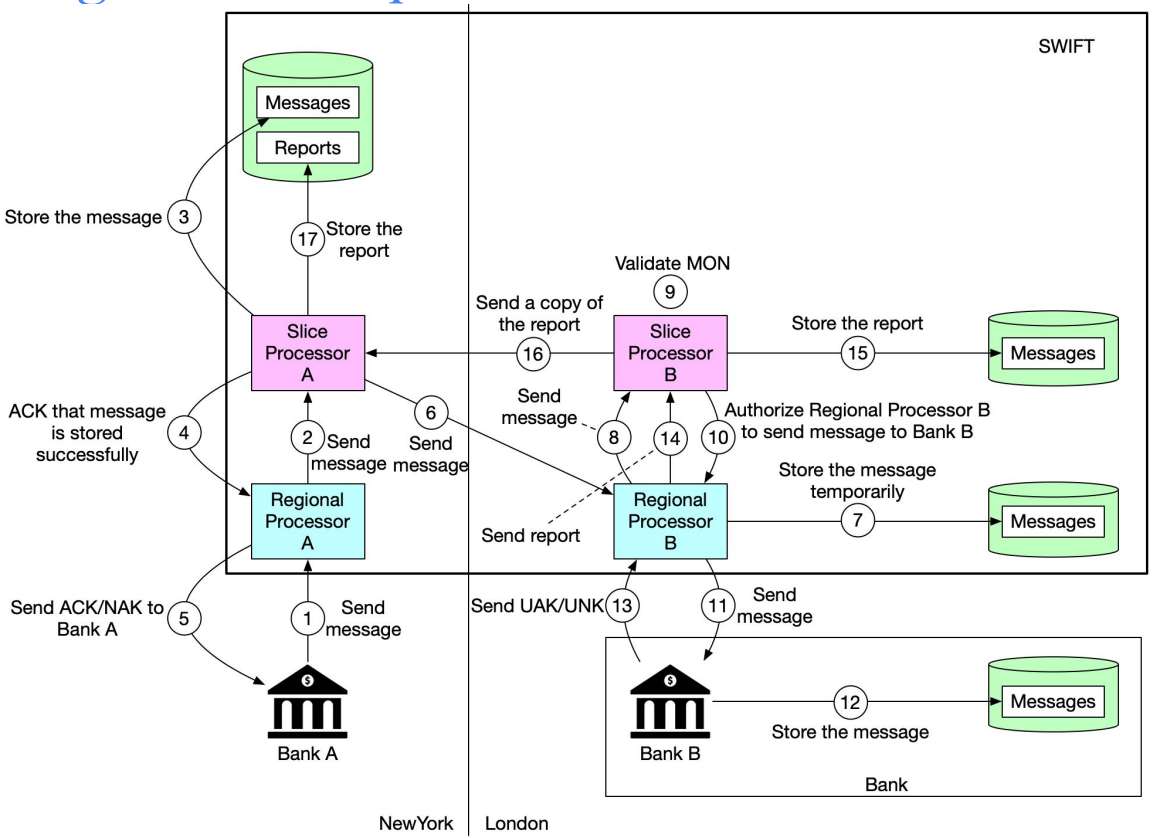
**Scenario:** Bank A (New York) → Bank B (London)

- **Regional Processors (RP):** Validate, queue, and route messages.
- **Slice Processors (SP):** Store and securely forward messages.
- **ACK/NAK & UAK/UNK:** Status signals confirming message receipt and correctness.

# SWIFT Message Flow Steps

1. **Bank A → Regional Processor A (NY) –** Sends payment details.
2. **RP A → Slice Processor A –** Validates format, queues message, stores securely.
3. **ACK/NAK to Bank A:** ACK = sent, NAK = error.
4. **SP A → RP B (London):** Message forwarded to destination region.
5. **RP B:** Assigns **Message Output Number (MON)**, stores message temporarily.
6. **SP B:** Validates MON, authorizes RP B to deliver message.
7. **RP B → Bank B:** Sends message to destination bank.
8. **Bank B:** Stores message, sends **UAK** (received OK) or **UNK** (checksum failure).
9. **SP B → SP A:** Final report sent back for storage and audit.

# SWIFT Message Flow Steps



# Acknowledgment Codes

- **ACK:** Message accepted for delivery.
- **NAK:** Message rejected (format/validation failure).
- **UAK:** User positive acknowledgment (received without error).
- **UNK:** User negative acknowledgment (received with checksum failure).

# SWIFT in Nepal's Context

- All major commercial banks in Nepal are SWIFT members.
- Used for:
    - **International remittances**
    - **Trade finance** (Letters of Credit, Bills for Collection)
    - **Foreign currency settlements**
- Key institutions: Nepal Rastra Bank (NRB) as a central player in foreign exchange management.

# Foreign Exchange Payments

- Foreign Exchange (Forex/FX) = Process of converting one currency into another during payment.
- Used in cross-border transactions where **buyer & seller use different currencies**.

**Example:**

- Buyer (Bob) pays **USD**.
- Seller (Alice) receives **EUR**.

# Example: USD → EUR Conversion via PayPal

**Step-by-Step Flow:**

1. Bob sends **100 USD** via PayPal.
2. Funds move from Bob's bank (**Bank B**) → PayPal's USD account in **Bank P1**.
3. PayPal needs to convert USD to EUR → uses **foreign exchange provider** (**Bank E**).
4. PayPal sends **100 USD** to its USD account in Bank E.
5. Bank E sells **100 USD** to its **funding pool**.
6. Funding pool exchanges 100 USD → **88 EUR** (exchange rate applied).
7. 88 EUR is credited to PayPal's EUR account in Bank E.
8. PayPal moves 88 EUR to its EUR account in **Bank P2**.
9. 88 EUR is paid to Alice's EUR account in **Bank A**.

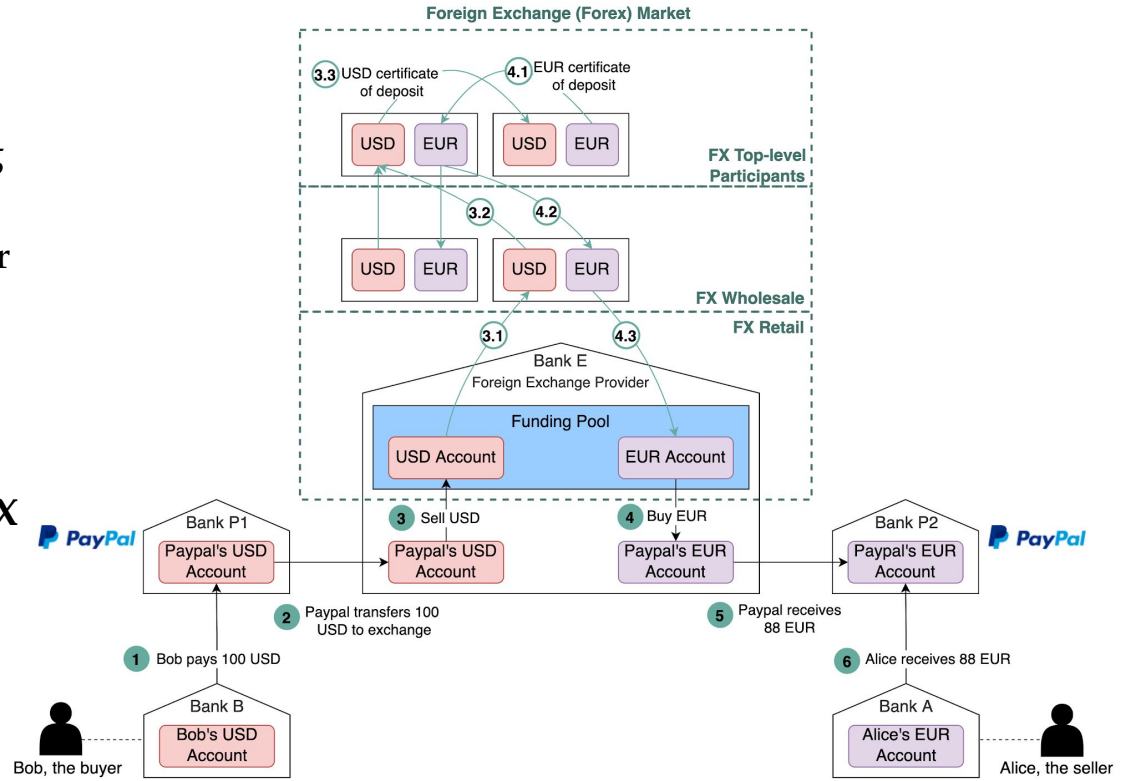
# FX Market Structure

**Three Layers of the Forex Market:**

1. **Retail Market** – Includes funding pools. PSPs like PayPal often **pre-purchase** foreign currency for efficiency.

2. **Wholesale Market** – Large transactions between **investment banks**, **commercial banks**, and **FX providers**.

3. **Top-Level Participants** – Multinational banks holding reserves in multiple currencies.



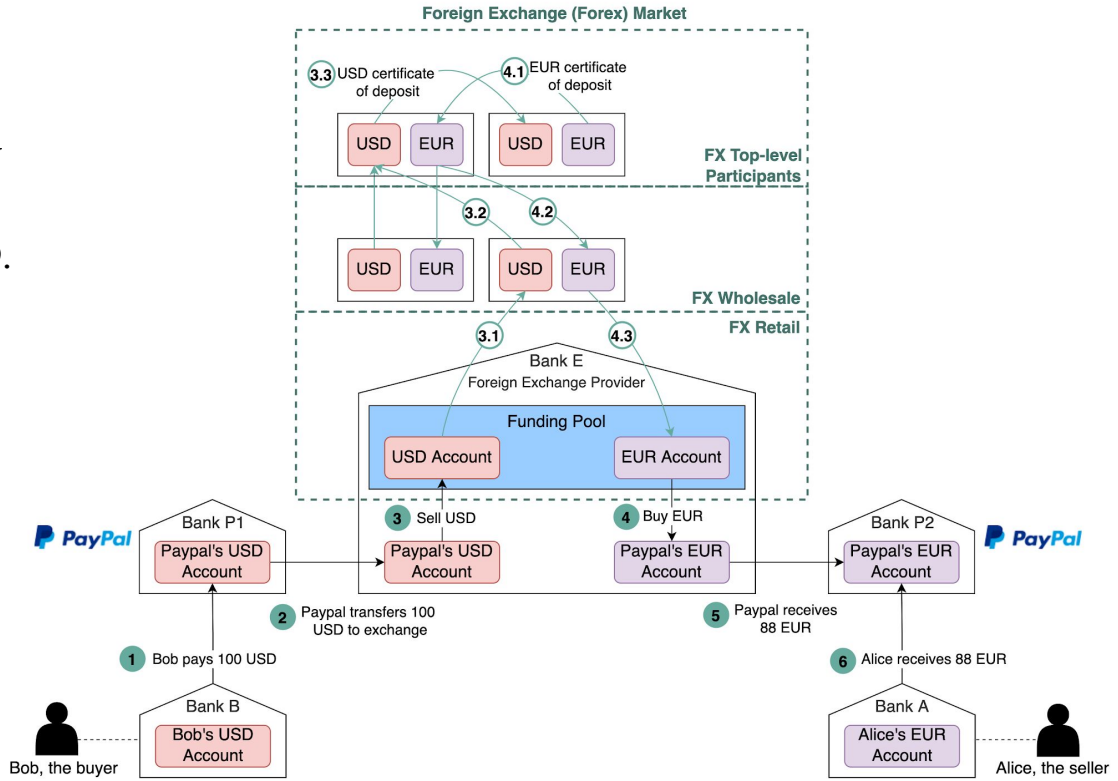
# How FX Market Liquidity Works

**Upward Flow:**

- If Bank E's funding pool runs low on EUR → Buys EUR from **wholesale market** by selling USD.
- Wholesale market aggregates orders → Goes to **top-level participants**.

**Downward Flow:**

- Top-level participants → Wholesale market → Funding pool → Retail PSP (PayPal) → Seller.



# FX in Nepal

- Foreign exchange handled by **Nepal Rastra Bank (NRB)** regulations.
- PSPs (eSewa, Khalti) mainly operate in NPR, but cross-border payments use **correspondent banks** with SWIFT + FX providers.
- Exchange rate differences and conversion fees apply for international card and PayPal transactions.

# Digital Wallets in Banking Systems

**Example: Bob using Bank of America (BoA)**

**Deposit:**

- Bob opens account **B1234** at BoA, deposits $100.
- Money stored in bank's vault, balance updated in wallet system.
- To use another bank (e.g., Citi), Bob must open a separate account.

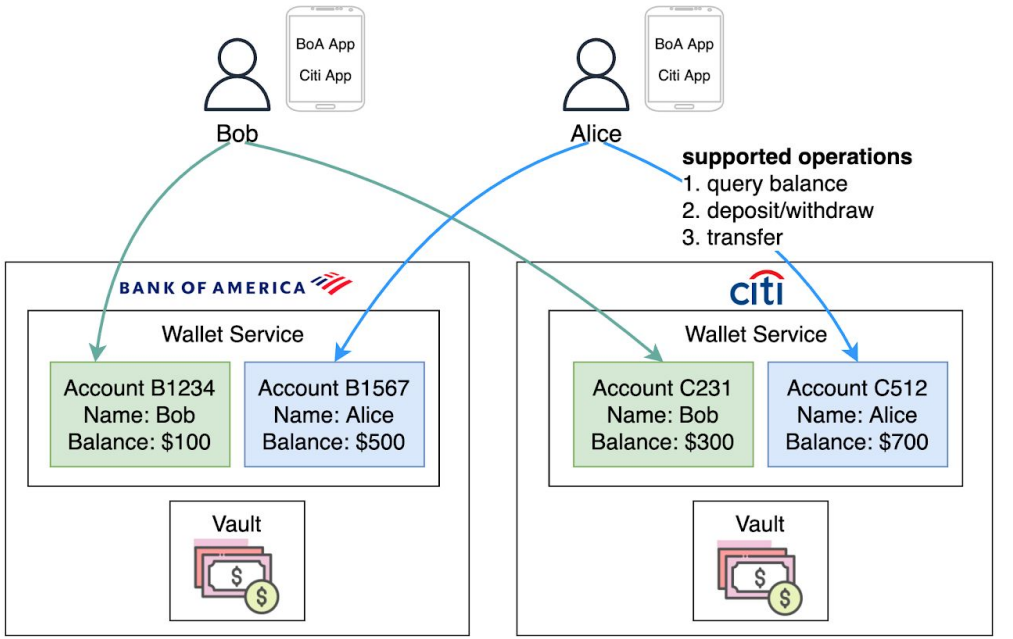
**Transfer:**

- Bob sends $50 from BoA to Alice at Citi.
- Balance changes instantly in wallet records, **but actual cash moves only after end-of-day settlement** between banks.

**Withdrawal:**

- Bob withdraws remaining $50 → Deducted from B1234 → Cash given to Bob.

# Digital Wallets in Banking Systems




Wallets in Traditional Banking Systems

# Digital Wallets on Blockchains

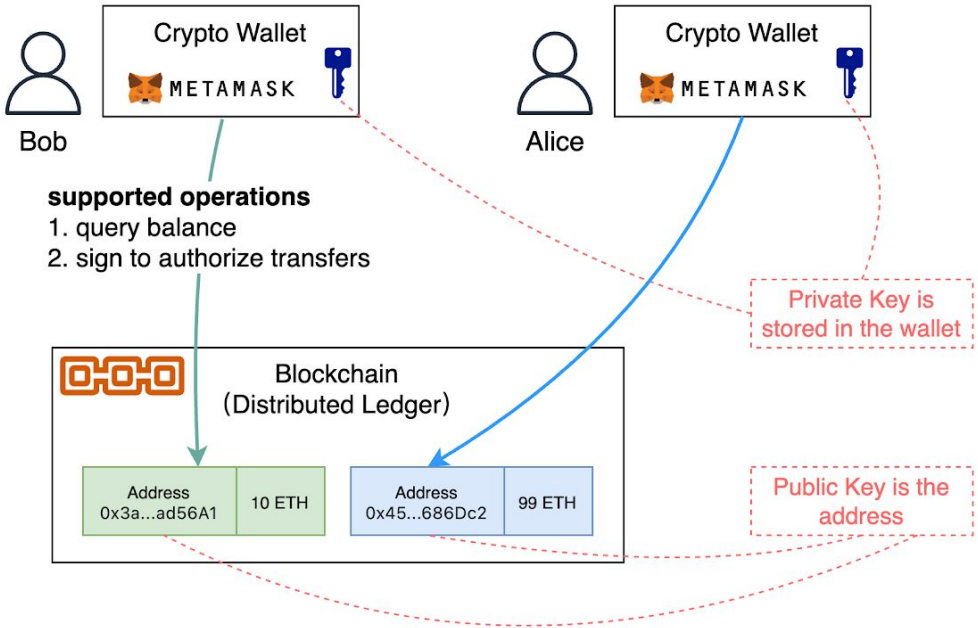
**Example: Bob using MetaMask (Ethereum Wallet)**
**Deposit/Withdraw:**

- Bob generates a **public address** and stores a **private key** in his crypto wallet.
- Can receive/send cryptocurrencies directly—no physical cash.

**Transfer:**

- Bob sends 2 ETH to Alice's address.
- Signs transaction with private key.
- Blockchain confirms transaction → Balances update in **pseudo real-time**.

# Digital Wallets on Blockchains



# Key Differences: Digital Wallets

| Feature | Bank Wallet | Blockchain Wallet |
|---|---|---|
| **Account Creation** | One per bank | One global blockchain address |
| **Settlement Time** | End-of-day netting | Near real-time (block confirmation) |
| **Interoperability** | Limited to bank's network | Global & borderless |
| **Custody** | Bank holds funds | User holds private keys |
| **Currency** | Fiat (USD, EUR, NPR) | Cryptocurrency (ETH, BTC, stablecoins) |
| **Trust Model** | Centralized (bank) | Decentralized (distributed ledger) |

# Digital Wallets: Banks vs. Blockchain

**Advantages of Blockchain Wallets**

- **Unified interface** – one wallet works globally.
- **Faster transfers** – no waiting for daily reconciliation.
- **Merges banking services** – cross-bank and cross-border without intermediaries.
- **User control** – ownership of funds via private keys.

**Why VISA & PayPal Invest in Blockchain**

- Faster settlement and reduced operational costs.
- New business models (crypto payments, stablecoin integration).
- Ability to serve **unbanked** customers globally.
- Competitive advantage in the evolving payment ecosystem.

# How Apple Pay and Google Pay Work

- Both use **tokenization** to protect card details.
- Both are **secure**, but have different storage and processing models.
- Two main flows:
    a. **Card Registration Flow**
    b. **Payment Flow**

# How Apple Pay and Google Pay Work

**Apple Pay**

1.User adds card to Apple Wallet.

2. iPhone sends card info → Bank.

3. Bank returns **Device Account Number (DAN)** to iPhone.

4. DAN stored in a **secure hardware chip** (Secure Element).

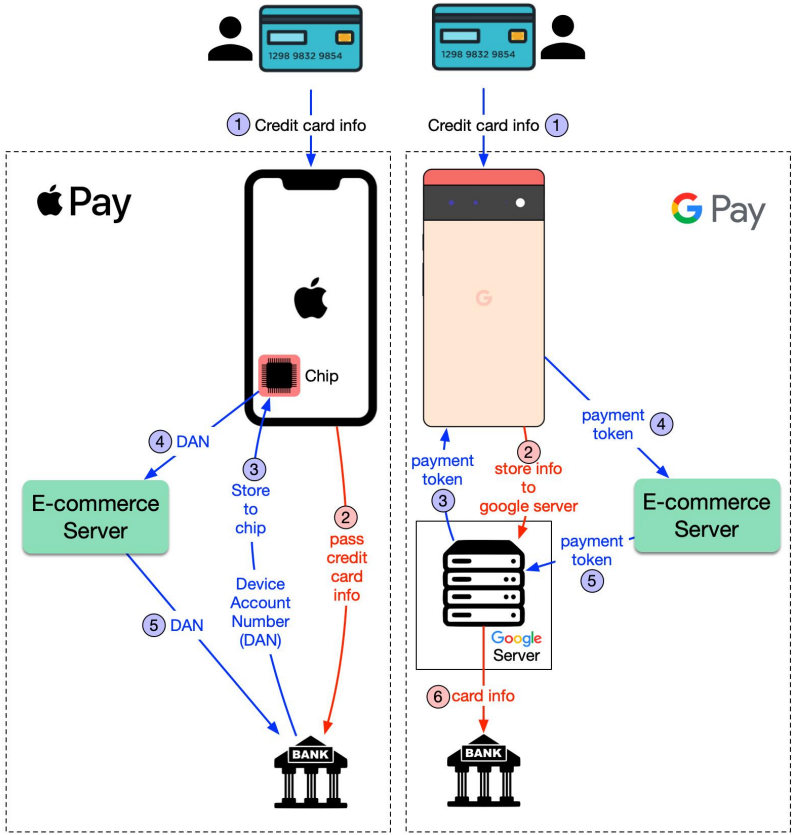
💡 **Apple does not store card info** on its servers.

**Google Pay**

1. User adds card to Google Pay.

2. Card info stored on **Google's secure servers**.

3. Google returns a **payment token** to the phone.



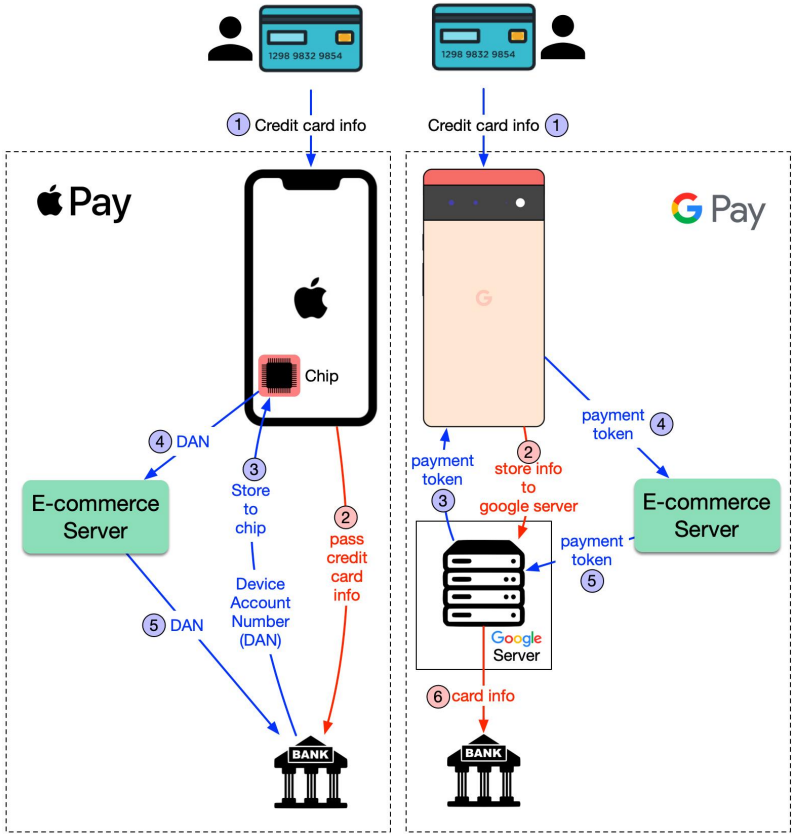
# Payment Flow

**Apple Pay**

1. User taps "Pay" → E-commerce server sends **DAN** directly to the bank.
2. Bank processes payment using the real card account linked to DAN.

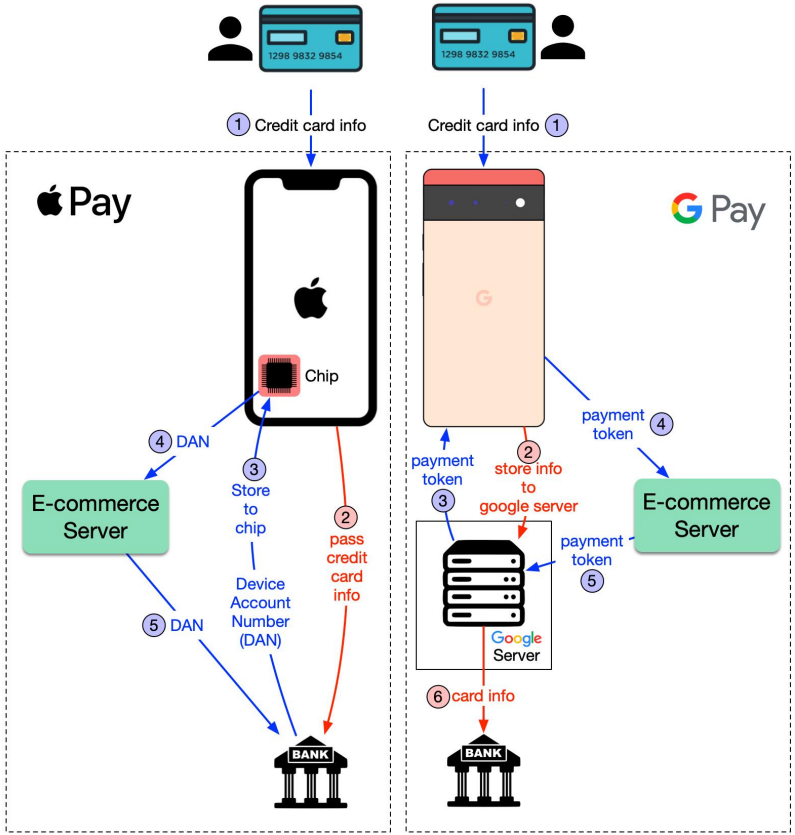
**Google Pay**

1. User taps "Pay" → E-commerce server sends **payment token** to Google server.
2. Google looks up stored card info, forwards it to the bank for processing.



# Security Notes

- **Apple Pay:** Card details never leave device; only DAN travels in network.
- **Google Pay:** Card details stored on Google servers, but **encrypted** before transmission.
- **Red Arrows in Diagram:** Show points where card details (encrypted) are transmitted over a public network.



# QR Code Payments

Widely used in mobile payment systems (PayPal, Stripe, Paytm, WeChat, Alipay, Fonepay, eSewa).

2 core questions define the type of QR payment:

1. **Who presents the QR code?** (Consumer or Merchant)
2. **Is the QR code static or dynamic?**

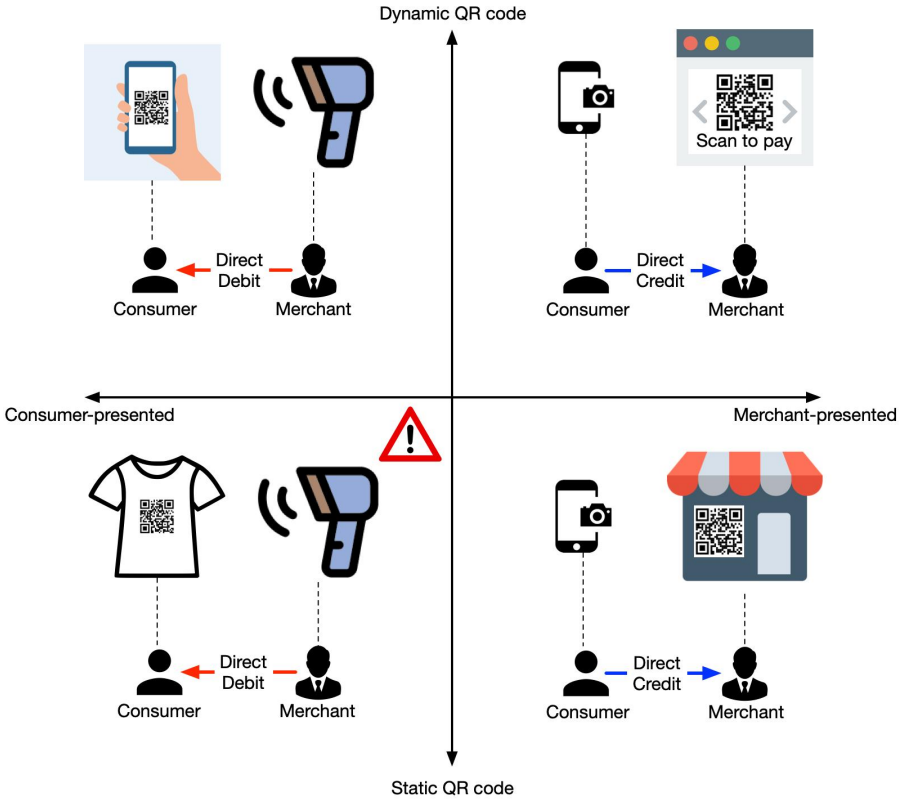
# Who Presents the QR Code?

**1. Consumer-Presented Mode**

- Consumer shows QR code → Merchant scans it.
- Allows **direct debit** from consumer's account.

**2. Merchant-Presented Mode**

- Merchant shows QR code → Consumer scans it.
- Allows **direct credit** from consumer's account tc merchant.



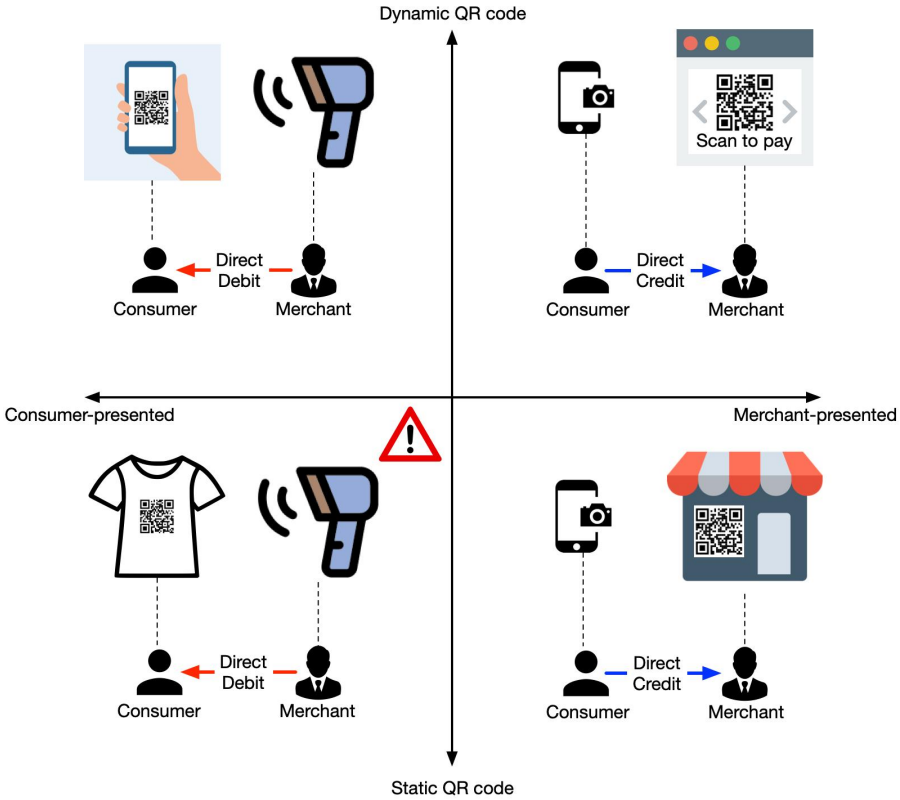
# Static vs. Dynamic QR

**Static QR Code:**

- Generated once, reused everywhere.
- Contains only account/payment ID information.
- Common in small shops, donation campaigns.

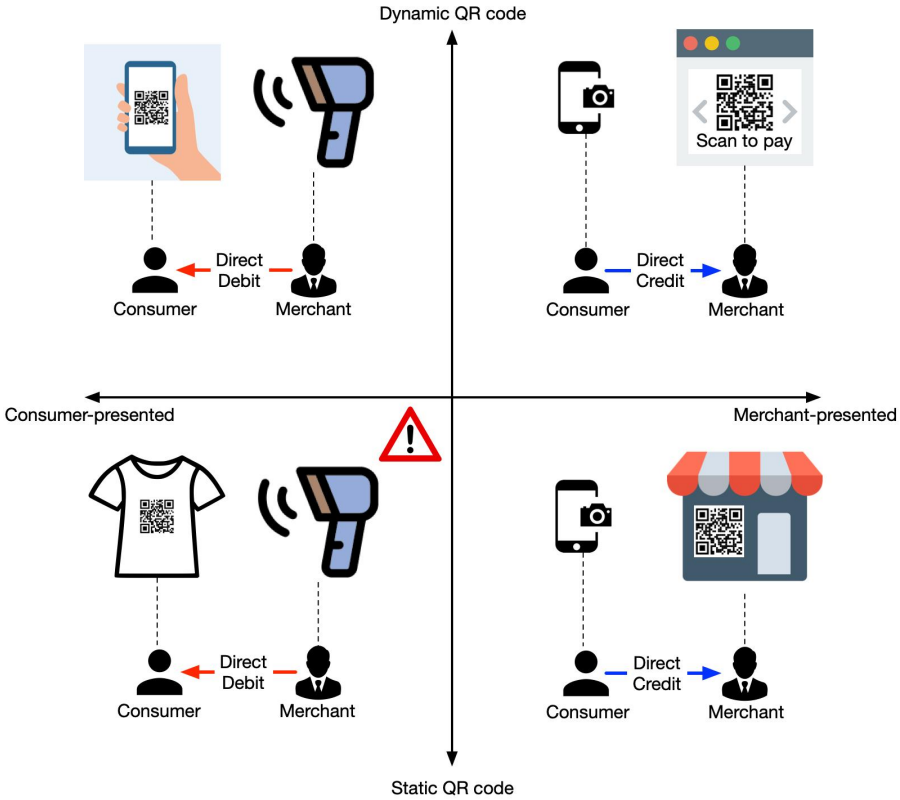
**Dynamic QR Code:**

- Generated in real-time for each transaction.
- Can include transaction amount, invoice ID, payment purpose.
- Expires quickly for security.



# The 4 Combinations

| Mode | Static QR | Dynamic QR |
|---|---|---|
| **Consumer-Presented** | Show same QR everywhere (linked to account) | QR refreshes for each payment |
| **Merchant-Presented** | Printed/sticker QR for all payments | On-screen QR with transaction details |



# Examples

**Consumer-Presented + Static:** Digital wallet QR in app reused for all payments.

**Consumer-Presented + Dynamic:** Payment app generates new QR for each purchase.

**Merchant-Presented + Static:** Printed Fonepay or Alipay QR code at store counter.

**Merchant-Presented + Dynamic:** POS machine or app displays one-time QR with amount and order ID.

**Nepal Context**

- **Merchant-Presented Static:** Common in small grocery stores with printed Fonepay QR.
- **Merchant-Presented Dynamic:** Used in supermarkets with POS systems.
- **Consumer-Presented Dynamic:** eSewa, Khalti in “My QR” section for P2P transfers.

# Thank you!
*Any Questions?*