27. Payment System

Real-life examples

- **Digital wallet**
    - Apple Pay
    - Google Pay
- **Online payment platform**
    - PayPal
    - Stripe
    - Square

Concepts

Payment options

- **Credit/Debit cards**

  Card network: Visa, MasterCard

    - POS terminal / Online
    - Contact / Contactless (NFC)
    - Online payment platform (PayPal, Stripe, Square)
    - Digital wallets (Apple Pay, Google Pay)
    - Direct debit (This method allows authorized parties to directly withdraw funds from a customer's bank account, often used for recurring payments like utility bills).
    - Checks / Money orders

  Payment Service Provider (PSP) / Payment processors / Payment gateway

- A third-party company that allows businesses to accept electronic payments, such as credit cards and debit cards payments.
- Examples: Paypal, Stripe, Square, Braintree, Adyen, Galileo

Reconciliation

- A method of bookkeeping that compares financial records that are logged internally with records that are logged in PSP, to make sure the accounting is accurate.

Double-entry bookkeeping

- A transaction always affects at least two accounts, always includes at least one debit and one credit, and always has total debits and total credits that are equal.

Account	**Debit**	**Credit**
Buyer	$5	
Seller		$5

Compliance

- Payment Card Industry Data Security Standard (PCI DSS): The standard outlines security requirements for organizations that handle credit card information.
- Anti-Money Laundering (AML): The laws prevent money laundering activities.
- Combating the Financing of Terrorism (CFT): The term prevents and detects the use of funds for financing acts of terrorism.

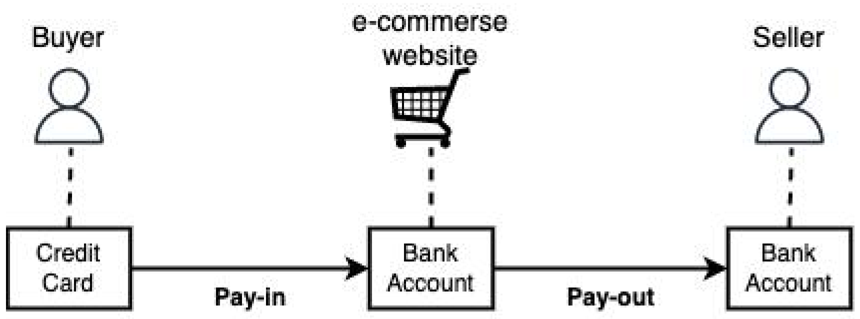
- Know Your Customer (KYC): The regulation verifys the identities of customers before allowing financial transactions.

Requirements clarification

Functional requirements

Support money movement for an e-commerce application:

- Pay-in flow: Receive money from customers on behalf of sellers once customers place orders.
- Pay-out flow: Send money to sellers once the products are delivered and money is released.



The payment system should support all the payment options

- Use third-party PSP for credit card payment processing.
- Non-functional requirements
- Fault tolerance (Failed payments need to be handled carefully).
- A reconciliation process between internal services (payment services, accounting services, etc.) and external services (payment service providers, etc.)

When a service fails, different services may run into inconsistent states. So we need to execute the reconciliation process to fix any inconsistent payment information among all services.

Estimation

Traffic estimation

Assume 1,000,000 transactions per day.

TPS = 1,000,000 transactions / 105 = 10 transactions per second.

Storage estimation

Bandwidth estimation

System interface definition

POST /v1/payments

Function: Execute a payment event.

Request body

```
{
   "buyer_info" : {},
   "checkout_id" : "",
   "credit_card_info" : {},
   "payment_orders": [
      {
         "seller_account" : "",
         "amount" : "",
         "currency" : "",
         "payment_order_id": ""
      }
   ]
}
```

Notes

payment_order_id: It is userd by the PSP as the deduplication ID, also called the idempotency key/token.

amount: The type of this field is string, not double.

GET /v1/payments/{:id}

Function: Get the execution status of a single payment order.

Data model definition

Schema

Table 1: Payment event

Description

Stores detailed payment event information.

Columns

Column Name	**Column Type**	**PK/FK**	**Description**
checkout_id	string	PK	
buyer_info	string		
seller_info	string		
credit_card_info	string		
is_payment_done	boolean		

Payment order

Description

Stores the execution status of each payment order.

Columns

Column Name	**Column Type**	**PK/FK**	**Description**
payment_order_id	string	PK	The unique identifier for payment orders.

Column Name	Column Type	PK/FK	Description
buyer_account	string		The account identifier of buyer.
amount	string		The amount of the transaction.
currency	string		The currency of the transaction.
checkout_id	string	FK	The foreign key to the payment event table's checkout_id.
payment_order_status	string		The status of the payment order. The value can be: NOT_STARTED EXECUTING SUCCESS FAILED
ledger_updated	boolean		
wallet_updated	boolean		

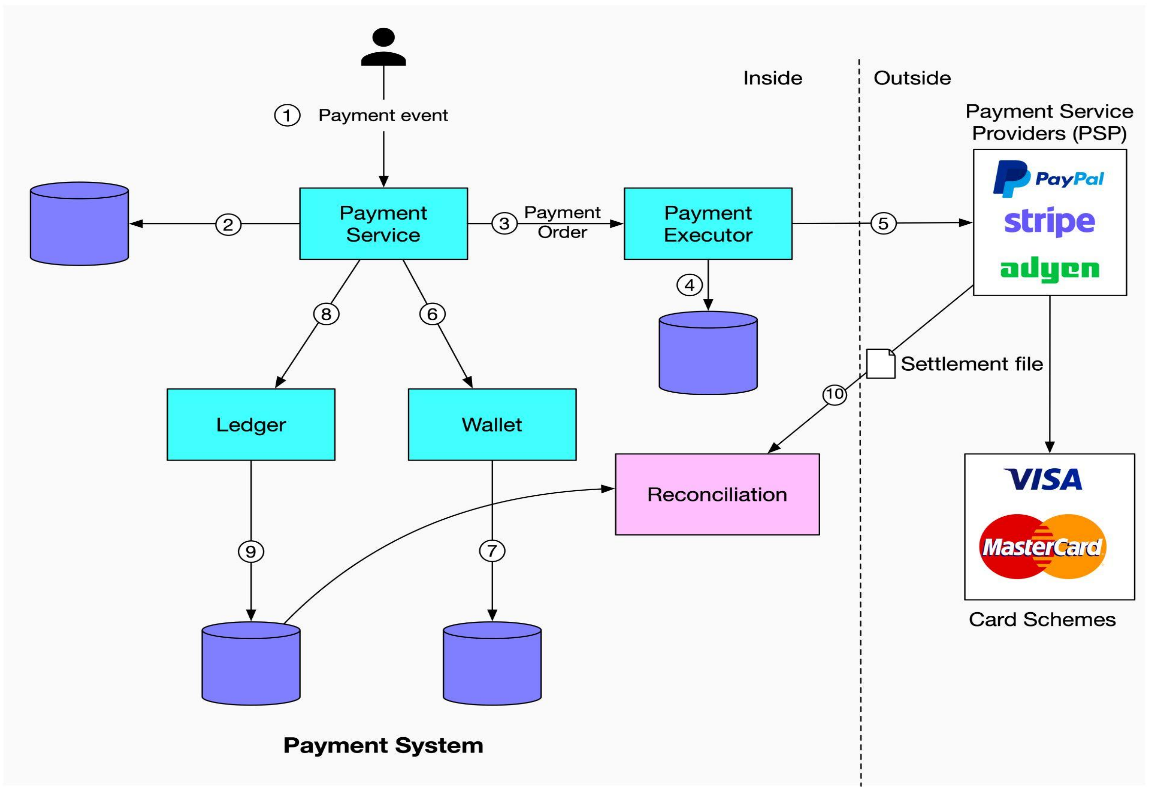
Database

Traditional relational database with ACID

Reasons

1. The financial sector requires greater uniformity rather than just accessibility.
2. Proven stability: Whether the storage system has been used by other big financial company for many years with positive feedback.

High-level design



Components

Payment service

- Accepts payment events (one event may contain multiple payment orders) from users and coordinates the payment process.

Payment executor

- Executes a single payment order via a Payment Service Provider (PSP).
- Payment Service Provider (PSP)
- Moves money from one account to another.

Card schemas

- The organizations that process credit card operations.

Ledger

- Stores transaction history.

Wallet

- Stores accounts' balances.

Reconciliation

- Parses the settlement file and compare it with the ledger system.

Key points

- ◆ Don't store the credit card information.
- ◆ Use hosted credit card pages provided from PSP
- ◆ Ledger will follow the double-entry bookkeeping principle.

Processes

Process after a user clicks the "place order" button:

(1) A payment event is generated and sent to the payment service.

(2) The payment service stores the payment event in the database.

(3) The payment service sends a payment order to the payment executor.

(4) The payment executor stores the payment order in the database.

(5) The payment executor calls an external PSP to process the credit card payment.

(6) After the payment executor has successfully executed the payment, the payment service will update the wallet to record how much money a given seller has.

(7) The wallet server stores the updated balance information in the database.

(8) After the wallet service has successfully updated the seller’s balance information, the payment service will call the ledger to save the transaction history.

(9) The ledger service adds the new transaction history to the database.

(10) The reconciliation system parses the settlement file and compare it with the ledage system.

Notes

Payment event and payment order

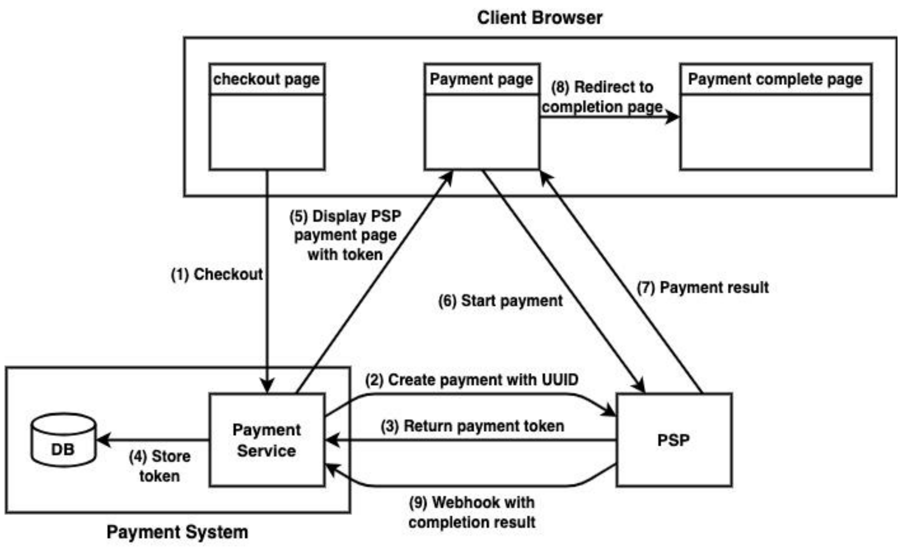
- A payment event may contain several payment orders.
- A user may select multiple products from multiple sellers in a single checkout process. The system splits the checkout into multiple payment orders.

Settlement file

- Every night the PSP or banks send a settlement file to their clients.
- Contains the balance of the bank account, together with all the transactions that took place on this bank account during the day.

Detailed design

PSP integration



Key points

- The payment system doesn't store the credit card information, it will use hosted credit card pages provided from PSP and let PSP collect credit card information directly.

Reconciliation

Purpose

- Fix the inconsistency between the financial records logged internally with the records logged in PSP externally.

Process

- Every night the PSP send a settlement file to the company.
- The reconciliation system parses the settlement file and compares the details with the ledger system.
- The financial team performs manual adjustments on the mismatches.