payment systems architecture. Content will be as follows

1. Typical Terms used In Payment Applications
2. How the card payment works ? Like Mastercard, Visa
3. Clearing and Settlement
4. System Design for payment

**Typical Terms used in Payment Applications:**

1)**Payment Gateway** : It allows to make to payment online while customer purchasing goods from any e-commerce website. It supports all types of payments like cards, wallets, net banking , UPI.

2)**Payment Service Provider** : This entity is responsible for transferring money from buyers account to the sellers account.

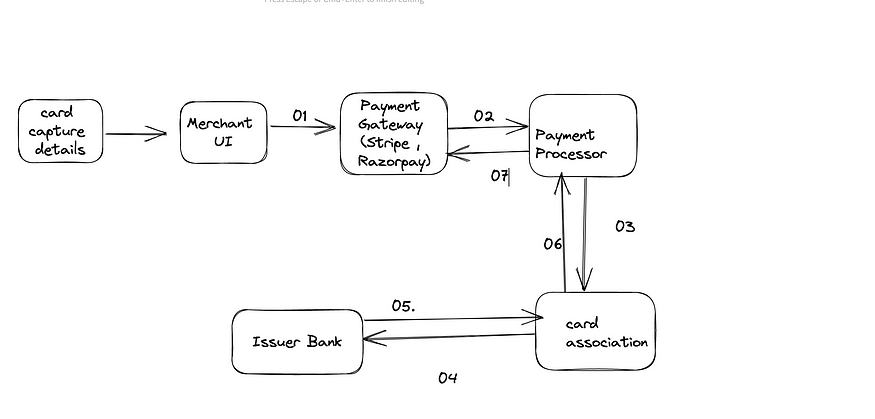
3)**Acquirer Bank**: It is the merchant’s /payee’s bank where amount will be credited once the customer buys any goods.

4)**Issuer Bank**: It is the customers bank from which amount will be debited on purchase of any goods.

5)**Card association** : It is an entity that links your card with the associated banks like Mastercard, Visa, Amex , etc.

6)**Payment Card Industry Data Security Standard (PCI DSS)** : It is the security standard that seller must have to get the credentials of having payment page generated on their website otherwise seller need to redirect the request to the payment gateway.

Now let’s understand **how card payment works** ?



Flow diag for Card payments

Let’s understand step by step each process from the above flow diagram

1. User purchases the goods and enters the card details on the checkout page provided by merchant having PCI DSS certified.

**01 Payment Gateway**: From merchant UI all the card details provided by customer are captured in form of XML/JSON and those are sent to the payment gateway like Razorpay, Stripe etc for carrying out the transaction.

**02 Payment Processor**: In payment systems the message format used is ISO Std 8583 that understands ETF switch). In this step, payment processor is responsible for converting XML/JSON message to ETF switch message format.

**03 Card Association**: This entity is responsible for transferring the card captured details to the associated banks . It maintains the directory with the first 4 digits on card and mapped banks to it and based upon the details that are presented by customer , it send the message to the associative banks. For eg: Mastercard, Visa etc

**04 Issuer Bank :**Issuer Bank will validate the card details presented by card association. It performs the following checks

i)Validate Card details

ii)Validate Card expiry

iii)Check customer balance

**05 Accept Transaction/Reject Transaction** : Based upon the checks mentioned in (04) It either accepts the transaction if all the checks are or rejects if any of those fails and sends the message back to the card association for transaction acceptance or rejection

**06/07 Sell Completed/Failed**

**Clearing and Settlement**

There are two standard protocols followed in clearing and settlement of funds.

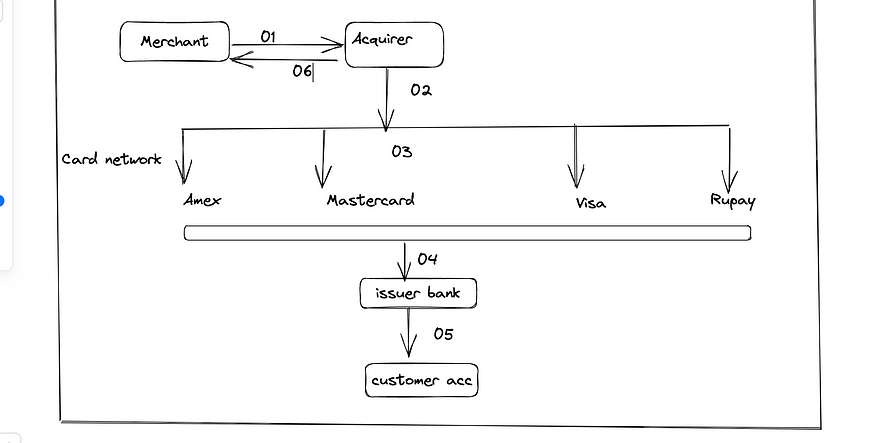
1. Dual Message System
2. Single Message System

Terms to Note:

**Clearing** : A signal indicating that funds should be sent from issuer bank to acquirer bank.

**Settlement** : A settlement indicates actual movement of funds overnight in batches.

Let’s understand dual message system clearing and settlement process:



Clearing and Settlement

Let’s decode the entities present here :

1. **Merchant** : The clearing instructions are formatted and sent to Acquirer bank (once the transaction approved by the issuer bank) at the specific time.
2. **Acquirer Bank** : The bank takes all the transaction in batches and sort them before sending to card network.
3. **Card Network**: It is responsible for forwarding the transaction in batches to the issuer bank.
4. **Issuer bank** : It is responsible for reading each transaction and further performing credit/debit the accounts. Card network then settles the fund cleared from issuer bank to acquirer bank.
5. **Acquirer Bank :**It settles the fund to merchants account.

**System Design for card payment processing**

***Functional Requirement***

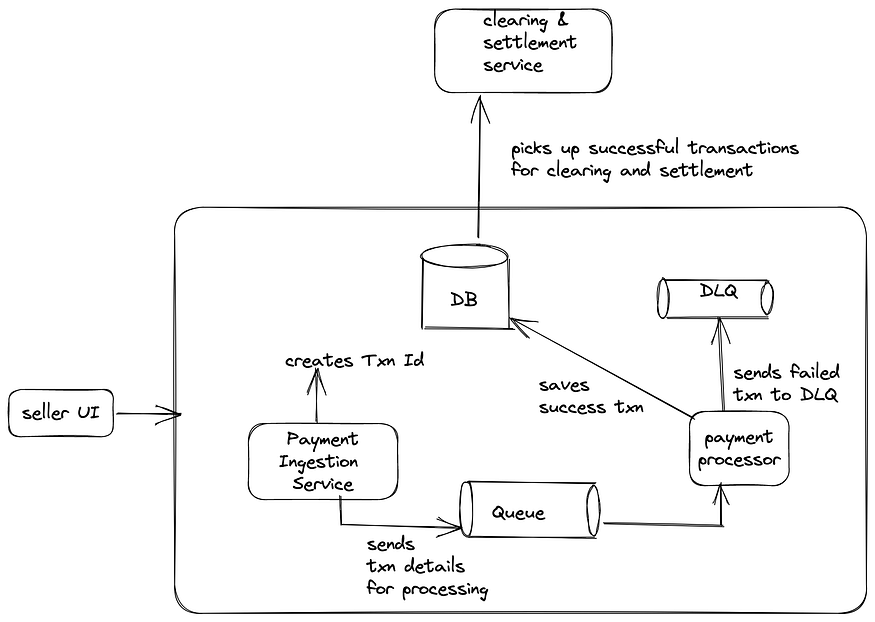
1) System should support multiple payment modes like Net banking, UPI, wallet payments

2) System should be secured

3) System should avoid duplicated payments

***Non Functional Requirement***

1. System should be highly available, scalable.
2. System should be consistent
3. System should be resilient



Payment System High Level Design

Let’s decode the architecture flow from above diagram

1. **Seller UI** : As mentioned earlier, seller will enter the card details for checkout
2. **Payment Gateway**: It will capture the card details and send to payment ingestion service.
3. **Payment Ingestion** : This service will create the unique transaction id which will be use to identify the payment in any case of failure or success.
4. **Messaging Layer**: The transactional message is send to the message queue which will be picked up by payment processor for conversion of messages to ISO std format that understands ETF switch.
5. **Payment Processor** : Processing of messaged from XML/JSON to ETF switch formats and send the details to Card Networks. If transaction is success , transaction is stored in database .If the transaction is failed then it is sent to DLQ for further retries/analysis purpose
6. **Data Layer** : Structured database used for achieving ACID properties . It will be holding all the relevant details against the particular transaction.
7. **Clearing and Settlement :**Final step of credit/debit merchant’s account.

Now , database in the system can be more under stress due often querying for the transactions details. In order to make our system highly available and scalable , I see our data layer will be the bottleneck. How can we make it more scalable considering the architecture above ?

Solution would be to partition the data based upon date so that all the transaction would be partitioned as per dates and to add further for each dates data can be sub-partitioned based upon modes of payments like UPI, Net Banking , wallets etc.

That’s all for this blog — I hope you might have gain insights on payment systems and how they typically work.

Thanks.