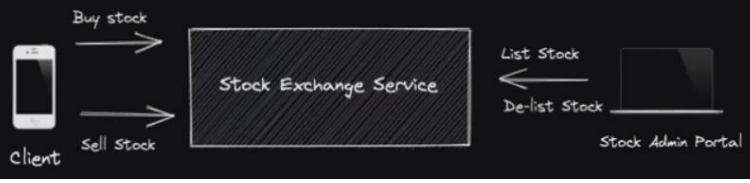
Design Stock Exchange

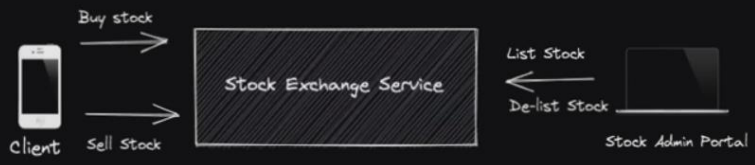
why is this problem challenging

* Financial System
* scale of stock buy/sell orders.
* Fairness in presence of retail & HFTs
* Handling limit order - so ensuring best deals.
* Creating a matching engine for trade matching.

High Level System Design



Requirements



Functional Requirement

* Sell/buy is the key feature to design
* users specify a price for buying/selling stocks
* orders are served as limit orders
* orders not served are valid till market ends
* 10 Hr market active time.
* Payment integration with 3rd party providers.
* Partial order matching is allowed.

Out of scope:
* updating price candle stick mark
* list/delist stocks

Non Functional Requirements

* highly available
* highly consistent
* trade fairness
* Scale:
  --> orders per second: 60k (buy + sell)
  --> stocks: 1000

Capacity Estimations

Database Schema

Order Request:
* order-id (PK) (Partition key)
* customer-id
* stock-id
* quantity
* price per unit
* order-type: (buy/sell)
* status:
  --> failed
  --> in-progress
  --> completely filled
  --> partially filled

Matched Orders:
* match-id (PK)
* buying order-id
* selling order-id
* sale quantity
* sale price

Stock Details
* stock-id (PK)
* stock metadata

APIs

* buy(customer-id, stock-id, quantity, price)
* sell(customer-id, stock-id, quantity, price)
* matchOrders(buyOrderId, sellOrderId, quantity)
* getOrderStatus(orderId)

Storage Estimates

Order Requests:
--> (100B x 60K x 86400 x 365) = ~ 100TB/year
--> ~ 500TBs for 5 years

Stock Details
--> (100B x 1000) = 100kB