#ASLI ENGINEERING

How Razorpay Scaled their Notification System

SWIPE

BY

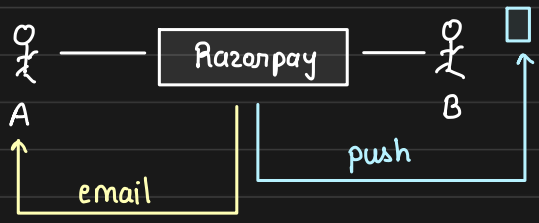
ARPIT BHAYANI

Scaling Notifications at Razorpay

Notifications System → Outbound communication

eg: SMS, Email, Push, Webhooks

Meeting SLA is super important



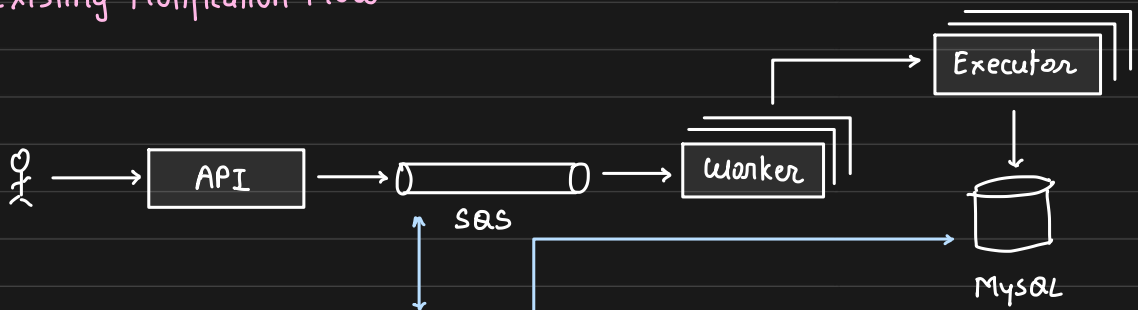
Razorpay notifies their customer payer and payee about the transaction

↳ SMS, Email, Push

↳ Webhook *

Given that the money is involved, maintaining a strong SLA is critical!

Existing Notification Flow



Schedules the message to be retried.

All about guaranteeing delivery { Executor is writing to the DB about notification sent

p99 drops from 2sec to 4sec for > 1000 TPS

Challenges while scaling

1. Read load on DB during peak
2. Scaling of worker POD is limited to IOPS of database
3. Surge during special events were hard to handle

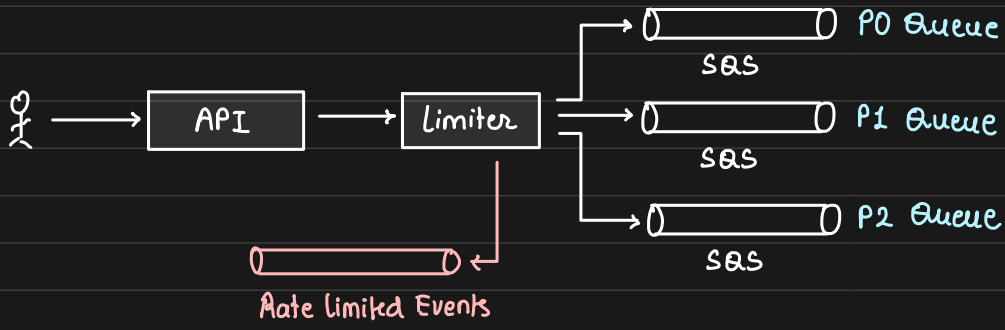
Rearchitecting Notification System

1. Prioritizing Incoming load

- Not all notifications are equal - Transactional >> Marketing
- One type of notification should not affect others

Solution: Queues

To ensure one customer's events are not affecting others, we rate limit. Each queue, event, customer has some configurable rate limit, breaching which the message goes in a separate queue

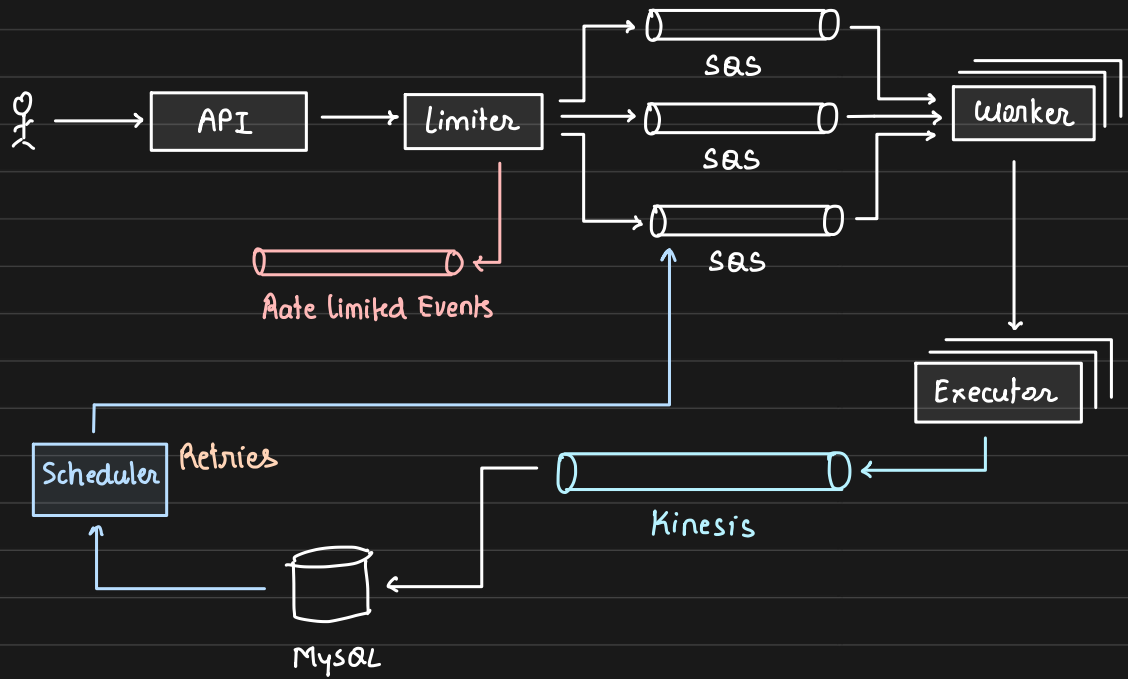


2. Reducing DB Bottlenecks

As load increased, the worker/executor increased but DB is not elastic and hence its IOPS became the bottleneck.

Vertical scaling is costly and we cannot do it forever

Solution: Write to database asynchronously



Observability: Dashboards on Grafana, Alerts, Anomalies, Health, Success rate, SLA measures