DESIGN A NOTIFICATION SYSTEM

A notification alerts a user with important information like breaking news, product updates, events, offerings etc.

A notification is more than just a mobile push notification.

3 types of notifications -

(i) Push notifications

(ii) SMS messages

(iii) Email

Clarification questions to ask the interviewer -

1) What types of notifications

2) Is it a real-time system?

We want user to receive notifications as soon as possible. However, if the system is under high workload, a slight delay is acceptable.

3) Supported devices? Android, IOS, Laptop/Desktop

4) What triggers the notifications?

Triggered by client applications. Can also be scheduled on the server-side. (Automatic/scheduled/manual)

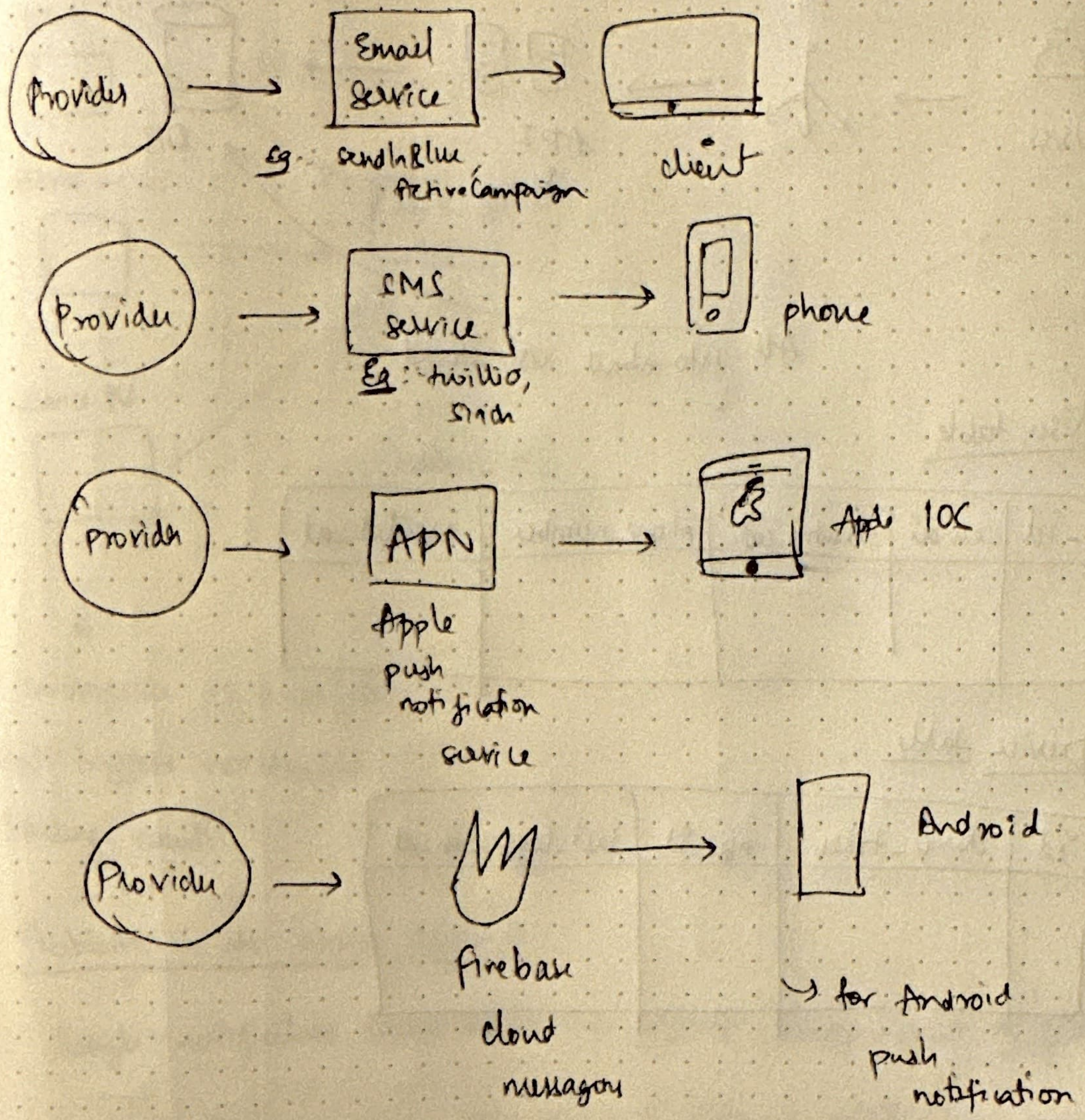
5) Will users be able to opt-out?

Yes, users will no longer receive notifications.

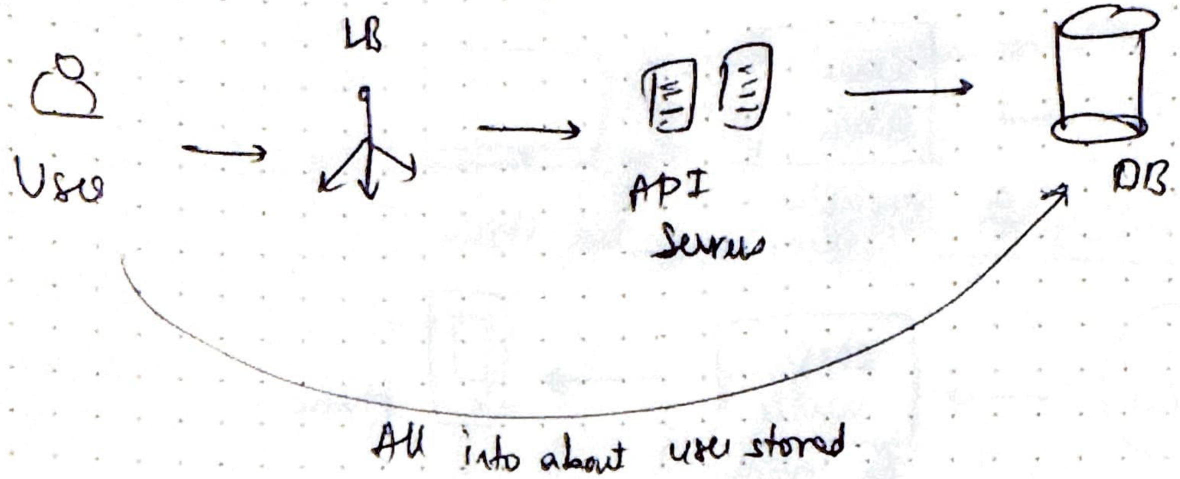
6) How many notifications each day?

10×10^6 push notifications, 1 million SMS messages and 5 million emails

High Level Design



Gathering Information



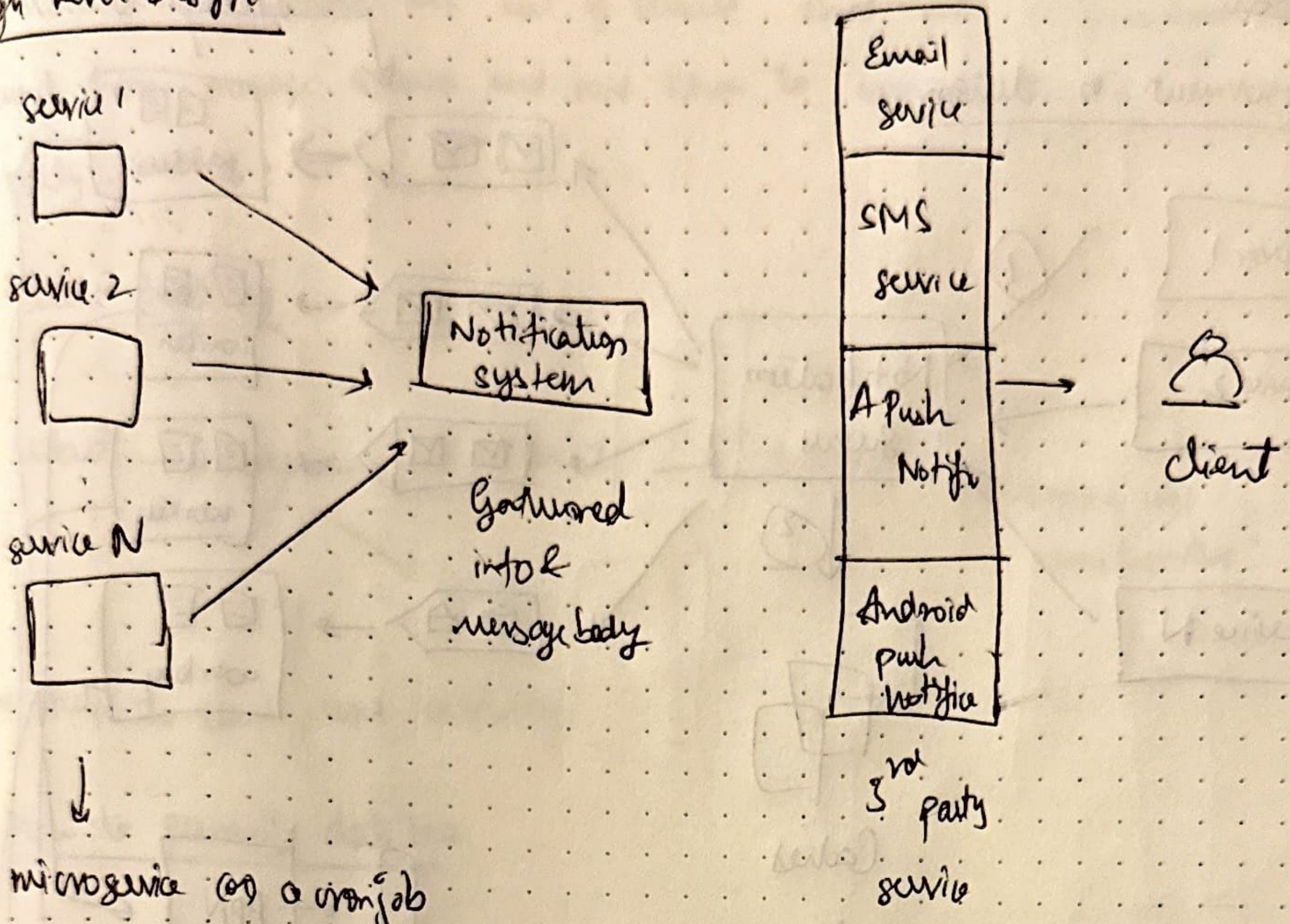
User table

user_id	email	country_code	phone_number	created_at

Device table

id	device_token	user_id	last_logged_in_at

High Level Design



a microservice (or) a cronjob that triggers notification sending events

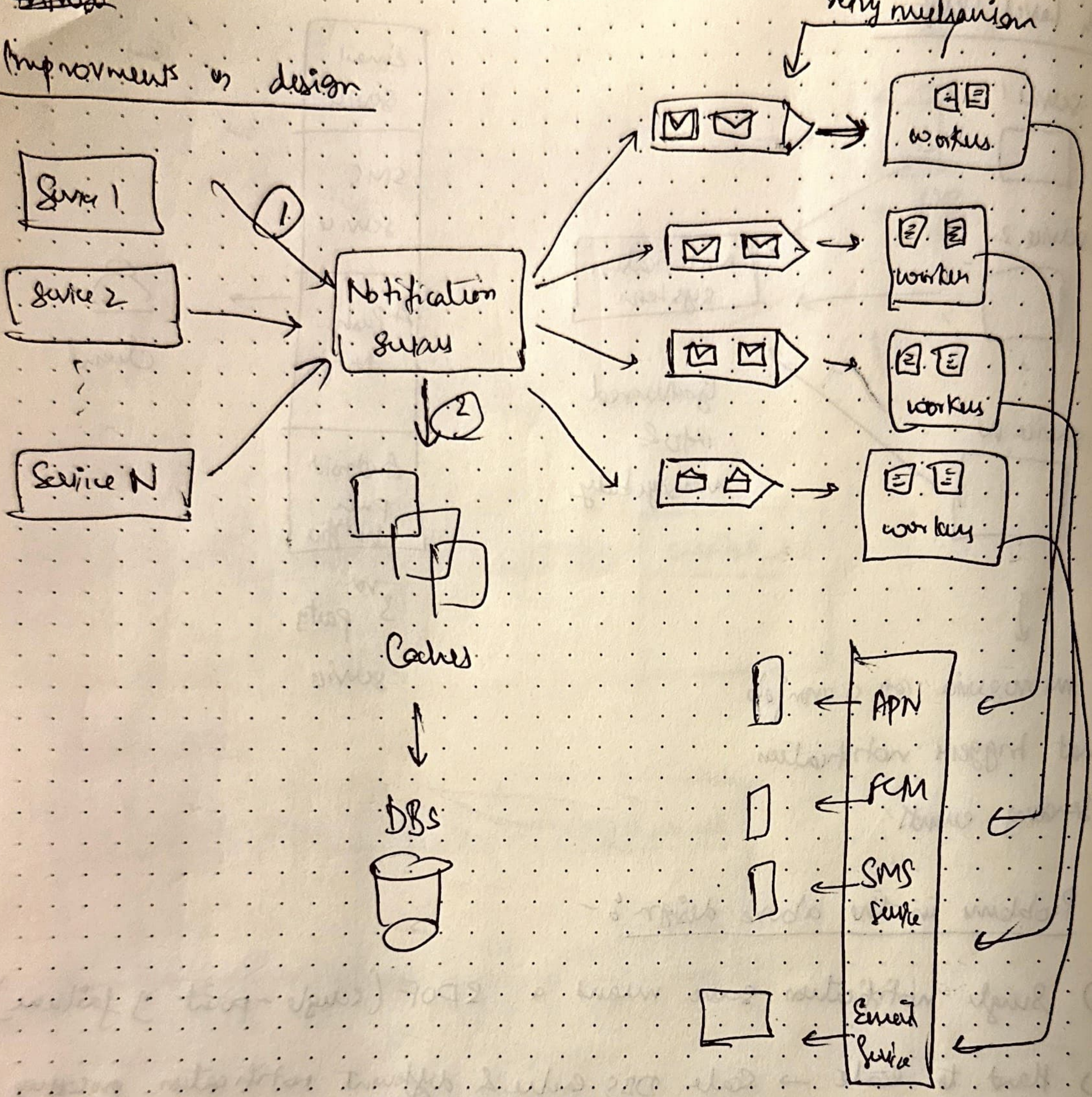
Problems in the above design :-

(i) Single notification server means a SPOF (single-point of failure)

(ii) Hard to scale → Scale DBs, cache & different notification processing components independently

(iii) Performance bottlenecks → Processing and sending notifications can be resource intensive. For example, constructing HTML pages and waiting for responses from 3rd party services could take time. Handling everything in one system results in system overload, especially during peak hours.

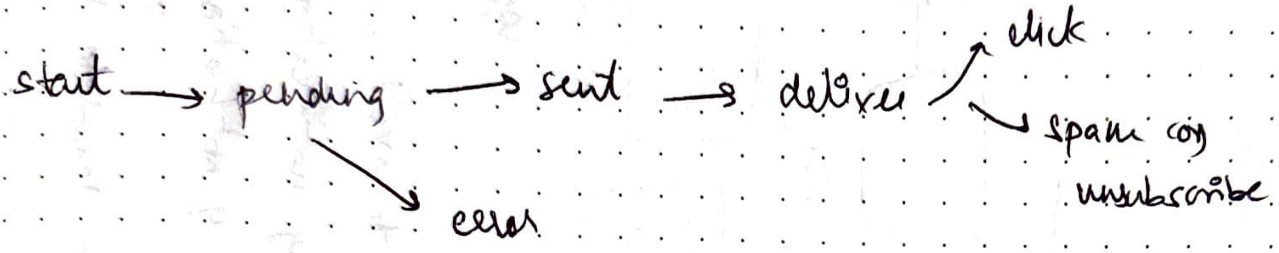
Improvments in design



Message queues: They remove dependencies between components. Message queues serve as buffers when high volumes of notifications are to be sent out. Each notification type is assigned with a distinct message queue so an outage in one 3rd party service will not affect other notification types

→ Workers → Workers are list of services that pull notification events from message queues and send them to corresponding 3rd party services.

Event tracking

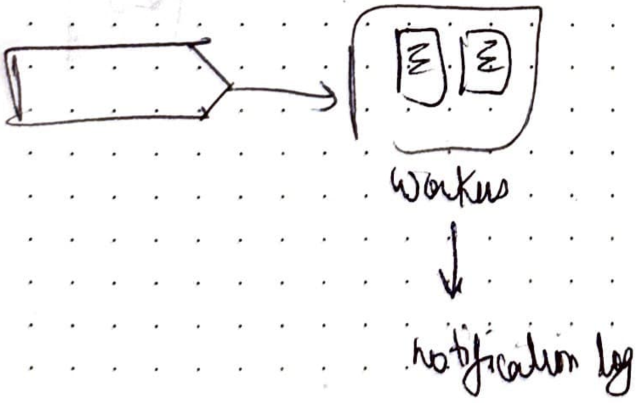


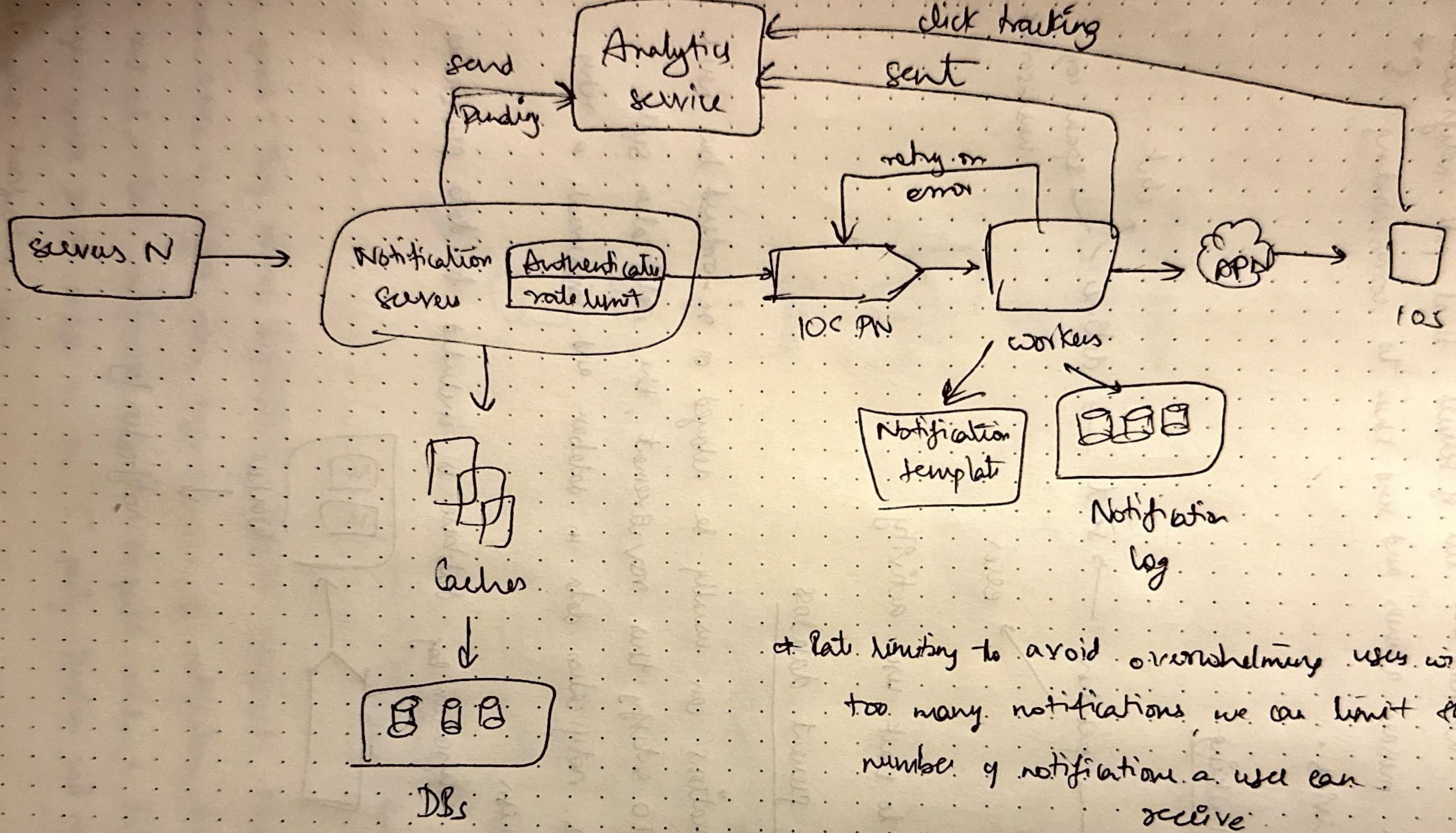
* Help to track user activity.

How to prevent data loss

Notifications can usually be delayed or re-ordered but never lost. To satisfy this requirement, the notification system persists notification data in database and implements a retry mechanism.

The notification log database is included for data persistence.





- Rate limiting to avoid overwhelming users with too many notifications, we can limit the number of notifications a user can receive
- Notification template to avoid building every notification from scratch