#ASLI ENGINEERING

Uber's Emergency SOS Service

SWIPE

BY

ARPIT BHAYANI

Uber's Emergency SOS Service

Emergency Button on Uber app is life saving and when in an emergency, every second matters

When the emergency button is tapped

1. critical details are captured and sent to the nearest local police
   - current location
   - trip details
   - vehicle details
   - driver / rider details
2. the location is continuously sent and kept updated
3. an optional SMS is compiled and sent to the First Responders
4. Uber's internal support team is notified and they keep an eye on the vehicle movement & co-ordinate

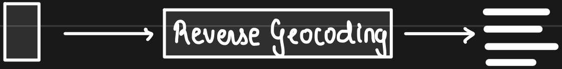
Capturing location

When SOS is pressed, we capture the GPS location.

To make it simple for the first responders, and the rider to communicate we show and send the address

to local police authority

on the phone



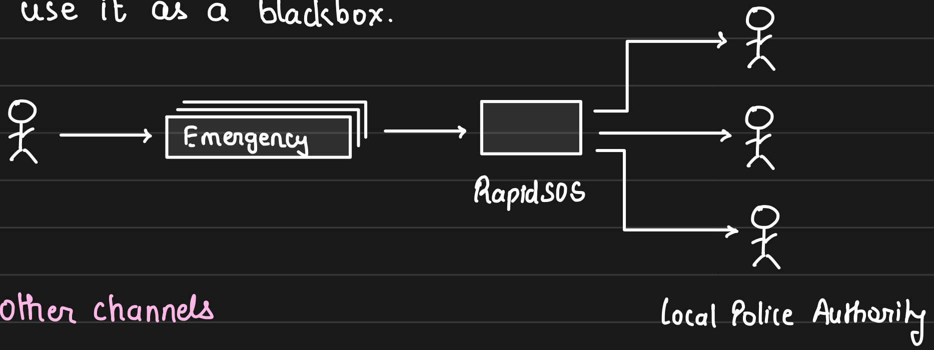
also, location should be continuously captured and broadcasted

ARPIT BHAYANI

How to notify local police?

A third-party service named RapidSOS comes handy

RapidSOS provides API to send data to the local police authority and we use it as a blackbox.

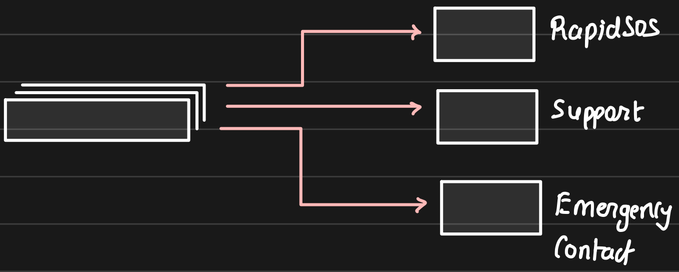


Notifying other channels

We should not only notify local Police Authority
but also internal support and emergency contacts

* all these communications should be decoupled
so that failure of one does not affect other

So, a simple broker can be used here to decouple the notifications



* Just a schematic repr

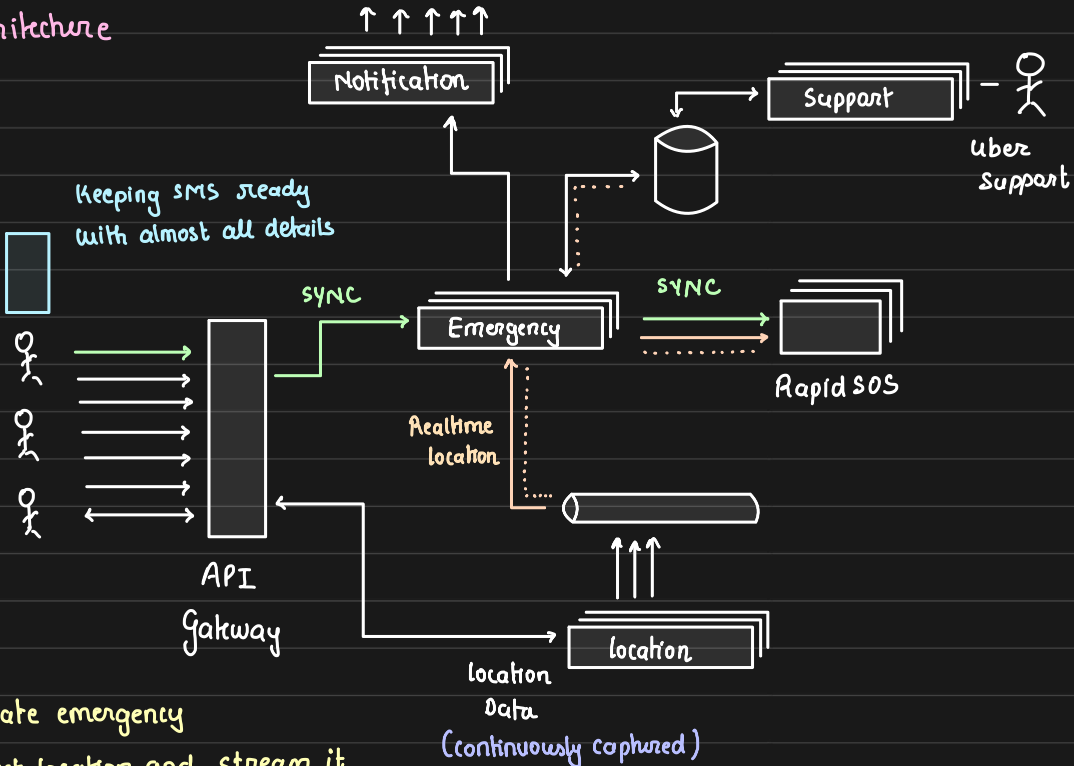
ARPIT BHAYANI

Reliability and Availability

Services involving emergency situations should be highly available & reliable

1. Kafka - persistence, at-least once delivery
2. if kafka down - fallback on sync APIs
3. no SPoF - horizontally scalable internal services & fallbacks for 3PT

Architecture



1. Create emergency
2. Collect location and stream it
3. Notify on various channel - Rapidsos, Support, Emergency Contacts

ARPIT BHAYANI