Major project → Payments App

→ Minorproject Application.java

Minor project

Single service
- → Transaction ops
- → Book related ops
- → User related ops

In a single service / Module / project

Different comps are tightly coupled

Monolithic architecture

||

microservice

100M
1000
50 ops
500 ops

User service

Signup

Book a cab

Find a cab

Update profi

Make payment

Uber

Monolithic Architecture -

1. If one service goes down and need to redeployed, others will face issue as well.
2. If you want to scale one service, you will have to scale everything

Sep Repo

single Repo

Monorepo but deployed as
microservice

multiple Services

Distributed

diff service

in terms of

Brainstorm

E-wallet application (Paytm → Paytm Wallet)

(Amazon —)

User Service — Module

Wallet Service

Transaction Service

Notification Service

Payment Gateway (×)

Service.java

Wallet Controller — Wallet Service — Wallet Repo

entities

Wallet Service

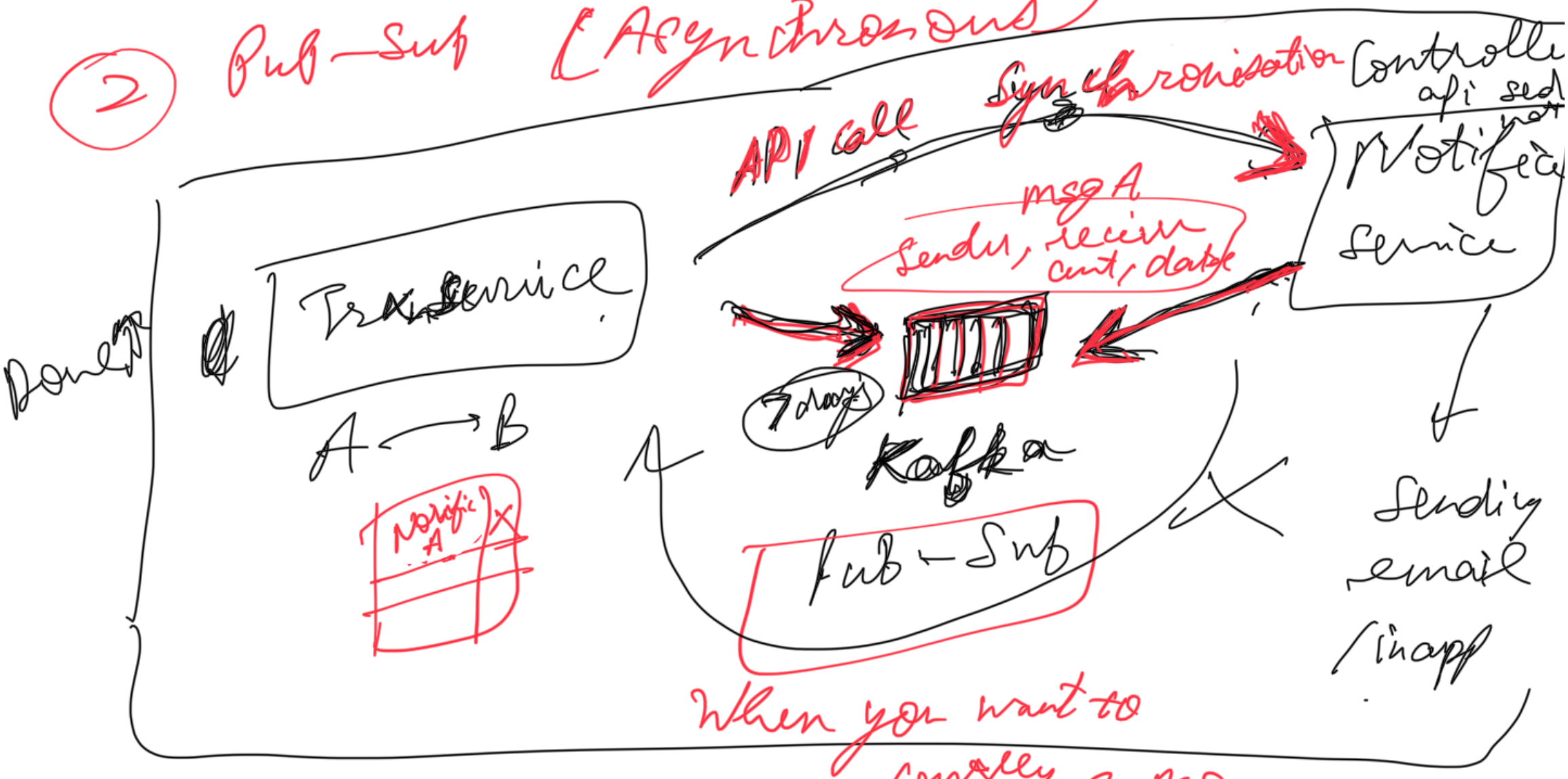
[illegible] microservice arch. / inter service

communication is very imp.

(2)

(1) 1:1 (Synchronous)

(2) Pub-Sub (Asynchronous)



When you want to convey a msg to and not to get data from

User A wants to send money to User B

Notification Service is down

1 → User Service

1
- POST
- GET
- Delete
- Put

User
- phone
- name
- pass

Module

Parent (Major Pj)
- wallet ser
- APP
- User serv

User DB

Flow

User onboarding Flow

User Service

POST

DB

User-created

Kafka

Notific service

Wallet Service

Wallet DB

2

Wallet Service →

↳ GET /

↳ POST /

Pay

Wallet

{ User Id
balance → Ratatype

↳ PUT

↳ Delete

walletId — Long

Float/double, but companies use

Integer/long

round it off → to two decimal

use lowest possible den

100 rs

100.50

10050 paise

100.50 equals (bal)

100.50 ≠ ا

(3) Transaction Service → Transaction & senderId

→ POST

→ PUT

→ GET

→ Delete

Receiver id
created date/time
payment type
→ status
msg

Asynchronous

A → B

transaction

Pending

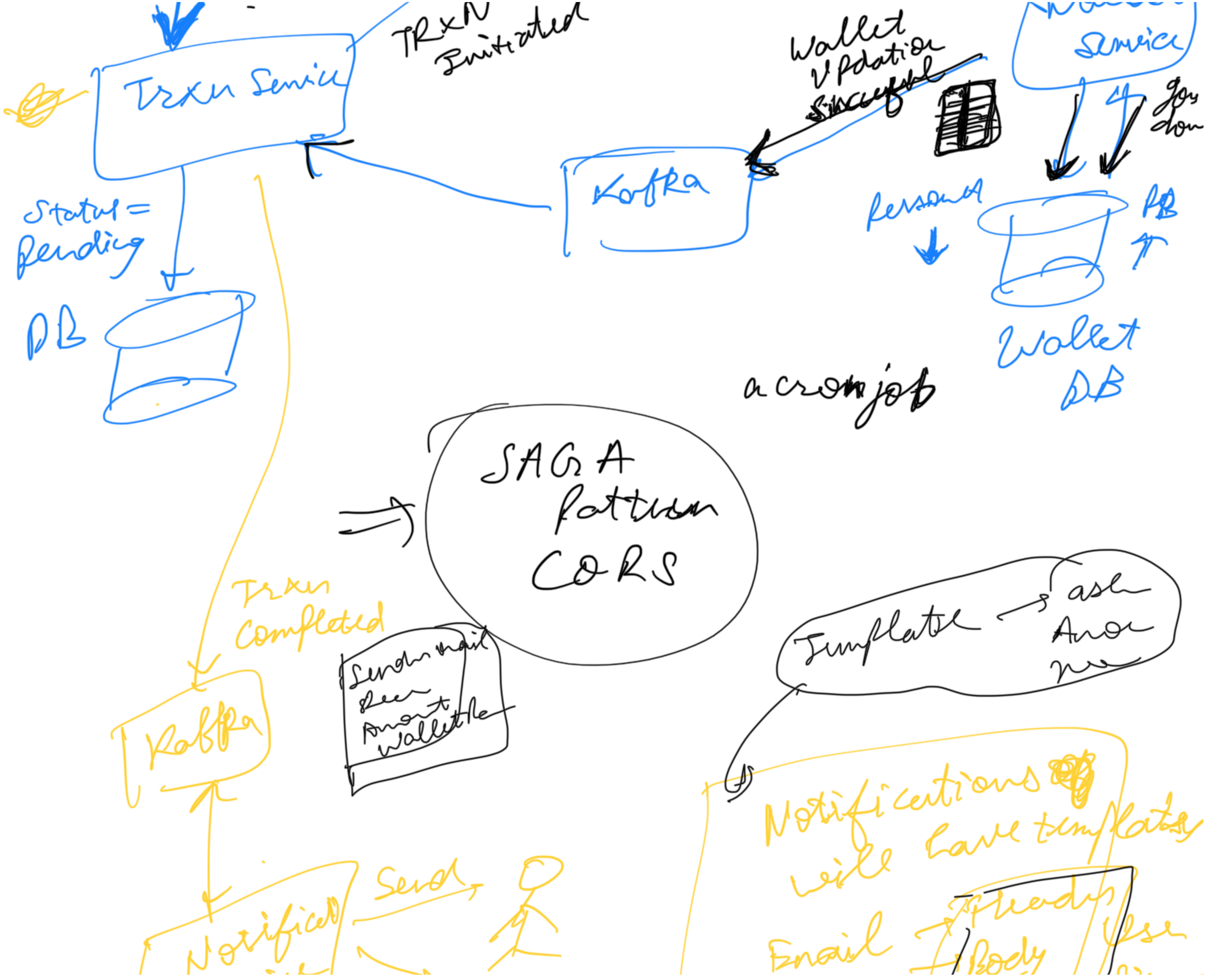
Status

Flow

→ producer
→ consumer

Kafka

→ Consum
→ producer


TRXN Initiated
Trxn Service
Wallet Updation Successful
Service
Kafka
Status = pending
DB
Wallet DB
a cronjob
SAGA Pattern CQRS
Template
Trxn Completed
Kafka
Send
Notifications will have templates
Email
Body