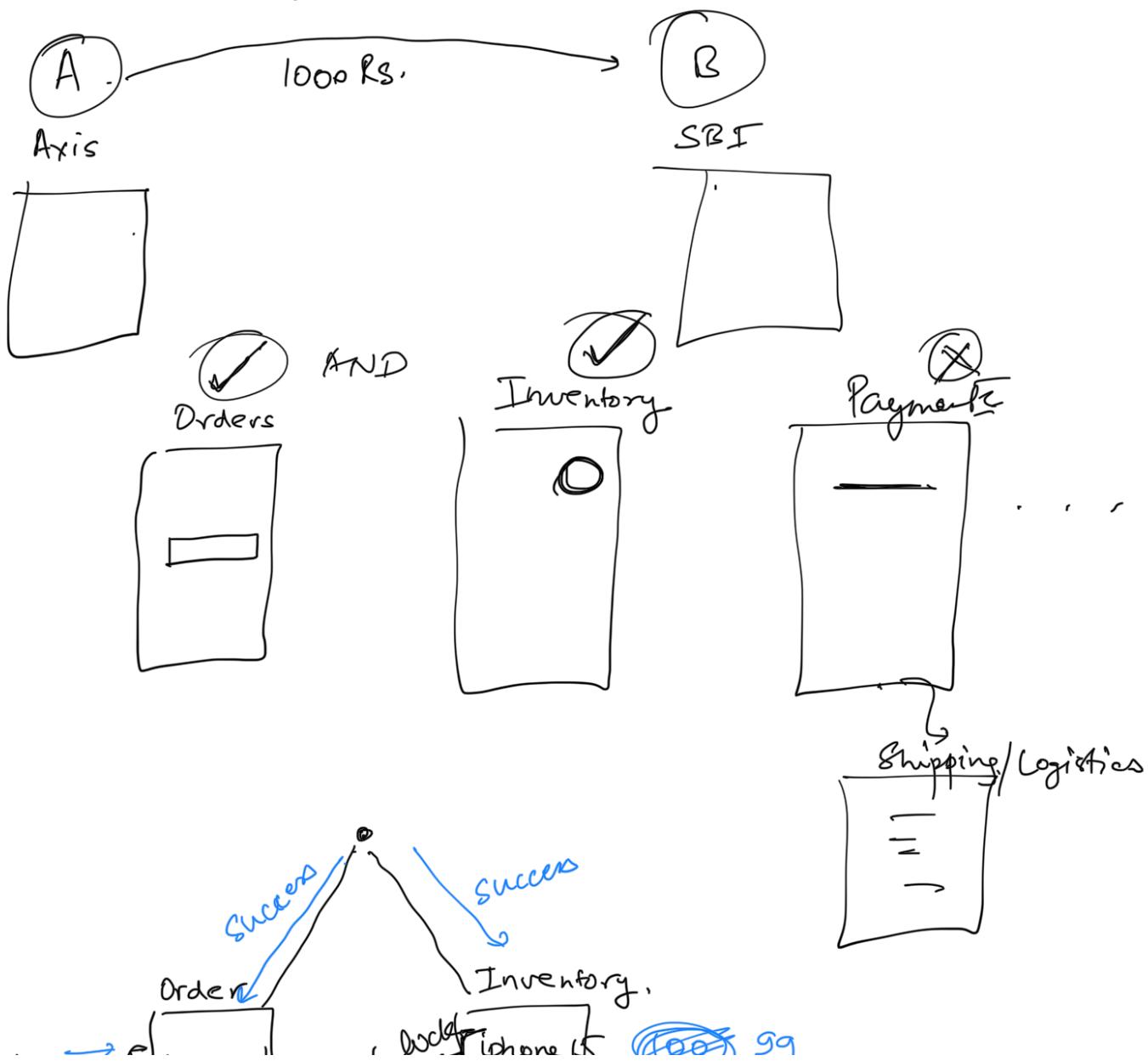
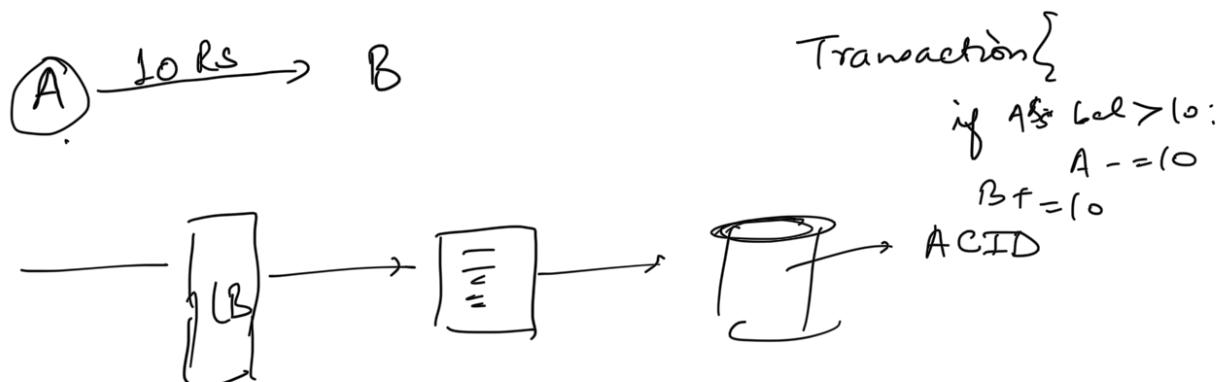
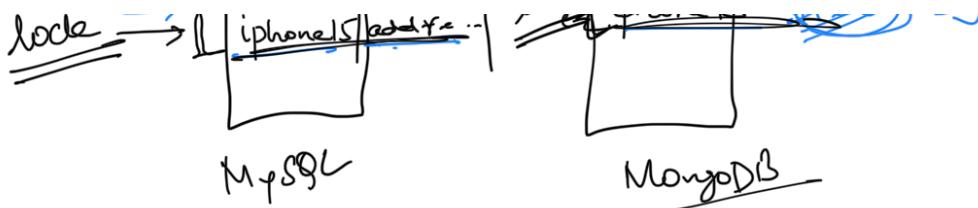


MICROSERVICES - 2

① Transactions

② Circuit Breaker





Step 1: Check with both
"Are they ready"

Step 2: Executions

Step 3: Release locks.

① long living transaction → lock

2 phase commit

SAGA Pattern

↳ Orchestration

↳ choreography.

Step 2:

Break it into steps

steps → 1 microservice



① $\text{Bal}_A > 1000$ → Query
 If $\text{Bal}_A > 1000$ → Yes → No

② $\text{Bal}_A = \text{Bal}_A - 1000$ if $\text{Bal}_A > 1000$
 If $\text{Bal}_A > 1000$ → Yes → No
 If $\text{Bal}_A > 1000$ → Error → Rollback

③ $\text{SBT} \rightarrow \text{Bal}_B = \text{Bal}_B + 1000$
 If $\text{Bal}_B = \text{Bal}_B + 1000$ → Yes → No
 If $\text{Bal}_B = \text{Bal}_B + 1000$ → Rollback

Step 1
Step 2

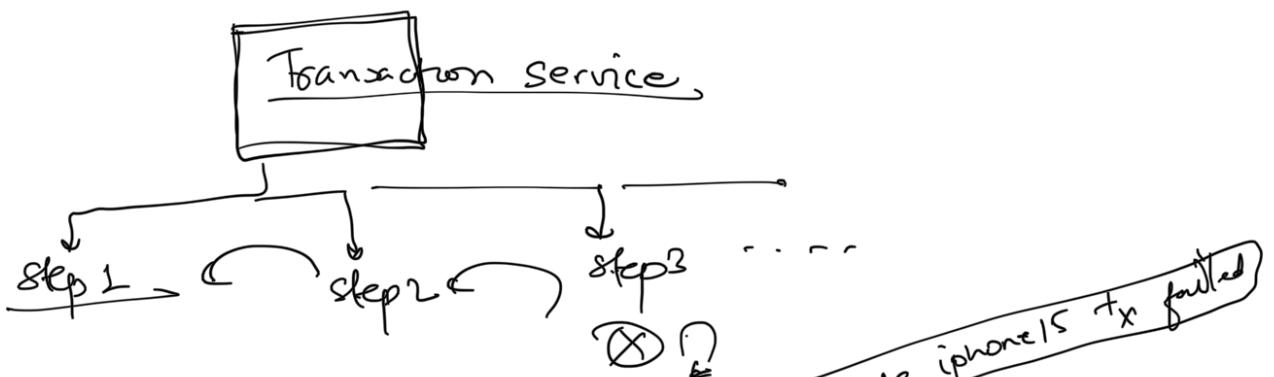
Step 2: Execute steps

On success,
make sure all
steps are executed

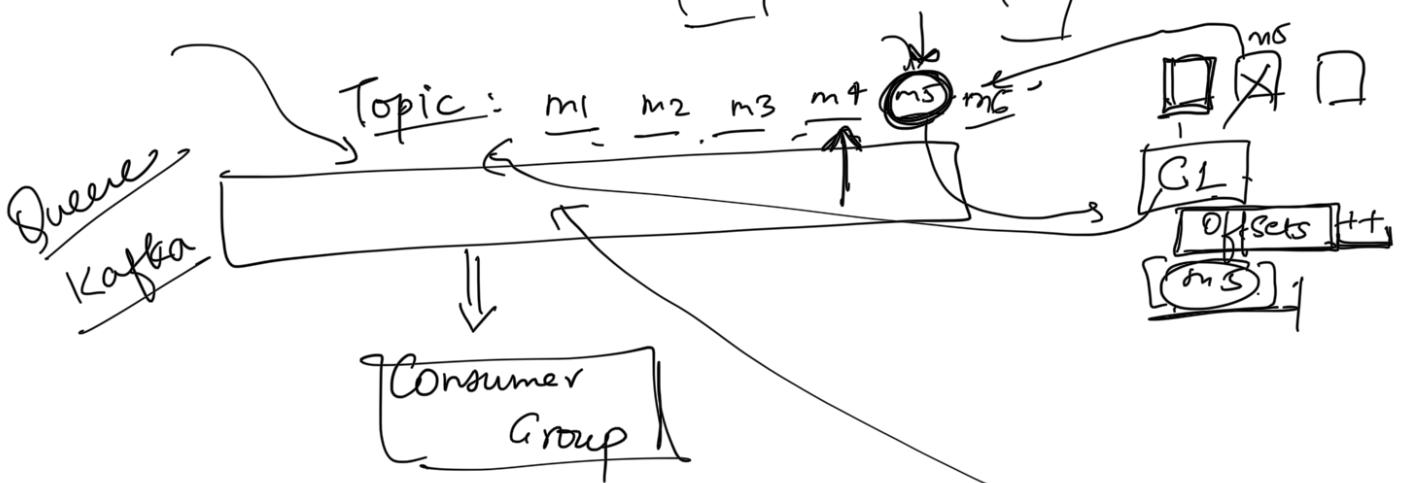
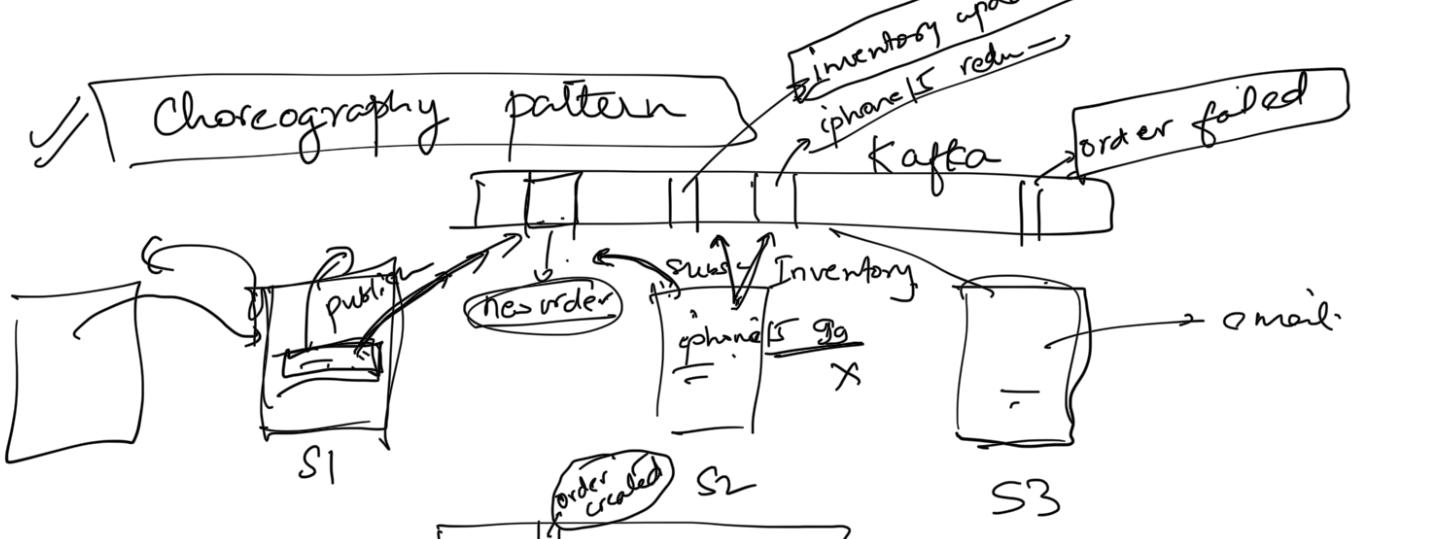
On failure, rollback
reliably on every
earlier step

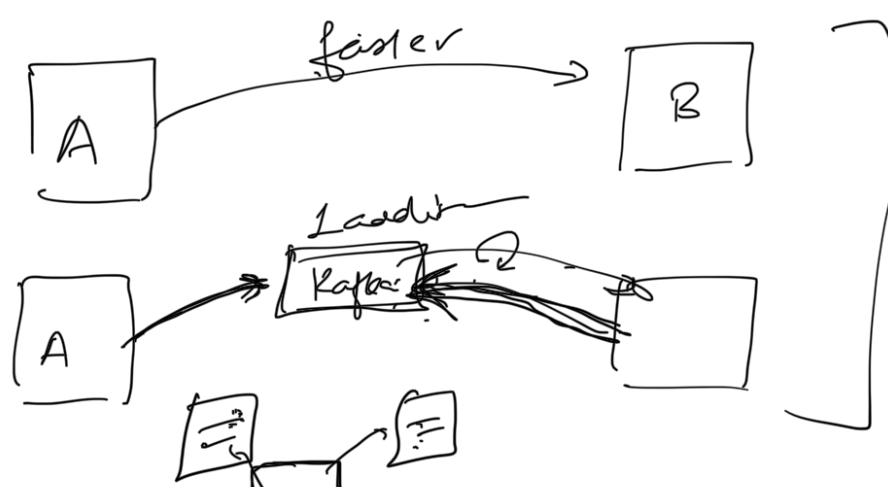
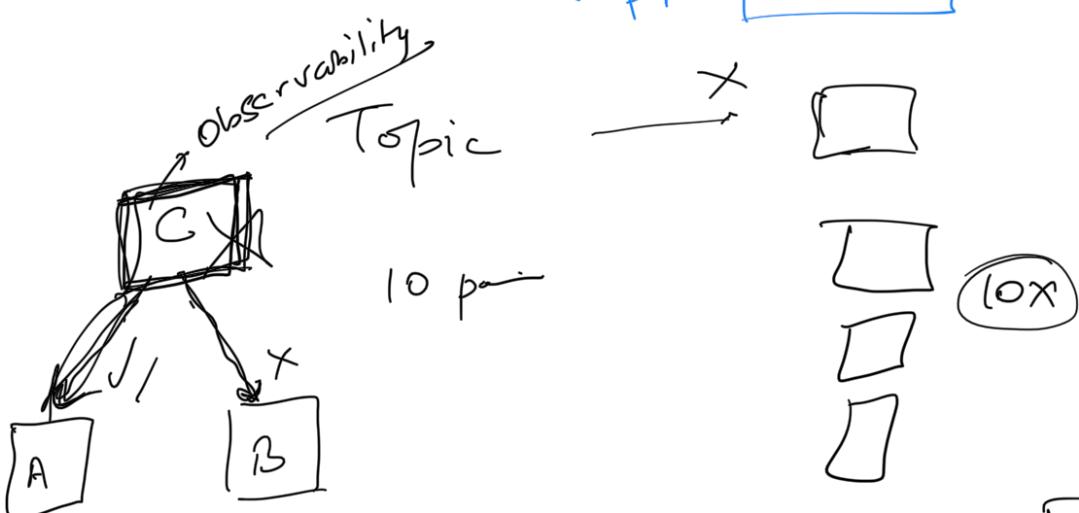
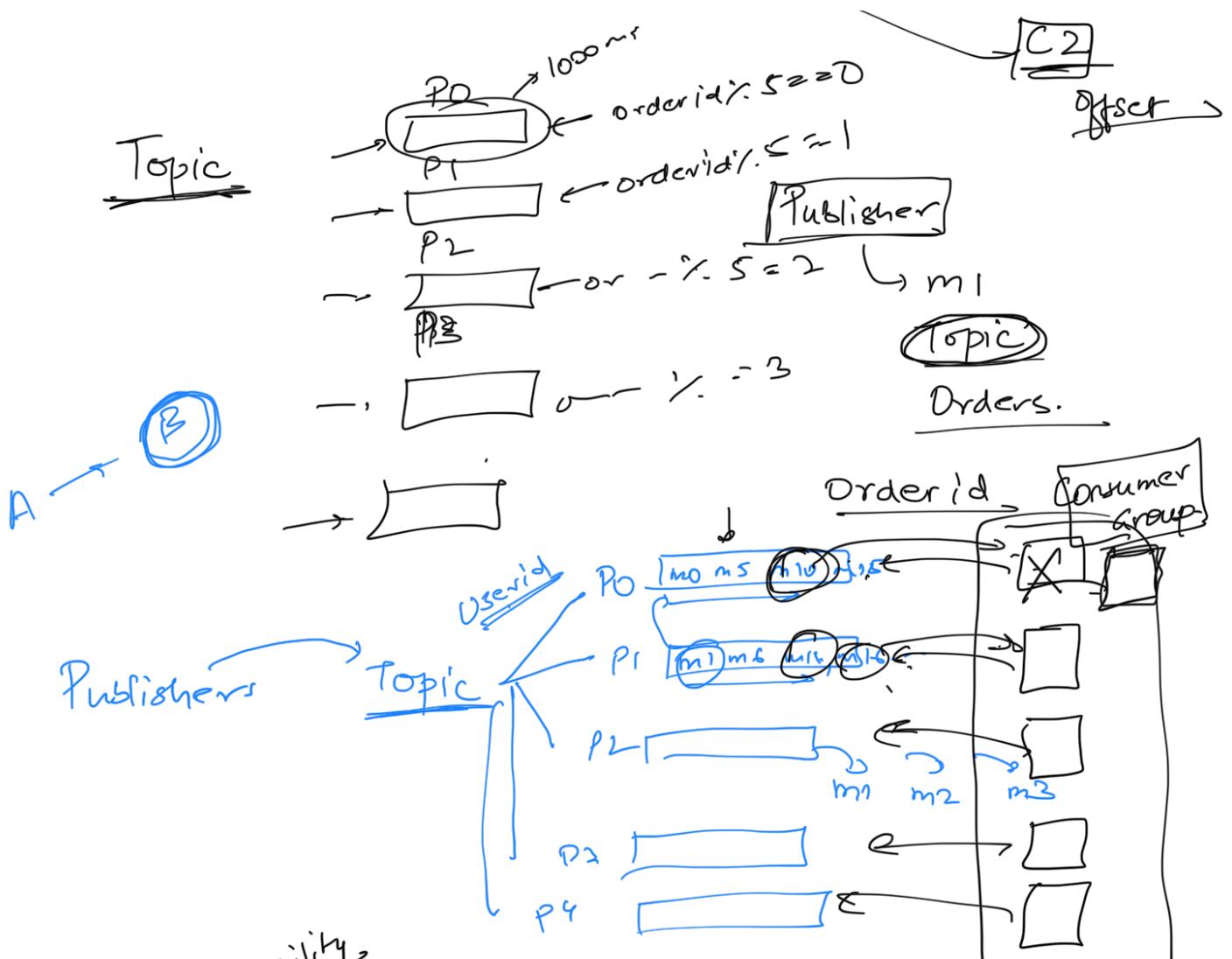
Orchestration pattern

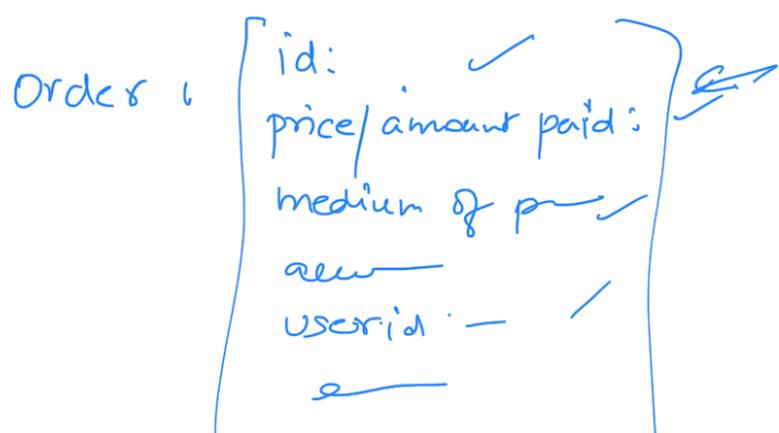
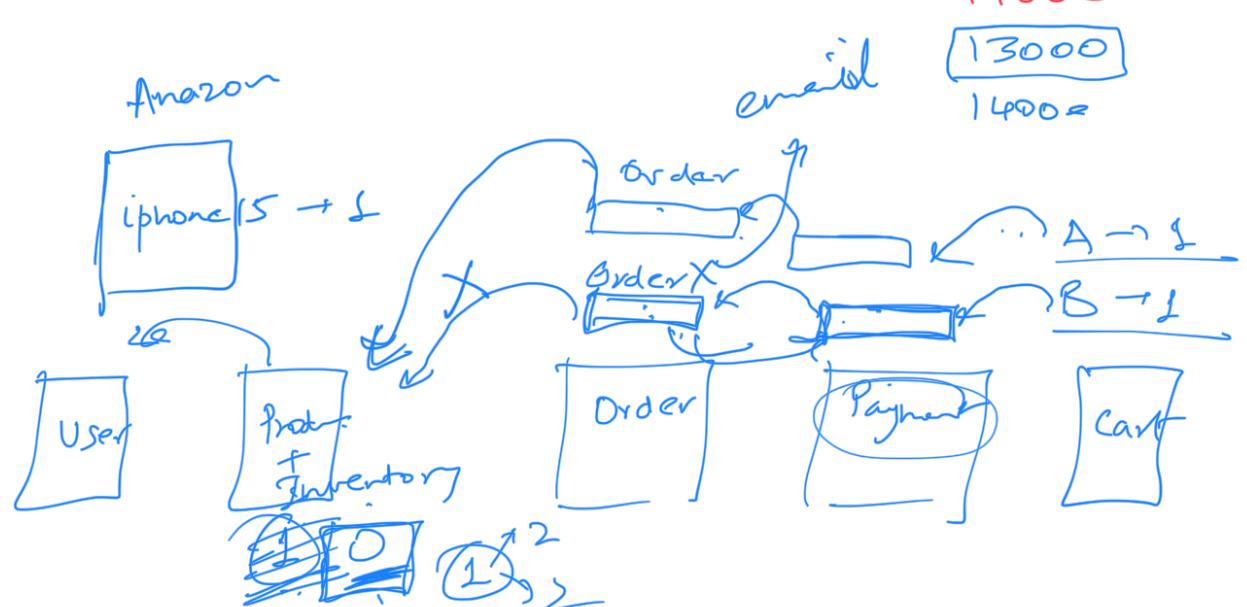
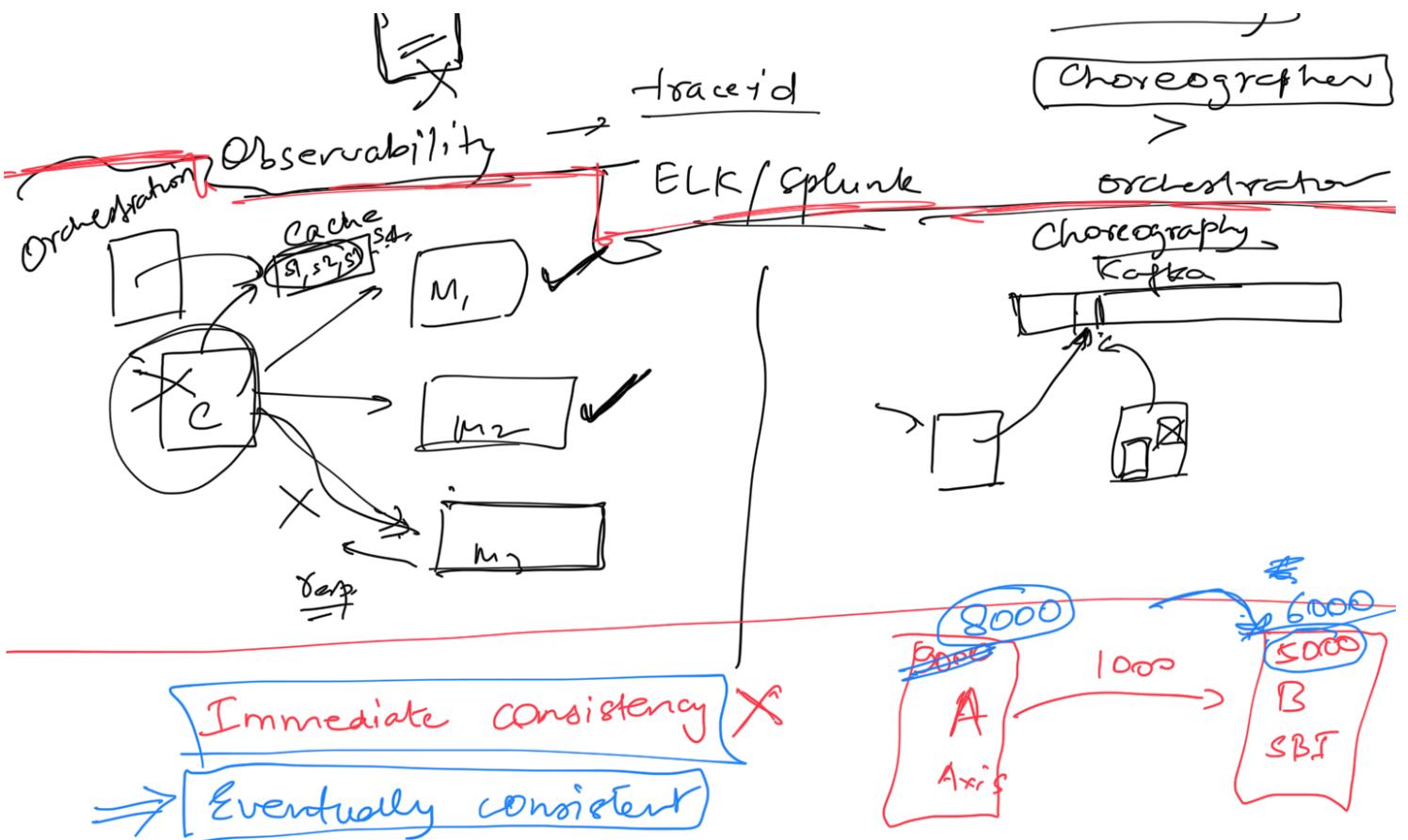
or retry
till success

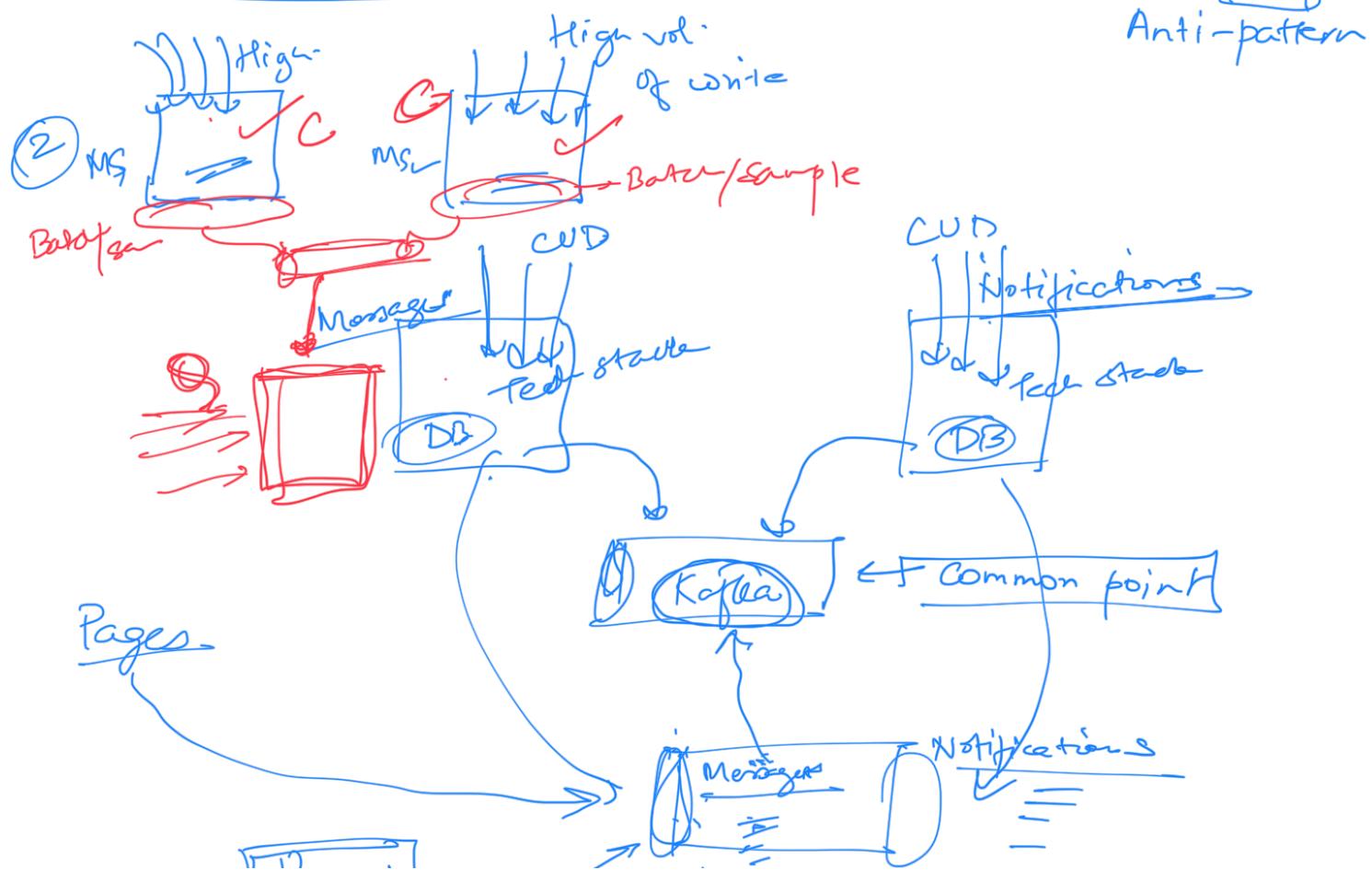
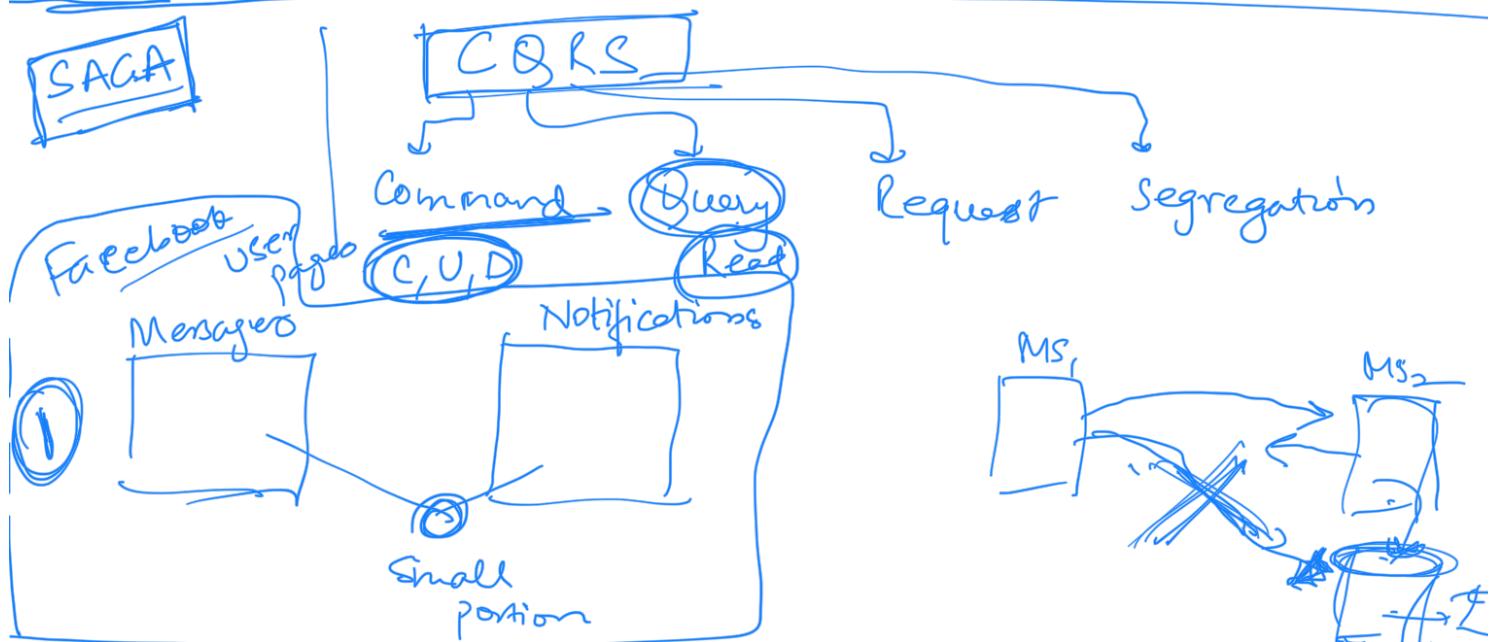
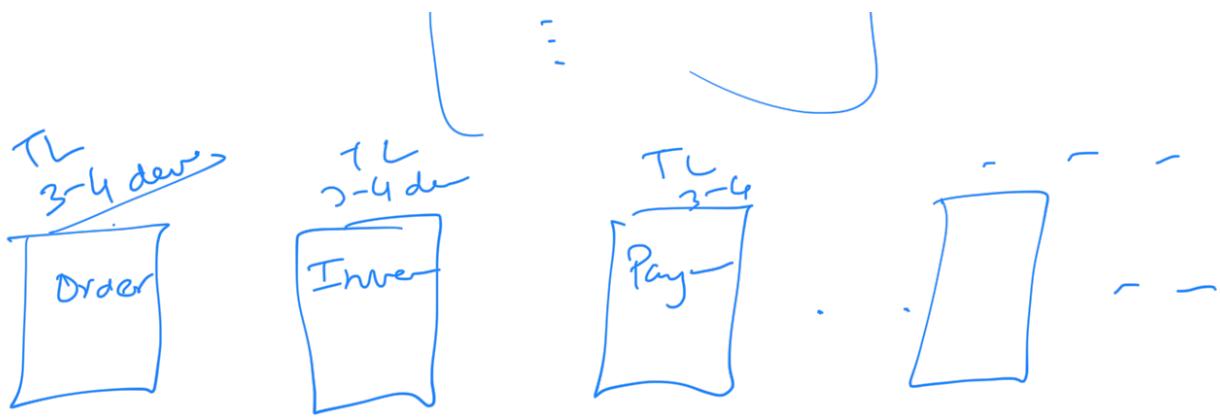


Choreography pattern





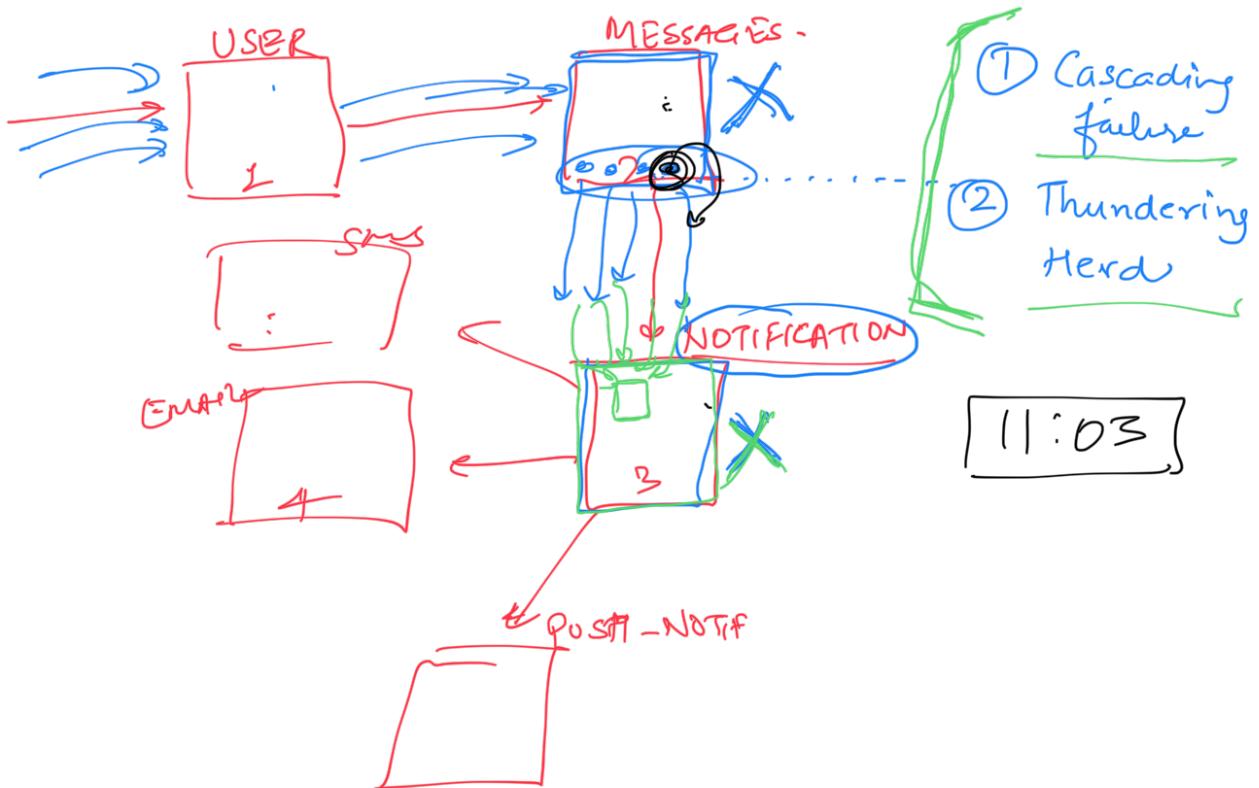




Read

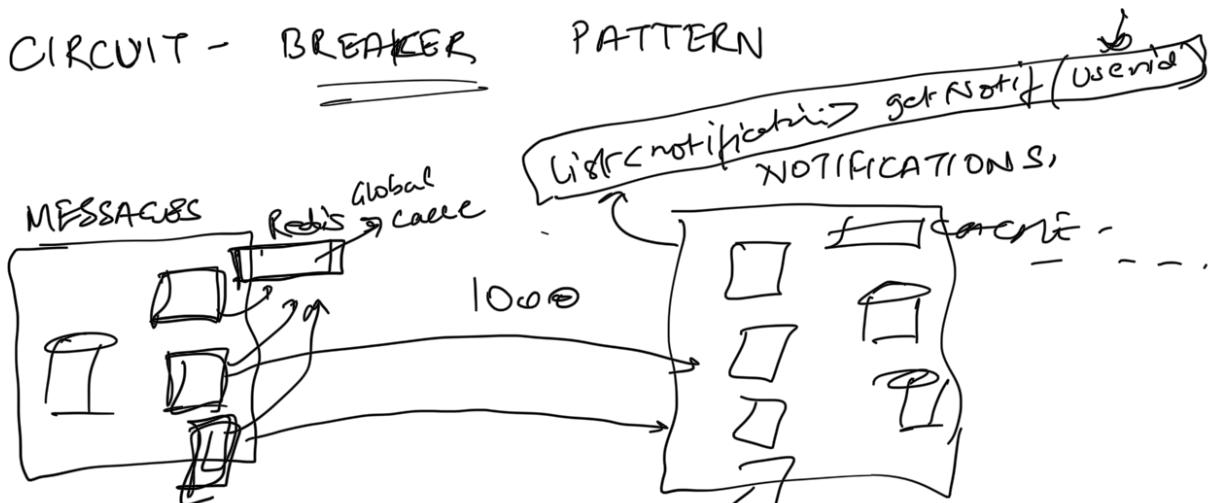
→ writes X

Problem



CIRCUIT - BREAKER

PATTERN



Step 1:

Failure category

Non-Transient failure

of failures > threshold

Timeout
Connections reset

Regular failures,

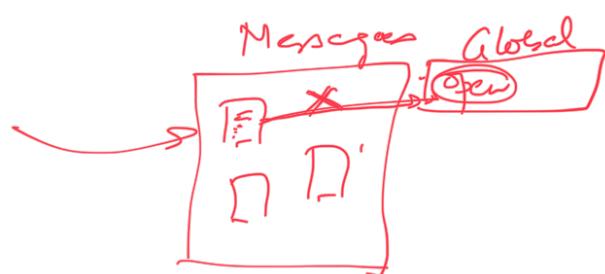
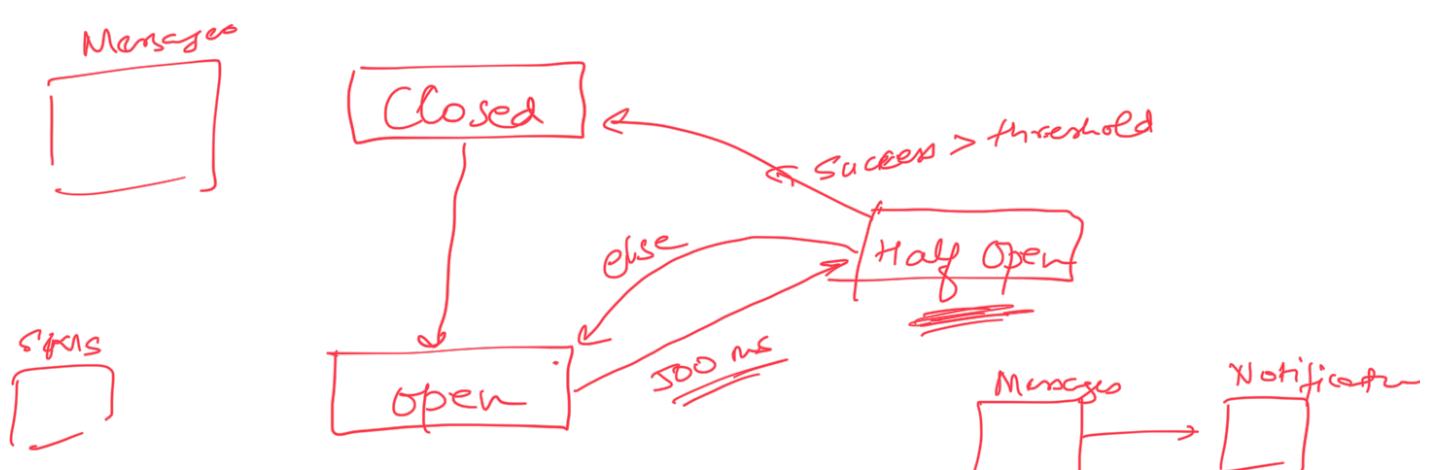
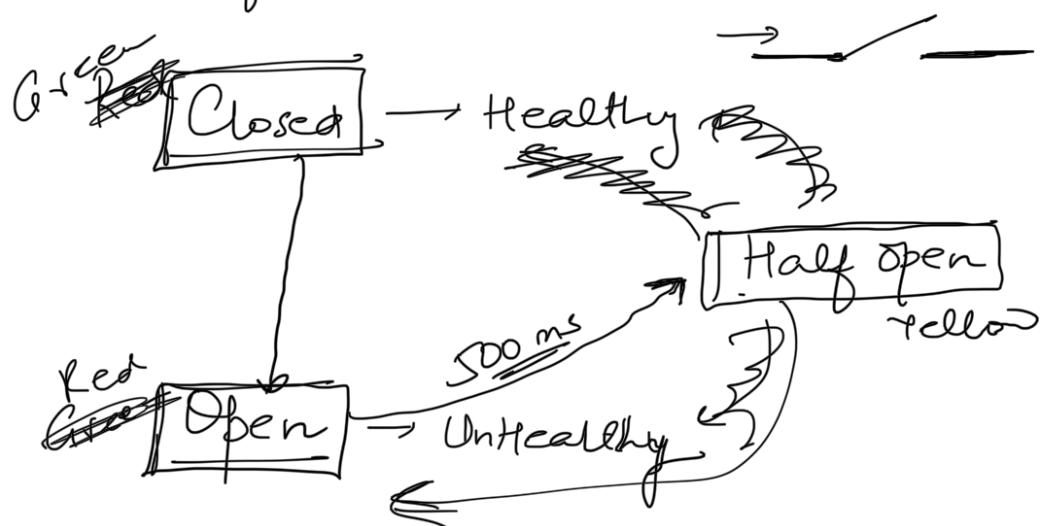
userid not found or invalid



Step 2

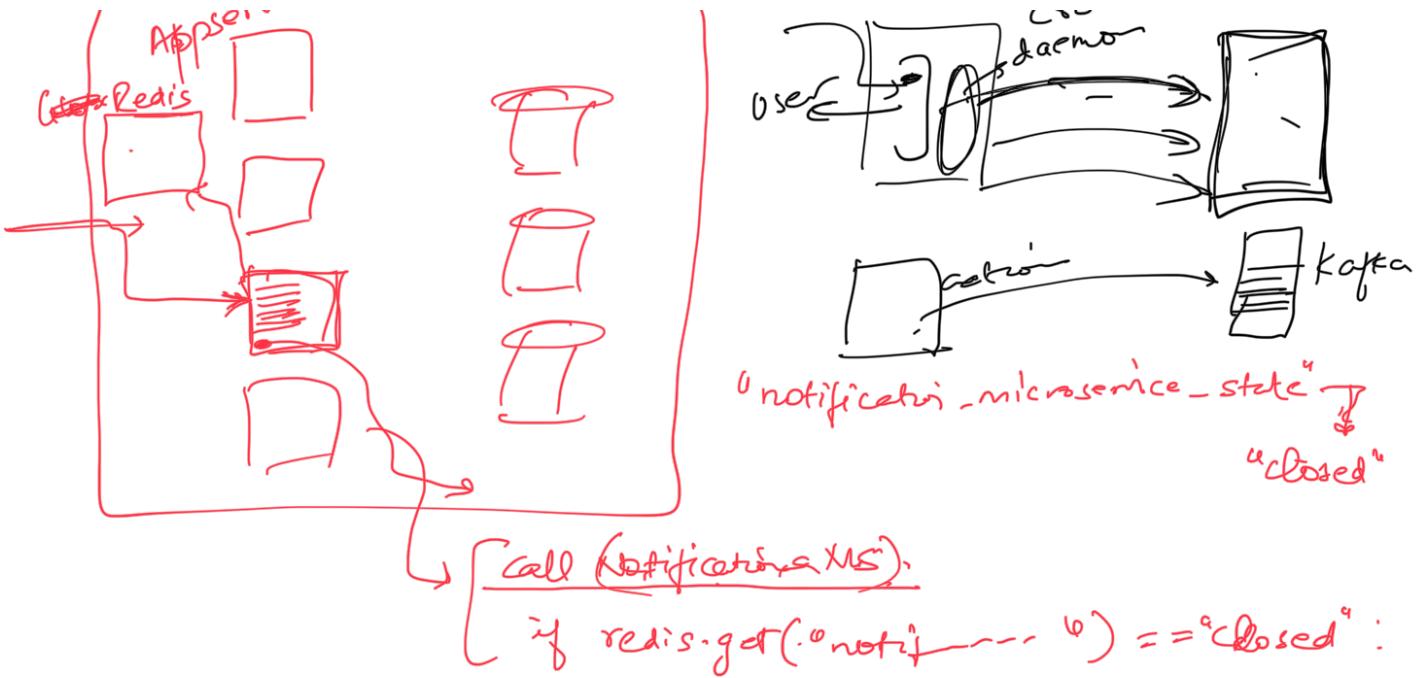
i) Detect if MS is down

ii) Detect if MS has recovered.



Failure → Rollback
Failure → Queue it for retry later

Message



~~Appse rec of ms.~~
~~Message~~
Cur-state = redis.get("...")

```

if cur-state == "open":
    // do nothing / fail

elseif cur-state == "closed":
    try {
        resp = notifications.getcluster(...).get...
    }
    catch (exception e) {
        cont = redis.get("cur-f--") + 1
        if cont >= threshold:
            redis.set("...", "open")
            redis.update("count", cont)
    }
}

```

else

// cur-state = "half-open"

0
1
2
3
4

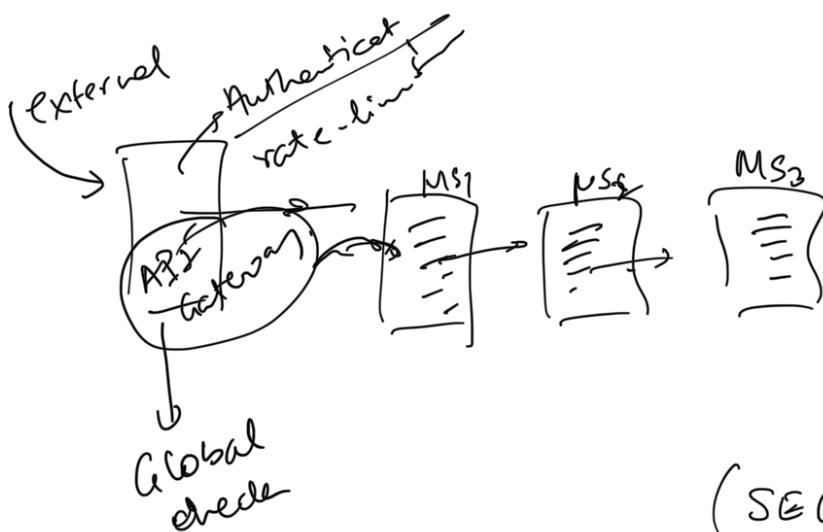
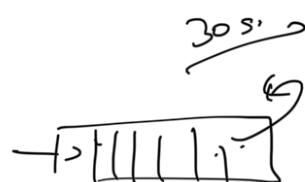
(if $\text{rand()} \% 100 = -\text{u}.$)
 to $\text{resp} = \dots$
 } except \dots
 else:
 // fail or ignore

}
 ;
 IS

cron 0 0 0 0 500 exec-sh
 ↓
 if $\text{redis.get}(\dots)$
 == "open"
 $\text{redis.set}(\dots,$
 $"half-open")$

Closed → Healthy.

open → unhealthy
 half-open → not safe



SQL Injections

(SELECT * FROM users

WHERE user_id = '12')

db.query("SELECT -- user_id = (?)", user_id);

