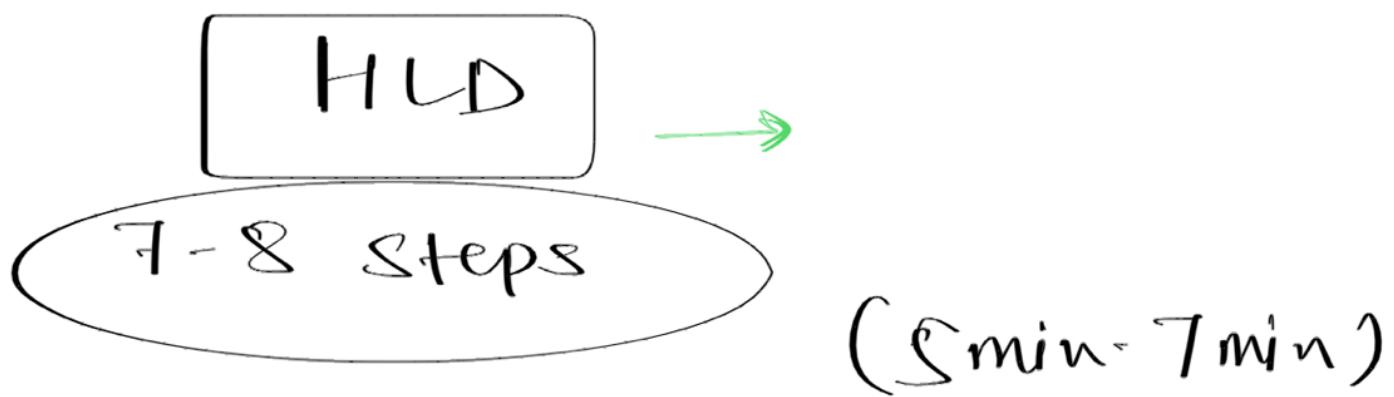


8 Aug 2025: BossCoder



①

Functional Requirement

→ Feature that your system will have.

Twitter (3-4 funcn)

- a) User should be able to post
- b) User should be able to follow
- c) User should be able to see tweet of whom they follow.

Assumption

- Profile creation
- Authentication

②

Non-functional Requirements

- Quality of System

(A)  → low latency (less lag)
of rendering feed under 200 ms

(B) CAP  → System should be
highly Available.

(C) Prioritizing Availability over
consistency)

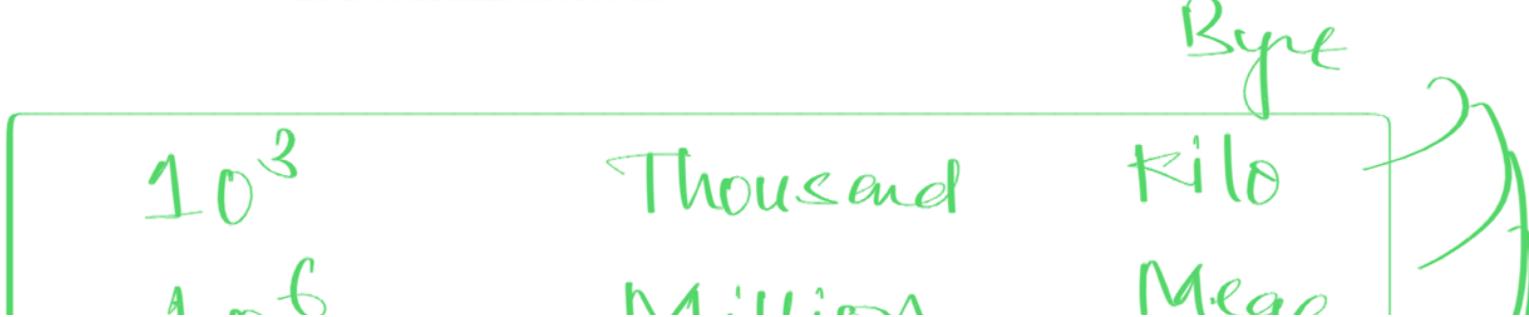
(C) System should be Scalable
to 100 M + DAU
(Daily Active User)

③ Capacity Estimation

Query per second

DAU

$$\text{DAU} : \boxed{300 \text{ M}} = 300 \times 10^6$$


 10^3 Thousand
 10^6 million
kilo
Mea

10^3	Thousand	Ja
10^9	Billion	Giga
10^{12}	Trillion	Tera
10^{15}	Quadrillion	Peta

$$DAO = 3 \times 10^8$$

(10 search) a day

$$\begin{aligned} \text{Total Search in a day} &= 3 \times 10^8 \times 10 \\ &= 3 \times 10^9 \end{aligned}$$

$$1 \text{ Day} = 24 \times 60 \times 60 = 86400 \text{ second}$$

$\approx 10^5$ second

$$\approx 1 \text{ day} = 10^5 \text{ second}$$

$$\begin{aligned} QPS &= \frac{3 \times 10^9}{10^5} = 3 \times 10^4 \end{aligned}$$

$$QPS = 30,000$$

→ Storage estimation
Amazon → 10 M product
↓
10 MB

$$\begin{aligned} &= 10 \text{ m} \times 10 \text{ MB} \\ &= 10 \times 10^6 \times 10 \times 10^6 \\ &\approx 100 \times 10^{12} \\ &= 100 \text{ TB} \end{aligned}$$

Store All product

④

API Design & Core Entities

⑤

Database Design.

→ which DB

→ How? (schema)

⑥

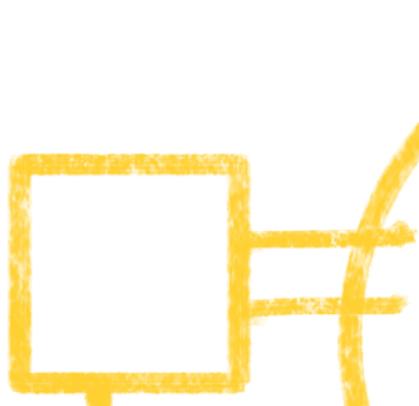
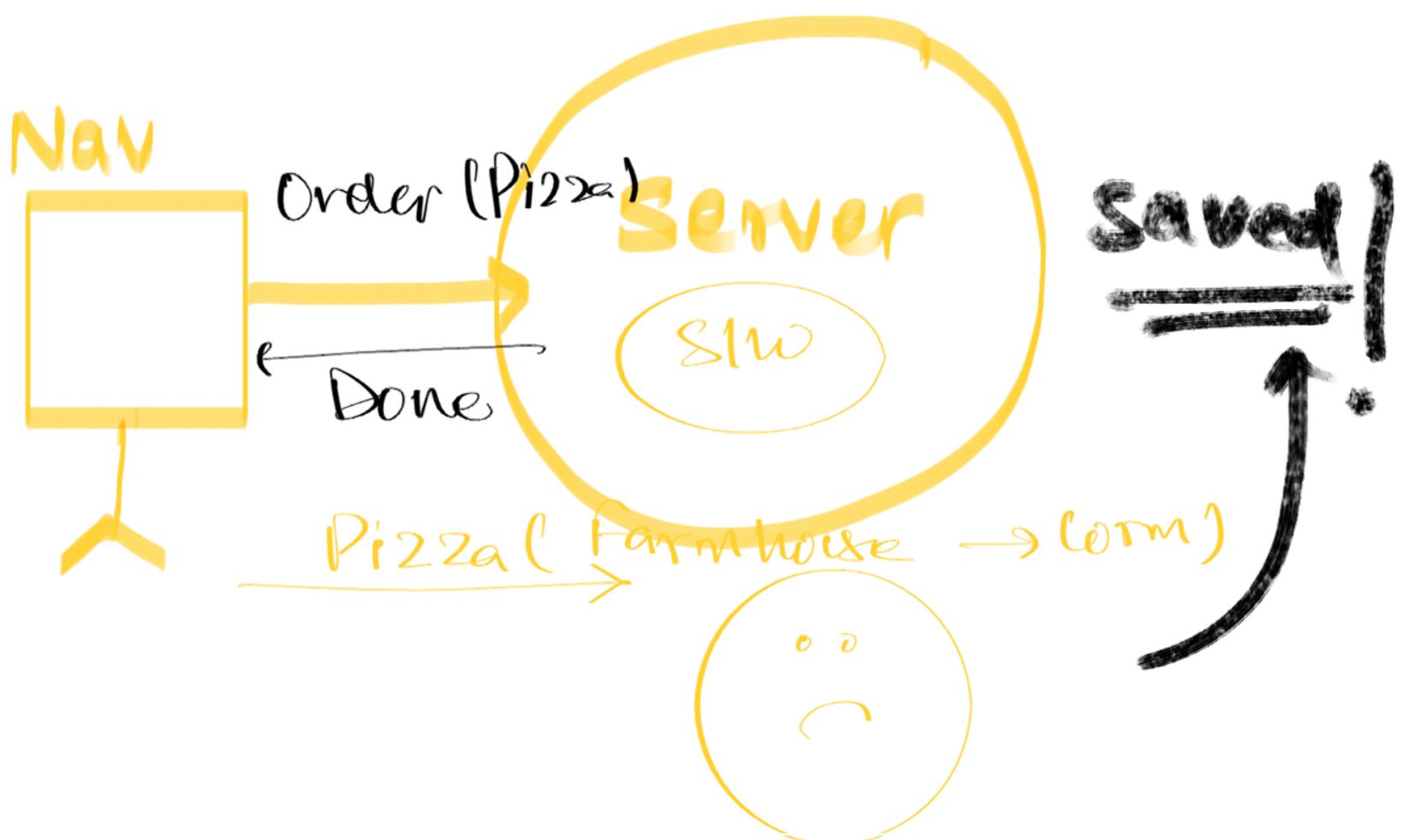
HLD DIAGRAM
(Architecture)

⑦

Deep Dive, Explain LB
Trade off

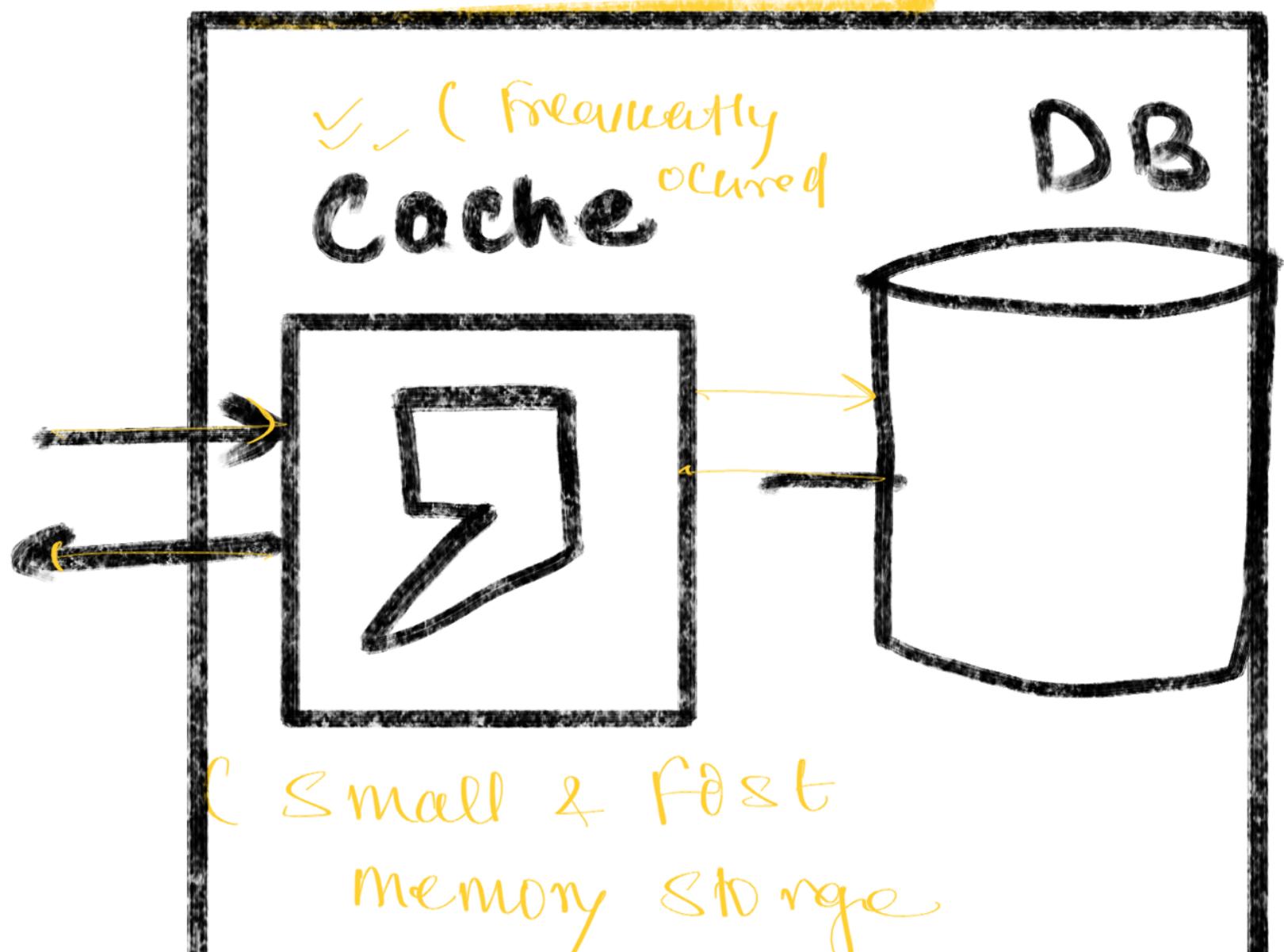
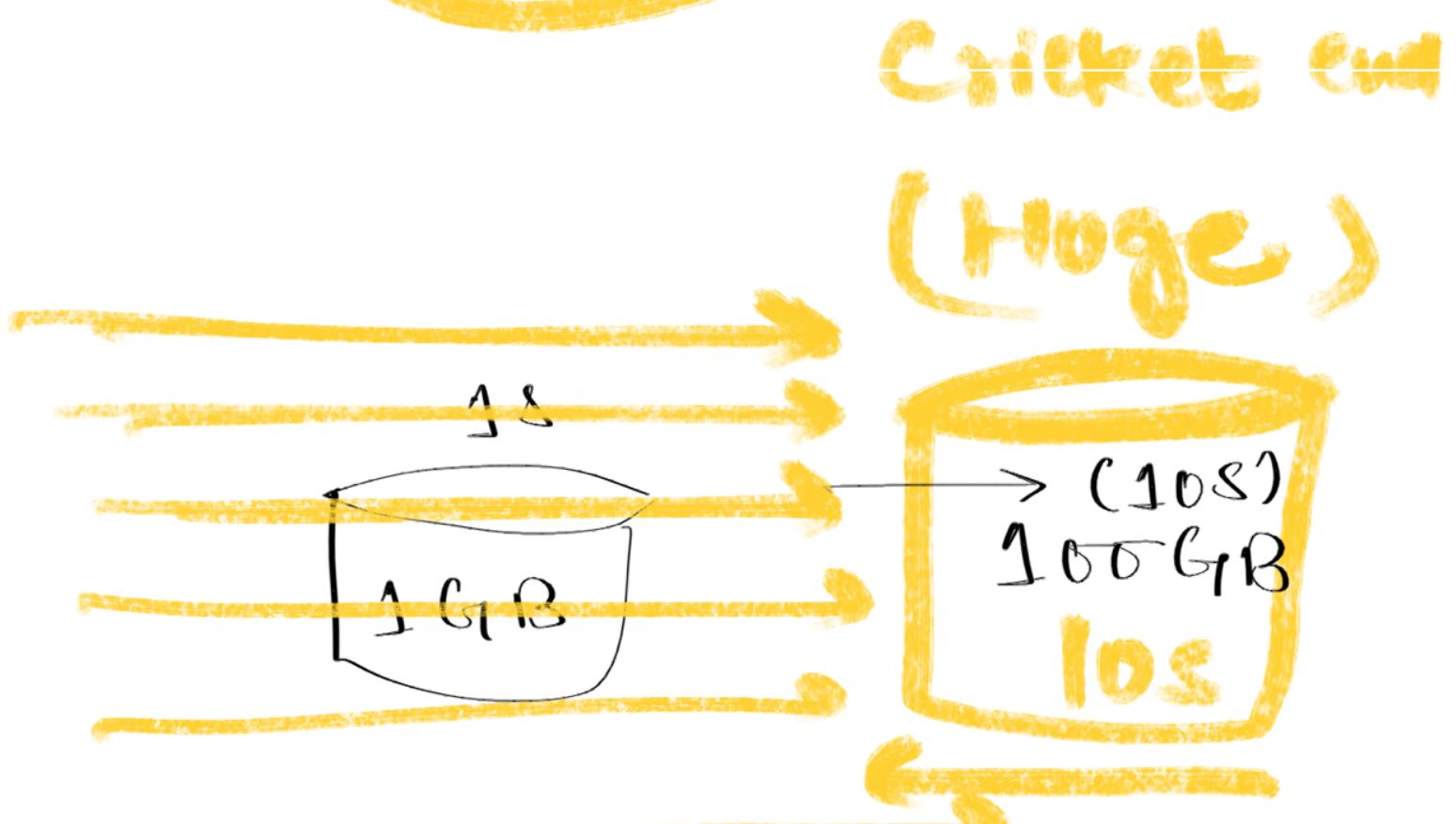
⑧

Scale for million
of user.

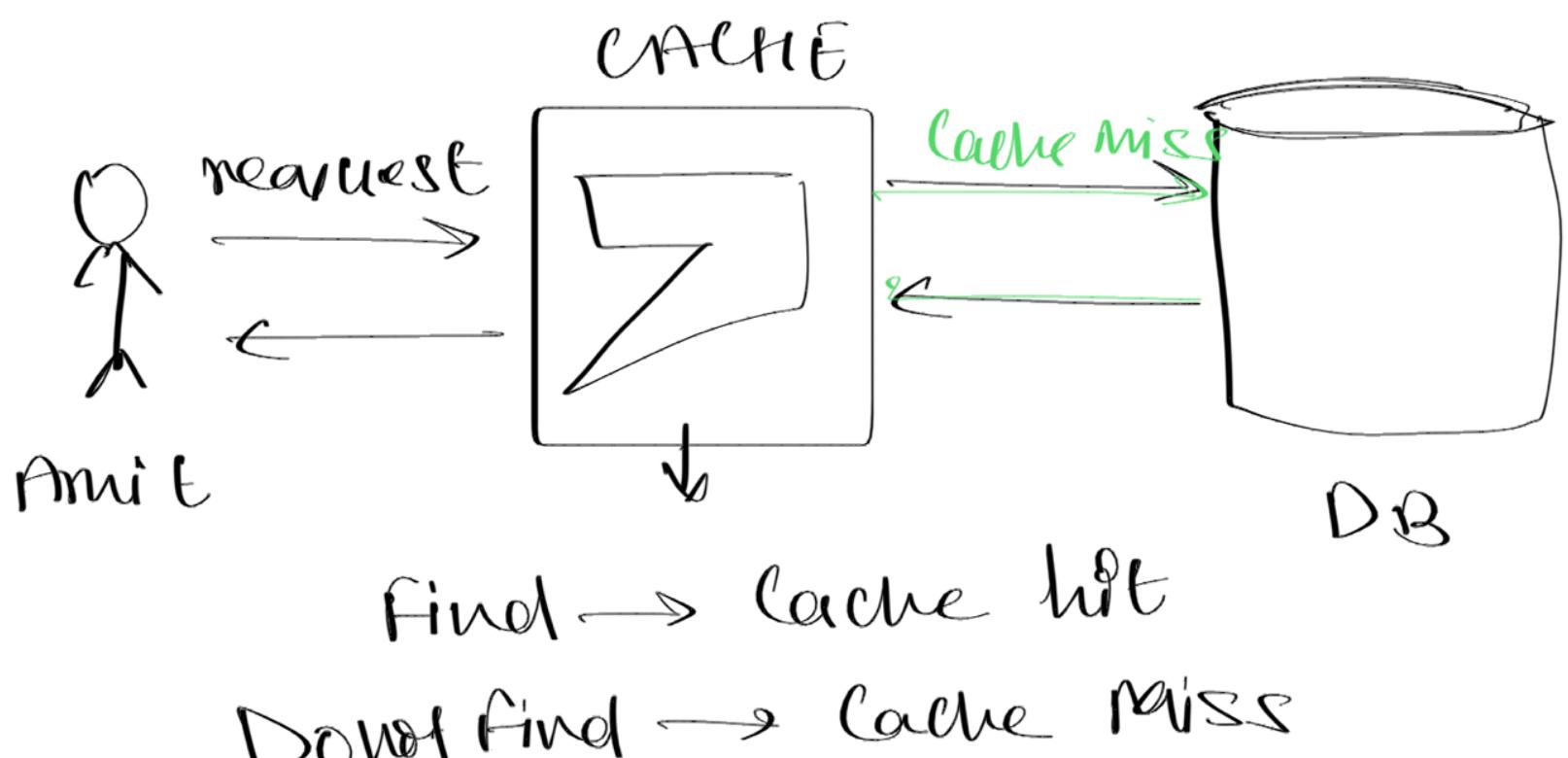


Server





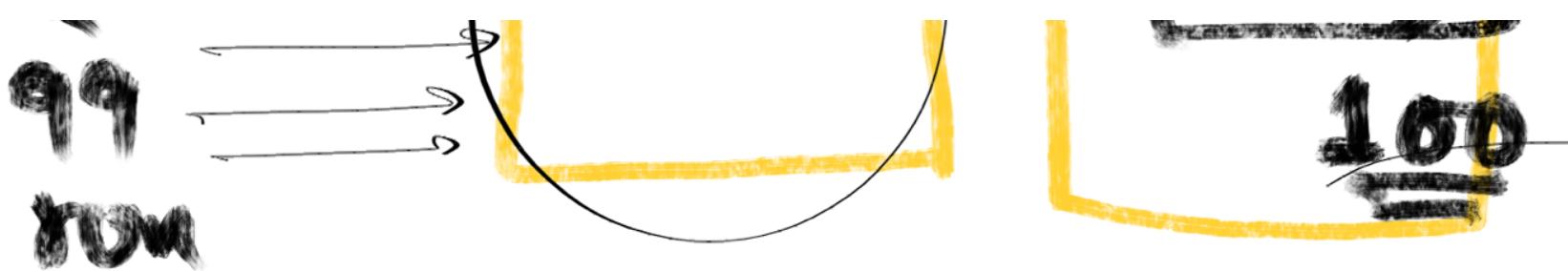
* Caching is process of storing data in high speed & small sized storage



Cache Eviction

Cache is full
& we remove
some data)

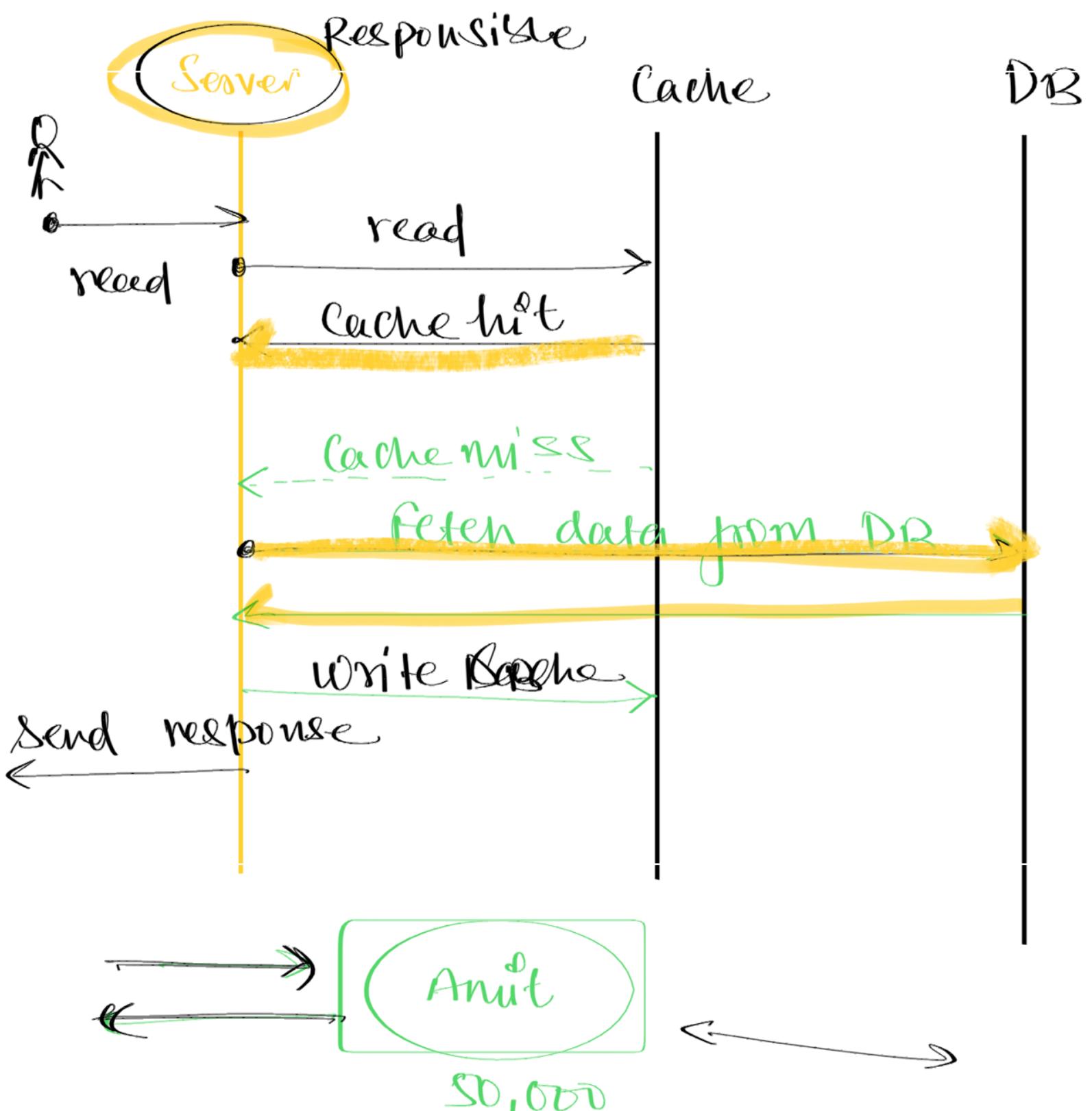




Cache Invalidation

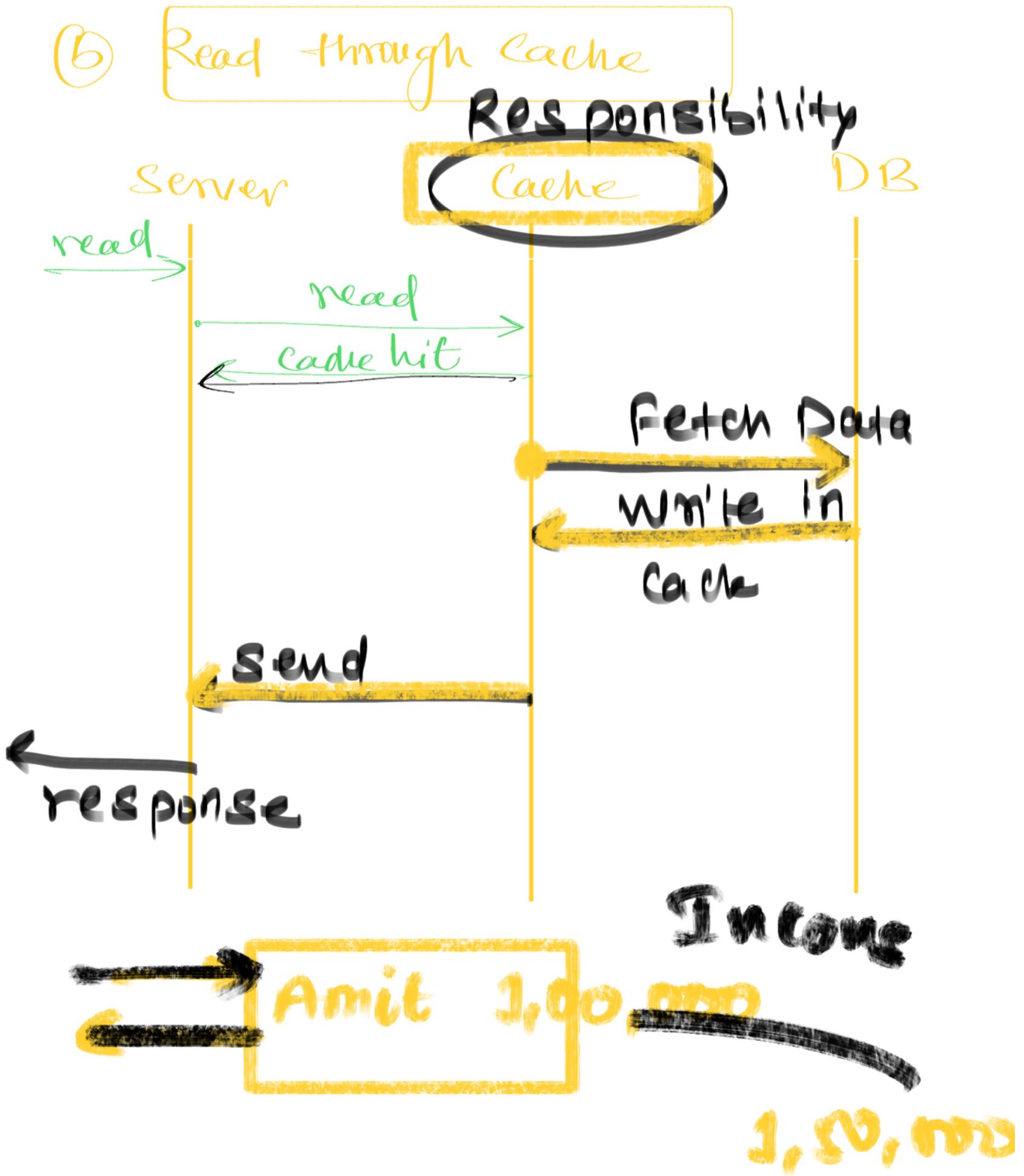
→ DB & Cache are inconsistent.

① Cache Aside



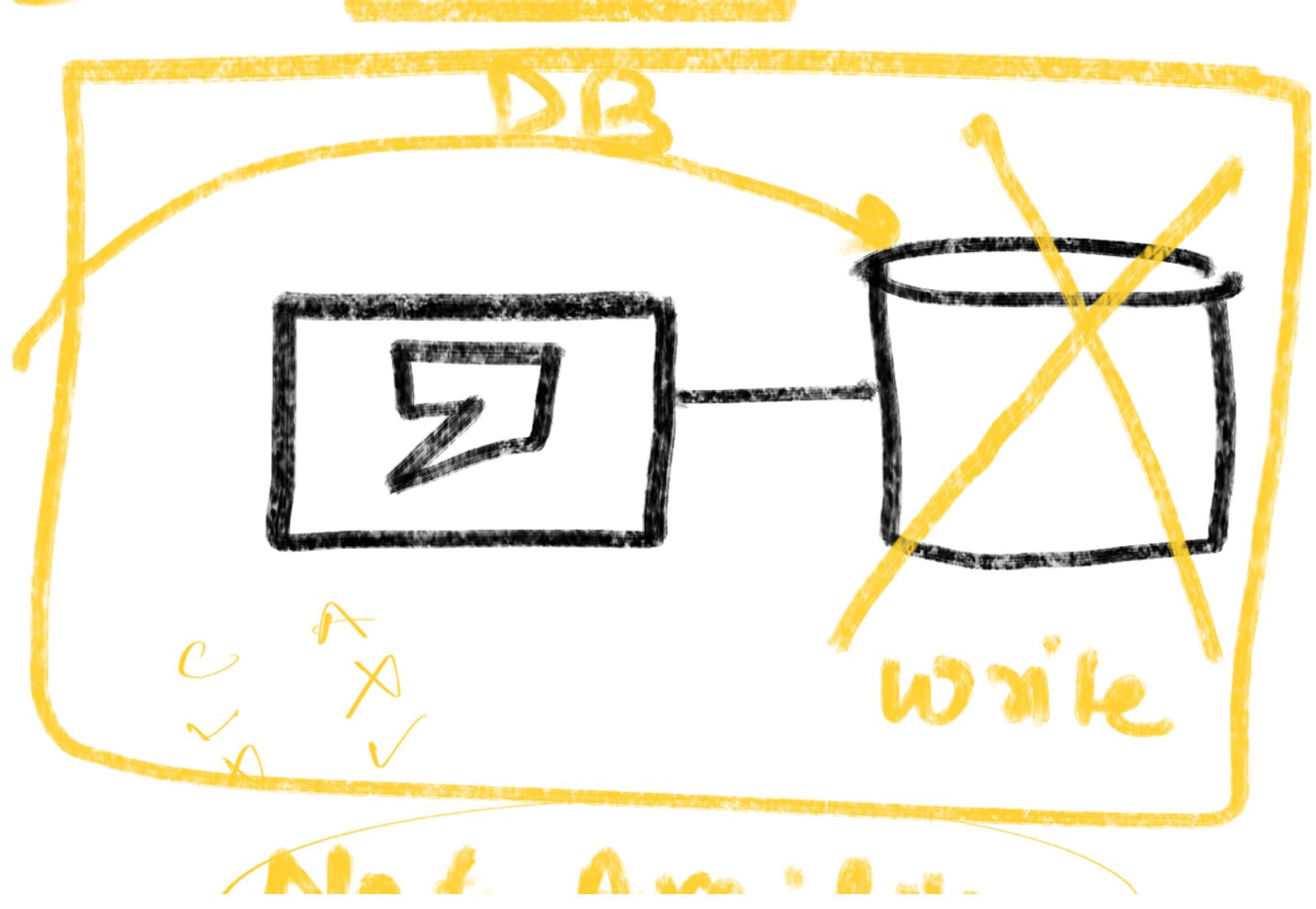
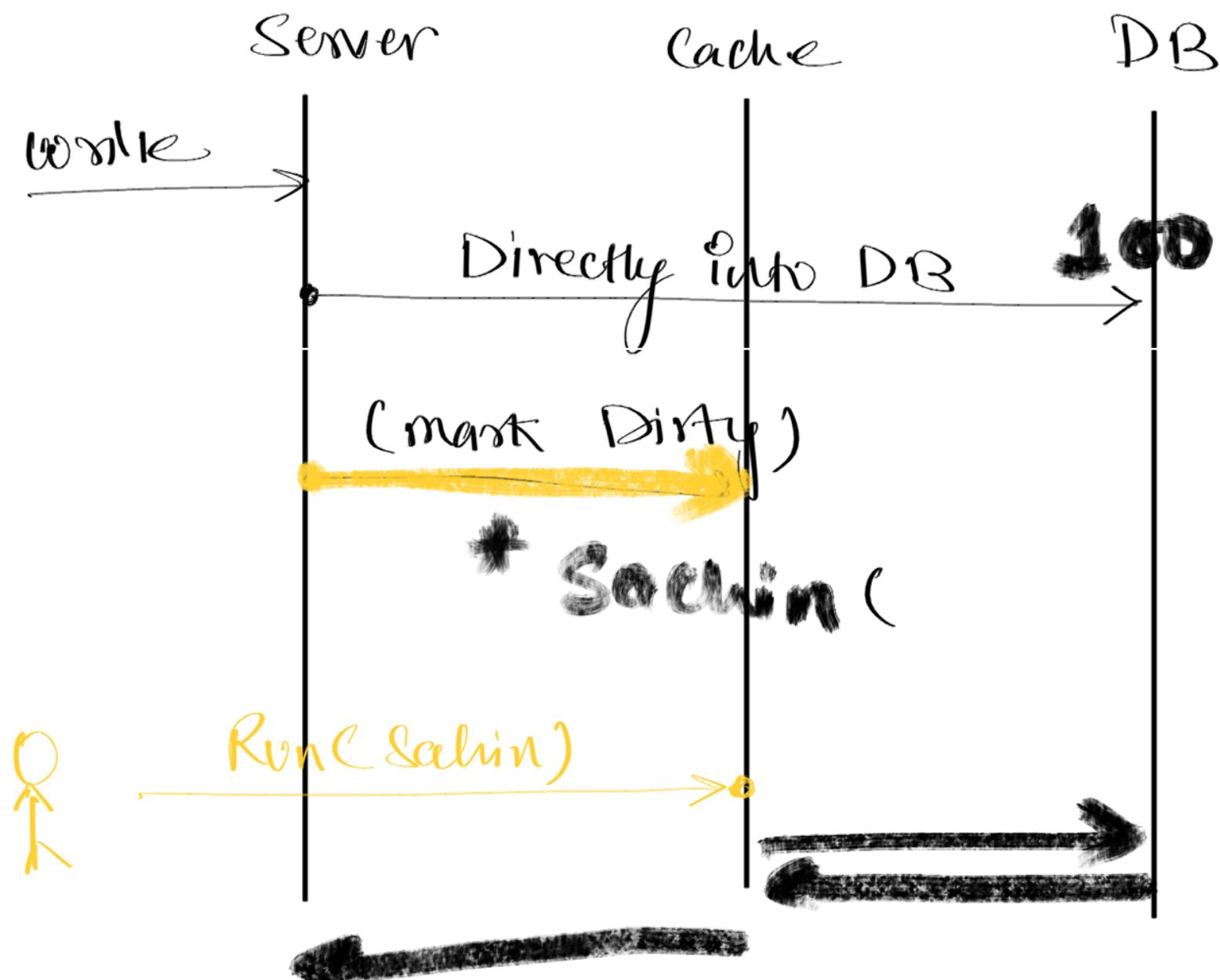
\$5,000

→ Inconsistency b/w DB & Cache.

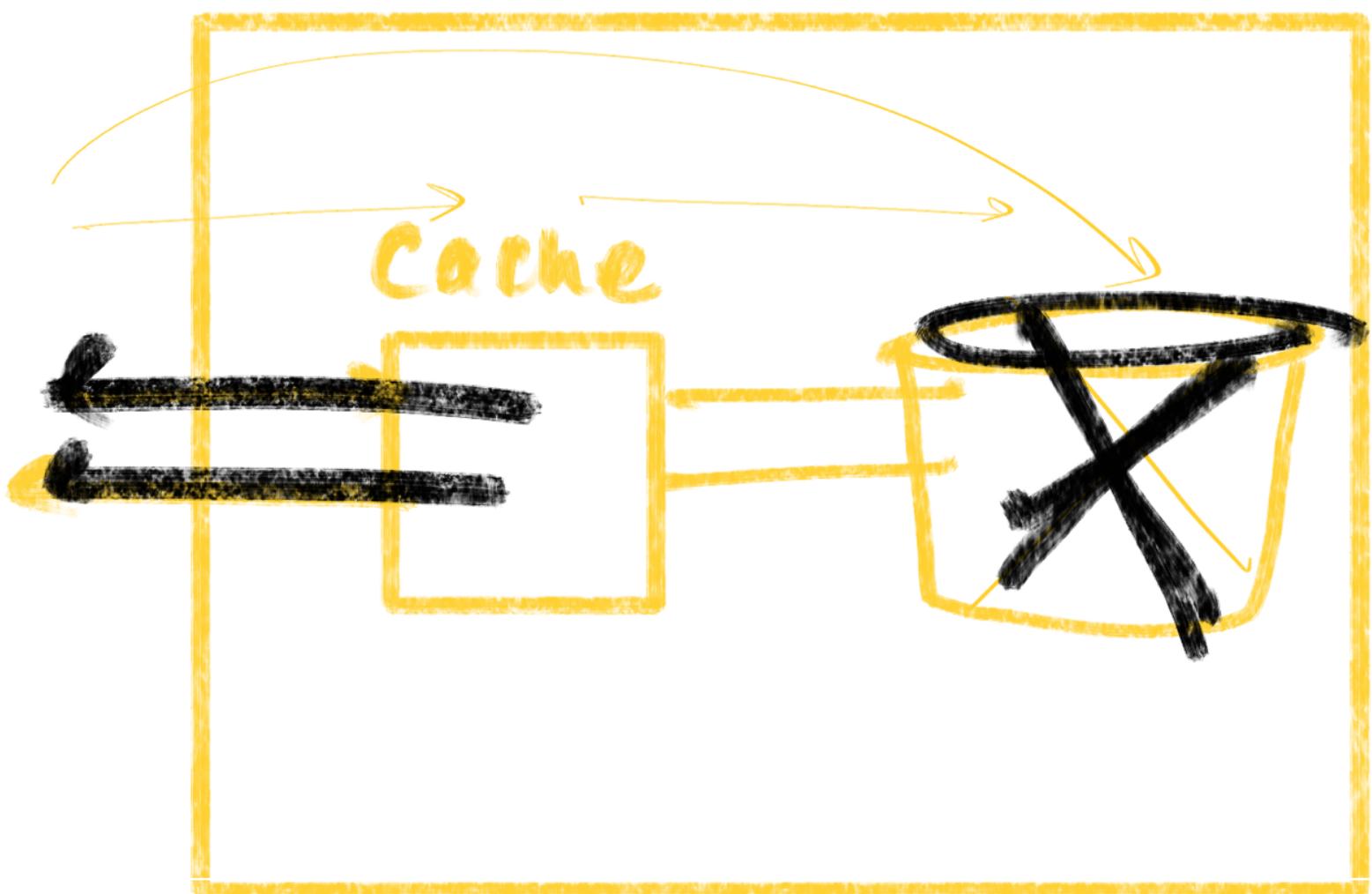


(3)

write Around Cache



(YOU ARE YOU)



Consistency

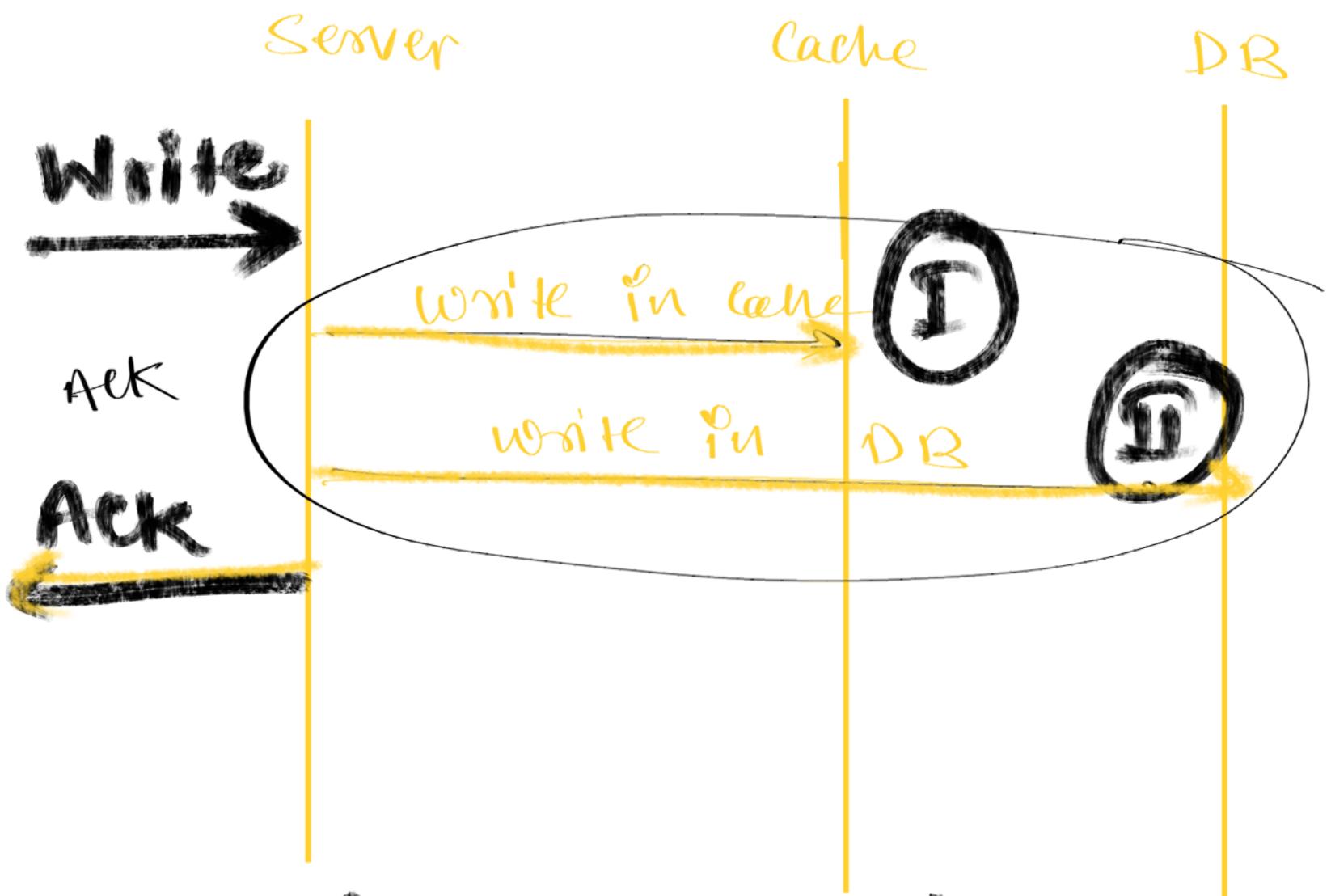


Avail



④

Write-through Cache

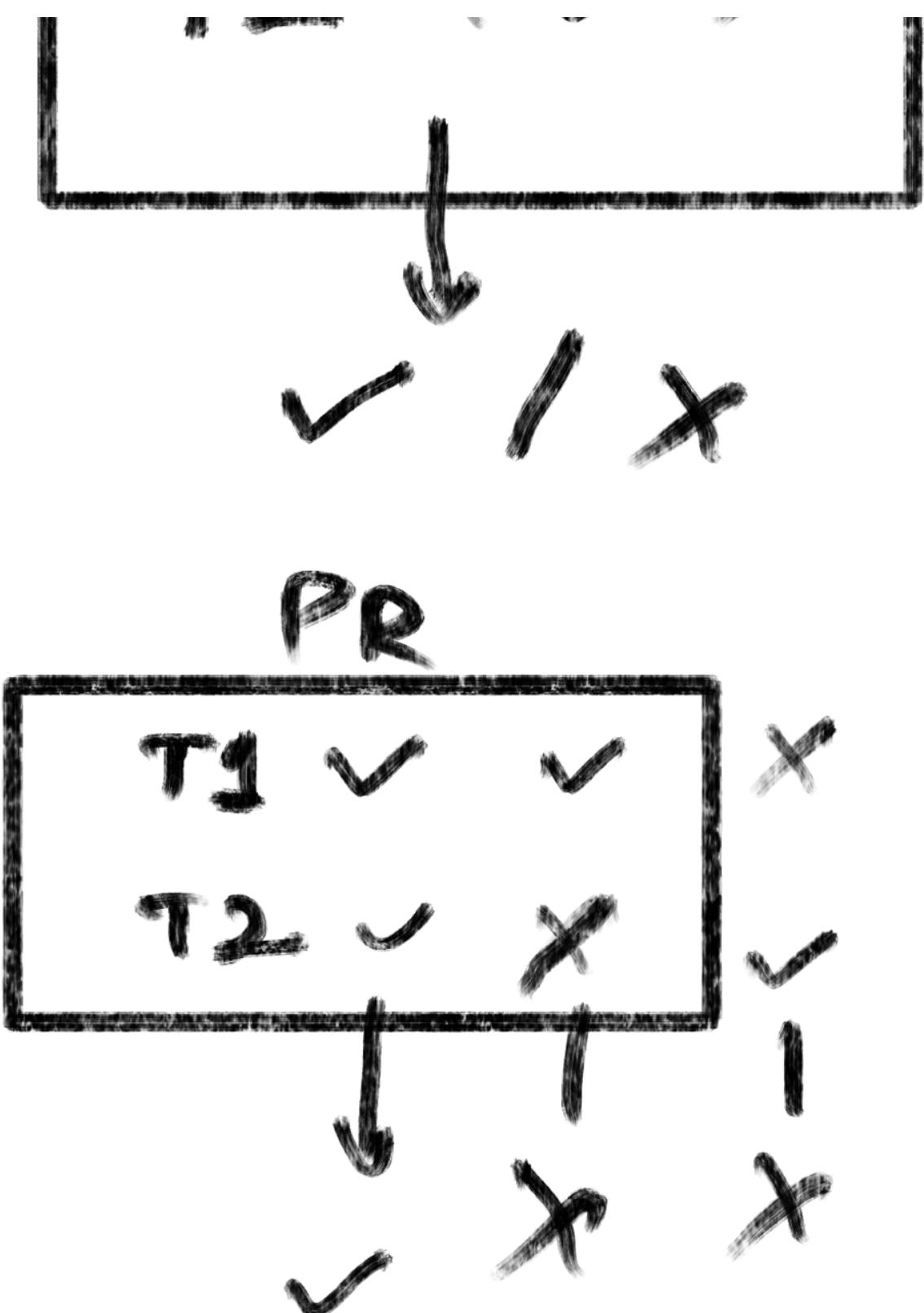


2 phase Commit

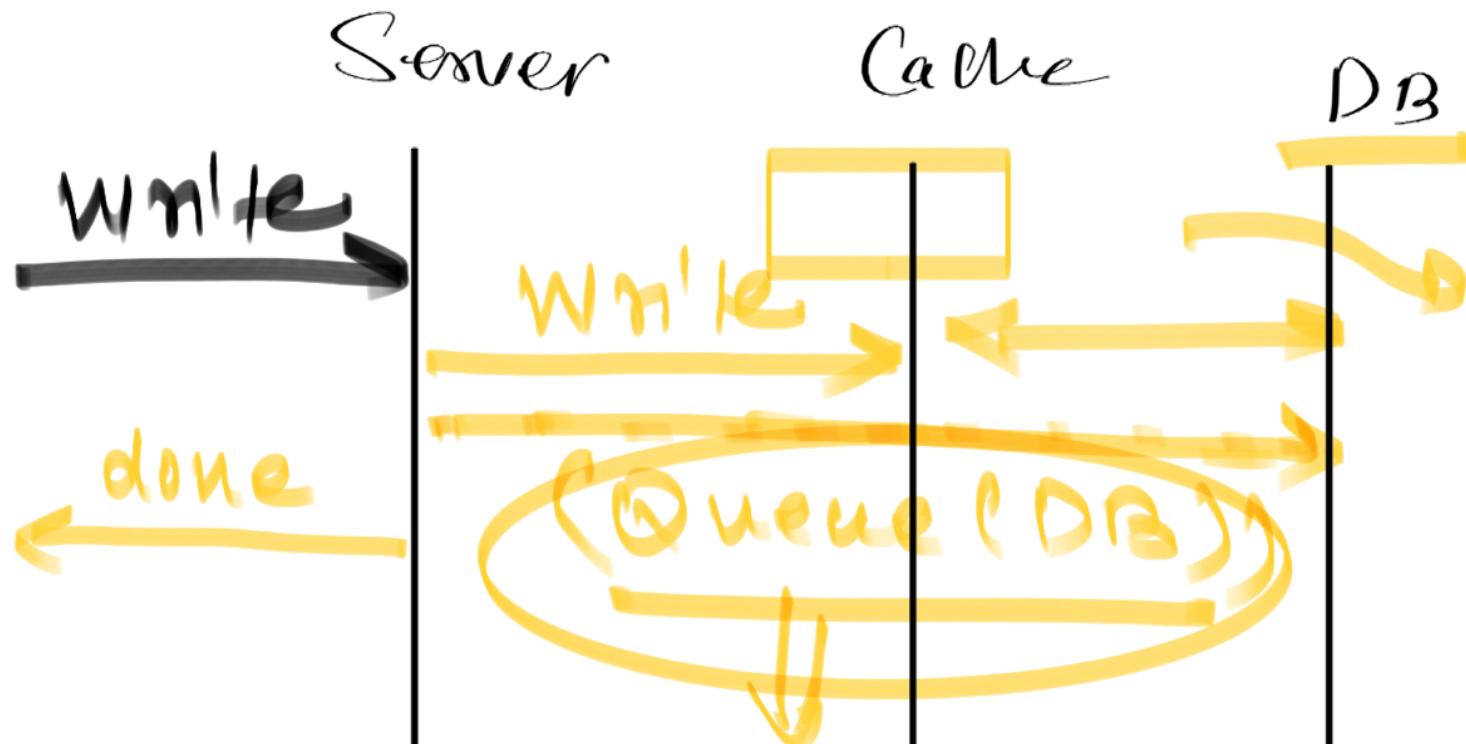
Any 1 is down
(System is down)

2 phase

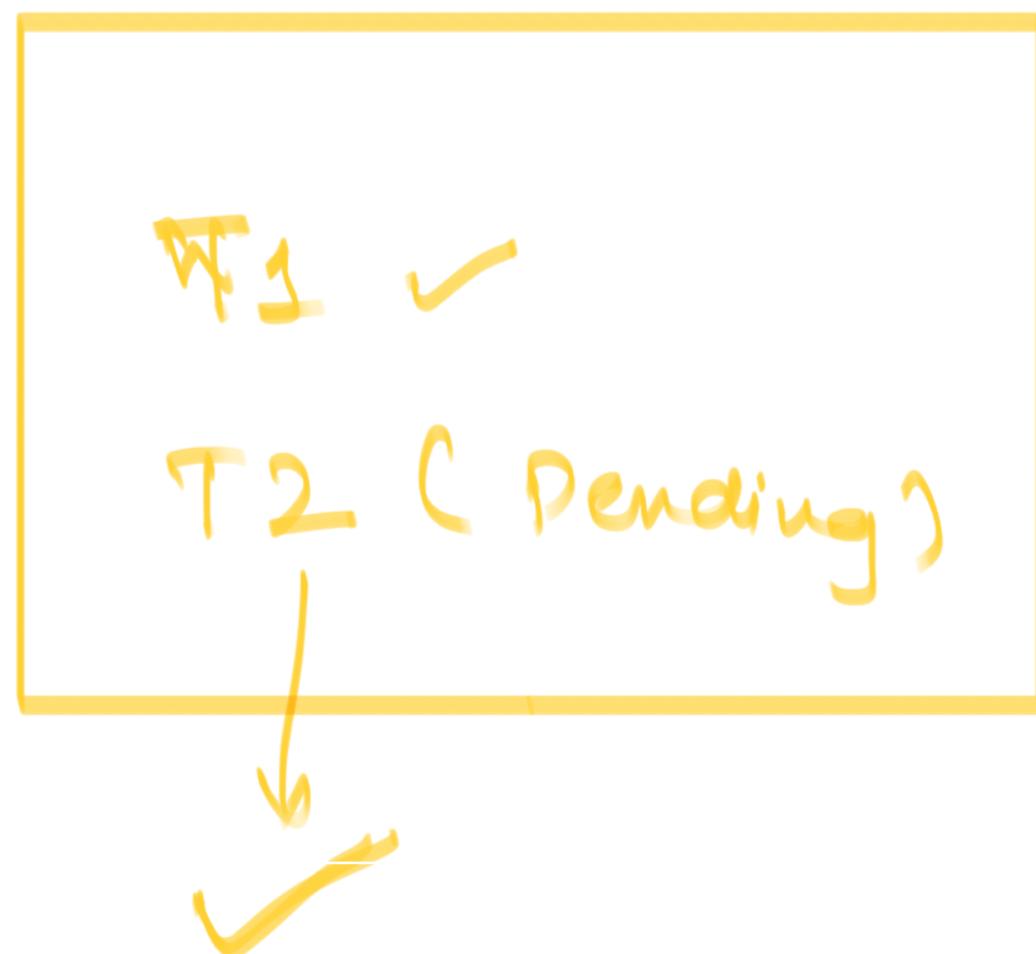
T1	(✓)
T2	(✓)



⑤ write Behind Cache



Messaging Queue



CRU cache
CPU cache
GPU
FAD

First In First Out

↑ last message

max



M₂

— · —

M₁₀₀

M₁₀₁

LIFO
(Last In First Out)

M₁

M₂

· · ·

M₁₀₀

M₁₀₁

(60%)

LRU Cache

M₁₀₁

Queue

146 ✓

30 min

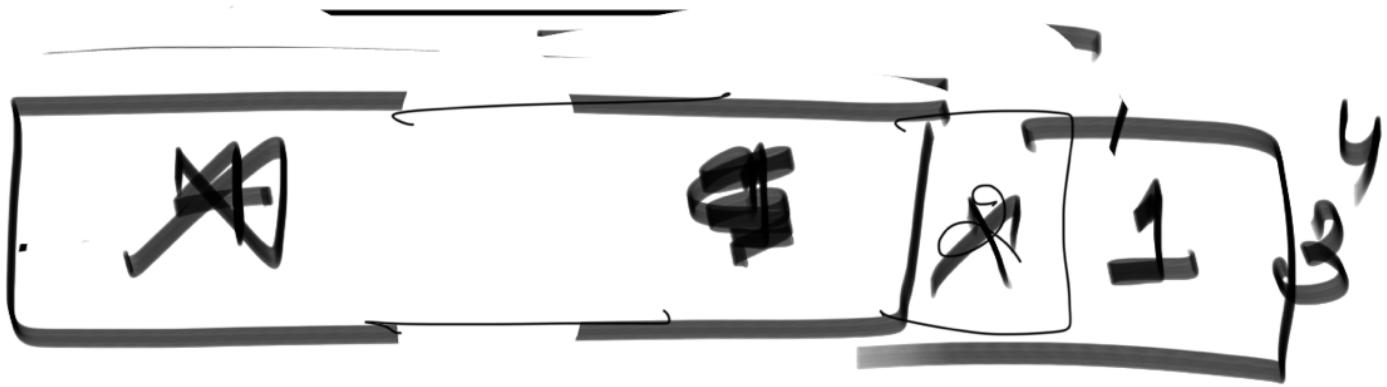
Theory

Stream

1 2 3 4 2 1 2 1 3 4

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

CACHE (size: 3)



FIFO (element do not exist
in cache)

✓ (we remove from
cache + enter Ab
Bu