

Solving Design Problems - Level I

Course on Solving Real World OOPs Design Problems

How to Approach a problem

Planning:

- Understand Problem First ✓ 5-10 min
- Ask Clarifying Questions →
- Don't directly jump to Implementation, think about the solution first
- Think about classes and possible design patterns/principles that can be applied
- You should be aware of the time too.

Implementation:

- Use Different Modules ✓
- Mark imp fields as Private and use getters/setters.
- You should have a driver class for testing your code
- Always have a plan to finish 30 mins before the time

Review:

- Start with different class u create and relationship between them
- Tell them about the principles/Patterns u have used → imp
- Interviewer can provide u custom test case
- There will be definitely some scope of improvement that Interviewer will find out,
So don't panic and do the necessary changes on the go.

This is not ~~Right~~ or ~~Wrong~~ Game, it is a game of discussing and optimising the code.

30-40
↓

easy

DP
X
X
X
X
X

Plan
5 min
20-30 min

manager

historian

model

practise

(2 hr) → (1.5) → (1.5) → (30 min)

→ Observer Pattern:-

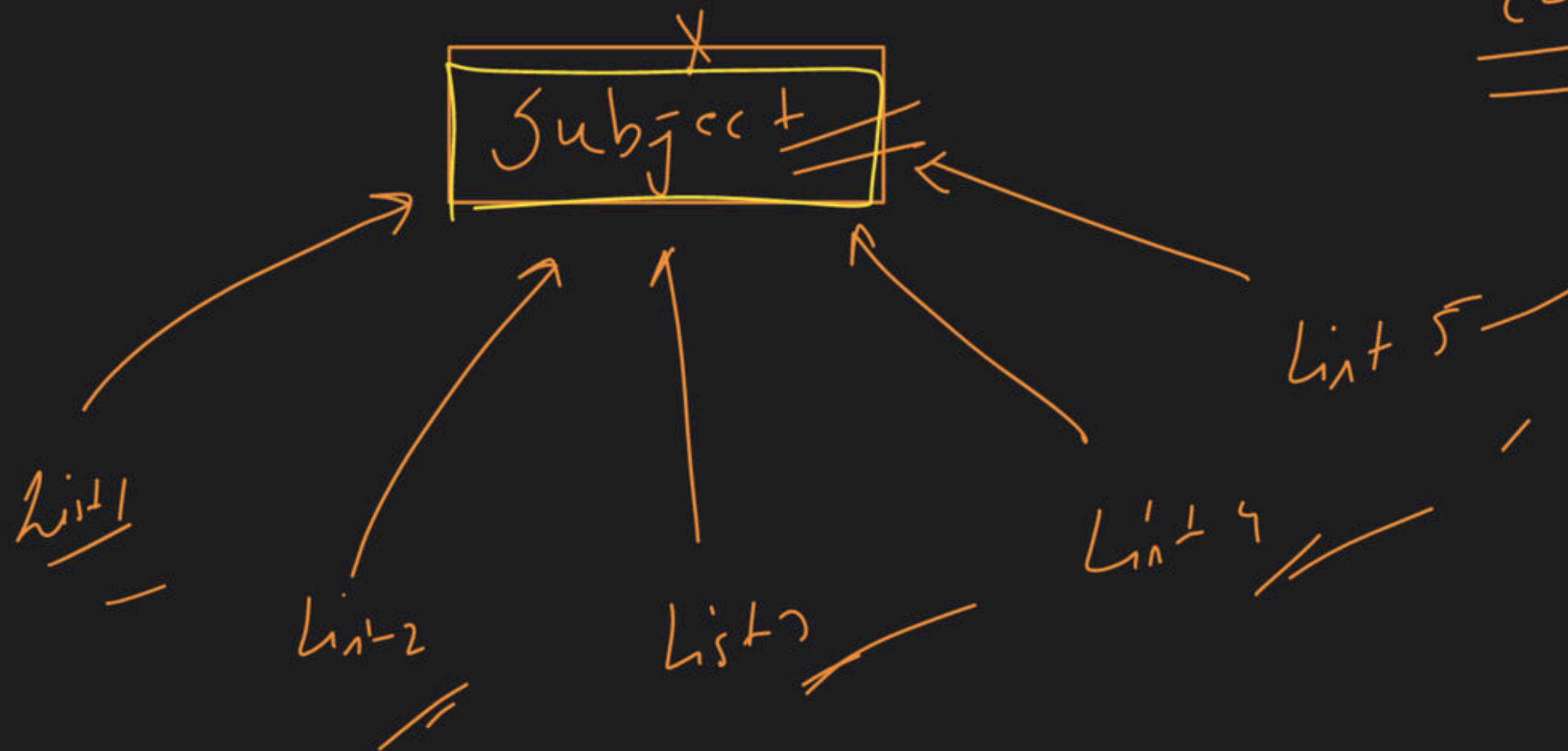
Publish
-Subscribe

→ one changed & all the subs get notified

↳ I¹ says

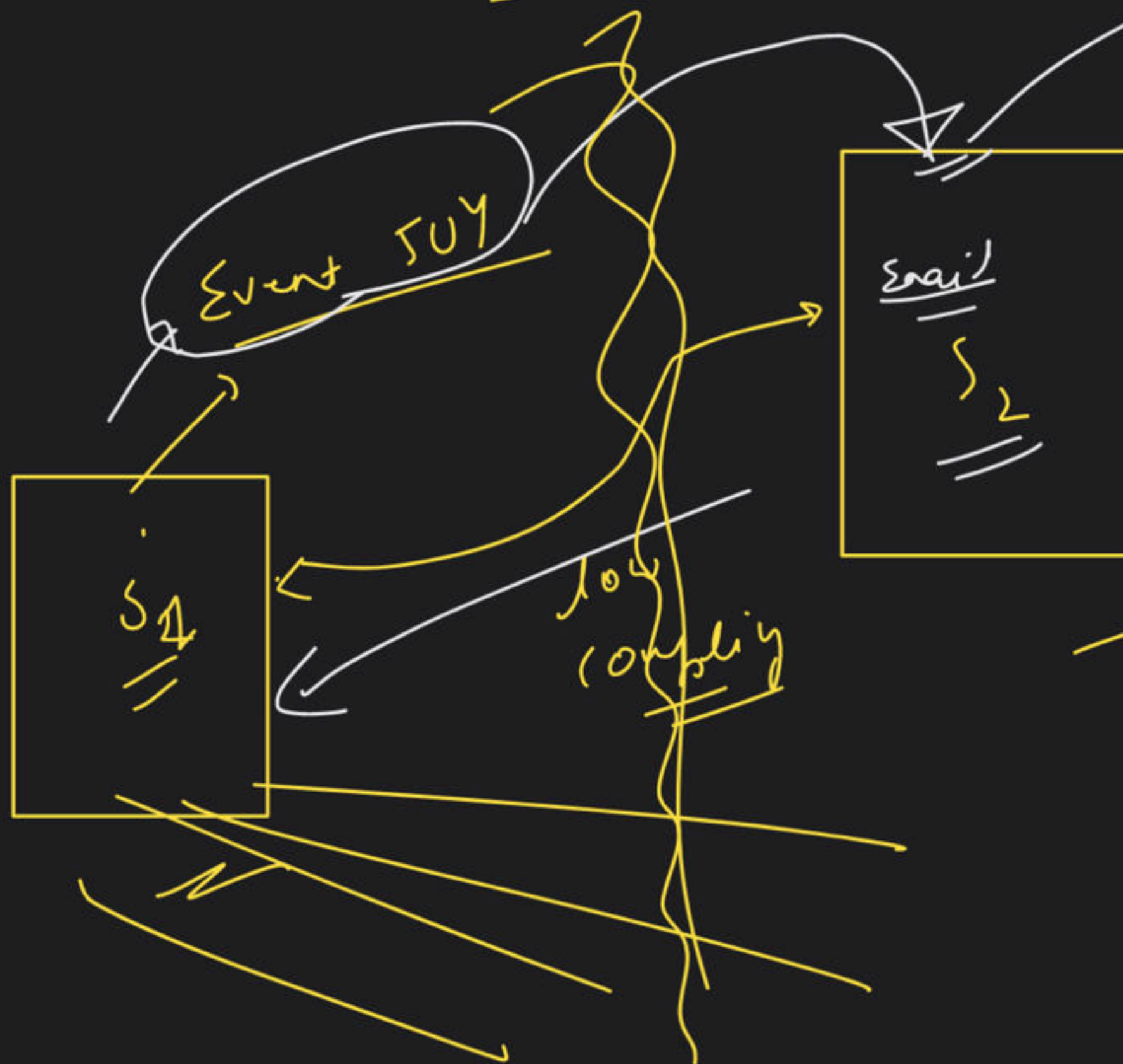
"Just define a 1:1 dependency so that when one object changes state, all its dependents are notified & update automatically"

Broadcast
comⁿ



Real world

Event-based Systems



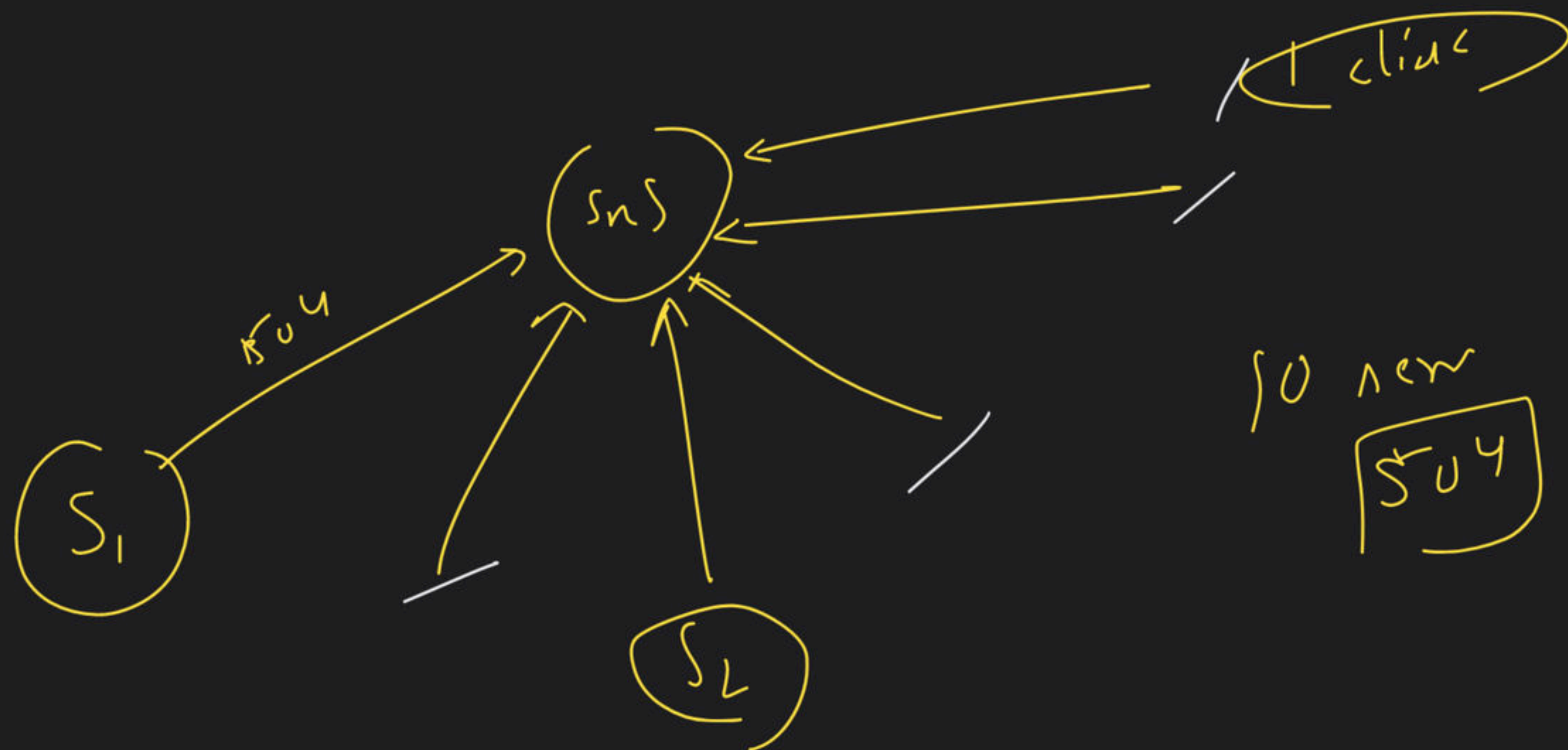
AWS

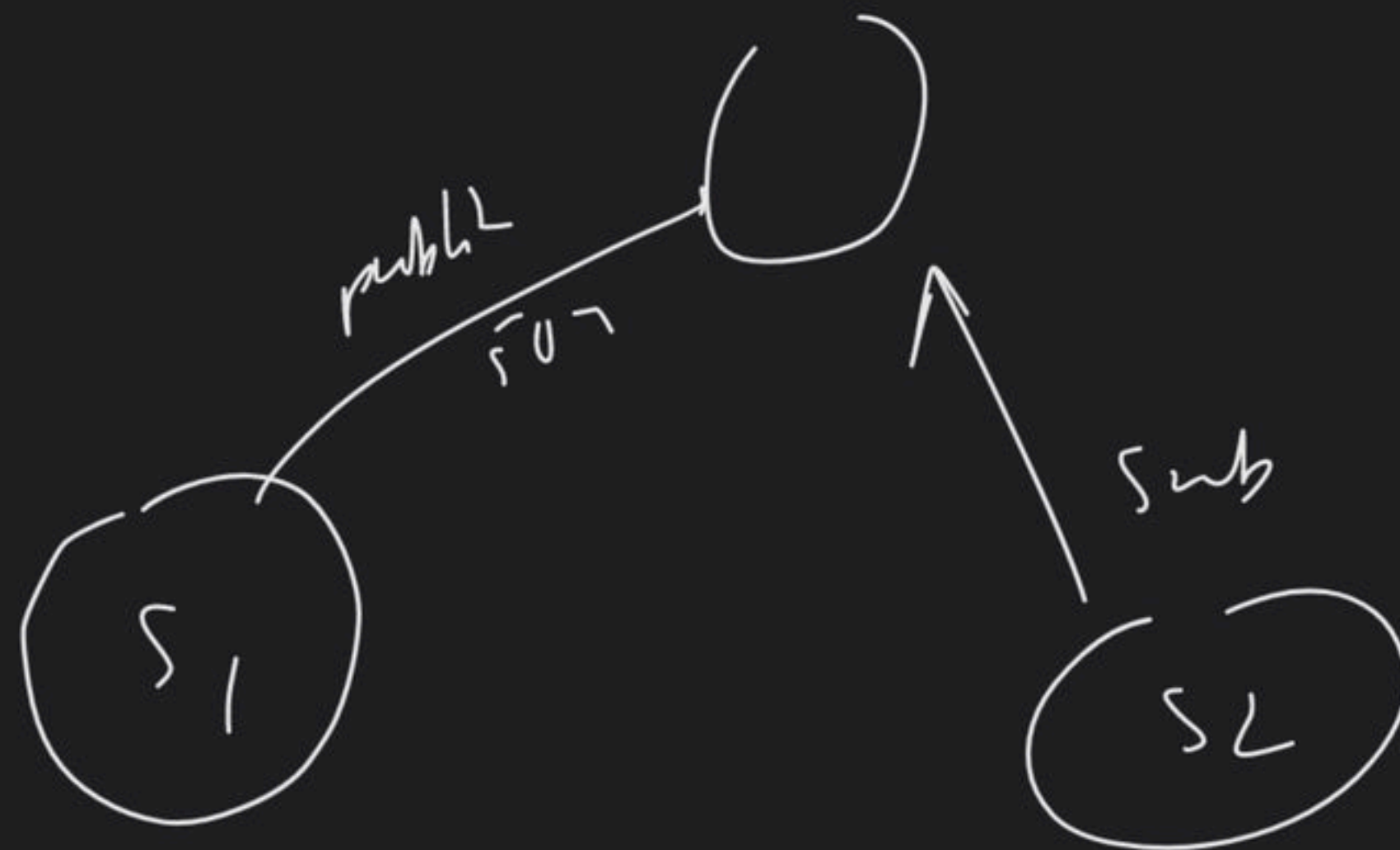
SNS

SQS

Event

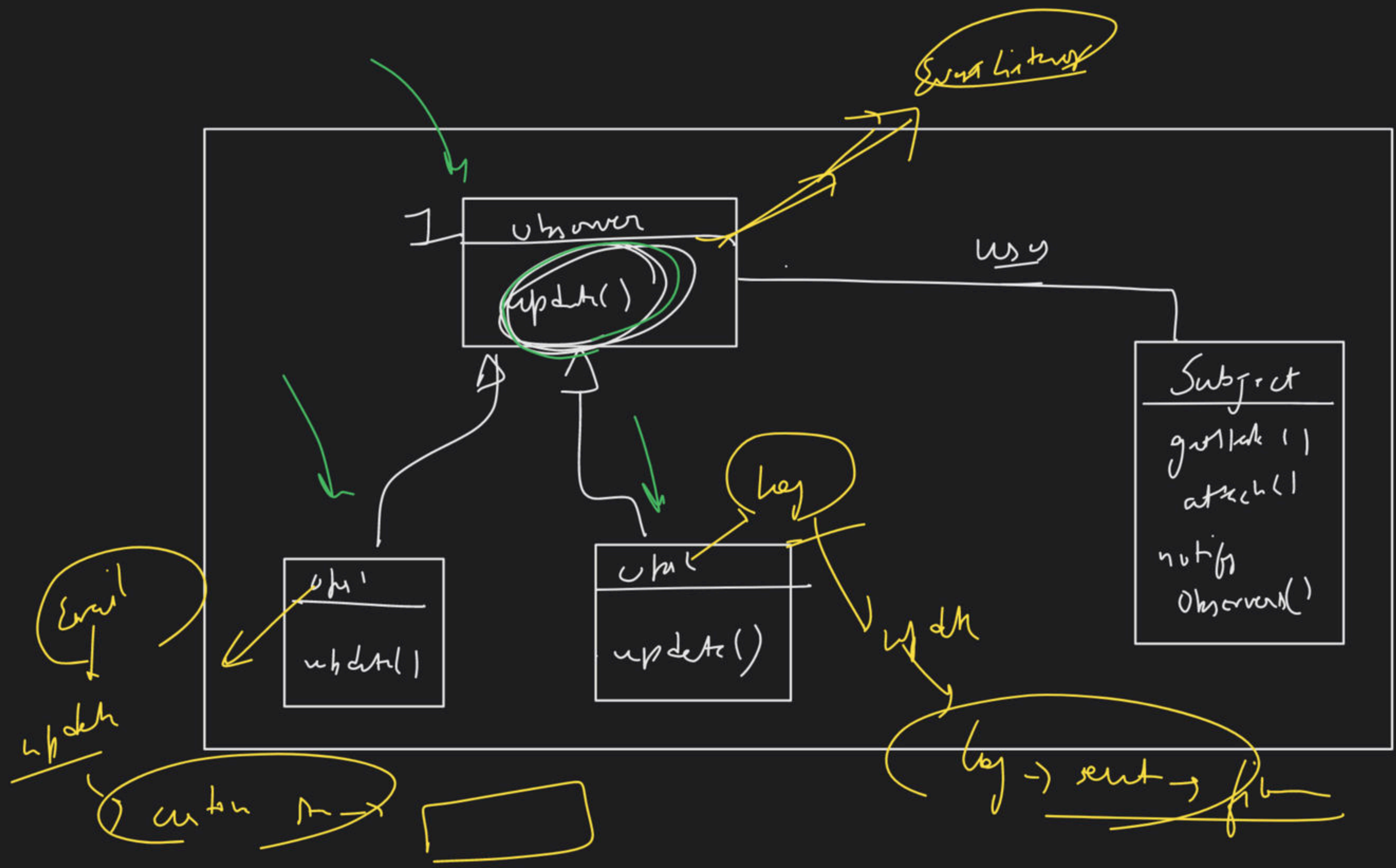
Action

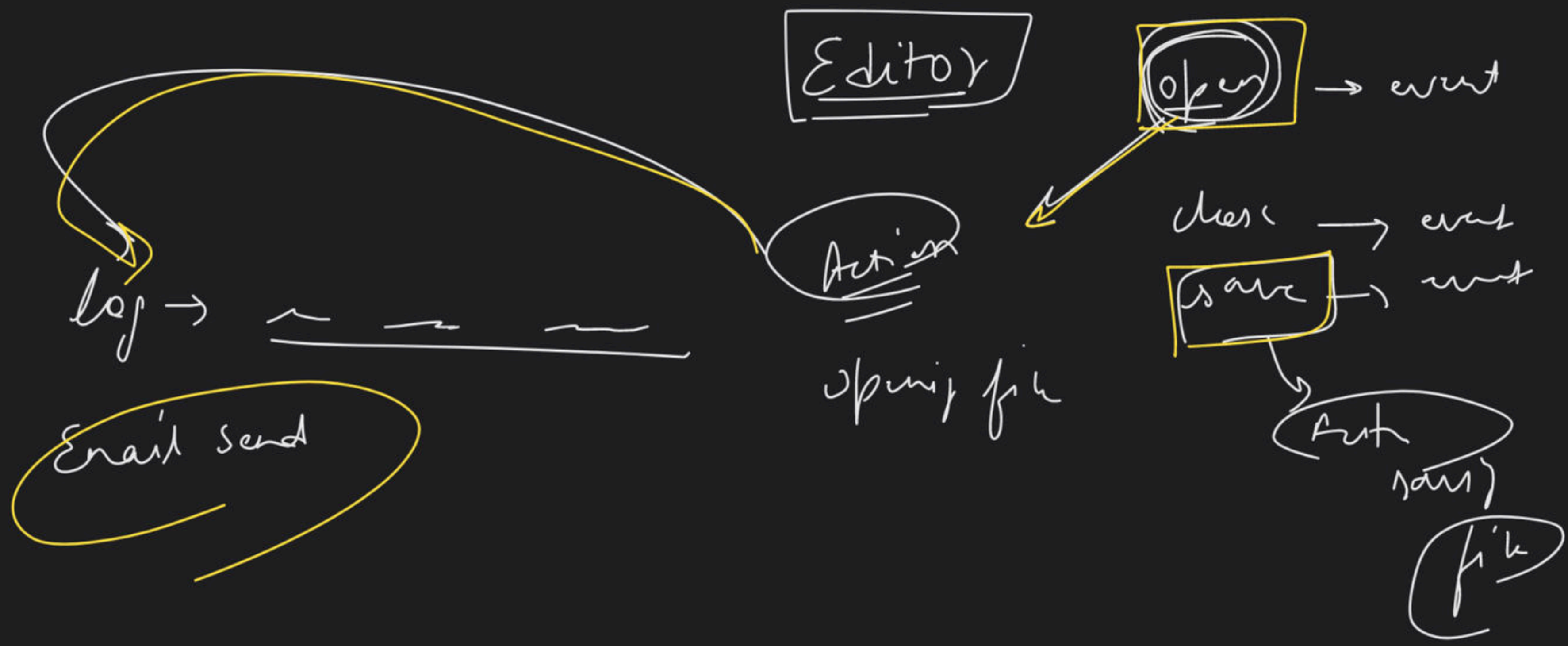




low - Couper
→ subjed - linker
→ Bradley
type
cont =



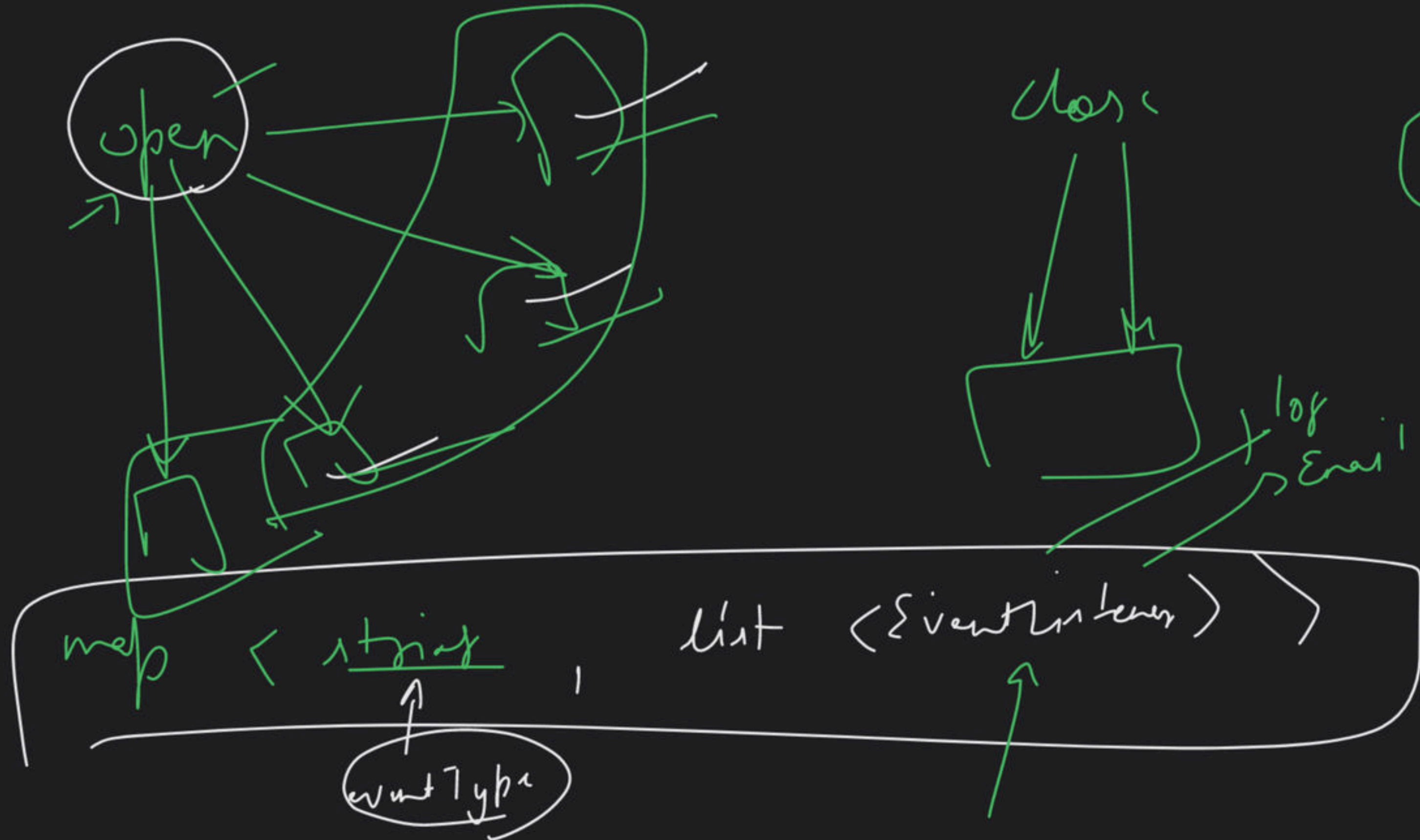


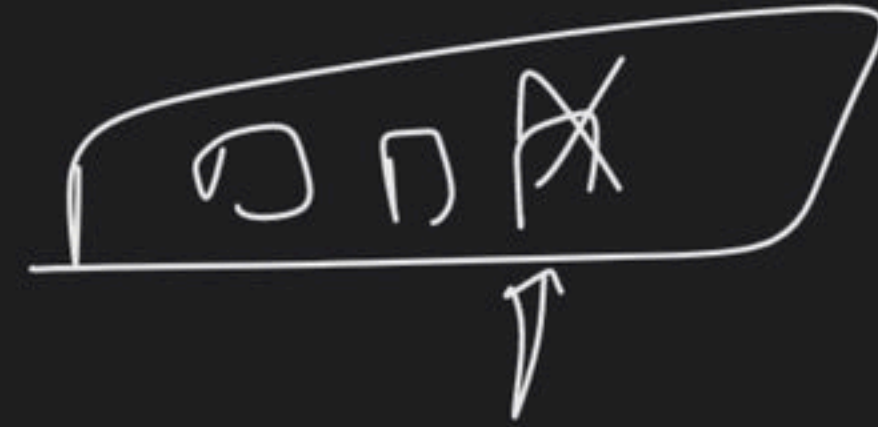
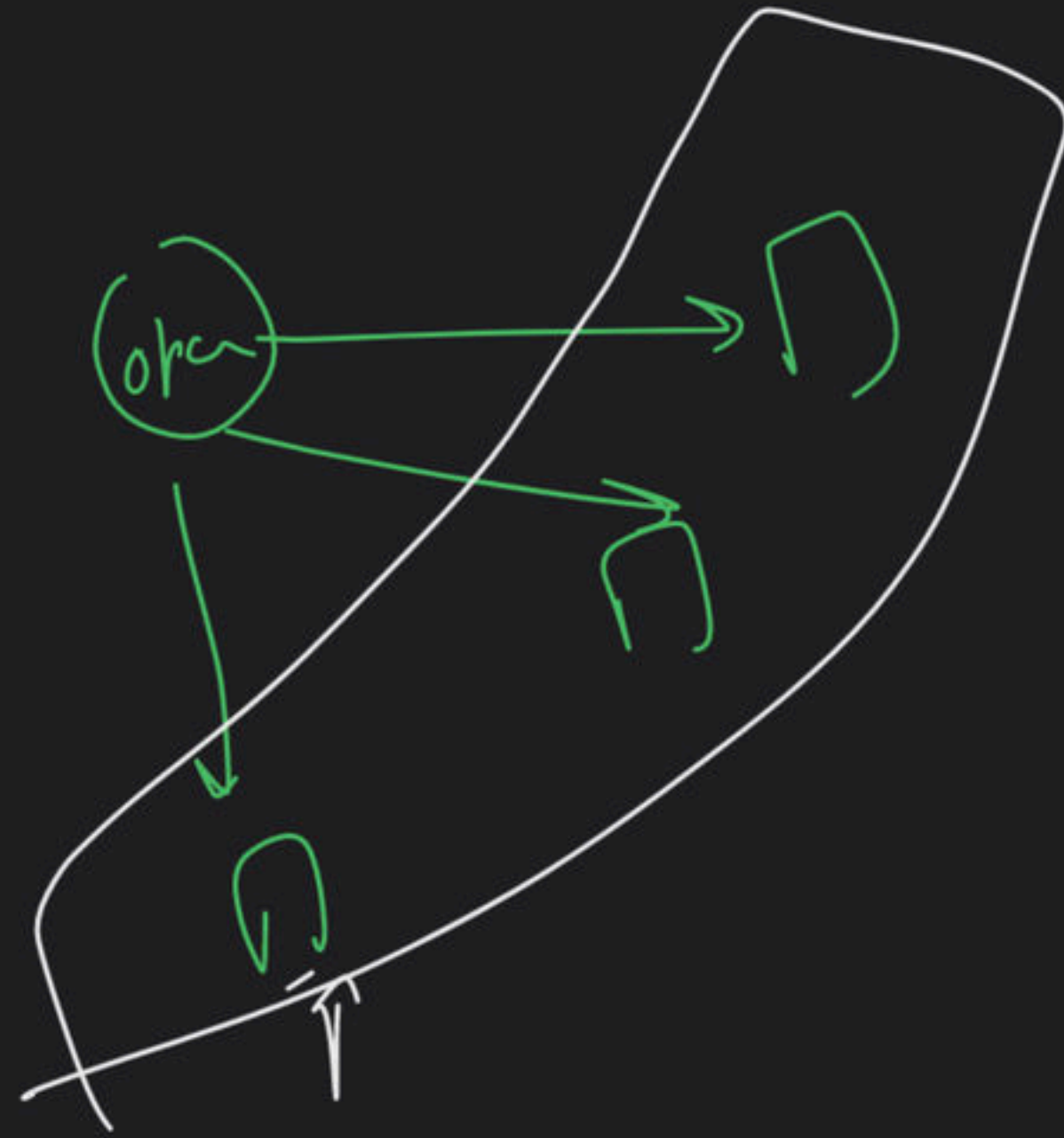


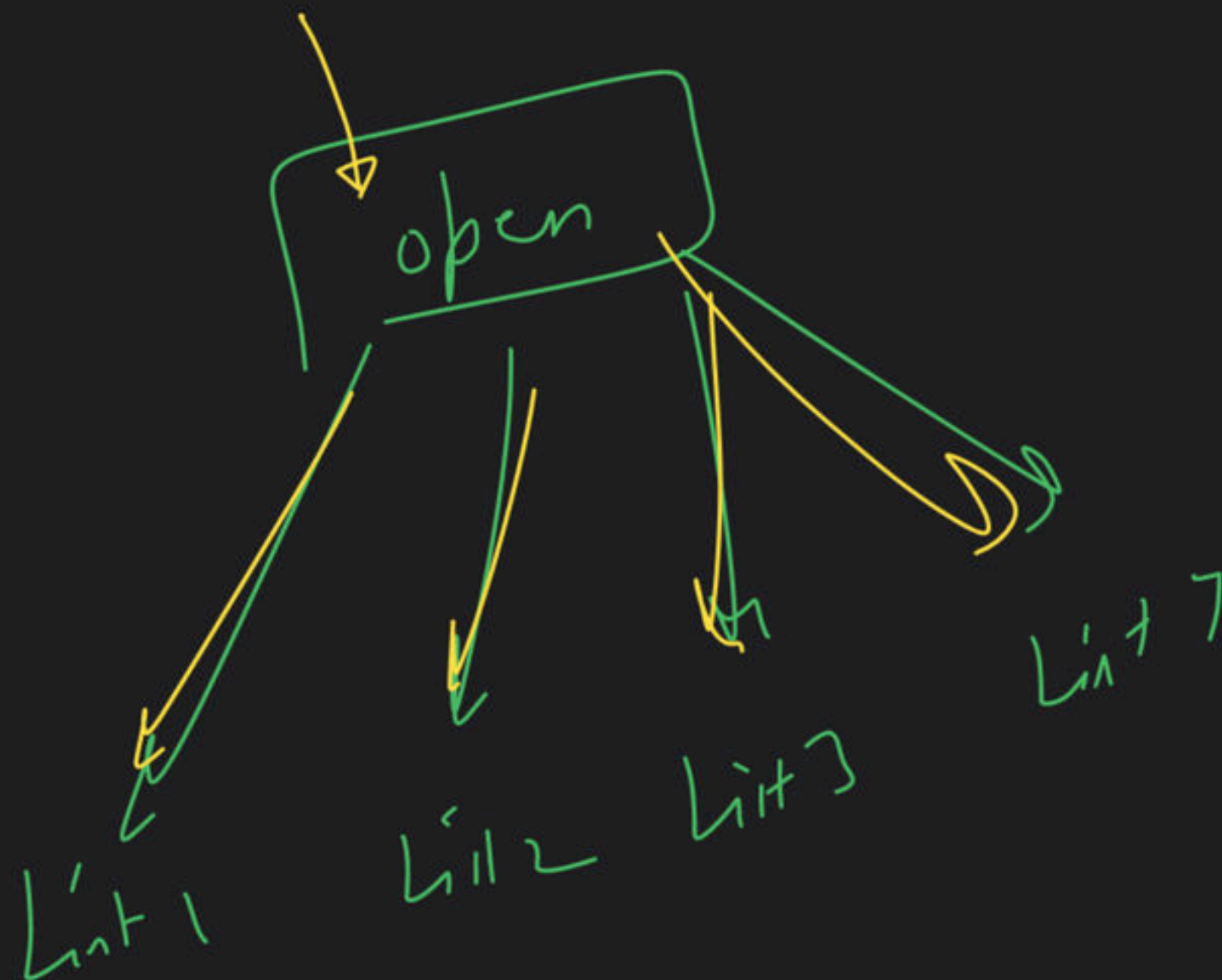
Event Manager

event → listener

save
↓
[]







Lit 4

→ Doubts:-

































In-Memory Cache:

Description:

Babbar Bhaiya got a new client for his web-services company. Clients wants to have a efficiently designed In-Memory Cache system, which he/she can invoke with the driver class. Help babbar bhaiya to deliver this task.

Requirements:

- create a user-defined size cache and with Eviction policy as "LRU"
- support fetch/Insert/Delete/Clear functionality in the system.

Least Recently
Used

Other Details:

- Do not use any database or NoSQL store, use in-memory store for now.
- Do not create any UI for the application.
- Write a driver class for demo purposes. Which will execute all the commands at one place in the code and test cases.
- Please prioritize code compilation, execution and completion.
- Please do not access the internet for anything EXCEPT syntax.
- You are free to use the language of your choice.
- All work should be your own. If found otherwise, you may be disqualified.

Expectations:

- Code should be demo-able (very important)
- Complete coding within the duration of 90 minutes.
- Code should be modular, with Object Oriented design.
- Maintain good separation of concerns.
- Code should be extensible. It should be easy to add/remove functionality without rewriting the entire codebase.
- Code should handle edge cases properly and fail gracefully.
- Code should be readable.
- Follow good coding practices: Use intuitive variable names, function names, class names etc. Indent code properly. Once the code is complete please zip the source code and upload it to: xxxyyyzzz.com.

2
minutes

→ Cache

* Size

* LRU

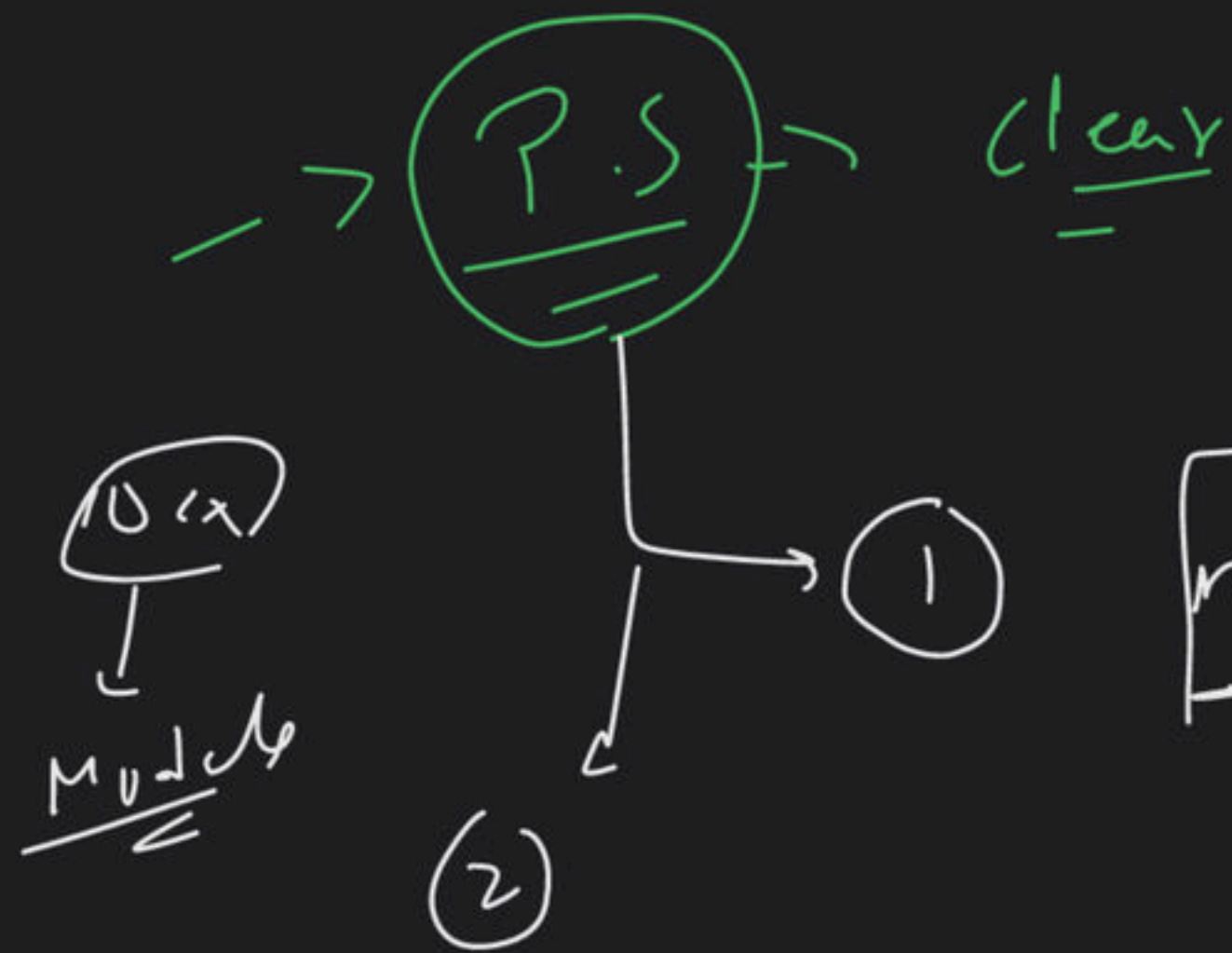
① get

② insert

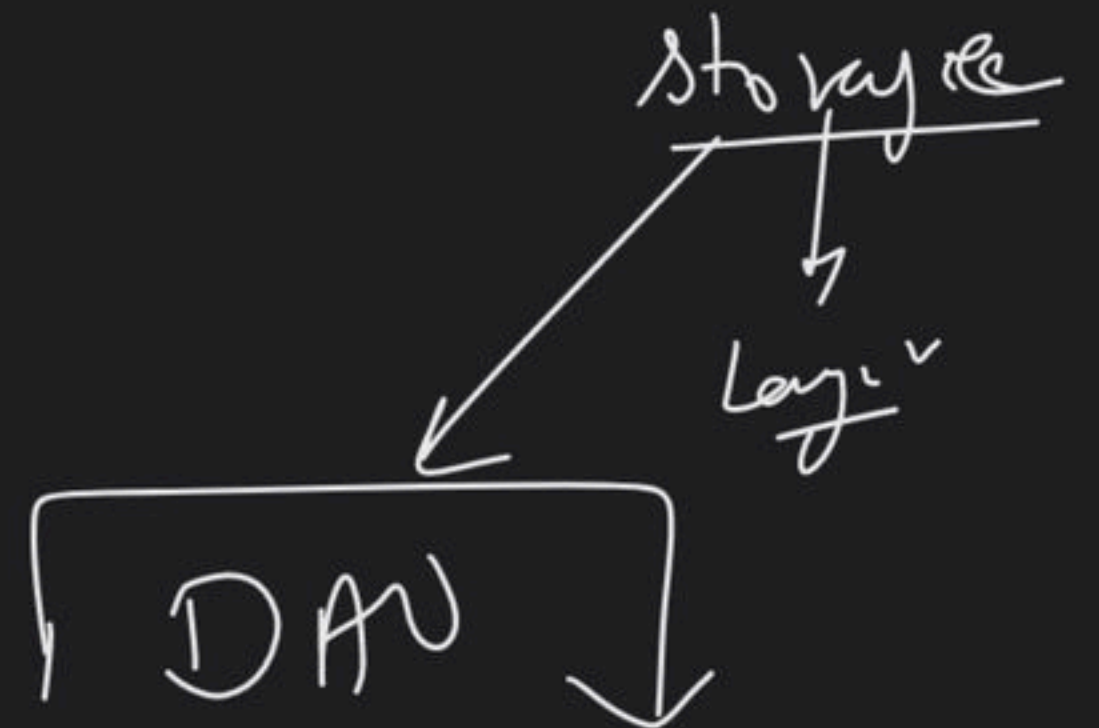
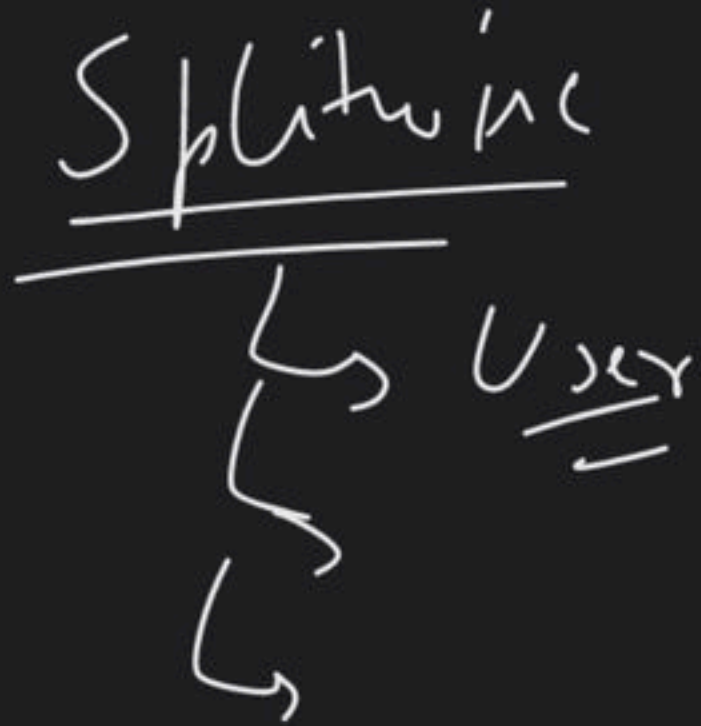
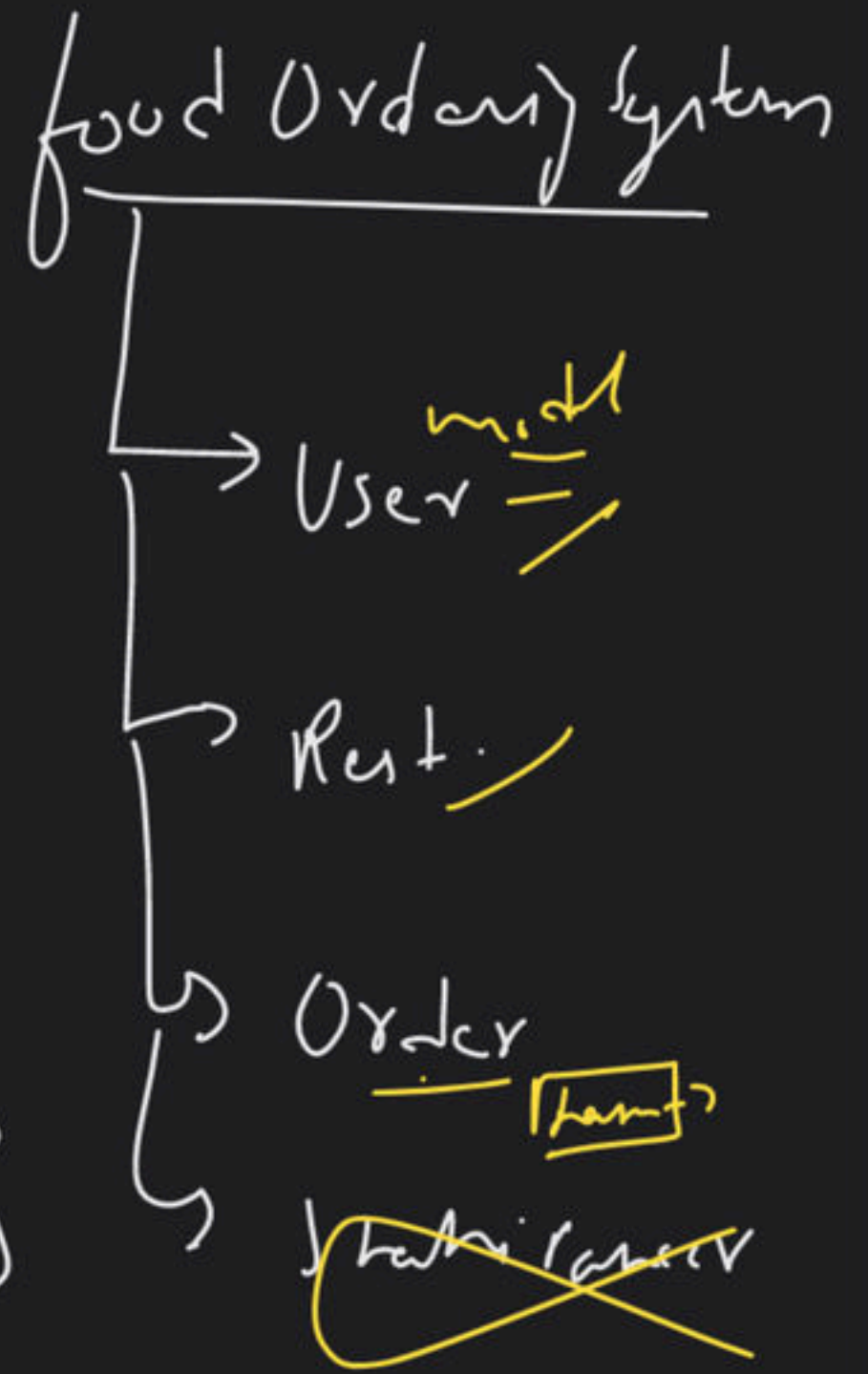
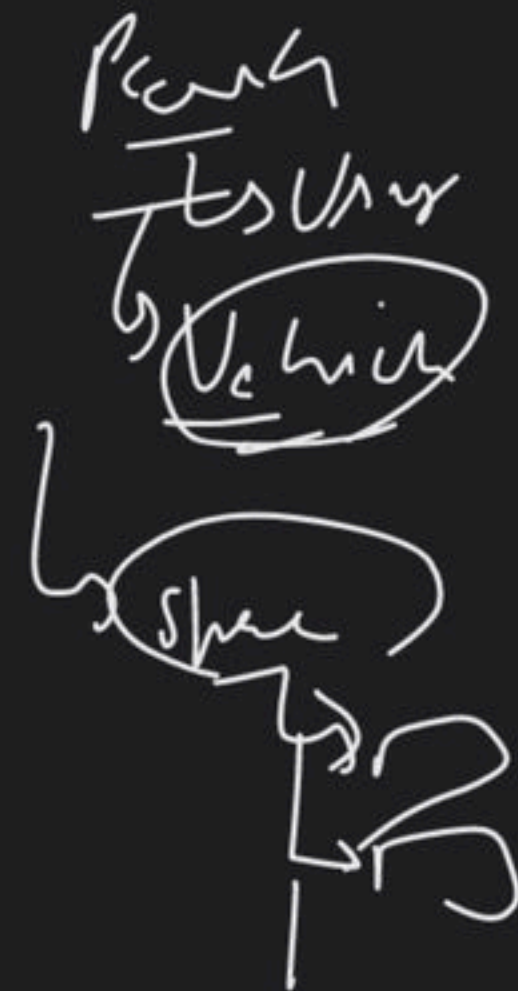
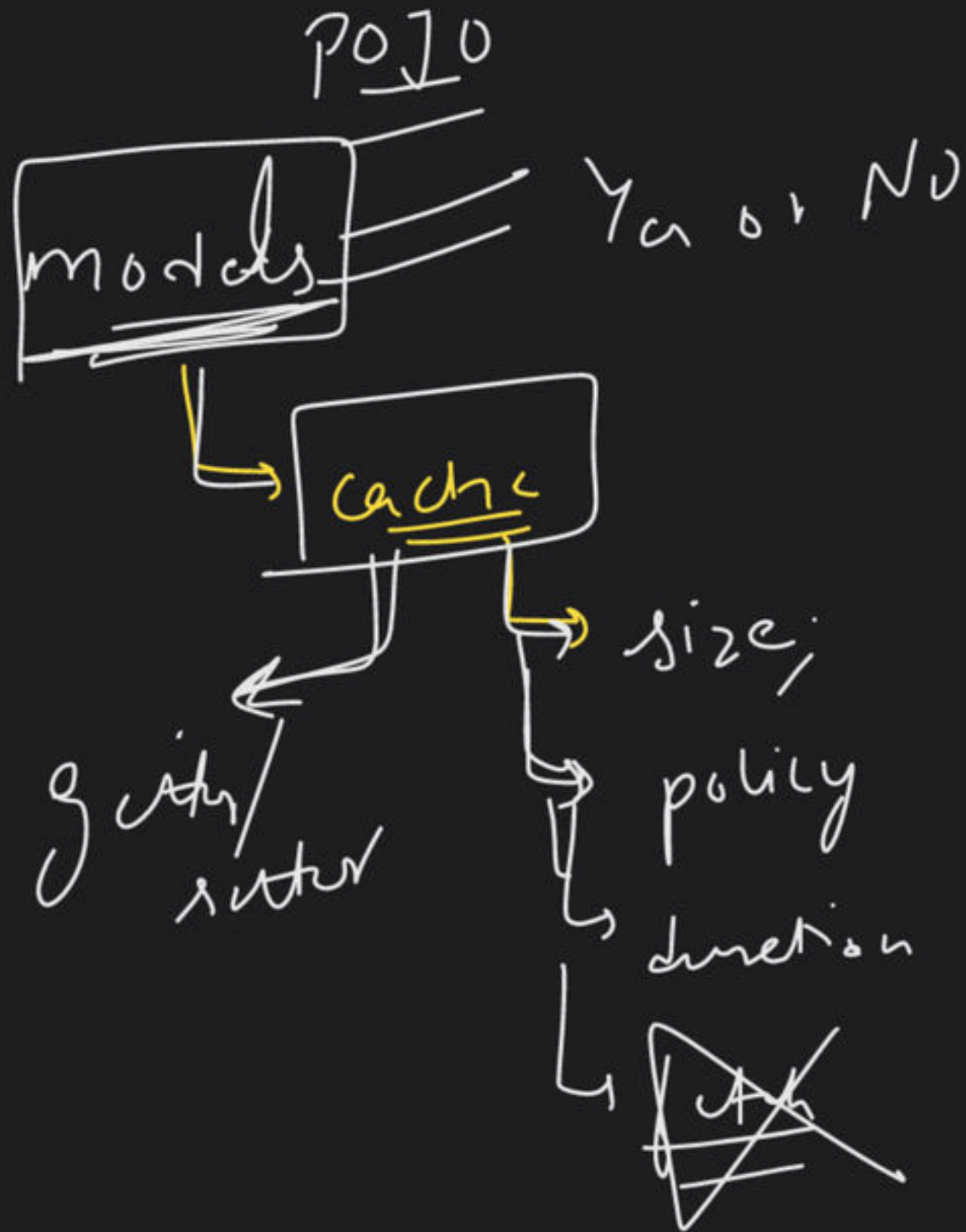
③ delete

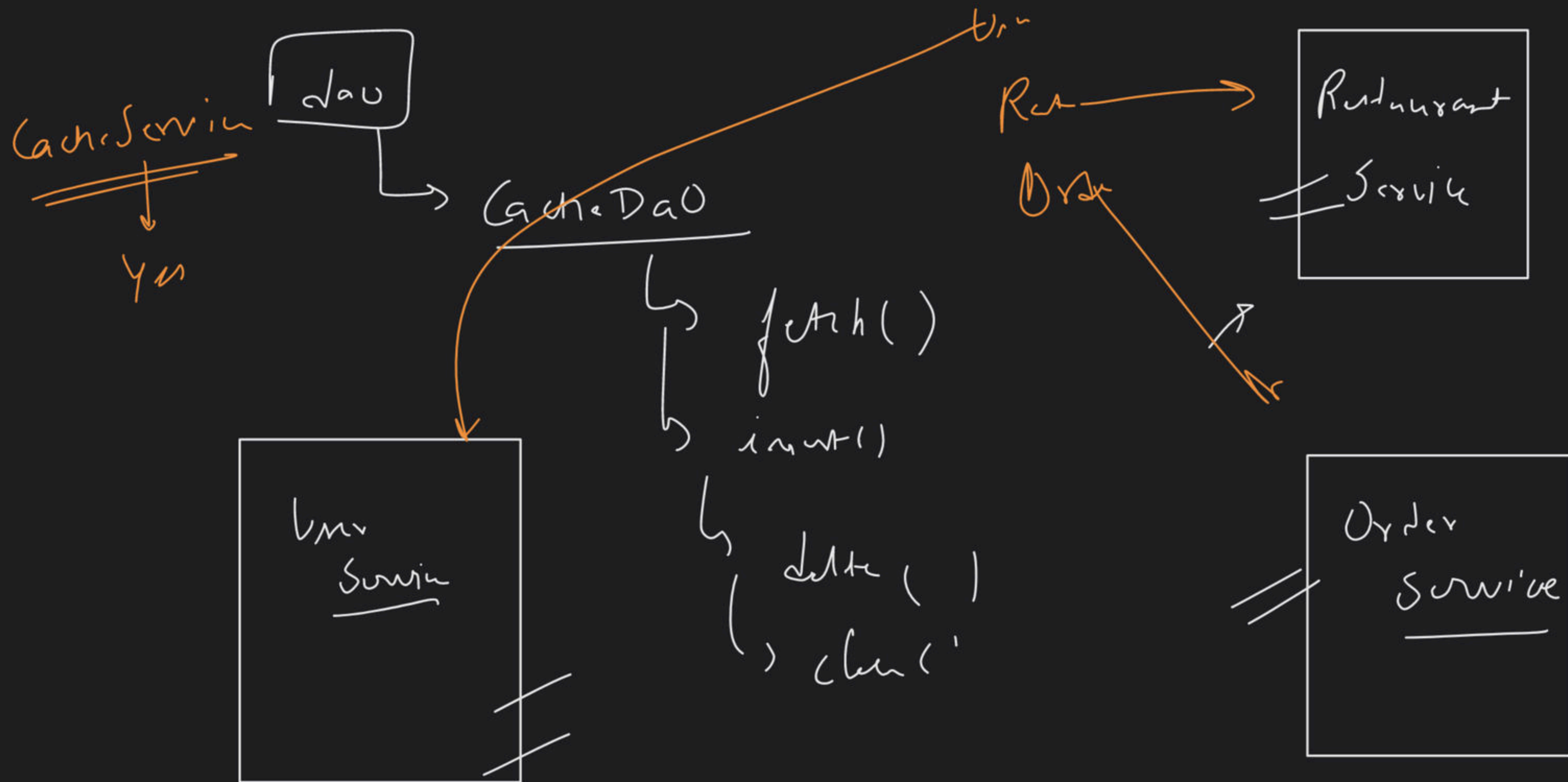
④ clear

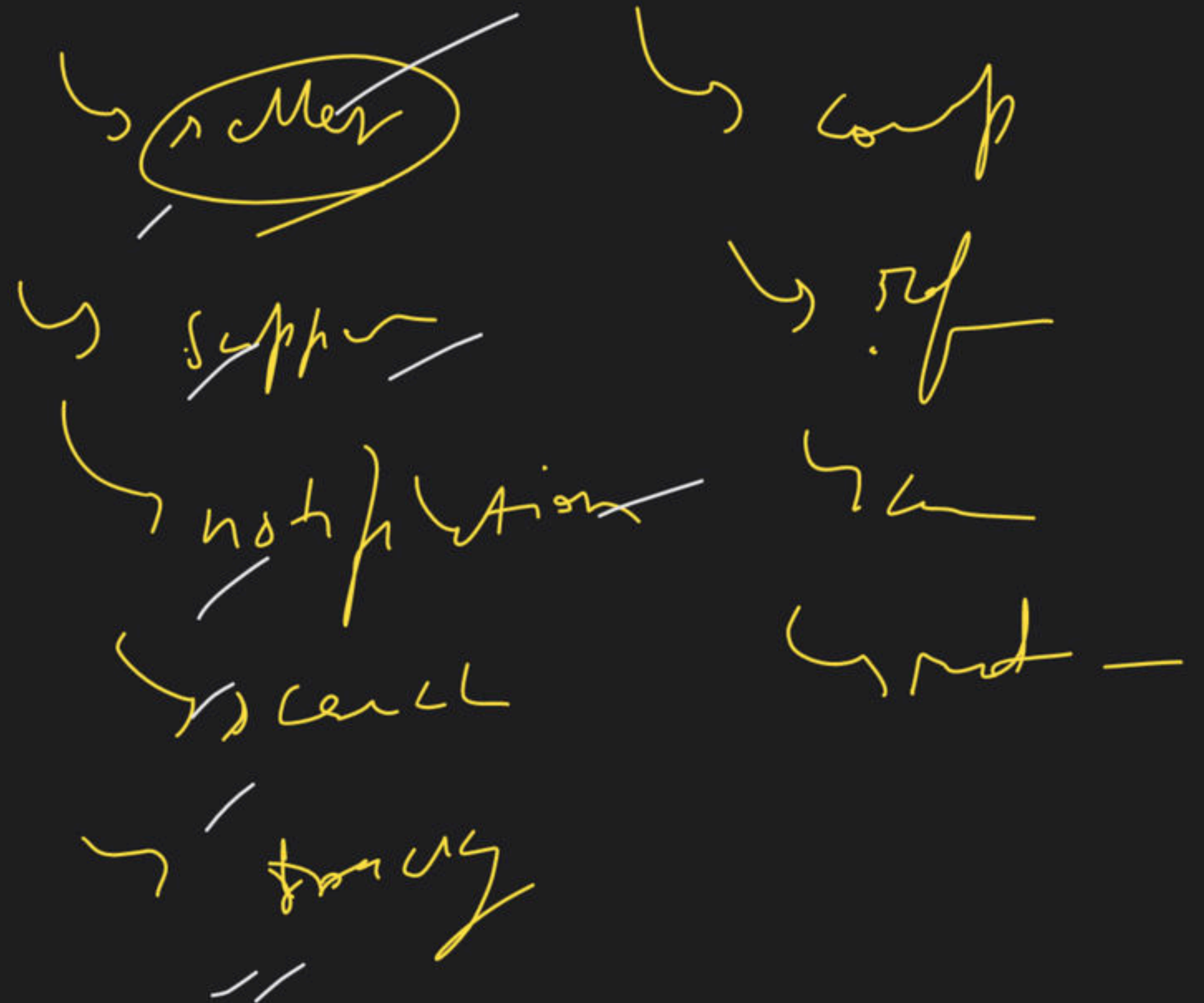
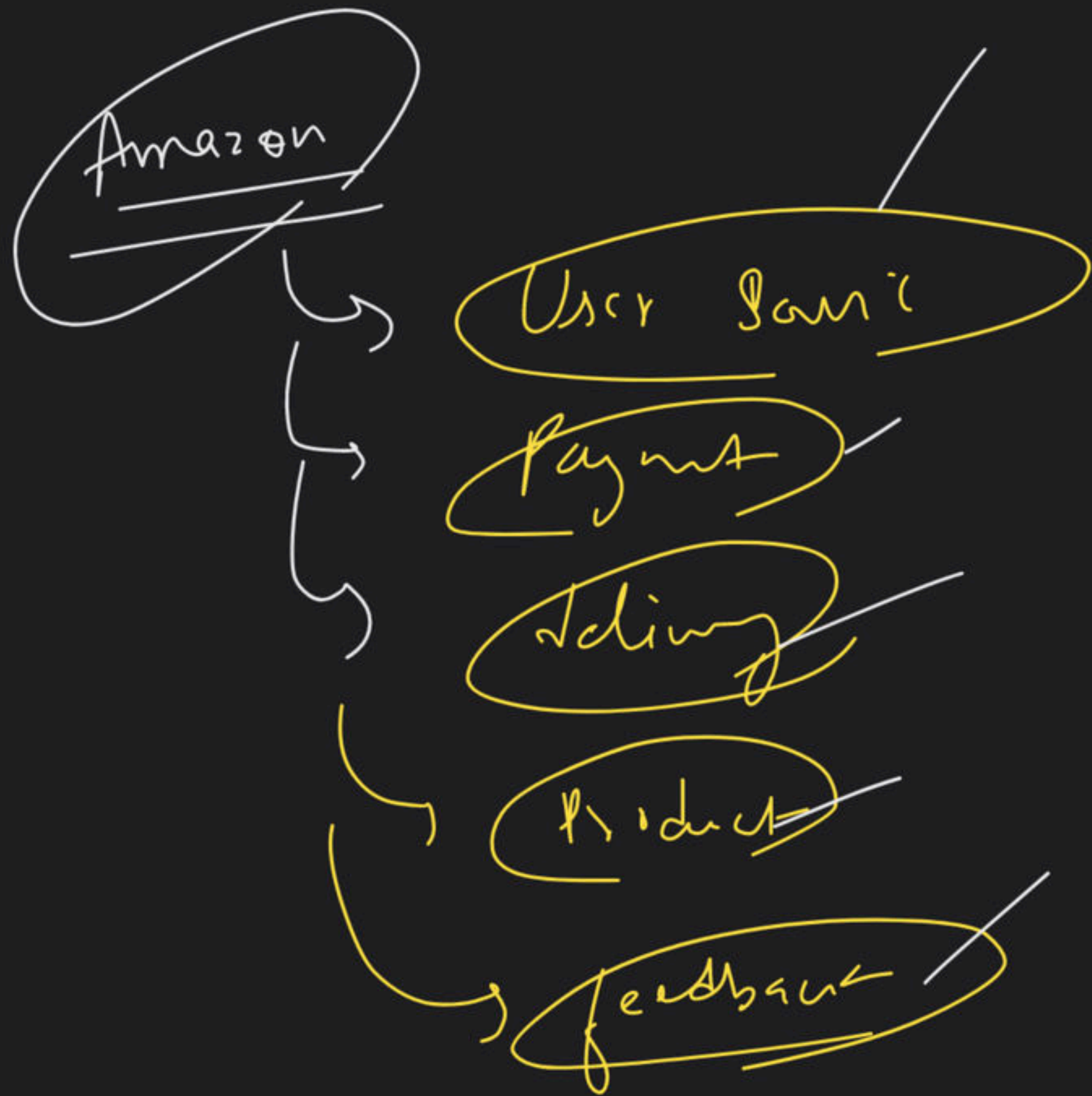
Split
four
Cats
PWL
console
Shur



Cache







for Shipping → date, user, pay, transaction, order, item, price, weight

→

RCE

LeetCode

→ User

→ Query

→ Sub

→ Review

Compiler engine

Subs --

Why

mode

test case

per

query bank

initial

→ In

US = v

(iv) ~~header~~

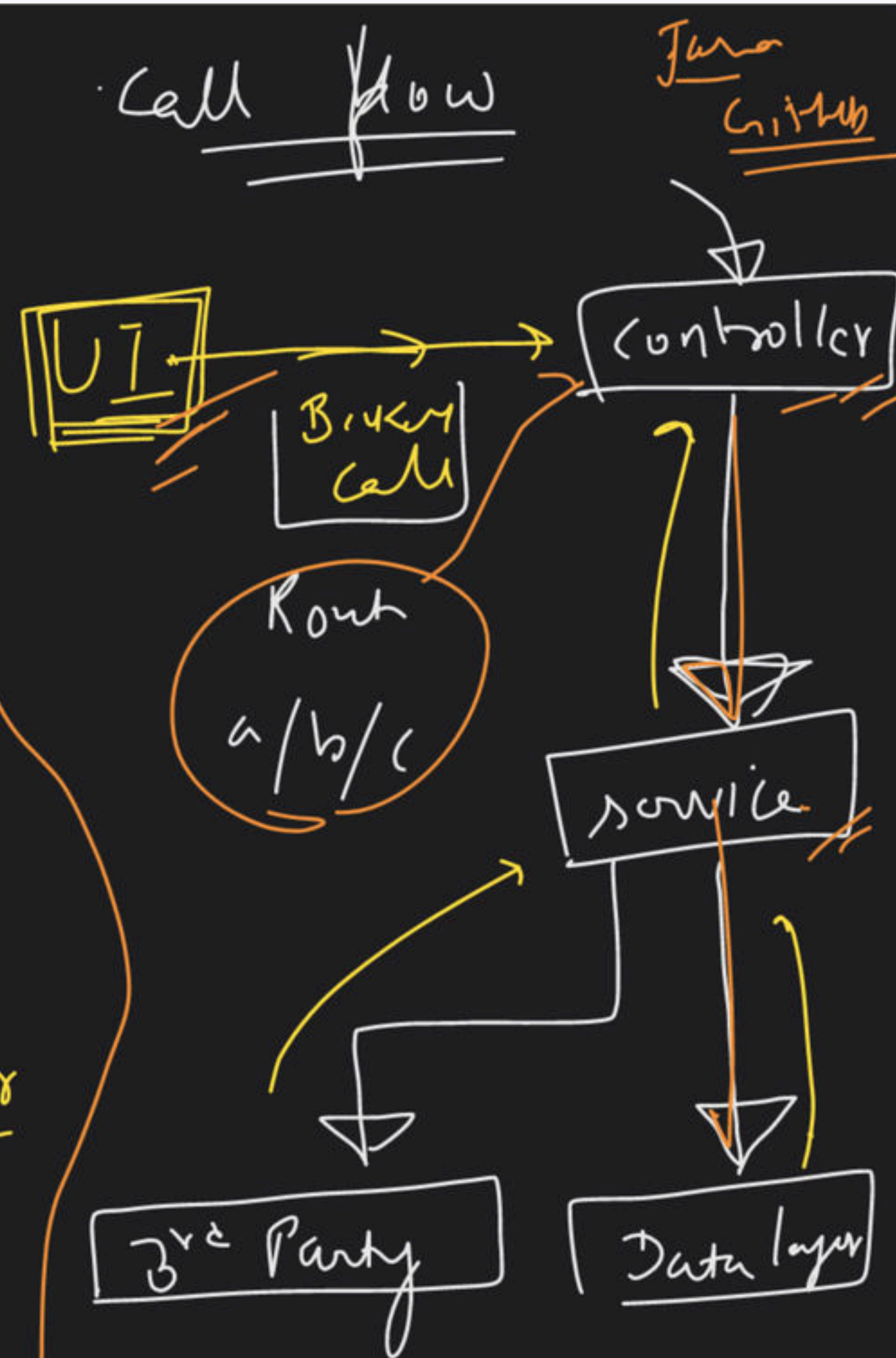
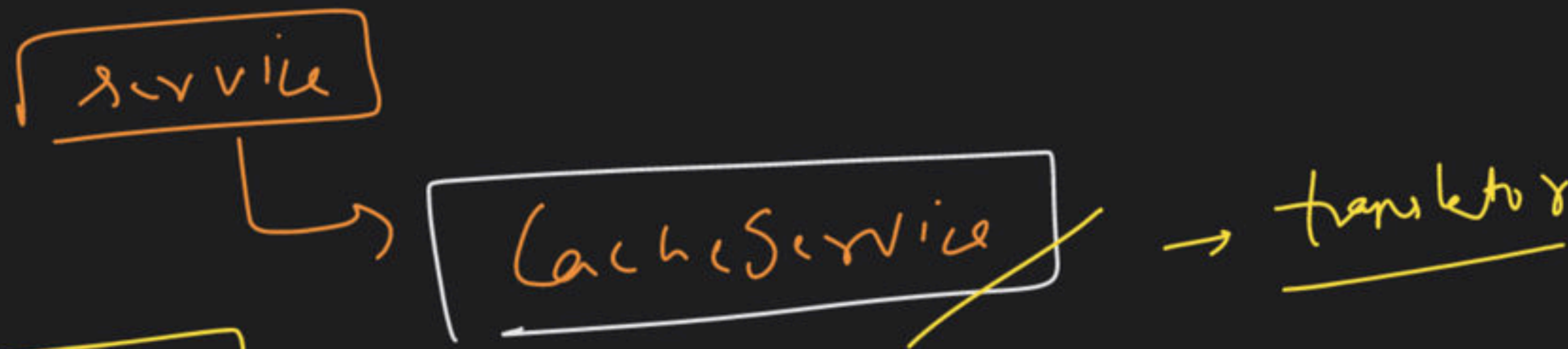
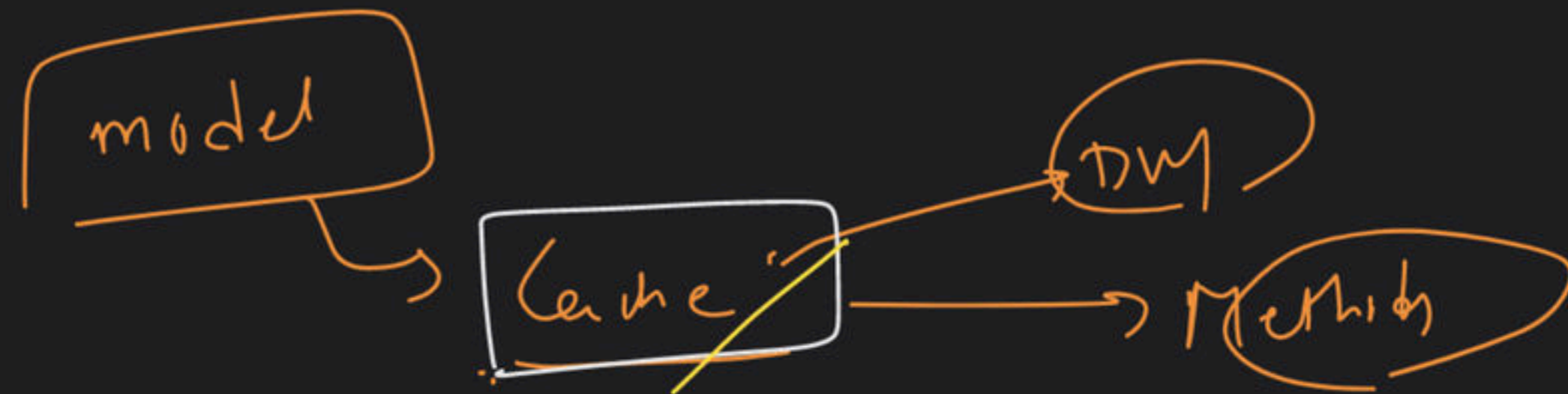
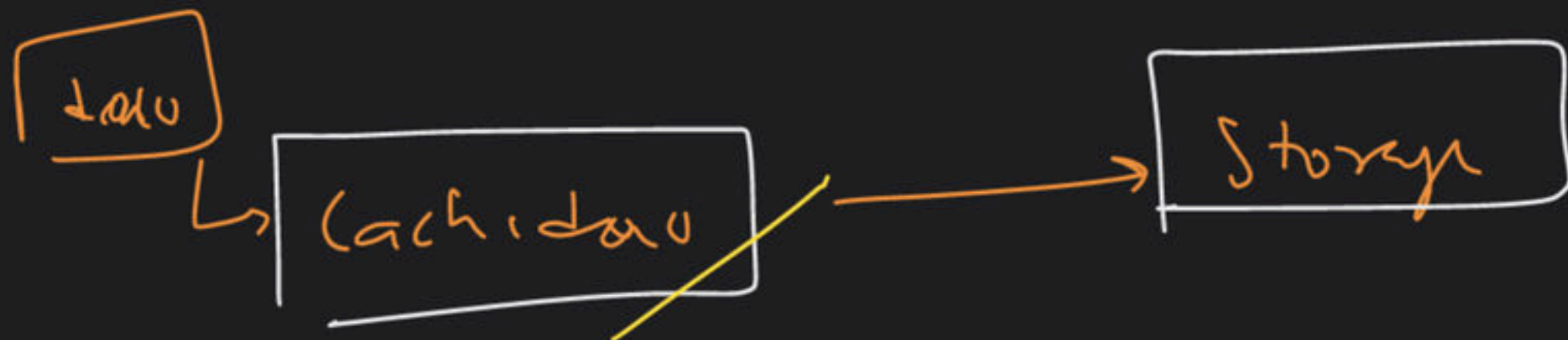
(iii) - content

(ii) - config

(i)

public enum Policy {
LRU, ...
}

LRU -> enum
anonymous -> bhoot
hard



on SI and hakea

DAU

Duty

6-7 yrs

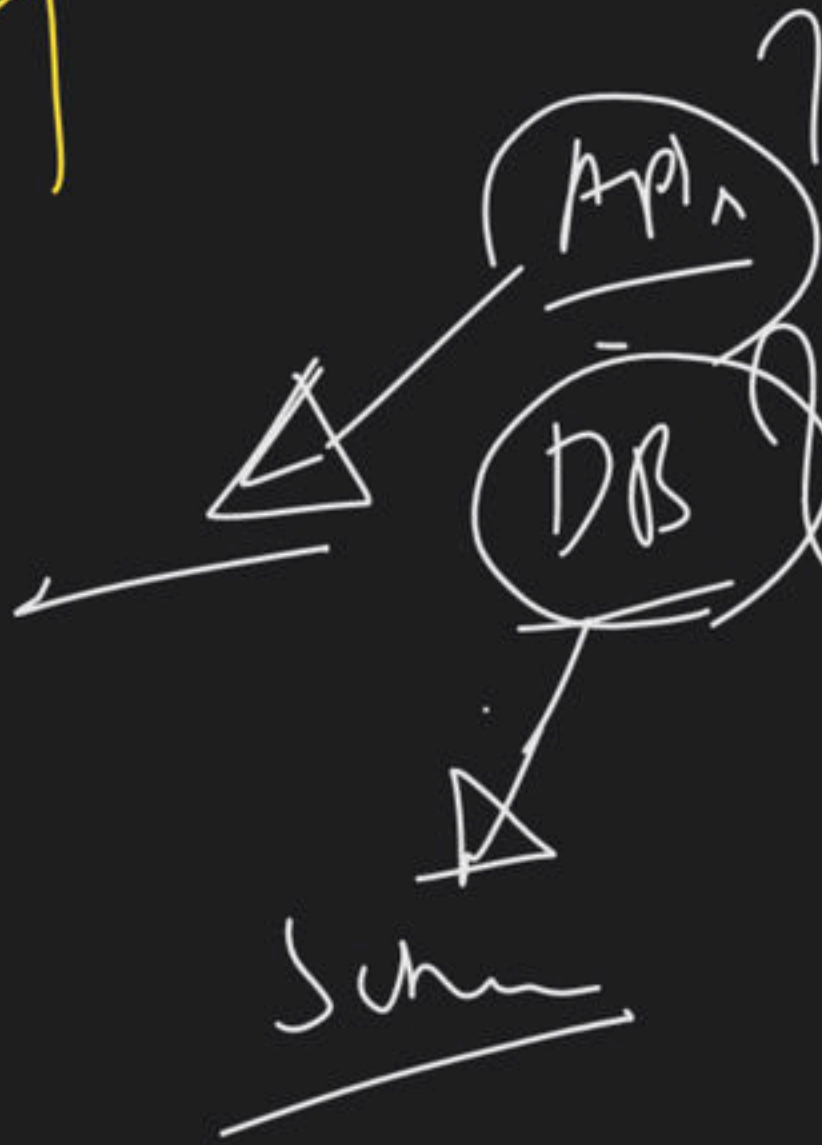
17 years

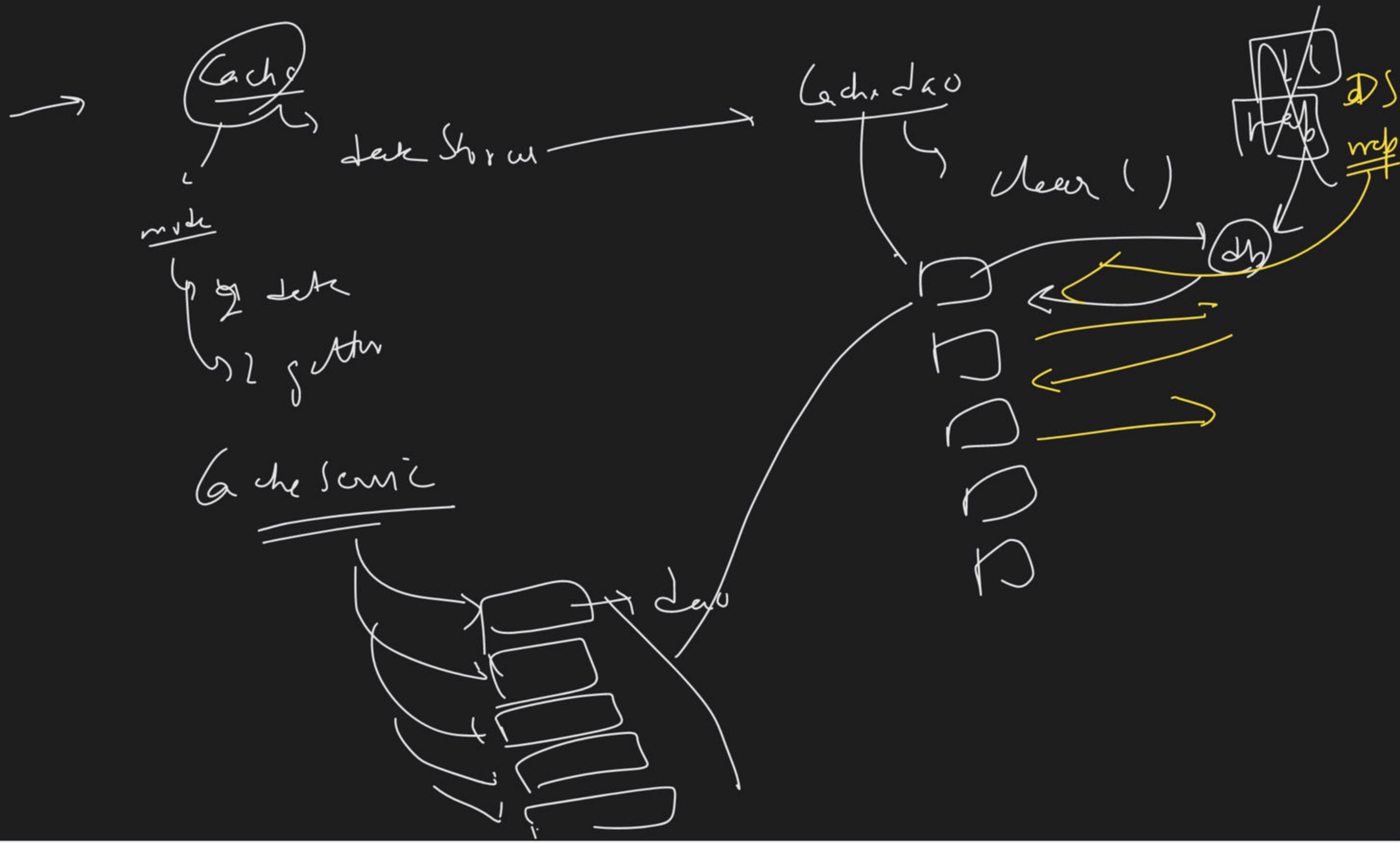
func()

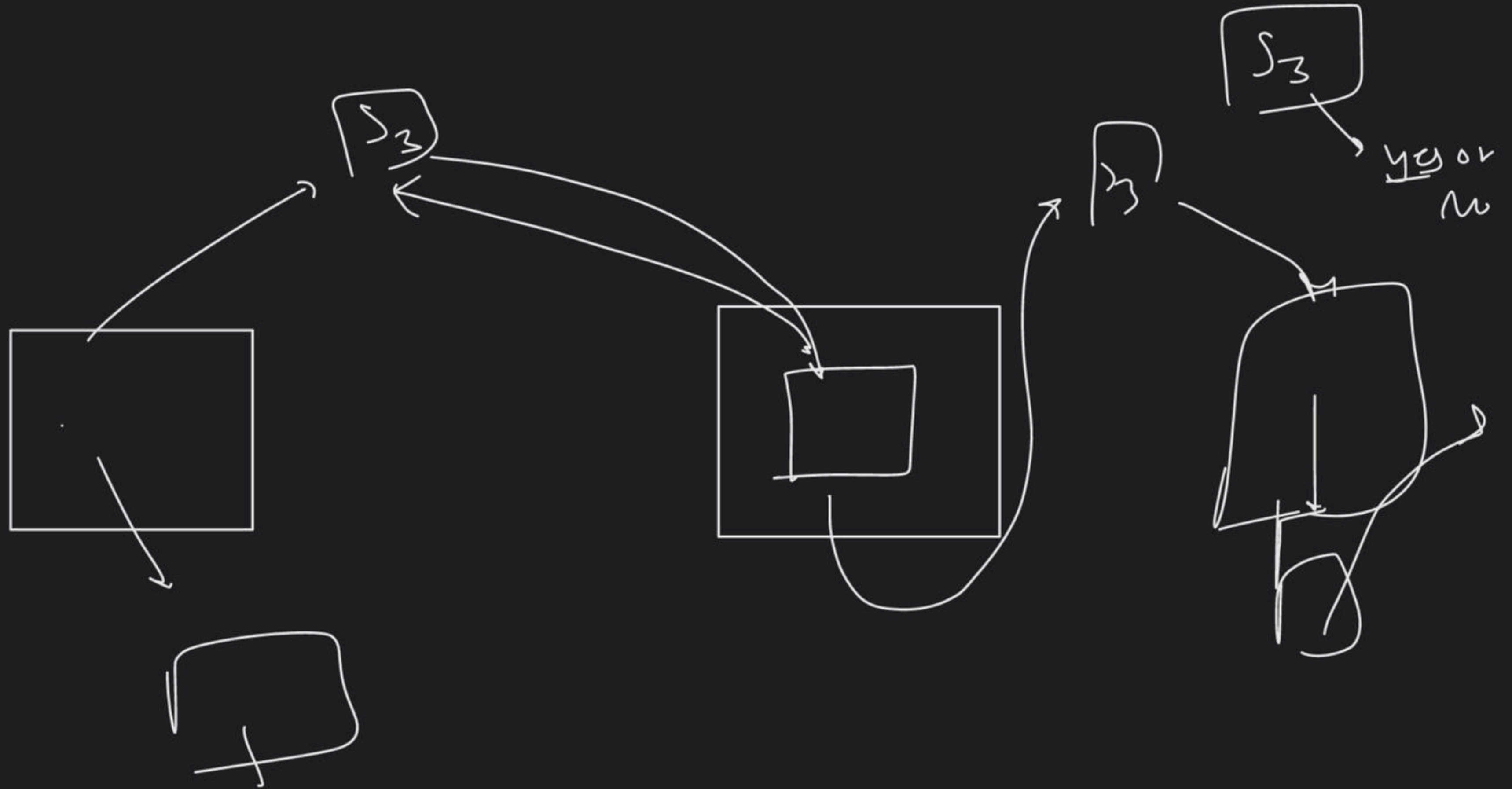
20.

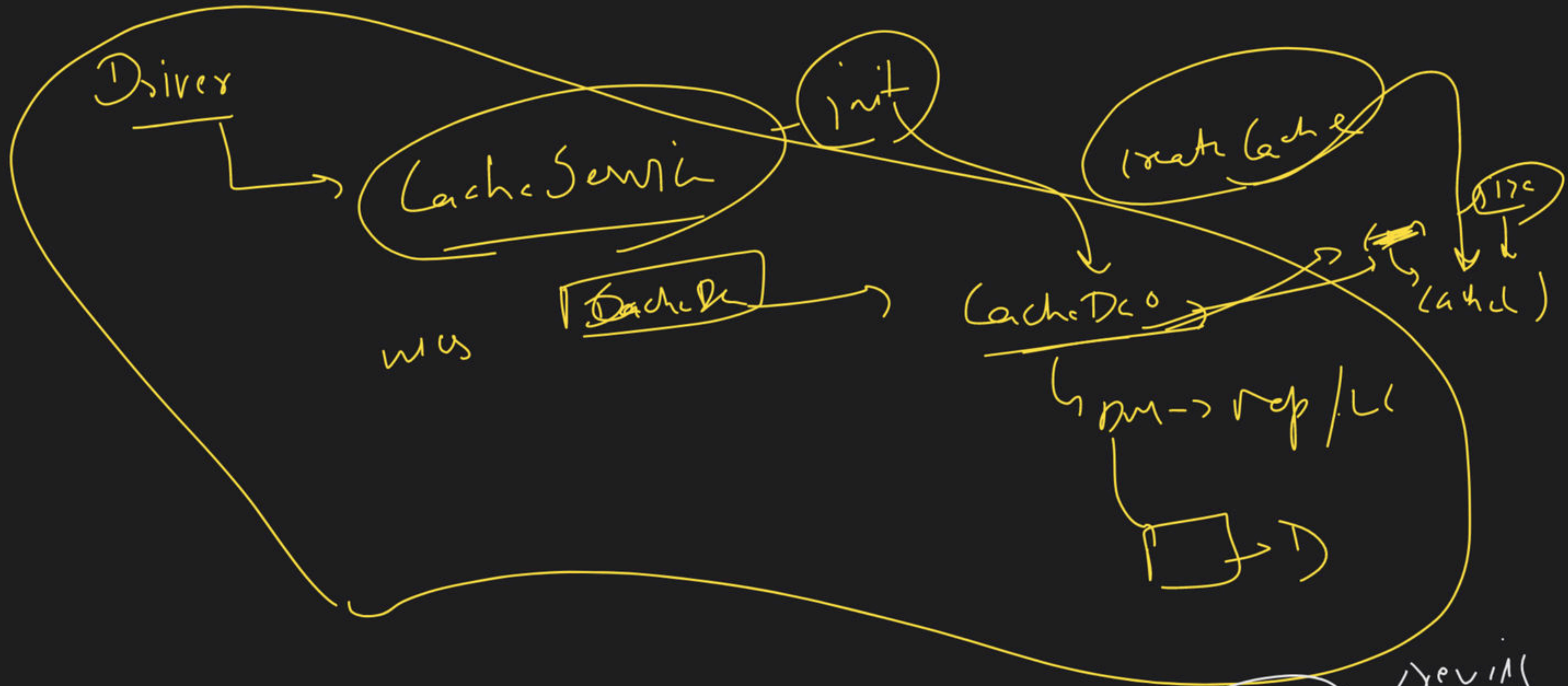
300M-1

→ painful → refactor / migration



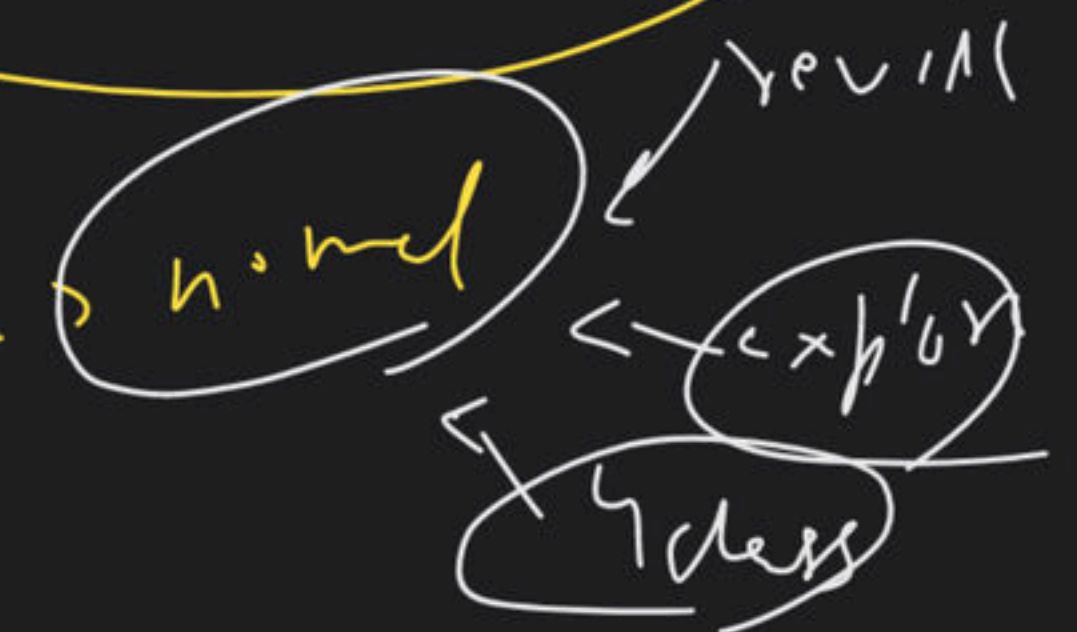






init →

gg. / -> tough
confirms



Repeat / Code

Driver

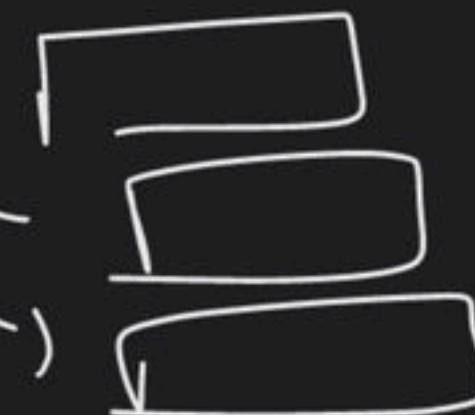
Ctrl + Click

CacheSize

Ctrl + Alt

with - LRU → 
X12

CacheD<0



Optimize

→ Sns/SQs/S3

→ D40

→ enum

→ POJO

→ MVC pattern

Code well through

Bonus

Marker /

Set

1

+

2

←