

Behavioural Design Patterns

Will start
at 9:10 PM

A gender

① Strategy DP

② Observer DP

③ Command DP

Case Study

BMS | PL

121 votes

18 Students

11 Faet



Ambala

It is fine

MWF → Namam

Behavioural Design Patterns

Agenda

- ~~1~~
- ① Strategy DP
 - ② Observer DP
 - ③ Command DP → BookMy Show

Behavioral

Creational

→ How will you create an object of a class

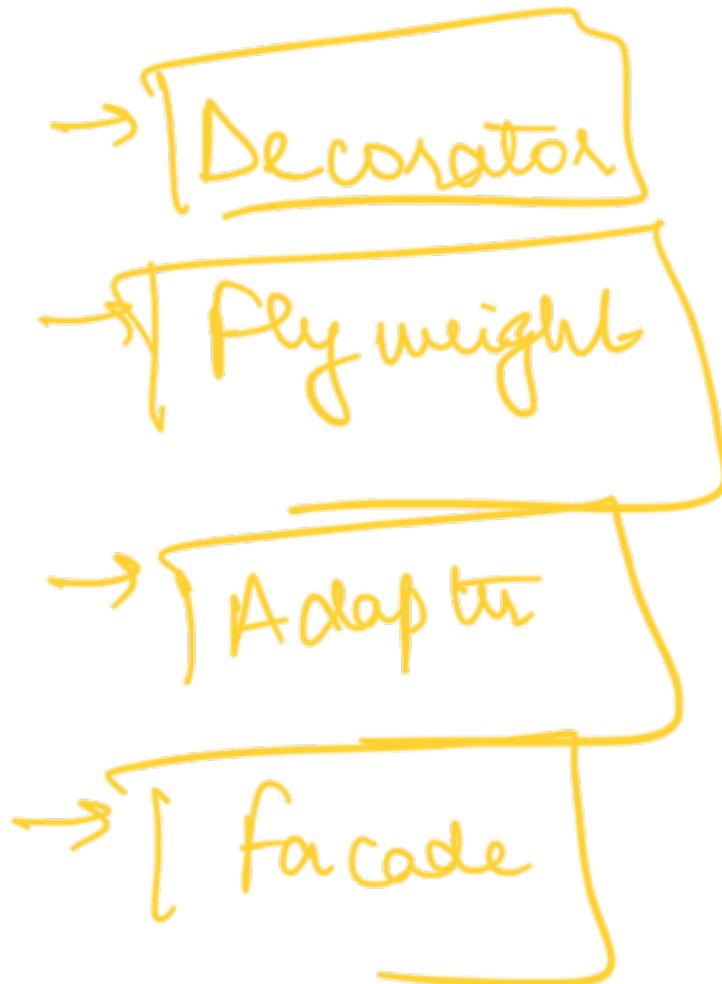
→ Singleton

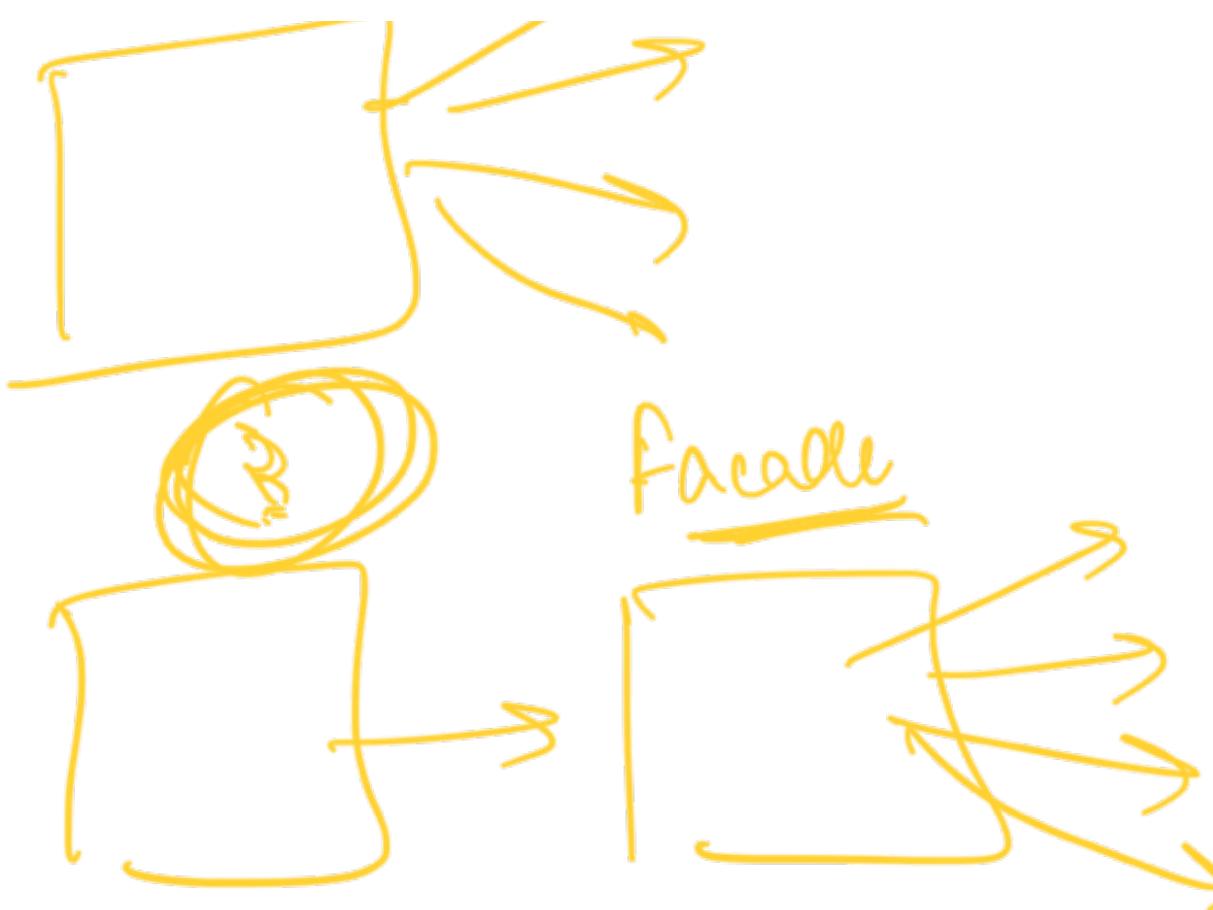
→ Builder



Structural DP

How will code be structured





Behavioural DP =

- ① Often there are some properties / requirements
with the behaviours of my system
any method

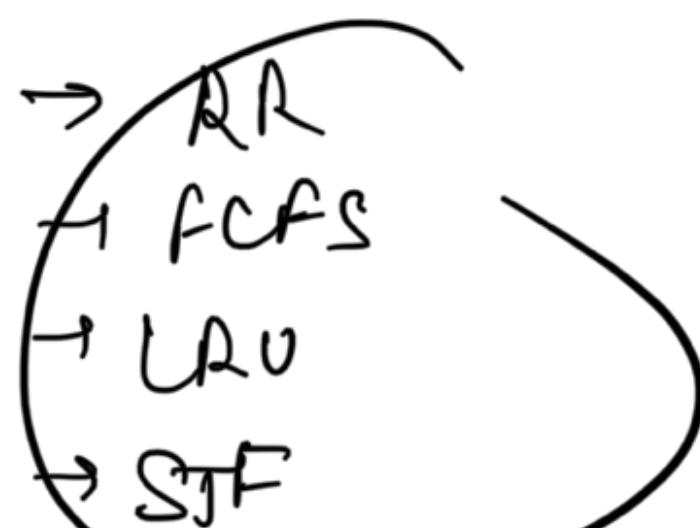
any action

- ① There are different variants/ types of that behaviour
(Diff ways to perform that behaviour)

Sorting Array Behaviour

↳ Quick Sort Sorting AB

↳ Merge Sort _____



② ~~When something happens~~

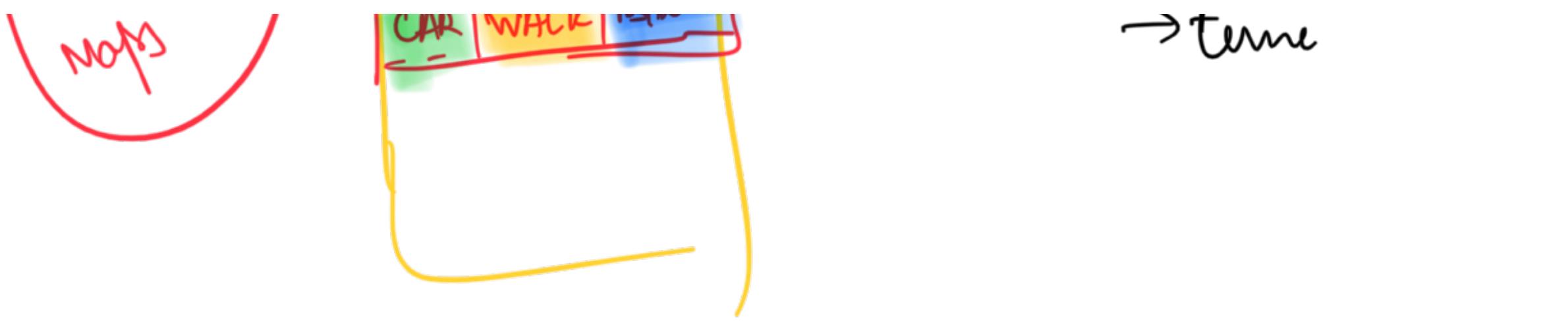
When an event occurs, multiple classes
might want to do some behaviour

STRATEGM DESIGN PATTERN



~~Hand
of
Google~~

→ Show the path from A to B



find Path (Location from,
Location to,

Mode of Transport

mode) {



if (Mode == car)



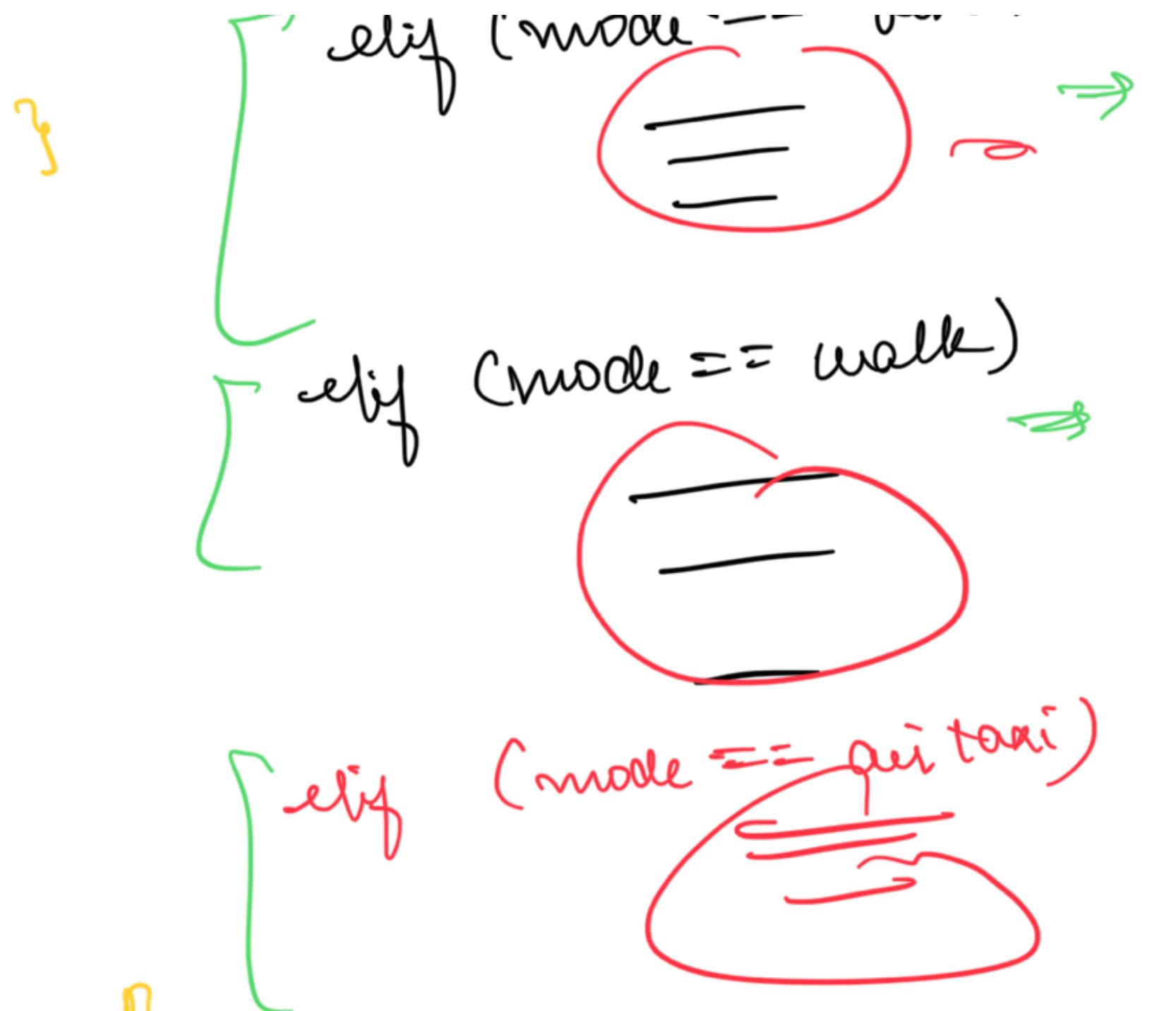
- - - - - = true)

Class

Bird

fly()

if (type == sparrow)



~~for~~ ~ 00

~~try~~ (~~type == crow~~)

}

}

O/C principle X

SAP X

Car Path Calculation {

 ⇒ Calculate Path (from, to), ~~return~~ {

}

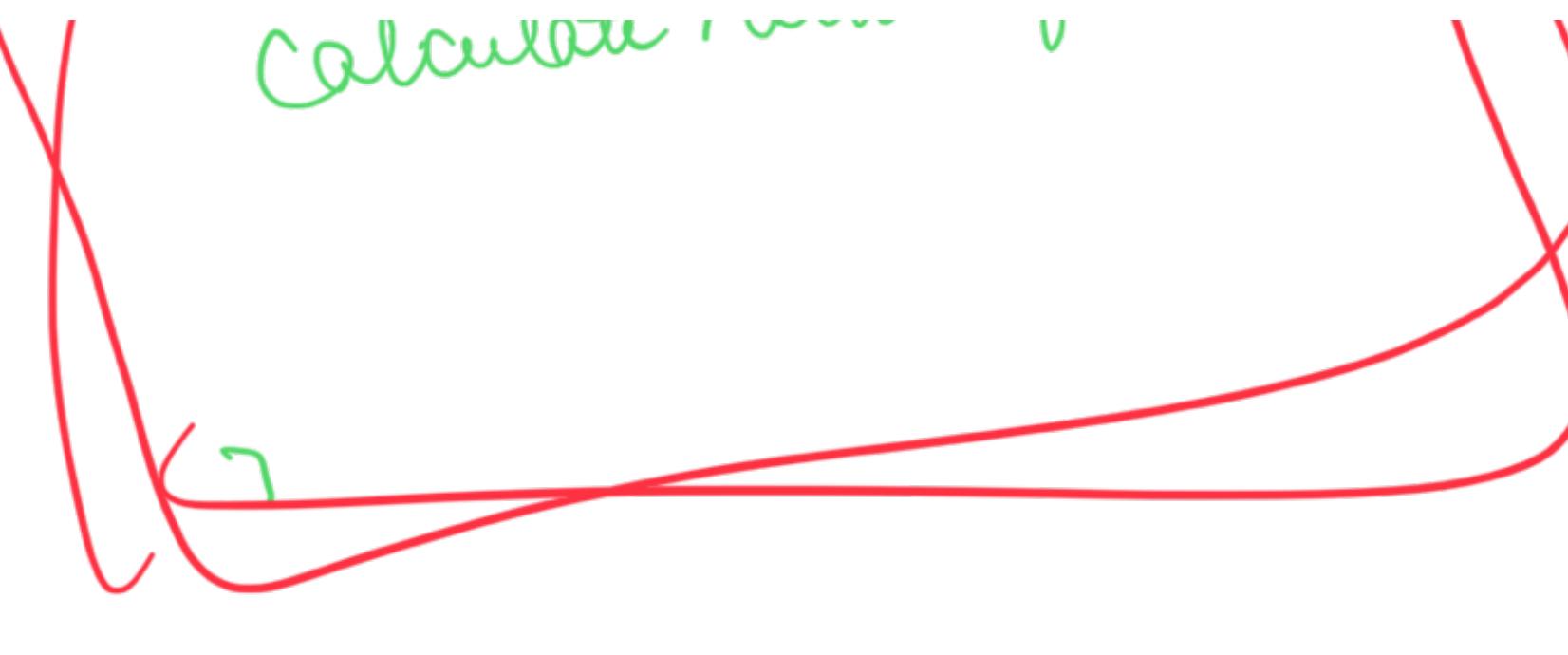
}

Bike Path Calculator {

 ... Path (from, to) {

=

Calculate Path



WalkPath Calculator {

Calculate Path (from, to) ↗

↗

}

Path Calculator

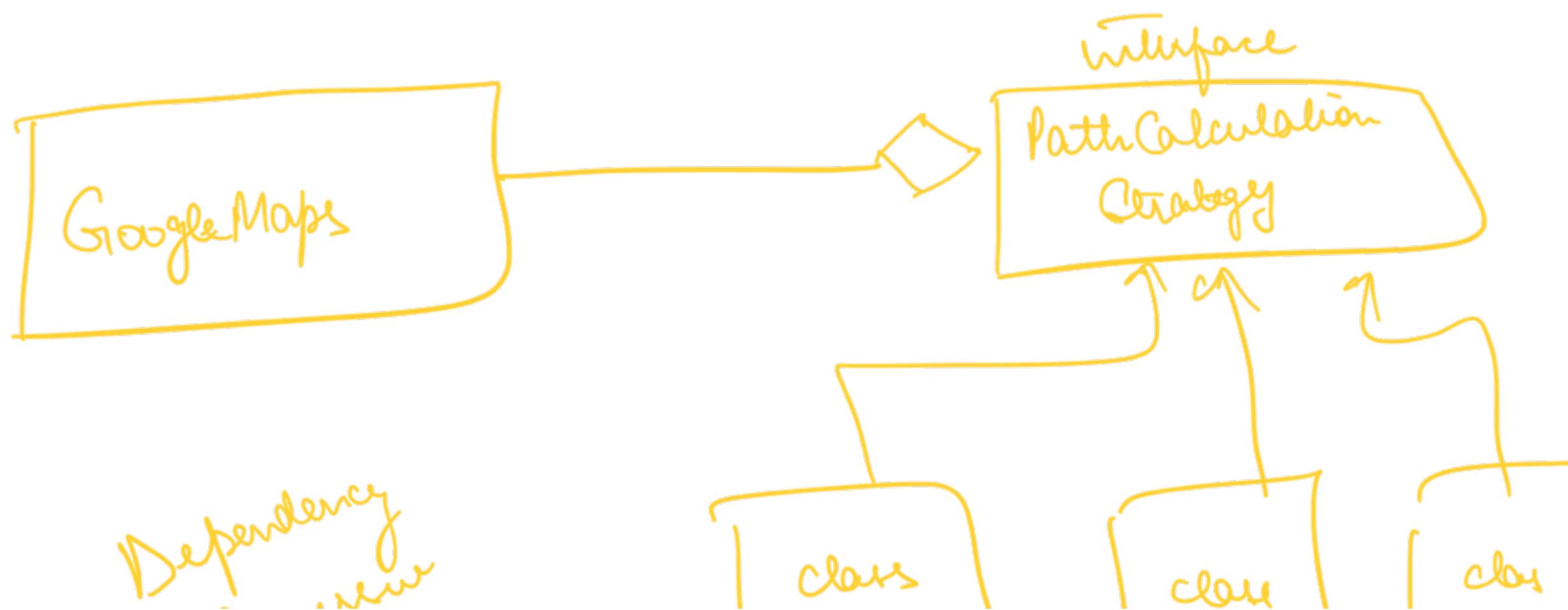
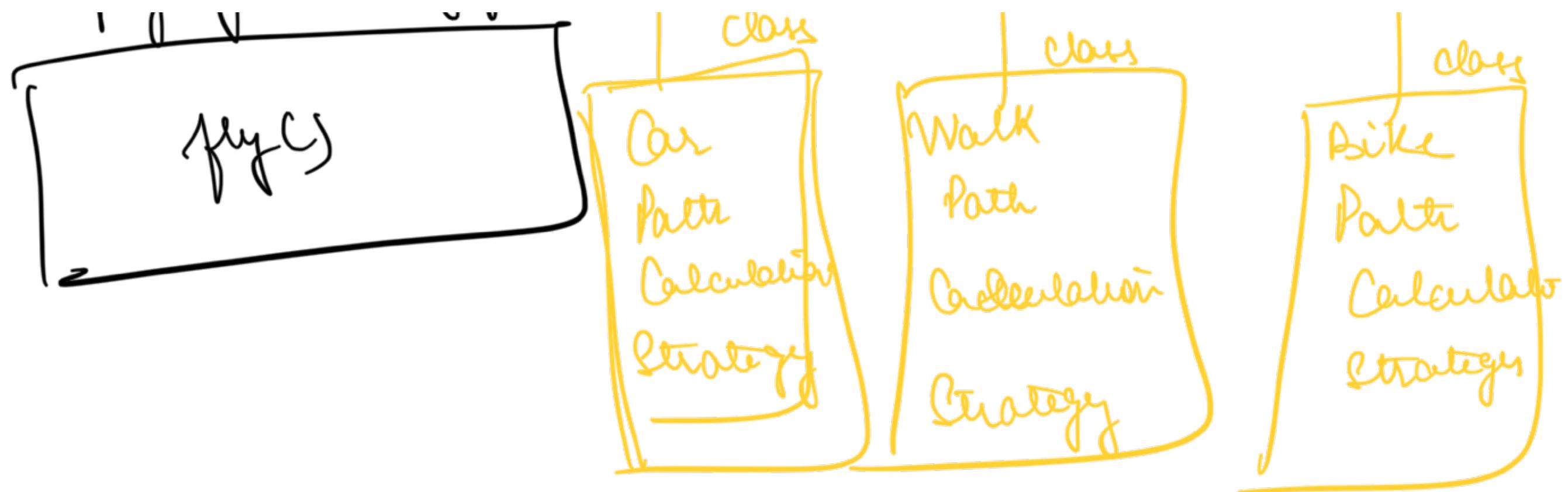
Calculate Path (from, to);

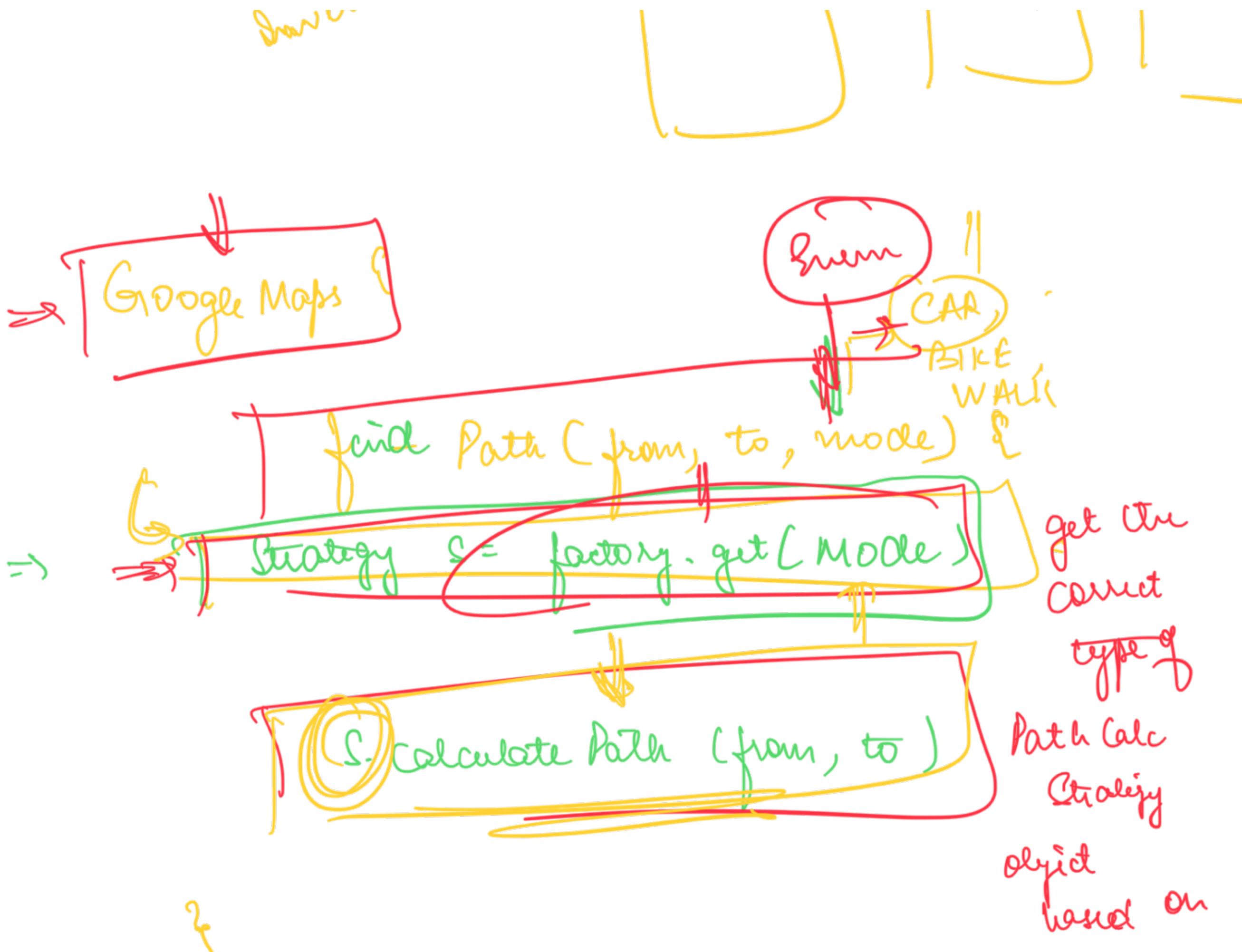
each of the inputs of path calculator is a

way to calculate path for a particular
mod

diff algo =
diff approach =
diff way =
→ diff strategy







mode

Path Calculation Strategy Factory {

get Path Calculation Strategy for Mode (Mode mode) {

if (Mode == car)

return new CarCalcStrategy()

elif (Mode == bike)

}

}

Strategy

Adapter

- Behav
 - A behaviour with multiple variants
 - Classes are just variants of behaviours
- Structural
 → How will my codebase talk to a 3rd party class
 → Classes wrap 3rd party classes





Assignment

RainWater Trapping Problem

area of water trapped

Solve (Left min > heights)

, Method)

$$\hookrightarrow O(N) O(1)$$

$$\hookrightarrow O(N) O(N)$$

$$\hookrightarrow O(N) O(1)$$

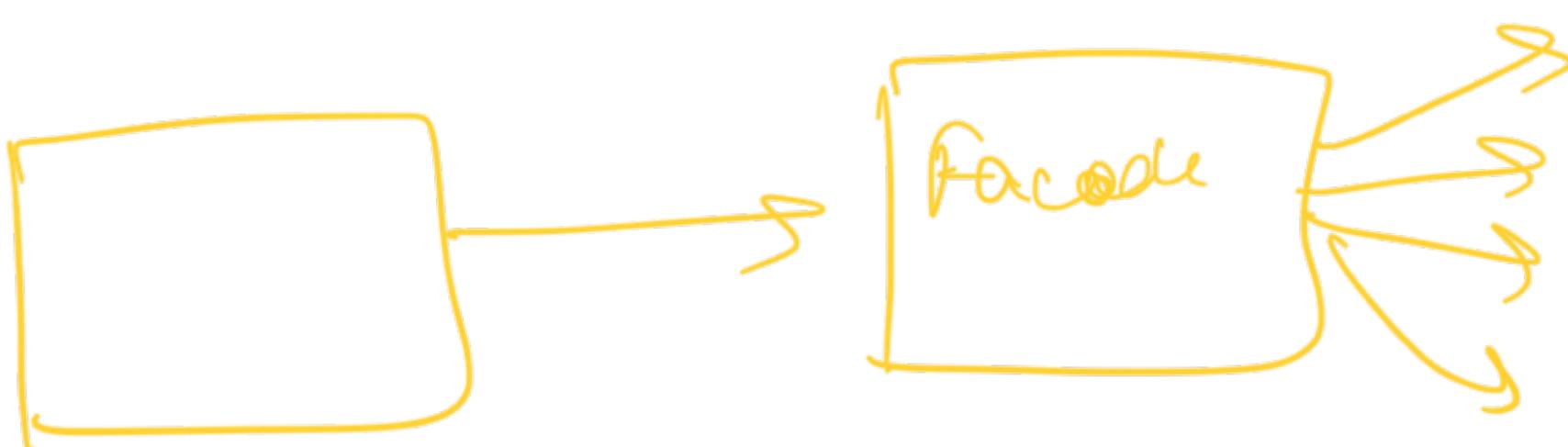
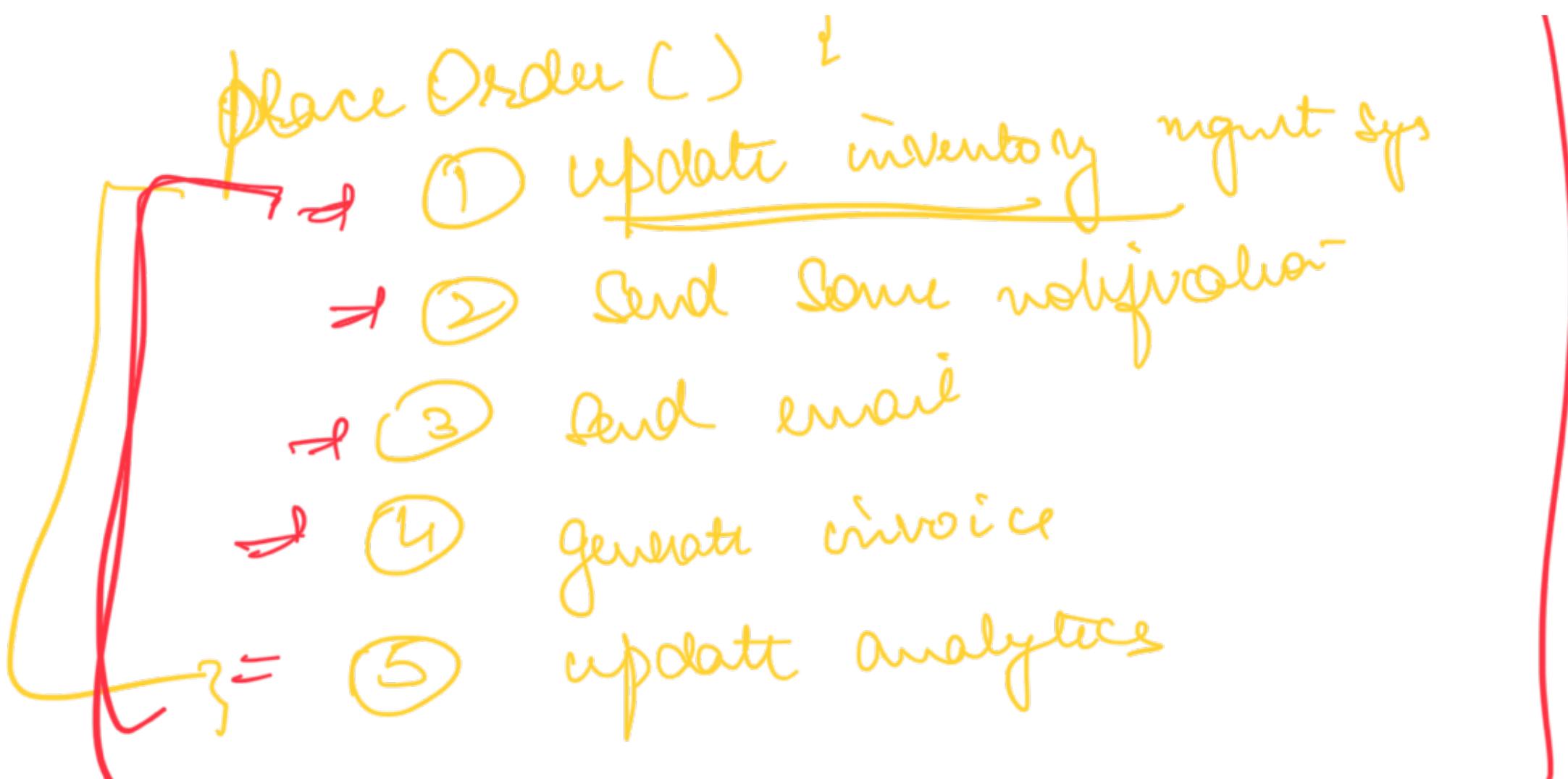
Distributed Cache

Paravys MC

= 11:05 PM

Observer Pattern

Flipkart



Variant:

- ① flipkart class itself doesn't know what all to do when an order is placed
- ② There can be more things that I might want to do at runtime

Way!

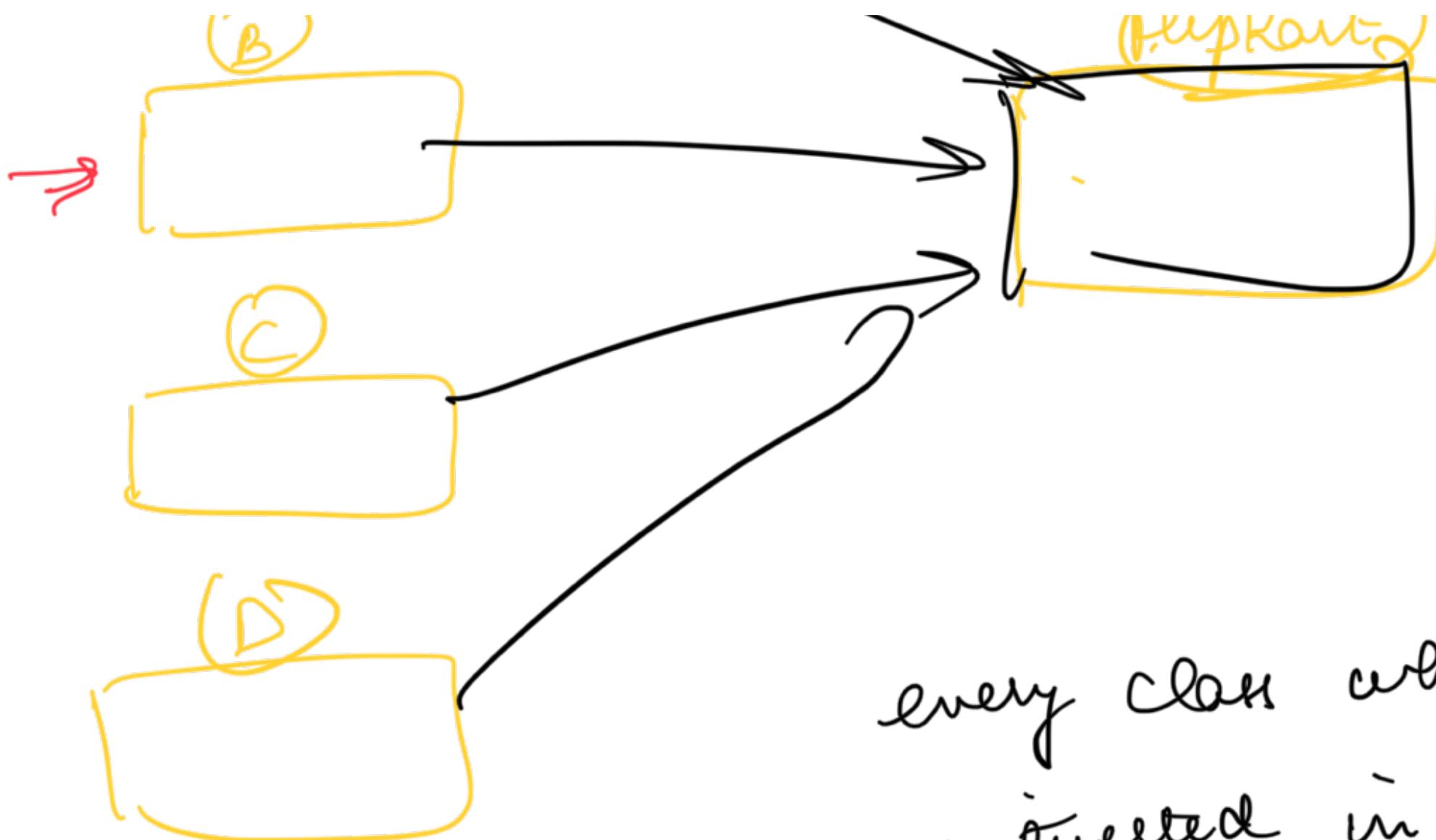


AG

fk. check New Orders()

See Order

Point 1



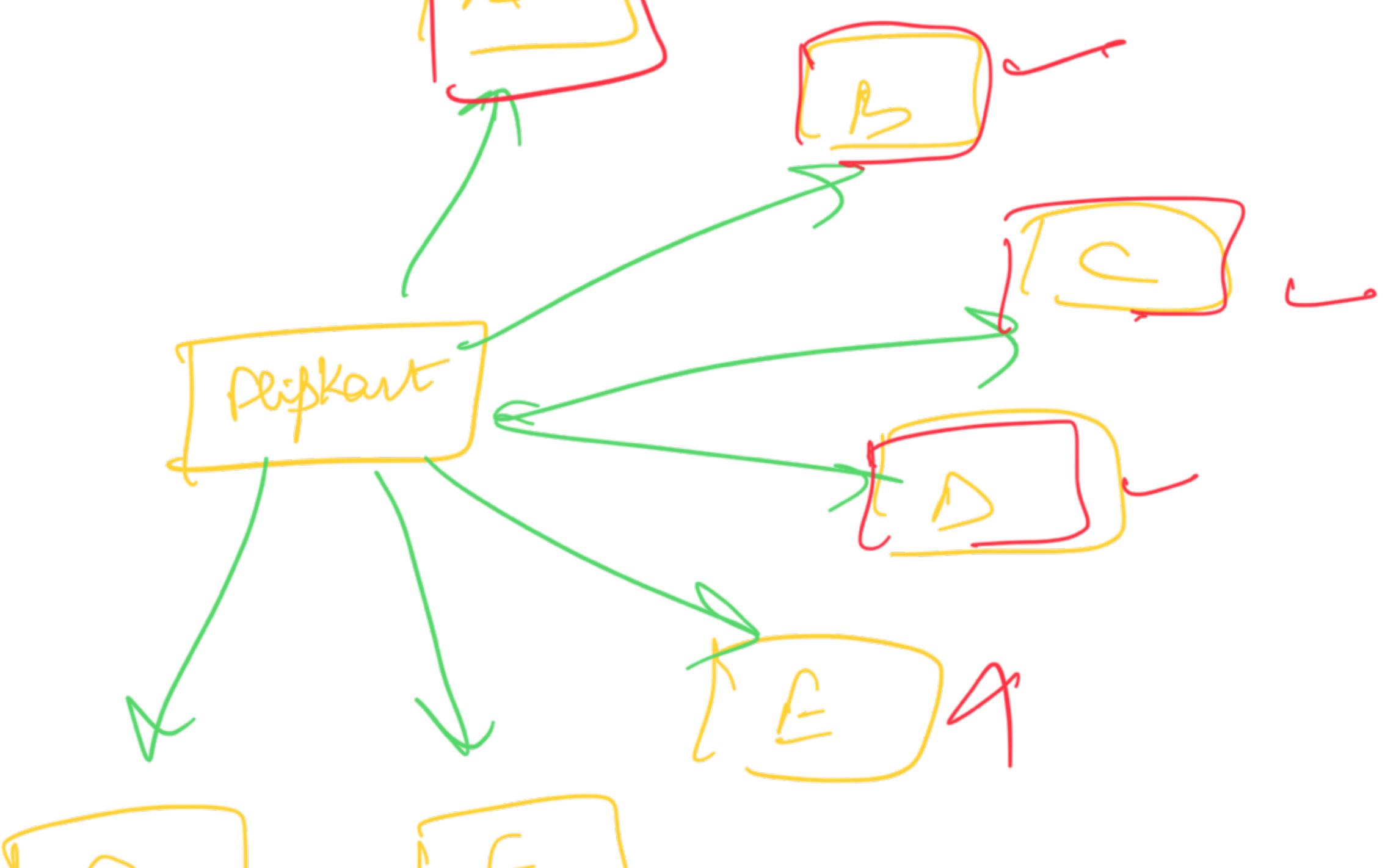
every class who is
interested in an event

asks you repeatedly if
that event has occurred



→ CPU expensive

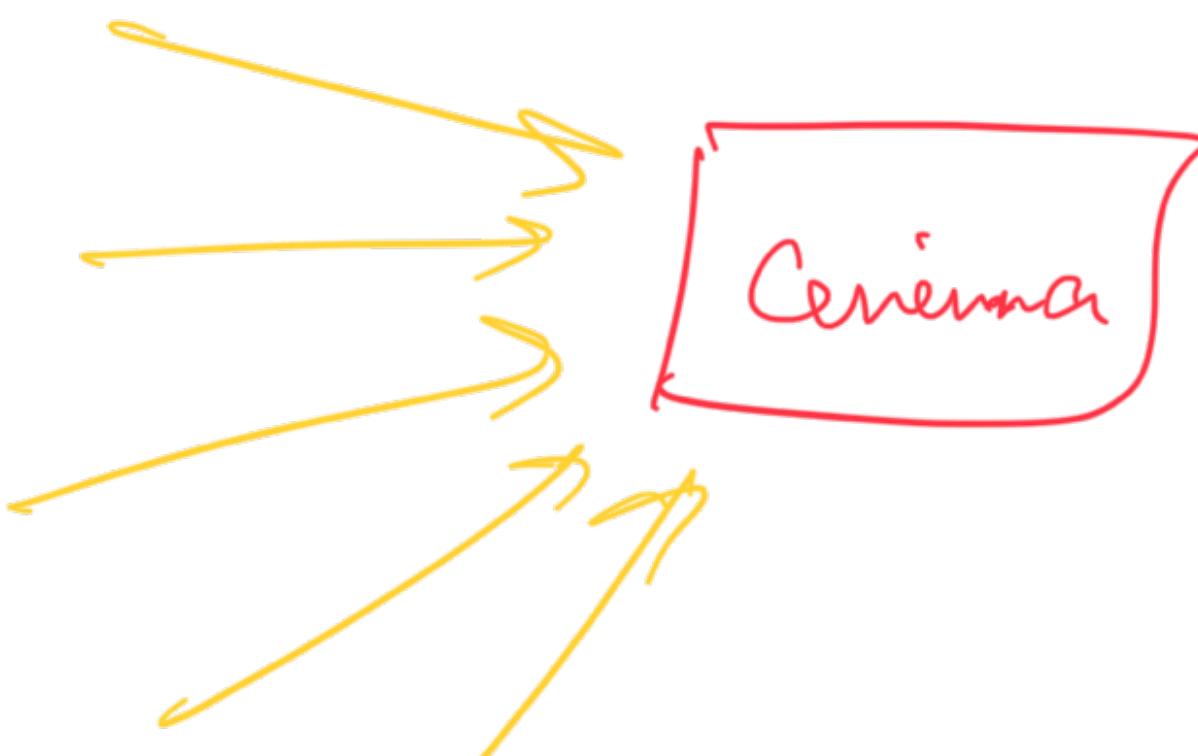
way 2

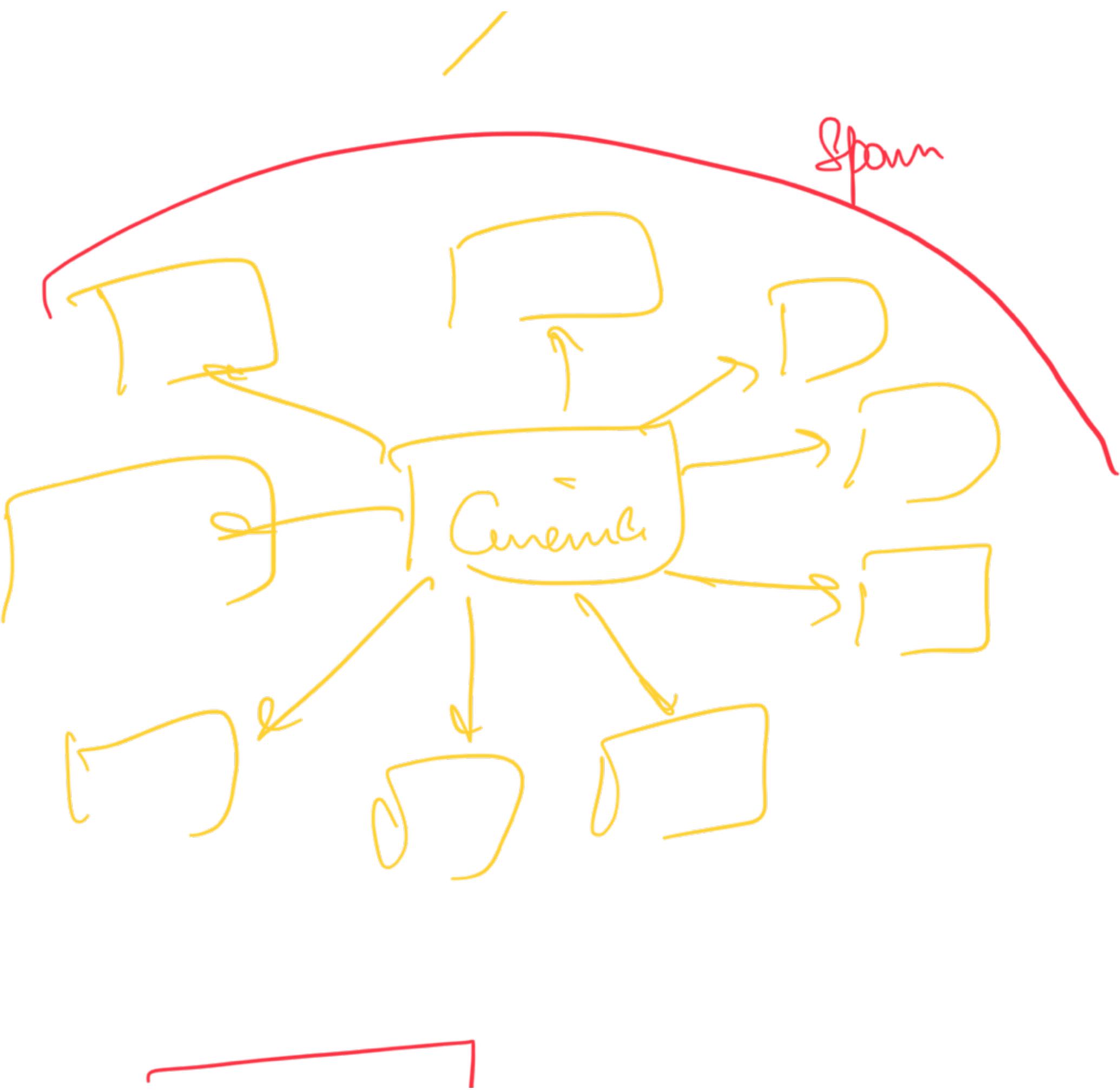


$\begin{bmatrix} \text{L} \\ \text{U} \end{bmatrix}$

$\begin{bmatrix} \text{L} & \text{U} \end{bmatrix} \alpha$

Let us say you are interested in booking
tickets of a movie





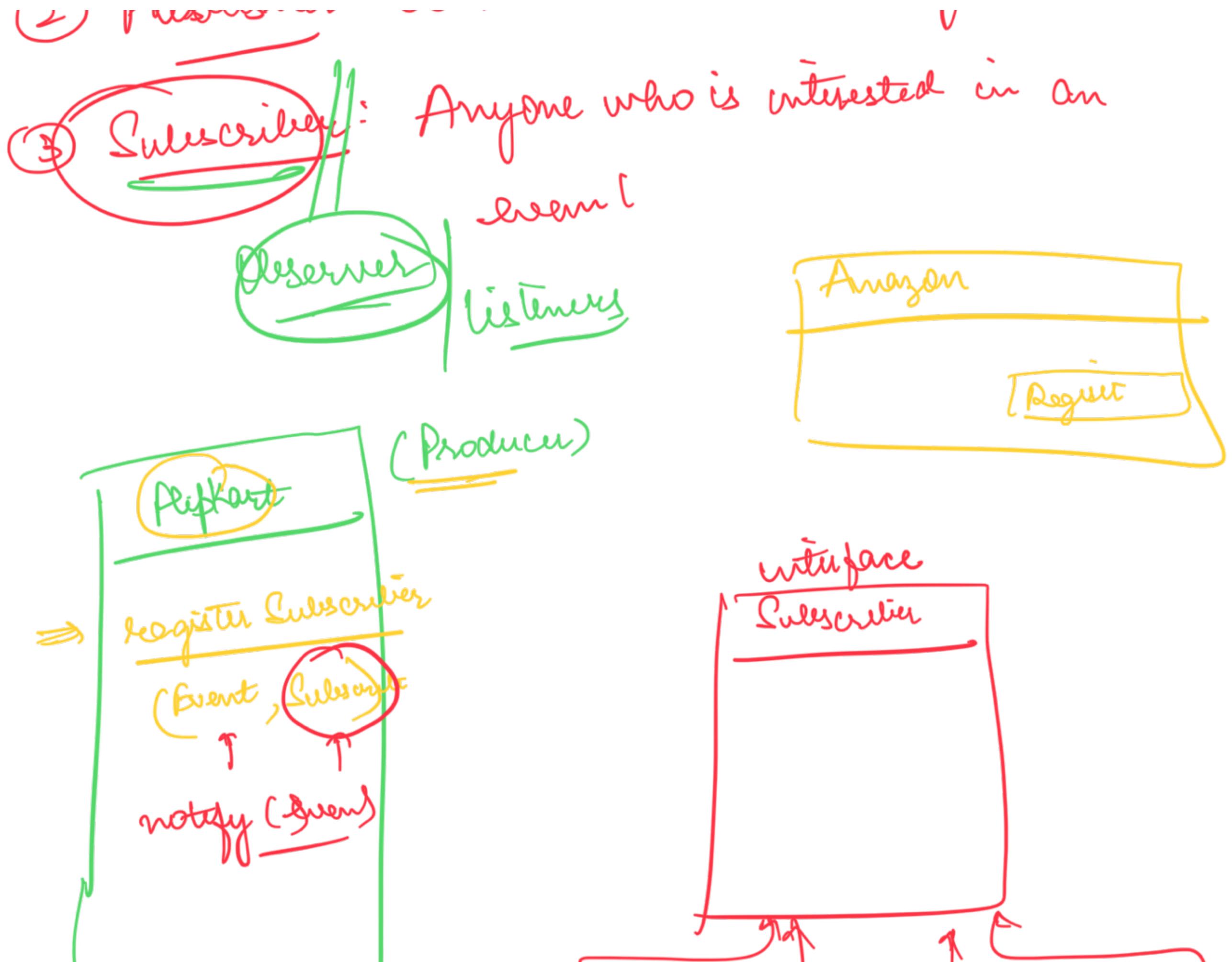


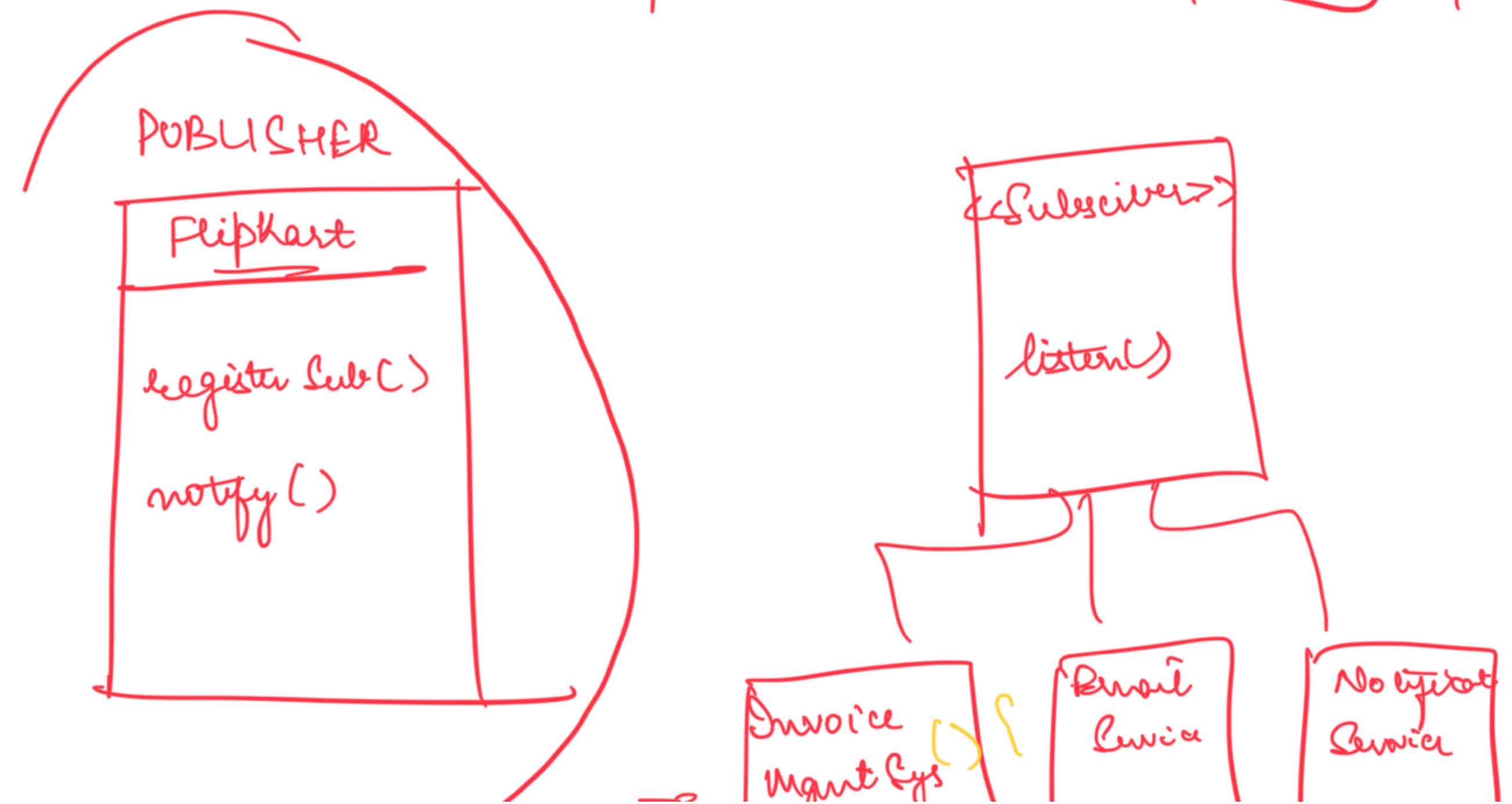
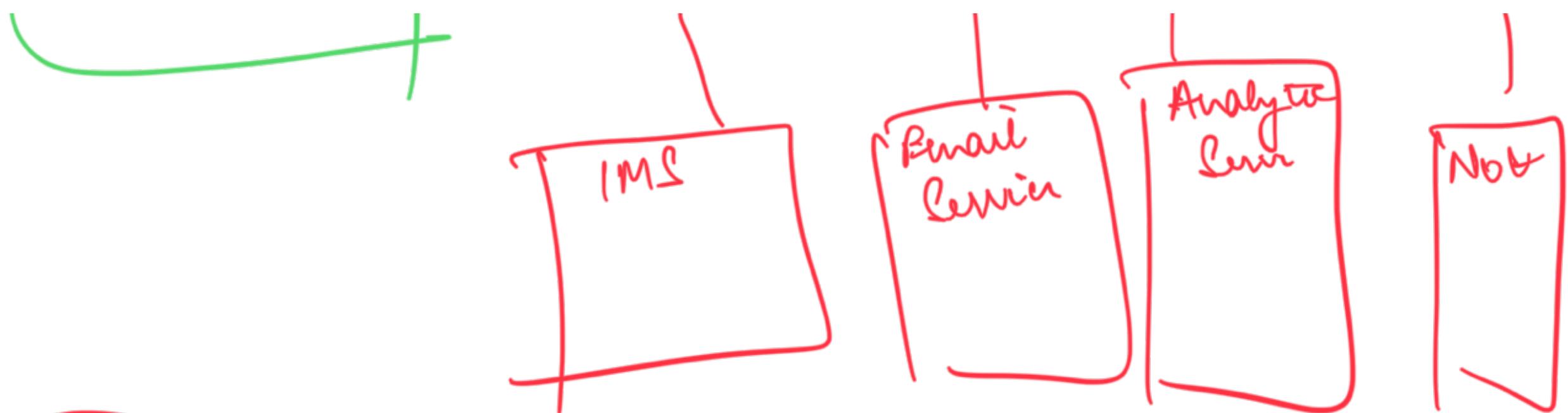
① Class that has an interesting event that
other classes want to know about

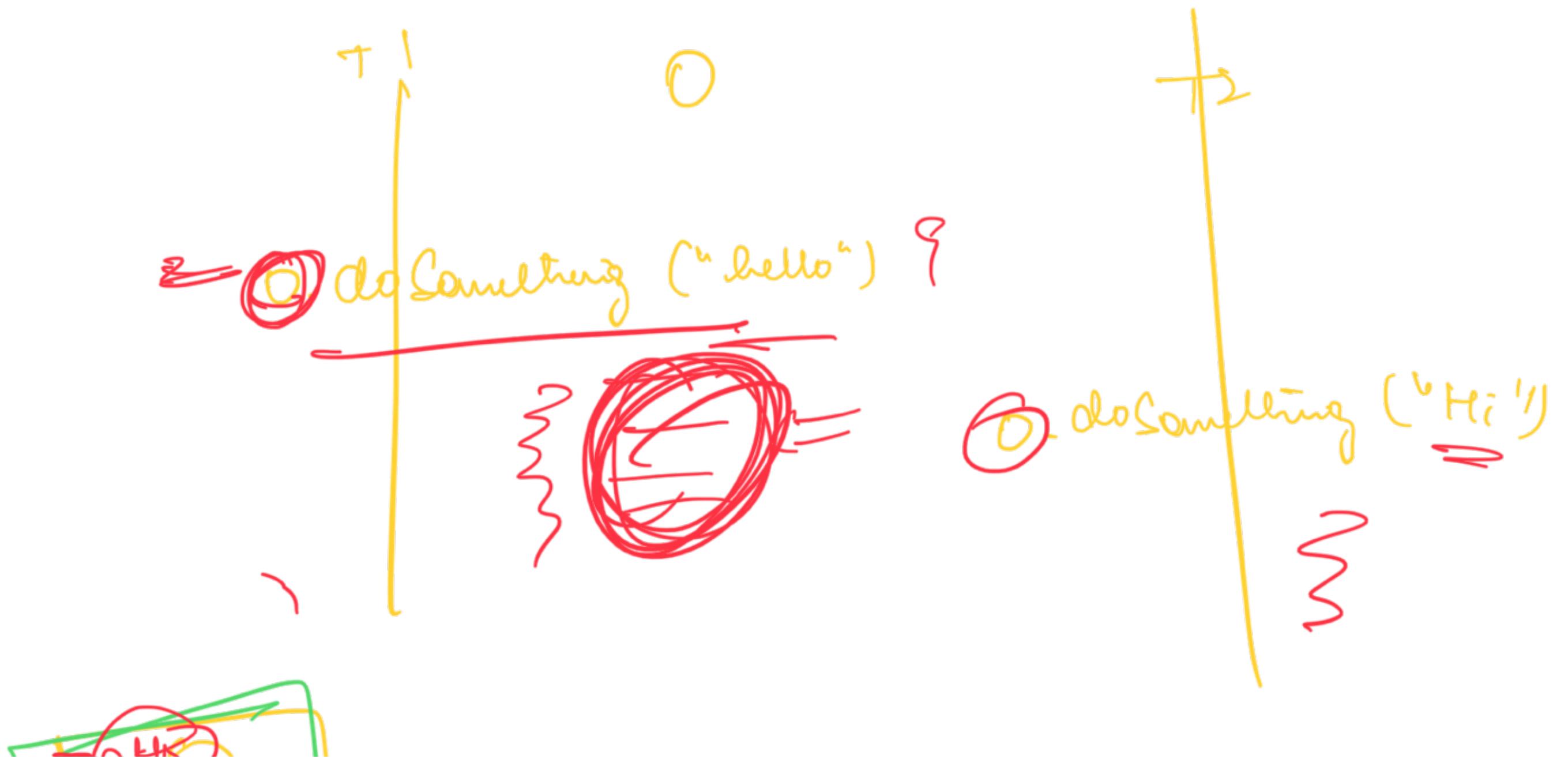
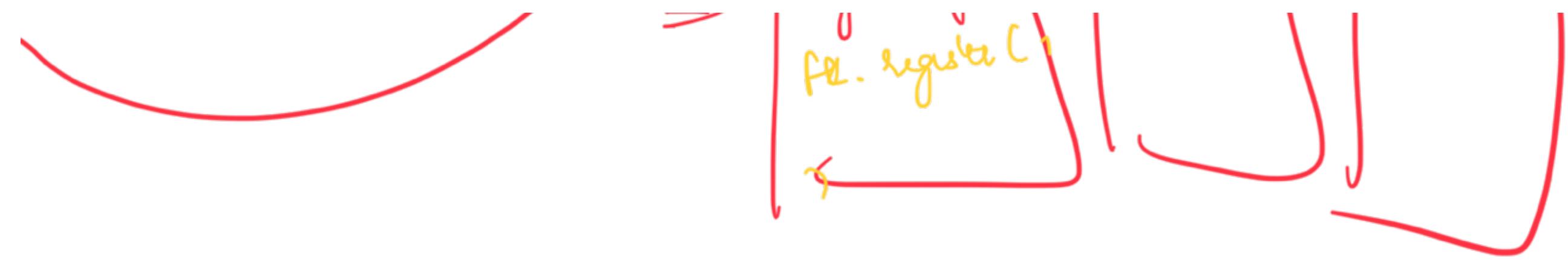
Observer DP
flipkart

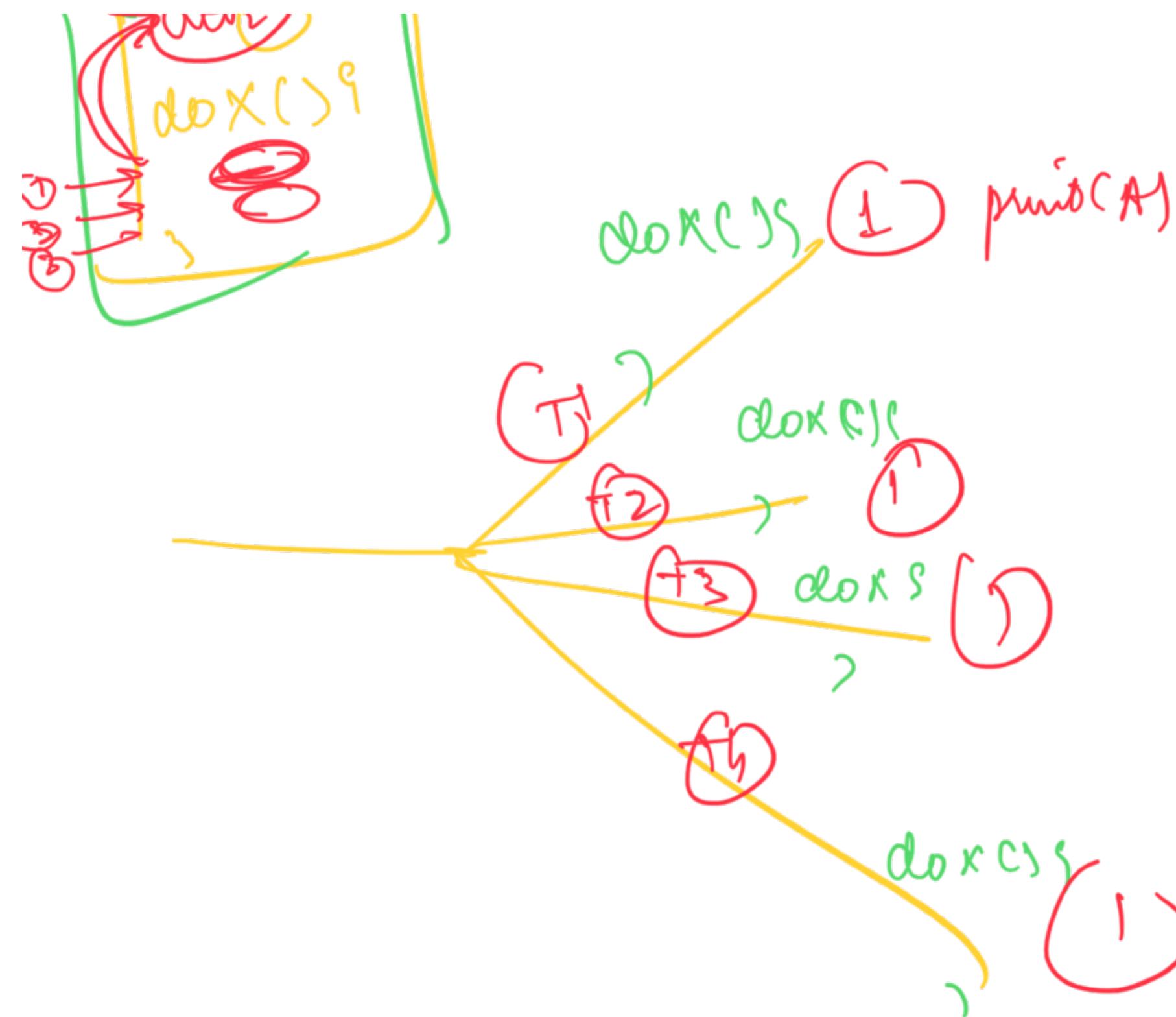
Order placed

→ D.O.Dieker: Class that has interesting event









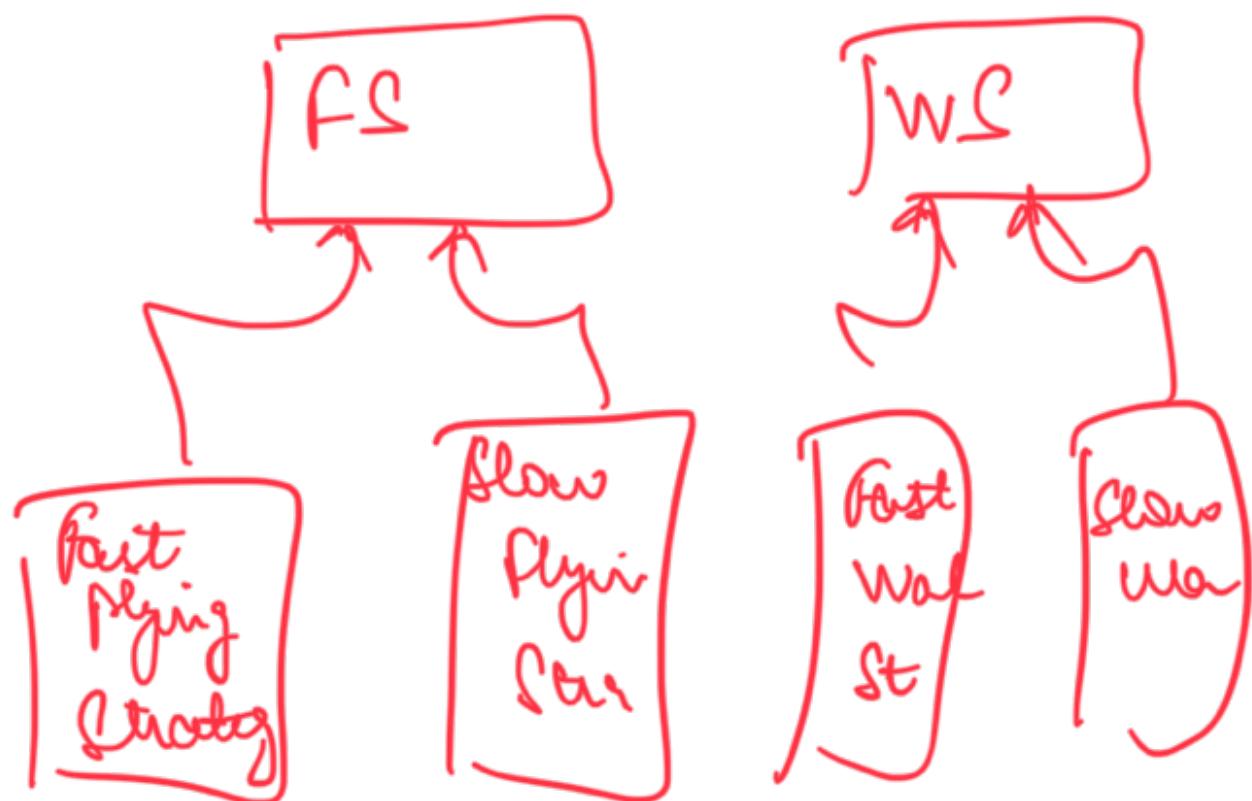
alte Bild 9

Flying Strategy fs;
Walk Strategy fw;

Distributed Order

Pigeon 9

FC fe = new fast FSC



7