Agenda.

⇒ Structural design patterns.

- ADAPTER
- FACADE.

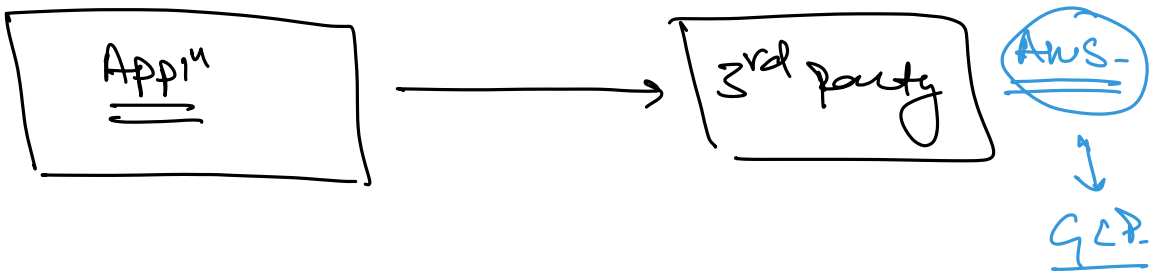
Structural Design Patterns.

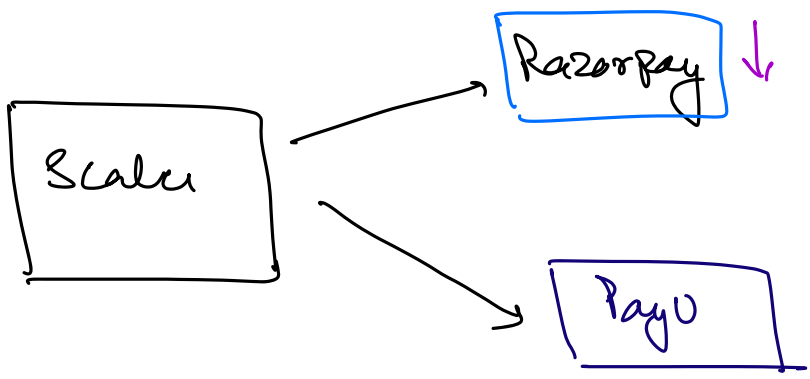
- Structure of our codebase.
- What all the classes/interfaces should be there.

ADAPTER PATTERN.

- Power Adapter.
- Converter.

⇒

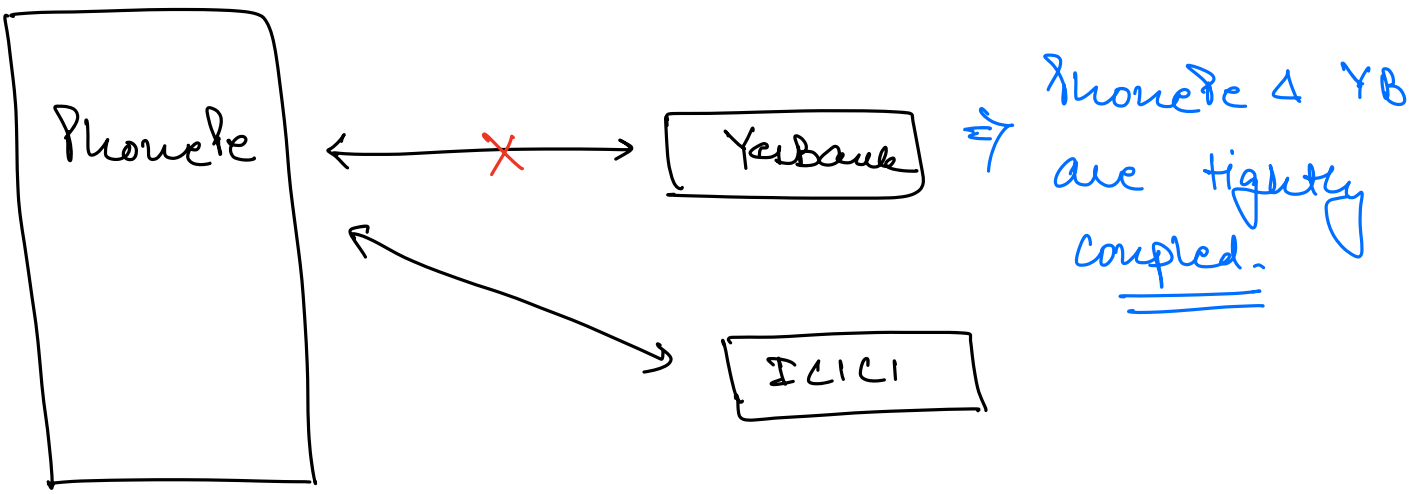




Problem Statement.

→ We might have to move from one 3rd party to another 3rd party.

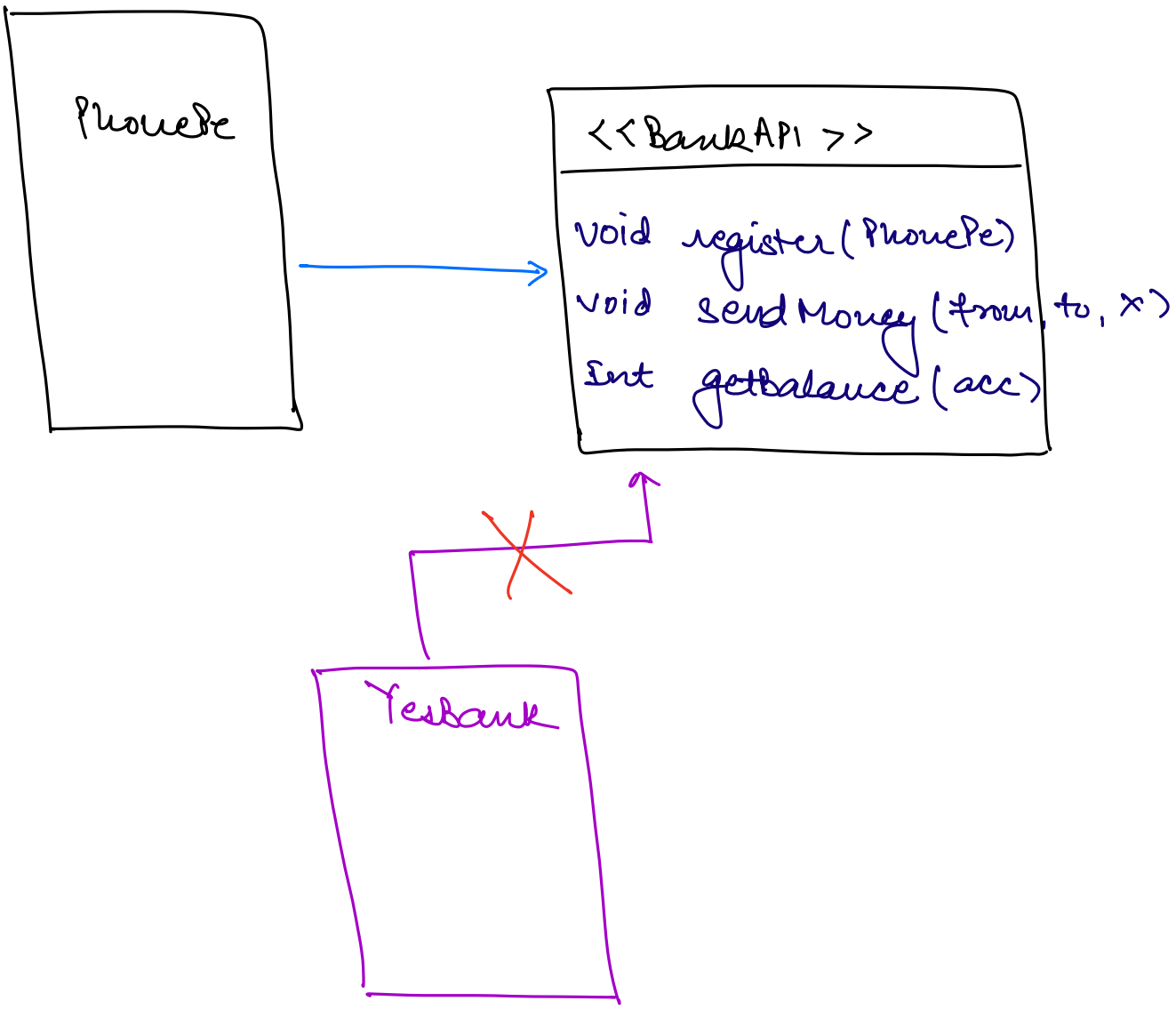
→ We might also want to integerate multiple 3rd party service providers.

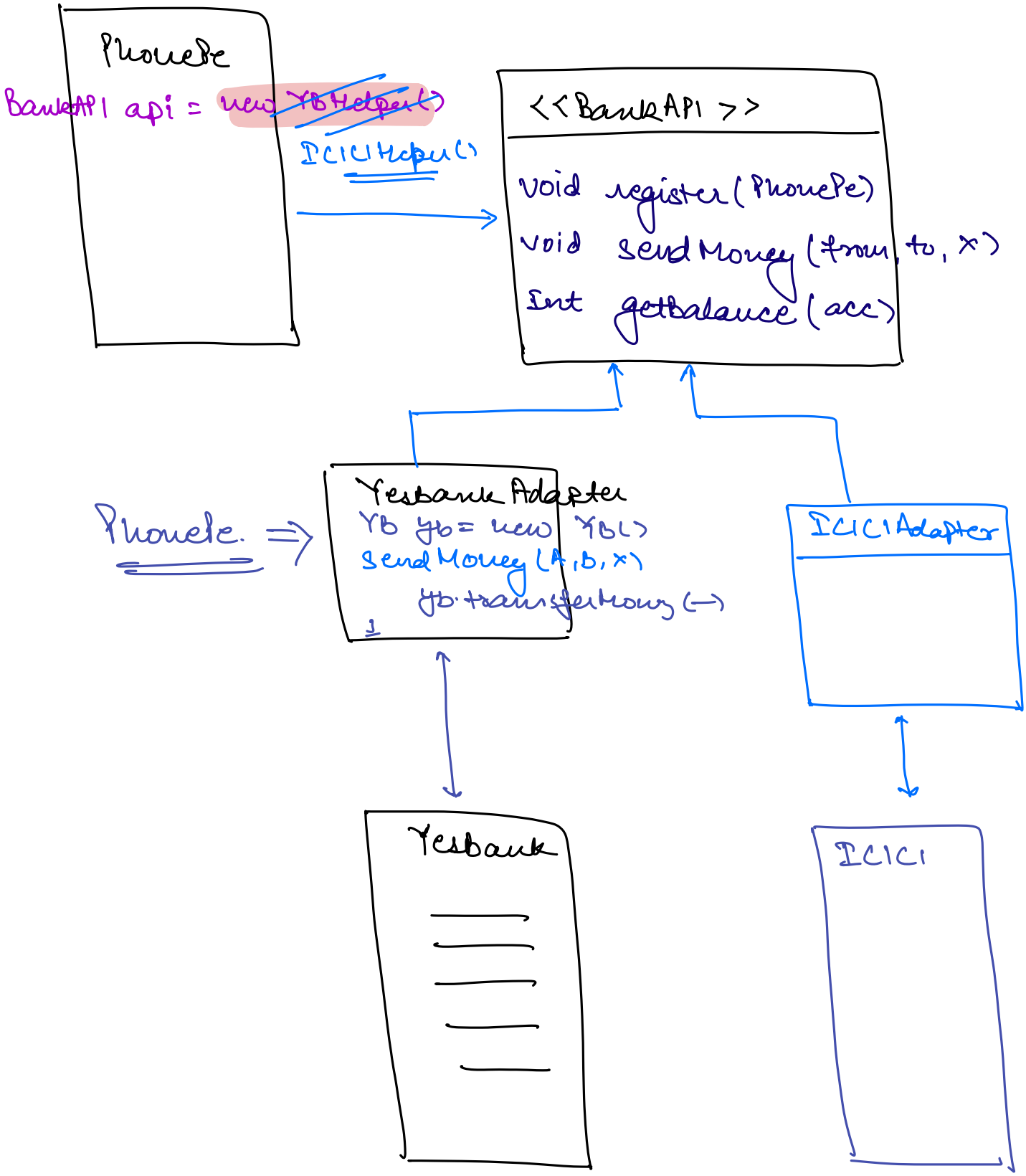


⇒ Adapter pattern ensures that our codebase remains maintainable whenever we are talking to 3rd party APIs.

⇒ If our codebase is directly talking to 3rd party libraries, it involves a lot of tight coupling b/w our code & 3rd party system & it makes our code less maintainable.

⇒ Whenever we are connecting to a 3rd party API, we should use an interface in b/w.

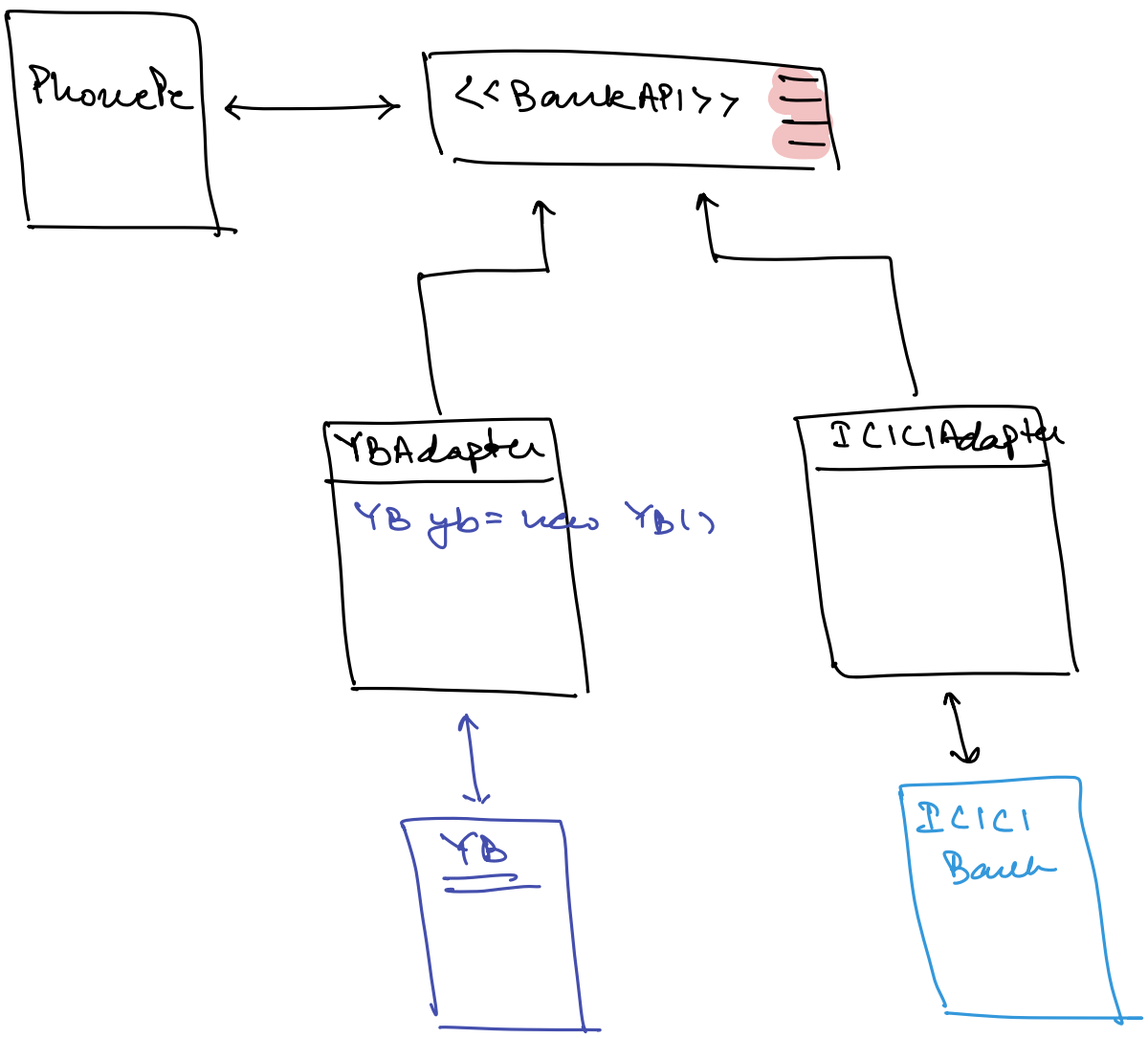


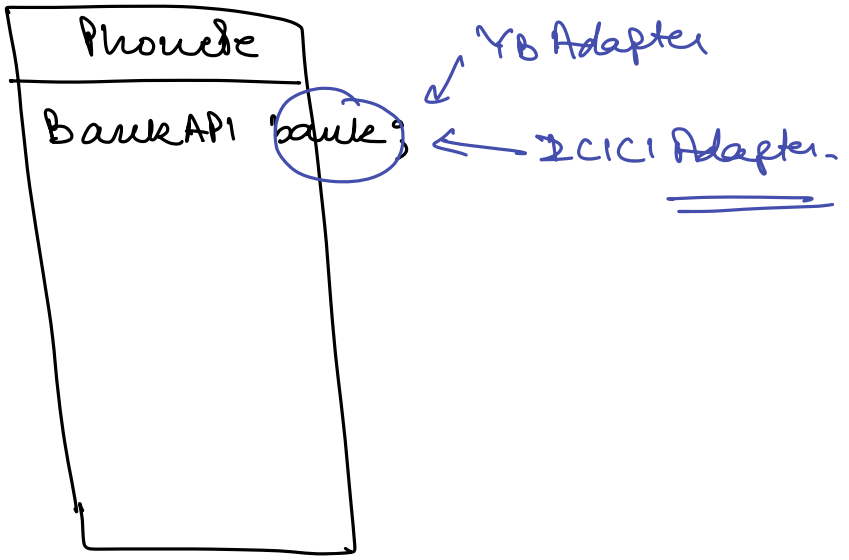

PhonePe
BankAPI api = new YBHelper()
ICICIHelper()
<<BankAPI>>
void register(PhonePe)
void sendMoney(from, to, x)
Int getbalance(acc)
PhonePe. ⇒
YesbankAdapter
YB yb = new YB()
SendMoney(A, B, x)
yb.transferMoney(→)
ICICIAdapter
Yesbank
ICICI

⇒ How to use Adapter.

① Whenever we are connecting to a 3rd party API, Create an interface with the methods that we need from 3rd party.

② As 3rd party will not implement this interface for us, we should create wrapper/helper/adapter that implements this interface using the 3rd party APIs.

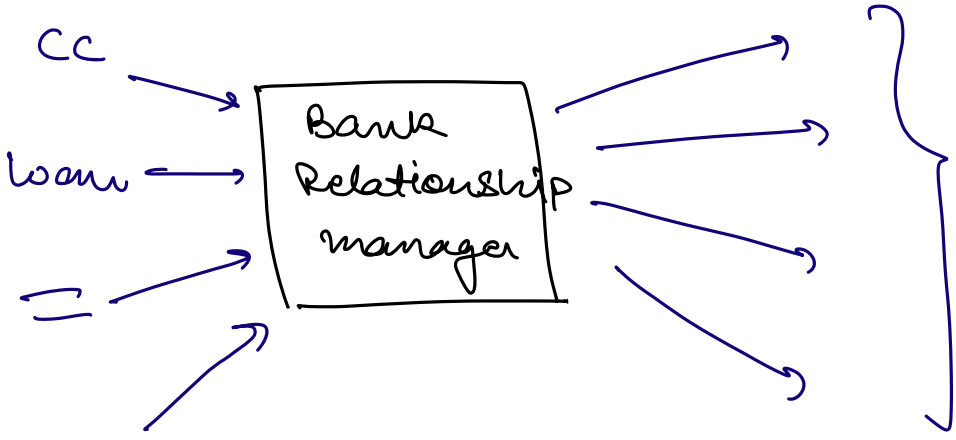




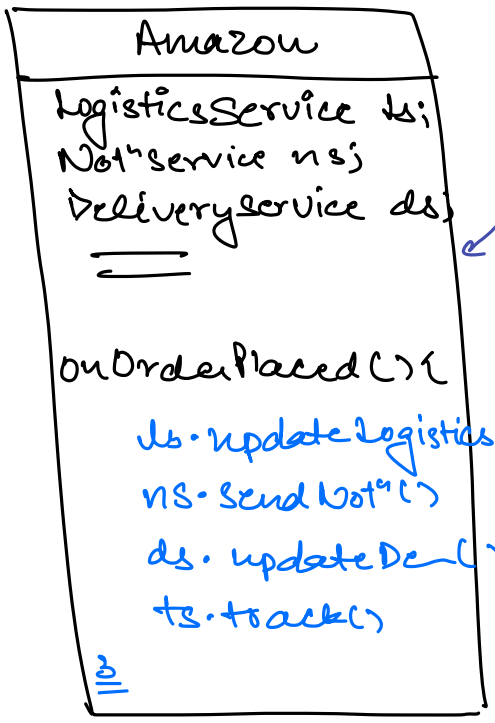
FACADE

↳ Exterior | Boundary of a Building

⇒ Provides a cleaner | simplified view of complex System.



⇒



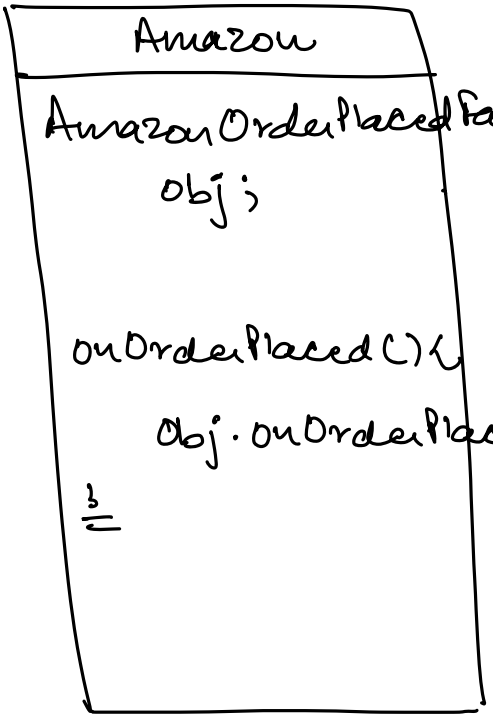
All the steps that should get executed if an order is placed.

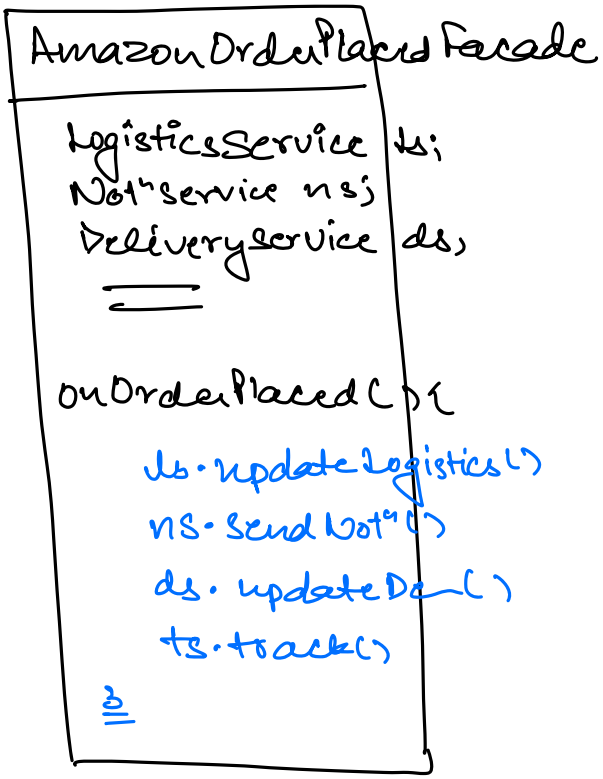
→ Update Seller
→ Send Notifications.
→ Update LogisticsService
→ deliveryService
→ TrackingService

⇒ Currently all the work of what will happen if an order is placed is being done in Amazon class.

⇒ This might make Amazon class to bulky

FACADE : Whenever you find a method / class that is doing too much of work then instead of doing that work within the same class/interface, Move that into some helper/facade class.





Conclusion.

ADAPTER ⇒ 3rd party

facade ⇒ provides simplified view of complex things.