Intro to springboot

Revisit MVC

How Req get served in SB

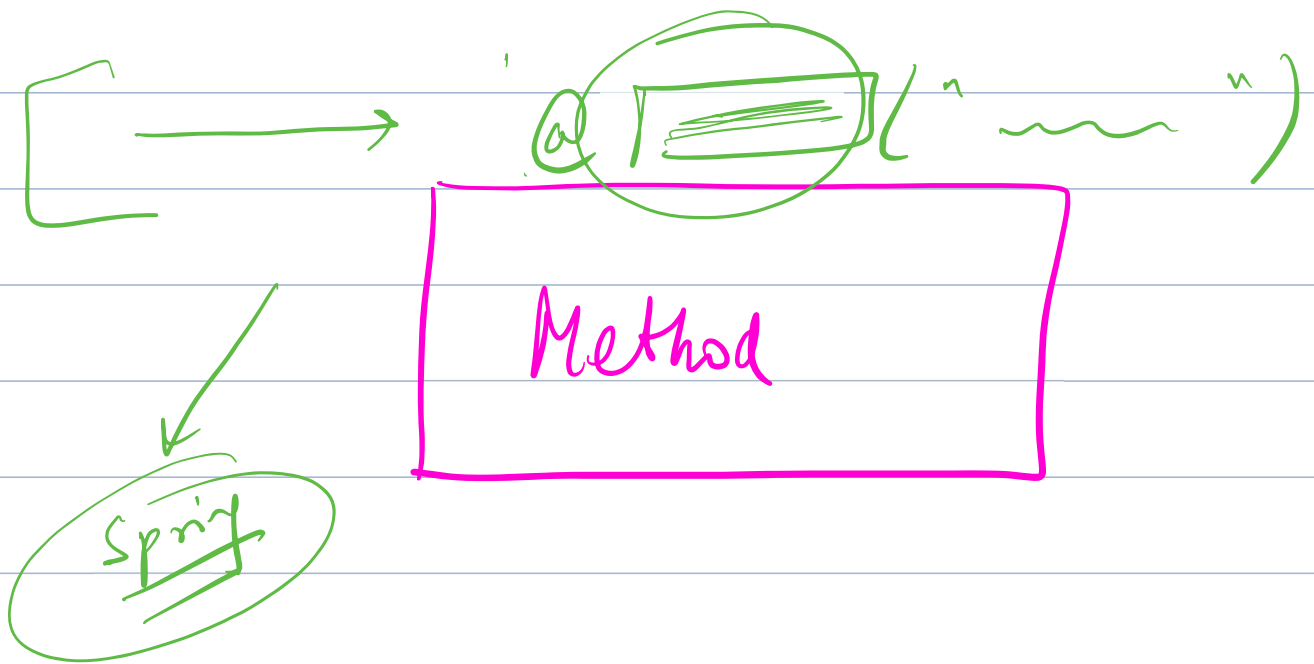Practical → Service Layer Impl

---

binary   0, 1

(HL Languages) → C++, Python, Java

↓

Helped us in

writing code in english

level text

Compilers —— Computer / Machine

↓

HLL → 0/1
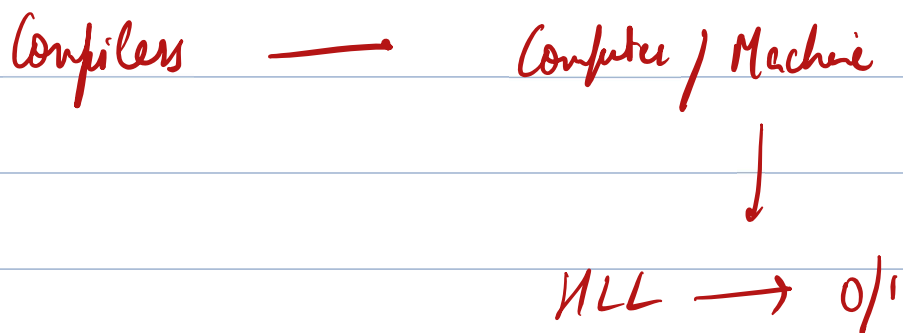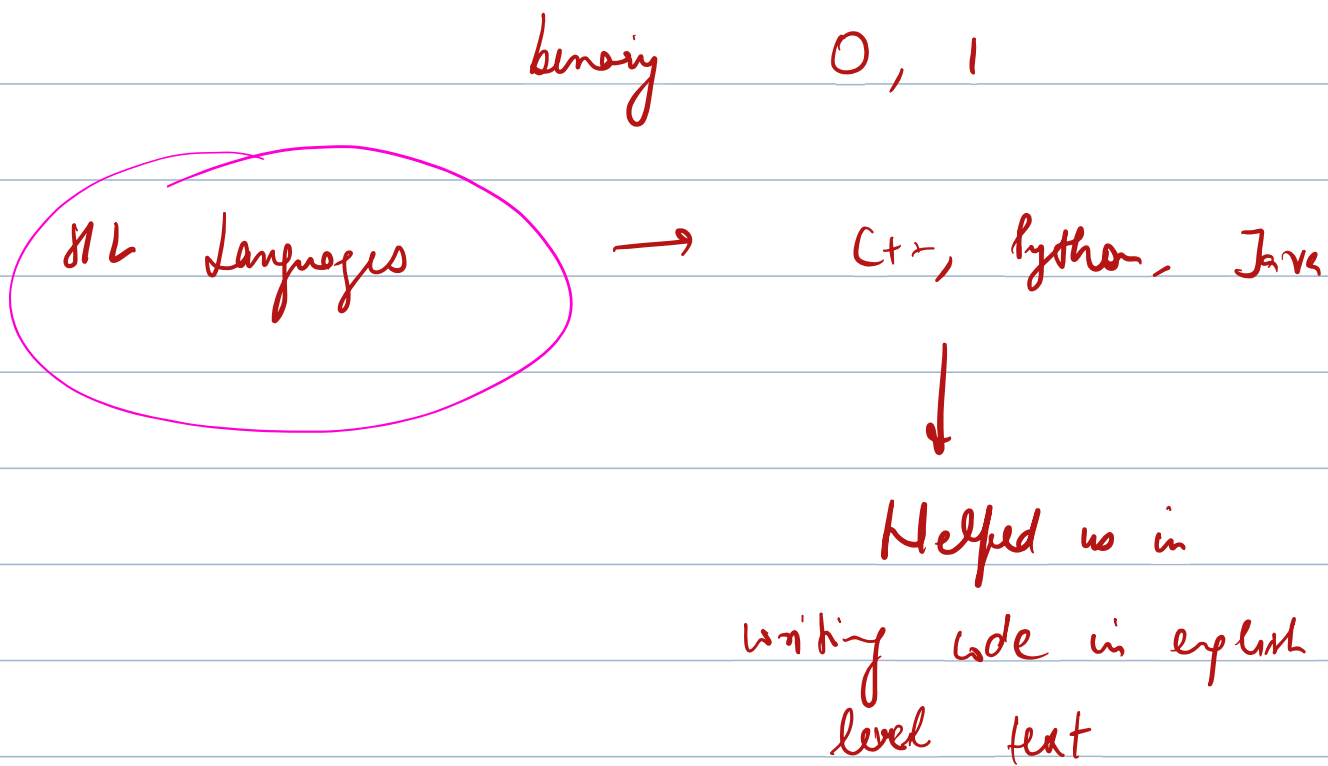
MySQL

RDBMS

Use existing tech until & unless we get significant ROI.

Method

Spring

Spring will help us in 3 main things

- Dependency Management

A ⟶ B

A (B b) {

```
                          }
                    2                    1
          3                          ┌─────────┐
    ┌──────────┐    ┌──────────┐     │ Repo A  │
    │Controller│    │Service A │─────└─────────┘
    └──────────┘      Service B  ──────  Repo B
                      Service C
```
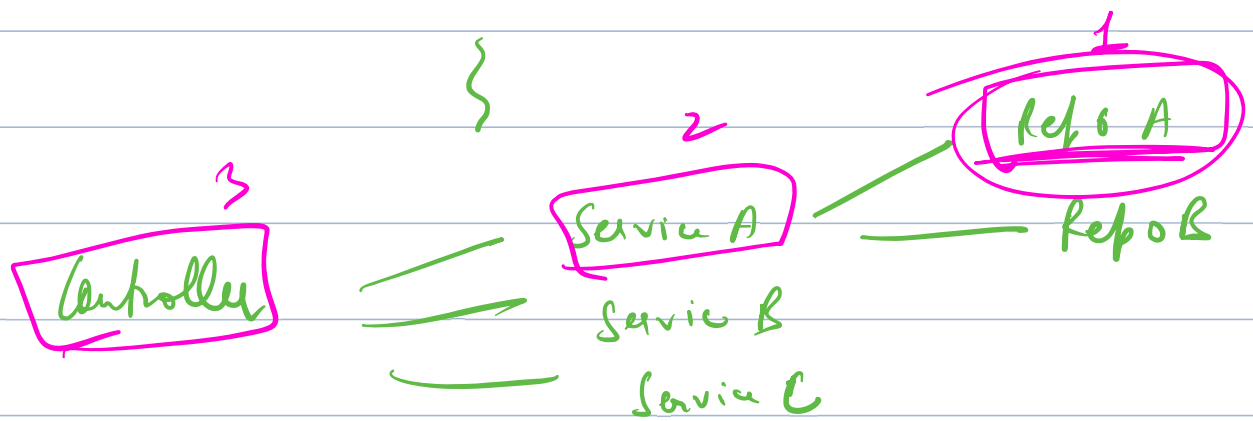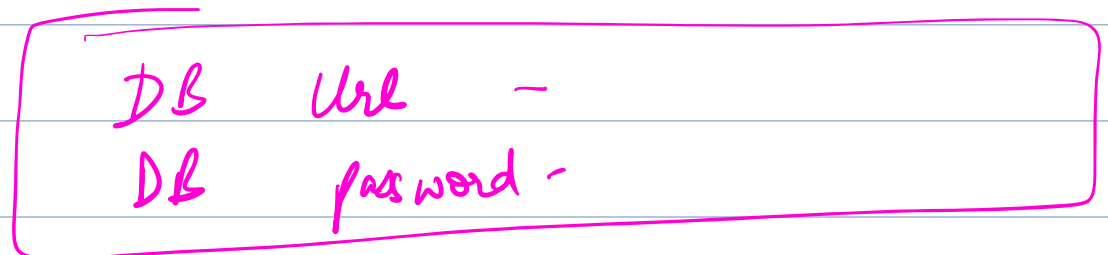
Inversion of Control → rather than us worrying about which object to create first, we have assigned this responsibility to spring

Maven / Gradle

② Config Management →

┌─────────────────────────────────┐
│ DB   Url   -   │
│ DB   password -   │
└─────────────────────────────────┘
                    │
                    ↓
            Test & Prod

```
if ( env == test )     {
    ↳ _____
    ↳ _____
}
else if ( env == prod ) {
    ↳ _____
    ✓ _____
}
```

↳ appsettings_ prod. json   ↙
↗ appsettings _ test. json   ←

Dedicated   Env specific file

③   class Management →

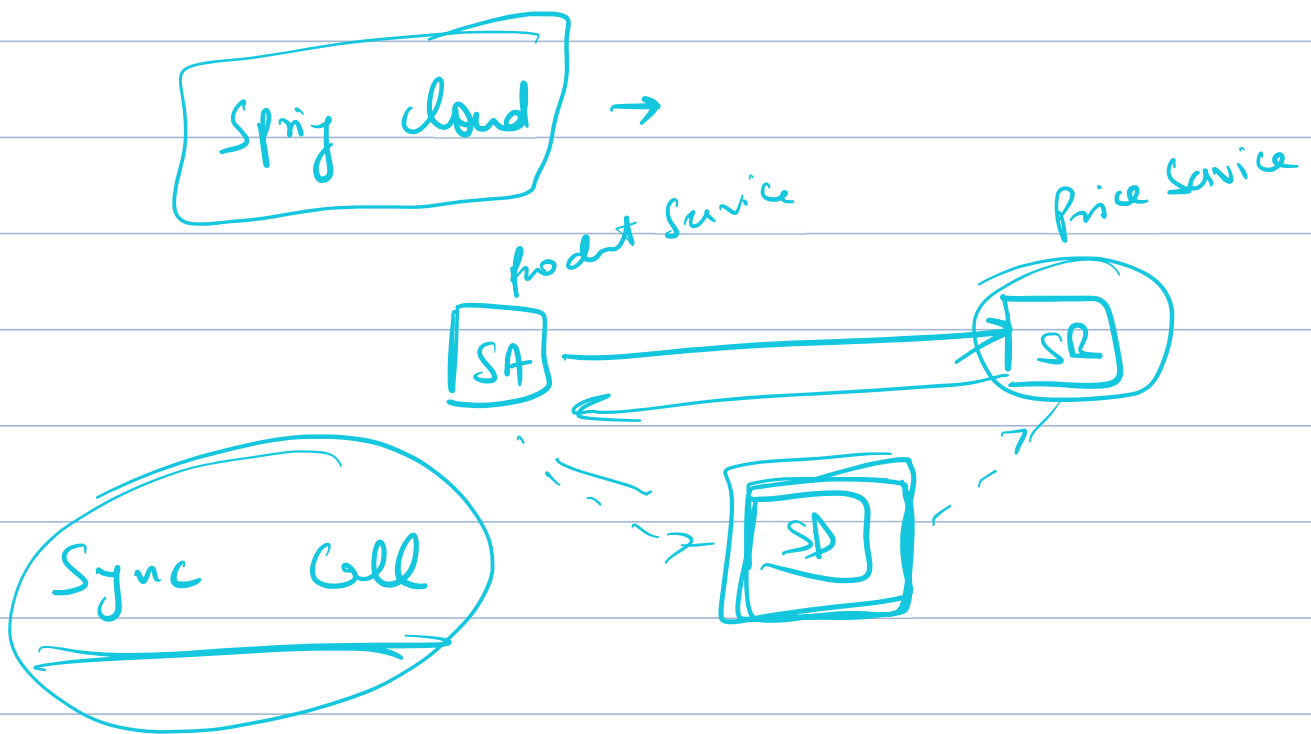Main Application
↑

Interface
fn( )

Type 1   fn( )
Type 2   fn( )
Type 3   fn( )
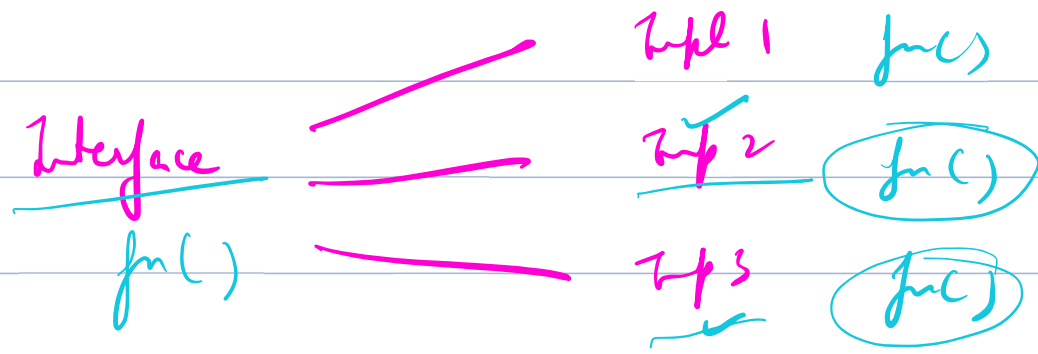
Spring is an umbrella of lot of tools

Spring Web / Spring MVC
↳ creating APIs
↳ connecting to 3rd P API

Spring cloud →

Product Service   Price Service

SA ──────────────→ SR

SD

Sync Call

Spring Security → Auth & Authorization

→ whitelisting / blacklisting

ZN Abuse Prevention Team

whitelist — [
tenant Ids

private preview   (1-2)
public preview   (10+)
GA → General   (100+)
        Availability

Spring boot   →   Spring Web / spring MVC
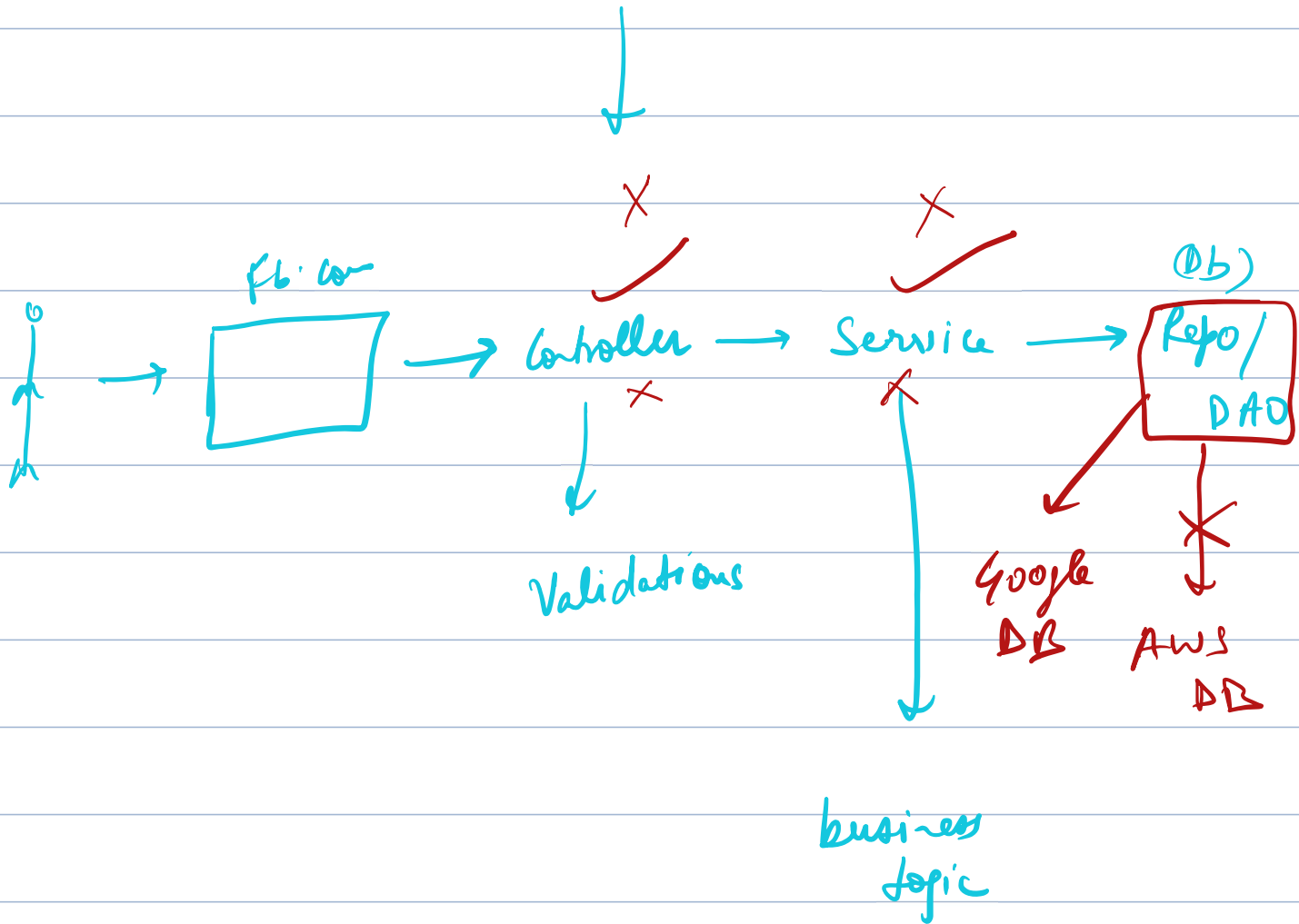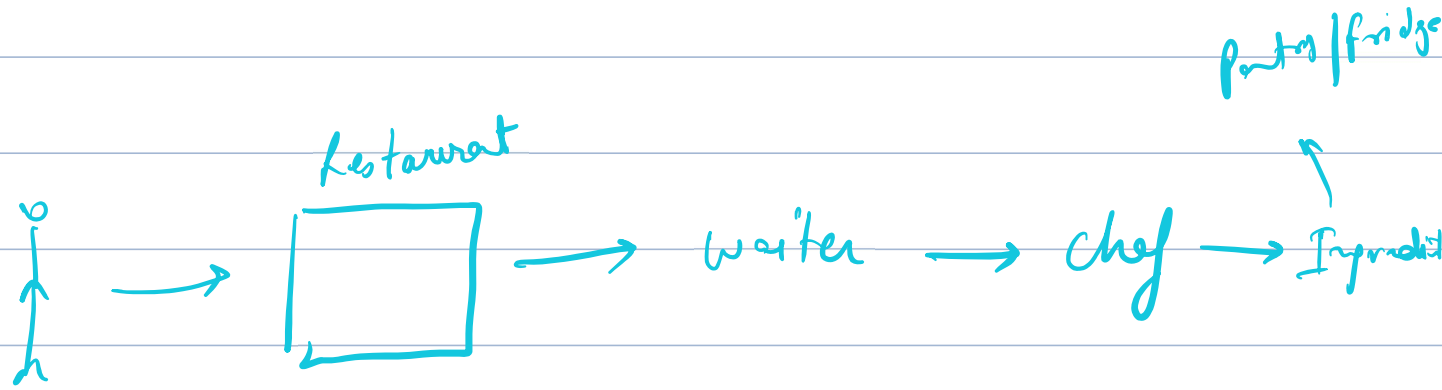
web server   (Tomcat)   ✓

        ↓

Server   Nginx
                +
                web server

/products

MVC

Pantry/Fridge

Restaurant

🧍 → ☐ → waiter → chef → Ingredient

↓

fb.com

🧍 → ☐ → Controller → Service → Repo/ (Db) DAO
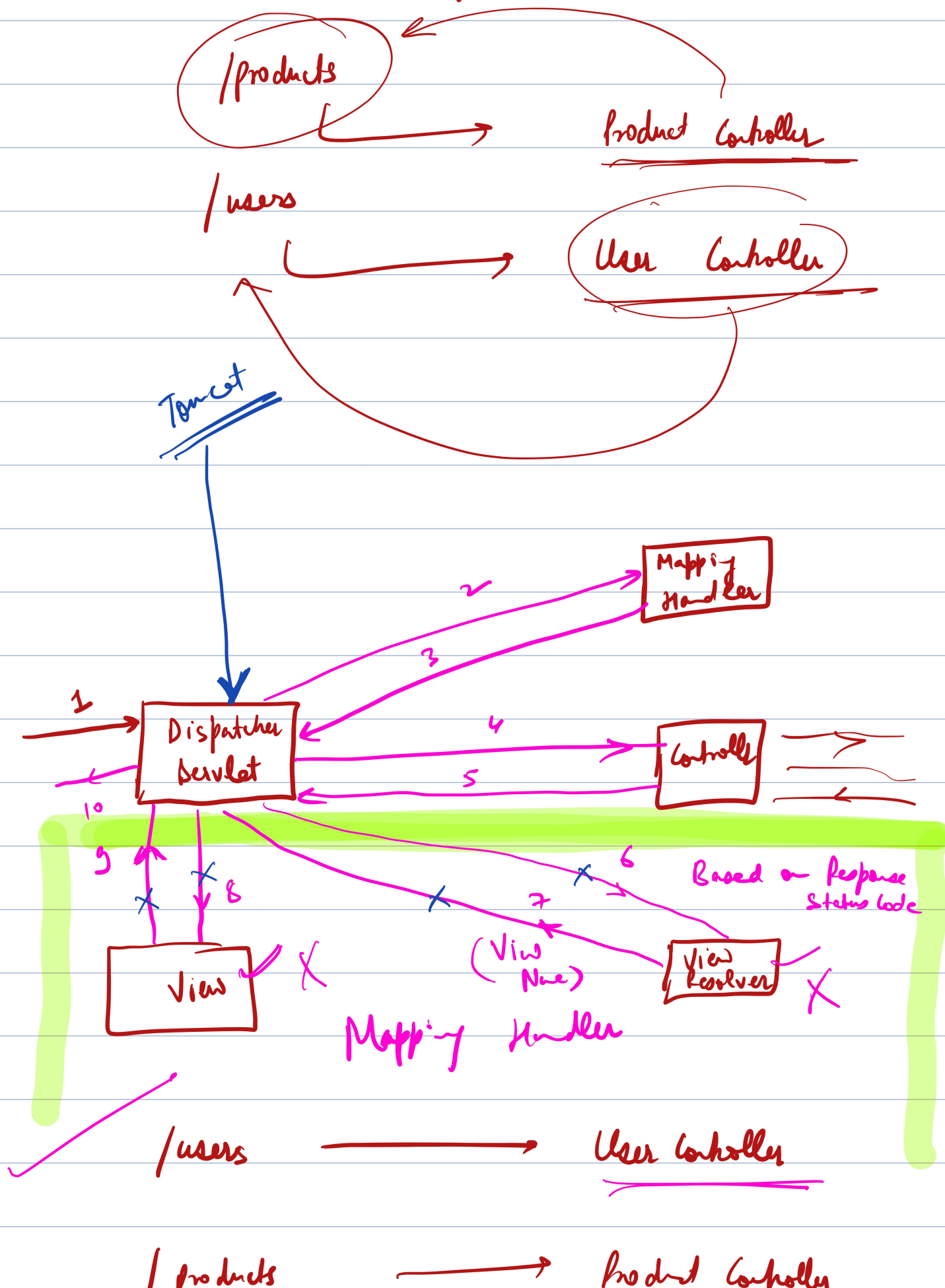
Validations

business logic

Google DB   AWS DB

DAO → Data Accessor Object

↳ Save

↳ getBy Id , getBy Name

↳ deleteBy Id

How our req get served in SB?

/products

Product Controller

/users

User Controller

Tomcat

Mapping Handler

1 → Dispatcher Servlet

2

3

4

5

Controller

Based on Response Status Code

10

9

8

X

7 (View Name)

6

View

X

View Resolver

X

Mapping Handler

/users ——→ User Controller

/products ——→ Product Controller

API Method
POST   → update_product

GET   → getproduct

POST   update_product