Client - Server Model
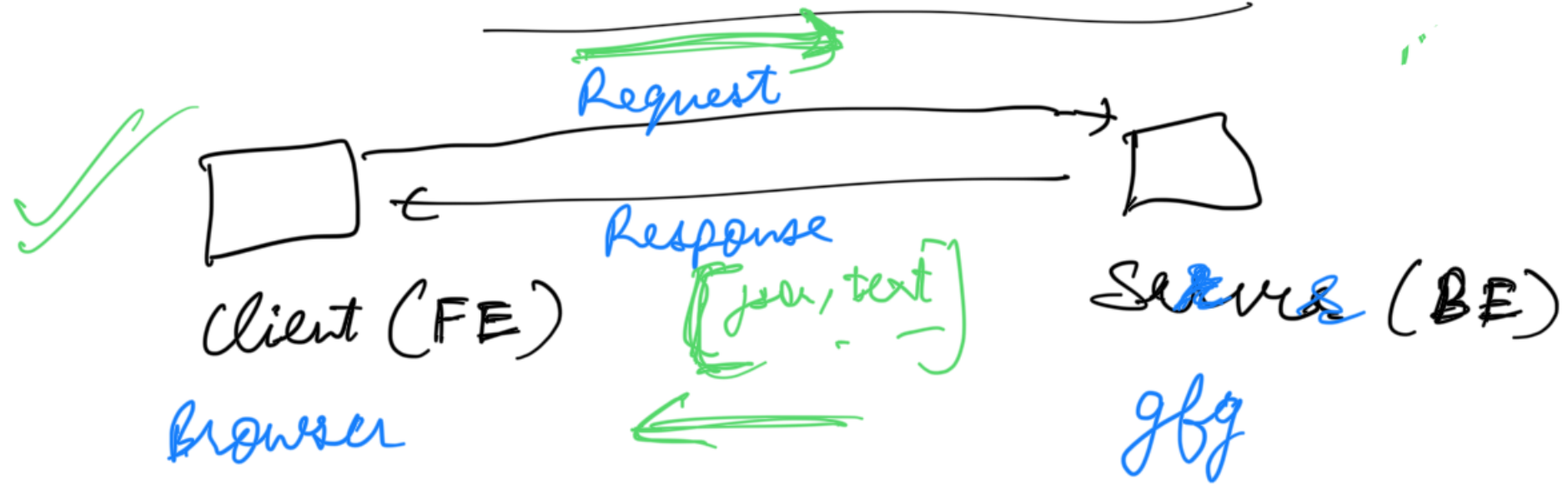
Request

Response
[json, text]

Client (FE)

Server (BE)

Browser

gfg

[dumb entity] brainless
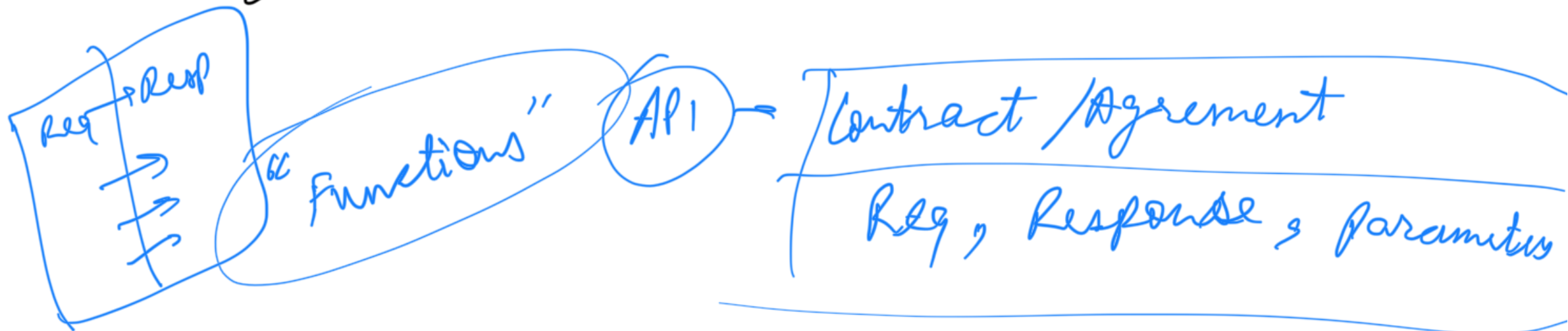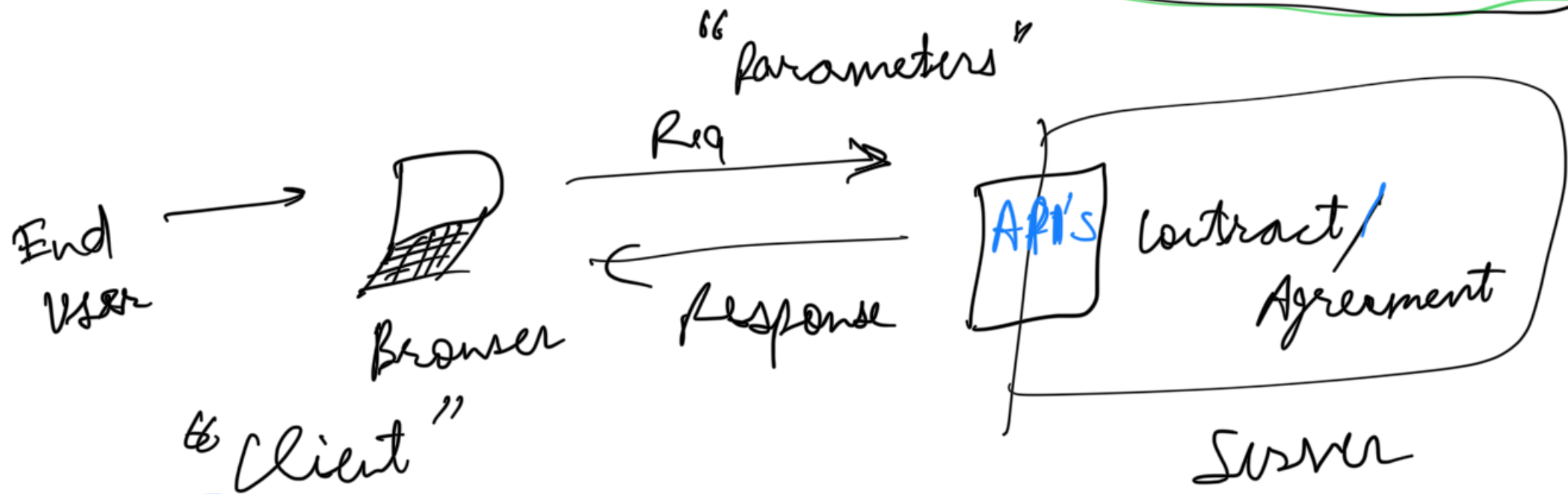
Webpage

[no use]

only FE

Tools

postman

to test the BE

FE client

Req
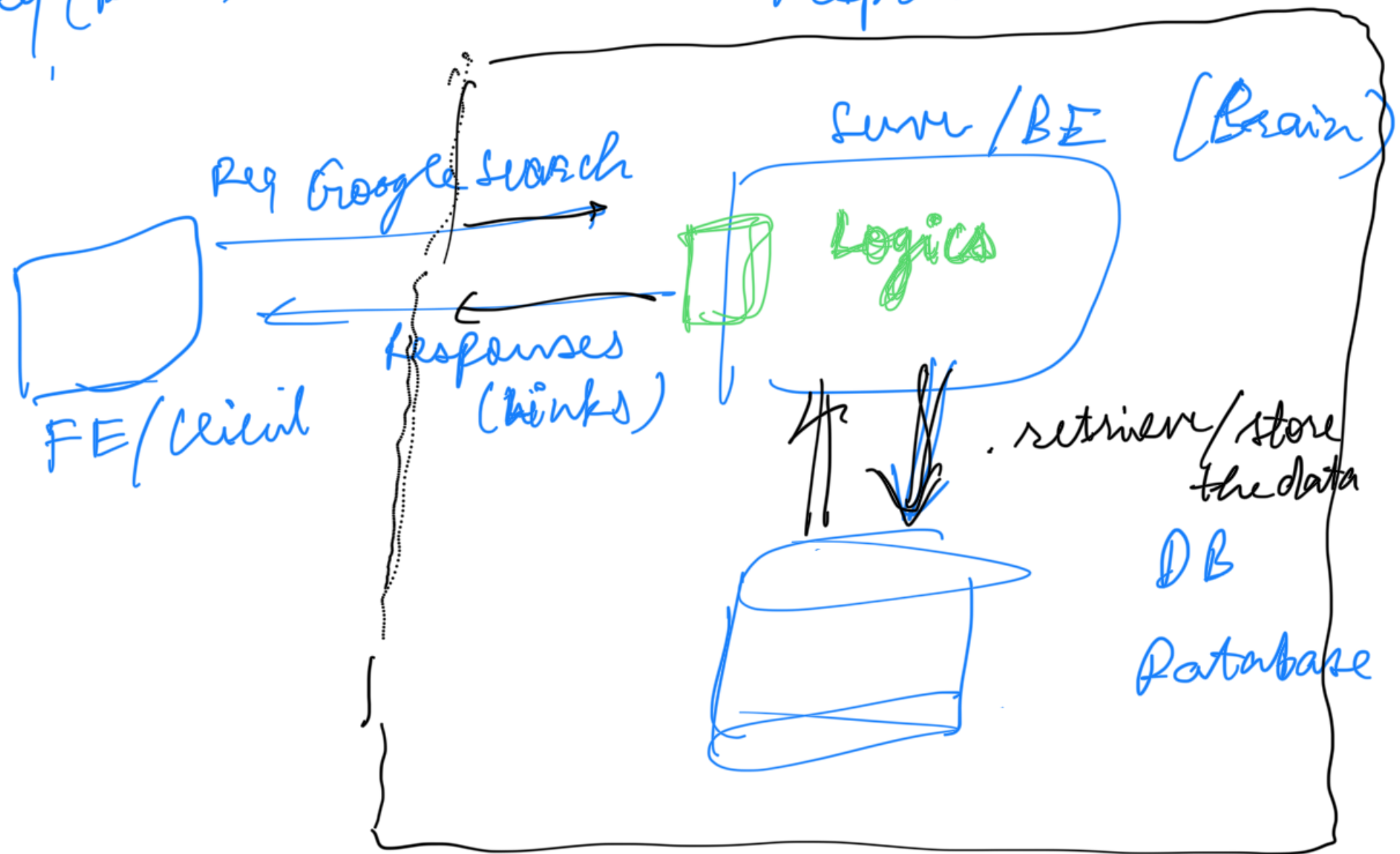
Response

WhatsApp

user 2
"Client"

"User1" "Client"

Req (M, User2)

"Response"

"W Server"

① Polling
② Long polling
③ Web Sockets

"parameters"

End User → Browser

Req →

← Response

"Client"

API's Contract/Agreement

Server

Req | Resp

"Functions" API = Contract/Agreement

Req, Response & Parameters

Input
Req (Params) → F(x,y) → Output
Response

Server/BE (Brain)

Req Google Search

Logics

FE (Client)

Responses (links)

retrieve/store the data

DB
Database

How data is stored/retrieved to/from DB
↳ BE job
be aware of it

(FE need not be ~~~~~)

───────────────────────────

Composition of URL

⇒ URL   Uniform Resource Locator

~~geekf~~
geeksforgeeks . ~~org~~

Client ——Req——→ Server

Any functionality on the
        web

protocol: Domain name

http   Sub domain   Main domain   Top level
https   geeksforgeeks.   domain

ftp
smtp

Optional par
www

.org/
.com/
.edu

Host number
localhost:8080

complete / search

API path endpoint

127.0.0.1

docs.google.com
geeksforgeeks.org / profile / create
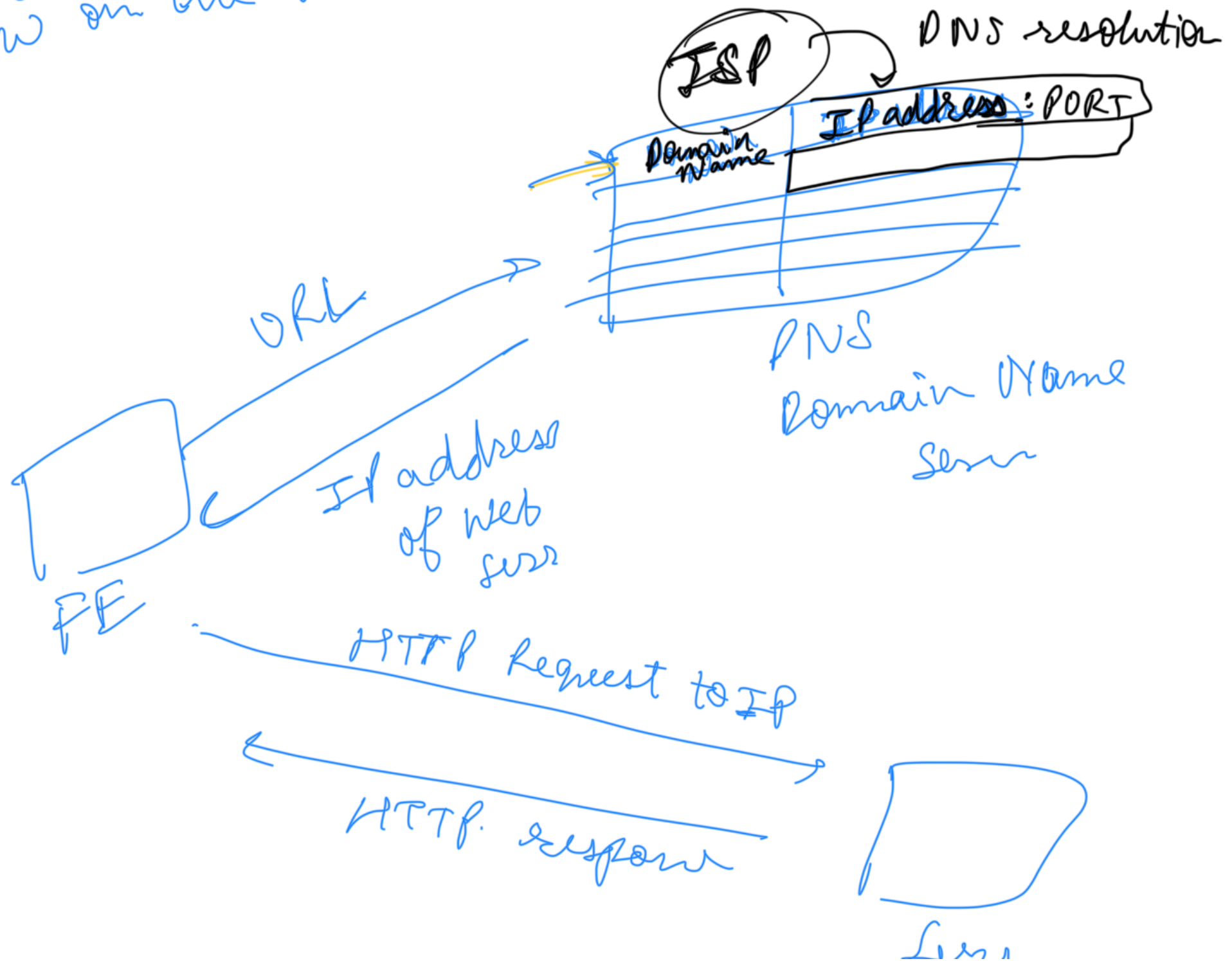save
list
delete

IP address : PORT

API path

Address → House No, Strct No, Area

~~Host~~
Address
of any
H/w on the web

$\longrightarrow$

IP address # PORT
Number

ISP

DNS resolution

Domain Name | IP address : PORT

URL

DNS
Domain Name
Sesn

FE

IP address
of Web
Sesr

HTTP Request to IP

HTTP response

IP address → Unique address of
every H/W on the internet

API gateway

C1
C2
C3
C4

API gateway

LB

[address]
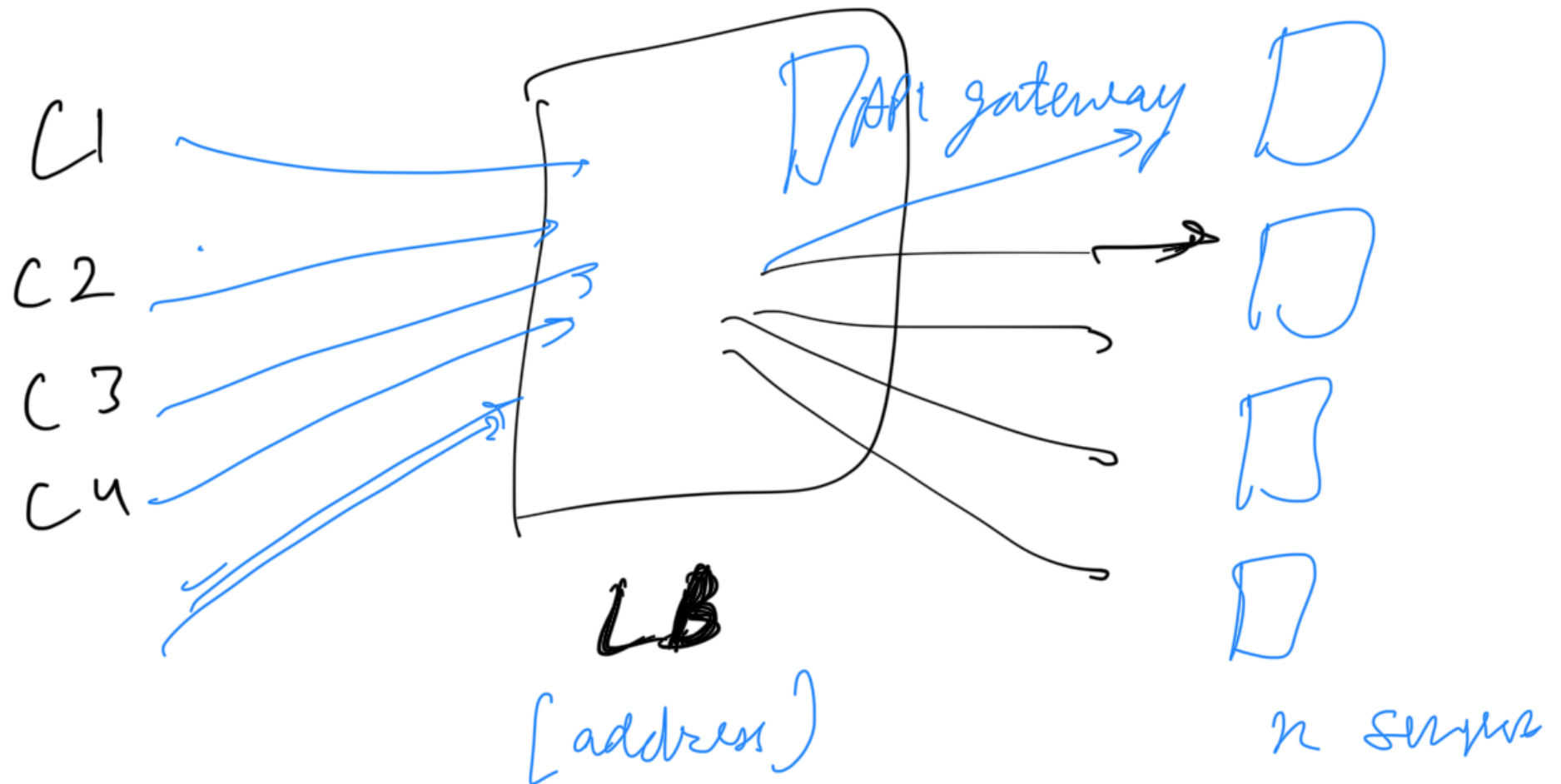
Load Balancer

Balances the load
b/w servers

n server

JAVA application	Server
Runs and stops	Running for an inf. amt of time

OSI layers

Support

Chrome

Application

↓

for

↓

Se

↓

h

Target

Application

(transport)

↓

IP layer   IP address

P physical
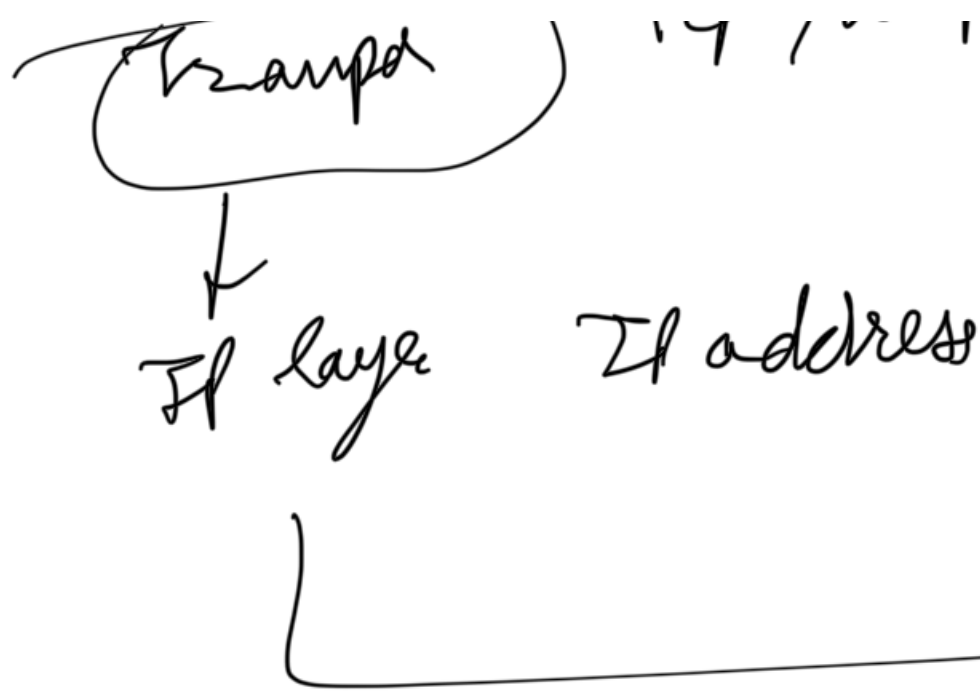
before spring boot → (framework)

logic

① Java Servlet

Java program that acts as a middle layer

client

→ servlet

← service()

init()
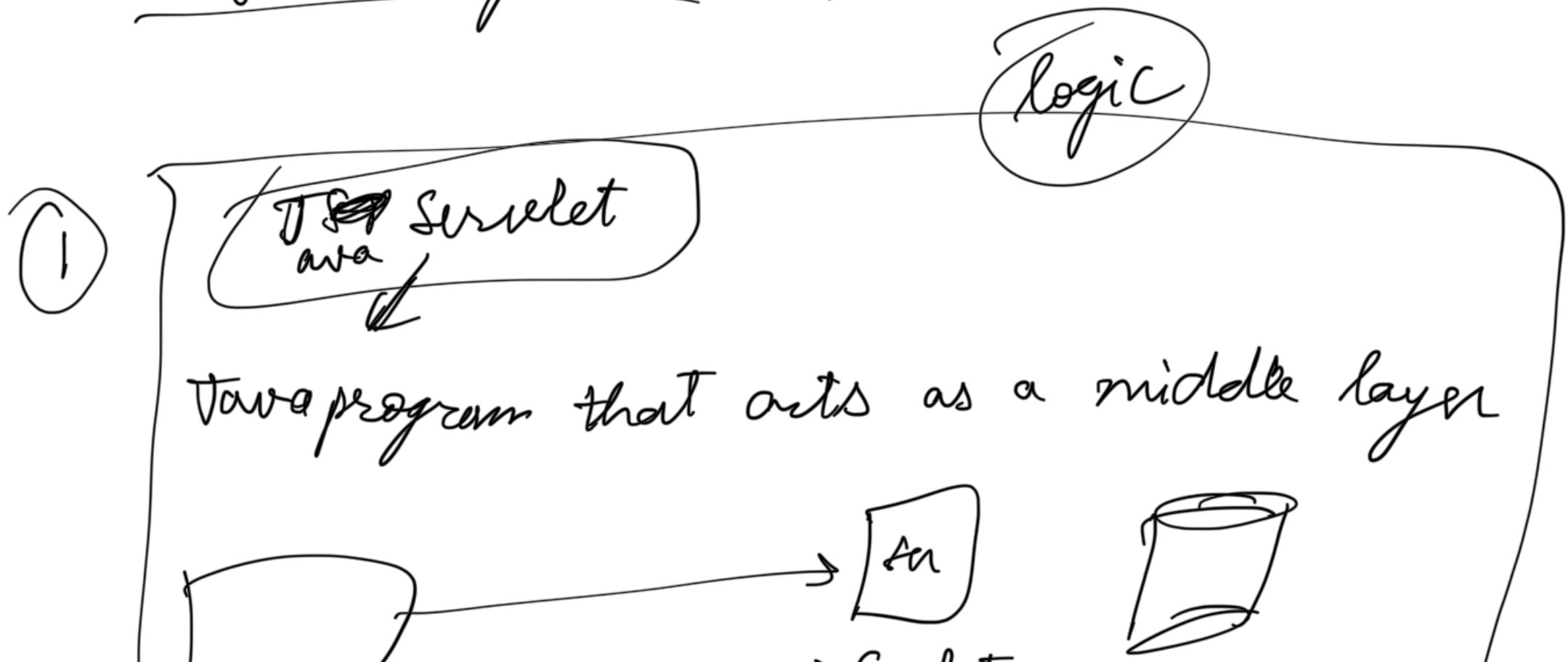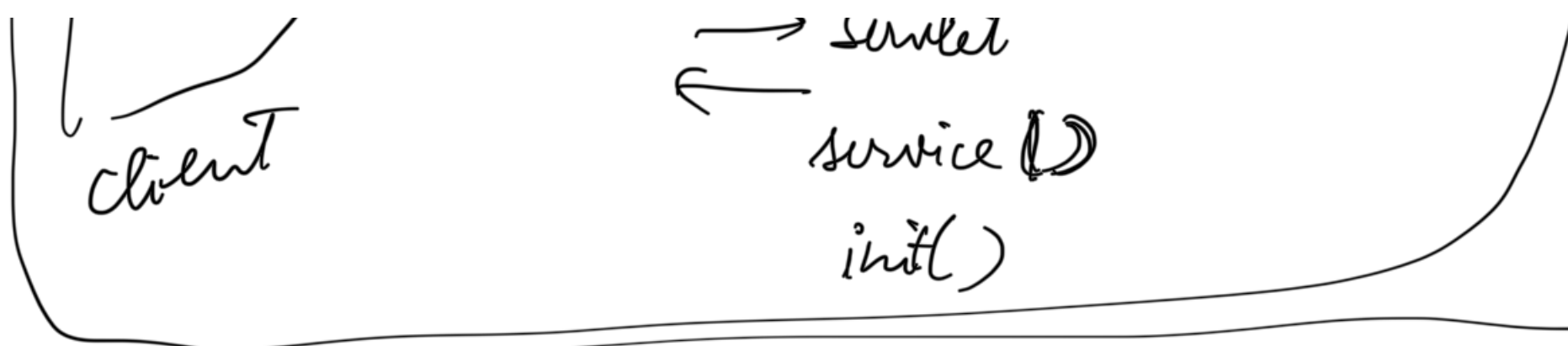
② (JSP) ~~Frontend~~ Dynamic Web content

③ JDBC → database interaction, connection
Driver
Manager
SQL statement

④ XML config.

⑤ Server → Tomcat, JBoss, Weblogic

Dependency Management

Spring boot $\neq$ Spring Module

Spring Framework $+$ Embedder issues (Tomcat, Jetty) $-$ XML config. @config

Dependency Injection IOC

JPA