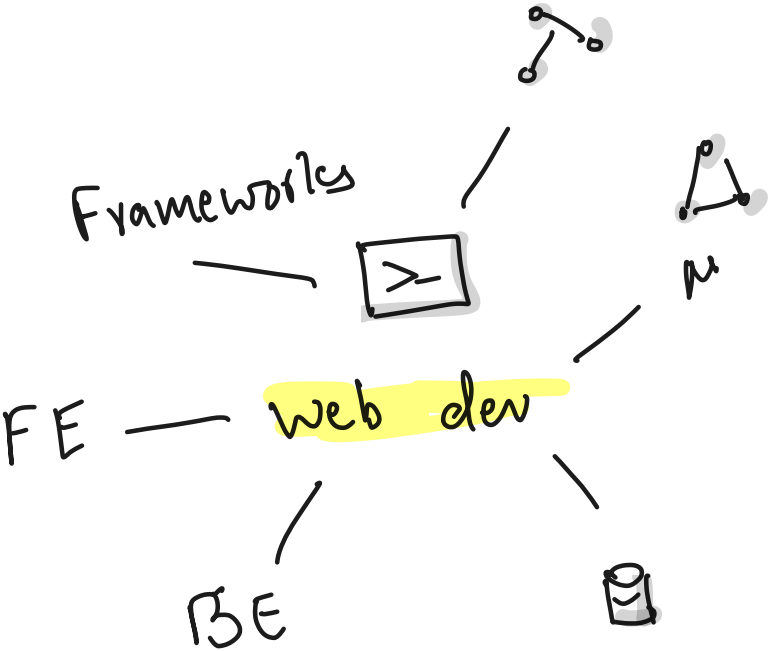
Everything You Need For Web Development

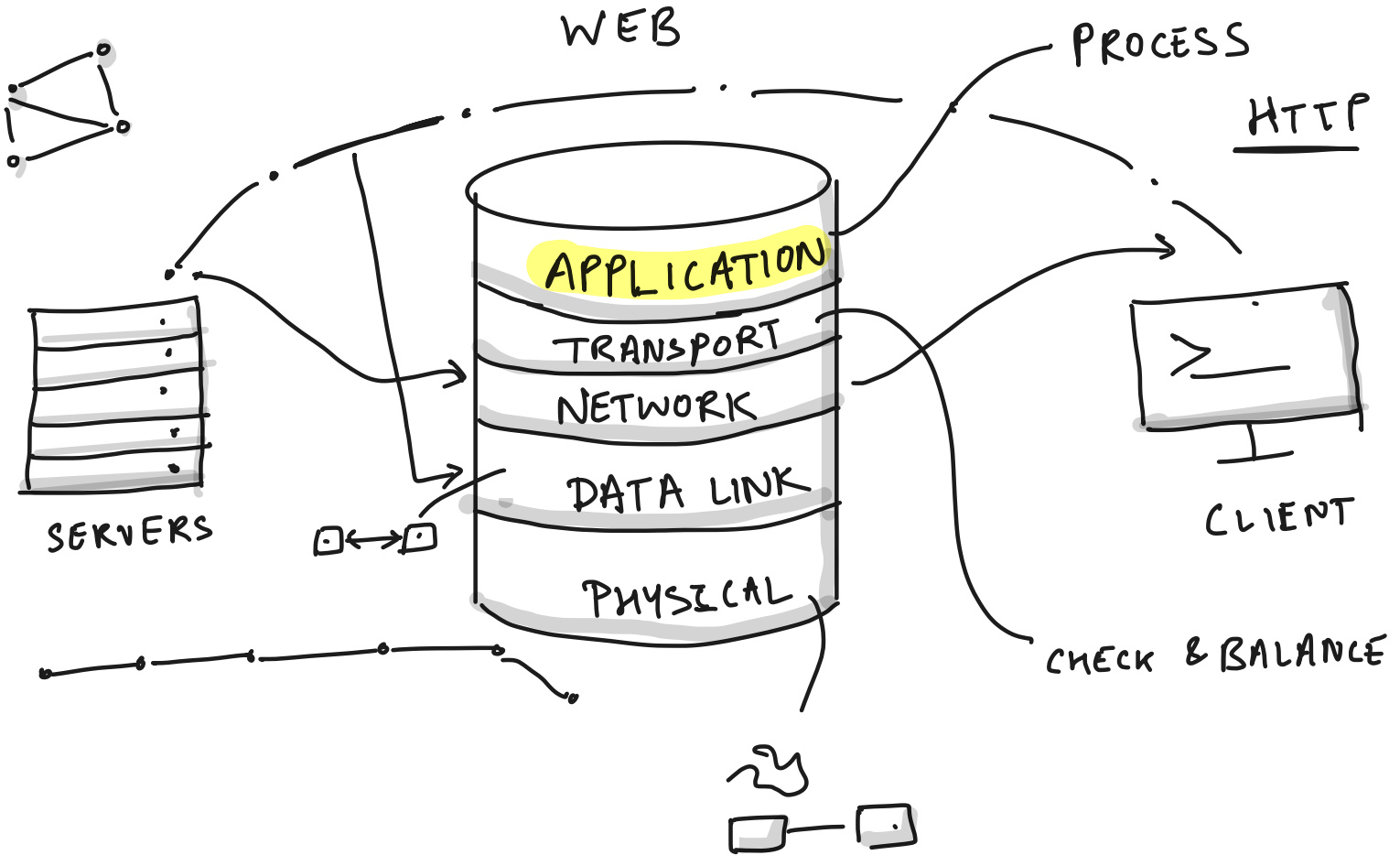
~ Anshul Yadav

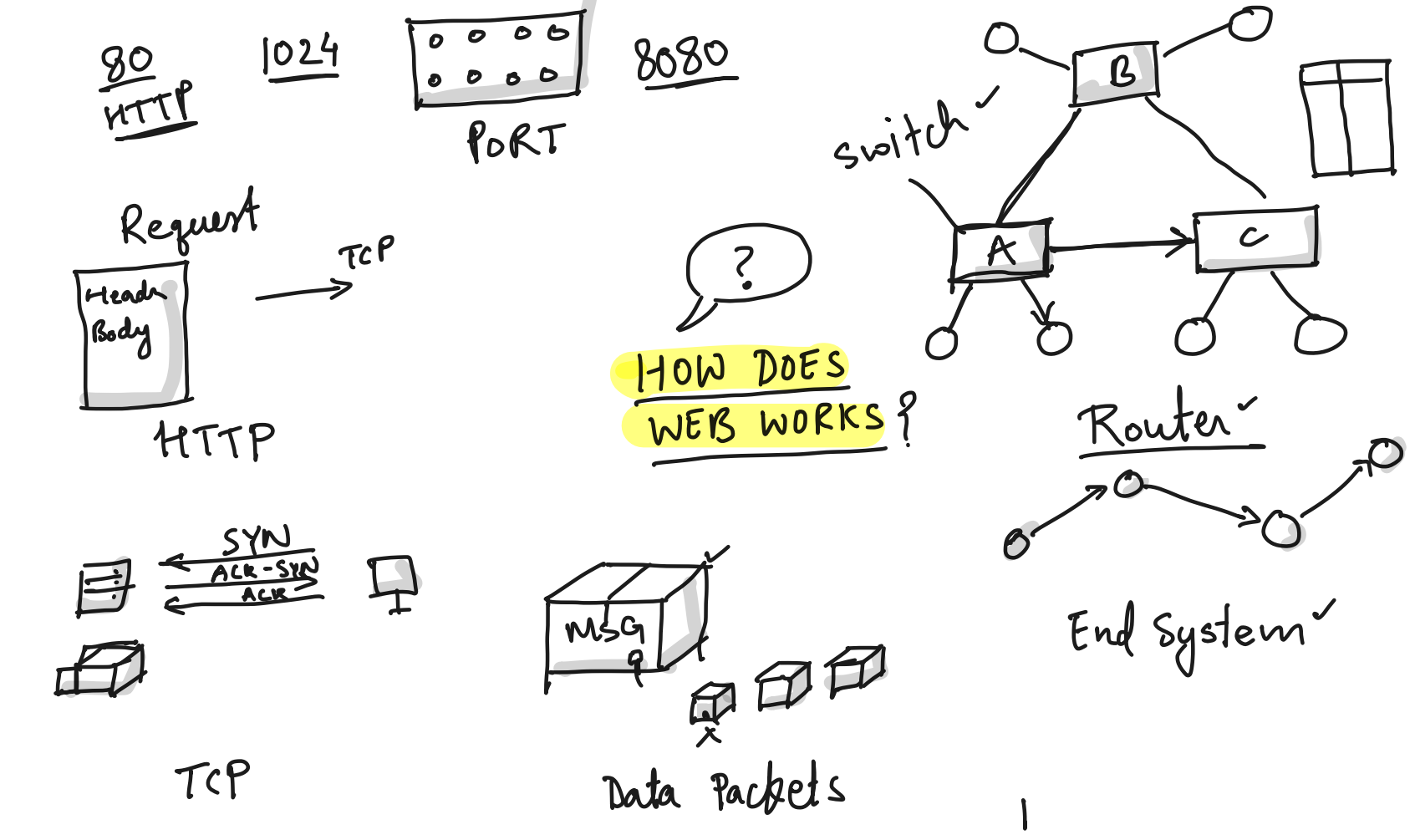
- 2x speed

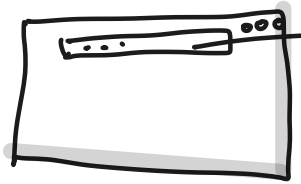
Who?

1. One Programming Language

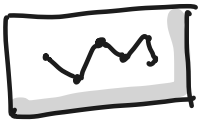


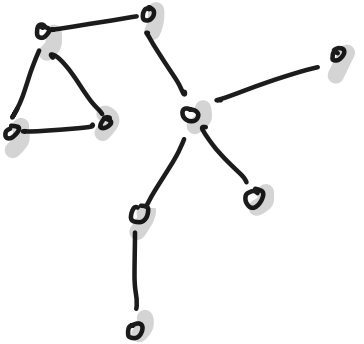

WEB
PROCESS
HTTP
APPLICATION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
SERVERS
CLIENT
CHECK & BALANCE


80
HTTP
1024
8080
PORT
switch
B
A
C
Request
Header
Body
TCP
HTTP
?
HOW DOES
WEB WORKS?
Router
SYN
ACK-SYN
ACK
TCP
MSG
Data Packets
End System



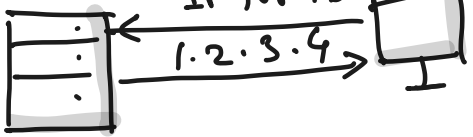
URL
||
Domain





Network

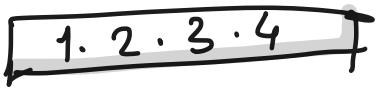
!! DNS



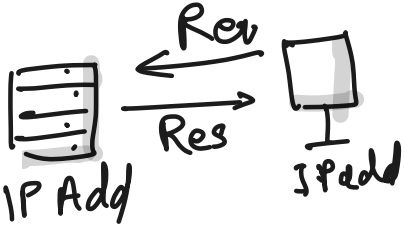
URL → IP

DNS SERVER

HOW DATA PACKETS FIND THEIR WAY?



IP Address



IP Interned Protocol

Ifconfig – Linux
ipconfig – window

— logic

INTRODUCING DATABASES

DB

WESITES

Static

No DB

DYNAMIC

DB

DB Types

SQL

- Relational Tables
- Rigid structure

- Eg.
  - MySQL
  - PostgreSQL
  - MariaDB

NoSQL

- Documents, Files, key-value graph
- Flexible

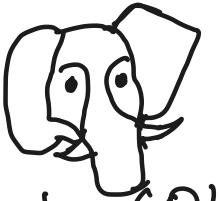
Eg:

- MongoDB
- Redis
- Apache CouchDB

SQL

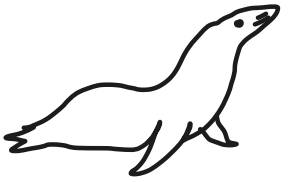
MySQL

- Oracle
- SQL
- ☑
- Easy



PostgreSQL

- Concurrent
- SQL
- Open Source
- macOS



MariaDB

- Fork of MySQL
- Replace MySQL

Speed ☺

- Cache

SQL ✗

NoSQL ✓

NoSQL ✓

NO SQL

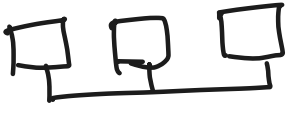
mongo

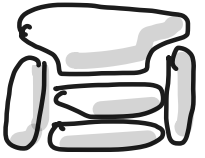
- Document Based

{ key : value }

- Scalable
- Open Sourse

```
use newDB;
```





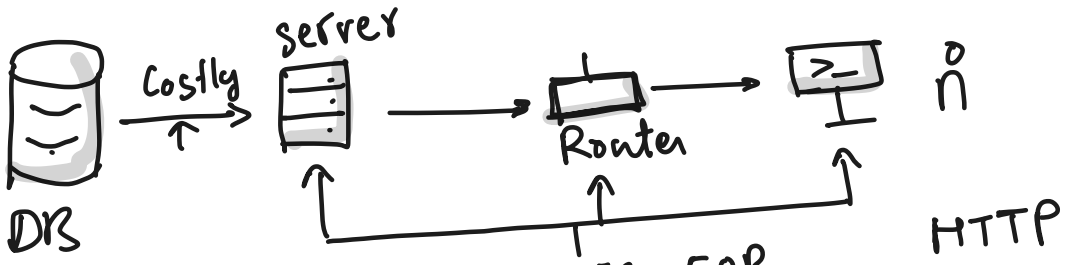
CouchDB

- Document Based
- Easy to use

redis

- In memory
- Store lists, strings, Hashmaps, sets, etc
- RAM

Web CACHING



Benefits

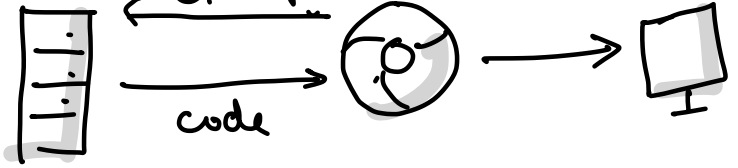
1. Low traffic
2. Speed
3. More data, Scale
4. Availability ↑

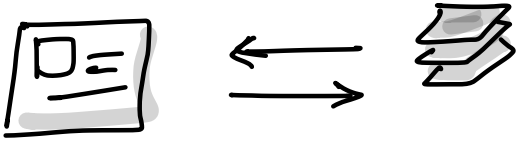
NoSQL ★ MongoDB ★ Redis

★ CouchDB

CODING FOR WEB

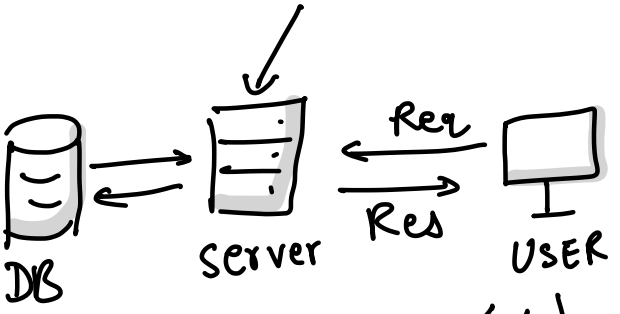
CLIENT SIDE

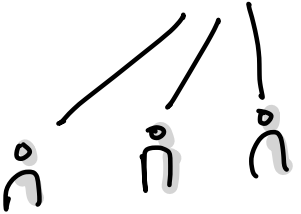




lang. { JS, HTML, CSS } → NEXT

SERVER SIDE





- Email

lang { JS, Python ~, <PHP>, Ruby, C# }

HTML

HyperText Markup Language

Document Structure

```
<!DOCTYPE html>
<html>
    <head> content</head>
    <body> content</body>
</html>
```

<!-- comment xyz -->

tag

eg:

 This

OT content CT

Element

selfclosing

Common tags

h1, h2, ... h5,

p, img, a,

ul, ol, li, br

CASCADING STYLE SHEETS

CSS

INLINE

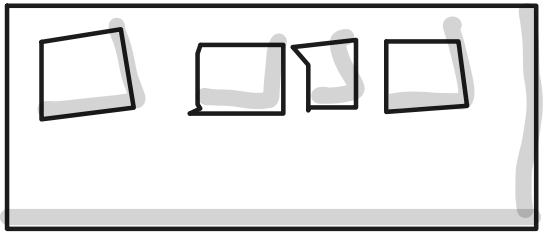
```
<h1 style="color: blue">Blue<h1>
```

INTERNAL

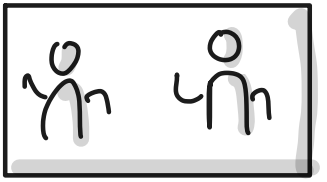
```
<head>
  <style></style>
</head>
```

EXTERNAL

```
<link></link>
```



LAYOUT



ANIMATIONS

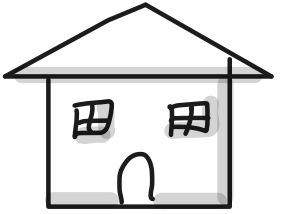
SASS & LESS

PREPROCESSOR

CSS

JAVASCRIPT

~~JAVA~~



ANALOGY

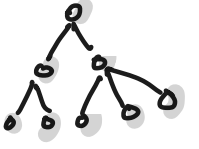
```
Console.log("Hello World");
```

Hello World

JS

INTERNAL

EXTERNAL

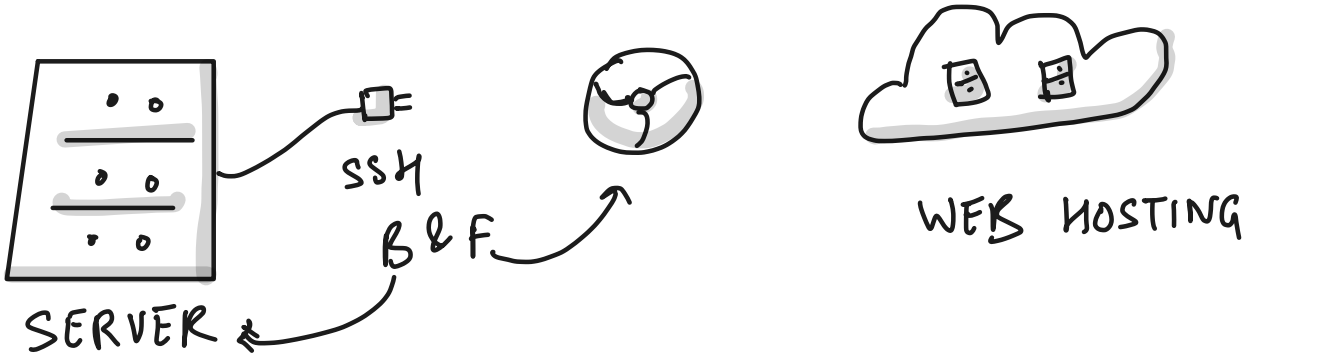


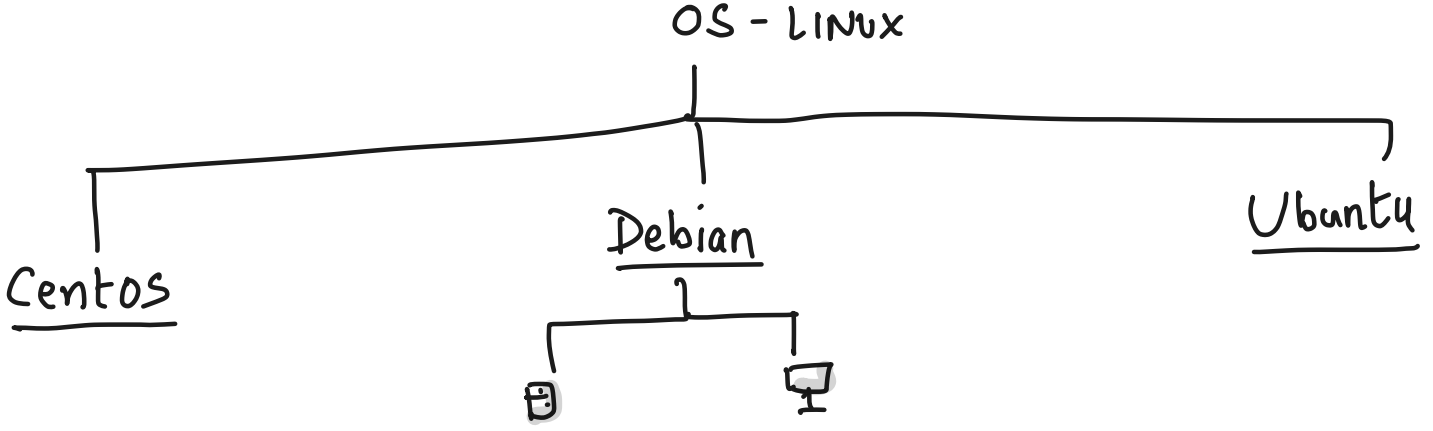
DOM

```
<script> consolelog(uuu); </script>
```

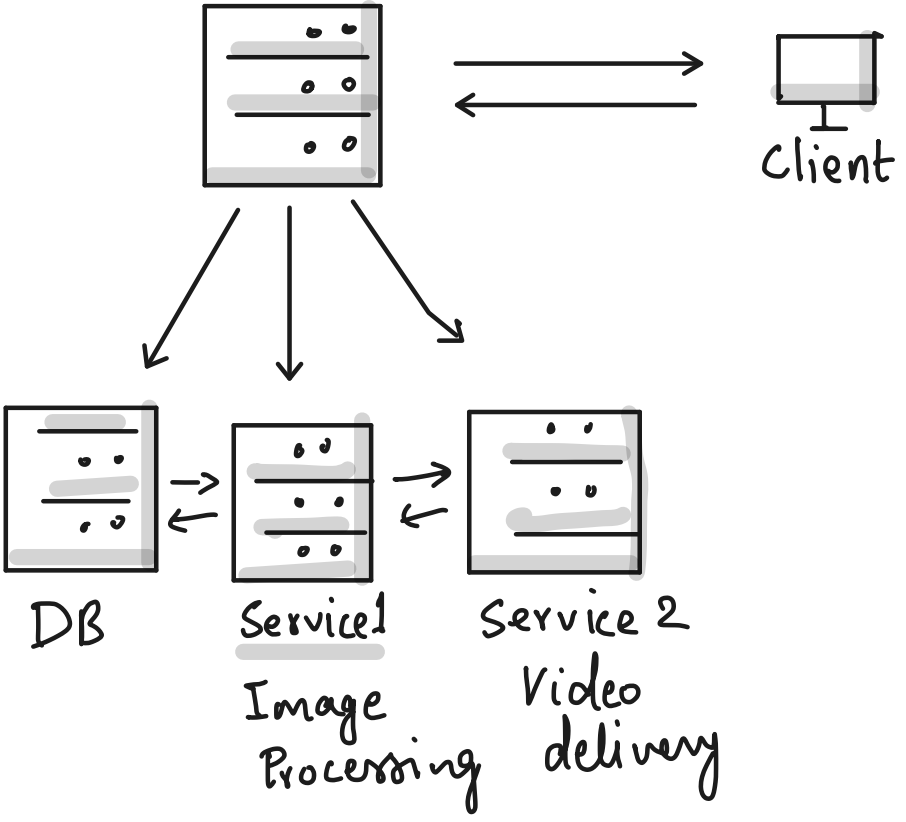
```
<script src="myjs.js"></script>
```

BACK END

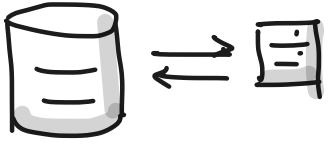




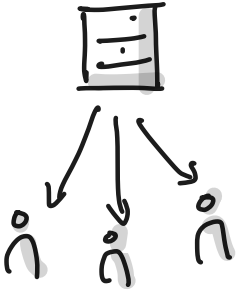
MICROSERVICE ARCHITECTURE



WHAT DO BACKEND ENGINEERS DO?

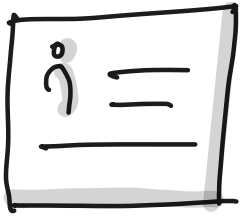


DB



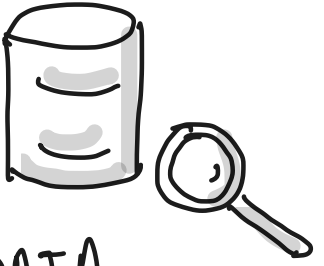
CUSTOMIZED

ACCESS CONTROL



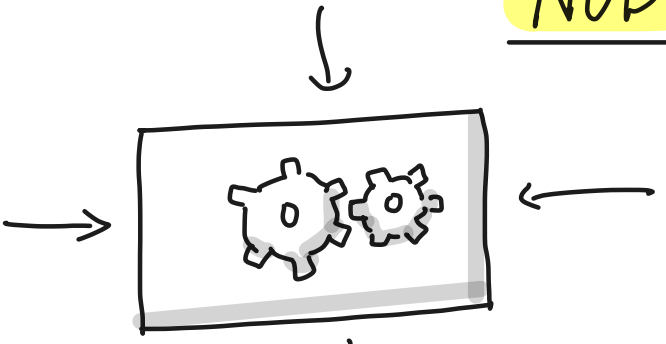
SESSION

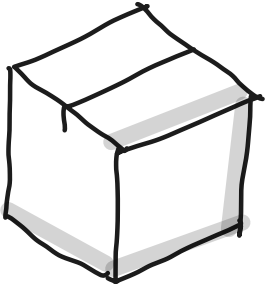
NOTIFICATIONS

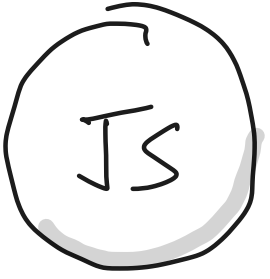


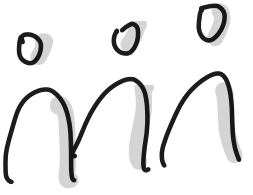
DATA ANALYSIS

NODEJS

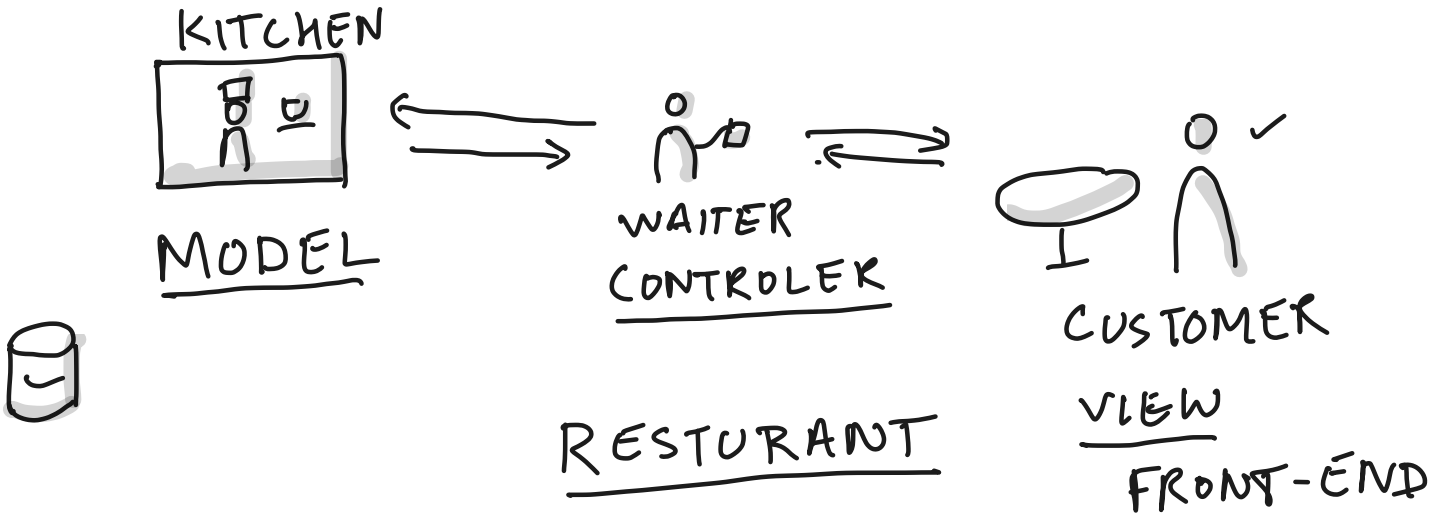




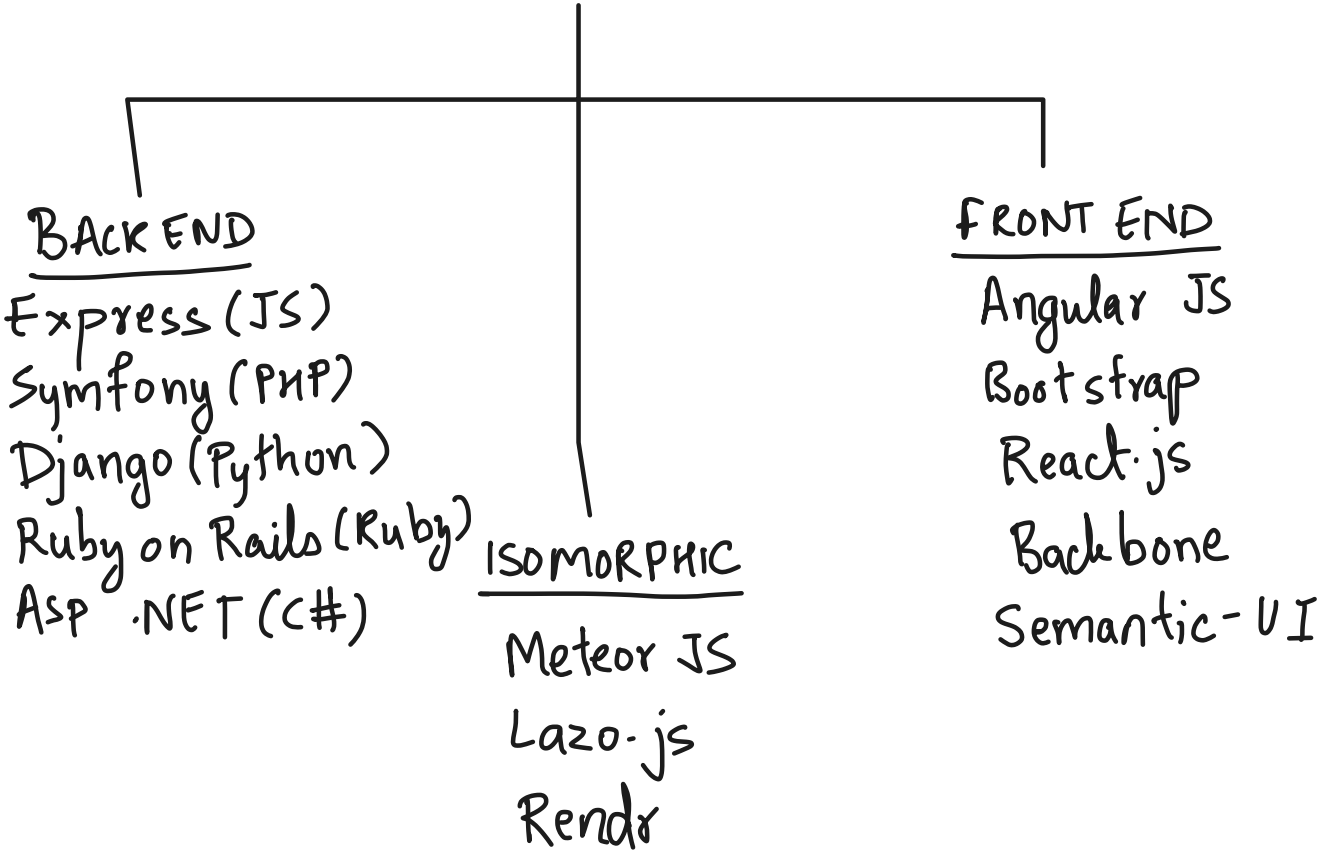




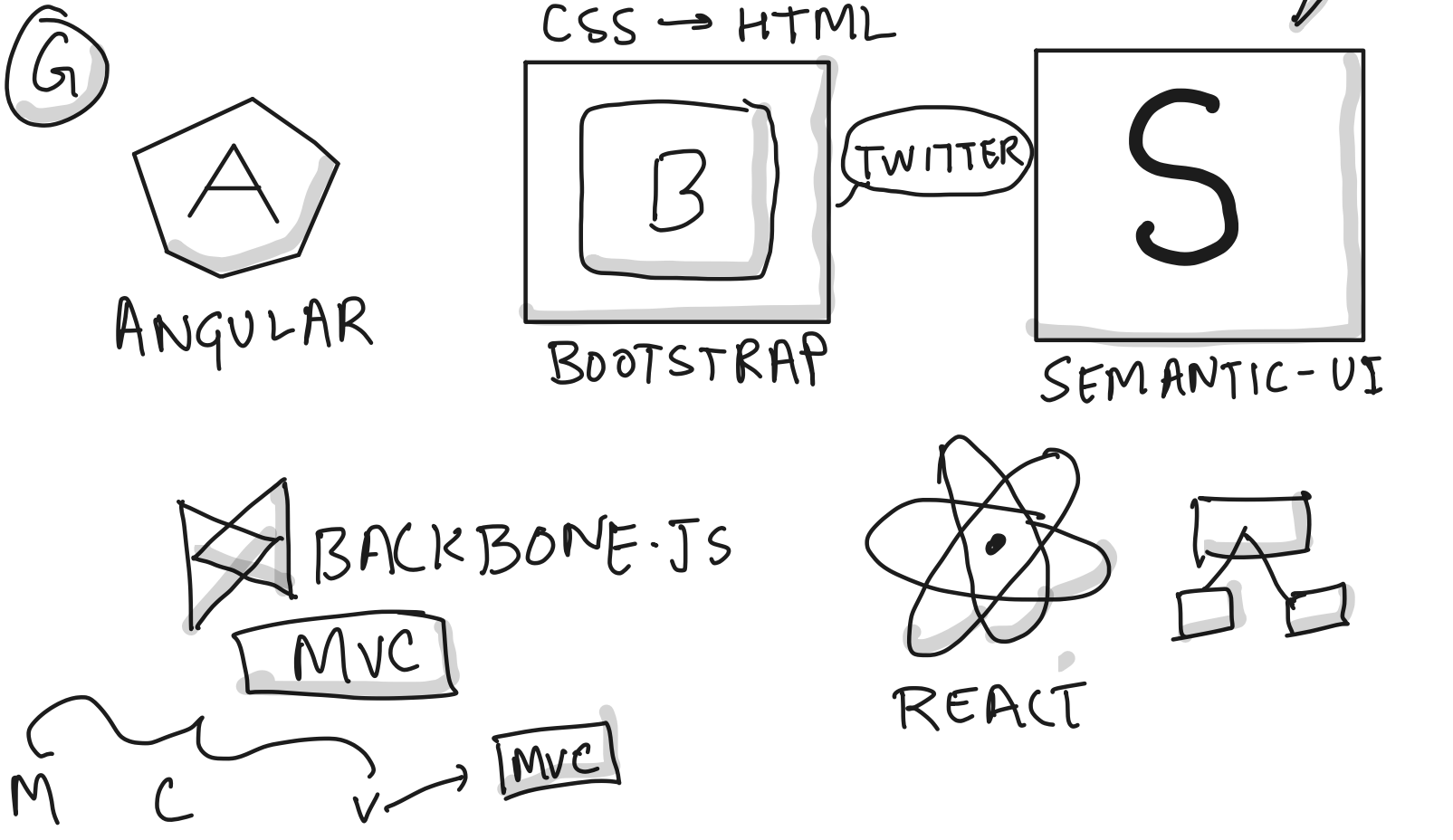
MVC ARCHITECTURE



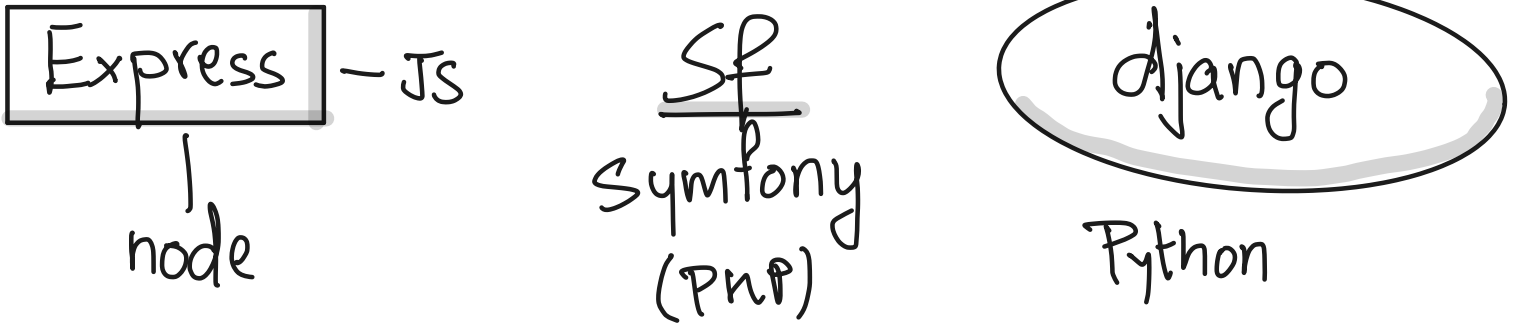
TYPES OF FRAMEWORKS



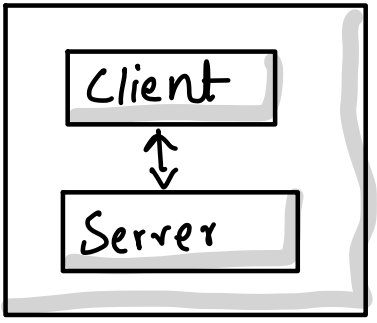
FRONT-END FRAMEWORKS



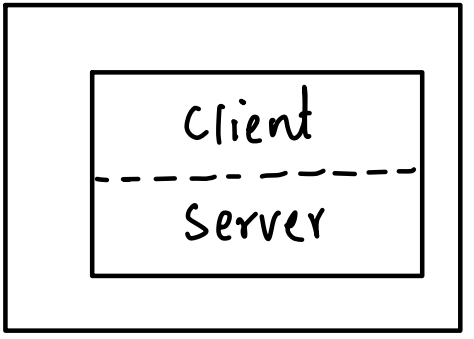
BACK-END FRAMEWORKS



ISOMORPHIC FRAMEWORKS



TYPICAL



ISOMORPHIC

MVC

Meteor JS

Blaze

MongoDB

Prototyping

Lazo.js

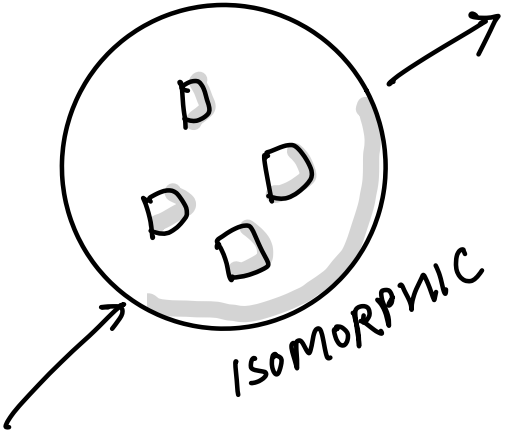
Backbonjs

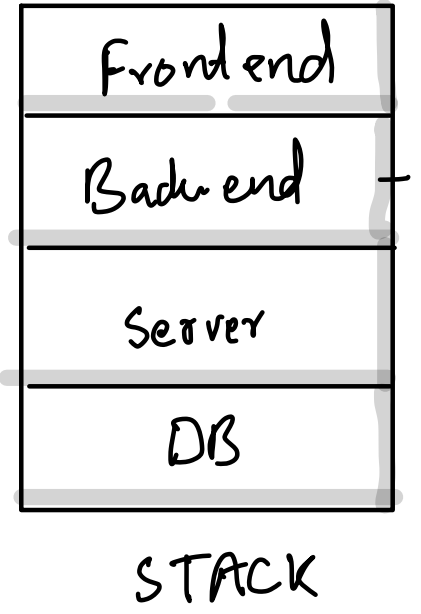
"Optimized First. Page load"

Rendr

Backbone.js

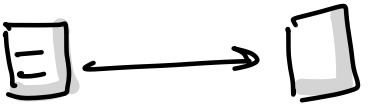
SOFTWARE STACKS



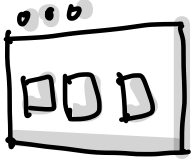


STACK

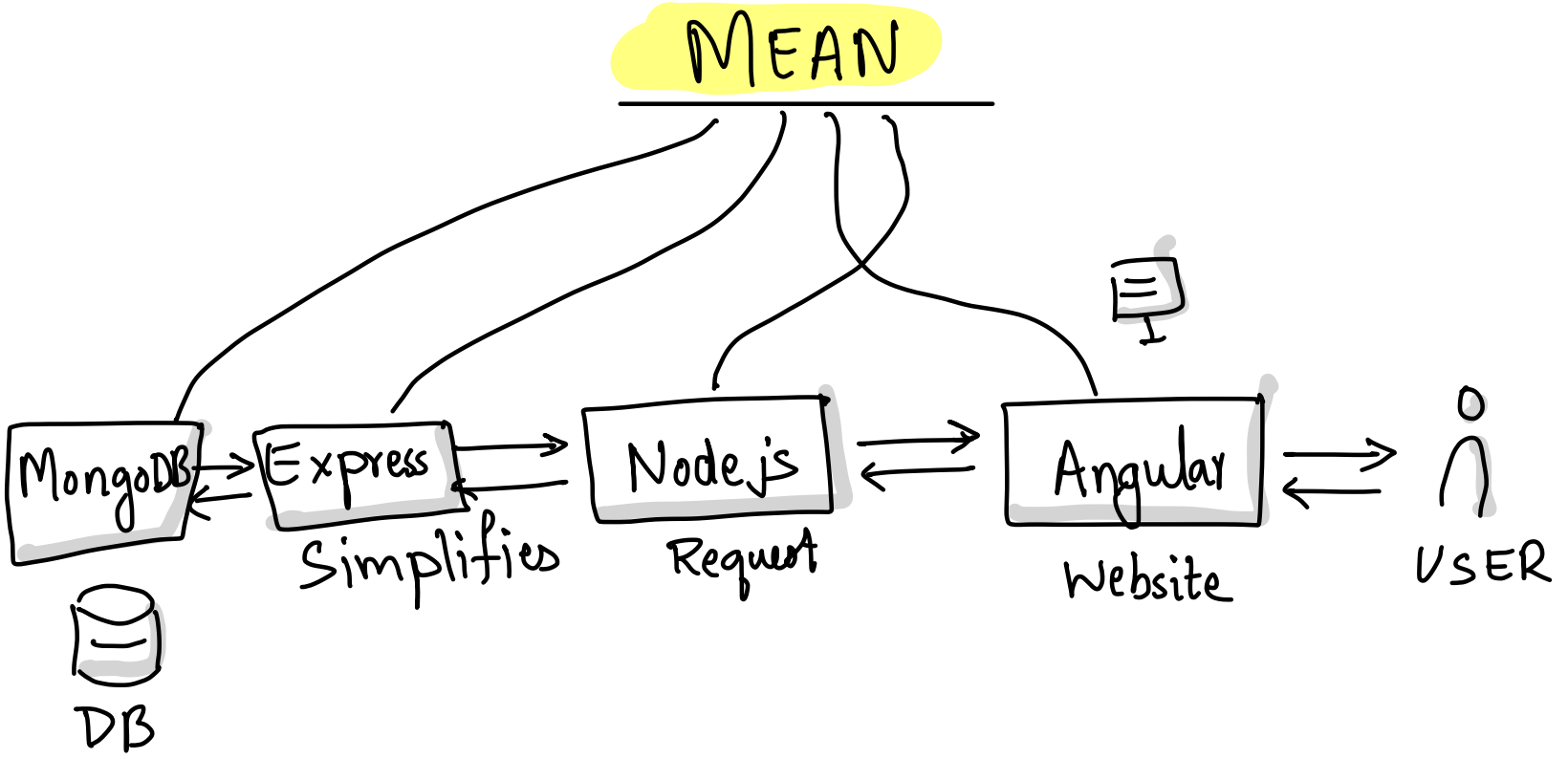
LAMP



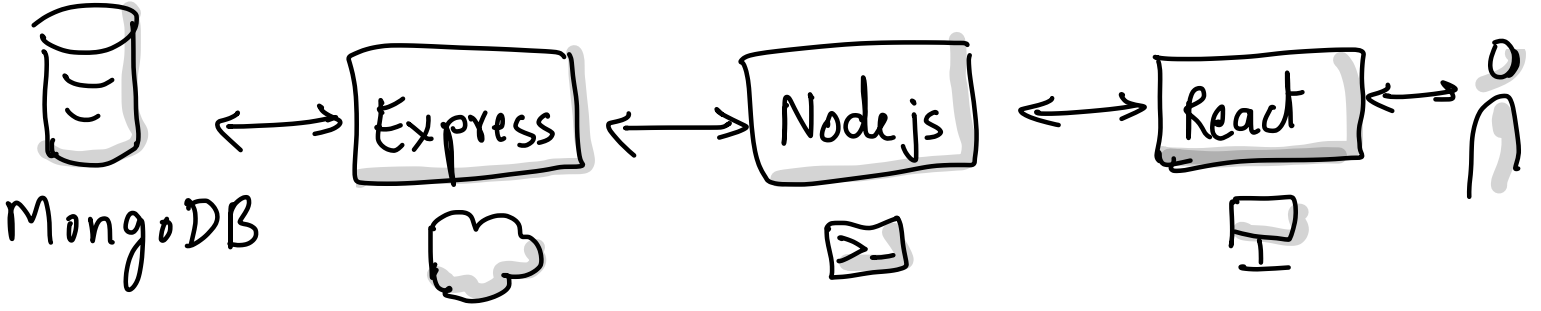
front end x



PHP	Programming
MySQL	DB
APACHE	HTTP Server
LINUX	OS


MEAN
MongoDB
Express
Node.js
Angular
USER
Simplifies
Request
Website
DB

MERN

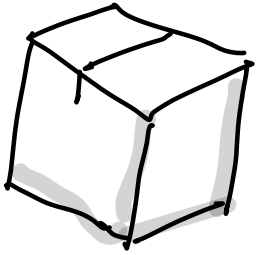


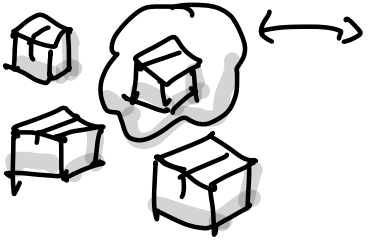
METEOR V/S MEAN

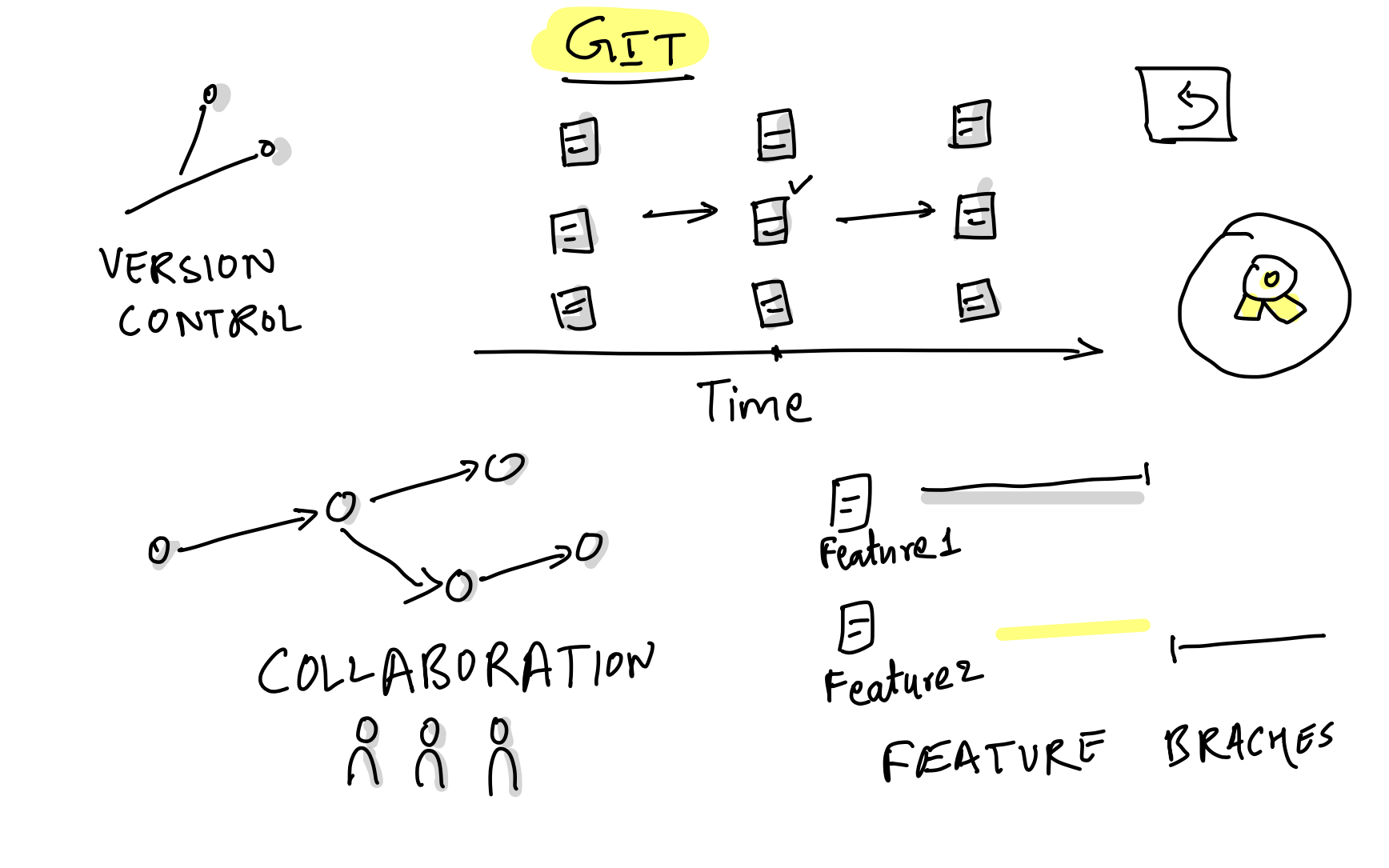
REALTIVITY

x

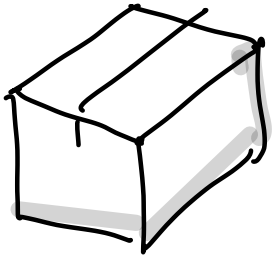
CUSTOMIZABLE



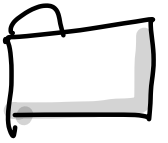



GIT
VERSION CONTROL
Time
COLLABORATION
Feature1
Feature2
FEATURE BRACHES

JARGON

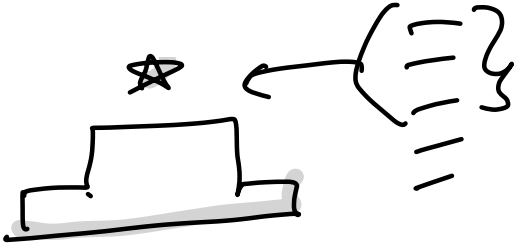


REPOSITORY



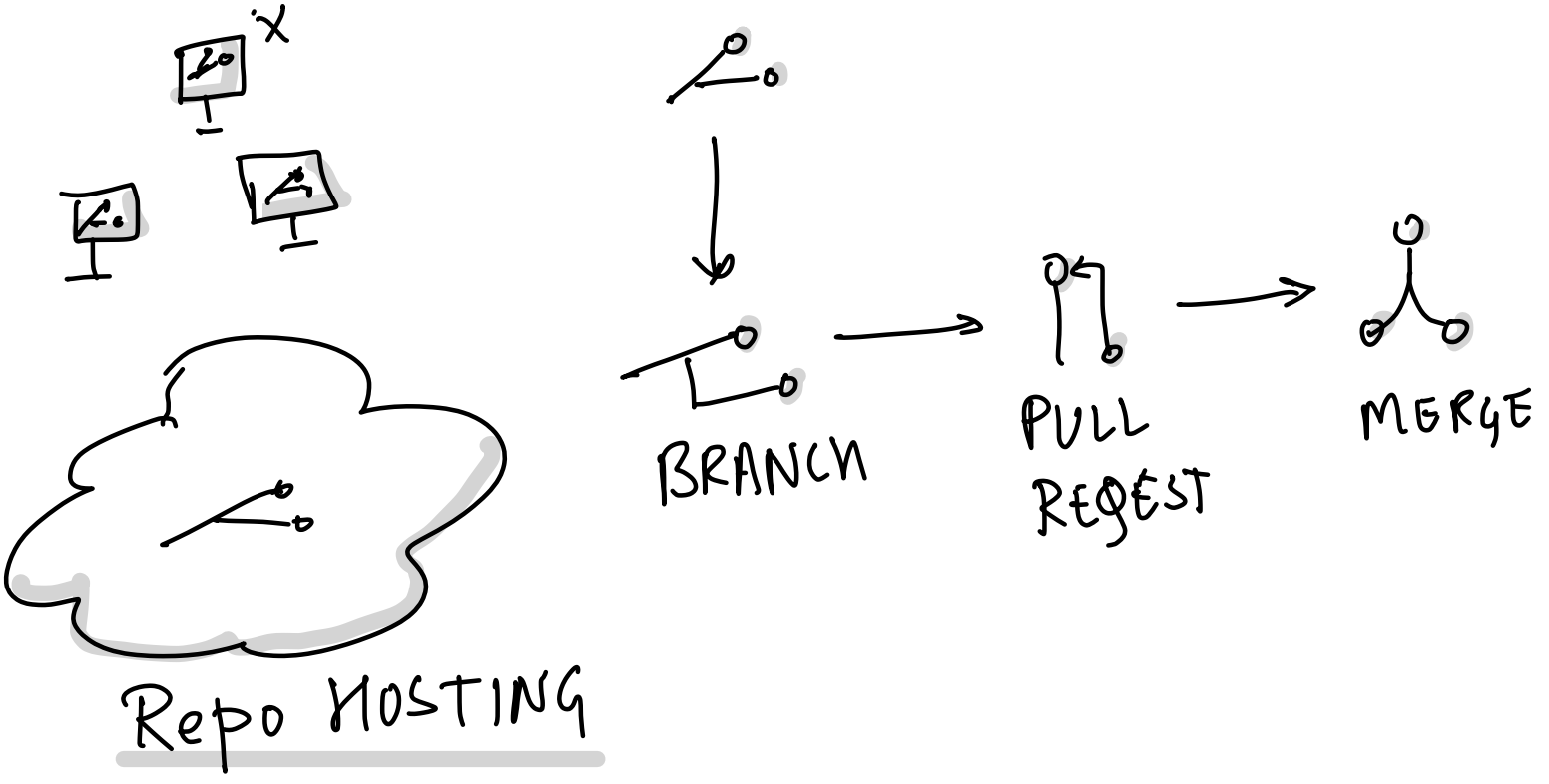
WORKING DIRECTORY

COMMIT

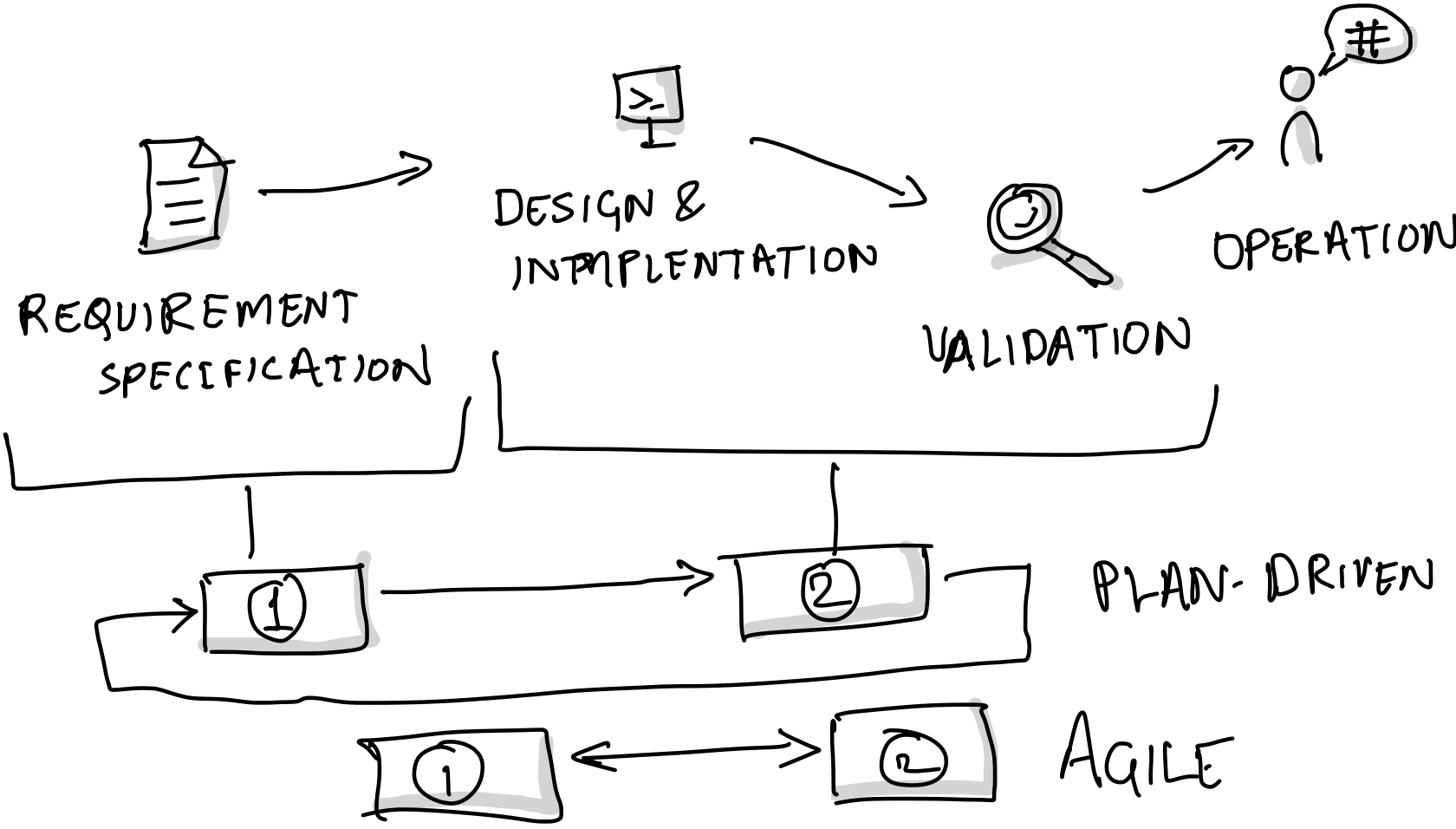


STAGING

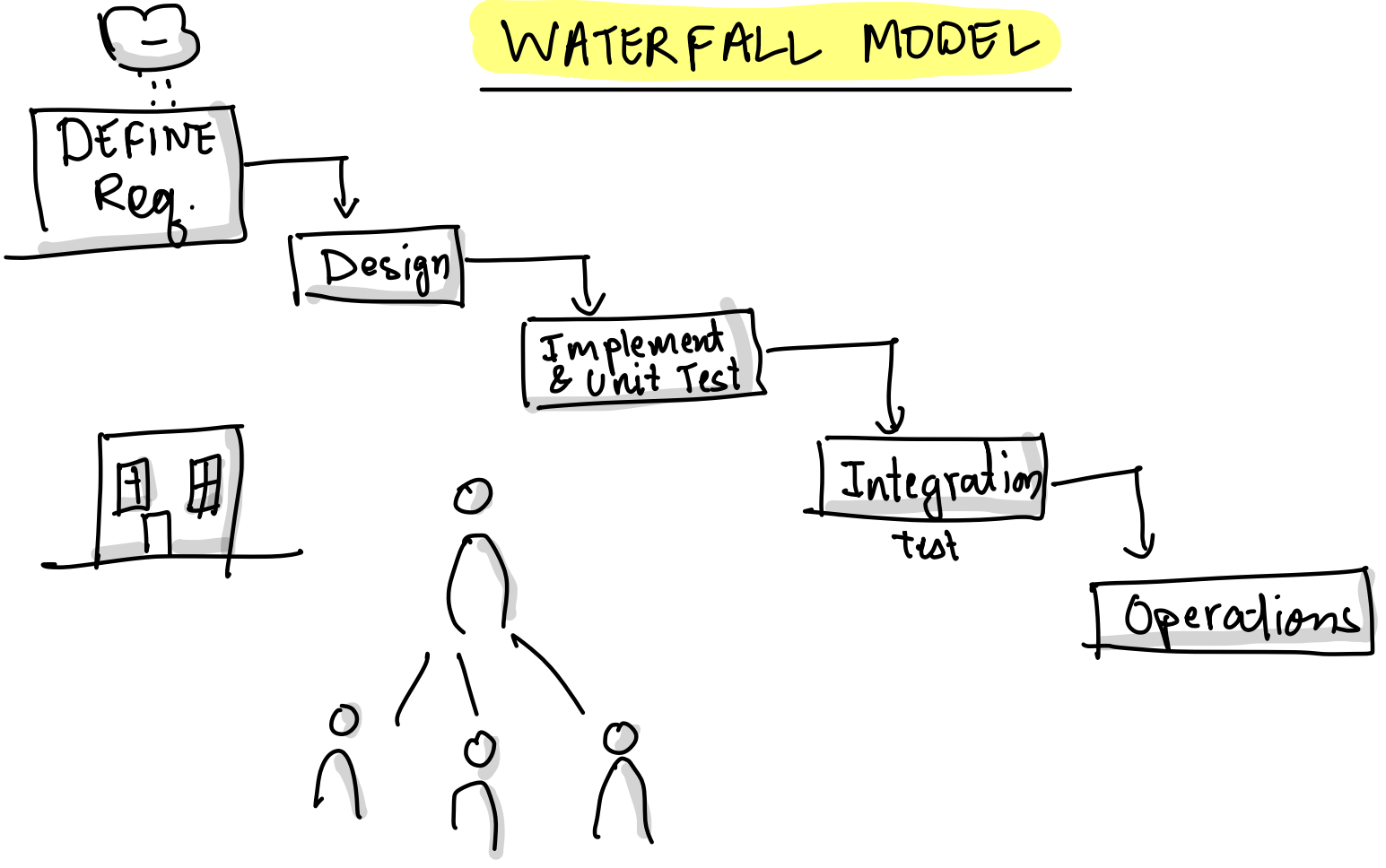
GITHUB



SOFTWARE PROCESSES

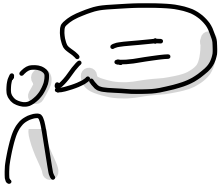


WATERFALL MODEL



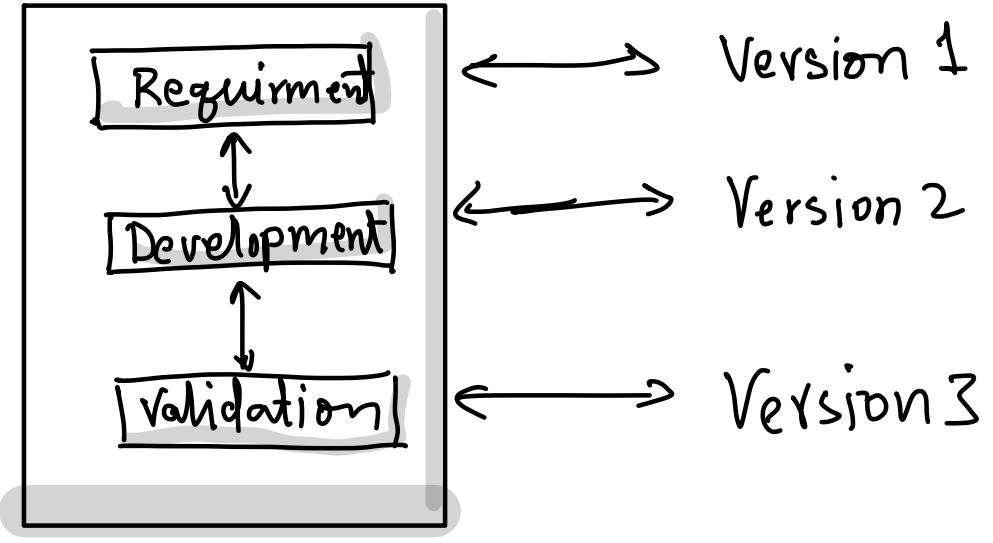
INCREMENTAL DEVELOPMENT

1.

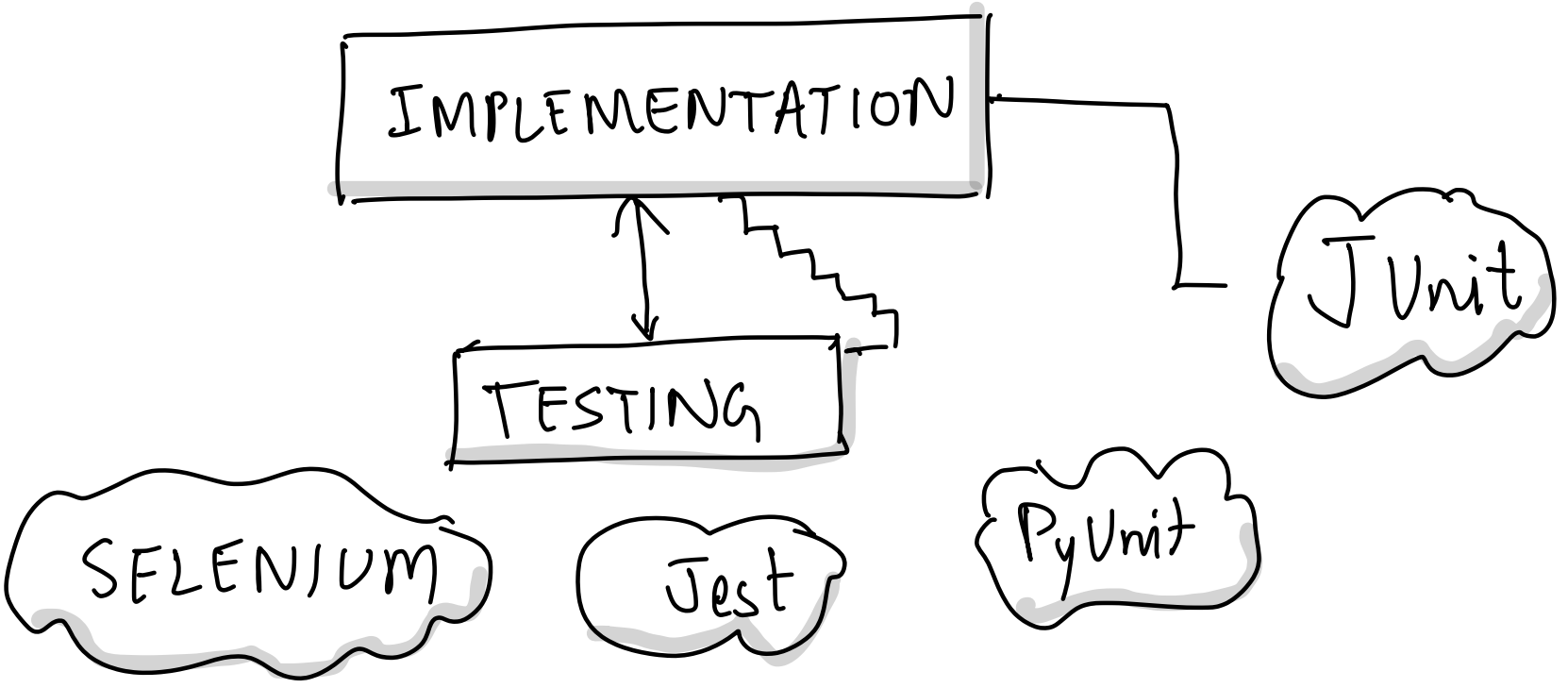


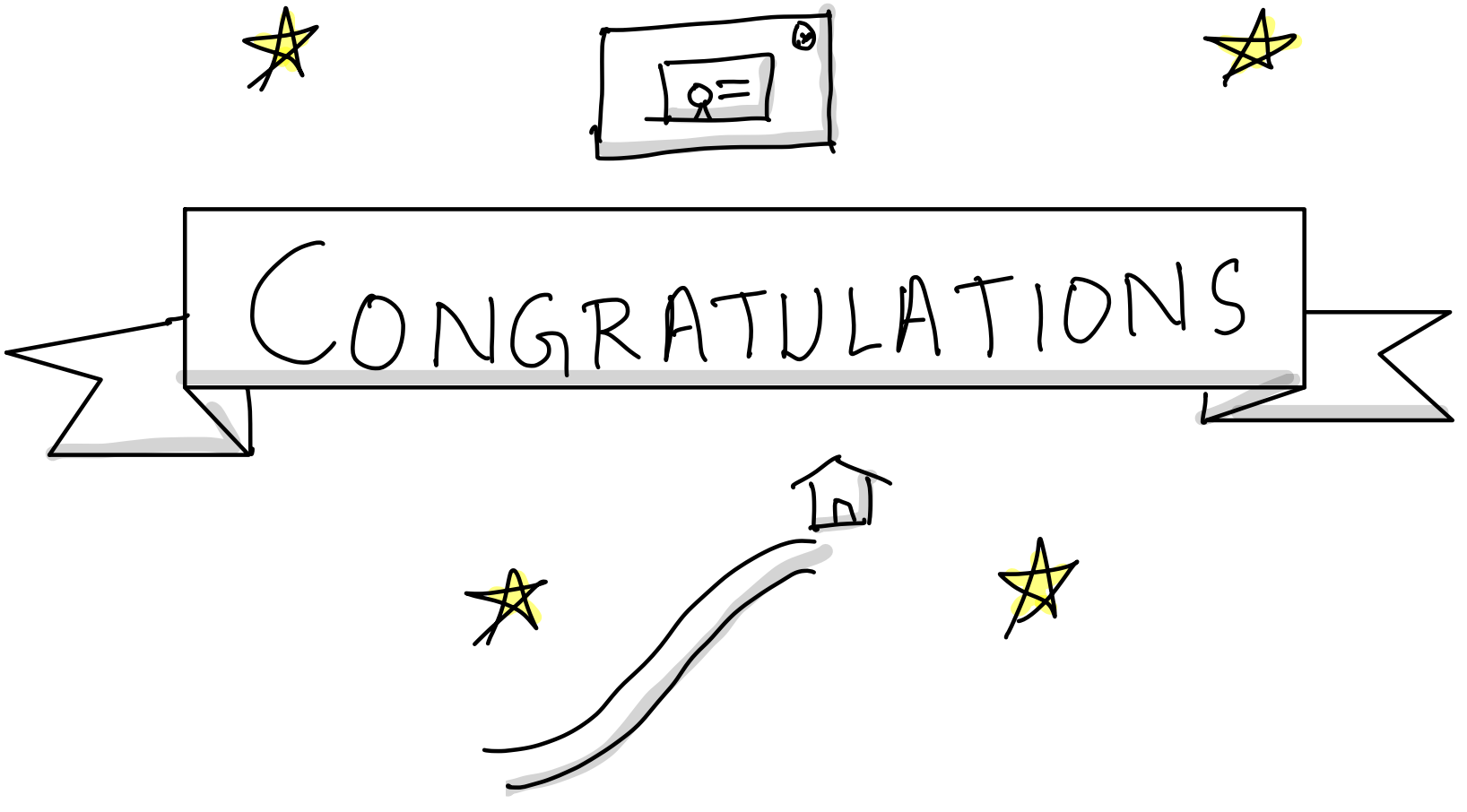
2.

3.



TEST DRIVEN DEVELOPMENT




CONGRATULATIONS