client side
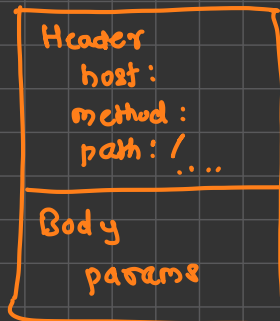## frontend
↳ html / css / JS
↳ static web pages
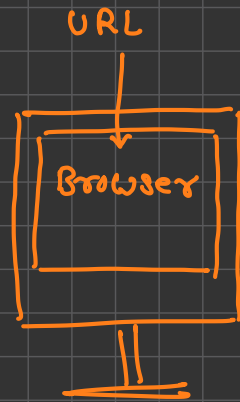
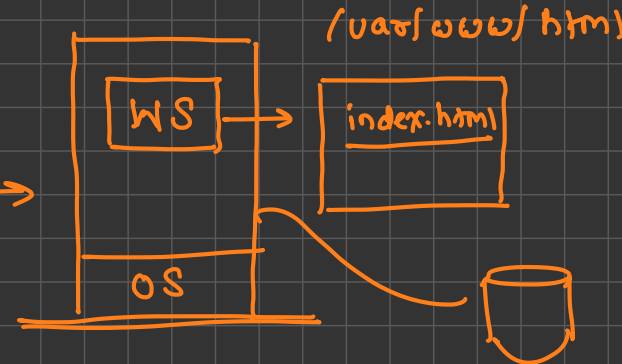# serror side
backend
↳ Java / c# / Python / PHP / JS
↳ dynamic web pages

Web Architecture
client - server

URL

```
┌─────────────┐
│  ┌───────┐  │
│  │Browser│  │
│  └───────┘  │
└─────────────┘
      │
```

Client

(var/www/html)

```
┌──────────────┐     ┌──────────────┐
│ ┌────┐       │     │ index.html   │
│ │ WS │ ───→  │ ──→ │ ‾‾‾‾‾‾‾‾‾‾   │
│ └────┘       │     └──────────────┘
│              │
│ OS           │
└──────────────┘
```

NIW

Server
15.2.5.6
mydomain.com

database

```
┌──────────────┐
│ Header       │
│   host :     │
│   method :   │
│   path : /   │
│       ....   │
├──────────────┤
│ Body         │
│     params   │
└──────────────┘
```
Request

```
┌──────────────┐
│ Header       │
│   serwer : ─ │
│ content type:─│
│   status:    │
├──────────────┤
│ Body         │
│    contents  │
└──────────────┘
```
Response

Browser → client, html/css + JS parser/interpreter

Components

- **Network** -

    - Network communication
    - creates & sends request to server
    → receives response from server

- **Layout Engine / Rendering Engine**

    - responsible for converting html/css to JS object
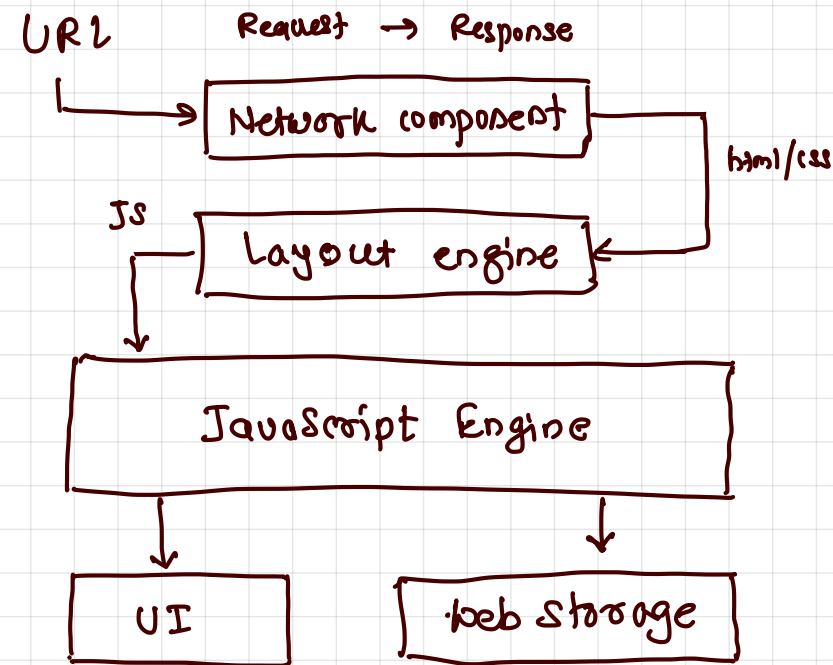
- **JS Engine**

    - responsible for executing JS code
    - heart of any browser

- **User Interface component**

    - showing the o/p to the user

→ **Web storage**

    - maintaining history / client side data
    → Session storage / local storage

URL   Request → Response

```
        ┌──────────────────┐
URL ───→│ Network component │────┐
        └──────────────────┘    │ html/css
   JS        ┌──────────────┐   │
      ┌─────→│ Layout engine │←──┘
      │      └──────────────┘
      ↓
 ┌──────────────────────────────────┐
 │      JavaScript Engine            │
 └──────────────────────────────────┘
      │                    │
      ↓                    ↓
  ┌──────┐          ┌────────────┐
  │  UI  │          │ web storage │
  └──────┘          └────────────┘
```

	Rendering Engine	JavaScript Engine
Internet Explorer / Edge	EdgeHTML	Chakra
Opera	same as chrome	same as chrome
Firefox	Gecko	Spider Monkey
Safari	—	Nitro / JS one
Chrome	Blink	V8

c, c++

* Server ⟳ process

→ s/w or app which runs continuously to serve some requests

→ **types**

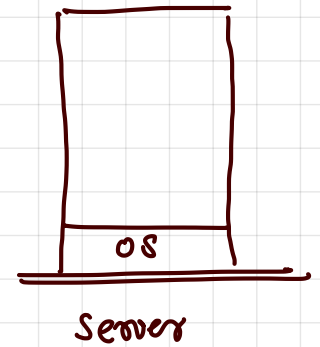→ web server → used to host web sites, eg. apache, nginx, IIS

+ database server
  → RDBMS → MySQL, MariaDB, Oracle, MS-SQL server, postgreSQL
  → NoSQL → MongoDB, Cockroach DB, firebase, couch DB, redis, ...

→ mail server → postfix, Dovecot, sendmail etc

→ file server → FTP, Samba (SMB), NFS

→ **Development stack (web)**

→ collection of multiple servers/ programs etc

→ platform + web server + database + user interface framework

eg ① AMP → Apache + MySQL + PHP/Perl/Python
   → LAMP (Linux), MAMP (macOS), WAMP (windows)

② WISA → Windows + IIS + SQL server + ASP.net
③ MERN → MongoDB + Express + React + NodeJS
④ MEAN → MongoDB + Express + Angular + NodeJS

OS

Server

MERN

NodeJS → Ryan Dahl
2009

server side   client side

platform → NodeJS

Express   driver   DB   React

http buffer https FS os

NodeJS

Operating System

Hardware

100% JS

program.js

http   https
FS   NodeJS   buffers
          os

50% JS
50% C++

30% JS
70% C++

V8   libuv   100% C++

Operating System