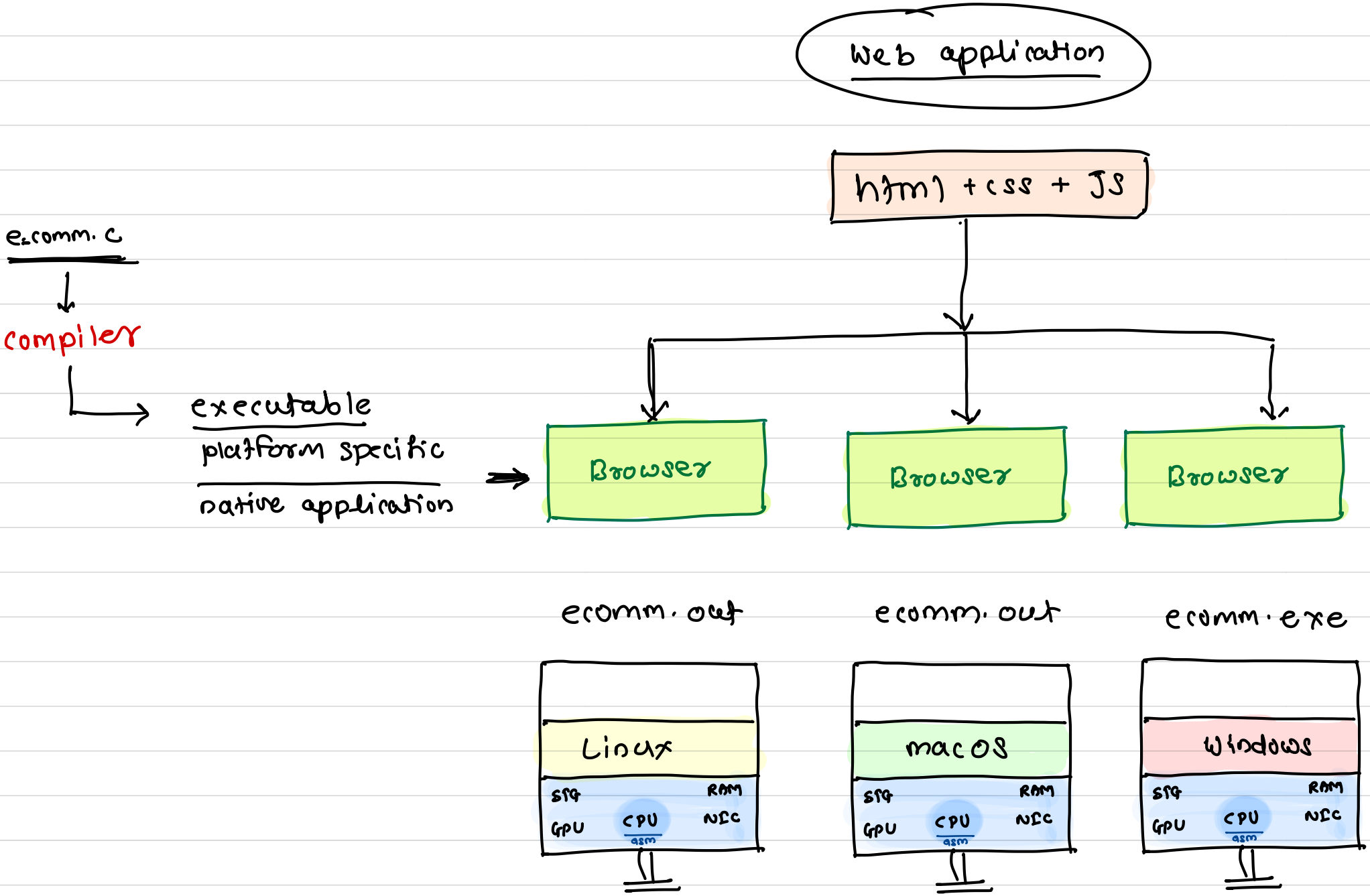
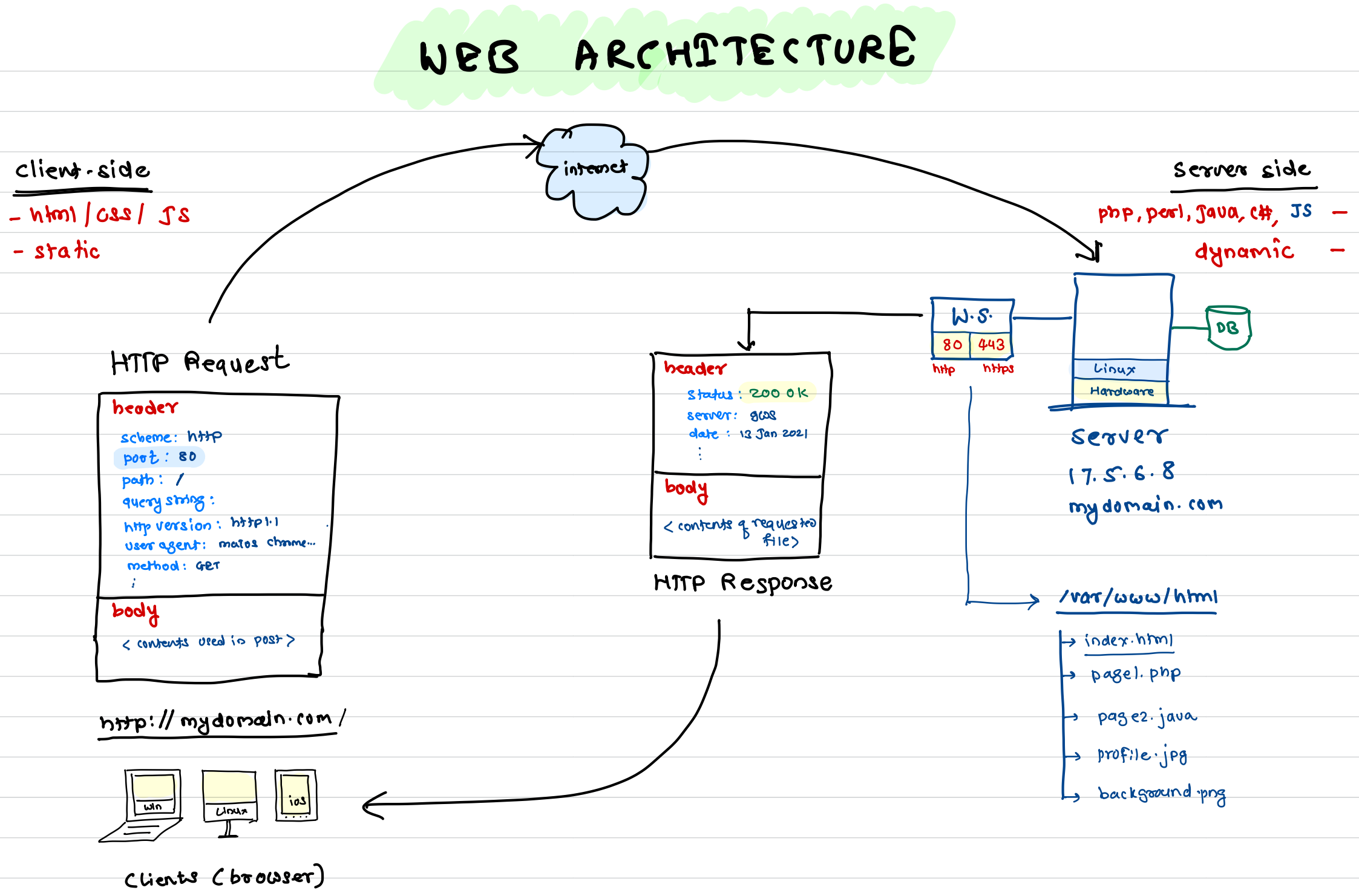

ecomm.c
compiler
executable
platform specific
native application
web application
html + css + JS
Browser
Browser
Browser
ecomm.out
ecomm.out
ecomm.exe
Linux
macOS
Windows
SIG
GPU
CPU
RAM
NIC

WEB ARCHITECTURE



Server

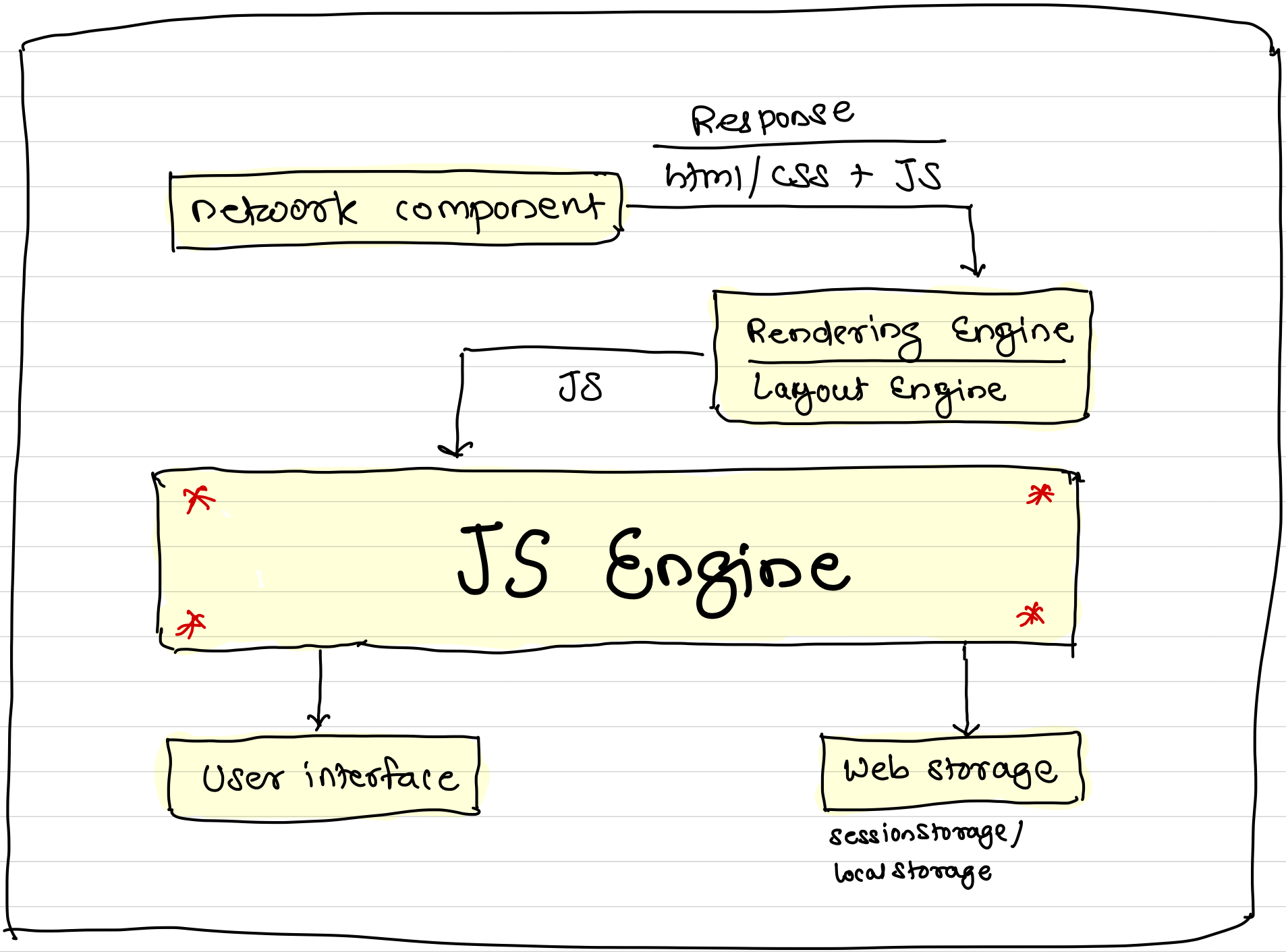
- machine used to serve requests
- types
  - web server
    - server used to serve http requests
    - eg. apache, IIS, Express
  - file server
    - used to share files within group of users
    - eg. NFS
  - database server
    - used to persist the data
    - types
      - RDBMS : MySQL, Oracle, SQL server, Postgres, IBM DB2 ...
      - NOSQL : MongoDB, CouchDB, Firebase ...
  - application server
    - used to serve application
    - eg. tomcat

* What is a browser?

- native application
- developed in C / C++
- OS dependent
- understands protocols
- interpreter for html / css / JS
  - understands html / css
  - executes the JS code
  - contains parsers for different formats
    - parser for html / css
    - parser for xml
    - parser for text format

* JS Engine : used to execute JS code
  - chrome : V8 [C++]
  - safari : Nitro JS
  - firefox : spidermonkey
  - Edge : chakra
  - interpreter for JS

* Rendering engine : converts html/css to JS
  - chrome : Blink
  - safari : webkit
  - firefox : Gecko
  - Edge : EdgeHTML


Response
html / CSS + JS
network component
Rendering Engine
Layout Engine
JS
JS Engine
User interface
Web storage
sessionStorage /
local storage

* What is URL ?

- Uniform Resource Locator
- Locator : used to locate
- Resource : any file hosted on server

http://mydomain.com

https://google.co.in

http://192.168.2.3

http://mydomain.com:8080

http://mydomain.com/mypage.html

http://mydomain.com/mypage.html?name=abc&address=pune

http://mydomain.com/mypage.html#top

- Uniform
  - irrespective of browser / OS URL remains same
  - components
    * scheme
      - protocol used to communicate with server
      - eg. http:, https, file, ftp .. ..

    * domain name or IP address
      - used to identify server machine uniquely

    * port number
      - port number of web server
      - if it is missing, by default
        - http: 80
        - https: 443

    * file name or path
      - resource to visit

* query string
  - a way to pass input to the page
  - it is always in the format :
    ? <key> = <value>

  - eg.
    ? name = abc & address = pune

* hash component (proxy)
  - used for linking multiple sections within a page

* what is status code in response object

- code represents the status of processing request
- types

  - 1xx : used to represent debugging / informational messages
    - 101 : switching protocol

  - 2xx : represents success
    - 200 : OK 201 : created

  - 3xx : represents redirection
    - 300 : temporary redirection

  - 4xx : represents client error
    - 400 : Bad request 401 : unauthorized 403 : forbidden 404 : not found

  - 5xx : represents server error
    - 500 : Internal server error

* framework (stack) ⇒ **MEAN / MERN**

① platform : NodeJS

② Database : MySQL + Mongo

③ client side framework : React / Angular

④ Web server : Express

⑤ OS : macOS / Linux / Windows

⑥ Language : JS