

04/02/23 /

→ Making the most of the bootcamp

→ watch lectures

→ offline review

→ Homework.

→ Discussions on Discord.

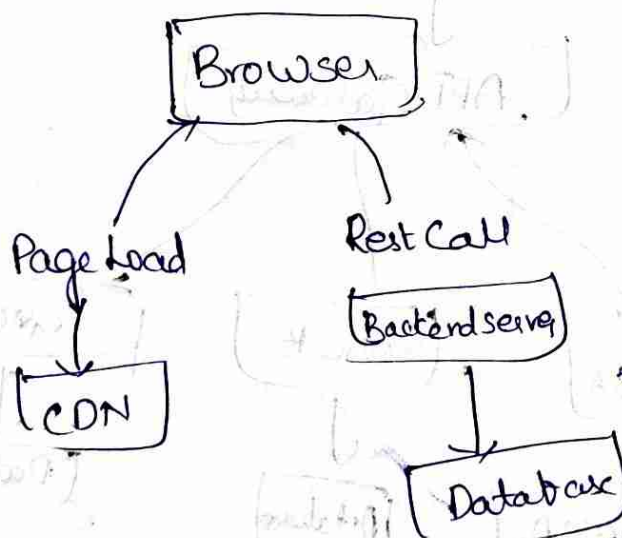
Notes Make Notes

→ Interactive

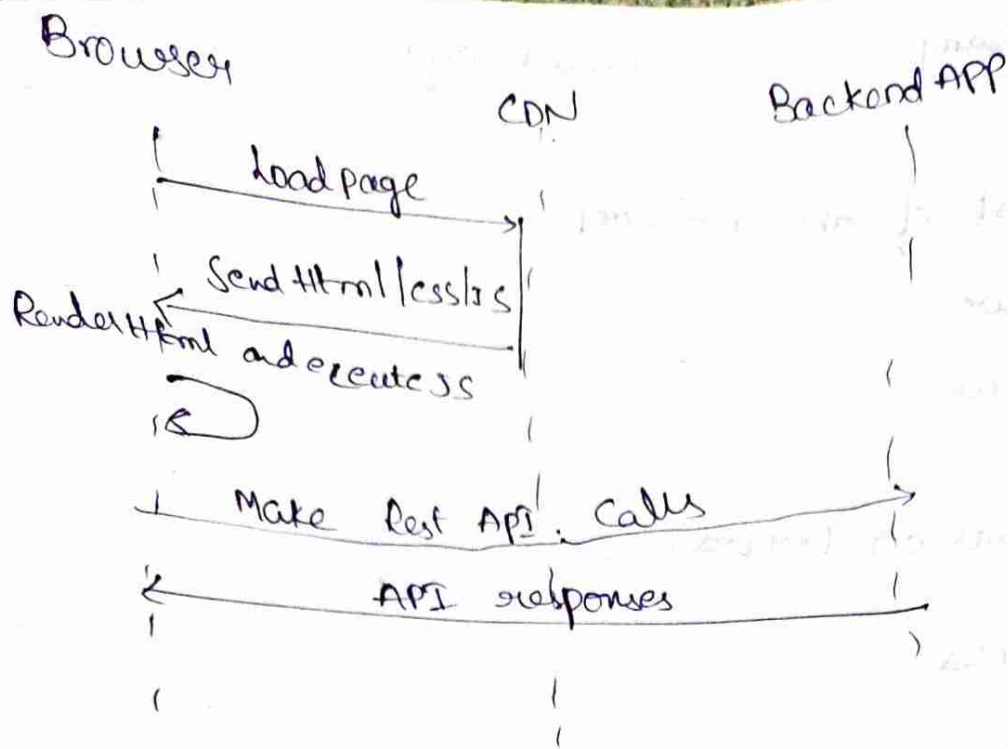
(opportunities to interact).

→ Most often Customers are other developers.

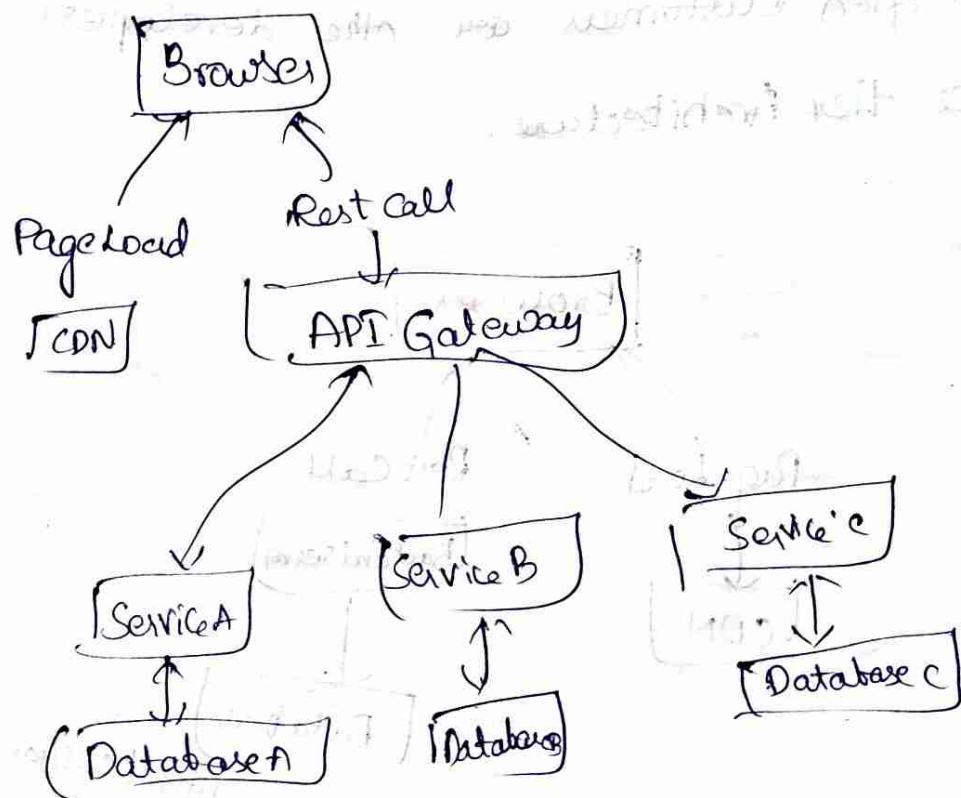
→ Three tier Architecture.



→ Browser makes call to CDN → which has some JavaScript embedded which will call Rest API's.



Microservice Architecture:-



CDN :-

Distributed servers that deliver web content to users based on their geographic location.

→ stored on multiple servers in various locations

How = does a CDN work.

→ origin server

→ Edge Server

→ user

↳ When a user visits a static website for the first time, a CDN Server will transfer the files from origin server.

Web Request

→ The browser sends a request to the DNS server, to resolve the domain name of the URL to its corresponding IP address.

Google.com → IP address
12.34... (Random address)

→ The DNS server returns the IP address of the closest CDN edge server's to user location.

→ The CDN edge server checks its cache.

→ If found the cache, the CDN edge server returns the cached version.

→ If not found a call happens to origin server and store in cache before serving the user. (Next page).

Edge Locations: - Very close to users which serve the request (CDN) it is related to

Edge servers: - The first server which responds to the request from server.
(Eg:- API Gateway)

AWS CDN Service: CloudFront.

A web page

→ HTML page

→ CSS assets

→ Image

→ JS assets

→ Fonts

Prerpage

→ The origin server returns the requested content to the CDN Edge Server.

→ The CDN edge server stores it in cache.

→ The CDN edge server returns the content to the browser.

REST request (Other such as GraphQL)

- Javascript Code Constructs an HTTP request.
- Adds endpoint API URL and HTTP method.
- Include necessary parameters or data.
- Servers gets request and returns response.
- Javascript receives response and processes it.
- Response manifest as a UI change.

A Capable backend

- Intercept requests.
- Extract Data from requests.
- process the request
- Pull necessary data from database
- Process the data and prepare response.
- Return response.

Handling Requests

HTTP :- (HyperText Transfer protocol)

↳ A message sent by a client (such as a web browser) to a server to request specific information.

HTTP request

- Request line (GET/index.html HTTP/1.1)
- ~~Req~~ Headers :- META info about the request.
- Body :- Data to be sent with the request.
- HTTP requests and we send HTTP response.
- Status Line :- First line eg. HTTP/1.1 200 OK
- 400 series → Error in the request from Client side
- 500 series → Server side Error.

HTTP

↳ Stateless Protocol :-

Cookies

- ↳ Allows multiple requests to be tied together.
- ↳ Server creates a cookie. while sending the response it sends a set cookie.
- This cookie is sent everytime when user sends the request.

Useful :- With server sessions

→ login flow

- user enters login credentials into a web form.
- form data is submitted using HTTP post.
- Server verifies the credentials.
- Server generates new session for the user and creates unique
- Server sends a response with SetCookie header
- Cookie Approach
 - Browser sends the cookie back to server in cookie header
 - Server looks up session info based on session ID.
 - Server generates personalized response
- Automatic
 - Browser memory.
 - Local and Session storage.

What we need

- A programming language. (eg Java)
- A Server Runtime. (eg: Jetty, Tomcat)
- A framework to handle common concerns (eg: spring boot)
- A framework to interact with the database (eg: JPA)
- A framework to handle security. (eg: spring security)
- A framework to manage infrastructure (microservices).

Source Control:

Git: A Version Control System for tracking changes in files and Coordinating work among multiple people.

- keeps track of different versions of your code.
- Collaborate with others.
- Revert back to previous versions if needed.
- Resolve conflict when multiple people work on the same code.

GitHub

- a web-based platform that provides hosting for Git repositories.

Additional features

pull requests

- issues
- wikis
- project management.
- actions
- discussions.

Open Source on GitHub

- Pull request workflow
- Fork the repository
- clone the repository
- create a new branch
- Make changes

→ push the changes

→ Create a pull request

→ Review and discuss

→ Merge or decline.

→ update and delete the branch