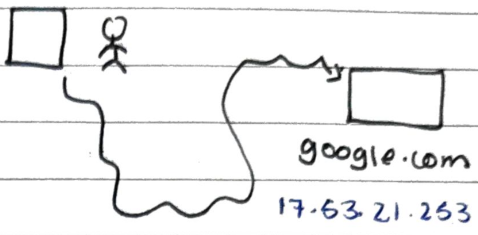
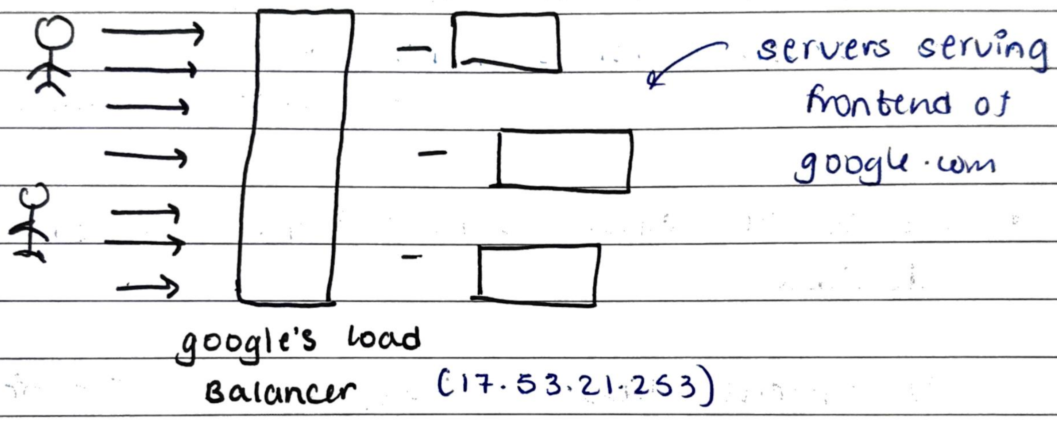
* How DNS works *

To connect to a machine, you need its IP address But how do we discover



google.com ⟶ 17.53.21.253



So, there would be a place where the mapping is stored :

www.google.com ⟶ 17.53.21.253
⋮

along with other records like gemini.gv → gemini.es

__/__/____

www.google.~~com~~ → 17.53.21.~~2~~53
bard.google.com → _ _ _ _
⋮
Mx, Txt, etc

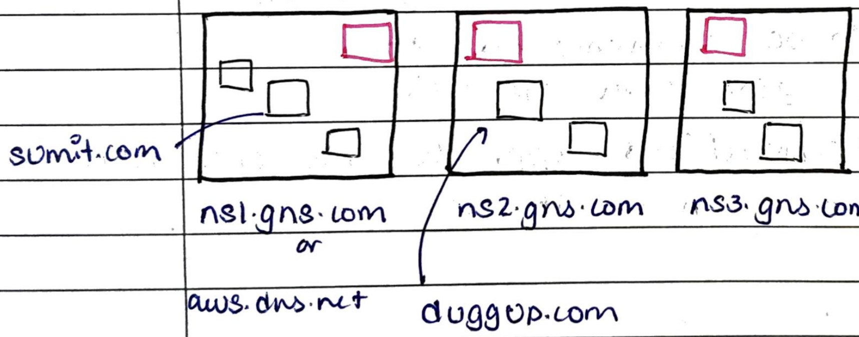
Zone
google.com

All DNS records of a domain are part of zone (logical entry)

→ what we configure on Route53, GoDaddy etc.

* Authoritative Name servers (zones are served via)

An authoritative name server hosts multiples zones, and it answer DNS questions for the zones it ~~co~~ occurs.

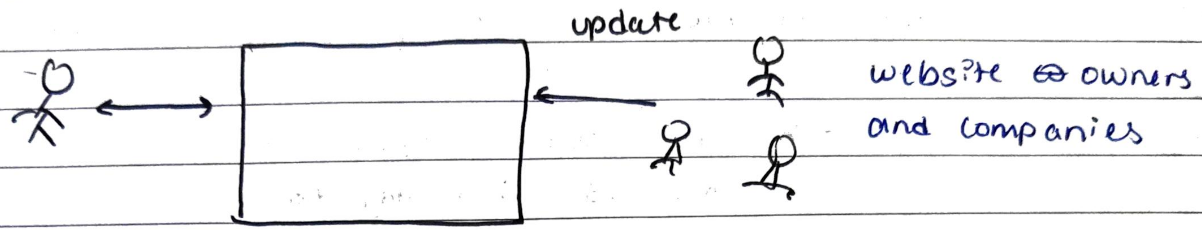


Zones are served from authroritutive NS

Typically offered by cloud providers or ~~god~~ godaddy, etc

Now, how would we reach to these servers?

Option1: let there be a single massive "system" that everybody reaches out to when they want this information.



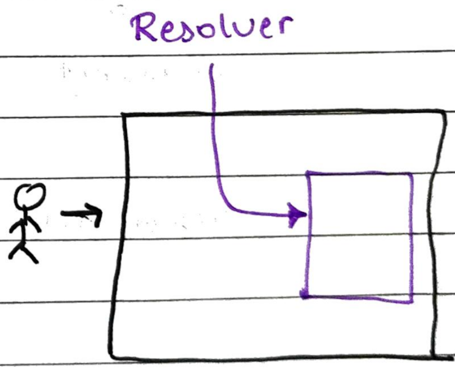
This centralised way is not scalable, fault tolerant, or even

concers: volume of data, number of requests and updates + any change in any info needs to be communicated to (world has gone with option 2)

Option 2: Decentralized approch - no one machine ~~Knows~~ Knows it all.

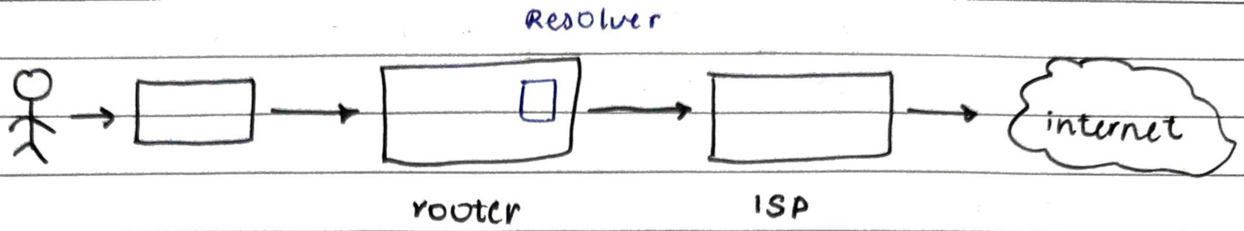
- DNS Resolver.

DNS resolver is a server that carrier out the resolution of Domain Name to IP address



* typically runs at isp. but you can run your own locally. This can be your internet router

* most home routers are real DNS resolvers.



$ ipconfig/all ~~fr~~ from your windows CMD

Two popular DNS resolvers are

Google : 8.8.8.8 Cloudflare : 1.1.1.1

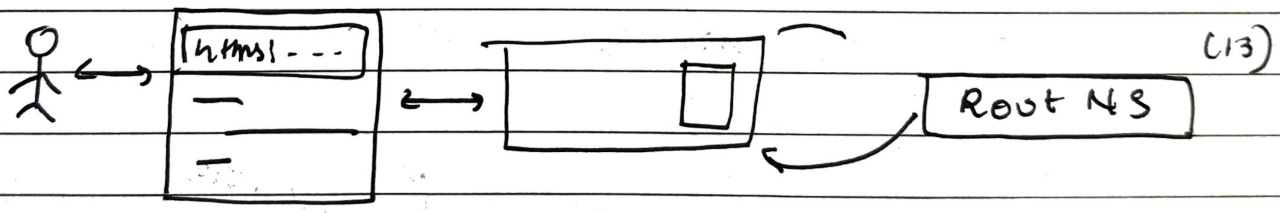
* SO, what exactly happens during resolution?

* How DNS resolution works?

say, we are looking for www.google.com

→ we need to reach it authouritative Name server

But we do not know where it is !!



Root Nameservers :

Total 13 root NS in the world, verisign ← a.root-server.net

fixed IP address USC ← b.root-servers.net

m.root-ser.net

* this ~~data~~ does not mean there are 13 servers

Root servers from single operator are distributed across the world and ~~&~~ leverage anycast ~~ea~~ (advertising same IP)

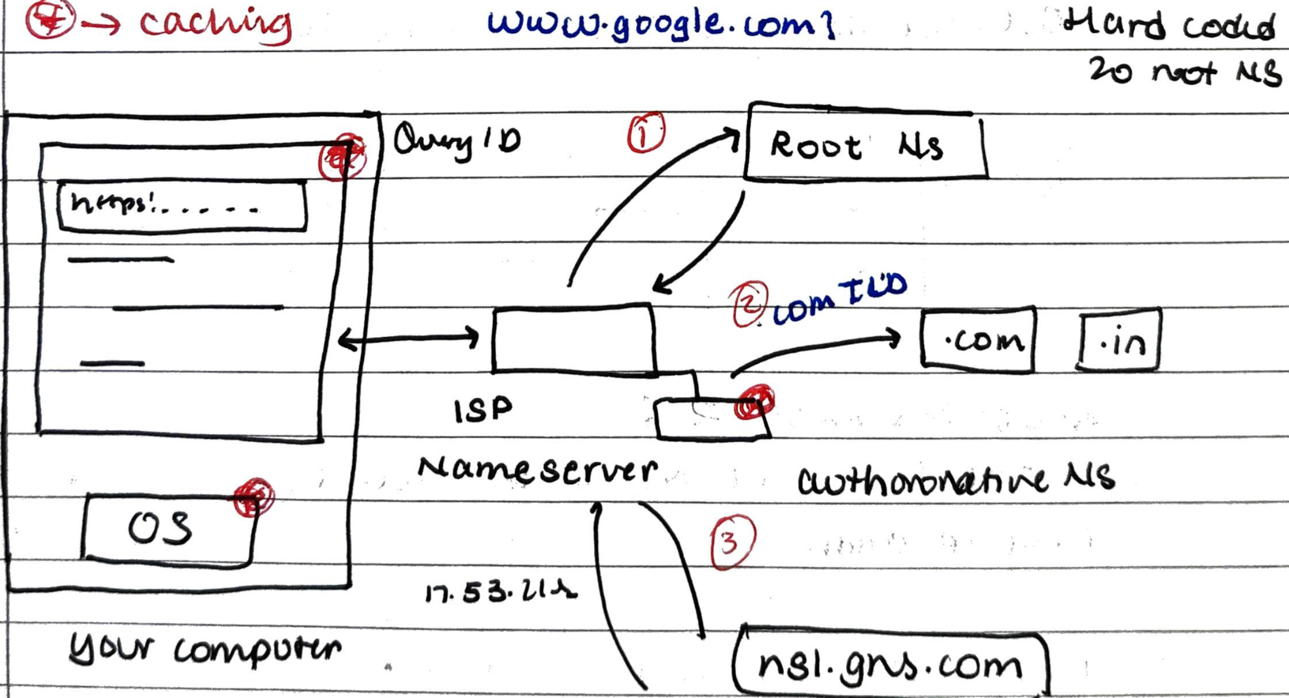
This way, incoming request connect to nearest Root Name server

when queried for domain name, it responds with IP address of some TDL (e.g .com, .in . edu, eh)

* calls to Root NS are infrequent, because even IP of TLD server does not changes often, so is heavily cached.
(you might be thinking every time we do this google.com this all process happens any, no!)

* each machine takes us closer to machine that knows it.

(4) → caching



* cached for a few hours

This recursion continues from Google Name server to Authoritative Name server at domain zone google.com } where zone are served

then it checks the record against 'www', which may point to IB DNS and it then resolves to IP of load balancer.

Then the browser connects to the machine & finds the request.