

## High Level Design

### Low Level Design

- writing good code

- scale

# of lines of code	2 Billion
complexity of code	
# of modules / file	20 Million
# of devs in team	10,000

### Scale w.r.t HLD

# users	100 → 2 Billion
Rps (Requests per second)	1000 → 10 million R/s
amount of data	few GBs → 100 PB

HLD

challenges @ scale  
& solutions

what we will NOT cover

- Dev Ops
- Cloud Tech - Aws/GCP/Azure
- Specific Technologies
- not about knowledge (about problem solving)

hands on  
for all this  
will be in  
the project  
module (backend / data engg)

HLD - Case studies

# Staff Engineer @ Google

India → upto 3 crons (Staff / Architect / Principal Eng)

US → 600K \$

## TLD for Staff Eng @ Google - Interview Question

given a list of strings sort them alphabetically.

Catch there is around 50 PB of data  
50,000,000 GB

1 bit

8 bits - byte

1000 b - Kb  $10^3$

1000 Kb - Mb  $10^6$

1000 Mb - Gb  $10^9$

1000 Gb - Tb  $10^{12}$

1000 Tb - Pb  $10^{15}$

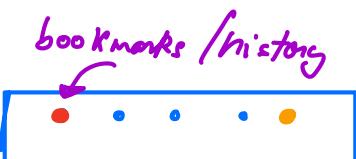
# Case Study - 1 - [://delicious/](http://delicious.com)

Joshua / 2003

youtube 2004

Alic 2006

Chrome 2008



Cyber Cafe

Bookmarking website - so that bookmarks can be accessed from any device

## Feature

Minimal Viable Product (MVP)

Proof of concept (PoC)  
demo version

v0

- identify user (account creation & login)
- User can add bookmarks  
*add Bookmark (user-id, URL)*
- User can view their bookmarks  
*get Bookmarks (user-id) → [list of URLs]*

Python + Django / Java + Spring / Ruby + Rails / PHP + Laravel  
JS + Express / ...

Code it on your local machine



**ISP:** Internet Service Provider

Jio / Airtel

Hathaway / Hayagi / ACT

Comcast / Verizon

Google Fibre / Starlink

**IP Address**

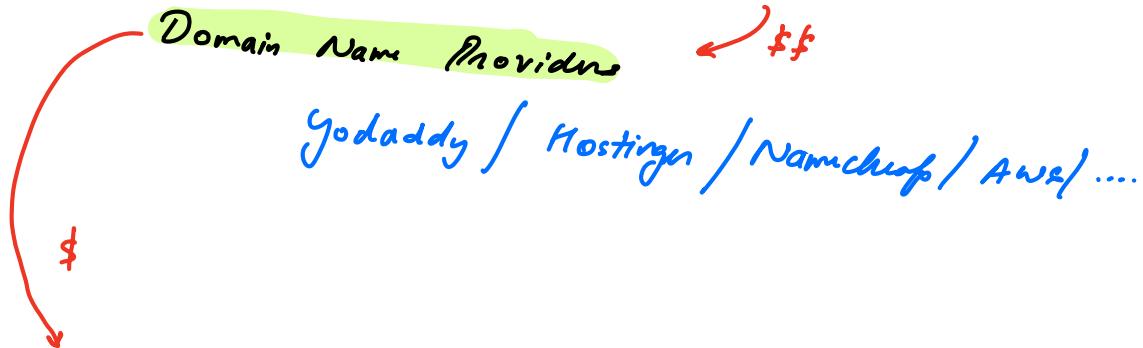
every device on the internet has an **IP address**

Public vs private

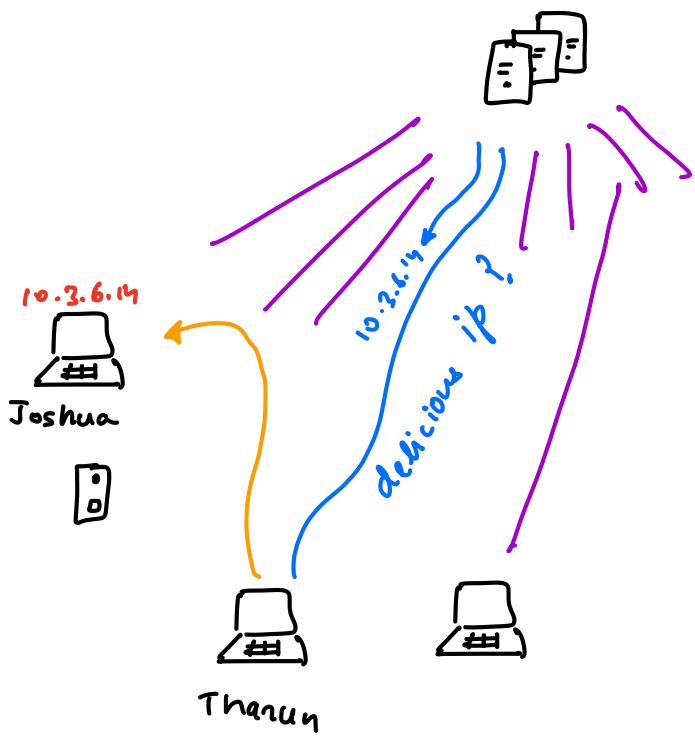
Static vs dynamic

# Domain Name Service (DNS)

① Joshua must purchase a domain name



## I CANN



Domain	IP Address
Scalen.com	1.2.3.4
google.com	4.2.2.1
delicious	10.3.6.14

World Population ≈ 8B  
# Internet users ≈ 5B  
# devices on internet  
≈ 100B - 1T

100 B devices ~~ICANN~~ → Internet goes down!!  
Crash! I CAN'T!

ICANN → single point of failure (spof)  
bottleneck

governments / google Amazon Netflix / Militaries / Records orgs)  
 edu institutions / ISP

depend on highly reliable internet

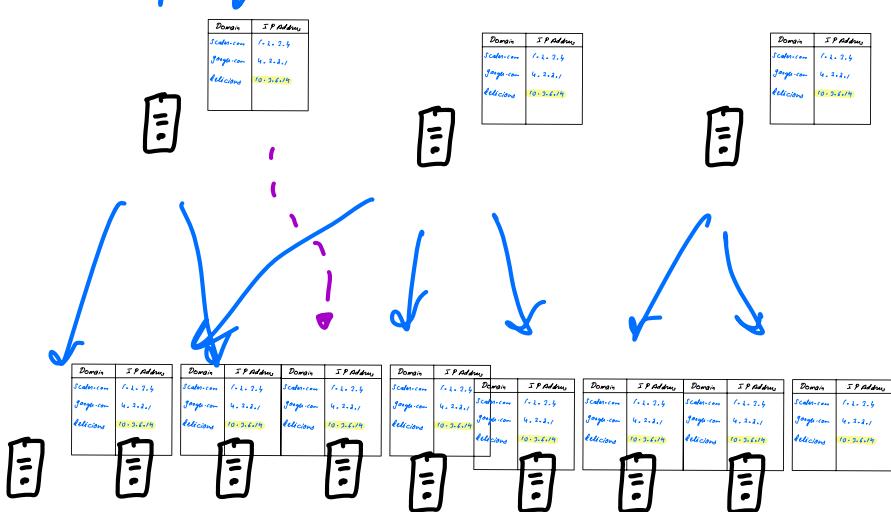
∴ set up copies of ICANN mapping  $\Rightarrow$  DNS servers

godaddy (Purchase)

ICANN

Periodic sync up

Domain	[IP Address]
Scaln.com	[1.2.3.4, 2.2.4.5]
google.com	[4.2.2.1]
delicious	[10.2.6.14]

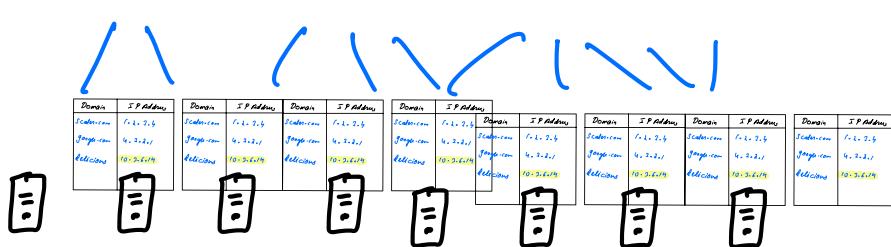


7 root DNS

servers

(Level 0)

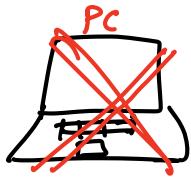
Level 1



Level 2

No spot  
no bottleneck!

## At present



webserver + database + OS etc.  
(app)

2007

35,000 Rs ( $< 500$ )  
CPU: 2.3 GHz 2 core  
RAM: 128 MB  
HDD: 90 GB  
N/W: 8 Kbps dial up

## Database

### User-bookmarks

user-id	URL
big int (8b)	random (1000) (10006) $\approx 1KB$

4 byte (72 bits)

Signed integer

$$= \begin{bmatrix} -2^{31} \dots 2^{31}-1 \end{bmatrix}$$

$$= \begin{bmatrix} -2 \text{ billion} \dots 2 \text{ billion} \end{bmatrix}$$

1 million bookmarks/day  $\rightarrow$  how much new data/day?

$$\frac{1 KB}{\text{bookmark}} * \frac{1 \text{ Million Bookmark}}{\text{day}} = \frac{1 \text{ million } 10^6 \times 10^3 \text{ KB}}{\text{day}} \Rightarrow 10^9$$

$$= \underline{\underline{1 GB / day}}$$

Q 9. how many days does the HDD get full?

$$\frac{40 \text{ GB}}{1 \text{ GB/day}} = 40 \text{ days}$$

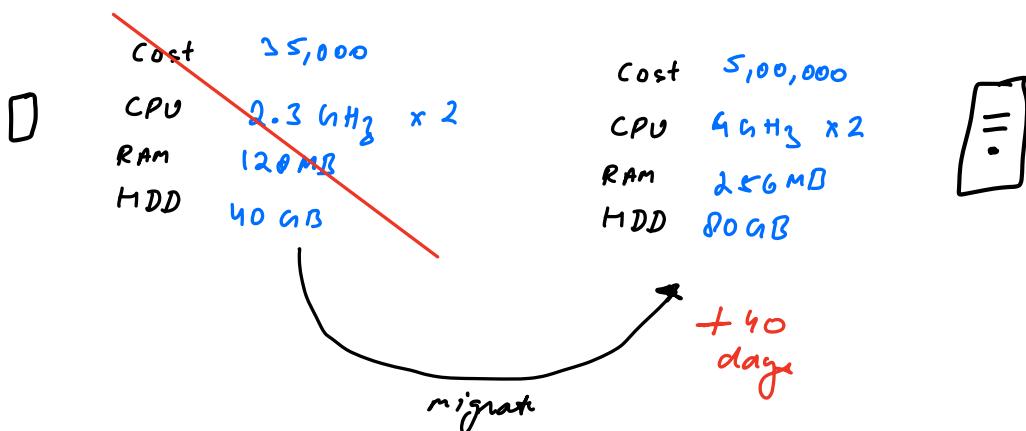
Joshua will run out of space in  $\leq 40$  days!!  
CPU/RAM/...

## Solutions

①

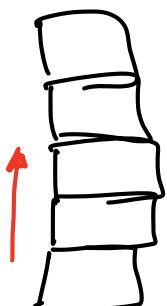
Vertical Scaling ↑ Scale up

Replace the current machine with a more powerful one



cost 2 crore  
CPU 2GHz x 10  
RAM 2 GB  
HDD 1 TB  
+ 3 years

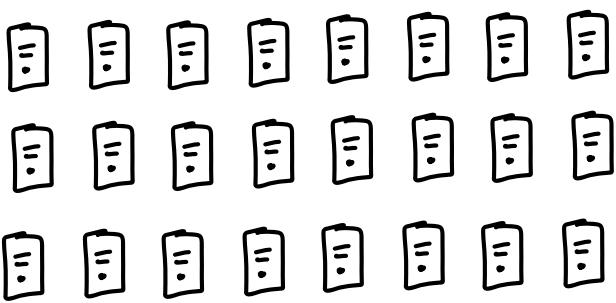
Build a custom supercomputer  
100 million \$  
5 years to build



Q

## Nonhorizontal Scaling

→ (Scale out)



1000 small servers

### Effective Config

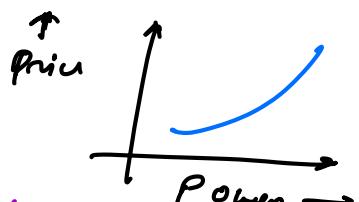
CPU	2.3 GHz	$\times$ 2000
RAM	128 GB	
HDD	40 TB	

Cost 2 crore

Google had ~ 10 million servers  
in 2020

## Vertical

- easy (throw money at the problem)
- limitations (after a certain point, it will stop working)
  - ∴ get limited by present day technology
- less cost effective
- computationally heavy



video rendering/  
machine learning (Chaos) / research / ...

## Horizontal

- no scale (for all practical purposes)
- more cost effective
- V.V.V.v.v.v... hand!!

HLD → what challenges arises → how to solve them

1.7 Trillion Parameter LLM (ChatGPT 4) ↗ 4M + /day

↙ 7TB of GPU & RAM

### Typical Server Config

(as of Aug 2024)

CPU 2-3 GHz × (1-8 cores)  
RAM 16GB - 32GB  
HDD 500GB - 10TB  
N/w 100 Mbps - 10 Gbps

### Max Server Config

On a single server

CPU 5GHz × (600 cores)  
RAM 12 TB  
HDD 2 PB  
N/w 10 Tb/s

baselines included framework  
(Django/Spring)

+ ORM + db Reads & writes +  
Auth + Middleware + logging

100 - 1000 reads/sec

10 - 100 writes/sec

bare metal framework

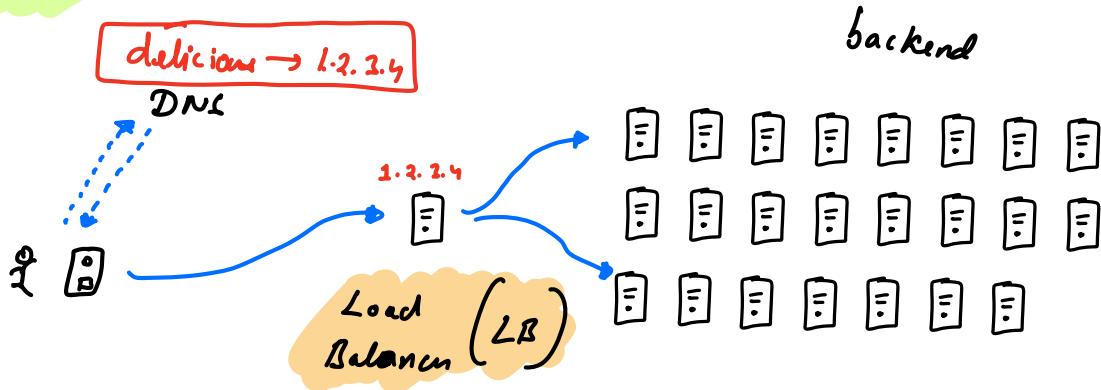
(C++ with Dragon/Rust with Actix web)  
+ simple requests

upto 200,000  
reads/sec

upto 10,000  
writes/sec

# Challenges with Horizontal Scaling

## ① DNS



## ② What if LB fails?

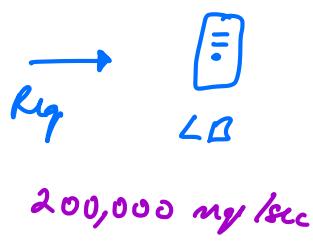
Solution : Have multiple LBs

## ③ Bottleneck

LB does not have to "process" requests  
it just forwards them

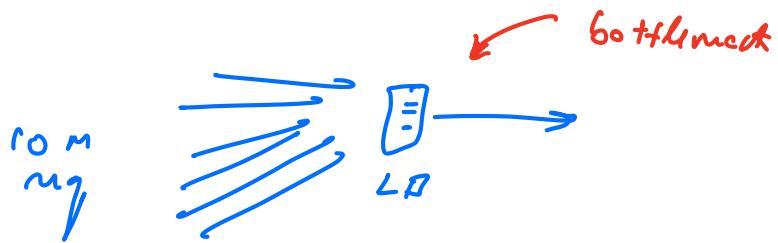


- ① decrypt & de-serialize request
- ② authentication & authorization
- ③ fetch db resources
- ④ process
- ⑤ generate response
- ⑥ logging
- ⑦ serialize & encrypt response



- ① look at headers
- ② figure out which machine to forward to
- ③ forward

still Q scale of Google (10 m user/sec)

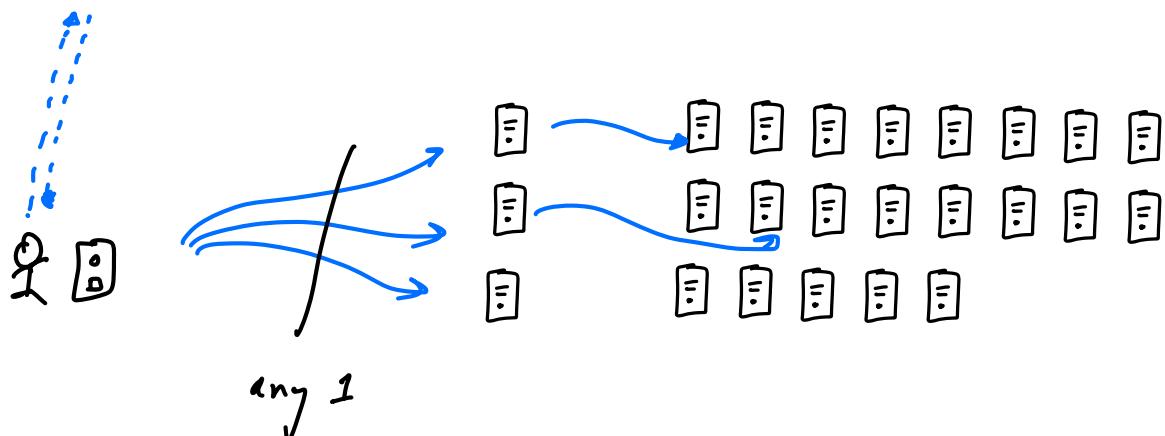


## ② SPOF

~~≥~~ ~~LB~~ website goes down.

has IP of all LBs

DNS



DNS acts as a LB in front of LBs

Note: DNS is a robust architecture

Next class → Load Balancing } Consistent  
                  & Sharding      Hashing

---

## Prerequisites for Mastering HLD

Note: it is okay to pick them up as you go

- ① Computer networks
- ② operating systems
- ③ Database
- ④ DSA

Entry Level

SDE 1

90% focus → DSA

8% focus → LLD + Projects + SQL  
(OOP)

2% → HLD / ...

Mid Level

SDE -2

80% DSA

10% LLD

(OO<sup>P</sup> / Design Patterns / ...)

Databases

(SQL + ACID + Normalization + Indexes)

5% Projects

5% HLD

Senior Levels

(SDE 2 & beyond)

DSA (is assumed)

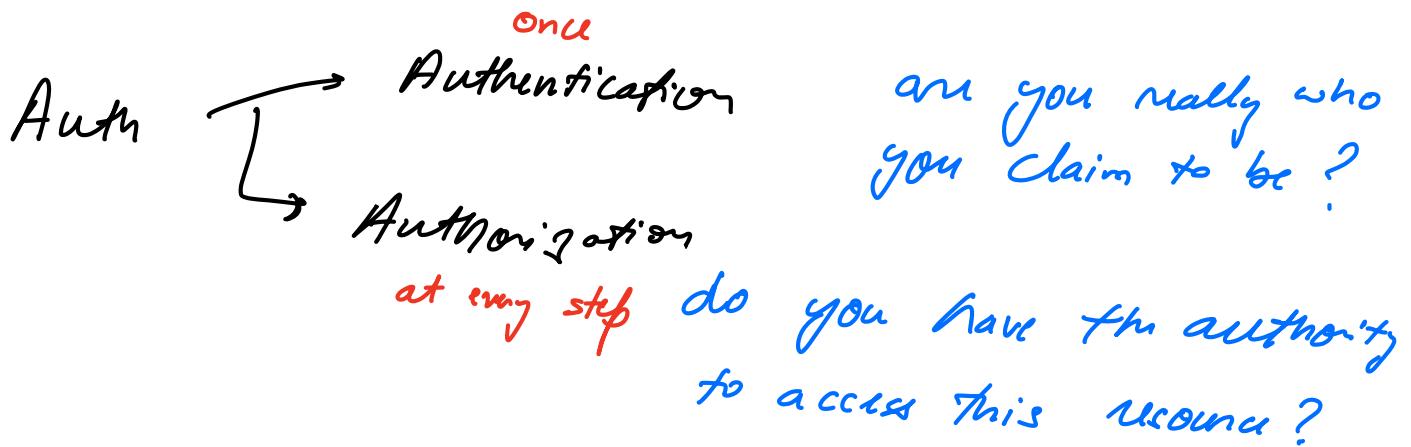
DSA will still be your  
first interview

LLD / Projects / HLD / Databases

equal weightage.

Imposter Syndrome

# Large Language Models (LLMs)

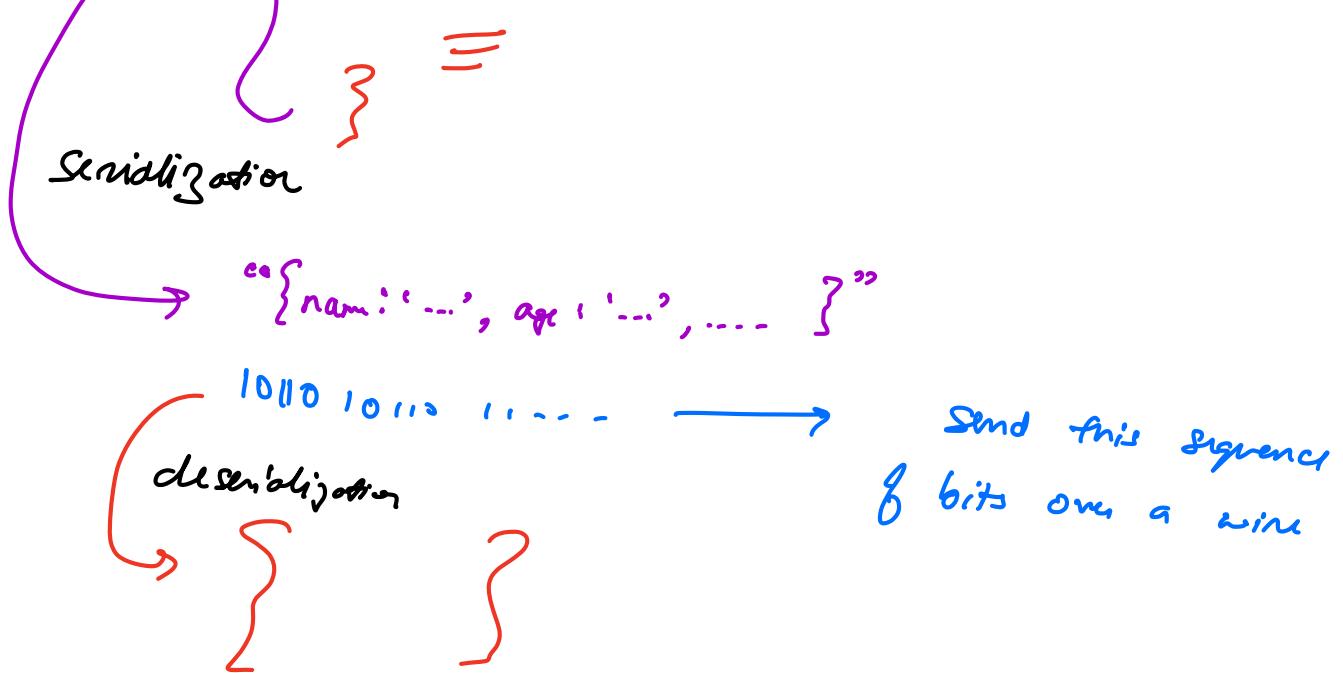


I'm ShahRukh Khan, so allow me to launch nukes!!  
authentication  
failure  
ShahRukh does not  
have the authority to  
launch nukes!

Serial → one after the other / linear

Person p = New Person()





Medium level bytecode → good enough for 95% companies  
 Codeforces rating  $\approx 1700$

top companies      ← Scaler 99%  
 Let code - hard  
 Codeforces rating  $\approx 1900$