Apache Kafka (LinkedIn ne banaya)

Zomato → Live location of rider (google map view)

driver: P. A → P. B | driver → current location → zomato server → customer

1s get curr location → Zomato DB insertion

But for all driver across jyada input ho jayega

DB inserts after every 1sec for single driver

Operation per/s → DB down

Database [throughput = OPS (operation per second)]

Database की throughput कम होती hai...

Discord → server → 5 members are chatting realtime

Database insertion operation takes sometime (Realtime nhin),

Throughput = amount of item passing through a system.

Kafka has high throughput, down nhi hoga

↳ But not alternative to DB,

Kafka has very low temp storage.

In Kafka you can't query data

Na toh ham sirf kafka ko use krenge, na DB

Kafka + DB

Data Producer → Kafka → Consumers (different services) → DB

X1 Lakh records — fast input — fast output — Bulk insert

Inside Kafka servers you have topics

partioning of data

inside topic we may have partitions - but partitioning is not done based on time... it is based on fields (North India data & South data is Area is a field)

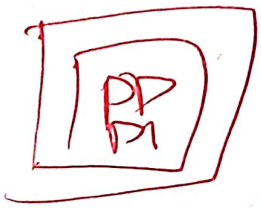
Kafka does auto balancing b/w consumers partitions

1 consumer ← Multiple partition

↑

But

1 partion → Multiple consumer ✗



Consumer groups (partitioning partition sharing happens first group level then individual level)

Queue (FIFO) , PUB-SUB

IP IC IC Multi C

Kafka Acts as Broker

Straightforward
OAuth
crossdomain
JWT { (CRUD)

Vid-8/

Mastering Access Control Patterns

Authenticate : To identify (login karwana, platform level)
Authorize : To give permission

Types of AuthZ

Admin → Table
Editor → Table
User → Table

1. Role Based Access Control... (Very small scale)
2. Fine-Grained RBAC (Resource level, not Platform level)
   → for every action we need to query the table & see user config.
   → Response time may increase
3. ABAC (Attribute)
   → "Resource Attribute" === "Person Attribute"
   Control over a resource is decided by their & persons attribute
4. Policy Based AC (Based on cloud Infra like AWS)
   Set policies for resource, user, team
   → set policies for user
   → set policies for team
   (Microservice)
5. Relationship Based Access Control
   If you have access to parent, you have access to child

P1 → Access
↓
P2 ✓
↓
P21 ✓

{4} Paper

(6) Google Zanzibar: Consistent Google Auth System

Graph Algorithm

(7) OpenFGA → Fine-grained Auth, Relationship Graph → ReBAC

Donated to CNCF by Google

Opensource look-alike of Zanzibar

V-{ → long-term payment

Kubernetes

(1) Bare Metal Deployment: Ghar pe hi server set-up karde.... IP address pe expose karde

(2) Server Rental → :

Server Physical Machine → on 24/7, Public / Static IP

(3) Code Behaviour on Server == That on Local Machine → Replicate dependencies + versions

AWS → Cloud was easily available → Cloud Native technologies (Static IP, LoadBalancer, Redis, Postgres yeh sab AWS dega, Cloudfront, ELB)

ASG → Auto Scaling Group...

→ Works on my machine issue

Virtualization

VMware, Own OS version, Own dependencies, Plus than bhi...

take them to cloud & deploy

later we removed OS from Virtualization

↳ containerization

↳ use kernel of host Machine

↳ Only Dependencies & Code

very light weight
config files

Same Behaviour Cross OS & platform.. (Scalable, shareable)

It was Hard to Manage Containers... → Creating & destroying, & even handling containers based on traffic of website..

↳ Google borg → Kubernetes

large scale cluster mgmt...

↳ Container Orchestration.

↳ Deployment, Scaling, Security

CNCF

IT → Greek

Kubernetes → Helmsman/

AWS ECS → Elastic Container Service

↳ Apni Docker image banao or upload karo

Vendor lock-in

Kubernetes - Cloud Agnostic (Generic)

- Yaml

Deployment config

↳ (Kubernetes)

Kubernetes Architecture

① Physical Machine → Control Plane
- ↳ API Server ↔ for Dev
- ↳ Controller
- ↳ etcd - KV store (Key-Value)
- ↳ Scheduler

Developer → (Config, Deploy) → API Server

(Yeh execute karta) Controller

(DB for req.) KV store

② Worker Working Node (2 min. for safety, saara kaam yeh hota hai.
- → Workload/container run karta hai
- Kubelet
- → Kube-proxy
- → CRI (Container Runtime Interface)

Another Physical Machine...

Key: Worker Node ⟷ Controller

Kubelet ⟷ API Server

③ You give instruction through ~~car~~ config file...

Config file → API Server → Controller

Controller will make Pods

Kubelet ↔ CRI

Scheduler → load balancing → load balance

→ CRI

Watch for new pods & assign to WN

RANKA
DATE / /
PAGE

firecracker VMs ~~or~~ CRI } AWS Lambda

CCMs Cloud Control Manager

API server → CCM (Load balancer / EC2 / Bucket
(AWS / Digital Ocean / GCP / Azure)

CCM (Provide support to ~~Specific~~ Specific Cloud Vendors)

Vid-6 (Scalable Notification System Design)

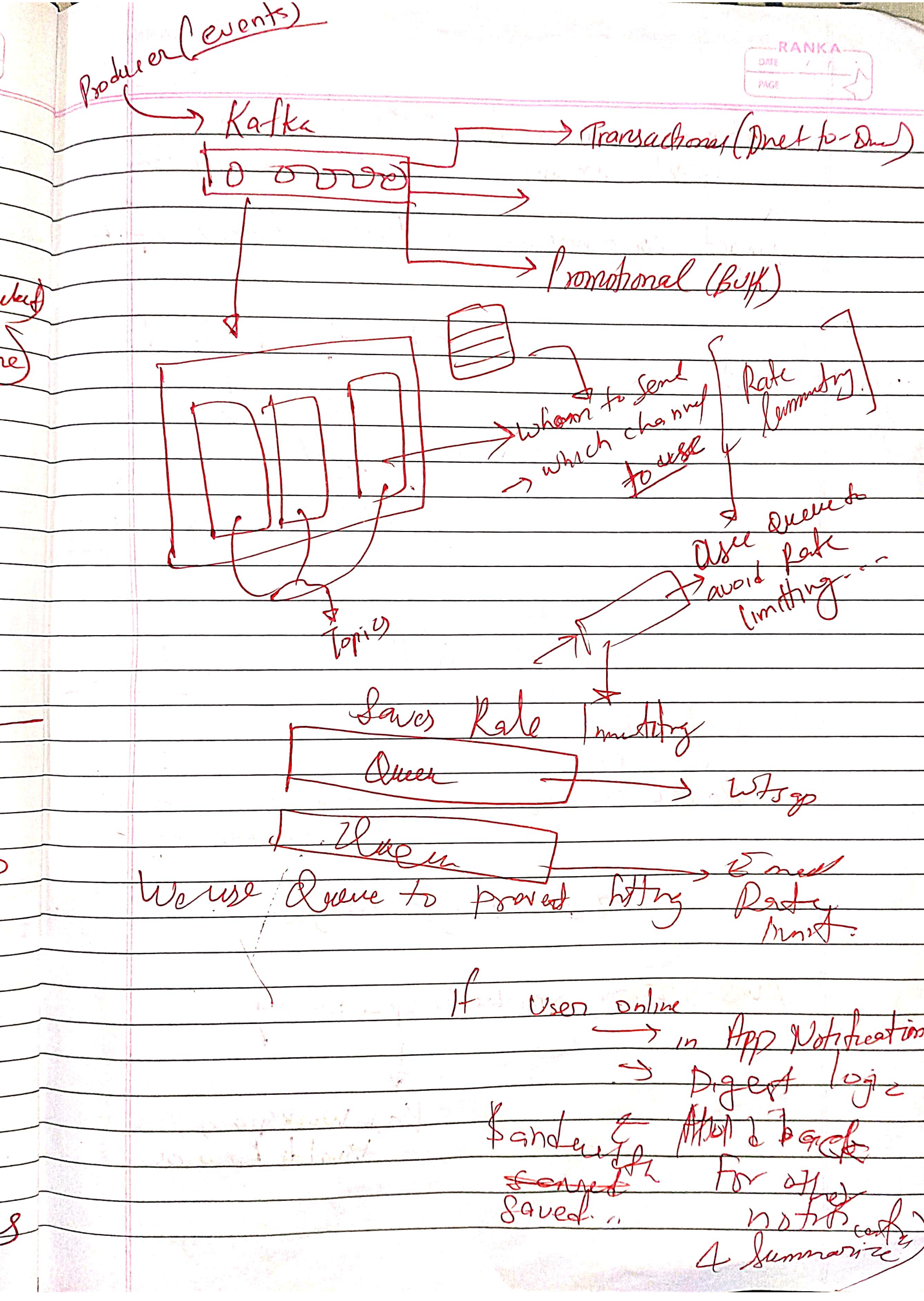
→ A lot of notification channel - Email, In App, SMS, WhatsApp

→ User given control to choose the channel

→ In fintech notification is imp

Sync → Not to use notification for every event
→ Backend slow ho jayega

Async → Best approach
→ High Throughput systems like Kafka or SQS
Lag


Producer (events)
Kafka
Transactional (Direct to-Dev)
Promotional (Bulk)
whom to send
which channel to use
Rate limiting
Use Queue to avoid Rate limiting...
topics
Saves Rate limiting
Queue
Queue
We use Queue to prevent hitting Rate limit.
If User online
→ in App Notification
→ Digest logic
Send
Saved
For offline
4 Summarize

priority queue for OTP logins for any other unique request notifications

Smart Consumers ... to first ~~and~~ ~~send~~ import Imp notifications

~~Pub-Sub not recommended~~

* Rate limit
* Priority

less notification — queue reduction

Host Your Browser & OS (7)

Local Machine

Docker की मदद (UI Browser & AWS Deployment)

& destroy after usage...

AWS → EC2 → make it up & running

↓

Kasm Workspace
Vivaldi Browser.

(8)

OWN VPN Server

VPN

Client - User (IP)

Internet (ISP)

Service (like Google)

Check for actual working diagram for VPN →

Client → VPN Client → ISP → Decrypt (add IP) → Service (for google clients & VPN)

encrypted route data to its remote server

extra hop add ho gya → speed slow ho jayegi...

VPN on Amazon EC2

↳ open VPN (Check for

EC2 → Different Region → Create new instance.

open VPN · AMI (Amazon Machine Image)

t2.large

launch instance

connect

Use Gitbash

Open session

follow instructions

(11) ~~How SSL Works~~ (Git / GitHub)

(12) ~~Git / GitHub~~

(9) Docker Mgmt API

AWS ECS → Docker Orchestration System → How many container Load balance } Auto matic

manual { CLI cmd → Docker run -it ubuntu bash entrypoint

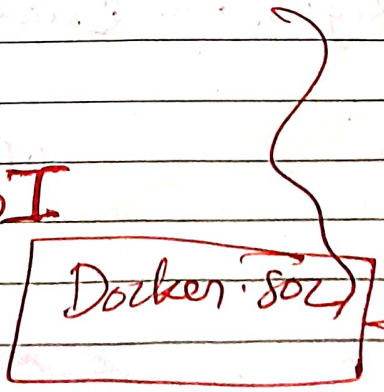
Cloud IDE

→ Docker container

3-4 users = 3-4 container spinning (60

→ container spin on demand kills after user gone

Docker App
Open Terminal
Docker Engine API



→ Entrypoint for communication

Docker Engine API :

Implement in Node.js using Dockerode...

Basically write code in js (node js) to create, spin, delete & hence manage containers

through get/post req in express server & call by through APIs & assigning ports

(15) Own Docker Hub...

Apna server jahan hum apne docker image ko push kr skte hain or baki log pull kr skte hain

similar to hub.docker.com

Steps AWS EC2 → Ubuntu → docker → docker compose → nginx → certificate ← SSL

AWS EC2 → t2.large → open the AWS SSH client on your terminal → install packages dependencies & docker → make docker registry folder → data folder to store all images push by public

make docker compose file (yaml file) listening to specific port

↓

wake up docker compose file...

↓

make authentication

↓

Setup nginx (HTTPS SSL certification)

↓

open ports on EC2...

↓

Push image to your own hub...

(11) SSL (Secured Socket Layer)

Client (User) → Server (Nginx)

(Get Req) ← Middleman can sniff... ← Response

(MMA - Man in the middle Act)

Secure bnana hai

[Symmetric Encryption: Same key used for both encryption & decryption.]

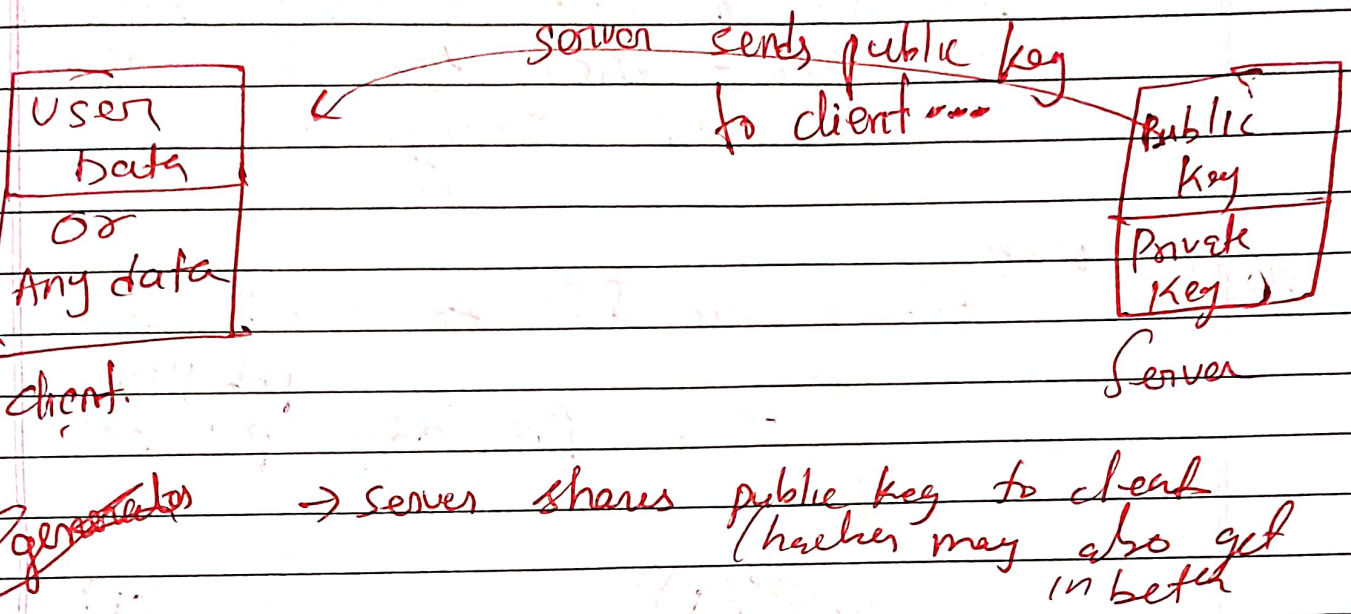
Encrypt data toh krlia, but ab key kaise securely transfer krein ki server hi decrypt kr paye.

Server can't read data

hence we use Asymmetric encryption (we have 2 keys)

→ Public key (can encrypt anything)

→ Private key (Can decrypt things)



Client generates → server shares public key to client (hacker may also get in between)

→ client generates own symmetric key & using public key of server encrypts the symmetric key & sends the symm key to server & servers decrypt using their private key & gets client's symm key

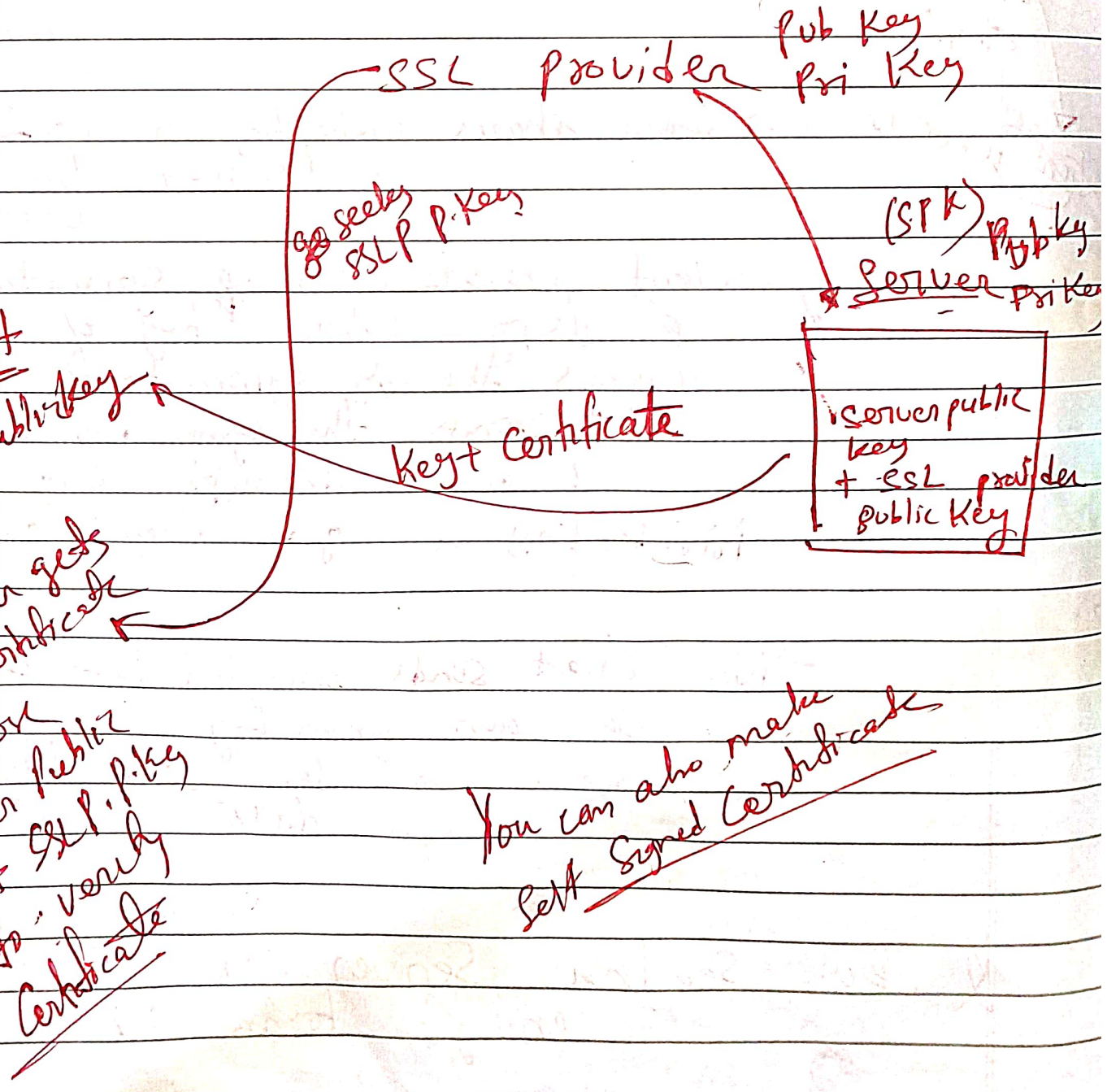
→ Now client sends encrypted data with own symm key encryption & server uses the same key to open data

Q. But Seedha Server key se hi data encrypt karta na ...

Isme bhi hacking ho sakta hai hacker may proxy as user in front of server & get its key & send a new key whose private key is with hacker to the user the encrypt users symmetric key, hence getting users key & data & sending users key to servers to look user proxy.

How SSL Works

Someone issues SSL certificate



CLI

(12)

What is VCS → Source Control ???

→ Git, Apache Subversion, Piper (Used by google)

→ Git Cheat Sheet by GitHub

→ git init

→ git graph extention install krlo

U = untracked

.git folder (don't touch it)

→ git diff (Diff dekhata hai)

To untrack files (git rm <file-path>)

Commit: Saving changes to make it a version

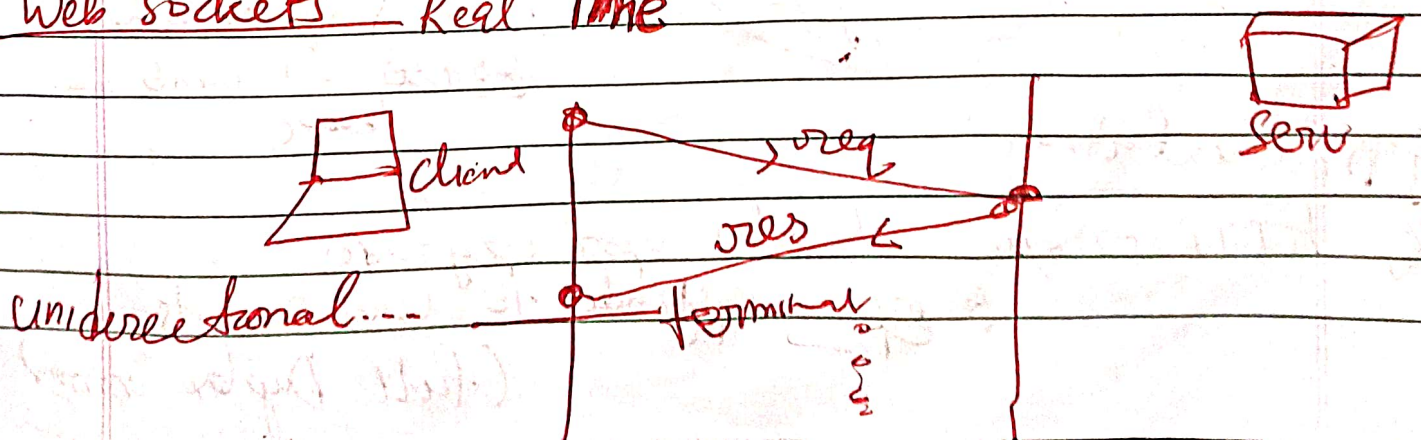
git commit -m "

→ git balam <file-path>

→ Reverting Back... (git reset --hard <commit no.>

removes the usage of to commit...

Web sockets — Real Time



Basic & most unoptimized way to achieve chat App ~~few~~ - polling
↳ Overkill

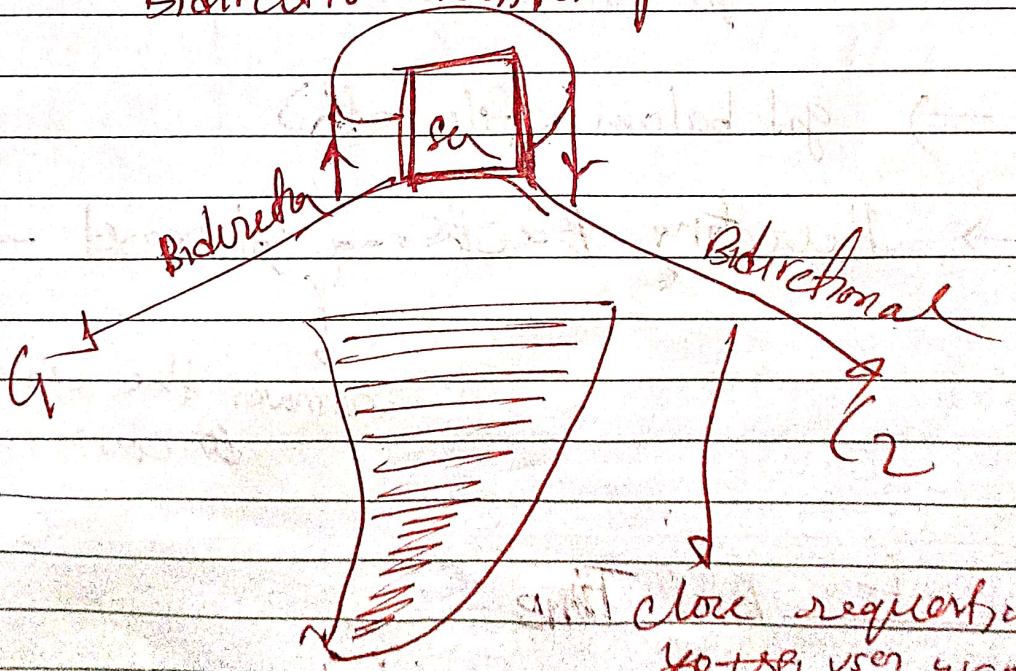
here comes web sockets

① Client sends HTTP req to server & say make a connection (Web Socket) & keep it active... (Upgrad Header)

(Connection close nhin hoga)

Connection close nhin hoga just after req & its response... (Unless you wait)

Bidirection lossless protocol...



#Documentations

① HTTP request header upgrade...
② eg upgrade to web sockets
(full Duplex comm)

Creating Discord Bot

1. Bot listen my tries & reply
2. Create My own Server for Club & Communities
3. Name it
4. General channel only sufficient
5. Open User setting → Advance → Turn on Developer mode
6. Go To Google → Discord Developer Portal

↓

Discord Developer Portal

1. Side bare → Applications
2. New Application, Name it...

Inside Application → Bot → Name it...

↓

Give Bot Admin Permissions

Add Bot to the Created Discord Server

Grant permissions

1. Inside Developer console → OAuth → URL Generator (Select Bot & Admin)

[Copy & paste URL in the Browser]

2. Select the Server & join...
3. To interact with bot we use Discord.js

RANKA

0 npm i discord.js

index.js → get recent token (very private)

import discord

discord gateway intent...

Kis tarah ke permissions de sakte hain...

(Guilds, Messages, Guild Messages, Message Content.

To give message permissions.

→ Turn on message content intent

client.on("messageCreate", (m) => {

console.log(m)

});

});

→ Make Discord commands on Discord.js

? Revise JS

(19) SSR (Node.js)

RANKA
DATE / /
PAGE

1. ejs (templating engine)
2. npm i ejs
3. app.set (view engine, ejs)
   import ejs

MVC

→ make views folder & ejs files inside views folder

Ref Ref Refer Re Notes, Usme, Khali
URL shortner Bnaya hai

MCP (Model Context Protocol)

Protocol = Rules...

HTTP → TCP

Anthropic

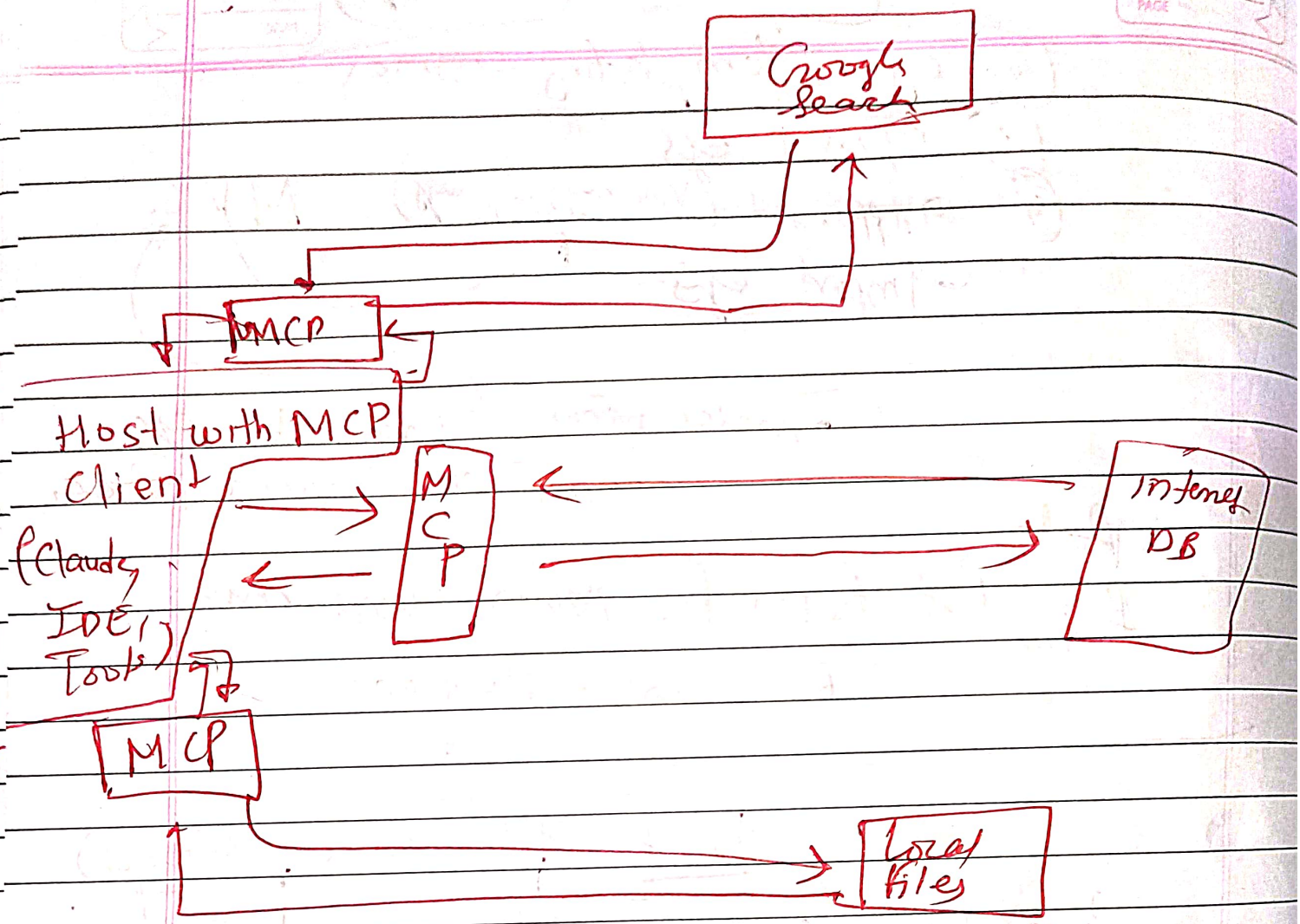
LLM — External info feeding — Rules

1. Context is imp for LLM to work.....
2. ~~Training~~ Training High & Costly & Re Training frequently Too Freq- Too High...
   → Context Window Bahut Chota (Limited)
3. How to efficiently (structure fully, give context to model)

MCP = USB-C port for AI Apps

MCP Provides Std. ways to connect to different data sources & tools...

MCP is next std



Link betn External World & AI (LLM) to feed Context.

#STDIO (Standard Input Output)

AI (LLM) ⟷ MCP ⟷ API

Bridge betn AI & Real time Data in external World

Same MCP server can be used cross model / cross IDE

RANKA

Initially MCP was built on STDIO (how we talk in terminal using echo)

MCP host (Claude, IDE or AI tool)
MCP client (Protocol maintain 1:1 connection with servers)
↳ Maintains I/O

MCP Server: Std. MCP server (Lightweight Program that exposes specific functionality)

Local Data Source: DB
Or
Remote Via APIs...

~~Steps~~ → import
→ server banao
→

MCP can make ~~to~~ tool, prompt, resources, API responses

Create ~~des~~ ~~resso~~ reusable prompt template & workflows.

STDIO ~~wont work~~ in cloud
hence we use SSE ~~transport~~
↓
Server Sent Event Transport

JWT

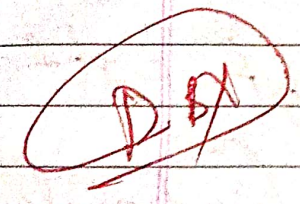
Cookies (for Authentication)

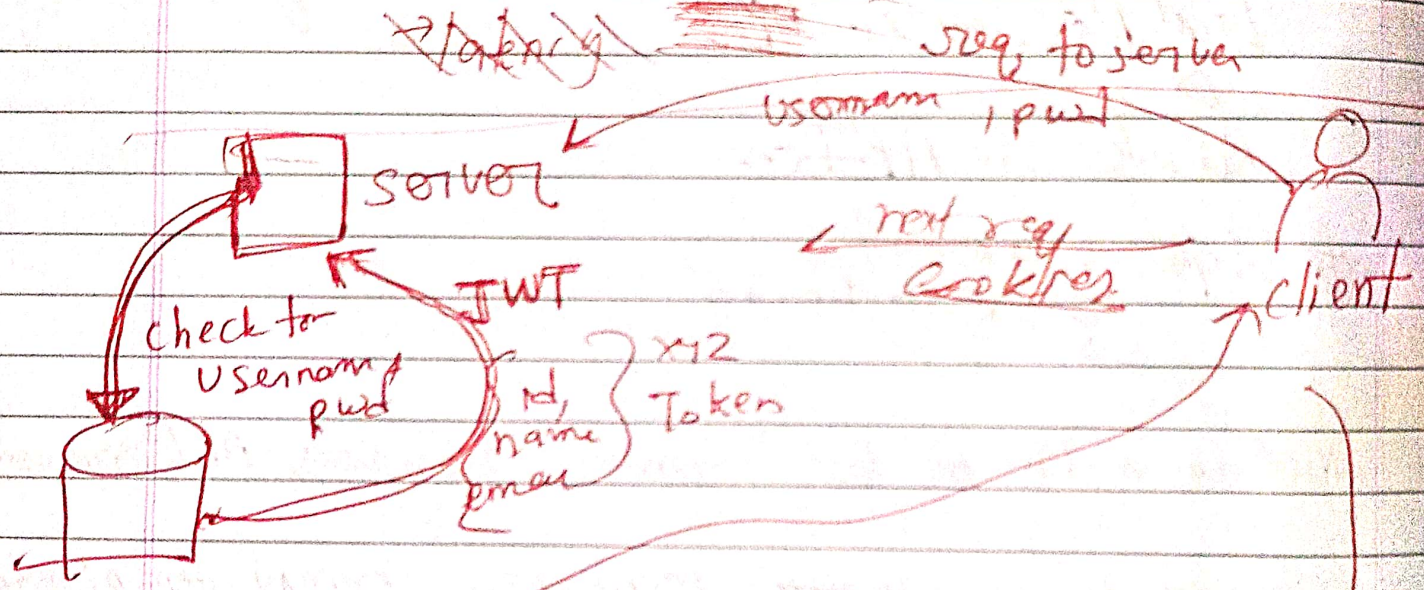
1. Stateless Authentication ...
2. Stateful Authentication ...

JWT Token → Session_id → user → memory

[Refresh (user gets logged out)]

[Session_id & User] we keep





Cookies = res.cookie (server data key)

data → uid token

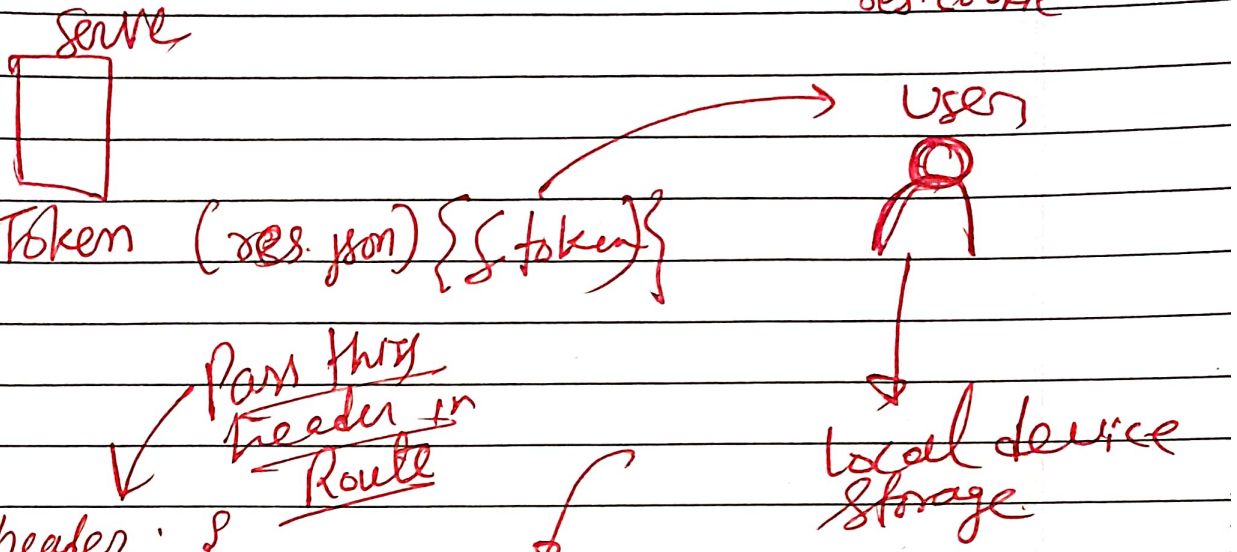
Browser store cookies

If cookie present (take token out of it & verify.

Cookies are domain specific transfer host (domain)

2800 | 3K

→ you can pass domain as option into res.cookie
" " " expiration date as option into res.cookie



Cookie work only for browser
Multiplatform Cross Platform

JSON → Header → Authorization