

## Push Notifications

- App ta device e running nai, but still notification ashtese. (Like Bkash)
- Triggered in 2 ways -
  - ◆ Can be triggered locally by an open application
  - ◆ Can be pushed from server to user when app is not running
- Push Notifications depend on 2 APIs -
  - ◆ **Notification API**
    - App open thak ba close thak, jekono notification display te show koray.  
*( Will discuss more further )*
  - ◆ **Push API**
    - Ekta service worker thake. Service Worker hocche ekta script jeta shob shomoy browser e run korte thake, eta khubi low amount of resources use kore, and shob shomoy listen korte thake kono push message/notification ashse kina. Push message ashle sheta Notification API er maddhome user ke show kore.

## **Important Terms**

- **Notification**

- A message displayed as pop up to user.

- **Push Message**

- A message sent from server to client.
  - Shob Push Message kintu Push Notification na.
    - Push Message ta ke Notification API er help niye user ke display korle tokhon shetake Push Notification bolbo.

- **Push Notification**

- A notification created in response to a push message.

- **Notification API**

- *Covered above*

- **Push API**

- *Covered above*

- **Web Push**

- Ei pura process e server theke client e message pass korbo. Ei pura process tay je je component involved thake, ei pura jinish ta ke boltesi web push. Ei pura...arre beda dhét.

- **Push Service**

- Ei service ta; the service/system for routing push messages from server to client.
  - Prottek browser er push service alada, and ekta. It's like a server!
    - For example, amar kache onekgula browser ase. Ekhon ami chrome theke facebook notification

on kore disi. Ami shudhu Chrome thekei tokhon facebook er push notification gula pai. Kokhono emon hoy na je onno browser thekeo same notification gula ashtese.

- **Web Push Protocol**

- Describes how an application or server interacts with the push service.

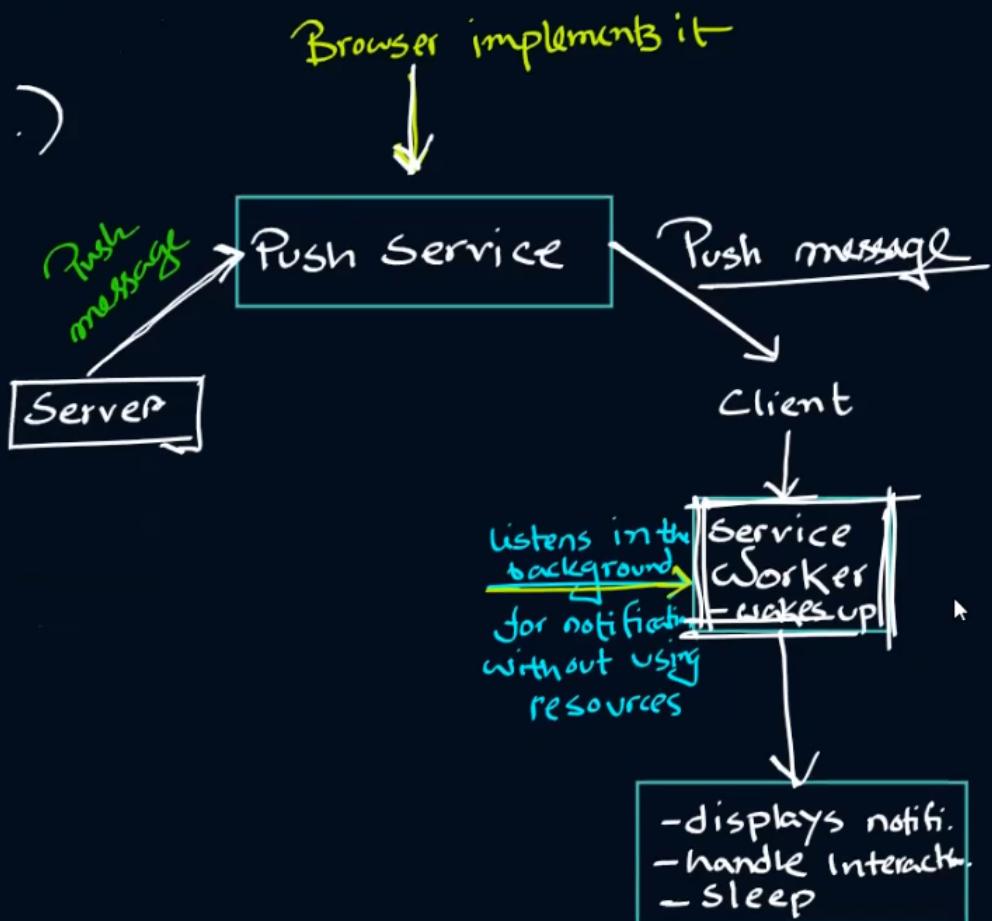
→ ***What can we do with a push notification?***

- ◆ Alert user to an important event.
- ◆ Display an icon and a small piece of text that the user can click to open up your site.
- ◆ Integrate buttons, sounds, vibrations etc.

→ ***Restrictions of Push Notifications on Web***

- ◆ Browser e push notification gula new, not advanced compared to Android, iOS.
- ◆ Ekek browser ekekvabe push service implement korse. Shob browser shob push functionality support kore na, like sound, vibration nai. Web er khetre ami jani na amar application ta kon browser e use hobe, but android/iOS er khetre sheta jani.

## How A Push Notification Works



Here, Server → Amar app er server

Client → User er Browser

## **Notification API**

- Displays notification to the user
- Can be divided into -

### **◆ Invocation API**

- Controls how to make your notification appear.
- Just notification ta show korbe.

### **◆ Interaction API**

- Controls what happens when user interacts with it.
- Notification tar sathe interact korle (e.g. swipe, click) ki ki hobe, sheta handle korbe.

## **How to Display Notification**

- Notification display korar jonno server sider er kono kaj nai, puratai device / client side er kaj.
- 3 steps -

### **◆ Request Permission**

- User er kach theke permission nibe

```
Notification.requestPermission(function(status) {  
    console.log('Notification permission status:', status);  
});
```

### **◆ Display Notification**

#### **● *Without Options***

- User er permission ase kina check korbe. Permission thakle service worker ke diye notification show korabe.

```

function displayNotification() {
  if (Notification.permission == 'granted') {
    navigator.serviceWorker.getRegistration().then(function(reg) {
      reg.showNotification('Hello world!');
    });
  }
}

```

without Options //

- With Options

```

function displayNotification() {
  if (Notification.permission == 'granted') {
    navigator.serviceWorker.getRegistration().then(function(reg) {
      var options = {
        body: 'Here is a notification body!',           ←
        icon: 'images/example.png',                   ←
        vibrate: [100, 50, 100],                      → time in ms
        data: {                                       ←
          dateOfArrival: Date.now(),
          primaryKey: 1
        }
      };
      reg.showNotification('Hello world!', options);
    });
  }
}

```

with Options //

Hello world  
Here is notifi body

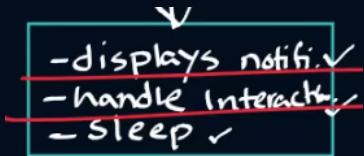
even indices - vibrate  
odd indices - silent

- Vibrate (100, 50, 100)
  - ◆ Even indices - Vibrate
  - Odd indices - Silent
- ◆ Ekhane first 100 ms vibrate korbe, next 50 ms silent thakbe, then next 100 ms abar vibrate korbe.
- Primary Key
  - ◆ Notification er sathe user er behaviour lock kore rakhte chaile.
  - ◆ Suppose, ekta notification 900 jon ke pathailam, 800 jon close kore dilo, baki 100 jon interact korlo. Prottek notification eri nijer

primary key ase. Sheta diye server e log rakhte pari, in order to research je kon notification kon user ke dekhabo ba dekhabo na. Also custom user er jonno custom notification banaite pari.

#### ◆ **Listen for Events** ----- Service Worker

- Service Worker notification er jonno listen korte thake, jokhon notification ashe tokhon sheta display kore dey.
- Notification display korar por service worker listens for events (notification close, key value notification). Then based on that, service worker oi event ta handle kore.
- Erpor ghumay jay.

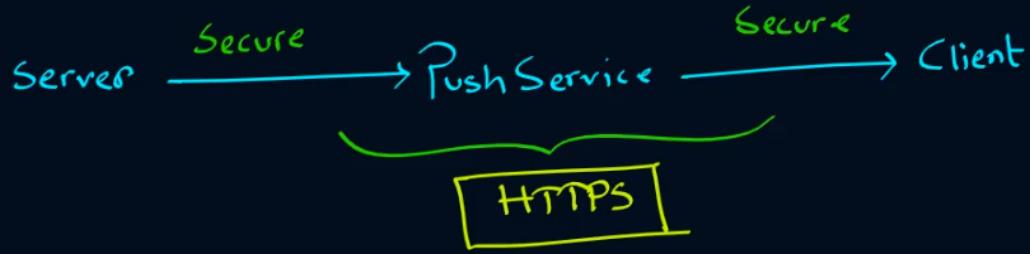


## Push API

### → How it works

- ◆ Push Service
  - Prottekta browser er nijeder push service ase.
- ◆ User grants permission
- ◆ Subscribe app to browser's push service
  - Client request accept korle client er push service server ke ekta object pathay. Object e thake -
    - **Endpoint URL**
      - ◆ Server theke client er browser e kono notification pathate chaile ei URL e pathate hobe. Ei URL er last e ekta unique identifier thake, jeta diye message ta kono ekta particular user kei pathaite parbo.
    - **Public Key**
      - ◆ Server jokhoni client browser er push service e kono notification pathabe, tokhon ei public key diye encrypt kore dibe.

```
{"endpoint": "https://fcm.googleapis.com/fcm/send", "keys": {"p256dh": "BLQELIDm-6b9B107YrEuXJ4BL_YBVQ0dvt9NQGGJxIQidJWHPNa9YrouvcQ9d7_MqzvGS9A1z60SZNCG3qfpk=", "auth": "4vQK-SvRAN5eo-8ASlrwA=="}}
```



Let's wrap up the whole fking thing -

**On the Client:**

- Subscribe to the push service
- Send subscription object to the server
  - ◆ Endpoint, public key

**On the Server:** (Application, not push service)

- Generate the message that we want to send to user
- Encrypt the data with public key
- Send data to the endpoint

After finishing all these, browser e jinish ta jawar por -

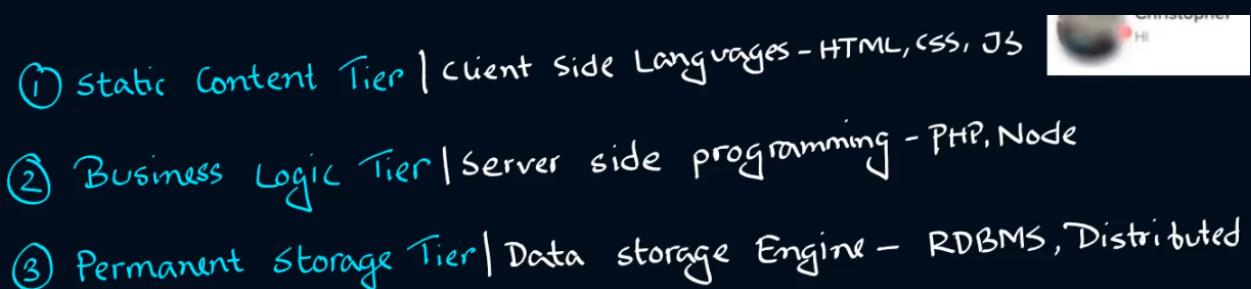
- Wakes up the service worker
  - ◆ Receives the message
  - ◆ Show the notification
  - ◆ Handle interaction
  - ◆ Sleep

## System Design, Performance Scaling & Load Balancing

We will be learning these topics through building a Tinder app with these features -

- Store Profile
- Recommend Matches
- Note Down Matches
- Direct Messaging

First e Monolithic Architecture hishbe start korbo. Then user base grow korle pore aste aste performance scaling korbo.



Ei 3 ta tier ekta server e rakhle Monolithic. Business tier ke bhanglei sheta Microservice hoye jabe, ar Monolithic thakbe na.

Aste aste user base grow korar por dekhlam unresponsive behaviour start hoilo, e.g. image load hoite time lagtese / load hocche na.

Eta fix korte first step nibo -

## **Performance Tuning**

→ Existing code base improve kora

→ **How can we do that?**

- ◆ Refactoring source code
- ◆ Convert unnecessary Synchronous functions to Asynchronous functions
- ◆ Logic improvement
- ◆ Caching

Performance Tuning korar capability shesh hoye gele next step -

## **Vertical Scaling**

→ Existing server kei spec up kora

## **Horizontal Scaling**

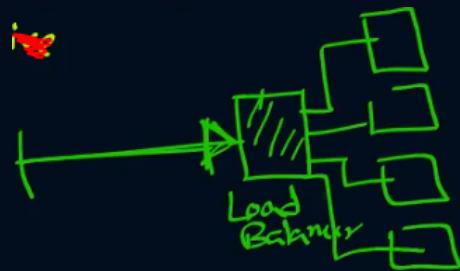
→ Number of server increase kora

Erporeo ek shomoy giye amar Business Logic Tier ar load nite partese na. Ekhoni Microservice Architecture implement korar kotha vabbo na, cz eta onek costly. Rather we will do at first -

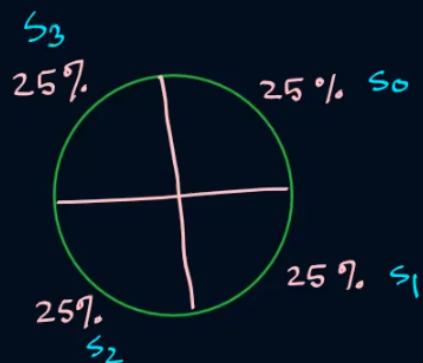
## **Load Balancing**

Business tier ta ke divide kore 4 ta server korlam suppose.

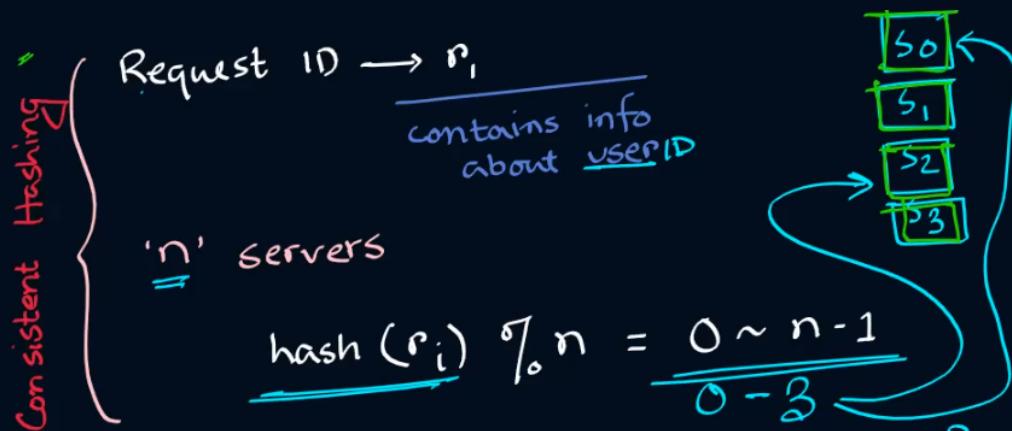
→ Ekta user er data ekta particular server e cache kore rakhlam to get faster access. Shob gula server e sheta cache kore rakha is not a good design.



→ User request first e Load Balancer e jay. Shekhane Consistent Hashing er madhhome decide hobe kon server e sheta forward hobe, then oi particular server e forward kore dibe.



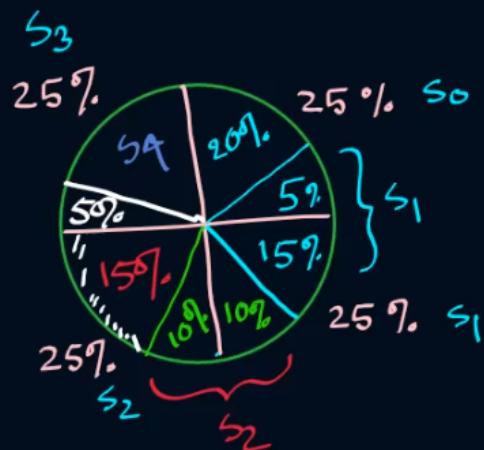
Business tier er 4 ta server e ekhon 25% kore load portese, jeta age purata ekta server er upor porto.



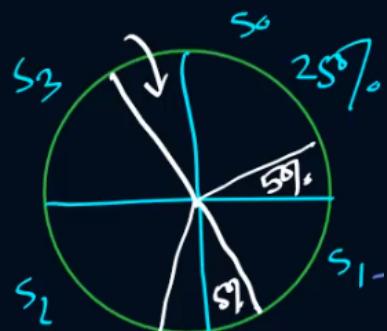
**What if we add a 5th server?**



Ager 25% gula ekhon 20% kore divide hoye jacche.



But this division has a problem. Ager 25% e jei cache data gula chilo, ei division er por shegula ar pabo na. Taile ekhon solution ki?



Rather we will divide like this. 4 ta server theke equal amount of space notun server ke dibo. Prottek 25% theke 5% 5% kore nibo arki. Although ekhaneo oi 5% user

er cache data lose hocche, but it is better than losing all user's cache data like before.

Load balancing koreo temon labh hoilo na. Ekhon koi jabo? MicroService Architecture

### Monolithic vs De-coupled Business Tier

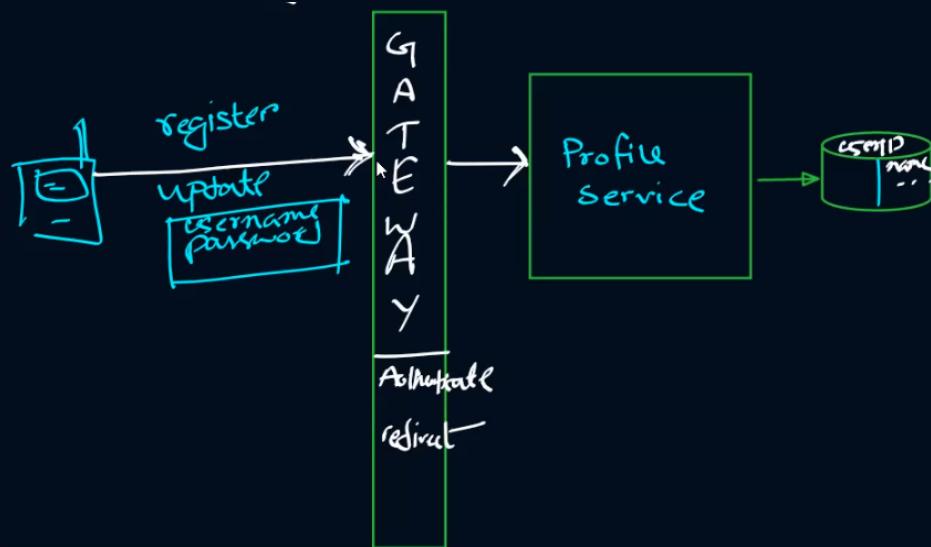
- Usually services use separate Database/Tables



- ① Store Profiles (Images per profile - 5)
- ② Recommend Matches (Active Users)
- ③ Note down Matches
- ④ Direct Messaging

Egula decouple kore koyekta service e vaag korlam.

Let's store profiles first -



Now, store images -

#### → File vs Blob

##### ◆ **Mutability**

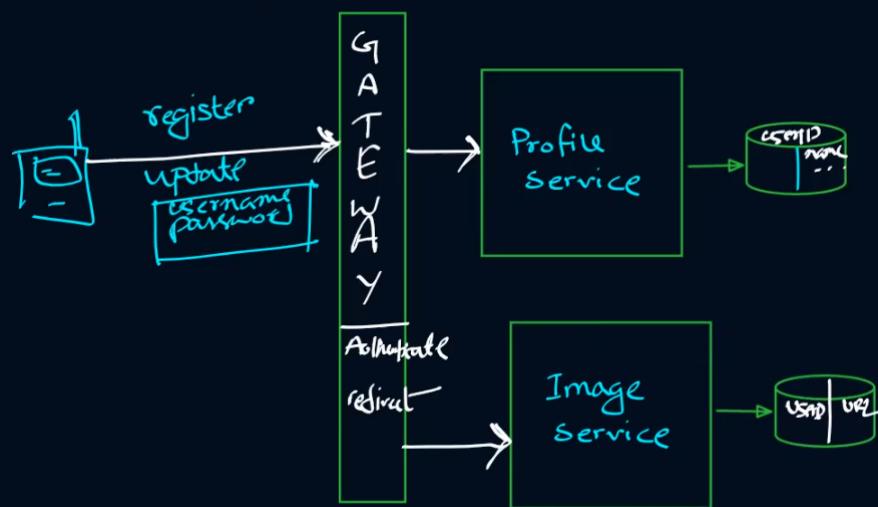
- Ekta image change korle pura file / whole image change kori, mukh rekhe shudhu amar dehokhan change kori na. So file diyei hoye jacche, Blob dorkar nai.

##### ◆ **Transaction**

- Eta kono secured emon kisu na je eta fail hoile database inconsistent hoye jabe. So Blob er dorkar nai.

##### ◆ **Indexes**

- Tinder e ami image diye search dibo na google image er moto. So Blob er dorkar nai.



Store profiles Done!

Let's note down matches now -

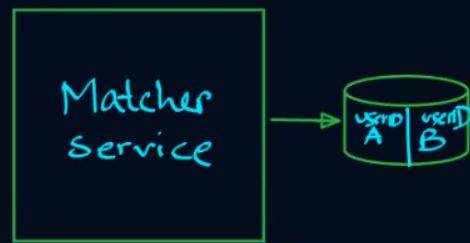
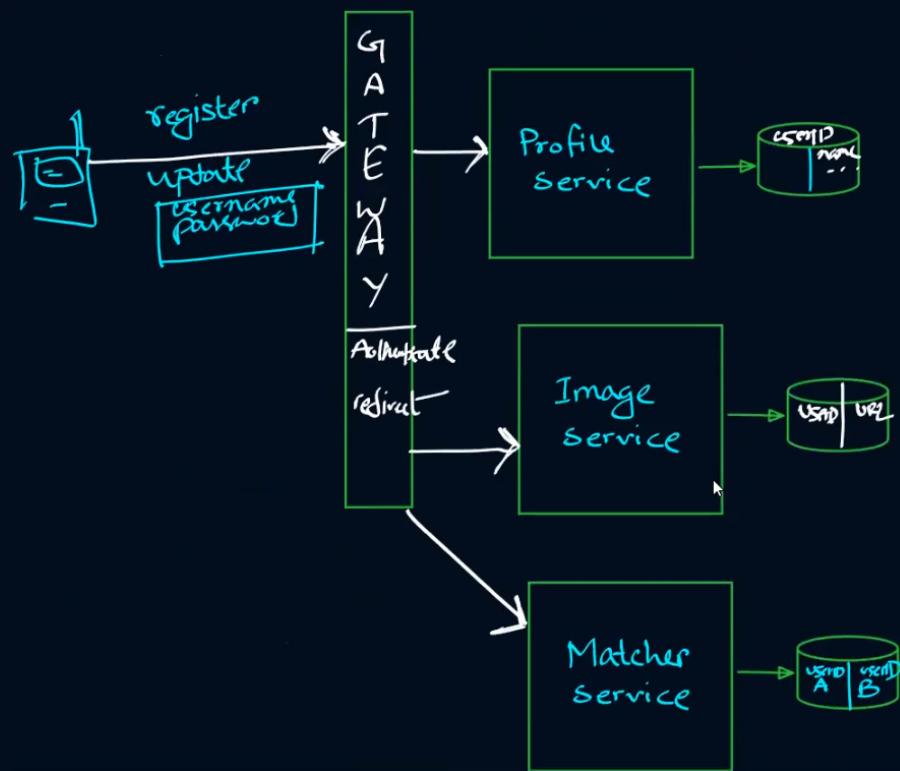


Table e userID A, userID B thakbe. That means A, B er sathe match korse. Why is it necessary? Bcz ekbar jehetu match korse, ekhon ami ar A ke B er kase recommend korbo na.

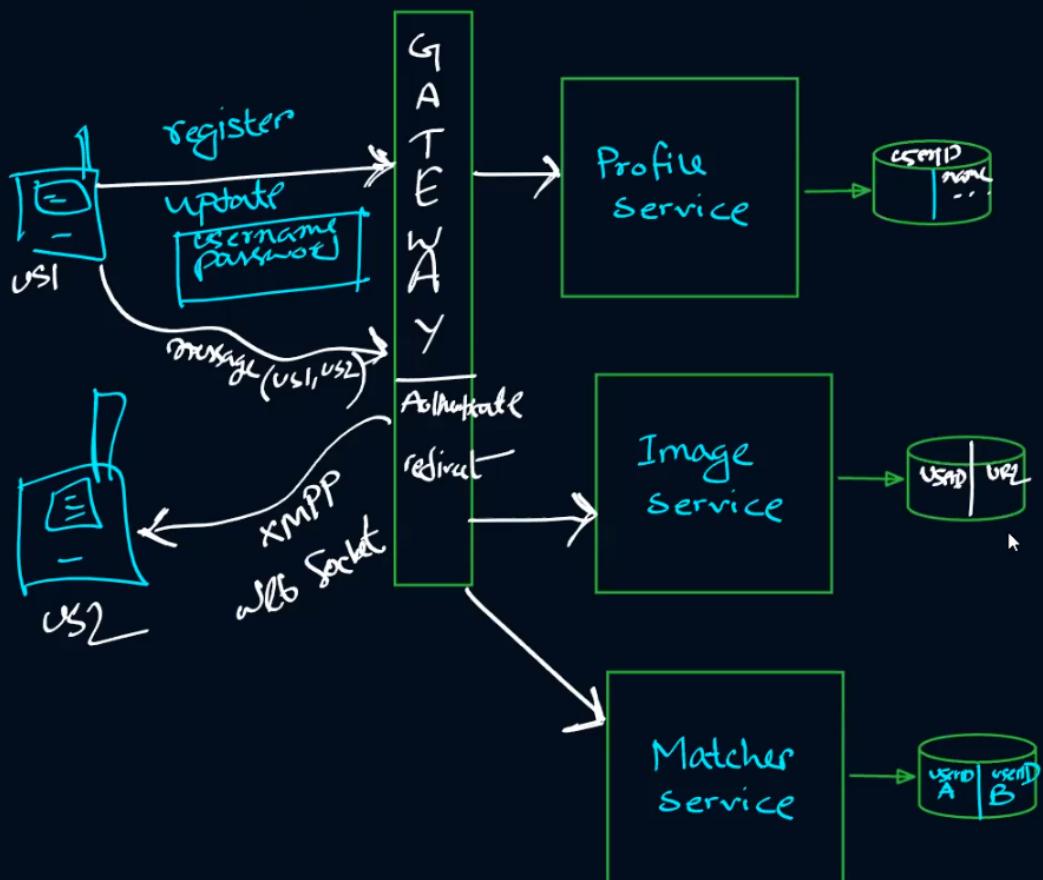


Note down matches Done!

Now let's implement direct messaging -

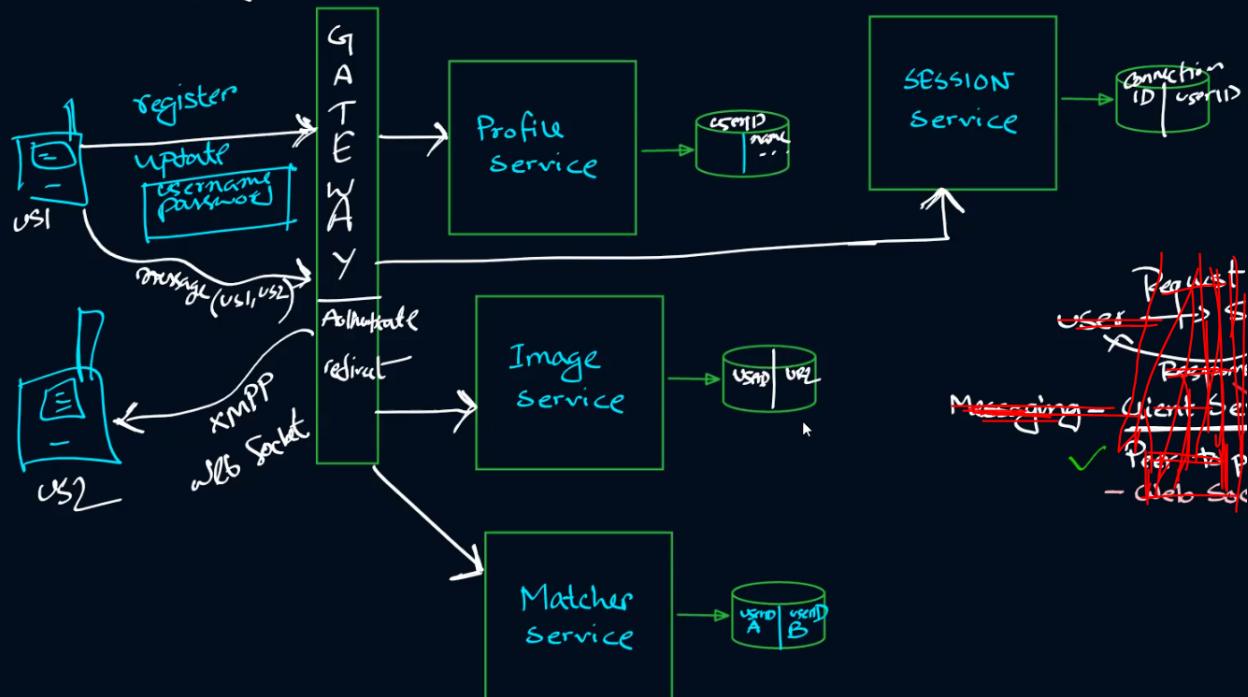
Age amra client-server protocol (e.g. HTTP) use kortam. User server e request pathaito, then server response korto, server nije theke user ke data pathaito na. But in this

case, we need something peer to peer (e.g. XMPP), je keu je karo sathe communicate korte pare.

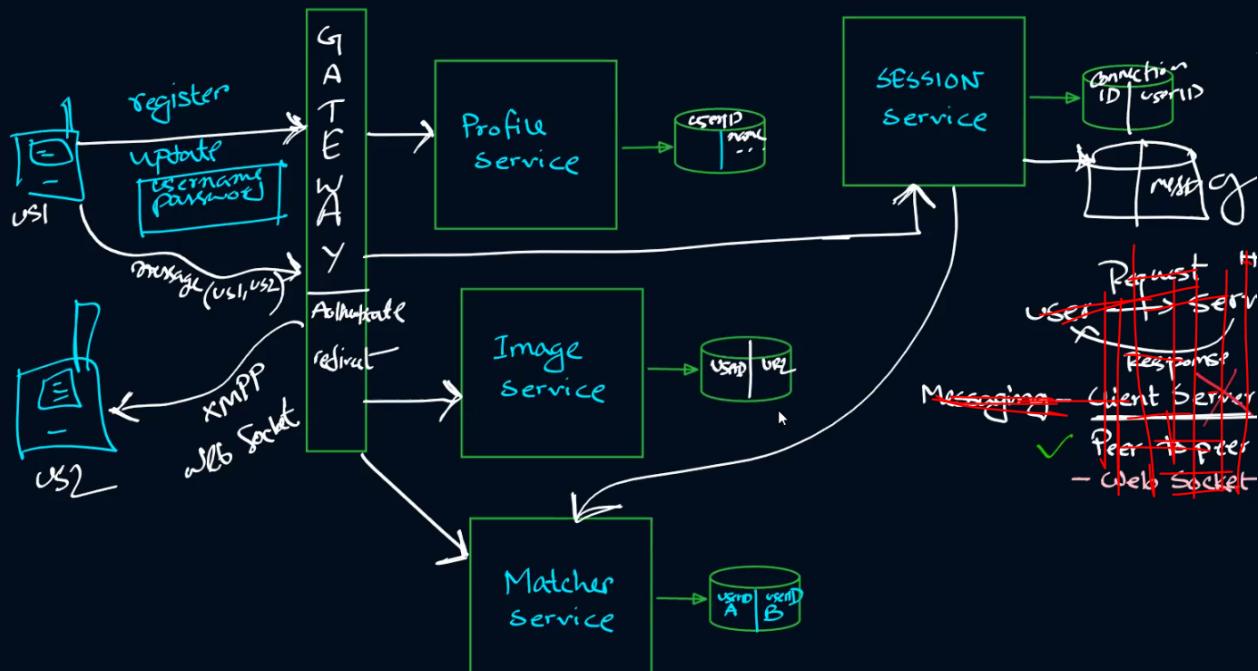


Tinder e to onek user ase. Jokhon duita user communicate korbe, tokhon gateway ke bujhte hobe kon connection diye kon user ke message pathabe. Ejonno ekta Session Service lagbe, jeta ekta connectionID ar ekta userID store kore, ja diye bujhay kon user kon connection use kortese.

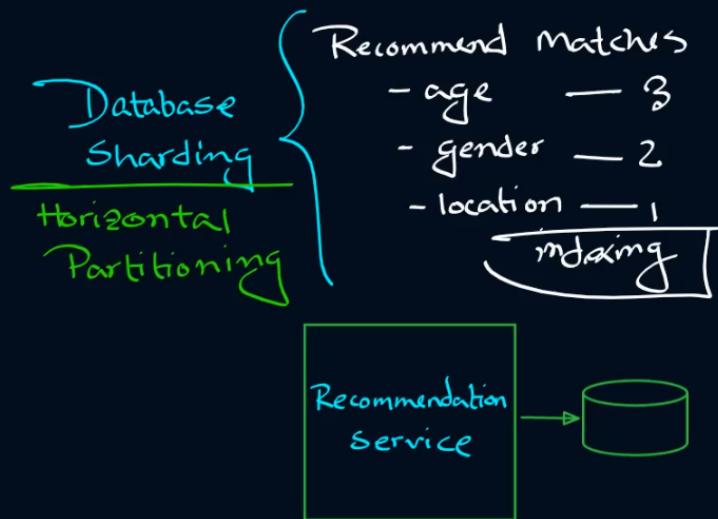
User1 ekta message pathailo, then gateway session service er kache dekhbe ei user1 je connectionID diye message pathaise, shei connectionID ta ar kon user use kortese. Emne user2 er sathe ekta connection establish korbe for the first time. Erpor theke ar connection establish kora lagbe na, oi connection diyei message pass korbe always.



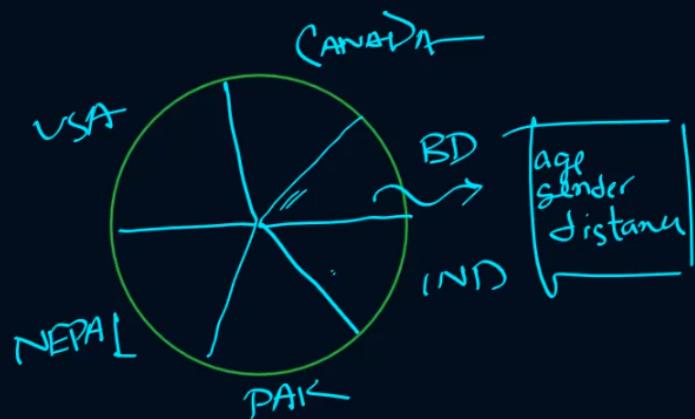
Ei Session Service e message pathanor age Matcher Service use kortese. Match na korle message pathaite dibe na.



Now it's time for Match Recommendation uWu -

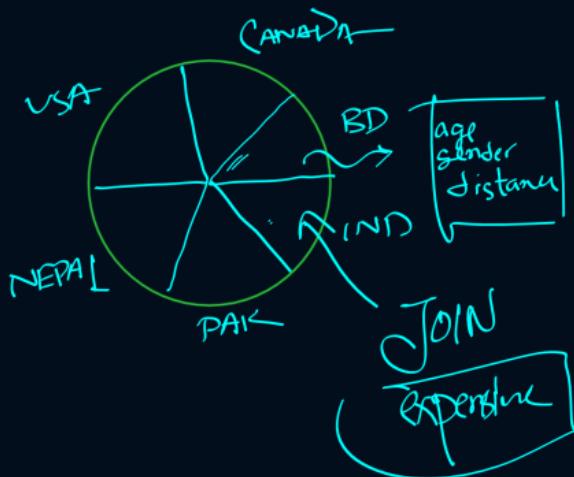


Bangladesh theke pura world er billions of users theke search korte to jaan bair hoye jabe.



Here is a concept, amra BD er jonno only BD er database theke search korbo.

**But what if ami border area te thaki?**



Tokhon dui region er database er moddhe JOIN request pathaite hobe which is very expensive. But kisu korar nai, it is what it is!

**What if ekta user Bangladesh theke India chole jay?**

Ei khetre sathe sathe database update korar luxury amader nai. So ekta fixed interval porpor region gular Recommendation Service er database update korte hobe, kono new user ashse kina. This method is called **Database Sharding**.

## Designing Instagram

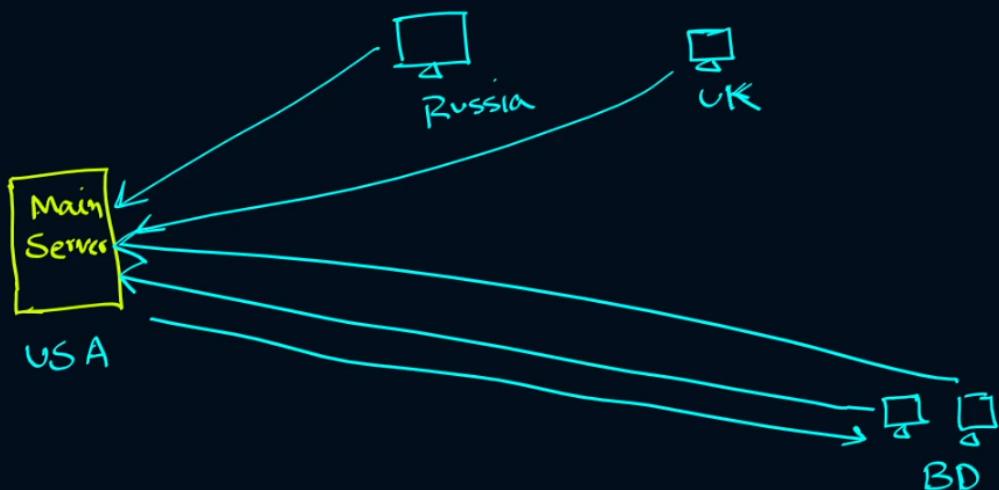
### Features

- Store/Get Images (File)
- Like/Comment on Images (Post)



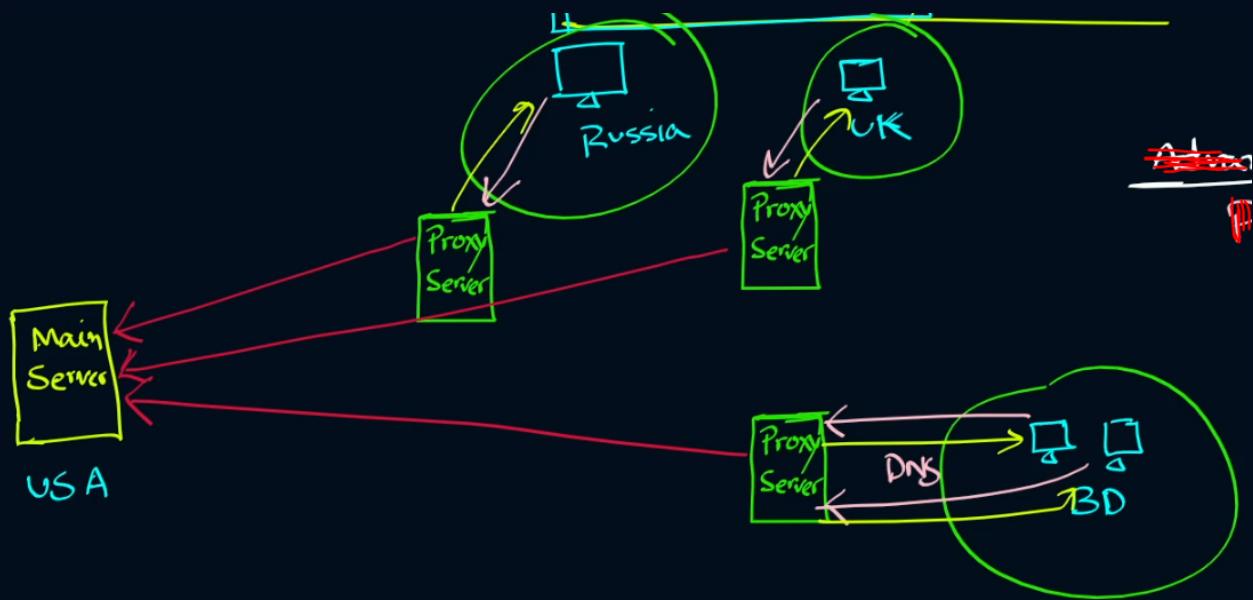
- Follow someone
- Publish a News Feed

### CDN (Content Delivery Network)



In this case, Round Trip Time onek beshi lagbe. Cz every static content (e.g. image) load howar jonno main server USA theke data load kora lagbe.

In case of CDN,



Onekgula region e vaag kore nibo. Eder sathe proxy server thake. Jokhon kono static content load kora dorkar hoy, tokhon ei proxy server theke load hoy.

### Advantages of CDN

- Performance
  - ◆ Round trip time kom lagbe.
- Less load
  - ◆ Age main server e millions of request ashto. Kintu ekhon jotogula proxy server, totogula request ashe.
- Security
  - ◆ CDN is provided by Amazon, Cloudflare etc. Oder server gular alada firewall ase.

## **Static Files**

- Image, Video, Pdf
- TTL (Time To Live)
  - ◆ Suppose, TTL 7 days hoile 7 days por proxy server main server ke request pathaye bolbe, "accha amar kache to ekta image chilo jeta ami 7 din age nisilam, oita ki ekhono same ase naki change hoise? Change hoile notun version ta amake pathay dao."

*Two types of CDN -*

### **Push CDN**

- Main server Proxy server ke notun content pathabe jokhoni kono ekta content change hoy. Mane user kono kichu change korlei Proxy server ke Main server update kore dibe, "tomar under e jei user gula ase, tar content e ei ei change ashse."

#### **→ Advantages**

- ◆ Notun content gula shathei shathei receive korte partesi.
- Suitable for -
  - ◆ Sites with small traffic. Server e load kom thakle time to time proxy server ke update korte pare.
  - ◆ Static files frequently update

### **Pull CDN**

- Fetch new content from server when first user requests it.
- TTL determines for how long content is cached.

#### → **Advantages**

- ◆ Main server er load kome jacche. Age ekta content change hoilei main server ke ekta response pathano lagtesilo proxy server ke.

#### → Suitable for -

- ◆ Sites with high traffic.

#### → **Disadvantages**

- ◆ Suppose ekta image er TTL 7 days. But user oita 5 days por delete kore dillo. Still proxy server e image ta aro 2 days theke jabe, and oi region er onno user ra shei expired content dekhbe.

### **Disadvantages of CDN**

#### → Cost

- ◆ Age data gula ekta server e rakhtam. Ekhon alada alada region e server e rakhte hocche. Increases cost.

#### → Require Changing URLs

- ◆ Common user er khentre Image gula Regional Proxy Server e cached kore rakhtesi. But celebrity er khentre content gula multiple proxy server e store korte hocche; waste of space. Also ekta image er jonno different URL generate kora lagtese.

#### → Expired Content

- ◆ Suppose ekta image er TTL 7 days. But user oita 5 days por delete kore dillo. Still proxy server e image ta aro 2 days theke jabe, and oi region er onno user ra shei expired content dekhbe.

Let's get back to designing Instagram.  
Store/Get Images is done by creating CDN.

Now let's design a database for Instagram -



Amar eshob like comment ke koyta korse eshob bishoye kheyal rakhte hocche. Ekhon suppose, keu ekjon post dilo -

Post				
ID	User ID	Text	Image URL	TS
—	—	—	—	timestamp

Ei post e jokhon keu like dibe -

Likes				
ID	ParentID	User ID	Timestamp	Active
—	—	✓	✓	False → true

Keu jodi like remove kore dey, tokhon active field ta false kore dibo. ParentID diye define korbo eta post er like naki comment er like.



Erokom khetre like ta post er naki comment er sheta janar jonno whole id er upor ekta string operation korte hobe, which is very expensive. As a solution, we can add an extra field -

Likes				
ID	ParentID	UserID	Timestamp	Active
	=	✓	✓	True → false

Ekhon jodi jante chai, ekta post e kotogula like ashse -

~~Select count(\*) from Likes where parentID = " "~~

But in case of Instagram, this is not feasible. Cz proti second e millions of users news feed refresh kore. Shobar jonno database e emne request pathaite gele database dead.

Rather we can maintain a separate table for counting likes and comments for a particular post.

### Activity

ID	Parent ID	Likes	Comments
	Post / Comment	+1	0/1
		+1	

Ekta like korle Likes +1 kore dibo, abar ekta comment korle Comments +1 kore dibo.

Likes and Comments table same.

### Comment

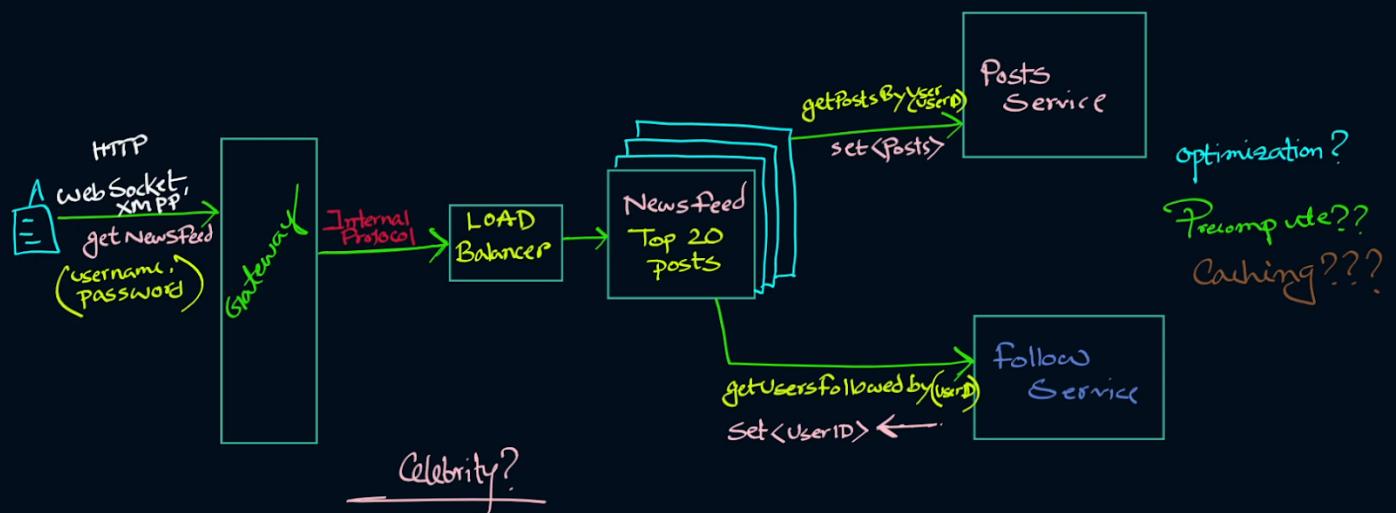
ID	Parent ID	Text	Timestamp

For followers -

### Follower

ID	Followers ID	Followee ID	TS
	x	x	

Now let's move on to core Instagram design -



User Gateway server er kache username, password, token etc pathaye authenticated hobe, othoba user1 user2 just message passign korte pare. Gateway server er kaj hoilo user ke authenticate kore deya, ba amar request ta kon service er kache jabe shekhane route kore deya.

Jehetu Gateway server internal server gular sathe communicate kore, tai ekta internal protocol use korte hobe, ja outside er kono server bujhbe na.

Load balancer decide korbe kon user kon News Feed service er kache jabe.

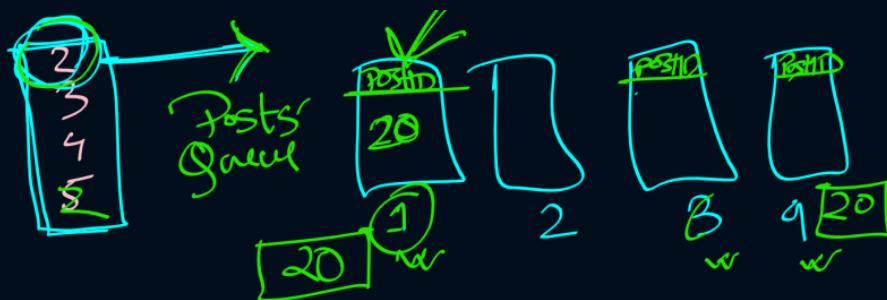
Ekhon dhorlam, user ke news feed e 20 ta kore posts dekhabo. News Feed service, Follow service ke userID pathabe. Oi user er follower der ID Follow service News Feed service ke diye dibe. Erpor Post service ke ei set of userID gula pathale Posts service oi user gular posts News Feed service ke return korbe, and then news feed e oder post gula show korbe.

There can be more services like these -

→ Image Service  
Likes Service  
Comment Service  
Activity Service  
Session Service

### Optimization

- Limit kore dibo je, jokhon news feed er request korbo, tokhon maximum 20 ta post nibo.
- News feed e jei 20 ta post dekhabo, shegula precompute kore rakhte pari.



For example, 4 ta user er jonno ekta queue maintain korte pari. (Ei user gula kinda ekjon arekjon ke follow kore type kisu ekta!) User 2 jokhon news feed refresh korbe, tokhon baki dero queue update hoye jabe. Next time, jokhon oi user gula news feed refresh korbe, tokhon queue er posts show kore dibo. Eta hocche News Feed Precompute kore rakha. As a result, server er upor load onek kome jacche.

Ei precompute dui jayga te kore rakhte pari - Cache & Database. Which is better? Jei user gula frequently news feed refresh kortese, tader ta cache te rakhbo. Frequent user na hoile Database e rakhbo.

## → Notification

Suppose, user kono post dile tar follower der kache push notification pathabo. But what if it's Kylie Jenner? Celebrity der million of followers ke eksathe pathaite parbo na. Solution?

### ◆ **Batch Notification**

- Batch batch kore pathaite pari, e.g. every 10 second 10k notification pathabo, next 10 sec abar 10k notification pathabo.

### ◆ **Polling**

- Amar device ta proti 1 mint por por check/poll kore kono notification ase kina. Ami only tokhon notification pathabo jokhon user ta poll korbe. Offline thakle chinta kora lagtese na, online ashle pathabo.

## Deployment

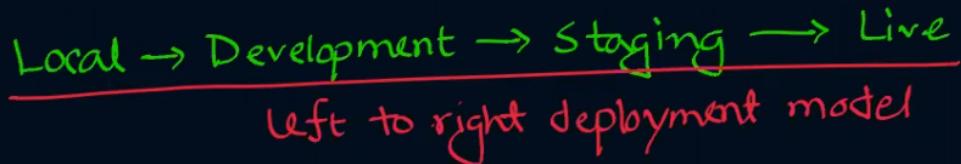
→ Push changes or updates from one environment to another.

→ Environment

- ◆ For example, amra jekhane code kori sheta Development Environment. Abar je environment theke client amar software/website ta access kortese, sheta Production Environment / Live Environment.

### Left to Right Deployment Model

→ Coding theke shuru kore Client er kache jei process er moddhe diye jabe.



→ Local Environment

- ◆ Emon ekta environment setup korsi je amar PC theke message pathai, sheta amar phone e show kore. Ei whole thing tai local environment.

→ Development Environment

- ◆ Amar jei messaging service ta (not whole app) local e thik moto run korlo, etake ekta development server e upload korlam.
- ◆ In most of the cases, Local and Development environment are the same.

- ◆ Might not contain all the microservices.

#### → Staging Environment

- ◆ Age jeta upload dilam, oita whole app er ekta part. Erokom onek gula service thakbe. Shob gula eksathe Staging Environment e upload dibo.
- ◆ It is actually the exact replica of the Production Environment.
- ◆ Contains all the microservices + newly added services.
- ◆ Here, QA people will do the testing.

#### → Live Environment / Production Environment

- ◆ Staging e jodi thikthak moto kaj kore, taile app ta Live / Production environment e deploy korbo.

*Ei local theke live prottek environment ei jokhon ekta theke arektay changes niye jabo, etai deployment.*

### Deployment Process Flow

1. Software deployment plan
2. The actual development
3. Testing your changes
4. Deploying changes to the live environment
5. Monitor your changes

### Different Types of Deployment

#### → Deployment of meta-data

- ◆ Codes, css etc
- ◆ Always controlled by developers.

#### → Deployment of content

- ◆ Photos, videos etc

- ◆ Content creators can upload.

## Deployment Best Practices

- Use Git
- Work in branches
- Review the differences before going live
- Deployment schedule
  - ◆ Less number of active users, enough developers - eishober upor base kore ekta deployment schedule choose korbo.
- User groups with different permissions
  - ◆ Shob developer ke deployment er permission deyar dorkar nai.
- Stay calm, even if something breaks

## Software Deployment Methodology

### Devops

- Methodology, set of best practices
- Primary Goals -
  - ◆ Shorten delivery times
  - ◆ Faster development
- 7 steps -
  - ◆ Coding
  - ◆ Building
  - ◆ Testing
  - ◆ Packaging
  - ◆ Releasing
  - ◆ Configuring
  - ◆ Monitoring

→ Depends on two frameworks -

◆ **Continuous Integration (CI)**

- New code is integrated in a shared repository on a regular basis. Sometimes it can be several times per day.
- Proti week / proti din new code repository te push kori.
- After the integration, the new code should be tested through an automated build process.

◆ **Continuous Deployment (CD)**

- Software Release Strategy ta ekta automated test er bhitor diye pass korbe before releasing into every production environment.
- Jokhonni kono change ashbe, ei whole change ta ekta whole setup environment er moddhe diye jabe with multiple sets of tests. Ei test er pore sheta production environment e push korbo.

### **Release vs Deployment**

- Ekta existng code ase with version number 4.0.1. New feature implement kore ekta version number dilam 4.0.2. This is called Software Release.
- Ei 4.0.2 version ta jokhon live ba kono ekta environment e push korbo, tokhon etake bolbo Software Deployment.