Fake Social - Decentralized Timeline

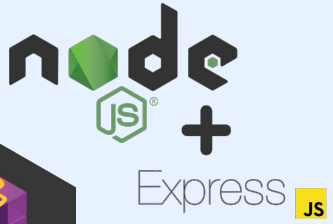
Large Scale Distributed Systems

Bruno Rosendo, up201906334
Henrique Nunes, up201906852
João Mesquita, up201906682
Rui Alves, up201905853

Technologies used

Frontend

Backend



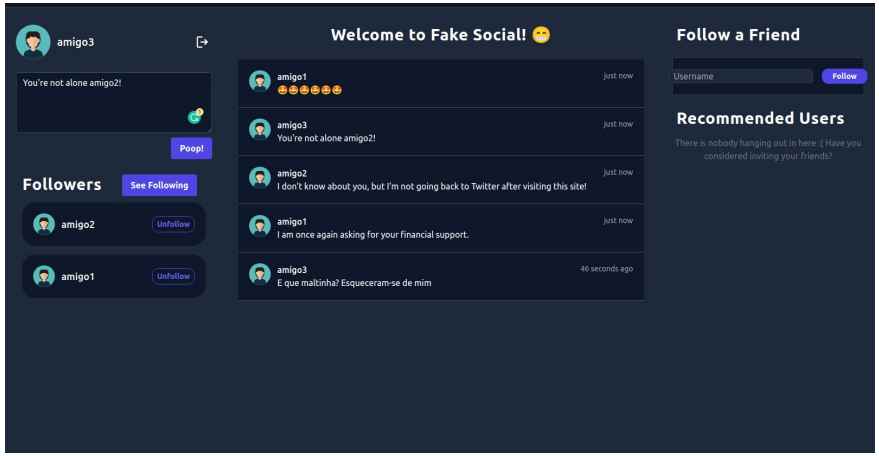
Application design

Backend

- API to handle client requests and execute the respective protocols.

Route	Method	Description
/	GET	Lifecheck
/auth/me	GET	Current user logged in
/auth/register	POST	Register new user
/auth/login	POST	Login user
/auth/logout	POST	Logout user
/posts/new	POST	Create new post
/posts/sse	GET	Create an event stream for SSE
/users/:username/timeline	GET	Get user timeline
/users/:username/info	GET	Get user info
/users/:username/recommended	GET	Get recommended users
/users/:username/follow	POST	Follow user
/users/:username/unfollow	POST	Unfollow user

Frontend



- Login/Register.
- Side panel with followers and following.
- Side panel with recommended users.
- User timeline & User profile.

Libp2p

- **Transport (TCP, Noise, Mplex)**
  - **Encrypted:** Connections must be end-to-end encrypted, so we use noise protocol of chainsafe/libp2p-noise.
  - **Multiplexed:** Multiplex multiple reliable streams over a single reliable transport connection.
- **Peer/Content Routing**
  - **Kademlia DHT:** Distributed hash table used for content and peer routing.
- **Peer Discovery**
  - **Bootstrap:** Nodes with known addresses that accept connections from the application nodes and allow them to join the network and find other existing and upcoming nodes.
  - **MulticastDNS:** Used to find nodes in our network and that are not connected to the bootstrap nodes.
- **Publish/Subscribe**
  - **Gossip PubSub:** Peers gossip about messages they have recently seen through lightweight metadata messages. This allows other nodes to notice whether they missed a message on the full-message network.

Data and Storage

USER RECORD

{

"followers":["amigo2","amigo1"],

"following":[“amigo3”],

"posts":[{"username":"amigo3","text":"E que maltinha? Esqueceram-se de mim\n","timestamp":1670933663549}]

}

Local Storage:

- Each node locally saves its own and followed users’ records.

DHT:

- All registered accounts.
- Trackers to indicate which peers provide a desired Record.

Garbage Collection

STORAGE LIMIT

Each user only stores up to 100 posts for each followed user. When this limit is reached, the older post is removed.

TIME LIMIT

Periodically, each node checks for posts older than 24 hours and removes them from local storage.

Pub/Sub Topics

FOLLOWED

Notify followers upon a follow action.

UNFOLLOWED

Notify followers upon an unfollow action.

WAS FOLLOWED

Notify followers when the user was followed.

WAS UNFOLLOWED

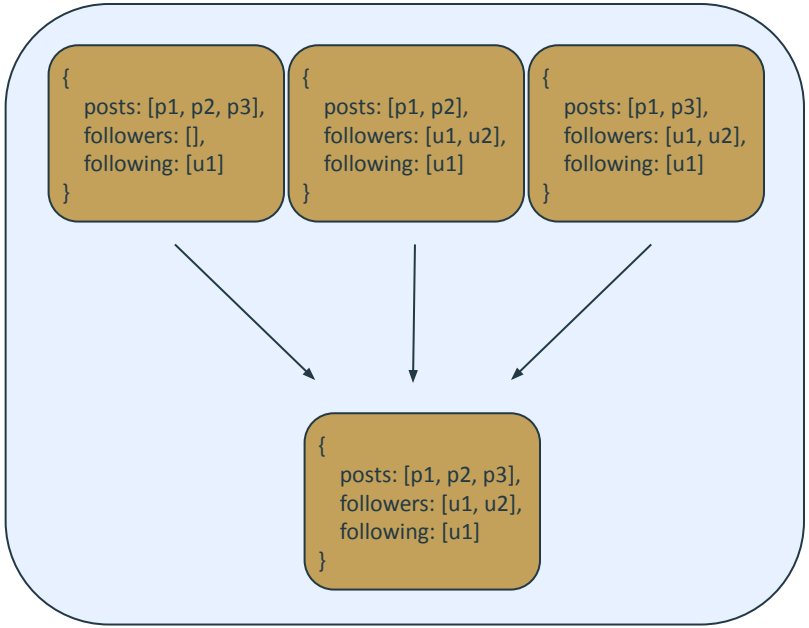
Notify followers when the user was unfollowed.

POST

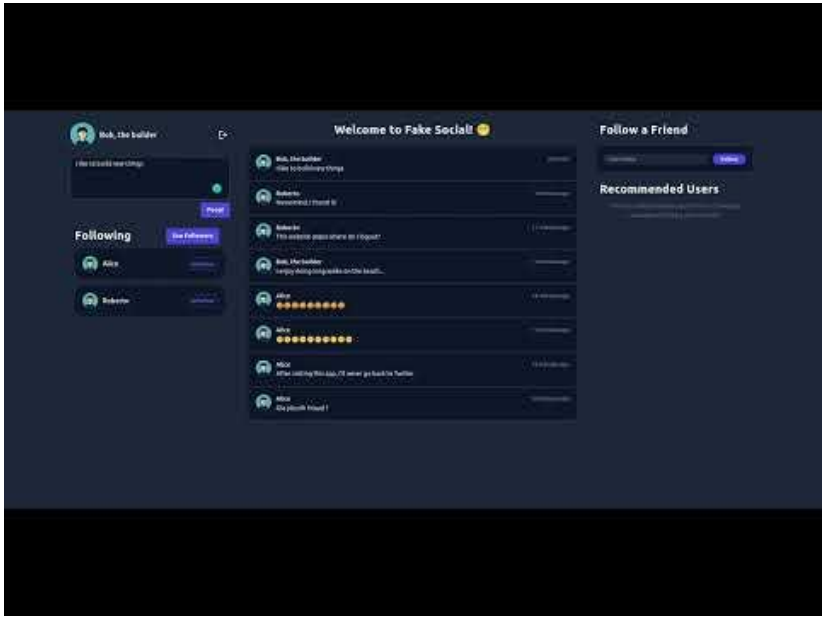
Notify followers when the user has a new post.

Records merging algorithm

- When requested, a peer tries to fetch a user record (posts, followers and following) from a maximum number of providers.
- Merge the fetched profiles by using a Majority Voting Algorithm for each field.



DEMO


Welcome to Fake Social!
Follow a Friend
Recommended Users
Following

Future Work

1

New topic for User Advertisement

Alternative way of discovering other users to avoid manually inserting the username.

2

UI/UX Upgrade

Improve the user experience, based on the observations made by the users.

3

Server-Sent Events

Add automatic updates for other events, such as follow actions and suggested users.

4

Improve the Merging Algorithm

Test our system under more extreme conditions to validate its stability.