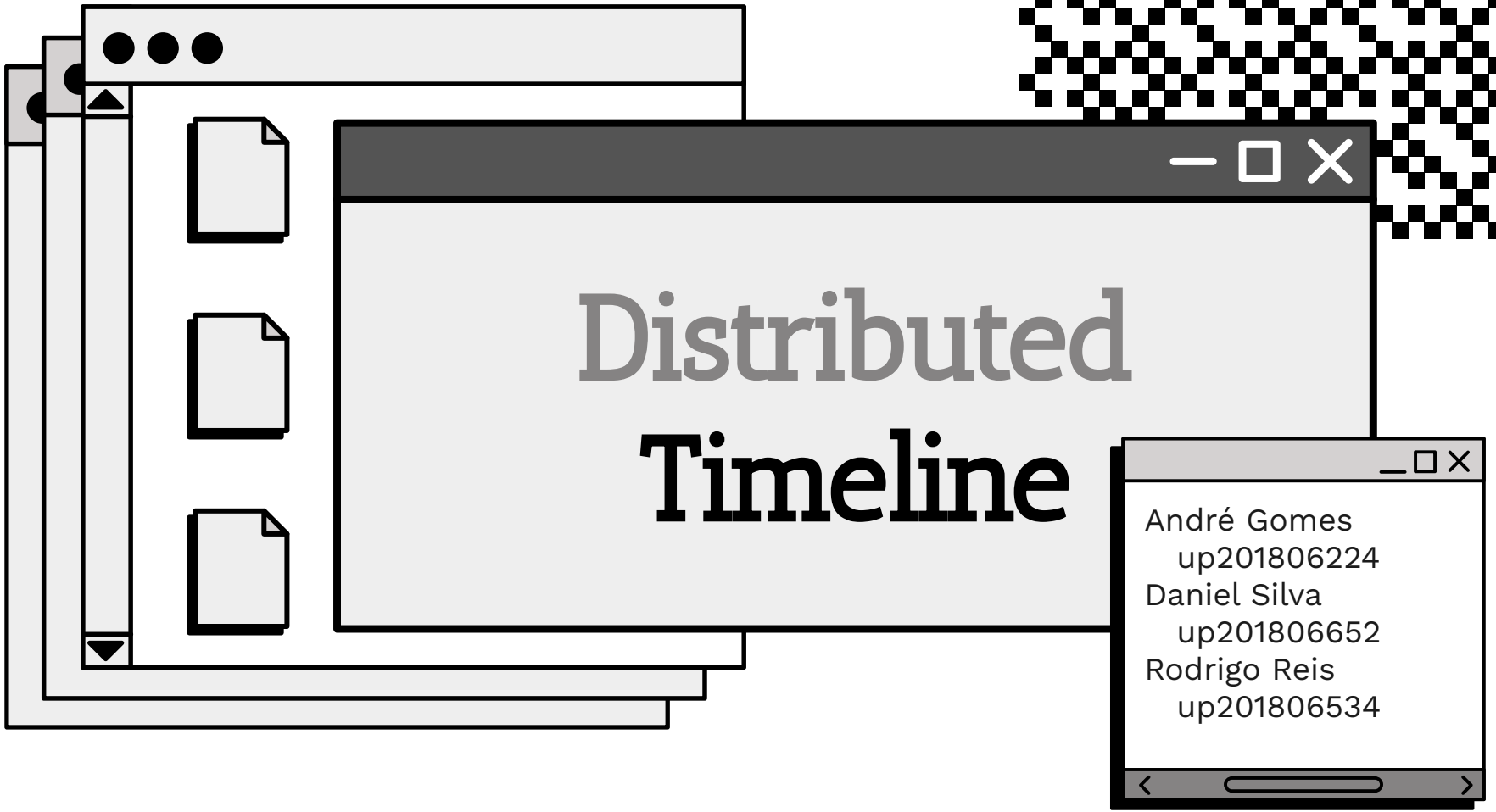

Distributed
Timeline
André Gomes
up201806224
Daniel Silva
up201806652
Rodrigo Reis
up201806534

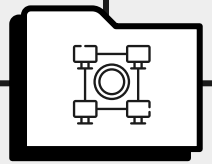
Problem Description

Specification #01

Decentralized timeline service

Specification #02

Users possess an identity and publish messages



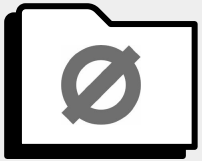
Specification #03

Users can subscribe to other users, having access to their timeline

Specification #04

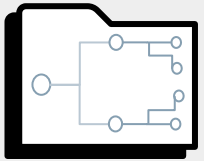
Users store and forward information relative to **subscribed** users

Options Considered



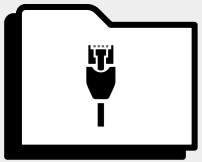
ZeroMQ

Need to design and create protocol, done in **SDIS**



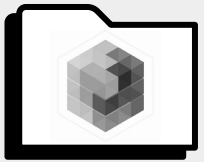
Kademlia

DHT using asyncio and python sockets



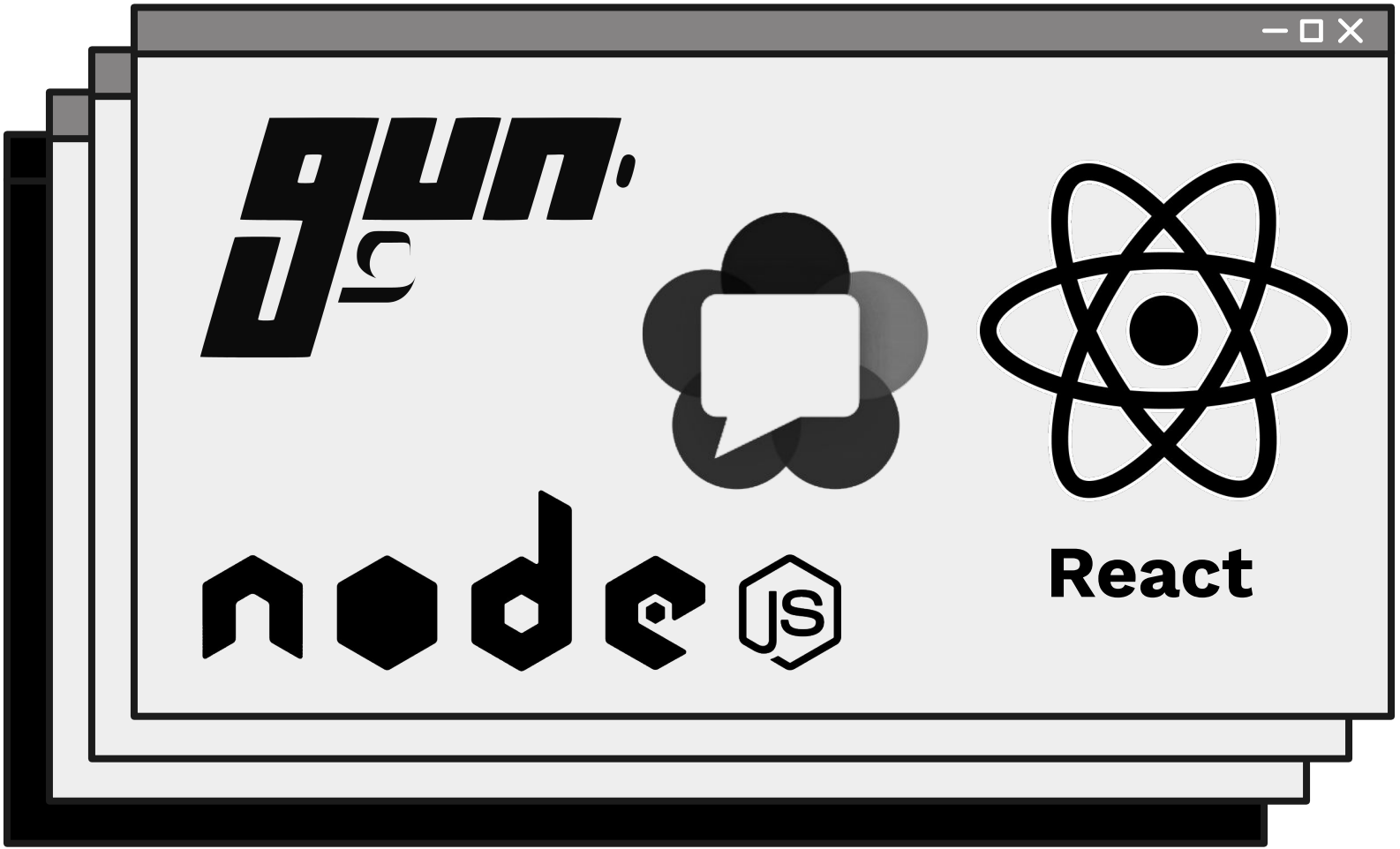
PeerJS

WebRTC wrapper to communicate between peers

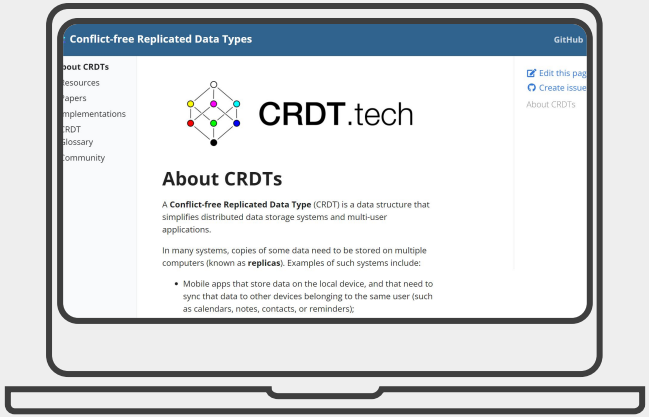


LibP2P

Variety of **Modules** implemented in Go/JS

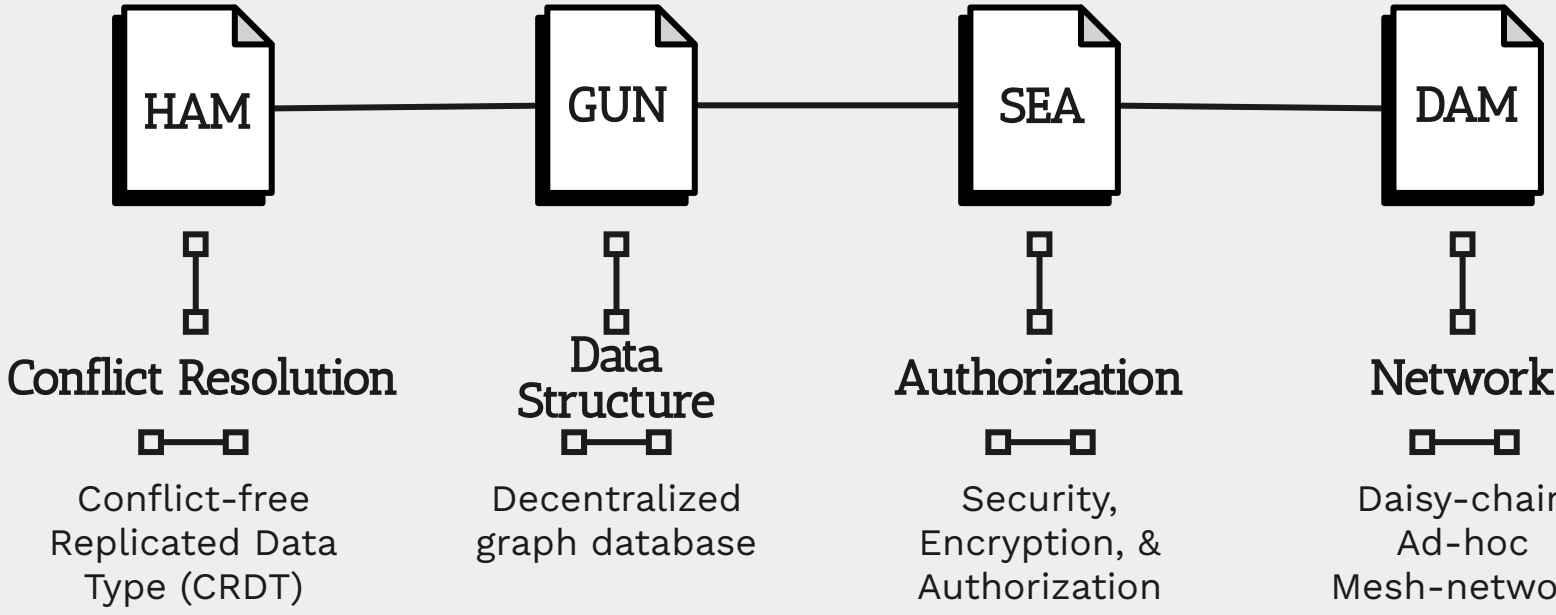

gun
node
JS
React

Conflict-free Replicated Data Types



<< For more details on CRDTs, you can read the article "Conflict-free Replicated Data Types" by Nuno Preguiça, **Carlos Baquero**, and Marc Shapiro (2018) >>

Gun Ecosystem



CAP Theorem

Consistency

Gun opts for eventual consistency instead of strong consistency

Availability

All requests retrieve a **non-error** response

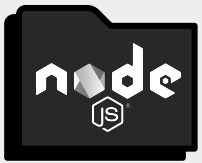
Partition Tolerance

Peers can be **offline** and still operate over their data

Gun Network

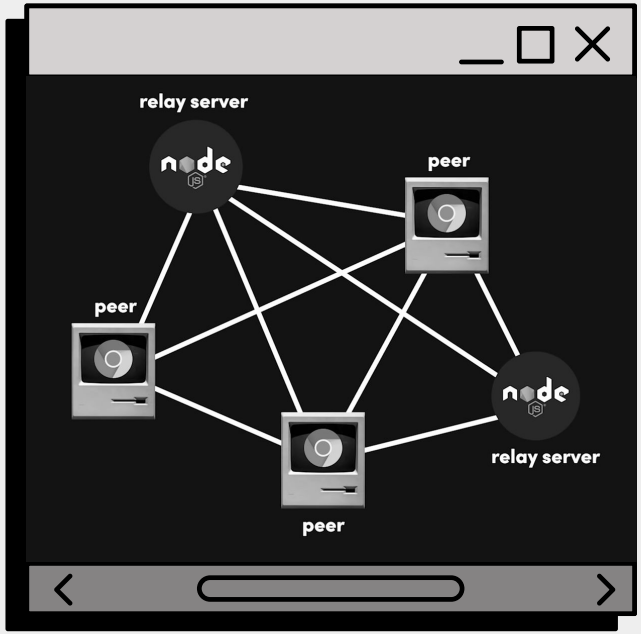
Peer

Keep localized graph in local Storage



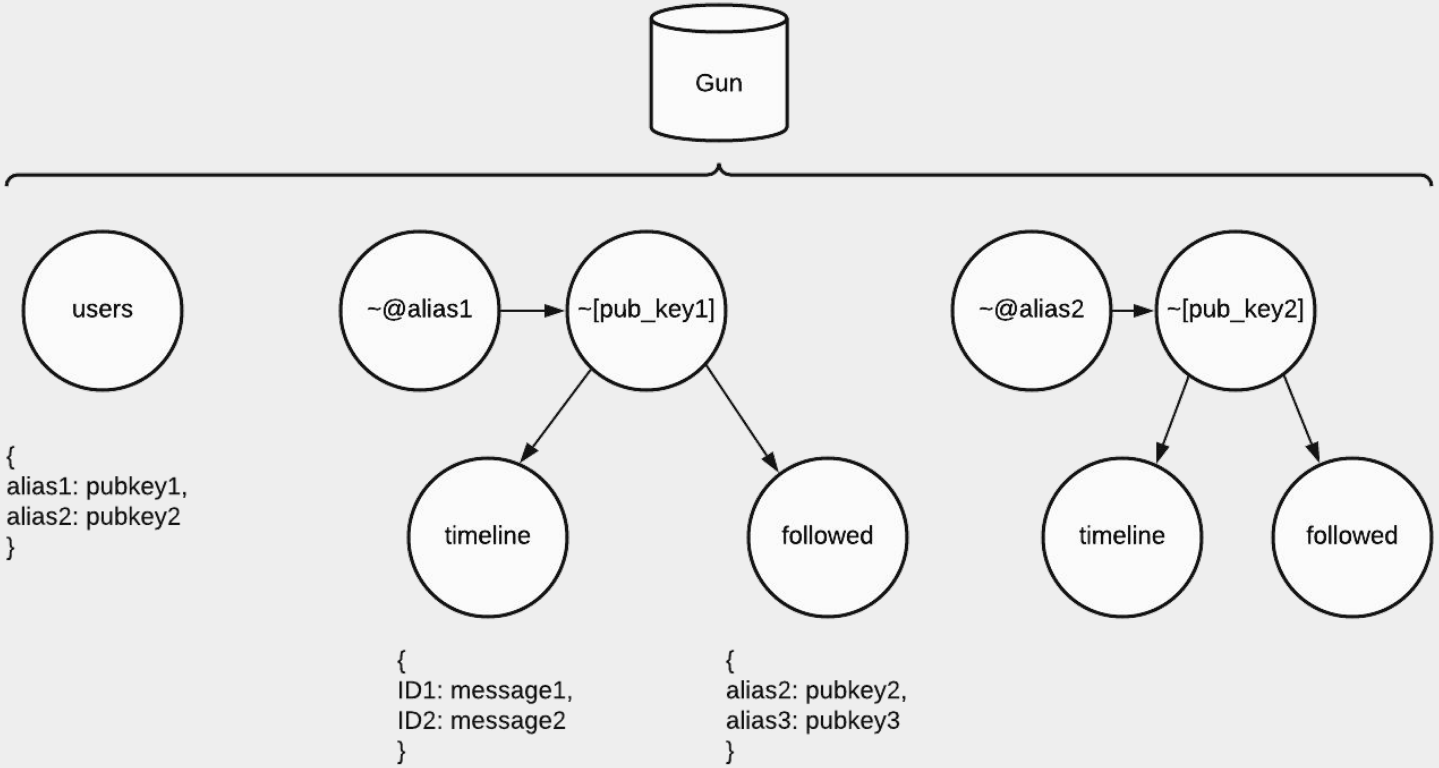
Relay Server

Persistent storage, serves as bootstrap



Source: GUN Decentralized Graph DB in 100 Seconds - https://youtu.be/oTQXzhm8w_8 (0:30)

Graph Structure

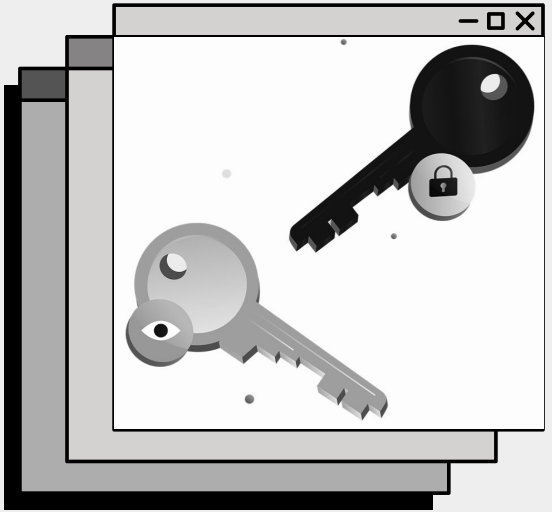


Authentication - SEA

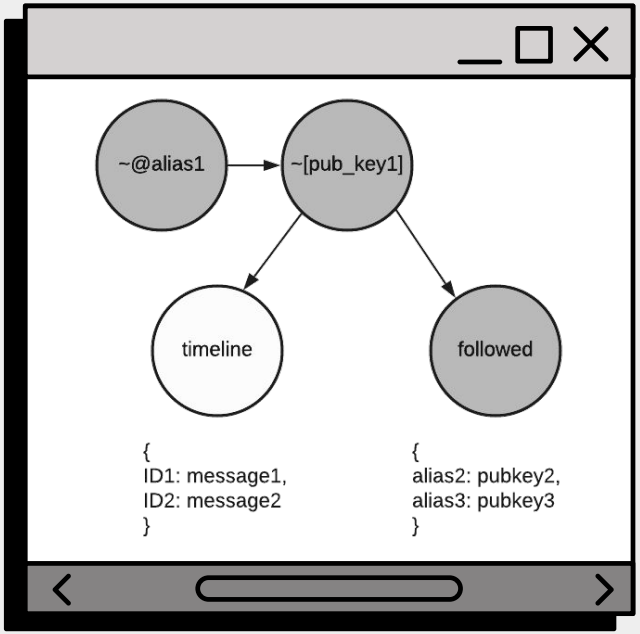
A user can create an account with an **alias** and a **password**.

If the alias is not taken, the user is created and authenticated. A **private** and **public** key pair is created and the user publishes his alias and pub key to the users node.

The user information is stored in the browser's session storage.



Timeline



Post a message

A new node is added to the timeline with its content and ID as Date.now()

Delete a message

Reference to the node becomes null, possible to restore message

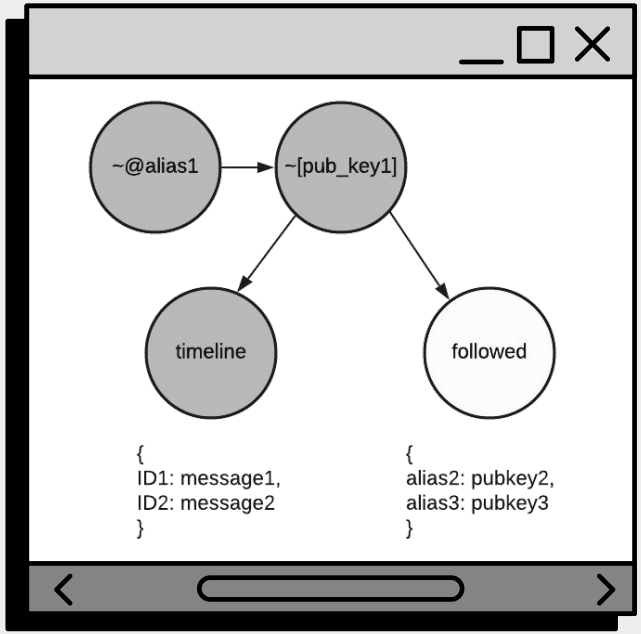
Followed

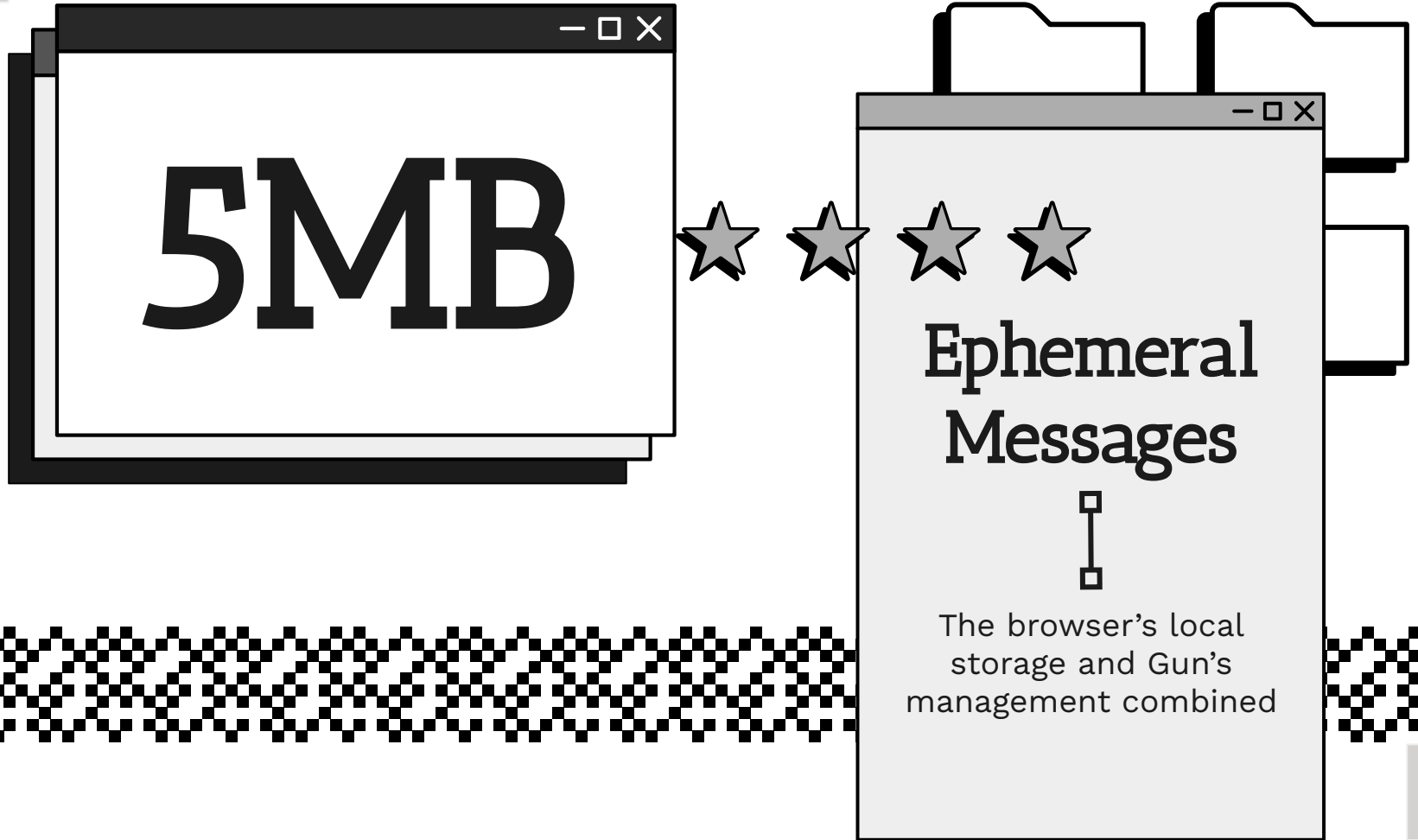
Following a user

A subscription to that user's timeline node is performed

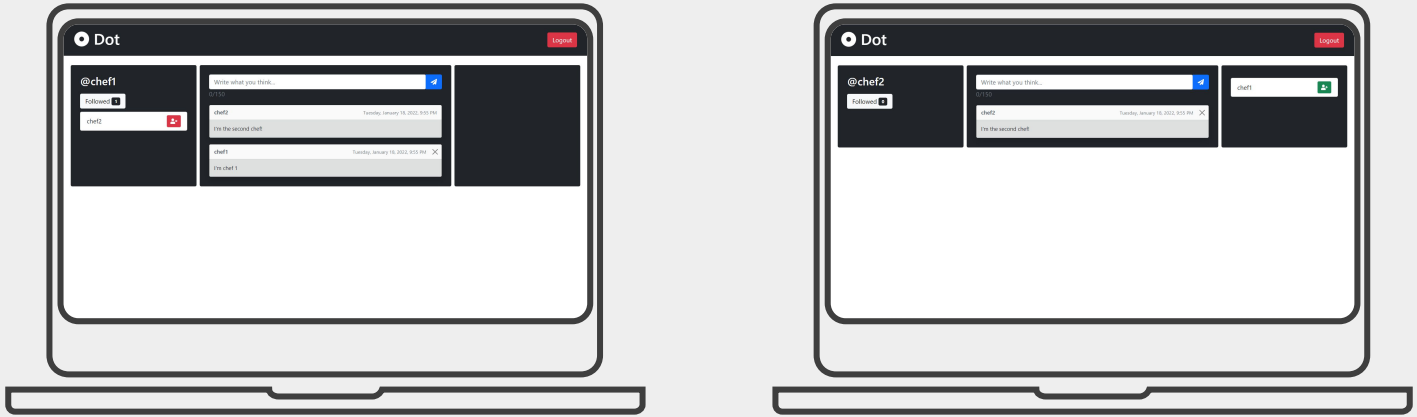
Followed node

Contains nodes with followed user's information




5MB
Ephemeral Messages
The browser’s local storage and Gun’s management combined

Realtime Updates

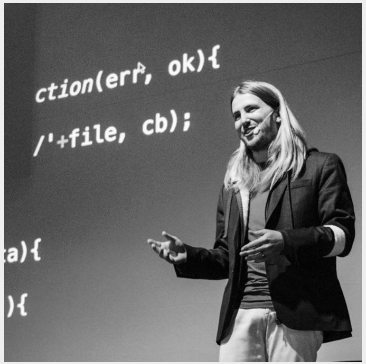


Due to GUN's use of **PTSD** (Performance Testing Speed Development), a strong focus on benchmarking ops/s makes real time updates possible

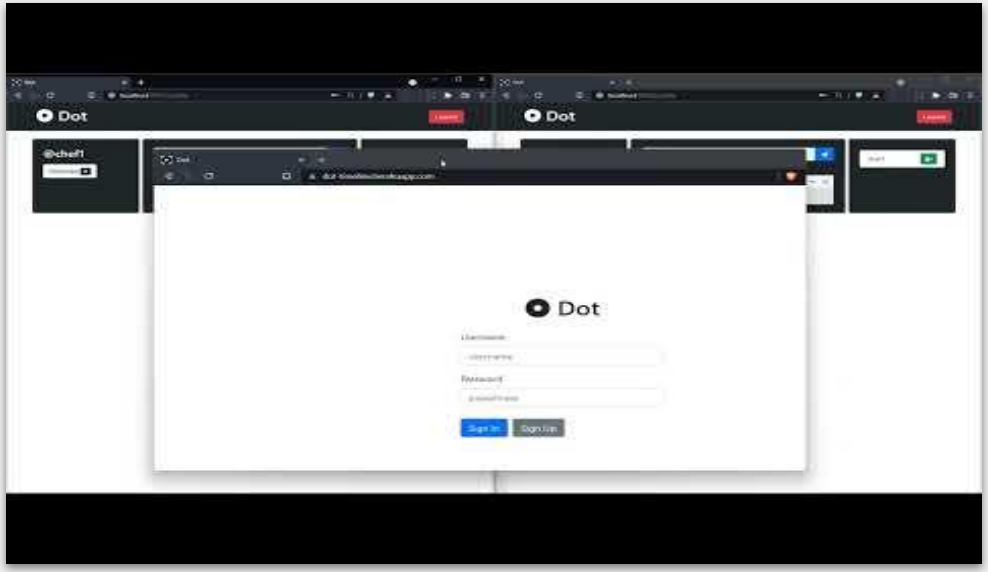
Thanks!

Do you have any questions?

[up201806224, up2018066552, up201806524]@edu.fe.up.pt



Special thanks to Mark Nadal and others on https://gitter.im/amark/gun

https://dot-timeline.herokuapp.com/

Demo