Hubs

Hubs are a distributed network of servers that store and validate Farcaster data.

A computer can run software to become a Farcaster hub. It will download onchain Farcaster data from Ethereum and offchain Farcaster data from other Hubs. Each Hub stores a copy of all Farcaster data which can be accessed over an API.

Hubs let you read and write data to Farcaster, and anyone building a Farcaster app will need to talk to one. Anyone can run a Hub on their laptop or on a cloud server. A full guide to setting up and running a Hub is availablehere .

Design

A Hub starts by syncing data from Farcaster contracts on the Optimism blockchain. It becomes aware of every user's account and their account keys.

The lifecycle of a Farcaster message looks like this:

1. Alice creates a new "Hello World!" message.
2. Alice (or her app) signs the message with an account key.
3. Alice (or her app) uploads the message to a Hub.
4. The hub checks the message's validity.
5. The hub sends the message to peer hubs over gossip.

Validation

Alice's message is validated by checking that it has a valid signature from one of her account keys. The hub also ensures that the message obeys the requirements of the message type. for example, a public message or "cast" must be less than 320 bytes. Message type requirements are specified in detail in the protocol spec

Storage

Alice's message is then checked for conflicts before being stored in the Hub. Conflicts can occur for many reasons:

1. The hub already has a copy of the message.
2. The hub has a later message from Alice deleting this message.
3. Alice has only paid rent for 5000 casts, and this is her 5001st cast.

Conflicts are resolved deterministically and asynchronously using CRDTs. For example, if Alice has no space to store messages, her oldest message will be removed.

Replication

Hubs sync using a two-phase process - gossip and diff sync. When a Hub receives and stores a message, it immediately gossips it to its peers. Gossip is performed using libp2p's gossipsub protocol and is lossy. Hubs then periodically select a random peer and perform a diff sync to find dropped messages. The diff sync process compares merkle tries of message hashes to efficiently find dropped messages.

Consistency

Hubs are said to have strong eventual consistency. If a Hub is disconnected, it can be written to it and will recover safely when it comes online. This is unlike blockchains where a node that is disconnected cannot confirm transactions. The downside is that messages may arrive out of order. For example, Bob's reply to Alice might appear before her "Hello World" message.

Peer Scoring

Hubs monitor peers and score their behavior. If a peer doesn't accept valid messages, falls behind, or gossips too much it may be ignored by its peers.

Implementations

- Hubble
-
    - a Hub implementation in Typescript and Rust

FAQ

1. Why should I run a Hub?

You may need a Hub if you are building an app that wants to read or write Farcaster data.

1. Are there rewards for running a Hub?

Farcaster does not provide rewards for Hubs and does not plan to. Warpcast, a company building on Farcaster, gives Hub runners minor rewards but may change this in the future.