FAIRNET: DECENTRALIZED SOCIAL MEDIA PLATFORM

CENG 3550, DECENTRALIZED SYSTEMS AND APPLICATIONS

Tuna Güven, Esin Güler
tunaguven@posta.mu.edu.tr
esinguler@posta.mu.edu.tr

Tuesday 20th January, 2026

Abstract

This report details the design and implementation of "FairNet," a decentralized social media application developed as a final project. The project addresses the limitations of traditional Web 2.0 social platforms, specifically focusing on censorship resistance, data ownership, and fair revenue distribution for creators. The system utilizes a hybrid architecture combining the Ethereum blockchain (Hardhat) for logic and transaction settlement, and IPFS (InterPlanetary File System - Kubo) for distributed content storage. Key features include Self-Sovereign Identity (SSI), an NFT-based marketplace, encrypted content delivery (Token-Gating), and a zero-fee tipping mechanism. We demonstrate data consistency and network persistence through a custom Host-Client topology using Tailscale, introducing a novel "Cooperative Pinning" protocol to ensure content survival.

1 Introduction

Traditional social media platforms rely on centralized servers, creating single points of failure and giving platform owners absolute control over user data. This centralization leads to issues regarding censorship, privacy violations, and unfair monetization models where intermediaries extract significant fees from content creators.

We propose a Web 3.0 alternative: a Decentralized Social Media DApp. By leveraging blockchain technology and distributed file systems, our application ensures that users retain ownership of their content and identity. The platform is built upon the Ethereum Hardhat environment for smart contract execution and IPFS for storage, ensuring censorship resistance. Furthermore, we implement a fair economic model where creators receive 100% of the tips (likes) directly from their audience, eliminating platform fees.

This report outlines the fundamentals of the technology used, the system architecture including the "Cooperative Pinning" innovation, implementation details of the smart contracts and frontend, and the experimental results of our network persistence tests.

2 Fundamentals

This section provides an overview of the core technologies utilized in the FairNet ecosystem.

2.1 Blockchain and Smart Contracts

The backbone of FairNet's logic is the Ethereum Blockchain. We utilize Smart Contracts written in Solidity to handle the "Table of Contents" for the application. This ensures that the registry of posts is immutable and tamper-proof. Specifically, we employ the ERC-721 standard to treat posts as Non-Fungible Tokens (NFTs), allowing for ownership transfer and monetization.

2.2 IPFS (InterPlanetary File System)

While the blockchain stores logic, storing images and large text data on-chain is prohibitively expensive. FairNet uses IPFS, a peer-to-peer hypermedia protocol. We utilize **Kubo**, the Go implementation of IPFS. In IPFS, files are addressed by their content hash (CID) rather than their location. This ensures that as long as one node in the network holds the file, it remains accessible to all.

2.3 Self-Sovereign Identity (SSI)

We implemented a Self-Sovereign Identity system to establish trust. User profiles (Bio, Display Name, Avatar CIDs) are not stored on a central database but are recorded directly on the blockchain. This allows users to cryptographically prove they are the owners of their profiles using their wallet keys, creating a root of trust without a central authority.

3 RELATED WORKS

Decentralized social media is an active area of research. Early platforms like Steemit introduced the concept of blockchain-based blogging but suffered from heavy on-chain data requirements. Newer protocols like Lens Protocol attempt to create a social graph on the blockchain.

FairNet differs from these by focusing on a lightweight, "View-to-Host" persistency model. Unlike Filecoin, which requires financial incentives for storage, FairNet relies on social cooperation—users inherently host the content they are interested in viewing, creating a natural redundancy for popular content via our Cooperative Pinning mechanism.

4 SYSTEM PROPOSAL / Architecture

The application follows a hybrid decentralized architecture. To overcome the high cost of storing large media files on the blockchain, we utilize IPFS for heavy static assets while using the Ethereum blockchain for state management.

The core innovation of our architecture is the "View-to-Host" protocol, which ensures data persistence through user participation. Figure 1 details the sequence of events between a Host (User A) and a Guest (User B) connected via a secure Tailscale mesh network.

As illustrated in Figure 1, the process follows these steps:

1. **Initial Upload:** The Host (User A) uploads a file (e.g., "Hello.png"). This is stored on User A's local IPFS node.

2. **Discovery:** The Guest (User B) queries the Blockchain to get the Content Identifier (CID).

3. **P2P Fetch:** User B's IPFS node requests the hash. The "Swarm Fetch" occurs, downloading the file from User A via the Tailscale tunnel.

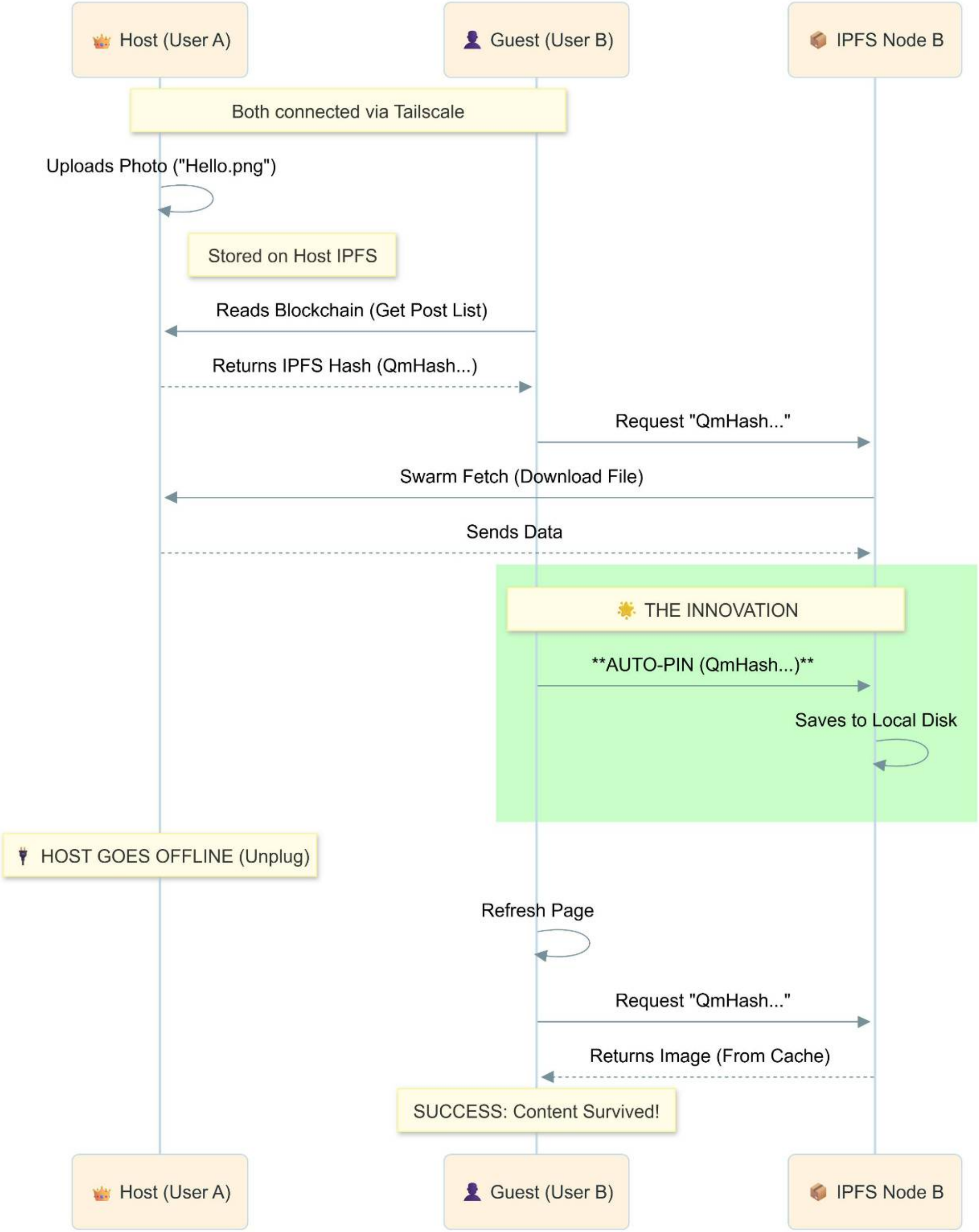
Host (User A)
Guest (User B)
IPFS Node B
Both connected via Tailscale
Uploads Photo ("Hello.png")
Stored on Host IPFS
Reads Blockchain (Get Post List)
Returns IPFS Hash (QmHash...)
Request "QmHash..."
Swarm Fetch (Download File)
Sends Data
THE INNOVATION
AUTO-PIN (QmHash...)
Saves to Local Disk
HOST GOES OFFLINE (Unplug)
Refresh Page
Request "QmHash..."
Returns Image (From Cache)
SUCCESS: Content Survived!
Host (User A)
Guest (User B)
IPFS Node B

Figure 1: Sequence Diagram of the Cooperative Pinning Protocol

4. **The Innovation (Auto-Pin):** Once viewed, User B's client automatically pins the content (`ipfs.pin.add`). User B is now a co-host.
5. **Fault Tolerance:** If User A goes offline (Unplug), User B can still access the content from their local cache/pin, ensuring the data survives.

5 IMPLEMENTATION

The prototype system works as a distributed network of nodes. We used **Hardhat** to execute our own smart contracts and **Kubo (Go-IPFS)** for storage.

5.1 Wallet Integration

Users create accounts and login using their Ethereum wallets via the **Metamask** extension. The wallet address serves as the user's identity. We utilized test accounts imported via private keys for demonstration.

5.2 NFT Minting & Marketplace

Creators can mint NFTs on their posts (music, painting, etc.). A smart contract function links the IPFS CID to a token ID. Creators can then sell these NFTs for a desired amount of ETH.

5.3 Encrypted Posts (Token-Gating)

For "Subscriber Only" content, we implemented client-side encryption:

1. **Encryption:** Media is encrypted with AES before IPFS upload.
2. **Token-Gating:** The decryption key is protected. The app checks if the user owns the specific NFT.
3. **Decryption:** If the user owns the NFT, the content is decrypted locally.

5.4 Zero-Fee Tipping

We replaced the "Like" button with a financial transaction. Every like triggers a **0.01 ETH** transaction via Metamask. This is a direct peer-to-peer transfer with 0% platform fees, realizing a fair revenue system.

5.5 Network Topology (Tailscale)

To achieve data consistency, two computers were connected via **Tailscale** to simulate a wide-area network. One acted as the Server (Host) and the other as Client. We used IPFS IDs and Tailscale IPs to manually peer the nodes, creating a private swarm for testing.

6 RESULTS and DISCUSSION

The primary success metric was the "Unplug Test." In our experimental setup, User A hosted an image and User B viewed it. After User B's client auto-pinned the content, we physically disconnected User A from the network. **Result:** User B could still load and view the image. This confirms that the "Cooperative Pinning" architecture successfully decentralizes the hosting burden and prevents data loss when the original creator disappears. The 0% fee tipping system was also verified with multiple test transactions on the Hardhat local network.

7 CONCLUSION

This project successfully demonstrates a functional prototype of a Web 3.0 social media platform. By integrating Ethereum Smart Contracts for logic and IPFS for storage, we achieved censorship resistance and data sovereignty. The implementation of SSI, Token-Gated encryption, and a zero-fee tipping economy offers a viable alternative to the exploitative models of Web 2.0. The successful networking test via Tailscale proves the system's capability to maintain data consistency across distributed nodes.

ACKNOWLEDGEMENT

We would like to thank our course instructors for their guidance on Distributed Systems concepts.

References

[1] Nakamoto, Satoshi. Bitcoin: A peer-to-peer electronic cash system. Manubot, 2019.

[2] Benet, Juan. "IPFS-content addressed, versioned, p2p file system." arXiv preprint arXiv:1407.3561 (2014).

[3] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." White paper 3.37 (2014).

[4] Entriken, W., et al. "EIP-721: Non-Fungible Token Standard," Ethereum Improvement Proposals, no. 721, Jan. 2018.