Emerald: A Decentralized Imageboard Platform

Alexander Sellström
alexander@sellstrom.me

Abstract

The author introduces Emerald, a decentralized imageboard platform powered by a delegated-proof-of-stake blockchain and a decentralized storage and delivery network. Emerald provides a marketplace for network participants to sell their knowledge of the state of the imageboard to end-users. Emerald also enables the creation of boards where humans are only allowed one account per self-sovereign or decentralized identity while preserving their anonymity with a zero-knowledge proof. Boards on Emerald can choose to have a decentralized autonomous organization for governance and moderation. Emerald's blockchain component will be implemented as an application-specific blockchain using the Cosmos SDK.

1 Introduction

Emerald is a decentralized imageboard platform that intends to disrupt the way people receive and impart information and ideas. Today our access to social media, our means of communication, is at the discretion of a handful of big tech companies. Platforms for hundreds of millions or even billions of users are heavily regulated or censored by a small number of people that have not been elected by their users.

It is not enough to live in a country with freedom and democracy if you also choose to surrender your freedom by using oppressive and centralized platforms. Centralized platforms in cyberspace can be likened to totalitarian states in the real world. It is time we had a social media platform that embodies Western ideals like democracy and freedom of speech.

2 General Structure

On Emerald, the service providers (back-end servers) can be consumer-grade desktop computers belonging to any private person, as opposed to a corporate data center. The blockchain itself only stores the hashes of media posted on Emerald due to bandwidth constraints inherent to the Ignite (formerly Tendermint) consensus algorithm. The media itself is stored by service provider nodes that make the content available to end-users for a small fee.

Emerald distinguishes itself from most other blockchain projects in that it has an application-specific blockchain. Other blockchains are often general-purpose blockchains that allow smart contracts to be deployed on-chain, enabling all kinds of decentralized applications to be built on top of them. Smart contracts increase complexity and attack surface which exposes users to risks such as fraudulent or even buggy smart contracts.

Emerald also distinguishes itself from decentralized (federated) platforms like Nostr, Bluesky, and Mastodon, in that the content is globally consistent (*logically* centralized) across all service nodes/servers. The shortcomings of federated protocols boil down to the fact that each individual server is sovereign and essentially centralized, and therefore inherits many of the problems of traditionally centralized networks. Blockchain technology also enables the creation of an incentive layer that compensates nodes that help support the network, and a disincentive layer to deter any would-be malicious nodes.

3 Posts and Threads

Posts and threads are identical in the way they're represented on-chain, but the first post with a certain thread identifier is treated as the original post by the application logic. A post is a transaction type that simply contains a thread identifier and a

content identifier (CID). The content identifier is a hash of the combined hashes of the text and any attached media files, also known as a root hash. A root hash of a hash list is used instead of the individual hashes to save precious bandwidth during the consensus process.

4 Content Storage and Delivery

Media in posts, such as text and images, are not stored on the blockchain. The blockchain merely stores content identifiers for the content which is handled by a content-addressable decentralized content delivery network (dCDN) similar to IPFS, which also uses libp2p. Blockchain full nodes that also want to be able to receive and broadcast media must also be network participants in the dCDN.

The dCDN differs from IPFS in that it does not use a distributed hash table (DHT) to find nodes. DHT lookup is a slow process involving finding a node that possesses a file so that it can voluntarily share it with you. Emerald's approach makes each node in the dCDN store and serve all of the files of a board. Clients only need to find service nodes once. When new CIDs are added to the blockchain, the client can simply request the corresponding files from the service nodes they're connected to.

The network is divided up into sub-networks, one for each board so that file nodes are not forced to handle data they are not interested in. This allows nodes with limited disk space to provide services. A defining characteristic of Imageboards is that threads disappear when new ones are created. A limit on the maximum number of active threads allows for very modest storage space requirements for service nodes in the range of 50-150 gigabytes for one board. In addition, nodes can opt out of hosting boards that are known to contain illegal or otherwise objectionable content.

After a post has been included on the blockchain, the node(s) that originally broadcast the transaction can start broadcasting the corresponding files to their neighbors that are also part of that board's dCDN sub-network. Nodes are kept honest by the threat of being blocked by their neighbors if they do not at least offer to transmit a file that their neighbors have seen.

5 On-chain Node Registry

When a node or a client needs to find nodes in the network they can consult an on-chain node registry. Entries in this registry specify the IP address of a node along with details such as which dCDN sub-networks they are part of, as well as the prices of their services for any clients that wish to employ them.

This registry facilitates the onboarding of new nodes for both the blockchain and the dCDN. The registry also allows clients to find suitable service nodes for the boards they are interested in viewing.

6 Service Contracts

If a client wishes to employ a service node a service contract is created. The service contract is cryptographically signed by both parties prior to being broadcast to show that both parties are consenting to the deal. The contents of the contract include the public keys of both parties, an expiration date, and a number of tokens held in escrow by the contract itself.

The client will send cryptographically signed messages, stating that they have received service, to the service node at regular intervals throughout the duration of the contract. The application logic dictates that the service node can broadcast these signed messages to collect the balance held in escrow. Each message works like a signed check for a fraction of the balance. All of the "checks" can be cashed using a single transaction after the contract is over.

The application logic forbids anyone other than the service node from withdrawing the tokens held in the contract to prevent the end-user from taking back the money. If the end-user's client decides that the node is no longer performing acceptably they can simply stop sending them "checks". The remaining balance in the contract is burned some fixed time after the expiration date of the contract.

With this system, malicious end-users are financially penalized if they enter service contracts that they don't intend on paying for. A rating function that takes unfulfilled service contracts into account can be used to avoid selecting malicious service nodes.

When interacting with other blockchains like Ethereum, users that cannot run their own node must typically rely on centralized API services. This defeats much of the purpose of using a decentralized network like Ethereum. In August 2022, the US Treasury put a ban on the Tornado Cash smart contract on Ethereum. This caused US-based Ethereum API service providers like In-

fura to block users all over the world from using it. Such censorship would not work very well on Emerald, since the user's client can simply rent a new service node if one starts acting up.

7 On-chain Governance

Emerald supports modifying protocol settings and the creation of new boards through on-chain governance. Anyone may submit a protocol modification proposal, for a fee, which will then be voted on by token holders using their tokens which are returned to them after the referendum has concluded.

8 Boards

Auctions will be held periodically for the leasing of board slots in a manner most similar to parachain auctions on the Polkadot and Kusama networks. Auctions are held for a week during which token holders may submit a proposal for a board containing a name and board-specific settings such as the maximum file size for images attached to posts. The proposal that receives the most backing will then become a board for the duration of the lease, at the end of which the board may have its lease extended in a new board slot auction. Whether or not the tokens used in the auction should be returned to their owners is not yet decided. Returning the tokens, effectively turning the auction into a vote, will likely lead to a higher turnout.

To combat bot spam and flooding, boards can require their posters to pay for the privilege to post there. It would also be possible for boards to require that users have a self-sovereign identity, and only allow one account per identity, to make it even harder for bots. In the future when there are robust blockchain-based decentralized identity (DID) solutions, such an identity could be transferred to Emerald using the inter-blockchain communication protocol. Such a solution could be implemented using the chip in biometric passports. The chip contains a public and private key pair that is signed by a well-known issuer.

9 Decentralized Moderation

Boards can choose different moderation models or even none at all. A complete lack of moderation is unlikely to result in a usable board, but appointing administrators with absolute authority to silence users reintroduces the very same problems inherent to centralized platforms that Emerald is supposed to solve. A solution to this would be to have a permissionless moderation council that functions in a similar manner to the consensus mechanisms that power the blockchain.

Users willing to help moderate the board and earn tokens for doing it can stake some tokens and vote on the legality of reported posts. These moderators will cast their encrypted vote and reveal the encryption key after the voting period has ended. Votes are encrypted during the voting period to prevent bots from simply copying the majority. The majority of the voting power decides the fate of the reported user and the post in question.

If a post is marked as deleted, the service nodes are no longer obligated to serve the file to their neighbors and clients. Moderators that voted against the majority are financially penalized by having a percentage deducted from their stake. This mimics how a proof-of-stake blockchain deals with malicious validators. The protocol can compensate moderators for their work by minting new tokens or paying them using posting fees collected by the board.

10 Token Mixing

A defining characteristic of imageboards is that all users are anonymous and do not have accounts. If users use the same identity for many posts they may end up "doxing" themselves. Simply generating a new key pair and transferring the tokens to the new address would leave an obvious paper trail leading back to the original one. A solution to this is to use a token mixer.

Token mixers would also work for transferring identity tokens like the ones described at the end of section 8. Users that have proven that they are unique humans using their tokenized DID could participate in a mixing session and have it exchanged for an anonymized "real human" token.