

Web3 5Degrees Protocol

- Make all DApps socializing

Background

In 2008, Satoshi Nakamoto released the Bitcoin white paper, which marked the beginning of the transfer of ownership of personal assets from an untrusted third party to the individual himself. Due to the strong correlation of BTC and other cryptocurrencies with finance, the industry was full of speculation, fraud, Ponzi, and other scams in the early days. But after decades of social experiments, the progress of the industry is obviously different.

Start from mining with the personal computer, to mining with professional mining machines. And finally, everyone works together to mine through the mining pool and then distribute the reward according to hashrate.

Also, with the launch of the Ethereum mainnet, and the establishment of protocol standards such as ERC-20, ERC-721, and ERC-1155[3]. The rise and the collapse of ICO in 2017, the rise of DeFi in 2020, and the popularity of concepts such as NFT, DAO, and Web3[1] in 2021. Which will continue to educate the market and the young generation about the ownership of assets and data and which enables individuals to pay more attention to asset ownership and the significance of data ownership. Every step of the development of the industry is reshaping the underlying structure of the value network.

Situations and thoughts

In Internet time, there isn't a unified identity system, and users have to create dozens of account and password pairs on every website or application.

Account information on every product is isolated, but as billions of people are now surfing online, its shortcomings have become more and more apparent. Even though it has been proved to be an insecure, inconvenient, and inefficient solution.

It could be possible for Internet generations to keep 70 to 80 pairs of accounts and passwords, resulting in a sharp decline in the user's experience. Some good products try to help people manage their accounts and passwords, like Okta, 1Password, and LastPass. But most importantly, users never own their online identities. Instead, they simply rent usage rights from companies and centralized entities. Therefore, they are easily faced with the risk of digital identity being hacked, manipulated, censored, or simply lost.

With the emergence of Web3.0^[1], people try to solve the unified identity problem by building products such as ENS^[2] and DAS^[2] that benchmark the DNS protocol in the Internet time. However, this solution also has its limitations. It attempts to reform the existing Internet problems by imitating the Internet DNS model and doesn't innovate from the core problems of the Internet. Let's imagine, if a function of Twitter has a certain prejudice against a group, the whole group is dissatisfied, but the group has no way to resist them, because they can't simply use something like ENS^[2], a unified alias system, to switch to another platform directly. The core problem here is that the personal identification system is not a simple alias system, it should include the individual's name and his overall social relationships. So, using an alias system like ENS^[2] seems to solve some of the problems, but not completely. And we think that we should think further about what Web3.0^[1] should bring to mankind, and what is the current predicament of Web2.0.

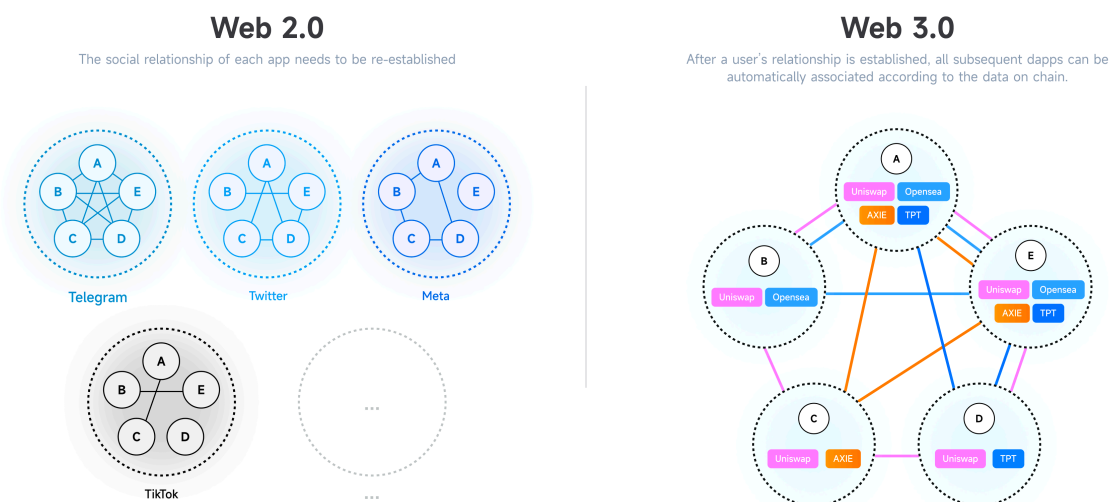
We think that the apparent problem faced by Web2.0 is a malicious monopoly, but the essence is the problem of data ownership, which causes the serious dislocations of rights and obligations of the entire Internet ecosystem.

Some rights that should belong to users are firmly controlled by the centralized platform, and individuals cannot express their oppositions. What's worse, the platform can use these data powers without restraint to suppress the budding competitors and cause their direct bankruptcy. To further prolong the cycle of malicious monopoly, if the platform is large enough, it can be a problem to society in a certain extent.

Therefore, we believe that the core mission of Web3.0^[1] should try to solve the problem of data ownership of Web2.0, and return that ownership to the users themselves through blockchain technology. All Applications share this set of data under the authorization of the user, since the underlying mechanism of user data ownership has been transferred, we can easily imagine that the various structures of the upper layer will undergo earth-shaking changes.

With the development of Web3.0^[1], the internet giants cannot take advantage of their cumulative user data to compete maliciously, and the changes of upper layers can also solve the problem of the malicious monopoly of these giants. Let the market players return to the normal competition of improving service qualities. What's a surprise, this also coincides with the current policies of various governments to combat monopoly.

The Evolution of the Web

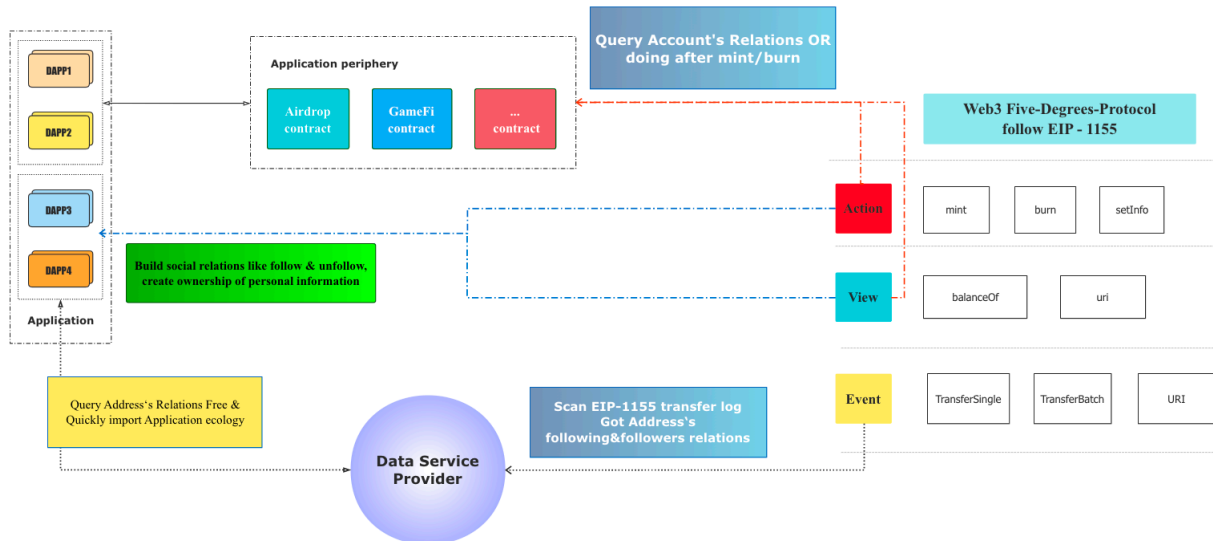


Solutions

With the core problems of Web2.0 described above, we propose a simple entity data ownership protocol and use the protocol to capitalize these core data. To make the protocol simple enough, permissionless, and composable, we choose to build the protocol under the existing ERC-1155^[3] standard. Compatible with the ERC-1155^[3] standard, meaning it will be easily implanted into any protocol or business of Web3^[1] that supports ERC-1155^[3]. The NFT generated in the protocol can be traded in any NFT market, also any smart contract that requires user relationships can track this data on chain. Any application that supports ERC-1155^[3], can easily integrate this protocol.

With the development of the Web3.0^[1] industry, entities build a relationship network by minting target NFTs, thereby forming a huge transparent, unified, and permissionless relationship network

infrastructure. All DAPPs focus on their own business, and combine the relationship network to realize various products. For example, a social module can be added to Uniswap, so that users can easily know the detailed data of the transactions of the people they follow, or a moment feature can be added to OpenSea, so that users can easily know what kinds of NFTs are their friends trading or holding, just like integrating Twitter or Discord into OpenSea, or if a social network product uses this solution to start its business, the user's friend relationship can be reconnected in Sandbox even the product failed.



Introduction to core methods

Set Entity's Info: Invoke setInfo to set entity's information

Get Entity's Info: Invoke contract's URI methods to get current entity's information

Set entity's followers limitation: invoke increaseMaxSupply method

Build the relationship between entities: Invoke mint method to mint an NFT of the following, hold his NFT means your relationship is connected

Destroy the relationship between entities: Invoke burn method to destroy a following's NFT to disconnect the relationship

Get entity's follower list: Query through check the who is holding the entity's NFT

Get entity's following list: Query through check the whose NFT is the entity holding

Web3 5Degrees Protocol

//This protocol follows ERC-1155 standard, EIP-1155 refers to: <https://eips.ethereum.org/EIPS/eip-1155> [EIP]

```
pragma solidity >= 0.8.0;
```

```
import "@openzeppelin/contracts/token/ERC1155/IERC1155.sol";
```

```
interface IFiveDegrees is IERC1155 {
```

```
    struct TokenURIInfo {  
        string name;  
        string image;  
        uint256 maxSupply;  
        string properties;  
    }
```

```
    event Mint(address indexed account, address indexed owner, uint256 tokenId);  
    event MintBatch(address[] indexed accounts, address indexed owner, uint256[] tokenIds);  
    event Burn(address indexed account, address indexed owner, uint256 tokenId);  
    event BurnBatch(address[] indexed accounts, address indexed owner, uint256[] tokenIds);  
    function setProtocolInfo(string memory name, string memory image, string memory properties) external;  
    function uri(uint256 tokenId) external view returns (string memory);  
    function baseInfo(address account) external view returns (string memory, string memory);  
    function metrics(address account) external view returns (uint256 tokenSupply, uint256 totalBalance);  
    function setPayProxy(address proxy) external;  
    function setInfo(string memory name, string memory image, string memory properties) external;  
    function increaseMaxSupply(uint newMax) external payable;  
    function decreaseMaxSupply(uint256 newMax) external;  
    function mint(address account) external;  
    function mintByOrigin(address account) external;  
    function mintBatch(address[] memory accounts) external;  
    function mintBatchByOrigin(address[] memory accounts) external;  
    function burn(address account) external;  
    function burnOrigin(address account) external;  
    function burnBatch(address[] memory accounts) external;  
    function burnBatchByOrigin(address[] memory accounts) external;  
}
```

References

[1] Web3 [<https://en.wikipedia.org/wiki/Web3>]

[2] Decentralized naming for wallets, websites, & more [<https://docs.ens.domains/ens-migration-february-2020/technical-description>]

[3] EIP-1155: Multi Token Standard [<https://eips.ethereum.org/EIPS/eip-1155>]