

# Web3 5Degrees Protocol

## - Make all DApps socializing

### Background

In 2008, the Bitcoin white paper was released by Satoshi Nakamoto, which signified that the ownership of personal assets had started to transfer 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 more than a decade of social experiments, the progress of the industry is obvious: from mining with early personal computers, to mining with professional mining machines. And finally, everyone works together to dig through the mining pool and then distribute the reward according to the hashrate.

Also, with the launch of the Ethereum main net, the establishment of protocol standards such as ERC-20, ERC-721, and ERC-1155[3], the rise and 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, constantly educating the market and the young generation about the ownership of assets and data enables individuals to pay more attention to asset ownership and attach importance to data ownership. Every step of the industry is reshaping the underlying structure of the human value network.

### Situations and thoughts

In the era of the Internet, there is no unified identity system for users, and users have to create dozens of account and password pairs on different websites or applications.

Account information on every product is isolated, but as billions of people are now surfing online, its drawbacks have become increasingly apparent, even though it has proven to be an insecure, inconvenient, and inefficient scheme.

Internet generations may have 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 likely to be faced with the risk of digital identity being hacked, manipulated, censored, or simply lost.

With the emergence of Web3.0<sup>[1]</sup>, people try to solve the problem of unified identity by creating products such as ENS<sup>[2]</sup> and DAS<sup>[2]</sup> that benchmark the DNS protocol in the Internet era. However, this solution has its limitations. It attempts to reform the existing Internet by simply imitating the Internet DNS model without innovating from the core problems of the Internet. Imagine if a function of Twitter had some kind of prejudice against a certain group, the whole group would be dissatisfied, but they could do nothing to resist it because they cannot switch directly to another platform simply by using something like ENS<sup>[2]</sup>, a unified alias system. The core problem here is that the personal identification system is not a simple alias system. It should include the uniqueness of the individual's name and his overall social relationships. So, using an alias system like ENS<sup>[2]</sup> seems to have solved some of the problems, but not completely. And we believe that we should think further about what Web3.0<sup>[1]</sup> should bring to mankind, and what the current predicament of Web2.0 is.

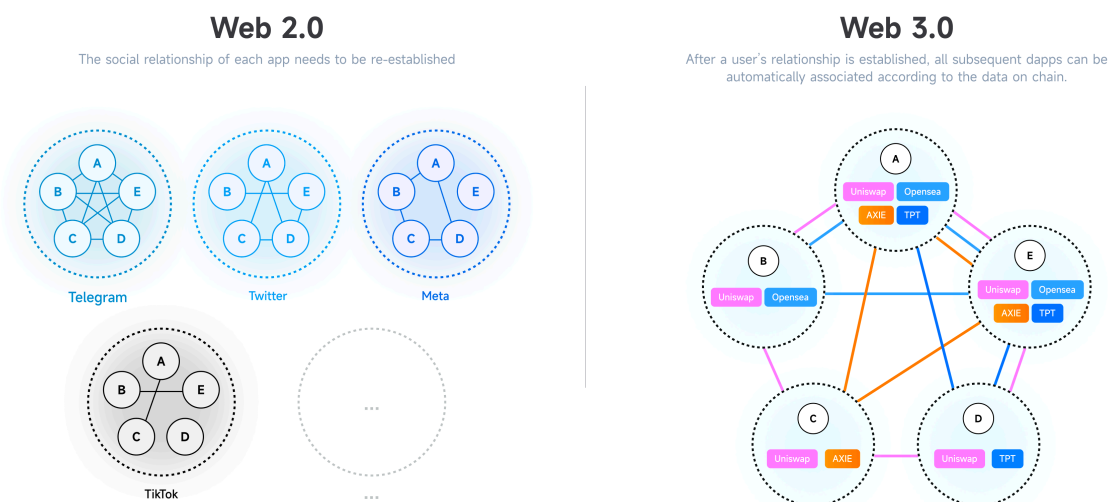
We maintain that the apparent problem facing Web2.0 is a malicious monopoly problem, but the essence is the problem of data ownership, which causes serious dislocations of rights and obligations of the entire Internet ecosystem.

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 exploit these data powers without restraint to suppress the budding competitors, further prolong the cycle of malicious monopoly, and directly lead to their bankruptcy. If the platform is large enough, it can even be a problem to the society to some extent.

Therefore, we believe that the core mission of Web3.0<sup>[1]</sup> should be trying 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 foresee that the various structures of the upper layer will go through earth-shaking changes.

With the development of Web3.0<sup>[1]</sup>, the Internet giants cannot take advantage of their cumulative user data for malicious competition. The changes of upper layers can also solve the problem of malicious monopoly from these giants, so that the market entities can return to the track of normal competition by improving service quality. What surprises us is that this also coincides with the current policies of various governments to combat monopoly and improve the status of data elements.

## The Evolution of the Web

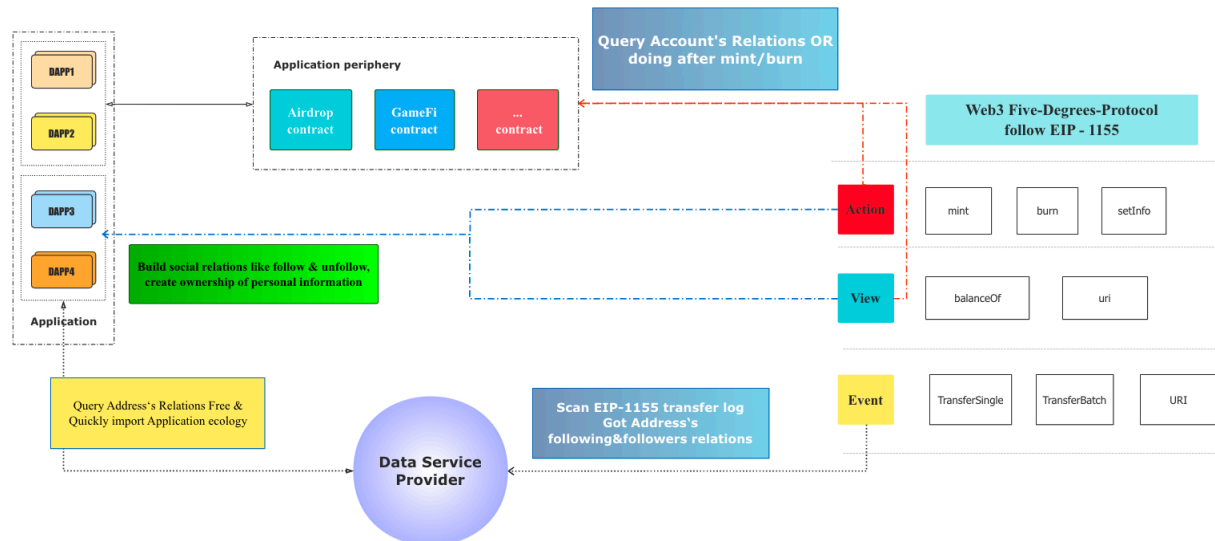


## Solutions

Based on 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 decided to build the protocol under the existing ERC-1155<sup>[3]</sup> standard. Being compatible with the ERC-1155<sup>[3]</sup> standard means that it will be easily implanted into any protocol or business of Web3<sup>[1]</sup> that supports ERC-1155<sup>[3]</sup>. The NFT generated in the protocol can be traded in any NFT market, and any smart contract that requires user relationships can track this data on chain. Any application that supports ERC-1155<sup>[3]</sup> can easily integrate this protocol.

With the development of the Web3.0<sup>[1]</sup> industry, entities can 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 create various products. For example, a social module can be added to Uniswap, so

that users can visually get the detailed data of the transactions made by the people they follow. A moment feature can be added to OpenSea, so that users can easily know what kinds of NFTs their friends are trading or holding, just like integrating Twitter or Discord into OpenSea. And if a social network product uses this solution to start its business, the user's friend relationships can be reconnected in Sandbox even when the product has broken down.



## 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 followed entity, holding his NFT means your relationship is connected

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

**Get entity's follower list** : Query by checking who is holding the entity's NFT

**Get entity's following list** : Query by checking 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>]