DOID Protocol: A Decentralized Self-Sovereign OpenID Protocol

Introduction to DOID Protocol

The DOID protocol is an open collaboration protocol for decentralized identities that users can control independently. The goal of the DOID protocol is to provide a more general and open identity solution for the decentralized world, and to build an ecosystem based on the open identity.

The issues DOID addresses

In a multi-chain system, existing identity solutions usually only support one chain and cannot interact

Although there are already many identity solutions, such as Ethereum's ENS and Aptos'upcoming ANS, these solutions usually only work on one chain and cannot interact with each other. For example, how to link or manage one identity on ENS with another identity on ANS. Not only that, how can blockchains without an identity solution work together?

Existing identity solutions do not help identify user uniqueness

Although existing identity solutions provide identity registration and identification, users can register many identities, and there is no direct correlation between these identities, which will become a problem for projects hoping

to identify the uniqueness of users and also cause a waste of resources. The project can only determine the uniqueness of users through other means, such as asking users to submit social network information, etc. These methods are complicated and facing high risk of disclosure of user privacy.

Benefits and conveniences that identity can bring to users in a decentralized world

An identity is more than just a wallet address. Behind the identity is a huge amount of user behaviors and data, the value of which is currently not well used in a decentralized world. another issue that needs to be solved is how to maximize the benefits of these behaviors and data while protecting privacy, so that users can match or even surpass the benefits and convenience of identity in the Web 2.0 era.

What could DOID achieve

Meeting the four essential CHARACTERISTICS of decentralized identity

The W3C'S CURRENT DECENTRALIZED IDENTITY STANDARD ADDRESSES four essential CHARACTERISTICS[1] that we believe all decentralized identity solutions need to meet.

1. **Decentralized**: there should be no central issuing agency;
2. **Persistent**: the identifier should be inherently persistent, not requiring the continued operation of an underling organization;
3. **Cryptographically verifiable**: it should be possible to prove control of the identifier cryptographically;
4. **Resolvable**: it should be possible to discover metadata about the identifier.

[1] https://www.w3.org/TR/did-use-cases/

Meet the ten principles of self-sovereign identity

When discussing decentralized identity, self-sovereign identity will inevitably be discussed, and DOID also needs to meet the ten principles of self-sovereign identity[2]:

1. **Existence.** Users must have an independent existence.
2. **Control.** Users must control their identities.
3. **Access.** Users must have access to their own data.
4. **Transparency**. Systems and algorithms must be transparent.
5. **Persistence.** Identities must be long-lived.
6. **Portability.** Information and services about identity must be transportable.
7. **Interoperability.** Identities should be as widely usable as possible.
8. **Consent.** Users must agree to the use of their identity.
9. **Minimalization.** Disclosure of claims must be minimized.
10. **Protection.** The rights of users must be protected.

Openness

DOID should support existing identity solutions such as ENS, ANS, BIT, etc., and support multiple blockchains, including registration from different blockchains and resolution on different blockchains.

Users should be able to manage all DIDs in one place, including unified management of address information on different blockchains and identities in different identity systems.

The privacy of users should be protected. The privacy of users, such as the relationships between different addresses and the relationships of behavioral data, should not be publicly obtained unless disclosed by users.

[2]https://github.com/ChristopherA/self-sovereign-identity/blob/master/ThePathToSelf-SovereignIdentity.md

Proof of uniqueness

When a user chooses to disclose, it should be possible to prove the uniqueness of the user's identity, that is, that a user is not using multiple accounts. This will greatly reduce the problems caused by account fraud such as zombie accounts and account masquerades.

User credit assessment

In the future, with the development of decentralized finance, a set of decentralized credit evaluation system will be constructed, and DOID-based identity identification and certification will greatly help the establishment of this credit evaluation system.

DOID SOLUTION

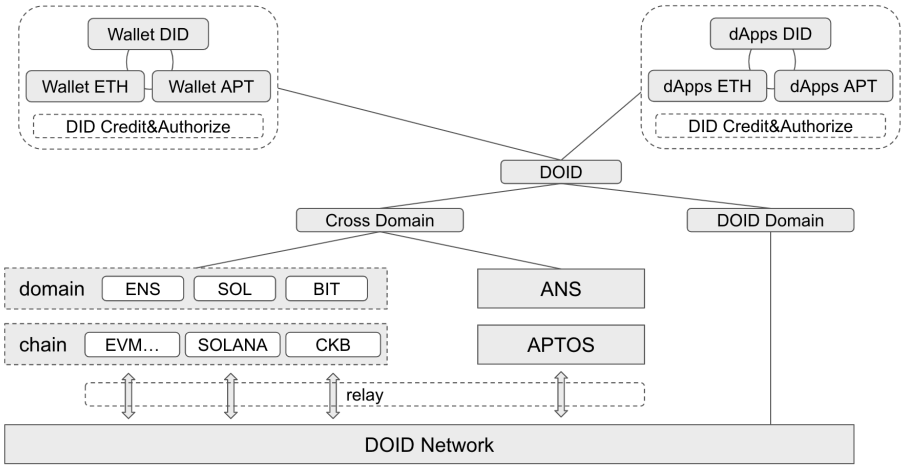
Decentralization and self-sovereign

DOID is built based on a decentralized network, which stores all DOID identity data. In this decentralized network, each node has full data, and these nodes maintain the data through a consensus mechanism to ensure that no node can independently control the network.

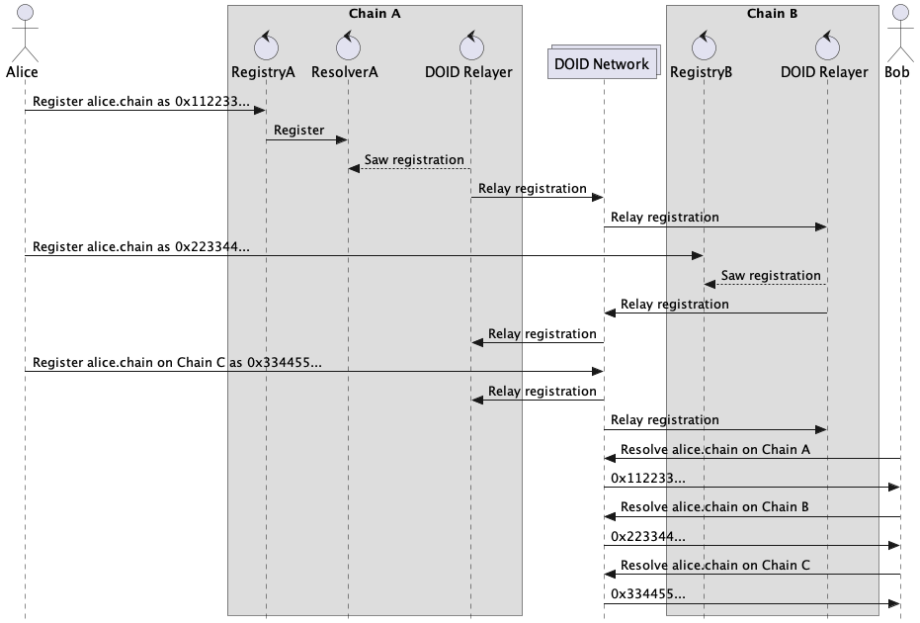
Users can achieve complete control over their identity data based on such decentralized network, combined with cryptography technology.

Multi-chain and multi-identity communication

The DOID Bridge Protocol describes how the decentralized network of DOID can communicate with other networks. Thanks to this, DOID network can communicate with other chains and other identity systems. At the same time, by providing a resolving interface on each chain, the identity of any identity system, including DOID's own identity, can be resolved on any chain. The overall structure is shown as below:

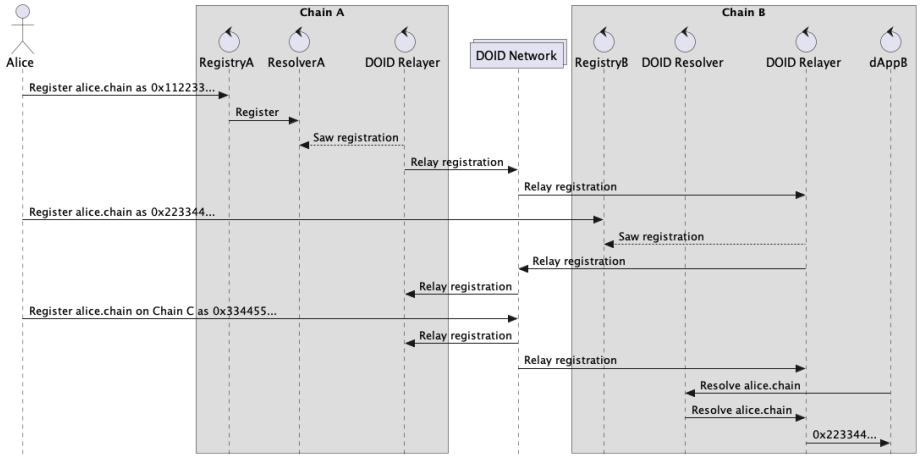


Users and wallets can use DOID to resolve identity information when making a transfer or other operations cross chains and cross identity systems. The communication process as follows:



When using DOID to complete cross-chain and cross-identity systems, dApp will resolve the results of the current chain by default, and can also selec-

tively resolve the addresses of other chains. For example, in some scenarios where the target addresses of other chains are expected to be verified, the communication process is as follows:



User identification with privacy protection

Thanks to zero-knowledge proof technology, users can prove their identity on each chain, and even prove their asset information and behavior information without disclosing the original information. Combined with social network information, users can also prove the uniqueness of their identity and credit evaluation information without disclosing their privacy.

The privacy preserving algorithm based on zero-knowledge proof is relatively simple and needs to be completed by a random challenge. This method can better protect the prover and the verifier, as follows

1. The prover generates the key, randomly chooses a number $a \in_R \mathbb{Z}_p$, calculates $V = g^a$ and sends V to the verifier. In this case, the prover' s key is a, the verifier's key is V, in this case,$a \in \mathbb{Z}_p, V \in G$. Of course, there are many decentralized and more private key exchange schemes available in this step, which will not be described here.
2. Introducing a random challenge, the verifier selects a random number $c \in_R \mathbb{Z}_p$ and sends it to the prover.
3. The prover computes the Hash value of the information as

$h = Hash(m)$, the prover computes $H = g^h, z = h + a \times c$ to generate the proof (z, H, V).

4. The verifier verifies the proof, and the verifier computes and checks whether $g^z = H \times V^c$ holds to verify the proof.

Of course, in order to reduce interaction, there are many alternatives to random challenges, such as taking $c = Hash(V)$, so that the verifier only needs to complete the verification.

Proof of uniqueness

Proving the uniqueness of a user's identity is more complicated, and requires the user to provide information such as social graph and complete probability evaluation. This probability evaluation mainly relies on the random walk in the whole social network. Some studies have proved that the probability of random walk ending early on the unique user node is higher than that on the non-unique user node. Therefore, this probability can be used as a confidence degree of user uniqueness.

Here, a power iteration algorithm can be used to calculate the trust value of user uniqueness, which is an efficient technique for calculating the landing probability of a random walk in a large graph. This technique discovers low uniqueness users in three stages. The first stage iterates through $w = O(\log n)$. Trust starts from a known unique user node and spreads to the entire network. This traversal is biased towards the unique user region. In the second phase, nodes are ranked according to their trust level. In the final phase, user nodes that are less unique are marked in the ranking list.

The following is a detailed description of the algorithm.

1. Let $T^i(v)$ denote the trust value of node v after i steps iteration. The trust value T_G of the whole network G will be evenly distributed among K trusted nodes $v_1, v_2, ..., v_k$, namely

$$T^0(v) = \begin{cases} \frac{T_G}{K}, & v \in K \\ 0, & \text{else} \end{cases}$$

2. Let's call the edge of the whole network G as E. At each power iteration, a node first distributes its trust value evenly among its neighbors. It then collects the confidence assigned by its neighbors and updates its own trust value accordingly. The process is shown below. Note that the total amount of trust value T_G of the entire network will remain the same.

$$T^i(v) = \sum_{(u,v)\in E} \frac{T^{(i-1)(u)}}{deg(u)}$$

3. With enough power iterations, the trust vector will converge to a stationary distribution: $\lim_{i\to\infty} T^i(v) = \frac{deg(v)}{2m} \times T_G$. However, only $w = O(\log n)$ power iterations are performed here, so they end before convergence. This degree is sufficient to achieve a sufficient uniform distribution of confidence, while at the same time limiting confidence propagation to nodes with low uniqueness.
4. After $w = O(\log n)$ power iteration, denote the trust value of v as $T_v = \frac{T^w(v)}{deg(v)}$, and then we can sort the nodes with confidence.

At this point, an estimate of user uniqueness can be derived. In the future, zero-knowledge proof can be combined to prove the uniqueness of users without disclosing their social networks. Similarly, the user's behavior data can be combined to further calculate the user's credit assessment without disclosing the privacy.