On-Chain DIDs

by Martin Riedel martin@identity.org

Background

Decentralized Identifiers (DIDs) are an identifier and document standard that enables an abstraction between an external identifier and the state (DID Document) that is resolved behind it. DIDs are conceptually a standard to break siloed solutions around identifier control, since various DID methods can freely be defined and work in an interoperable way. Therefore, from a web3 (or web5 perspective as Block's TBD might put it), DIDs are an **off-chain** concept and have no real value or meaning in any on-chain applications.

Interpreting the DID spec "on-chain"

Recently, however, there is a movement in the web3 community that the introduction of an abstraction layer between an identifier owning your digital assets and the represented control behind it (e.g. what keys, or combination of keys are able to control the identifier and or specific assets) is a valuable concept. Several targeted solutions exist in the space with Gnosis Safe being one of the most well known platforms.

When Identity.com first published the did:sol: method spec and implementation, we soon discovered that the on-chain representation of the DID state would be valuable for not only off-chain DID Document resolution but also for on-chain evaluation that could offer the controller of the DID the same identifier abstraction on-chain that they are familiar with in their off-chain use-cases.

In order to demonstrate the concept Identity.com introduced Cryptid a wallet abstraction on Solana that evaluates ones `did:sol` state on-chain in order to execute generic transaction on the Solana Blockchain. However, with the implementation came questions around nomenclature and spec compliance. For example, is it a correct statement to say "one transact with their DID on Solana"? For now, we call the on-chain representation of the `did:sol` a `Cryptid Account` in order to differentiate the on-chain from the off-chain applications. However, we would love to bring these applications and nomenclatures closer together, since after all, they are operating on the same state.

Questions & Concepts to discuss further at RWOT 11

- How does a DID-spec conforment evaluation of the DID state look like on-chain?
- What nomenclature should be adopted for a DID Identifier/State that is evaluated on-chain?
- How to model extended DID state from on-chain use-cases within the off-chain DID Document? (Example: `did:sol` allows to proof "ownership"

of the private key of a certain verification method)

- Are DID Document a good vehicle to link wallets (on-chain and cross-chain)?
- Should DID methods allow a “reverse” lookup to identify all DIDs that contain a specific key/verification method?

Goals

As an outcome for the workshop I’d love to get to some kind of understanding how DID state that is managed within smart-contract blockchain platforms can be used and evaluated on-chain.