2020-11-03 Indy DID Method Specification Call

Summary

- Recap of IIW Discussion
- Collaboration Tools
- About the <network> element of the DID

Hyperledger is committed to creating a safe and welcoming community for all. For more information please visit the Hyperledger Code of Conduct.

Antitrust Policy Notice

Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

HYPERLEDGER
BLOCKCHAIN TECHNOLOGIES FOR BUSINESS

Recording from the call: 20201103 Indy DID Method Call Recording

Welcome and Introductions

Announcements

- Status of relevant, unresolved issues in the DID Core Spec
  - A `type` attribute

Discussion

- Recap of discussion at IIW
- Collaboration Tools:
  - Current hackmd document
  - indy-did-method on RocketChat - https://chat.hyperledger.org/channel/indy-did-method
  - indy-did-method repo
    - ReSpec vs. SpecUp
- `About the <network> element of the DID – did:indy:<network>:<id>`
  - What is the goal of the structure of the network?
    - Hash (543F4) – unrecognizable, verifiable with the ledger, short, non-discoverable/requires a registry
    - Domain Name (example.com) – recognizable, discoverable, not tied to the ledger, dependent on DNS

- Arbitrary Name (SovrinStaging) - recognizable, non-discoverable/requires a registry, not tied to the ledger
- Combination of hash and domain name (this is what TrustBloc does)
- Combination of arbitrary name and hash <arbname>:<hash> e.g. did:indy:sovrin:<hash>:<id>

Approach	Discoverability	Decentralized Control	(Limited) Verifiability	Human Friendly	Conciseness	Dependencies
Hash of Domain Genesis File	No	Yes	Yes	No	Yes	Registry or Config
Domain Name	Yes	Yes	No	Yes	No	DNS
Arbitrary Name	No	Yes	No	Yes	No	Registry or Config
Hash and Domain Name Alias, as in TrustBloc	Yes and No	Yes	Yes	Yes and No	Yes and No	DNS and Config
Arbitrary Name + Hash	No	Yes	Yes	Yes	No	Registry or Config

        - What is the easiest way for agents to use this?
            - DNS is a hard sell per Dan Gisolfi
            - A registry implies centralization - e.g. GitHub, DIF, ToIP
            - Today it will be just a manual list of name - config files
            - Does readability matter? Who sees a DID?
                - Should be no one. "If anyone sees a DID, we've failed at our job" - quote from RWoT

Discussion ended here. The following was not discussed in detail

- First 5 characters of a hash – of what?
    - Genesis File
        - What Genesis File? Domain (does not change - first n transactions on the ledger), Pool (does change - inevitable as it contains IP:port of nodes)
            - Proposal: Use the Domain Genesis File hash
            - Pool file is required to contact nodes of the network.
            - If Domain, what to do if there is a fork?
                - Proposal: Domain Genesis file contains the first n records after the fork, as the sequence number is the same
- Should an "alias" be allowed as TrustBloc uses?
    - From Troy Ronda: A quick update on our did:trustbloc handling of multiple networks. With the ability to specify a canonical DID in the DID document, we are adding the ability to have both discoverable domains in the DID - e.g., did:trustbloc:domain:suffix and also to have a stable consortium genesis identifier - e.g., did:trustbloc:<consortium genesis hash>:suffix. The canonical DID would become the <consortium genesis hash> version such that the resolution of discoverable domain DIDs would point to this canonical DID in the resolution result.
    - TrustBloc alias example:
        - did:trustbloc:testnet.trustbloc.dev:s3did12345
        - https://testnet.trustbloc.dev/.well-known/did-trustbloc/org1.sandbox.trustbloc.dev.json
            - Configuration file – equivalent to the Indy Pool Genesis File
    - So Indy might use:
        - <domain> alias is :example.com, https://example.com/.well_known/did-indy/ ??
            - Perhaps a folder with current ledger pool genesis file (to find the ledger) and ledger domain genesis file (to find/check the hash)
- If the DID to be resolved is NOT using an alias, how is the Pool Genesis File found?
    - Known by all that need to know it?
    - Registry? GitHub?

Attendees: Stephen Curran Alexander Jonsson Kumaravel N Paul Bastian