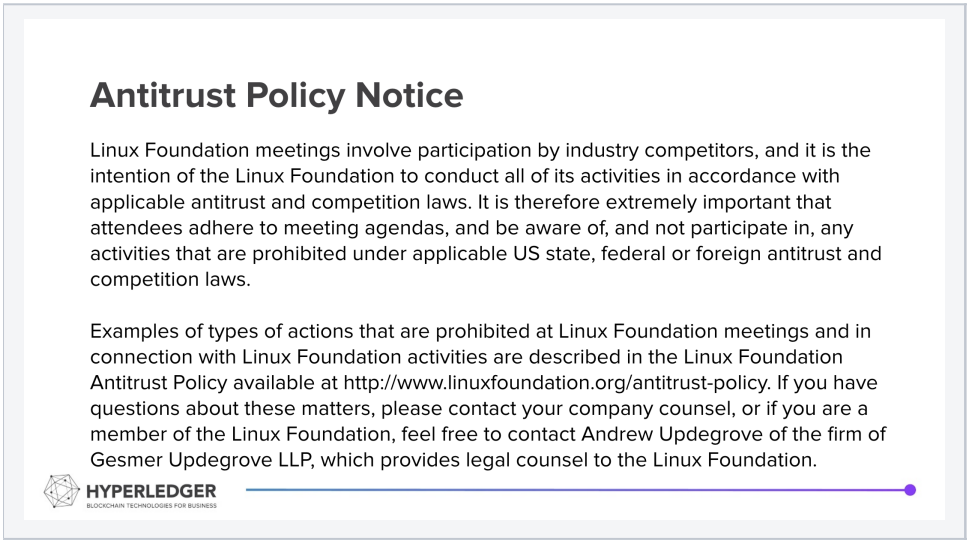
2021-04-20: CANCELLED - IIW - Indy DID Method Specification Call

Summary

- Cancelled because of IIW

Recording from the call: None



Hyperledger is committed to creating a sa

community for all. For more information

please visit the Hyperledger Code of Con

Welcome and Introductions

Announcements

Attendees

- Stephen Curran

Collaboration Channels

- Current hackmd document
- indy-did-method on RocketChat - https://chat.hyperledger.org/channel/indy-did-method
- indy-did-method repo: ~~ReSpec vs.~~ SpecUp

Agreed Upon:

- See HackMD Document for most of what we have discussed

Online Discussion (from RocketChat this week)

- Serialization formats

This Week's Discussion:

- DIDDoc Construction:
  - Generate template and merge in diddocContent if present, or
  - ~~If diddocContent is present, it is the DIDDoc -- check to be sure that the NYM key is in the DIDDoc~~
  - Good idea? At minimum, we need to be able to add elements of `diddocContent` to all of the NYM-based DIDDoc
- Other Indy Ledger Objects as DIDs (e.g. schema, etc):
  - Code and documentation:
    - https://github.com/hyperledger/indy-node/blob/master/indy_node/server/request_handlers/read_req_handlers/get_claim_def_handler.py
    - https://github.com/hyperledger/indy-node/blob/master/docs/source/requests.md

- Only require namespaced ledger references for DIDs, CLAIM_DEFs, and SCHEMA. Implications:
  - All objects must be on the same ledger as their creating controller
  - CLAIM_DEFs may reference SCHEMA on the same or another ledger
  - REV_REG_* objects must be on the same ledger as the related CLAIM_DEF
- Add resource resolution via Sequence Number: did:indy:sovrin:56495?resource=true, which returns the ledger object at that location
  - A resolver would call the existing GET_TXN Indy call.
- Change identifiers outside of the ledger to the following:
  - Schema: `did:indy:sovrin:F72i3Y3Q4i466efjYJYCHM/SCHEMA/npdb/4.3.4`
  - Claim Definition: `did:indy:sovrin:5nDyJVP1NrcPAttP3xwMB9/CLAIM_DEF/npdb`
  - Revocation Registry Definition: `did:indy:sovrin:5nDyJVP1NrcPAttP3xwMB9/REV_REG_DEF/npdb/TAG1`
  - Revocation Registry Entry: `did:indy:sovrin:5nDyJVP1NrcPAttP3xwMB9/REV_REG_ENTRY/npdb/TAG1?version_id=12345`
    - Note that the Rev Reg Entries all have the same name, but are actually deltas – meaning that all are necessary in doing a proof of non-revocation. The data needed across the transactions are picked up using the transaction: GET_REVOC_REG_DELTA
- In the examples above, the items after the object type are all controller (client) defined.
  - For all, on right, the following rules would be enforced:
    - For Schema, Claim Def and RevReg Def, duplicates are NOT permitted on the ledger (pre-write check).
    - For RevReg Entry, duplicates are allowed as they are deltas – e.g. all are "live".
- Questions:
  - Do we need to have the SCHEMA ID within the CLAIM_DEF ID?
    - Sort of not needed – it is included in what is returned.
      - The current GET_CLAIM_DEF call includes the SCHEMA ID as a parameters.
    - However, the Schema is returned as "Ref" and is the sequence number for the schema on the current ledger
    - That would need to be adjusted
  - Could we use a path for the object, e.g: **`did:indy:sovrin:`**`5nDyJVP1NrcPAttP3xwMB9/3/CL/56495/npdb`
    - `New Transactions:` **`did:indy:sovrin:`**`5nDyJVP1NrcPAttP3xwMB9/CLAIM_DEF/npdb`
    - Legacy Transactions: did:indy:sovrin:CLAIM_DEF?id="5nDy..:3:CL:56495:npdb"
    - **`did:indy:sovrin:`**`F72i3Y3Q4i466efjYJYCHM/SCHEMA/npdb/4.3.4`
    - ~~**`did:indy:sovrin:`**`5nDyJVP1NrcPAttP3xwMB9?CLAIM_DEF=npdb`~~
    - ~~**`did:indy:sovrin:`**`F72i3Y3Q4i466efjYJYCHM?SCHEMA=npdb:4.3.4`~~
  - What does a minimal DIDDoc look like that we could use if we reference on object without "resource=true" or with "resource=false"?
    - Could return just the identifier of the object.

- JSON vs. JSON-LD
  - Here's a google doc that captures my thinking. I am not so emotionally or intellectually caught up in my own perspective here that I will balk if I am out-voted, but I would appreciate knowing that a thoughtful discussion about it occurred before a decision was made.
    - Link to Google Doc
- The "close-to-finished" DID Method Spec – please review
  - Perhaps not close to finished – doesn't talk about other objects yet – the conversation above
  - At risk – DNR/DND, KERI

Future Discussions:

- DNR and DND discussions
  - To find networks we will require at least the first and perhaps the second of these approaches, while the rest are suggested:
    - Config files for one or more known networks
    - A mechanism for a ledger operator to register discovery information for other ledgers (aka "human gossip")
      - A DID/DIDDoc on a ledger will contain cross-registry information
      - A mechanism is needed for finding the DID(s) that contain the registrations – ideas have been put forward - a DID Name Directory (DND) is the likely approach.
      - Document about the DND and DNR records
    - Decentralized registries based on verifiable credentials
    - Other registry mechanisms, such as the DDNR proposal
    - The DID Method Spec will include a reference to a repo (likely) "indy-did-networks" within Hyperledger that will be a lightly managed, structured repository of folders per Indy network with at least the config file(s) for the networks. Use of the repo is voluntary, but provides a convenient way for networks to publish information about the network. Maintainers will be selected from the community and should exhibit a light hand in accepting PRs, being concerned mainly with structure of the data (not content) and that contributors are not being malicious about updating the information of other network operators. The Hyperledger governance structure may be used for disputes as appropriate. This is not a replacement for the Governance that a specific network should implement.