Cordentity Labs Project
Indy Identity on R3’s Corda
Emerging standards for decentralized identity

DECENTRALIZED IDENTITIES

Anchored by

BLOCKCHAIN IDs

Linked to

ZERO-TRUST DATASTORES

That are

UNIVERSALLY DISCOVERABLE

for people, organizations, apps and devices.
Decentralized Identifiers (DIDs) and DID Documents

Key enablers for decentralized self-sovereign identity

- Decentralized
- Privacy
- Self-Sovereignty
- Security
- Proof-based
- Portability
- Discoverability
- Simplicity
- Interoperability
- Extensibility

**DESIGN GOALS**

Also Service end-points, Proofs, Extensions, etc

See [https://w3c-ccg.github.io/did-spec/](https://w3c-ccg.github.io/did-spec/) for details
By decoupling the trust between the identity provider and the relying party, a more flexible and dynamic trust model is created such that market competition and customer choice is increased.
Verifiable credentials

How DIDs become (use-case specific) Identities

DMV – issuer

Traveler – Holder/Subject
DID: EXAMPLE:EBFEB...

Credential

Bar - inspector/verifier

W3C Example

{  "id": "http://example.gov/credentials/3732",  "type": ["VerifiableCredential", "ProofOfAgeCredential"],  "issuer": "https://dmv.example.gov",  "issued": "2010-01-01",  "credential":{"id": did:example:ebfeb1f712ebc6f1c276e12ec21,  "ageOver": 21},  "signature": {  "type": "RsaSignature2018",  "sigalg": "2048-SHA256",  "signatureValue": "Bavell0/I1zpYwBXNiibyVg/sCneO4JukezBBwDg/+ MCRVpJoDoDeO4SXXXjkcOVK/CHGDvC4kqi6Z1nBufqxxGfmatCuFlbcC1wps PRdWgGsUtP7LzvueMmFhWymfIfpb8u95t50l+rSLHIUujm/+PXr9Cky6Ed +W3JT24="}}
Dedicated, public but permissioned ledger
Pair-wise DIDs
Agents based claims/proofs exchanges
Implements Attribute Based Claims (ABCs)
ZKPs for selective disclosure & revocation

A global public utility for self-sovereign identity should be used even more than DNS. Although the DNS is capable of much more, the vast majority of DNS requests are simply to look up the IP address for a domain name. With over 1 billion host computers now available on the Internet, the DNS is serving over 100 billion lookups per day. The same should be true of a global public utility for DIDs—only instead of looking up IP addresses from domain names, it will be looking up public keys from DIDs. If you imagine every person, organization, or thing needs a collection of DIDs—one for every relationship they have—then it is easy to imagine that there could be trillions of DIDs in a global decentralized identity system. Sovrin is explicitly designed to achieve this scale.

DNS is simpler to scale (yet more susceptible to attack) because it does not use a consensus protocol to create an immutable blockchain. Yet all consensus protocols can only scale to a limited number of validator nodes. To overcome this hurdle, the Sovrin Network is designed to use two rings of nodes: a ring of validator nodes to accept write transactions, and a much larger ring of observer nodes running read-only copies of the blockchain to process read requests. In addition, the Sovrin blockchain is engineered to be able to return a state proof with any response. This is a very lightweight cryptographic proof—capable of being processed on a smartphone—that the response is valid according to the current state of the ledger, which should prevent man-in-the-middle attacks on Sovrin queries.

The network must have the performance and scalability of DNS.
Corda + Indy = Luxoft’s Cordentity

- **Cordentity** is an *utility CorDapp* which exposes high level APIs hiding complexity of Hyperledger Indy.
- **Cordentity** doesn’t require deep knowledge of cryptography or Indy’s functionality. It operates with basic primitives: schema, definition and proofs.
- **Cordentity** utilizes Corda’s *flows*, *states* and *contracts*.

**CORDAPPS**
Corda applications running on private client’s node

**CORDAPP**
Application specific flows to implement required business process

**INDY- CREDENTIAL**
CredentialRequest & Credential

**INDY-CREDENTIAL-PROOF**
ProofReq & Proof

**HOLDER/ISSUER**

**Prover/Verifier**

**PRIVATE WALLET**
- Keys
- Credentials
- Validity Proofs

**INDY SDK**

**CORDIDENTITY**

**FLOW**
Indy specific flows to work with the Credentials and Proofs

**FLOW**

**CORDAPP #Y**

**FLOW**

**CORDAPP #X**

**FLOW**
Cordentity – usage overview

- Every individual Corda Node that uses Cordentity has a **private wallet**
- Cordentity’s flows interact with a specified Indy network
- Indy network should be specified via file with genesis transactions. There are 3 type of networks: docker powered (development), STN and production.
- Authority or Issuer have to get permissions. AssignPermissionsFlow provides suitable interfaces.

### Authority & Issuer

**CREATE SCHEMA FLOW**
Legal entities create new schema as a definition of future credentials

```kotlin
class Authority(
    private val schemaName: String,
    private val schemaVersion: String,
    private val schemaAttributes: List<String>
) : FlowLogic<String>()
```

**CREATE CREDENTIAL DEF FLOW**
Authorities authorized to issue user’s credentials create credentials definition on top of the existing schemas

```kotlin
class Authority(private val schemaId: String)
```

### Prover & Verifier

**ISSUE CREDENTIAL FLOW**
User requests new credential from one of the authorities

```kotlin
class Issuer(private val identifier: String,
    private val credDefId: String,
    private val credProposal: String,
    private val proverName: CordaX500Name) : FlowLogic<Unit>()
```

**VERIFY CREDENTIAL FLOW**
Two users check/verify credentials

```kotlin
class Verifier(
    private val identifier: String,
    private val attributes: List<ProofAttribute>,
    private val predicates: List<ProofPredicate>,
    private val proverName: CordaX500Name
) : FlowLogic<Boolean>()
```

Two type of data in a proof: **Predicates** and **Attributes**.
- A **predicate** is never revealed and just checked on a criteria.
- An **attribute** will be revealed.
Trust / Attestations Mapping

BASIC RELATIONSHIPS MODEL

AUTHORITY

ISSUER #1

ISSUER #2

ISSUER #3

PROVER

VERIFIER

USE-CASE SPECIFIC RELATIONSHIPS MODEL

AUTHORITY

INSURANCE

GOVERNMENT

ISSUER

INSURANCE CREDENTIAL DEFINITION #1

GOVERNMENT CREDENTIAL DEFINITION #2

CREDENTIALS

CREDENTIAL DEFINITIONS

PROVER

PATIENT

TREATMENT CENTER

CREATE SCHEMA
CreateSchemaFlow

ASSIGN PERMISSIONS
AssignPermissionsFlow

CREATE CREDENTIAL DEFINITION
CreateCredentialDefinitionFlow

ISSUE CREDENTIAL
IssueCredentialFlow

DATA VERIFICATION
VerifyCredentialFlow
Complete Sample Application

Personalized Medicine End-to-End Ecosystem

- Personal data is kept privately and not shared with external participants
- Participants’ data is verifiable and immutable against fraud
- Selective data visibility
- Isolated pairwise relationships between participants

1. Patient is prescribed with a personalized medicine therapy
   DID: SOV: 12345689ABCDEFGAB

2. Insurance company confirms the coverage of therapy costs
   DID: SOV: 937473838JFGDFEDH

3. Treatment Center places pers. medication order to the assigned Manufacturer
   DID: SOV: 135473839JFGDFEDH

4. Manufacturer produces and ships the pers. medication package

5. Courier delivers the package to the Treatment center

6. Patient receives the therapy at the Treatment center
   DID: SOV: 123456789ABCDEFGAB

Sample Application

Personalized Medicine
Conclusion

- Self-Sovereign Identity & Verifiable Credentials is a very powerful mechanism
- Scalable, DLT-enabled Business Ecosystems benefit from SSI integration
- Corda is the next generation DLT that simplifies integration with other technologies
- Cordentity makes it easy to use Hyperledger Indy / Sovrin powered SSIs / Credentials from CorDapps

Is Open Source – please use it and let us know how to make it better!
Thank you

r3.com