Indy

2017-03-17

# HIP identifier

Indy

# Sponsors

*Sovrin Foundation:*

*Steve Fulling*

*Executive Director*

Sovrin Foundation

Code Maintainer

steve@sovrin.org

*Phil Windley*

*Chairman and President*

Sovrin Foundation

Code Maintainer

pjw@sovrin.org

*Evernym:*

*Jason Law*

*CTO*

Evernym

Code Maintainer

jason.law@evernym.com

*Nathan George*

*Software Architect*

Evernym

Code Maintainer

nathan.george@evernym.com

*Please reach out to Phil Windley and Nathan George with your contact information if you wish to sponsor this project.*

# Abstract

*Indy* is a distributed ledger purpose-built for decentralized identity. It provides tools, libraries and reusable components for providing independent digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any other “silo”.

# Context

The Sovrin Foundation (<http://www.sovrin.org>) was formed as a neutral open source public trust organization to manage a global public utility for decentralized identity for any individual, group, organization, or thing. The initial code for the Sovrin ledger was primarily developed by Evernym, Inc. (<http://www.evernym.com>), who open sourced it under the Apache 2.0 license and donated it to the Sovrin Foundation in order to provide the core engine for a global public utility for decentralized identity (also called *self-sovereign identity* or *independent identity*).

# Motivation

Identity is one of the most widely recognized applications of distributed ledger technology—there are at least a dozen active “blockchain identity” projects in development around the world. However, identity is in fact a specialized use case for distributed ledger technology just like cryptocurrency, deed registries, healthcare data exchanges, or other popular uses. Moreover, because distributed ledgers cannot be altered after the fact, it is essential that use cases for ledger-based identity carefully consider foundational components, including performance, scale, trust model, and privacy. In particular, [Privacy by Design](https://en.wikipedia.org/wiki/Privacy_by_design) and privacy preserving technologies are critically important for a public identity ledger where correlation can take place on a global scale.

For all these reasons, the Indy Project has developed specifications, terminology, and design patterns for decentralized identity along with an implementation of these concepts that can be leveraged and consumed both inside and outside the Hyperledger Project.

# Status

The work on the Indy code base (originally called “Plenum”) started in September 2015, with public participation starting in September 2016. The first go-live stage of the Sovrin Network, a specific instance of Indy designed to operate as a global public utility for identity governed by the Sovrin Foundation, is scheduled for April 2017, with general availability starting later in 2017.

Current work items for the “Indy” code contributors include:

1. Refactor existing interfaces to make the code-base more accessible to new contributors.
2. Improve the performance and efficiency of the consensus protocol (Plenum). This includes adding support for new transport protocols such as: a) CurveZMQ, b) an EdDSA version of ZeroMQ over TLS, or c) QUIC. (Currently Plenum only supports RAET, a python-specific secure messaging protocol.)
3. Finish adding support for observer nodes (auxiliary nodes that act as edge-cache hot-standby nodes for large-scale deployments such as the Sovrin Network). Also automate the process of promoting nodes from validators to observers.
4. Improve the APIs and interfaces for pairwise decentralized identifiers ([DIDs](https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2016/blob/master/draft-documents/did-implementer-draft-10.md)), verifiable claims, anonymous credentials and cryptographic key management.
5. Abstract the Indy Identity layer to a transaction family, so that Indy can function as an identity service within another distributed ledger platform (depending on whether the platform can support the special privacy, scalability, and trust model requirements).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hyperledger INDY Roadmap** | | | | |
| Phase 1 | Phase 2 | Phase 3 | Phase 4 | Phase 5 |
| **INDY Core**  **Dev**  10/16-3/17 | **Minimum Go-Live**  4/21 | **First General**  **Availability Milestone**  TBD | **Second General**  **Availability Milestone**  TBD | **Third General**  **Availability Milestone**  TBD |

**Phase 1:** This was the initial development phase after the public announcement of the Sovrin Foundation at the Ctrl-Shift Personal Information Economy Conference on Sept. 29, 2016.

**Phase 2:** Complete the stories needed to reach the minimum requirements for the first stage of the Sovrin public identity ledger, called the Provisional Network, to go live by the end of April 2017.

**Phase 3:** First General Availability Milestone (TBD - 6/17): Windows server development, stability, agents, upgrades and enhancements for the live Sovrin DLT, etc.

**Phase 4:** Second General Availability Milestone (TBD): Documentation, scaling, enhanced spam protection, development tools, consent receipts and adding new Stewards, etc.

**Phase 5:** Third General Availability Milestones (TBD): P2P DID connections, consumer platforms, wallets, etc.

# Solution

## The Concept of Using a DLT for Digital Identity

Digital identity has evolved through three key stages:

1. **Centralized**—where the identity is controlled by a single provider.
2. **Federated**—where identities can be shared across a set of cooperating providers.
3. **User-centric**—where the users of federated identities are given greater control over their usage (the broadest example being [social login](https://en.wikipedia.org/wiki/Social_login)).

However in all of these stages, the underlying digital identity is fundamentally controlled by an “identity provider”, meaning this provider is ultimately in control—i.e., can take the identity away—and that the identity is still part of a silo (even if that silo is very large).

The emergence of distributed ledger technology—and the indisputable strength of its distributed security model—finally enables the fourth stage of digital identity: *decentralized identity* (also called *self-sovereign identity* or *independent identity*). In this model, individuals, organizations (and things, when needed) can have their own set of independent identities that are controlled by their own set of public/private key pairs. The “root” of this identity system is no longer a silo, but rather a distributed ledger that can serve any size community of participants—in the case of Sovrin, a global public utility that can serve the entire Internet.

## Interoperability of Decentralized Identity Across DLTs

Since the promise of ledger-based identity is to break out of silo-based identity, it would be self-defeating to have DLT-based decentralized identities become another set of silos. Sovrin identity architecture has focused on establishing a core set of decentralized identity specs and artifacts that are independent of any particular ledger and will enable interoperability across any DLT that supports them. Specifically these include:

1. **DIDs (decentralized identifiers)**—identifiers that are globally unique and resolvable (via a ledger) without requiring any centralized resolution authority. DIDs are specified in the [DID Data Model and Generic Syntax](https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2016/blob/master/draft-documents/did-implementer-draft-10.md) specification.
2. **DDOs (DID descriptor objects)**—the simple JSON object to which a DID resolves. The DDO exposes the public keys and service endpoints for interacting the entity identified by the DID. DDOs are also specified in the [DID Data Model and Generic Syntax](https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2016/blob/master/draft-documents/did-implementer-draft-10.md) specification.
3. **Verifiable claims**—this is the interoperable format for exchange of digital identity attributes and relationships currently in the standardization pipeline at the W3C. See the [Verifiable Claims Task Force](https://w3c.github.io/vctf/).
4. **Agents**—a standard for communication between the agents that support off-ledger activities.

It is the hope of the Sovrin architects and Indy code contributors that these and other specifications and standards for ledger-based identity can be brought to the Hyperledger community where they can continue to grow.

## Interoperability of Digital Certificates

Digital certificates are a particularly critical point of interoperability for ledger-based identity. Conventional X.509 certificates are the most widely established format for binding cryptographic key material with the identifiers and other metadata necessary to establish chains of trust. This type of certificate is typically deployed by CAs operating in a hierarchical PKI model.

DIDs and DDOs, by contrast, were designed for a “flat” DLT model, where digital identities interoperate as peers, each self-signing its own DDO and serving as its own root of trust. Then, through interacting, connecting, and sharing verifiable claims, the peers begin forming multiple overlapping “webs of trust” as required.

These two models—the X.509 certificate model and the DID/DDO model—are not mutually exclusive by any means. DID and DDO key material and metadata can be used to generate X.509 certificates, and X.509 certificates can be used as Verifiable Claims with the DID/DDO model. In this way we believe hierarchical/federated identity systems can be bridged with decentralized identity systems to the benefit of both.

## The Sovrin Ledger

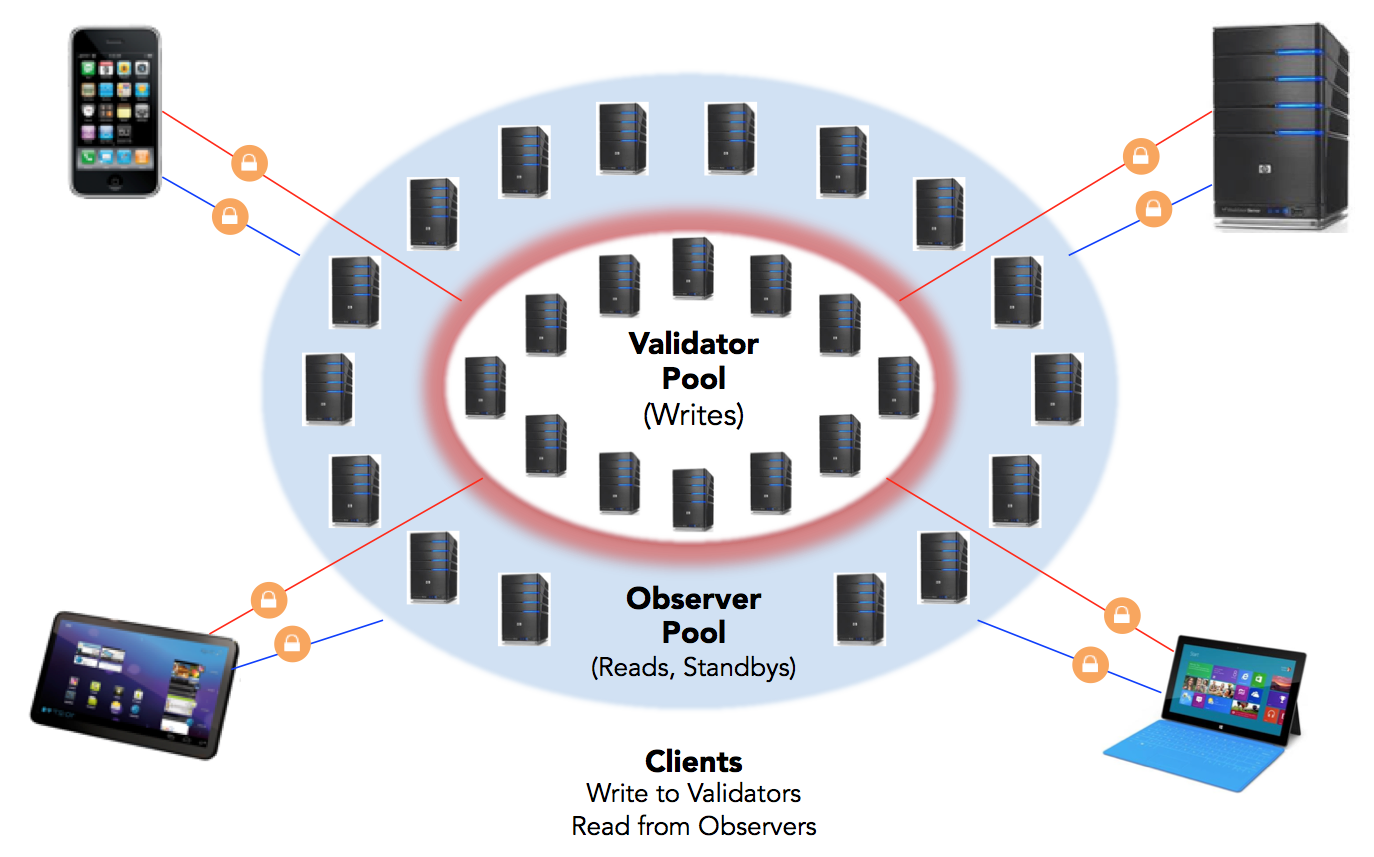
Again, the Indy code base was initially developed by Evernym specifically to power a public permissioned ledger for decentralized identity. Once the governance model was developed and the first 11 trustees were confirmed, the Sovrin Foundation was officially announced on September 29, 2016, and the code base was contributed to the Foundation. More information about the Sovrin Foundation is available at <http://www.sovrin.org/>.

As a public ledger, what most distinguishes Sovrin is that every aspect of the architecture, including the consensus protocol, governance, and trust model, has been built from the ground up exclusively for the purpose of decentralized, self-sovereign, independent identity.

Thus the Indy code base includes numerous features that are very specific to the requirements of a globally-scalable identity ledger. For example, because reads of the Sovrin ledger are expected to outnumber writes by several orders of magnitude—much like DNS—the Sovrin Network includes two node types:

1. **Validator nodes** participate in the Plenum consensus protocol and validate all writes to the network. As a public permissioned ledger, the Sovrin Network is designed to achieve diffuse trust by having validator nodes operated by trusted institutions all over the world—never having too many concentrated in any one country, industry, or hosting facility.
2. **Observer nodes** are read-only copies of the ledger that replicate transactions from the validator pool. These nodes take the much larger load of reads from the network.

This deployment architecture (which again applies specifically to the Sovrin deployment of Indy) is shown in the following diagram.

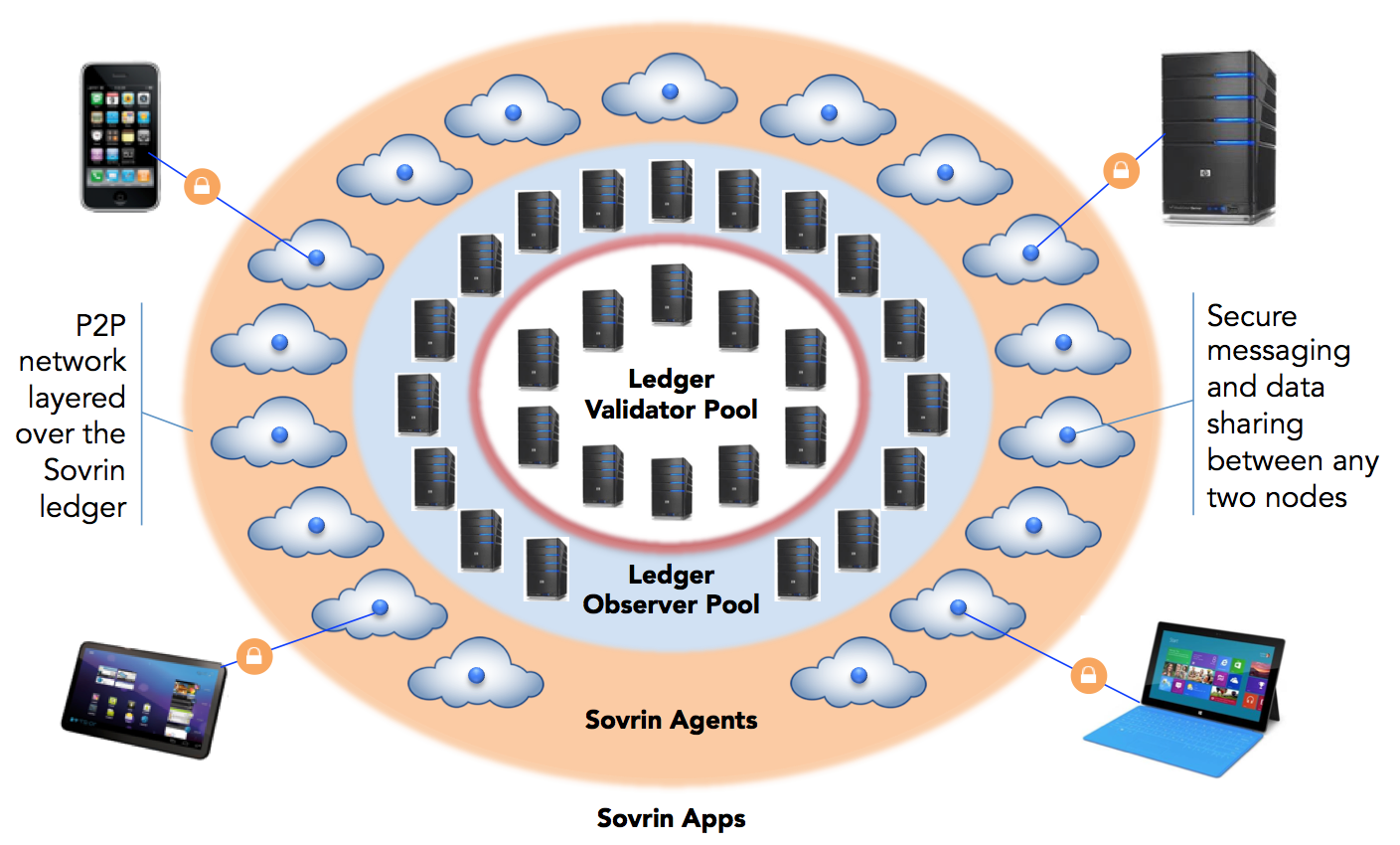


## Off-Ledger Interactions

One of the other critical features of a public identity ledger is that it **must not** expose any PII directly on the ledger. To meet this requirement, Sovrin stores only the following information on the ledger:

1. **DIDs**—which in the vast majority of cases will be pairwise-unique
2. **Public keys**—to enable encrypted communications with the identity owner
3. **Service endpoints**—to enable interaction with the identity owner via an associated agent (see below)
4. **Proofs**—hashed or zero-knowledge proof artifacts that enable issuers, identity owners, and relying parties to prove that certain information existed as of a point in time

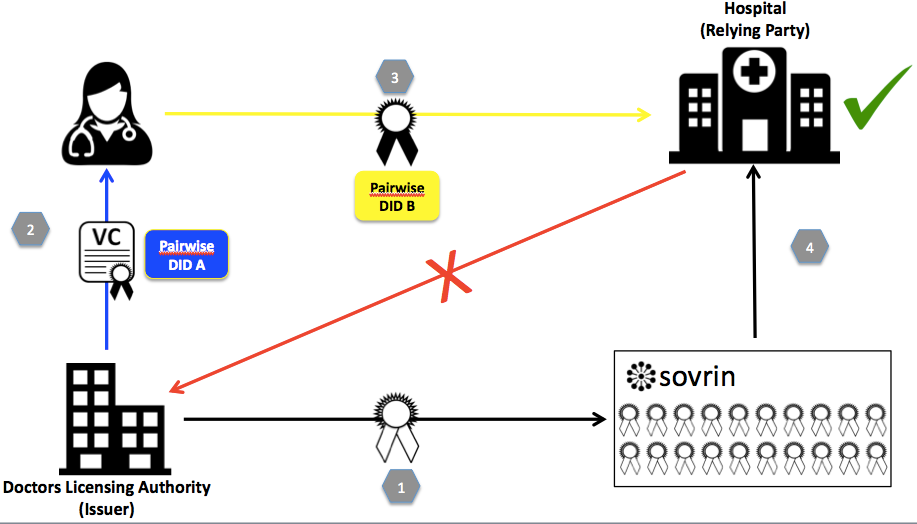
With these four basic types of data, identity owners can bootstrap peer-to-peer interactions directly between *off-ledger agents* operating on the identity owner’s behalf. In addition to facilitating off-ledger exchange of claims and proofs, agents can be extended to support a wide variety of secure applications. This next layer of peer-to-peer off-ledger agent interaction is shown in the following diagram:



## Verifiable Claims and Zero Knowledge Proofs

To establish trust in a peer-to-peer, decentralized identity system, identity owners must be able to exchange identity claims that relying parties can verify. This ecosystem of “verifiable claims” is in fact ideally supported by a distributed identity ledger. Claims can be issued against pairwise-unique DIDs registered on the ledger, signed with the issuer’s private key, and verified by the issuer’s public key from their DDO. Claims exchange and verification can all be handled by off-ledger agents using the service endpoints discovered in the DDO.

Furthermore, this architecture can also support zero-knowledge proofs (revocation material for which can also be stored on the ledger), giving all identity owners the power of selective disclosure at a scale that has never been possible before. The following diagram steps through a typical example of a verifiable claim interaction:



1. Before issuing claims, a Doctor's Licensing Authority (Issuer) creates a claim schema (or uses an existing one), creates public keys, and creates a revocation registry entry on the ledger.
2. Doctor's Licensing Authority (Issuer) supplies Doctor Anna with a Verifiable Claim confirming she is a Medical Doctor via Pairwise DID A.
3. Doctor Anna presents Proof of a subset of her claim—whatever she chooses—to a Hospital (Relying Party) she’s applying to work for via Pairwise DID B. Anna also includes proof that her claim has not been revoked by the issuer.
4. Hospital verifies the authenticity of the claim without contacting the Issuer. (Although in this scenario, Anna will be sharing attributes that strongly identify her to the hospital, the protocol also allows for attribute-based authentication where she may not be strongly identified.)

## The “Indy” Code Base

The initial set of tools provided in the proposed Indy project includes:

* **Indy:** A repository that points to the other repositories and may include tools to help developers get the relevant parts of the project. (This project does not exist in the current repositories),
* **Indy-common:** Common utility functions and other tools needed across the Indy project, formerly known as [sovrin-common](https://github.com/sovrin-foundation/sovrin-common).
* **Plenum-consensus:** This is the implementation of BFT used by the default ledger of the Indy project.
* **Plenum-ledger**: A general purpose ledger formerly known as the [Evernym ledger](https://github.com/evernym/ledger). It implements a view of state using the ledger entries provided from the BFT component. It also implements tools for managing node membership, voting for network management and a minimal number of features for servicing identity transactions. Though it could be used as a general ledger, it is intended to serve as a base for identity, and over time could be replaced by another general purpose ledger, provided the functionality needed to preserve identities privacy, establish diffuse trust and service identity transactions can be accomplished in a simple/reliable way.
* **Indy-anoncred:** Implementation of base tools for supporting a privacy preserving identity on a ledger, formerly published as [Evernym’s anoncred project](https://github.com/evernym/anoncreds). This includes tools for verifiable claims (such as those being worked on by the W3C Verifiable Claims Task Force), and privacy-preserving credentials used for creating selectively-disclosing proofs about verifiable claims and their contents such that an identity can prove in zero knowledge that claims associated with different distributed identifiers are associated with the same individual.
* **Indy-node:** The main code for running a node for an identity-specific ledger (this could be considered a cross-anchored side chain to any use-case specific ledger, or the primary chain used to create multiple use-case specific ledgers), formerly known as [sovrin-node](https://github.com/sovrin-foundation/sovrin-node). This code currently extends Indy-ledger to implement identity transactions and the appropriate genesis blocks for starting a new Indy network.
* **Indy-agent:** The off-ledger components required to accomplish basic key management, data sharing and secure identity to identity interactions, formerly known as [sovrin-agent](https://github.com/sovrin-foundation/sovrin-agent).
* **Indy-client:** This is a cli interface for talking to an Indy-node or Indy-agent. As the API matures this may be folded into Indy-node and Indy-agent, formerly known as [sovrin-client](https://github.com/sovrin-foundation/sovrin-client). This project contains the current reference agent code, which is being refactored in the new sovrin-agent repository.
* **Indy-client-rust**: Currently under development using the name [sovrin-client-rust](https://github.com/sovrin-foundation/sovrin-client-rust), this is a C-callable library written in Rust that can be used to embed indy functionality into other systems. It is similar to indy-client in functionality.

These projects are either currently licensed as Apache License v2.0 (or will be as they are ported to new Hyperledger/Linux Foundation repositories).

# Effort and resources

There are existing working groups at the Sovrin Foundation working to improve these code bases, provide documentation and support to users and new developers.

* **Ledger WG:** Jason Law is coordinating efforts around the ledger nodes
* **Agent WG:** Nathan George is coordinating work around agents (the part of an identity that exists off-ledger)
* **Trust Framework WG:** Drummond Reed is coordinating the development of legal and technical documentation around building and managing an instance of Indy for the Sovrin Network.

Evernym has several engineers working to improve the project, and there is participation from a variety of other organizations interested in using the project for their own use cases. Sovrin is currently in the process of bootstrapping it's own Community program (developer engagement, meetups, hackathons, etc.) and INDY will be critical in these efforts.

# How To

The [Getting Started Guide](https://github.com/sovrin-foundation/sovrin-client/blob/stable/getting-started.md) provides a quick way to install and test against the Sovrin development sandbox. There are other documents describing how to set-up a new network with its own nodes.

The project is now hosted at [the Sovrin Foundation's github repository](http://github.com/sovrin-foundation) and [several public Evernym github repositories](http://github.com/evernym).

CI is currently being managed internally by Evernym and run in association with all pull requests. Moving to a publicly visible CI system is currently under discussion within the project (options include the official Hyperledger Travis CI system as well as something hosted by the Sovrin Foundation, if the Travis CI system cannot meet the current requirements).

The codebase has significant tests and reasonable test coverage; improvements in the form of pull requests are welcome.

The code includes documentation in the form of README.md files and other git-managed documentation. User documentation and discussion can be found on the [Sovrin website](https://www.sovrin.org/docs/), [Sovrin Forums](http://forum.sovrin.org/) and [Sovrin Slack](https://sovrin.slack.com).

# Closure

This project is expected to be the home for development efforts for the Indy codebase for the foreseeable future. If the Sovrin Foundation itself were to shut down or decide to move back to its own branch of the codebase and Hyperledger’s interest in the project as a home for identity tools that can be leveraged across Hyperledger projects diminishes, the project could be closed.

# Acknowledgements

Thank you to all reviewers for the insightful comments. Please feel free to add yours.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Organization | Email |
| 1 | Haranath Tirumalasetty |  | [tirumaha@gmail.com](mailto:tirumaha@gmail.com) |
| 2 | Brian Behlendorf |  | [bbehlendorf@linuxfoundation.org](mailto:bbehlendorf@linuxfoundation.org) |
| 3 | Drummond Reed | Evernym, Sovrin Foundation Trust Framework Working Group Chair | [drummond.reed@evernym.com](mailto:drummond.reed@evernym.com)  [drummond.reed@sovrin.org](mailto:drummond.reed@sovrin.org) |
| 4 | Sean Bohan | Evernym, Sovrin Foundation | [sean.bohan@evernym.com](mailto:sean.bohan@evernym.com)  [sean.bohan@sovrin.org](mailto:sean.bohan@sovrin.org) |
| 5 | Markus Sabadello | Danube Tech, Sovrin Technical Governance Board | markus@danubetech.com |
| 6 | Sam Curren | Sovrin Foundation | telegramsam@gmail.com |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |