2019 Q4 Hyperledger Indy

Projects

Distributed Ledger

- Indy-node
- Indy-plenum

Client Tool

• Indy-sdk

Shared Components

- Indy-hipe
- Indy-test-automation

Project Health

Indy is a healthy project. Indy's codebase has 23962 commits from 177 unique contributors. This represents an increase of 7 contributors this quarter and about 2,000 additional commits. We had commits from 41 different authors in the last three months. Forums and chat channels are monitored and a variety of participants are contributing helpful responses to questions.

Questions/Issues for the TSC

We continue to track the same issues as in previous quarters.

Incompatible agent implementations

Update:

The launch of project Aries had the intended effect of focusing attention on standards for agent implementations. We will continue to support that effort as we migrate portions of Indy SDK into Aries core libraries. We will not be tracking this as an issue in our future quarterly reports.

Measuring the size and make-up of our user community

Update:

We played a bit with tools for counting contributors, but no other significant progress.

Future work planned:

- · We expect that the Indy contributor community will continue to shrink as much of the attention shifts to Aries.
- Work with Hyperledger to get analytics about web sites, and Rocket Chat usage.
- Begin measuring usage of the Sovrin forums: new contributors, questions asked, and questions answered
- We will also explore the use of Github's contributor tool.

Build Issues

Update:

Our transition to the GitLab CI build pipeline managed by the Sovrin Foundation is going slower than expected. The Indy SDK pipeline is complex, and GitLab's integration with GitHub is not yet mature. We have found credible solutions to the problems we have seen, but the project has not had sufficient developer focus to come to completion. We are still relying on the Jenkins instance managed by the Sovrin Foundation for our production CI / CD pipeline.

Future work planned:

· We need a new commitment from the various teams who have expressed interest.

Releases

August 2019:

Indy SDK 1.11.0

- Enhancements and bugfixes added for TAA and Endorser txns
- Updated Indy-SDK CI/CD pipelines to test, to build and to publish Android artifacts for Libvcx.

- Improved state proof verification to support pagination.
- Bugfixes

Indy SDK 1.11.1

- Supported endorsing of transactions in Indy-CLI and Libvcx.
- Added new functions to Anoncreds API to rotate credential definition.
- Added sign/verify with payment address functions to Libvcx.
- Supported state proof verification for GET_TXN request.
- Extended config parameter of indy_open_pool_ledger function.
- Extended Libvcx initialization config to accept pool configuration.
- Supported new platforms Ubuntu 18.04 and Centos:
- Bugfixes

Indy Node 1.9.1

- New DIDs can be created without endorsers
- Transaction authors don't need to be endorsers
- TAA acceptance should use date, not time
- Bug fixes

Indy Node 1.9.2

- Stability fixes
- Endorser support fixes and improvements
- Improving GET_TXN to be able to query just one node the same way as for other GET requests

September 2019:

No Releases

October 2019:

Indy SDK 1.12.0

- Minimal EXPERIMENTAL support of Fully-Qualified identifiers:
 - omit or set "1.0" to use unqualified identifiers.
 - set "2.0" to use fully qualified identifiers.
 - added correspondent did qualify command to Indy-CLI.
 - general format of fully-qualified identifier is <prefix>:<method>:<value>.
 - extended did_info parameter of indy_create_and_store_my_did function to accepts optional method_name filed. This field should be used to create fully qualified DID.
 - ° all functions can work with fully-qualified identifiers (new way) as well as with unqualified.
 - added a new function -- indy_to_unqualified -- that gets unqualified form of a fully qualified identifier.
 - proof requests now support versioning (ver field) -- now it specifies whether restrictions are full qualified or not.
 - The same format of identifiers will be used in generated proof and must be used for proof verification.
 - ° added a new function -- indy_qualify_did -- that updates DID stored in the wallet to make it fully qualified, or to do other DID maintenance.
 - all functions in Ledger API can accept fully-qualified identifiers but always return results in an unqualified form.
 - extended VCX provisioning config to accept optional did_method filed. This field should be used to create fully qualified DIDs.
- Migrated Android onto the API v21 and NDK 20.
- Supported MacOS builds for Indy CLI.
- The default value of Protocol Version was changed on 2. Henceforth indy_set_protocol_version function should be called if you are going to work with Indy-Node 1.3 and less.
- Bugfixes

Indy Node 1.10.0

- PBFT View Change implementation (not enabled yet) and corresponding code improvements
- BLS multi-signature fixes and improvements
- The latest version of ZMQ library support
- Stability fixes

Overall Activity in the Past Quarter

Key activity over the past quarter:

- Migrating large parts of indy-agent, indy-hipe and indy-sdk to Aries, including infrastructure such as Jira issues.
- Improving the ledger to make it more stable, performant, provably correct, and easy to maintain. We made a lot of progress and are really pleased with Plenum.

Our contributor community collaborates a lot on Jira issues, in pull requests, using Rocket Chat, and in weekly meetings. But we don't use the mailing list very much. We do not have clear analytics, but the majority of questions appear to receive answers within a few days.

Current Plans

- We are increasingly pushing contributors who want to add new features to the Indy SDK to focus instead on the Aries libraries. This impacts the language wrappers the most.
- However, we are continuing to maintain Indy SDK and wrappers for existing users to allow them a smooth transition, including adding compatibility for Aries protocols to LibVCX.
- The next round of ledger work will remove the RBFT replicas, instead implementing Aardvark BFT. This addresses a number of issues we see with replicas, and should significant improve performance and scalability giving us headroom in the adoption of production Indy networks for some time to come.
- Other initiatives include improving CI / CD (as described above), adding support for W3C DID and Verifiable Credential primitives, and adopting the new approach to revocation provided by Ursa's anoncreds 2.

Maintainer Diversity

The weekly Indy Maintainers call continues to be the medium by which maintainers coordinate work, discuss critical issues to the Indy codebase, and agree on HIPEs. In the past quarter, the teams at the Sovrin Foundation and at Kiva have been much more involved in maintaining Indy. There is a proposal to start cross-organizational reviews of pull requests in order to increase knowledge sharing, but we don't yet have commitments from sufficient organizations.

Contributor Diversity

Fewer people work on Indy now that many contributors focus instead on Ursa and Aries. Evernym still sponsors the majority of development, but the developers at the Sovrin Foundation are confident in the code base. Developers from the government of British Columbia, Deutsche Telekom, and Kiva are increasingly familiar with the code base on a deep level.

POCs, Pilots, Projects

There are now too many Indy projects to list them all here. Recent public disclosures include:

- Onfido and UK FinTech ecosystem
- esatus
- CULedger and Desert Financial
- Farmer Connect

Additional Information

- Key channels on Rocket Chat: #indy, #indy-sdk, #indy-node, #indy-maintainers
- Join the Indy Mailing List: https://lists.hyperledger.org/g/indy

Reviewed by

- Angelo De Caro
- Arnaud J Le Hors
- Christopher Ferris
- Dan Middleton
- Gari Singh
- Hart Montgomery
- Mark Wagner
- Nathan George
- Swetha Repakula
- Tracy Kuhrt
- Troy Ronda