Hyperledger Indy Besu as a permissioned ledger in Self-sovereign Identity

Alexander Shcherbakov

Abstract: Self-sovereign Identity (SSI) represents an approach to digital identity that prioritizes privacy and empowers individuals to maintain control over the information associated with their identity. This approach aligns with GDPR and similar regulations and is gaining adoption across various governments, non-profit organizations, and commercial entities worldwide. A foundational element in SSI is a Verifiable Data Registry (VDR), which serves as a trusted repository for registering and accessing public keys, schemas, identifiers, and other data. A natural choice for a VDR is a distributed ledger or blockchain. Among the most stable and popular frameworks for SSI is Hyperledger Indy. Indy includes a custom implementation of a public permissioned ledger as a VDR. The Indy community has been developing a new experimental approach for a VDR in Indy: a permissioned ledger based on Hyperledger Besu. In this paper, we are going to discuss the importance, benefits, and technical details of this initiative.

Keywords: SSI, Self-sovereign Identity, Decentralized Identity, Verifiable Credentials, DID, W3C VC, VDR, AnonCreds, Hyperledger Indy, Hyperledger Aries, Hyperledger Besu, Distributed Ledger Technologies, Blockchain, Permissioned Ledger, Ethereum

1 About Self-Sovereign Identity and Verifiable Data Registry

Self-sovereign Identity (SSI), or Decentralized Identity, is an approach to digital identity focusing on privacy and enabling individuals to maintain control over the information associated with their identity [PR21]. SSI is not tied to a specific framework or library; instead, it encompasses multiple specifications, standards, frameworks, and tools that implement SSI principles. Three key concepts are fundamental in SSI: Verifiable Credentials (VC), Decentralized Identifiers (DID), and Verifiable Data Registries (VDR).

A credential is a set of one or more claims made by an issuer. Generally, these claims describe certain properties of the credential holder. A verifiable credential (VC) [VC24] is a tamper-evident credential with authorship that can be cryptographically verified. Verifiable credentials can be used to build verifiable presentations, which can also be cryptographically verified.

A Decentralized Identifier (DID) [DI24] refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.). In contrast to typical, federated

1 DSR Corporation Europe, Rua Sa da Bandeira 819, 2 DTO, Porto, 4000-438, alexander.sherbakov@dsr-corporation.com

doi:10.18420/OID2024_11
identifiers, DIDs may be decoupled from centralized registries, identity providers, and certificate authorities. DIDs are URIs that associate a DID subject with a *DID document* allowing trustable interactions associated with that subject. DID document is a set of data describing the DID subject, including mechanisms, such as cryptographic public keys, that the DID subject or a DID delegate can use to authenticate itself and prove its association with the DID.

Finally, a *verifiable data registry (VDR)* [VC24][DI24] is a system that facilitates the creation, verification, updating, and/or deactivation of decentralized identifiers, DID documents, keys, and other relevant data, such as verifiable credential schemas, revocation registries, issuer public keys, and so on. In other words, VDR can be considered as a trusted place used to publish and access public keys for verification of verifiable credential signatures.

Example verifiable data registries include trusted databases, decentralized databases, government ID databases, and distributed ledgers. Distributed ledgers, in particular, are popular options for VDRs due to their effective combination of decentralization and trust. While the use of public permissionless ledgers and blockchains (such as Ethereum Main Net) as a VDR can be a reasonable option for many use cases, there are scenarios where a permissioned ledger becomes the only viable choice. This is particularly evident in government-driven ledgers or situations where the use of crypto tokens is deemed undesirable.

![Fig. 1: SSI Workflow](image-url)
Let's explore a standard SSI-based workflow (Fig. 1) by integrating all three concepts [VC24], [DI24]:

1. **Pre-requisites.** The Issuer possesses a private-public key pair used to sign verifiable credentials. The key is linked to a DID (DID1) by including the public part in the corresponding DID Document. The DID and DID Document are then published to a Verifiable Data Registry (VDR), such as a distributed ledger.

2. **Issuance.** The Issuer issues a credential for a holder identified by a DID2. The credential is signed by a key associated with DID1. Both DID1 and DID2 are included in the credential, and hence signed. The Issuer may also require the Holder to prove ownership of DID2 keys before issuance. The issued verifiable credential (claims and issuer’s signature) is sent to and stored by the Holder, usually in the Holder’s mobile or web wallet.

3. **Presentation.** When the Verifier requests proof from the Holder, the Holder examines the wallet and selects verifiable credential(s) that can satisfy the proof request. The Holder then creates a verifiable presentation comprising required information from the selected verifiable credential(s), as well as the corresponding Issuer’s signature(s) (such as DID1’s signatures) and the Holder DID2’s signature (a proof of VC ownership by the Holder). The Verifier resolves Issuer’s public keys associated with the Issuer’s DID (DID1) via the VDR and verifies the signatures.

## 2 Hyperledger Indy Besu

The Indy community has been actively working on an experimental approach for a Verifiable Data Registry (VDR) within Hyperledger Indy—a permissioned ledger based on Hyperledger Besu [IB24]. To provide a comprehensive understanding, we will begin with an overview of both Hyperledger Indy and Hyperledger Besu. Subsequently, we will introduce the Indy Besu approach, delving into its importance, benefits, and technical details.

### 2.1 About Hyperledger Indy

Hyperledger Indy [HI24] is an open-source project under the Linux Foundation's Hyperledger umbrella. It is a decentralized and self-sovereign identity (SSI) management platform designed to give individuals, organizations and devices control over their digital identities. Indy provides the tools and protocols necessary to create, manage and verify digital identities in a secure and privacy-preserving manner. Many
Hyperledger Aries frameworks depend on Indy or some API from it under their hood. Indy consists of two main components, ledger and client SDKs:

- Indy Ledger [IN24] is a public permissioned distributed ledger providing a decentralized, secure and tamper-evident infrastructure for managing identities. It's based on RBFT consensus protocol implemented as part of the Indy Plenum [IP24] project. Indy ledger can be used as a verifiable data registry (VDR) with did:indy [ID24] and did:sov [SD24] methods. It can also be used as a registry for CL AnonCreds [AC24] verifiable credentials to publish credential schemas and issuer's credential definition, public keys, revocation registries, etc.

- Indy SDK [IS24] is a collection of software libraries, tools and APIs that developers can use to build applications and systems that incorporate SSI features based on Indy. It includes the components for communication with the Indy Ledger, managing CL AnonCreds [AC24], verifiable credentials, establishing pairwise connections, wallet functionality, CLI, etc. The code is written in Rust and contains wrappers for all popular programming languages and platforms, including mobile.

Indy is a graduated Hyperledger project since 2019. Indy Ledger has successfully run in production for many years without significant issues as part of Sovrin [So24] and other networks.

2.2 About Hyperledger Besu

Hyperledger Besu [HB24] is a Java-based Ethereum client that has been an active and graduated Hyperledger project since 2020. This versatile framework serves two main groups of use cases:

1. **Public Networks**: These include public Ethereum networks.

2. **Private Permissioned Networks**: These are often associated with enterprise or supply chain ledgers.

For each category, Hyperledger Besu adopts different approaches concerning consensus protocols, supported features, components, etc.

Hyperledger Besu features a pluggable architecture and encompasses multiple implementations of consensus algorithms, including proof-of-stake (PoS), proof-of-work (PoW), and proof-of-authority (PoA). In private permissioned networks, consensus algorithms such as IBFT 2.0, QBFT, or Clique, which fall under the proof-of-authority category, are commonly employed. For implementing application-specific business logic and transactions, developers can use Solidity smart contracts.
2.3 Why New Indy Ledger

The Indy Ledger [HI24] stands out as one of the most stable and widely adopted frameworks for decentralized and self-sovereign identity. It has been successfully deployed in production systems and played a pioneering role in the early days of SSI projects, significantly contributing to the adoption and popularization of the self-sovereign identity concept. Despite its reputation for 'just working,' it is worth noting that there has been a lack of ongoing maintenance and implementation of new features.

The Indy Ledger project was initiated in 2016. At that time, stable frameworks for a public permissioned ledger were not readily available. Consequently, Indy Ledger not only incorporates the business logic of SSI-specific transactions but also encompasses the implementation of the auxiliary blockchain framework itself, including components such as the consensus protocol, permissioned logic, ledger, storage, etc. This dual role contributes to a sizable and intricate codebase, making maintenance a non-trivial task.

When the Indy project commenced, it offered a functional implementation of SSI principles at a time when modern SSI standards, such as W3C VC [VC24] and W3C DID [DI24], had not yet been established and finalized. To maintain its role as a driving force in the SSI space, Indy now requires the implementation of new features and support of recent specifications.

While there are various options for CL AnonCreds registries beyond Indy, such as cheqd or Cardano [AM24], it's noteworthy that all of these alternatives are built on permissionless proof-of-stake ledgers. While a permissionless ledger can be a reasonable choice in many cases, there are specific scenarios where a permissioned ledger, like the existing Indy Ledger, serves as a more suitable alternative.

The current Indy Ledger has limitations concerning performance (throughput, latency). Typically, performance is not a critical factor for SSI cases, as the primary actor who needs to write to the ledger is the verifiable credentials Issuer. In many cases, writing just a couple of transactions (e.g., publishing issuer’s public keys, credential schemas, etc.) is sufficient. However, if support for the revocation of verifiable credentials is required [AC24], issuers may need to write to the Indy ledger frequently (e.g., on each VC revocation). This increased frequency places higher demands on the ledger's throughput.

Decentralization is another crucial aspect. While the number of nodes in a permissioned network is typically limited, having more nodes participate in the validation of new transactions is advantageous for trust. The current Indy Ledger network often operates under the assumption of a limit of 25 nodes [So24], beyond which performance experiences a significant drop.

In response to the challenges observed in the current implementation of the Indy Ledger and the recognized need for a public permissioned ledger as an option for VDR, a new Indy Besu [IB24] initiative was proposed, and the first MVP was implemented.
The primary objective of this initiative is to preserve the favorable characteristics of the current public permissioned Indy Ledger while simultaneously reducing complexity, simplifying maintenance, enhancing performance and scalability, accelerating the development of new features, improving the end-user experience, and lowering the operational costs of Indy nodes.

A key enhancement involves replacing the custom consensus protocol implementation with Hyperledger Besu [HB24], a stable and maintained framework. This shift allows Indy Ledger to concentrate on SSI-specific business logic.

2.4 Indy Besu Benefits

In contrast to the current Indy Ledger [IN24][IP24], the new Indy Besu [IB24] has a more compact and simpler codebase. This is achieved by encapsulating blockchain complexity within the Hyperledger Besu framework, upon which the new Indy Ledger is built.

It has the following benefits and advantages for the Indy community, maintainers and users:

- Business logic (transactions) is implemented in Solidity, one of the most popular and adopted languages for smart contracts. Solidity smart contracts are easy and understandable. This will attract new developers, make it much easier to support the code, and add new technologies that are common on the market.

- The new consensus protocol significantly increases network throughput (up to 10 times) [FLK22], which can be especially beneficial for revocation features where verifiable credential issuers may publish quite significant number of transactions to the ledger.

- Hyperledger Besu allows to increase the number of validators in the network, which improves decentralization properties and trust of the network [FLK22].

- Permissioned mode (proof-of-authority consensus) follows the same principles as the current Indy ledger implementation.

- Hyperledger Besu is part of the Hyperledger family, so it is a logical bridge between two graduated Hyperledger projects.

- The new Solidity-based Indy contracts can be run on a public Ethereum Main Net as an alternative to the permissioned case.

- There is a possibility to deploy SSI/Indy logic into existing private permissioned deployments based on Hyperledger Besu (for example, to extend supply chain cases).
• A possibility to implement light client solutions [LC24].
• Indy Besu has lower hardware requirements [BD24][So24][IN24], contributing to a reduction in the overall cost of operating and maintaining Indy nodes.

Moreover, the new implementation is compatible with the old one (did:sov [SD24] and did:indy [DI24] methods), and have clear migration guides for existing deployments.

2.5 Indy Besu Technical Details

Indy Besu ledger [IB24] serves as a Verifiable Data Registry (VDR) for verifiable credentials, supporting both W3C VC format [VC24] and Hyperledger AnonCreds format [AC24]. Indy Besu, like the current Indy Ledger, can be utilized as a VDR for Hyperledger CL AnonCreds.

Additionally, similar to the existing Indy SDK and Indy VDR libraries, Indy Besu features a client SDK written in Rust, complemented by wrappers for popular languages and seamless integration with Hyperledger Aries projects.

Fig. 2: Indy Besu Components

Indy Besu supports both the did:ethr and an extension of did:indy (did:indy:besu) methods. The choice of an appropriate DID method depends on particular deployment specifics and other requirements, such as permissionless or permissioned ledger, the need for data migration, etc. We expect that one or another method will be selected as the main one as a result of adoption.

The did:ethr DID method [ED24] is a well-known, standardized approach for DID management on Ethereum networks. This method adheres to ERC-1056 [ERC24] and is designed to utilize Ethereum addresses as fully self-managed DIDs. The method is
preferrable for permissionless deployments, as it requires lower gas costs.

The new did:indy:besu (extension of did:indy) method [IB24] is designed to be compatible with the legacy did:indy and did:sov by supporting DID aliases, allowing legacy identifiers to be mapped to and associated with the new identifiers. Additionally, it utilizes Ethereum addresses as fully self-managed DIDs and provides support for light client approaches similar to the existing Indy Ledger. This method is recommended for permissioned deployments.

A compatibility layer is in place to facilitate the migration of existing Indy Ledger deployments (with did:indy [ID24] or did:sov [SD24] identifiers) to the new did:ethr or did:indy:besu based networks.

Fig. 3: Indy Besu Smart Contracts

Much like the existing Indy Ledger, Indy Besu operates as a public permissioned ledger, allowing public access to read requests while restricting write requests and the setup of new validator nodes. Leveraging the permissioned capabilities of Hyperledger Besu, Indy Besu extends these functionalities with roles-based authorization for accounts.
Not all identity owners may have permissions to write transactions to a permissioned ledger. Therefore, similar to the existing Indy Ledger, Indy Besu has transaction endorsement support. The transaction endorsement is a mechanism for executing transaction writes to the ledger by a special party with an Endorser role while preserving
the original transaction author as the entity owner. This approach is applicable to all transaction types, including Hyperledger AnonCreds, did:ethr and did:indy/did:indy:besu transactions.

![Diagram](image_url)

Fig. 6: DID transaction endorsement sequence diagram

3 Summary

The Indy Besu initiative is currently in an experimental state and resides in a separate Indy repository [IB24]. A minimal valuable product (MVP) has been already implemented. This project is attracting considerable attention from the Indy community and existing Indy deployments. There are promising indications that it has the potential to gradually replace the legacy Indy Ledger implementation and contribute to Self-Sovereign Identity adoption.

Bibliography


Hyperledger Indy Besu as a permissioned ledger in Self-sovereign Identity


