A New Approach to Client Onboarding Using Self-Sovereign Identity and Distributed Ledger

Conference Paper · March 2019

DOI: 10.1109/Cybermatics_2018.2018.00205

Reza Soltani
York University

Uyen Trang Nguyen
York University

2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing,
Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics

A New Approach to Client Onboarding using Self-Sovereign Identity and Distributed Ledger

Reza Soltani
Department of Computer Science and Engineering
York University
Toronto, Canada
rts@cse.yorku.ca

Uyen Trang Nguyen, Aijun An
Department of Computer Science and Engineering
York University
Toronto, Canada
utn@cse.yorku.ca

***Abstract*— Existing client onboarding and Know Your Customer (KYC) processes are typically slow, expensive and often accomplished in-person. Moreover, the current identity management models in practice deprive users from having complete control over their digital identity data. Users' identity attributes are stored on multiple centralized repositories, which often follow inadequate security policies. In this paper, we take advantage of Hyperledger Indy, a public and permissioned distributed ledger technology (DLT), to develop a digital onboarding framework based on the Self-Sovereign Identity (SSI) principles. With this framework we take a step towards tackling a number of weaknesses in current KYC processes and identity management models, while addressing the requirements associated with SSI, Privacy by Design and European Union's General Data Protection Regulation (GDPR).**

Keywords—blockchain, identity management, distributed ledger, self-sovereign identity, hyperledger indy, KYC

I. Introduction

The current identity management models that rely on central identity providers to manage user data, have led to frequent security and privacy concerns, and the creation of honeypots for hackers. Recent breaches such as the Equifax incident [1] show users' personal identifiable information (PII) are not always protected by adequate security measures. Moreover, the current identity management landscape revolves around the needs of identity and service providers rather than identity owners and users.

A. Problem Motivations

There are a number of driving reasons behind our proposed client onboarding framework. The introduction of GDPR, the challenges faced by underbanked and under-identified individuals, and the increasing cost of KYC are all key motivations of this research.

1) GDPR

General Data Protection Regulation (GDPR) [2] is European Union's new set of policies on data protection. Its purpose is to reshape the way organization across the Europe handle citizens' personal data. This law officially took effect on May of 2018. While this regulation focuses on the citizens of European Union (EU), any organization outside of EU that collects or processes data on EU's citizens will also be affected. The introduction of GDPR necessitates the development of solutions with adequate security and privacy measures in place. Section V discusses the key principles of GDPR related to personal data.

2) Underbanked and Under-identified

The magnitude of underbanked people living in rural areas, with no access to any banking infrastructure is staggering. Based on the latest reports by World Bank, about 2 billion people world wide do not have access to financial services [3]. The number of people who are living in urban centers in developed countries, but are underbanked is also surprisingly high. A 2015 report suggests 20% of U.S. households are underbanked [4].

The World Bank has set an objective of providing universal access to financial services by 2020 [5]. These challenges necessitate the exploration for an interoperable, multi-jurisdictional and scalable identity management system, accessible by segments of the population without easy access to brick-and-mortar financial institutions.

3) Cost of Know Your Customer (KYC)

Know-your-customer (KYC) is the process of having financial institutes engage with their prospective clients to verify their identity [6]. Through this process, financial organizations obtain better insight about their clients, and establish a rapport by which they gain a better understanding of how their clients' funds are obtained, accessed and withdrawn. KYC is a necessary process to prevent money laundering and terrorism financing.

The cost of complying with KYC and anti-money laundering (AML) regulations is increasing each year. Banks that do not comply with these regulations are penalized and may be put out of business. On the other hand, KYC is a slow, inconvenient, and at times, insecure process for the clients [7]. A typical client registeration process involves the manual submission of physical identity documents, which can be easily forged or lost. Furthermore, banks are sometimes organized in product silos which lead to an increase in complexity of the onboarding process. These issues encourage us to seek an overhaul of the existing KYC processes.

B. Contributions

The Self-Sovereign Identity (SSI) model [8] is the next evolution of identity management paradigm in which users have complete ownership and control over their digital identity. The distributed ledger technology (DLT) provides a decentralized consensus-based approach to transaction processing. DLT assists in the realization of SSI by providing

978-1-5386-7975-3/18/$31.00 ©2018 IEEE
DOI 10.1109/Cybermatics_2018.2018.00205

the necessary functionality to perform distributed identity management tasks.

In this paper, we develop a client onboarding framework named *KYC2*. This framework is based on Hyperledger Indy [9], a public permissioned, open source, distributed ledger technology. Our framework simulates a financial client onboarding scenario among a client and two banks. This is followed by an assessment of the framework based on the principles of SSI, General Data Protection Regulation (GDPR) and Privacy by Design (PbD) [10].

The remainder of this paper is organized as follows. Section II provides a background on SSI, blockchain and distributed ledger as well as the technologies underpinning SSI. Section III provides a summary of related work on SSI and DLT. Section IV describes our KYC2 framework and the KYC2 simulation in detail. In Section V, we present an evaluation of the framework against SSI, GDPR and PbD principles. In section VI, we summarize the paper and outline our future work.

II. Background

To address the problems discussed in Section I, we utilize Hyperledger Indy which facilitates an SSI architecture, by relying on DLT, Verifiable Credentials [11], Decentralized Identifier (DID) and DID Descriptor Object (DDO) [12]. These concepts are discussed in the following sub-sections.

A. *Identity Management Models*

An identity is a representation of an entity in a specific application domain [13]. A digital identity is a partial identity in digital format. For every given entity there can be one or more unique or non-unique digital identities [14]. The growth of the Internet and online services has ignited the quest for practical, secure and privacy preserving digital identity and access management (IdM) architectures [15]. This has led to the development of a series of identity management models. Fig. 1 depicts the evolution of these identity models.

The most basic model is the *isolated identity model* in which the service provider (SP) and the identity provider (IdP) are paired together, that is, the service provider is also responsible for managing the identification, authentication and authorization of its users.

The second model is called the *centralized identity model* [13] in which the IdP is decoupled from the SP. In this model, the digital identities are stored and managed by the IdP. The users authenticate with the IdP prior to accessing the SP.

The third model is known as the *federated identity model* [16] in which multiple SPs form a federation with one or more IdPs, allowing the user to use the same credential to authenticate and access any of the federated SPs.

Central Identity
Federated Identity
User Centric Identity
Self-Sovereign Identity

Fig. 1. Evolution of identity management models

The fourth model is known as the *user-centric identity model* [13]. This model lays emphasis on the user experience and privacy control. It places the users in the center of decision-making by allowing the user to define the policies by which their identity attributes are shared with SPs.

Self-Sovereign Identity can be seen as the next evolution of identity management models. SSI revolves around ten core principles, namely: existence, control, access, transparency, persistence, portability, interoperability, consent, minimization and protection. These principles are developed by Christopher Allen, and are influenced by Kim Cameron's laws of identity [17]. The SSI principles are further discussed in section V of the paper.

B. *Benefits of the Self-Sovereign Identity Model*

Self-Sovereign Identity is the latest iteration of identity management models. The SSI model attempts to address the inherit problem found in existing identity models in which users' identity data are scattered among multiple distinct identity providers, and without the users' direct control. The goal of SSI is to provide users complete control over their identity data. This model reduces the likelihood of data breaches and identity frauds affecting the IdPs, as it is no longer compulsory for the identity providers to store users' identity data.

SSI users have the liberty to manage their identity data on their mobile devices or on cloud repositories. Mobile devices have become an essential part of our lives. We use mobile devices to store our credentials and payment. Therefore, while physical documents and storage of identity attributes on the cloud and third party identity providers may exist for the years to come, storing identity data on mobile devices is the next natural step towards the realization of Self-Sovereign Identity.

SSI is better realized with the use of distributed ledger technology (DLT). The decentralized nature of the network ensures data integrity and availability, as well as privacy for the users, as there is no need for the continuous involvement of the identity issuer, for identity access, resolution or verification. While SSI provides a number of advantages, it begets a number of technical challenges and complexities, including effective key management mechanism. Adequate response to these challenges is a prerequisite to the adoptability and growth of the SSI model.

C. *Blockchain*

A blockchain is a cryptographically secure, decentralized and distributed ledger of information. Blockchain underpins the majority of cryptocurrencies by allowing for transactions to take place without central intermediaries [18], [19]. Bitcoin [20] and Ethereum [21] are common examples of blockchain-based cryptocurrencies. However, blockchain use cases have expanded beyond the financial services industry.

Under the hood, blockchain relies on the distributed ledger technology (DLT). In this paper, we use the term DLT to encompass all implementations of such architecture. DLT defines an immutable time-stamped append-only distributed ledger, which contains a set of cryptographically hashed transactions. DLT implementations can be divided into two categories: public and private.

A public DLT typically has a lower transaction throughput as the network is public and larger, therefore the consensus mechanism requires more time and resources [22]. Conversely, in a private DLT, only a selection of verified entities have the privilege to access the ledger and consequently, the transaction approval rate is higher. A private DLT provides more privacy but less transparency on the content of the transactions. The above statements do not infer one approach is better than the other, but rather it is an indication that they are different tools that solve different problems.

DLT implementations can be divided into two sub-categories: permissioned and permission-less. In a permission-less DLT, any entity can theoretically participate in writing into the ledger. In a permissioned blockchain, however, only the authorized entities are permitted to participate in validating and adding transactions to the ledger.

Blockchain is considered as one of the most ingenious technical inventions of today. There is a large amount of hype and excitement around blockchain and the growing field of cryptocurrency. There are new cryptocurrencies and prototypes sprouting in the market regularly, which leads to excitement but also confusion for the public and the industry. Despite its rapid growth, blockchain is still in its infancy and, consequently, requires extended and rigorous evaluation, analysis and standardization. Blockchain is not the panacea to every problem. The blockchain use cases that show potential should be articulated accurately, and undergo thorough evaluation and scrutiny, to address deployment and operational concerns such as scalability, governance, security, incentivization, potential for centralization, and power consumption.

D. Hyperledger Indy

The Hyperledger Consortium [23] is an umbrella project managed by the Linux Foundation. Hyperledger houses various DLT implementations including Hyperledger Fabric [25] and Hyperledger Indy.

Hyperledger Indy is a public, permissioned distributed ledger, purpose-built for decentralized identity. Indy is commonly associated with the Sovrin Project [25]. Sovrin is an instantiation of Indy and is managed by the Sovrin Foundation.

The following sub-sections describe the key technologies underpinning Hyperledger Indy. They include DID and DDO, Verifiable Credentials, and Anonymous Credentials [39].

1) DID

Decentralized Identifier (DID) is a globally unique identifier specification similar to universally unique identifiers (UUID). This specification does not require a centralized authority to register, resolve, update or revoke the identifiers. The ownership of DID can be cryptographically verified. Each DID resolves to a *DID descriptor object* (DDO). A DDO is a machine-readable document that contains the service endpoints, verification keys and metadata required to prove the ownership of the DID. An example of a DID is depicted in Fig. 2.

2) Verifiable Credentials

Verifiable Credentials - also known as Verifiable Claims - is a working draft specification developed by W3C Verifiable Claims working group. It describes a data structure for representing cryptographically verifiable and tamper-proof claims about a subject. A subject is an entity (e.g., a person, an organization or a device) for which claims may be made. A claim is a statement made by an entity about a subject. The claims are requested and verified by verifiers. The content of a basic Verifiable Credential is shown in Fig. 3.

By leveraging Verifiable Credentials, all identity attributes can be stored in an interoperable digital format, cryptographically signed and in control of the user.

3) Anonymous Credentials

Anonymous Credentials are a type of Zero Knowledge Proof (ZKP) [27]. They provide a protocol to prove a claim while maintaining anonymity, by not revealing the underlying identity data. Anonymous Credentials can represent a particular claim, or a predicate derived from a claim, such as "*claim A is less than value B*". For example, rather than transmitting the actual date of birth to a verifier it is possible to transmit the statement "age is over 21*"* that is cryptographically verifiable. Such exchanges maintain user privacy and prohibit the verifier to potentially impersonate the identity owner. There are a number of implementations that provide ZKP functionality. They include Microsoft U-Prove [28] and IBM Identity Mixer (Idemix) [29]. Indy builds on Idemix and extends its functionality by providing an anonymous revocation protocol using accumulators [30].

did:kyc2:21tDAKCERh95uGgAbCNHYc

Fig. 2. An example of a DID

```
{ "@context": "https://w3id.org/security/v1",
  "id": "http://example.org/credentials/1234",
  "type": ["Credential", "ProofOfAgeCredential"],
  "issuer": "https://www.bank.com", "issued": "2017-12-17",
  "claim": {
   "id": "did:kyc2:12f44b1f712ebc6f1c276e12ec22",
   "ageOver": 21
  },
  "revocation": {
   "id": "http://example.com/revocations/787",
   "type": "SimpleRevocationList2017"
  },
  "signature": {
   "type": "LinkedDataSignature2015", "created": "2017-12-18T21:00:10Z",
   "creator": "https://example.org/bank/keys/1", "domain": "json-ld.org",
   "nonce": "598c63d6",
   "signatureValue": "BavE…3JT24="}  }
```

Fig. 3. An example of a Verifiable Credential

4) Indy Architecture

The main components of an elementary Indy based framework consist of *validator nodes*, *agents* and *wallets* [25]. The validator nodes run the Plenum consensus algorithm. Plenum is an enhanced implementation of the Redundant

Byzantine Fault Tolerance (RBFT) consensus protocol [31]. The agents are software programs assigned to act on behalf of identity owners (e.g., users) to interact with other agents. The agents typically have access to a digital wallet to store cryptographic keys, and to perform cryptographic operations.

Inspired by the Sovrin Project [25], the Indy architecture supports various built-in roles such as *trustee*, *steward*, *trust anchor* and *identity owner*. A trustee is a high privileged role entitled to adding new stewards, other trustees and new trust anchors to the network. A steward is an entity that operates a validator node. A trust anchor is an entity with the permission to register new identifiers on the ledger.

The Indy architecture relies on various types of data documents to facilitate secure communication and issuance of claims. In order to issue a Verifiable Credential, a *schema* and a *claim definition* are required. A schema is a data structure that defines the fields enclosed in a Verifiable Credential. A claim definition is a data structure that contains a reference to the claim issuer and its keys, a reference to the schema, and details about the revocation procedure.

III. Related Work

A number of researchers have focused on developing decentralized public key infrastructure (DPKI). Al-Bassam [32] introduced the SCPKI framework, a decentralized public key infrastructure framework that relies on Ethereum and InterPlanetary File System (IPFS), to provide identity management capabilities. In contrast to KYC2 which relies on Hyperledger Indy, the SCPKI framework does not address all key management operations, including key recovery. Morever, [32] supports the storage of user PII on Ethereum and IPFS networks. While this is suitable for public claims such as a university certificate, it is a problematic approach for personal identity attributes.

Faisca and Rogado [33] propose a decentralized identity management model and authentication system based on Namecoin blockchain [34], WebID [35] and IPFS. Their solution uses Namecoin blockchain to manage public key based authentication and to store WebID profile addresses. Each WebID address points to a WebID profile documents stored on IFPS. In contrast to [33], KYC2 relies on DID and a permissioned DLT built for identity management use cases.

There are a number of initiatives that aim to address the current identity management challenges through DLT and SSI [36]. Blockcert [37], ShoCard [38], OpenBadges [39], Civic [40] and UPort[41] are among the most popular projects.

OpenBadges provides the ability to manage verifiable digital badges representing achievements and qualifications. The badges are stored in JSON-LD format and contain components such as assertions and profiles that describe user claims and user identity attributes respectively. As opposed to KYC2, OpenBadge does not directly work on a DLT to store identifiers. Moreover OpenBadge usage revolves around the use cases related to management of achievement badges.

Blockcert is another open standard based on SSI that facilitates the creation of applications that can issue and verify official records and certificate of achievements. The records are stored on Bitcoin public blockchain. In contrast to Blockcert, KYC2 relies on a DLT specifically built for identity management use cases, with support for data minimization features such as Zero Knowledge Proof that are not yet available in Blockcert.

ShoCard provides a meachanism to issue and verify identity claims by adding a digital fingerprint of user identity attributes on Bitcoin and other blockchains. ShoCard users can use ShoCard apps to share their identity claims with claim verifiers such as airport checkpoints. ShoCard servers are a necessary part of the ShoCard ecosystem. Shocard is not an open source project.

uPort is an SSI framework in which public keys are stored on a decentralized system such as IPFS. Similarly to Indy, it supports a method to assign unique IDs to each user and provide the users with the ability to share their identity attributes. Unlike Indy, uPort relies on Ethereum, a public permissionless blockchain. Since uPort is based on Ethereum, it supports smart contracts and smart contract IDs to represent users. There is a cost associated with the creation and exeuction of Ethereum smart contracts.

Civic is an Ethereum based identity management service with a built-in incentive mechanism. Civic support the creation, consumption and verification of identity information. Civic servers and Civic apps are essential components of the Civic ecosystem. As oppose to Indy, Civic is not an open source system.

The answer to which platform is best suited for developing an identity management framework depends on the use cases and the requirements. However, Hyperledger Indy's reliance on an open source DLT, purposely built for identity management use cases, its use of new technologies such as Verifiable Credentials, DID and Anonymous Credentials, its independence from propriatory software, and the support offered by its community, make Indy a suitable choice for our KYC2 framework.

IV. KYC2 Framework

KYC2 introduces an identity management ecosystem in which banking clients' identity attributes are stored on their mobile devices. The banks enrolled in the KYC2 ecosystem are able to issue Verifiable Credentials to the clients once their identity is successfully verified.

A. Architecture

The KYC2 architecture consists of three components. These components are:

1) *The banks A and B*
2) *A bank client Q*
3) *Validator nodes 1, 2, 3 and 4*

As shown in Fig. 4, the client and the banks, each have an agent and a wallet. Four validator nodes are responsible for the management of the ledger. For a system to tolerate at least one Byzantine fault there should be at least four validator nodes [31]. The banks and the client have access to the validator nodes through their respective agents.

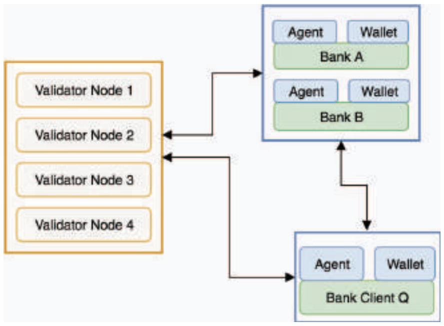

Fig. 4. KYC2 architecture composed of two banks, a client and four validator nodes.

B. *Simulation*

In our KYC2 simulation, there exist two banks *A* and *B* and a client *Q*. Client *Q* is interested in opening an account with a bank. In order to open a bank account, *Q* provides a bank, say *A*, with all the necessary identification documents. Bank *A* proceeds with verifying the identity of *Q*, opening an account for *Q* and providing *Q* with a Verifiable Credential signed by the bank's private key. This Verifiable Credential is held in *Q*'s possession and can be used by *Q* to interact with other banks. Next, client *Q* decides to open a new account with bank *B*. Instead of presenting the physical identification documents and reiterating the same process, *Q* uses his Verifiable Credential to prove his identity to bank *B*. Once bank *B* verifies *Q*'s claims, it proceeds with opening a new bank account for *Q*.

The following paragraphs describe the simulation in more detail. Fig. 5 illustrates the main interactions that take place in our simulation. The simulation is divided into two phases: *Bootstrap* (signified by the top area in Fig. 5) and *Interaction* (depicted by the bottom area in Fig. 5).

1) *Bootstrap Phase*

The bootstrap phase consist of registering the necessary data to the ledger so that the entities can locate, identify and establish trust among each other. This process begins by adding a trustee and the stewards to the ledger. In our simulation, there is a single trustee. The responsibility of this entity is to register the stewards, who are legal entities responsible for managing the validator nodes. The bootstrap process proceeds by the addition of banks *A* and *B* as trust anchors. The banks must have a trust anchor role to be able to register the DID of their new clients.

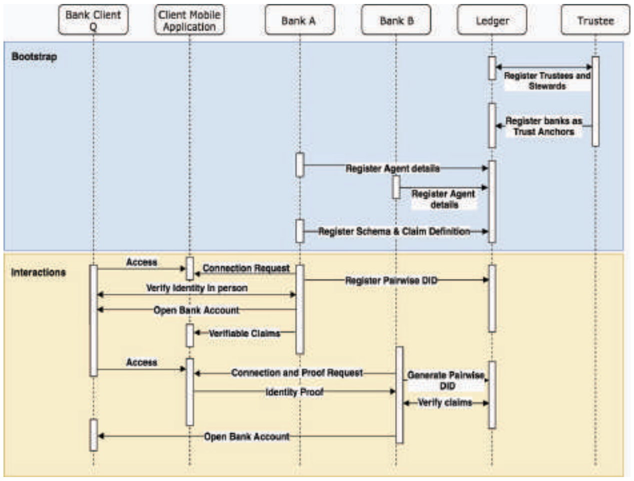

Fig. 5. Interaction among client, banks and the ledger during bootstraping and interaction phases

Next the agents' communication metadata, including their network address is broadcasted to the ledger, to make them accessible across the network. The last step of the bootstrap phase involves the registration of the KYC schema and claim definitions, used to issue the Verifiable Credentials. The schema consists of the following attributes: *first name, last name, phone number, email* and *address*. The claim definition consists of the public key and revocation details related to bank *A*. Since *A* is the only bank in our simulation that issues Verifiable Credentials, it is the only entity that registers a claim definition with the ledger.

2) *Interaction Phase*

The interaction phase begins by client *Q* accessing his KYC2 mobile app to accept a connection-request from bank *A*. This request contains the DID and the necessary connection information required to establish a secure link with the bank.

Upon the acceptance of the request by client *Q,* he generates a new *pairwise* DID and key pair specific to the communication between *A* and *Q*. Bank *A* is a trust anchor; thus it is able to append the new DID and key produced by *Q* to the ledger. Next, *Q* presents his identification documentation to *A*. This is typically a process done in person or through a method supported by the bank. Once *Q* is fully identified, the bank will open an account for *Q* and issue a Verifiable Credential through the established secure channel.

Note that client *Q* can also issue claims about himself, but these claims do not have the same level of trustworthiness and hence may not be trusted by the banks. Only the banks enrolled in the KYC2 framework are trusted as valid issuers.

At this point, client *Q* has a Verifiable Credential, which he can use to prove his verified identity with any other bank. Next, *Q* attempts to open an account with bank *B*. This begins by *Q* opening the KYC2 mobile app and accepting a connection-request from bank *B* as done before with bank *A*. The request contains the lists of verifiable identity attributes expected from client *Q* by bank *B*.

Since client *Q* has a Verifiable Credential in his possession, he responds to the connection-request with the required claims. Bank *B* then verifies the signatures on the claims and returns a success or failure message to *Q*. Upon successful identity verification, bank *B* may proceed with opening an account for *Q*. Optionally, the details of the usage consent given on the data offered by *Q* to the banks, are registered on the ledger for future reference and audit.

Going forward, client *Q* has the ability to register with any other member bank in a similar fashion. As *Q* interacts with additional banks, he is able to increase his reputation score. Higher score leads to *Q* having access to financial services that require higher identity assurance.

V. Discussion

This section provides a discussion around decentralized trust and current identity management systems. That is followed by a high-level evaluation of the KYC2 framework against a number of identity principles.

A. KYC2 and Decentralized Trust

Hyperledger Indy provides a decentralized public key infrastructure (DPKI). One of the major hurdles of traditional PKI models is their certificate revocation mechanism. The existing approaches to certificate revocation, such as Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP) are costly, slow, privacy infringing and require the availability of the certificate providers. In contrasts, KYC2 supports the storage of the revocation detail on the distributed ledger. This translates to diffused trust among many nodes, and verification of claim signatures without the presents of claim issuers.

The ability for every KYC2 entity with a trust anchor role to register DID and cryptographic keys onto the ledger enable the formation of an authenticated and encrypted channel among any pair of entities. While the same process can be implemented on PKI-based protocols such as TLS in the client-server web architecture, it is not common nor the default behavior of the clients to produce and use client-side digital certificates.

While KYC2 has a decentralized architecture, it is nevertheless dependent on trusted sources, such as the government and banks, to perform the initial client identity proofing. Moreover, the current iteration of the KYC2 framework relies on the existence of a supervisory entity with the *trustee* role, which possesses absolute control over the network. Future iterations of the KYC2 framework require the development of a *trust framework* agreement that describes the decentralized governance policies and processes recognized by all entities prior to participating in the KYC2 ecosystem.

Finally, the incentive for the deployment and operation of KYC2 nodes is an important topic. While it is enticing for the validator nodes to benefit from faster access to the ledger content, operation of a node is an expensive commitment.

B. Evaluation of KYC2 against Various Principles

In this section, we assess the KYC2 framework against a number of identity and privacy criteria. In order for the KYC2 framework to successfully implement an SSI model, it must address its ten principles. Table I maps each SSI principle with a specific requirement of the framework.

GDPR and other new legal frameworks have expedited the need to improve the status quo identity management models. Given that GDPR consists of 11 chapters, 99 articles, and 173 recitals, it is not a minor adaptation. Data controllers and data processors (jointly called data consumers in this section) are subject to GDPR regulations. In the context of KYC2, data consumers include banks, agents, wallets and ledger nodes. Table II lists the key principles within Article 5 of GDPR that are related to personal data, and their corresponding KYC2 requirements.

Privacy by Design (PbD) is a holistic approach of taking privacy into consideration at every step of the engineering process. To evaluate the KYC2 framework against PbD, Table III maps each of the seven foundational principles of PbD, as defined by Ann Cavoukian, to the appropriate KYC2 requirement.

TABLE I. KYC2 Framework and SSI Principles

SSI Principles	KYC2 requirements
Existence	The existence of a real world identity is verified during the initial interaction between the client and the bank.
Control	The client controls their identity data and keys on their own device, and have the capability to create, read, update or delete their identity attributes. They can also selectively disclose their identity attributes with the banks. The wallet must support secure storage of the identity claims and keys.
Access	All identity attributes and keys are accessible by the client through the wallet. The ledger should be available to the client (optionally through agents) to resolve DIDs to DDOs, and to obtain public keys and other public data.
Transparency	The KYC2 framework is an open source project operating on a public permissioned distributed ledger. No PII attributes, however, are written to the public ledger. The key revocation information, access consent receipts, and schemas, are all accessible through the ledger. There should be an onboarding process in place for any entity that wishes to have a write access to the ledger.
Persistence	The DID of each entity, as well as the agents metadata, consent receipts and public claims are recorded on the ledger, making them permanent, accessible and valid until explicitly invalidated by their owners.
Portability	Client identity data is transportable through the use of smartphones. Alternatively, the client can store their data on cloud-based solutions chosen by the client.
Interoperability	The KYC2 relies on open data structures and technologies such as DID and Verifiable Credentials to ensure full interoperability among parties and to support network growth. The framework also relies on standard communication protocols to maximize interoperability.
Consent	The client should explicitly define the terms and policies by which the identity attributes are shared with the banks. A receipt of client's consent should be registered on the ledger for future audit. The receipt should preserve client privacy by utilizing ZKP and data minimization techniques not susceptible to current or quantum-based cryptographic attacks.
Minimization	Only the identity attributes required by the banks in order to comply with their KYC policies and regulations are required from the client. Moreover, the client can take advantage of ZKP techniques to minimize the data being shared with banks.
Protection	The policies and responsibilities associated with every entity of the framework should be clearly stated within the KYC2 trust framework. The validator nodes should be distributed among multiple geographic jurisdictions and no node should be able to censor a valid transaction or operation.

TABLE II. KYC2 FRAMEWORK AND GDPR PRINCIPLES

GDPR Principles	KYC2 requirements
Data to be processed in a lawful, fair and transparent way	Every KYC2 data consumer is required to clearly articulate to the clients their data processing practices, and obtain client's permission before processing client PII. The client and the data consumers involved in an interaction should produce a privacy-preserving consent receipt referencing their usage agreements. The consent receipts can also be used to describe the interactions among data consumers. The format of the consents placed on the ledger should be clearly defined so that permitted entities are able to interpret them.
Purpose limitation	The data consumers should only obtain and process the identity attributes that are required to operate. For example, the agents may only require the public key credentials and DID of the clients in order to establish a secure communication channel with other agents. The banks on the other hand, require the identity data that satisfy their KYC and AML compliance requirements.
Data minimization	Only the relevant and necessary data is shared with the data consumers and only for the required duration of time. The clients should have the ability to perform granular selection of identity attributes and possess the capability to utilize data minimization techniques such as ZKP. The claims issued to the client by the issuing banks should be atomic, making it possible for the client to select only the required minimum attributes to disclose with the data consumers.
Accuracy	The clients should have the ability to update their basic identification records such as cryptographic keys and DIDs stored on the ledger and on the banks. The banks issuing the claims to the clients should ensure accuracy and integrity of their identity verification and claim issuance processes.
Storage limitation	While the banks have regulatory obligations to maintain their clients data, the data should be stored only for the duration of time mandated by the regulation and related policies. Article 17's 'Right to be Forgotten' provides the clients the right to request their data to be removed from the data consumers. To satisfy this article, there should be no PII stored on the ledger. However other data points such as DIDs, DDOs and consent receipts added to the ledger cannot be removed, although their usage with the assigned identity can be blocked. Any identity data stored by the banks; agents or wallets should be removed per client's request in a timely fashion. If the identity data cannot be promptly removed, its reason should be clearly conveyed to the client.
Integrity and confidentiality	The data consumers should enforce proper security posture by implementing appropriate security controls including controls related to secure storage and transmission of client's personal data. For example, by relying on the public keys and DIDs found on the ledger, the client and banks have the ability to establish a bidirectional secure connection. Moreover, great consideration should be given to wallet and agent security, due to their critical role in holding and processing client identity information.
Accountability	Any data consumer of the framework should implement appropriate technical and organizational measures, such as implementation of accurate record keeping, and incident management processes. Upon the discovery of a data breach, the target entity must promptly notify the regulatory bodies, and the clients involved. Moreover each data consuming organization should have a data protection officer responsible for ensuring that all organizational and technical processes and policies comply with the current and future GDPR regulations. The KYC2 ecosystem involves a complex architecture that includes various types of entities. These entitles are potentially located in different jurisdictions. Consequently this may lead to confusion, and challenges, in determining the applicable regulations for each entity, and holding the right entities accountable should a security breach occur.

TABLE III. KYC2 FRAMEWORK AND PRIVACY BY DESIGN PRINCIPLES

PbD Principles	KYC2 requirements
Proactive not Reactive, Preventative not Remedial	There should be strong commitment by all entities within the KYC2 ecosystem to establish and apply adequate privacy and security controls, to proactively protect against privacy infringements. These commitments should be reflected in the KYC2 trust framework to foster a culture of continuous safeguarding of users' privacy.
Privacy as the Default Setting	By default only the minimum amount of required data should be obtained from the clients. By default, privacy-preserving processes should be adopted to protect against unnecessary exposure of client identity data. These processes include the use of ZKP, anonymization methods, and strong cryptography. To maximize anonymity on the network every connection with a new bank should lead to the creation of a new client DID. The banks should have the option of using a pairwise DID specific to a client, or a publicly know persistent DID.
Privacy Embedded into Design	The Indy architecture has privacy controls baked in its core. Examples include the support for selective disclosure of claims and ZKP.
Full Functionality, Positive-Sum, not Zero-Sum	The privacy measures should be complimentary to the security of the framework and not contrary. For example the ledger should only record public keys, schemas, claim definitions and revocation material, and allow verification of signatures and claims, while preventing the claim issuer from being involved in signature validation queries. The DIDs and the signatures registered on the ledger should not lead to exposure of identity through data correlation. While SSI allows clients to control their identity data on their devices, proper security measures must be implemented to protect the clients identity data and keys.
End-to-End Security	From the point the identity claims are issued by a bank to a client, and sent as proof to another bank by the client, to the time those claims are revoked, proper security measurements must in place to protect the confidentiality, privacy and integrity of the data. This rule also applies to the data stored on the ledger, on client wallets and agents, and on banks servers, during transmission or at rest.
Visibility and Transparency — Keep it Open	The framework is open source and relies on open standards. Before joining the framework every entity must agree to the KYC2 trust framework, and every entity should be verified before they can be trusted as a data consumer. The clients have the liberty to choose their preferred agents and wallet implementations. The entities should be open to independent verification and audit.
Respect for User Privacy — Keep it User-Centric	KYC2 implements the SSI principles, through which the clients are in complete control of their identity data. The interface by which the clients interact with the KYC2 framework, including the mobile applications, wallets, and agents, should be user-centric, intuitive, privacy preserving and easy to use. The mobile app should provide clients with prompt notifications about any type of access to their data.

VI. Conclusion and Future Work

KYC is a resource intensive and complex process. This paper lays the groundwork for a digital client onboarding system based on Hyperledger Indy, and principles of Self-Sovereign Identity. Through this work, the paper aims to address a number of challenges. First, the current identity management models do not always consider the user as a principal stakeholder. Secondly, users' digital identity data are scattered among multiple identity providers, which oftentimes are not on par with industry's best security practices. This is manifested in the growing number of cyber attack incidents. Furthermore, the introduction of GDPR and other new regulations impose new angles on how we should view digital identity rights. This paper takes a step towards empowering online users in the on-going race for their digital data, and providing a new digital approach to the KYC process.

Our future research work can take multiple directions. First, we are interested in extending our research to outline the key design decisions for secure and privacy preserving digital wallets and agents. Such research must consider critical usability issues such as practical key recovery mechanism.

Secondly, we plan to develop the relevant trust framework documents, by which entities can join the KYC2 ecosystem. This trust framework should cover the policies and processes applicable to all entities of the ecosystem.

References

[1] "Equifax breach: total number of Canadians impacted by cyber attack passes 19,000 - National | Globalnews.ca." [Online]. Available: https://globalnews.ca/news/3886756/equifax-canada-data-breach/. [Accessed: 12-Apr-2018].

[2] "EU GDPR," *EU GDPR Portal*. [Online]. Available: http://eugdpr.org/eugdpr.org.html. [Accessed: 12-Apr-2018].

[3] A. Demirgüç-Kunt, L. F. Klapper, D. Singer, and P. Van Oudheusden, "The global findex database 2014: Measuring financial inclusion around the world," 2015.

[4] S. Burhouse and Y. Osaki, "FDIC National survey of unbanked and underbanked households," *Fed. Depos. Insur. Corp. Recuperado En Marzo De*, 2014.

[5] "UFA2020 Overview: Universal Financial Access by 2020," *World Bank*. [Online]. Available: http://www.worldbank.org/en/topic/financialinclusion/brief/achieving-universal-financial-access-by-2020. [Accessed: 12-Apr-2018].

[6] "How to conduct proper customer due diligence (CDD)," *AML-CFT*, 19-Mar-2017. [Online]. Available: https://aml-cft.net/conduct-proper-customer-due-diligence-cdd/. [Accessed: 12-Apr-2018].

[7] "Thomson Reuters 2016 Know Your Customer Surveys Reveal Escalating Costs and Complexity." [Online]. Available: https://www.thomsonreuters.com/en/press-releases/2016/may/thomson-reuters-2016-know-your-customer-surveys.html. [Accessed: 12-Apr-2018].

[8] C. Allen, "The Path to Self-Sovereign Identity," *Life With Alacrity*. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html.

[9] "Hyperledger Indy," *Hyperledger*. [Online]. Available: https://www.hyperledger.org/projects/hyperledger-indy. [Accessed: 12-Apr-2018].

[10] A. Cavoukian, "Privacy by design," *Take Chall. Inf. Priv. Comm. Ont. Can.*, 2009.

[11] "Verifiable Claims Data Model and Representations." [Online]. Available: https://www.w3.org/TR/verifiable-claims-data-model/. [Accessed: 12-Apr-2018].

[12] "Decentralized Identifiers (DIDs) v0.9." [Online]. Available: https://w3c-ccg.github.io/did-spec/. [Accessed: 12-Apr-2018].

[13] A. Jøsang and S. Pope, "User centric identity management," in *AusCERT Asia Pacific Information Technology Security Conference*, 2005, p. 77.

[14] "Common Terminological Framework for Interoperable Electronic Identity Management," *modinis IDM*. [Online]. Available: https://www.cosic.esat.kuleuven.be/modinis-idm/twiki/bin/view.cgi/Main/GlossaryDoc#4_13_Digital_Identity. [Accessed: 12-Apr-2018].

[15] M. Dabrowski and P. Pacyna, "Generic and complete three-level identity management model," in *Emerging Security Information, Systems and Technologies, 2008. SECURWARE'08. Second International Conference on*, 2008, pp. 232–237.

[16] D. W. Chadwick, "Federated identity management," in *Foundations of security analysis and design V*, Springer, 2009, pp. 96–120.

[17] K. Cameron, "The laws of identity, May 2005," *Microsoft Corp.*, 2005.

[18] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of cryptocurrency systems," in *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*, 2016, pp. 745–752.

[19] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 3, pp. 2084–2123, 2016.

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[21] "Ethereum Project." [Online]. Available: https://www.ethereum.org/. [Accessed: 12-Apr-2018].

[22] P. Marc and others, "Blockchain Technology: Principles and Applications," 2016.

[23] "Home - Hyperledger." [Online]. Available: https://www.hyperledger.org/. [Accessed: 12-Apr-2018].

[24] "Hyperledger Fabric," *Hyperledger*. [Online]. Available: https://www.hyperledger.org/projects/fabric. [Accessed: 12-Apr-2018].

[25] "The Technical Foundations of Sovrin, A White Paper from the Sovrin Foundation." Sovrin, 29-2016.

[26] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[27] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *J. Cryptol.*, vol. 1, no. 2, pp. 77–94, 1988.

[28] "U-Prove," *Microsoft Research*. [Online]. Available: https://www.microsoft.com/en-us/research/project/u-prove/. [Accessed: 16-Apr-2018].

[29] "IBM Research - Zurich, Identity Mixer." [Online]. Available: https://www.zurich.ibm.com/identity_mixer/. [Accessed: 12-Apr-2018].

[30] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Annual International Cryptology Conference*, 2002, pp. 61–76.

[31] P.-L. Aublin, S. B. Mokhtar, and V. Quéma, "Rbft: Redundant byzantine fault tolerance," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, 2013, pp. 297–306.

[32] M. Al-Bassam, "SCPKI: a smart contract-based PKI and identity system," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, 2017, pp. 35–40.

[33] J. G. Faísca and J. Q. Rogado, "Decentralized semantic identity," in *Proceedings of the 12th International Conference on Semantic Systems*, 2016, pp. 177–180.

[34] "Namecoin." [Online]. Available: https://namecoin.org/. [Accessed: 12-Apr-2018].

[35] "WebID Specifications." [Online]. Available: https://www.w3.org/2005/Incubator/webid/spec/. [Accessed: 12-Apr-2018].

[36] P. Dunphy and F. A. Petitcolas, "A First Look at Identity Management Schemes on the Blockchain," *ArXiv Prepr. ArXiv180103294*, 2018.

[37] Blockcerts, "Blockchain Credentials," *Blockcerts*. [Online]. Available: http://blockcerts.org/. [Accessed: 12-Apr-2018].

[38] "Secure Enterprise Identity Authentication | ShoCard." [Online]. Available: https://shocard.com/. [Accessed: 12-Apr-2018].

[39] "Open Badges Homepage." [Online]. Available: https://openbadges.org/. [Accessed: 12-Apr-2018].

[40] "Civic Identity Verification | Secure & Protect Identities," *Civic*. [Online]. Available: https://www.civic.com/. [Accessed: 12-Apr-2018].

[41] "uPort.me." [Online]. Available: https://www.uport.me/. [Accessed: 12-Apr-2018].