DESIGNING HYPERLEDGER INDY BLOCKCHAIN TO ELECTRONICALLY CERTIFY STUDENTS' ACADEMIC CREDENTIALS

Tikajit Rai[1] and Poonsri Vate-U-Lan[2]

Email: tikajit@ieee.org, poonsri.vate@gmail.com

Received: October 4, 2021; Accepted: November 23, 2022; Published: December 1, 2021

Abstract

Purpose of the research paper is to investigate available blockchain technologies and identify one that can be adapted as a solution to electronically verify academic credentials of a person. The general objective is to investigate types of open-sources and Decentralized Identifiers (DIDs) compatible cloud-based blockchain platforms in the market with specific research objectives to identify security features and consensus mechanism, and systems architecture of the solution, respectively. The methodology used for the research is qualitative in nature to reviewing literatures. Hyperledger Indy is found to be a good fit for the solution because it is DIDs compatible, and its inherent security features and consensus mechanism are highly secure. A high-level use case of the solution and systems architecture is created based on the security features that utilizes Zero-Knowledge Proof (ZKP) scheme and Redundant Byzantine Fault Tolerant (RBFT) consensus mechanism. This study itself is and evidence that the free resources made available by the Hyperledger Indy project enables organizations to design and develop prototypes per their requirements. Benefits of the research paper are further validating the potential use of DIDs compatible blockchain framework to electronically certify verifiable credentials, and realization of shifting the ownership of credentials from third-party entities to the users themselves through use of Self-sovereignty Identity (SSI) feature.

Keywords: Blockchain, Academic Credential, Distributed Ledger, Hyperledger

1. INTRODUCTION

Blockchain technology-based solutions have the potential of reshaping how business transactions are conducted, including identity management by enabling trust, providing transparency and traceability, and eliminating third party dependency (Costello, & Rimol, 2019). Identity management can be integrated into blockchain for a number of real-life solutions, ranging from verifying an individual's age to employment record; from purchasing airline tickets to certifying academic credentials. The context of the study is on how a blockchain technology can be adapted as a solution to electronically certify (e-Certify) a person's academic credentials.

[1] M.S. ICT, Computer Communications and Network, Graduate School Business and Advanced Technology Management, Assumption University of Thailand. tikajit@ieee.org

[2] Ed.D., Assistant Professor at Assumption University of Thailand. poonsrivlt@au.edu https://orcid.org/0000-0002-4200-0469

The general objective of the study is to investigate types of open-source and Decentralized Identifier (DID) compatible blockchain platforms in the market. The specific objectives are to identify security features and systems architecture, respectively, of the blockchain based solution that replaces the traditional process in which a person is required to mail sealed hardcopy of academic credentials to the entity that seeks evidence of the fact. Student applying for further studies in new academic institute or employment are real-life scenarios in which the applicant requests the certificate issuer to send the sealed hardcopy of credentials directly to the concerned institute or the employer.

Scope of the study is limited to reviewing available literatures to investigate blockchain technologies in the market, their security features, consensus mechanism, and then to develop a use case of the solution and its high-level architecture. Hyperledger Indy is of a particular interest in this study, because Hyperledger Indy is a framework specifically created for managing DIDs, which are a type of a digital identifier of a subject, including, of a person, which is verifiable and are decentralized over its distributed ledger (W3C, 2021a).

Significance of the study are two-fold. First is to further validate the potential use of DIDs compatible blockchain framework to electronically certify verifiable credentials. If the solution can electronically verify academic credentials, it can also do the same for verifying person's age when needed, without disclosing any information at all. Second and last is to showcase realization of shifting the ownership of credentials from third-party entities to the users themselves through use of self-sovereignty identity feature.

1.1 Purpose of the study

Purpose of the paper is to investigate available blockchain technologies and identify one that can be adapted to e-Certify academic credentials of a person and can replace the inefficient and inconvenient traditional process of mailing certified hardcopy of academic transcripts to an academic institute or an employer. The general objective of the study is to investigate types of open-sources and DID compatible blockchain platforms in the market. Specific objectives are deduced to following:

i) To identify security features and consensus mechanisms integrated in Hyperledger Indy.
ii) To examine cloud-based systems architecture of blockchain applications in the market.

2. LITERATURE REVIEW

During the literature review, following key terms are investigated to have a thorough and complete understanding so the study objectives are achieved with expected depth and adequacy.

2.1 Blockchain: Private, Public, and Hybrid

Evidently, blockchains can be implemented in various ways to solve business problems. They can either be implemented either as private, or public blockchain. An example of a former case blockchain based solution for an organization and network participants are the employees or certain pre-identified stakeholders. They, as nodes, can both participate and control the decentralized network as well as being able to offer permissions to various nodes.

In contrast, the latter, as the name already suggests, can be a public solution to which wider stakeholders interact. Since such blockchains are open to the public, all records are visible to every participant, and the network is always open to all new nodes. Table 1 provides specific characteristics of private and public blockchain.

Table 1: Private and Public Blockchains

	Private	Public
Network	Centralized	Decentralized
Security	Approved Participants	Open Network
Access	Permissioned	Permission-less
Identity	Known Identities	Anonymous/ Pseudonyms
Speed	Faster	Slower

In recent years, however, hybrid blockchains have evolved and fall under public blockchains category. A consortium blockchain is very similar to a public blockchain. The difference however is that only a group of permissioned nodes are allowed to participate in the consensus process of the blockchain. Hyperledger Indy is such a blockchain. To participate in Hyperledger Indy, the network participant must have permission to become an active node, albeit the blockchain being public (Blockchain technology overview, 2020). Table 2 provides some of the popular cloud-based blockchain brands that compete in the industry. Hyperledger Indy and Ripple are the ones that are both consortium types of blockchain technologies.

Table 2: Blockchain: Types of Access and Validation (Hyperledger Indy SDK, 2018)

Blockchain Types	**Permissionless**	**Permissioned**
Public	Bitcoin Ethereum	Hyperledger Indy Ripple
Private	Hollochain LTO Network	Hyperledger Fabric R3 Corda

2.2 Blockchain: The Genesis and Data Block

Irrespective of the types blockchains, as shown in Table 2, it retains the same fundamental components, which are Genesis Block (GB); Data Block (DB); and Blockchain (Pourmajidi et al., 2020). Primary purpose of the GB is to indicate the beginning of a new chain (Azaria et al., 2016). To that effect, GB is always the first block of any blockchain and has a predefined set of characteristics. The *index* and *previous hash* are, as there are no preceding blocks, both set to zero.

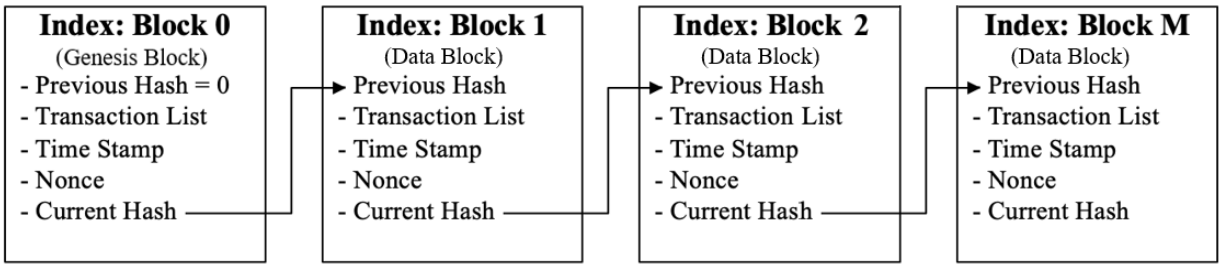

Figure 1: Blocks of Blockchain

As shown in Figure 1, a DB, or simply a block, contains the following variables: *index*, *timestamp*, *data (or transaction list)*, *current hash*, and *previous hash*. *Index* is a unique sequential ID for each block to identify each successive block. The *timestamp* stores the time when the block is created. The data contains information which blockchain makes it immutable, rendering it the most important element of a DB.

Essentially, as Figure 1 depicts, a blockchain is chain of blocks that are linked together through binding the hash – the process also known as *hash binding* – of preceding blocks. Also, in Figure 1, the last block "M" illustrates the arbitrariness of number of blocks that are hashed and liked after the GB to the blockchain at a given time due to subsequent completed transactions. The *nonce* is an arbitrary random number that is used to generate a specific *current hash*. To achieve *hash binding*, each block includes a *previous hash* element. The *previous hash* is the exact duplicate of the *current hash* of the previous block (Pourmajidi et al., 2020).

2.3 Blockchain Nodes

Nodes in a blockchain are computers, or devices, that participates in its distributed network as distributed ledgers. Each node maintains the accuracy and security of the information by retaining a copy of the complete set of ledgers of all the past transactions. When a new block is successfully created by an authorized node, the new block then is broadcasted to the entire blockchain network, allowing all the participating nodes to update their respective ledger (Tschorsch & Scheuermann, 2016).

2.4 Consensus Mechanism: Redundant Byzantine Fault Tolerance (RBFT)

RBFT is a consensus mechanism that is modified version of Byzantine Fault Tolerance (BFT). The replication protocols that it uses tolerates arbitrary faults within a fraction of the replicas, or instances, of the protocol themselves (Aublin et al., 2013). In it, the redundancy is achieved through running multiple instances of the same BFT protocol, each with a *primary replica*, simultaneously on multiple nodes. All the instances then order the requests for execution, but only the node that has what is called a *master instance* gets to execute the instance exclusively. During this time, rest of the instances is closely monitored to check if the instance being executed yields expected performance, or not – for example, time taken to complete the execution. If the outcome is unfavorable, the instance is considered malicious and discarded. Then another *primary replica* of next node in random order is executed. This new approach increases the performance when compared to other existing most robust protocols (Aublin et al., 2013).

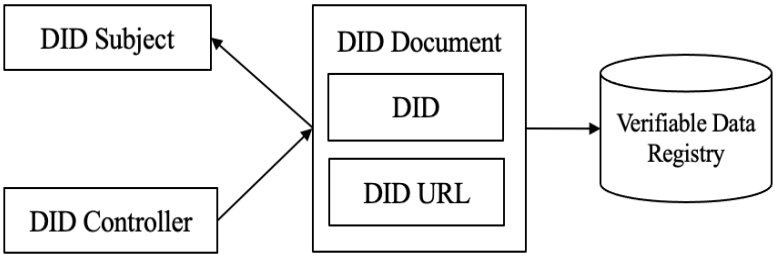

Figure 2: High-level View of Interacting DID Entities

2.5 DID and its Components

A DID is a new type of identifier that enables verifiable and decentralized digital identity of a subject, including that of a person (W3C, 2021b), over a given distributed ledger. This is also a globally unique persistent identifier that does not require a centralized and third-party registration authority because it is generated and/or registered cryptographically (W3C, 2021b). As shown in Figure 2, a DID has other supporting components to make the application practically feasible. DID document stores necessary data and information of a subject and mechanism as to how they can be updated. Whereas Verifiable Data Registry is any system that supports recording DID and returning data necessary to produce DID documents, for example, distributed ledgers, decentralized file systems, databases of any kind, peer-to-peer networks, and other forms of trusted data storage (W3C, 2021b).

2.6 Digital Ledger Technology (DLT)

DLT is a form of digital database that is updated and shared a copy to every qualifying node of the blockchain network. The nodes therefore hold the ledger collectively but creates and updates it independently. In contrast to centralized location to hold official copy of the ledger, all network participants would have their own (Norman, 2017).

2.7 Hyperledger Indy

Hyperledger Indy is one of the frameworks produced by the open-source community and Hyperledger project that is supported by the Linux Foundation in developing a suite of stable frameworks for enterprise-grade blockchain deployments (Hyperledger, 2020).

2.8 Hyperledger Indy Nodes

Hyperledger Indy Nodes are distributed servers. In addition to nodes, there are also Full and Master nodes, as well. Each node houses ledger to store records and identifies specific transaction in the nodes. Full node, on the other hand, is a client operating on the network which maintains full copy of the blockchain. Master node is governance around decentralization (Hyperledger, 2020).

2.9 Hyperledger Indy Plenum

Hyperledger Indy Plenum is the network system that maintains a replicated ordered log of transactions called the ledger. Participants of the network which maintain this log are called the nodes. The nodes run RBFT consensus protocol (described further in next section) to agree

on the order of transactions and maintain several ledgers, each for a distinct purpose (Hyperledger, 2020).

2.10 Identify Correlation Resistance

Any system that does not allow two similar IDs in the DLT of a single user (W3C, 2021b). In other words, in blockchain implementation, there is always one and only one DID of a single user. A user may create multiple DIDs, however (W3C, 2021b).

2.11 Public-Key Cryptography

Public-key cryptography uses public and private keys to encrypt and decrypt electronic messages, respectively. The public key may be shared widely in the public domain, and in contrast, the private key remains secretive to the owner. In contrast to symmetric key calculation that uses a single key to scramble and unscramble data, each key has a distinct feature, consequently calculating the private key from the public key is computationally infeasible (Jegadeesan et al., 2021).

2.12 Security and Privacy in Blockchain

In the context of online blockchain transactions, specific security and privacy related requirements need to be considered to prevent undesired exploitation of possible vulnerabilities. Based on Zhang et al. (2019), there are seven broad categories of security and privacy one need to consider:

i) Consistency of the ledger across blockchain nodes always needs to be maintained.
ii) Integrity of transactions must be always guaranteed.
iii) Availability of system and data must be guaranteed to all network nodes governed by its established rules and policies.
iv) Duplication of the same transaction must always be prevented with.
v) Confidentiality of all transactions must be guaranteed.
vi) Identities of users behind nodes must always remain anonymous.
vii) Ability to link the transactions of the node also must be guaranteed.

2.13 Self-sovereignty Identity (SSI) and its Relevance to Decentralized Identity System

SSI is a feature of an identity system, whereby individual users maintain control over when, to whom, and how they assert their identity. It also includes a use case in which the users are given greater control over how their identity (ID) that are issued by a third and formal party (Houtan et al., 2020).

2.14 Software-as-a-Service (SaaS)

SaaS is a centralized software service delivery model in which subscription by a user is licensed. As part of the three delivery models proposed by National Institute of Standards and technology (NIST) (Mell & Granc, 2011) – other two being, Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) – it provides services to users without installing any application locally. In other words, services are hosted in cloud-based architecture and using thin clients, such as web browser alone, users can access SaaS services that include, but not limited to, management information systems, enterprise resource planning, office applications, collaboration tools, and service desk management. Such cloud-based service is leveraged by

service providers to integrate blockchain services in their offerings, which are termed as Blockchain-as-a-Service (BaaS).

2.15 Verifiable Credential (VC)

VC is the W3C data model specifications to express credentials on the Web and establishes standards to make it cryptographically secure, privacy respecting, and machine verifiable (W3C, 2019). A VC can represent any information that a physical medium can, but the former is more tamper-evident and trustworthy with the use of technologies, such as electronic certification or digital signature, blockchain and DLT.

2.16 Verifying VC Using Blockchain Technology

Figure 3 below illustrates high-level use case of blockchain as to how a VC is issued and verified when an individual purchases an airline ticket. The *Issuer* is the third-party and formal entity that issues the credential. A good example is the U.S. Department of Motor and Vehicles (DMV) in the U.S. which issues a certain category of driver's license to a qualified vehicle driver. The *Issuer* executes two tasks in the process. First, it asserts the claim about the driver's name, age, and driving credentials by writing VC in the blockchain enabled VDR (see 1.a arrow in Figure 3). Second and last, the DMV transmits the VC to the *Holder*, or the driver (see 1.b arrow in Figure 3).

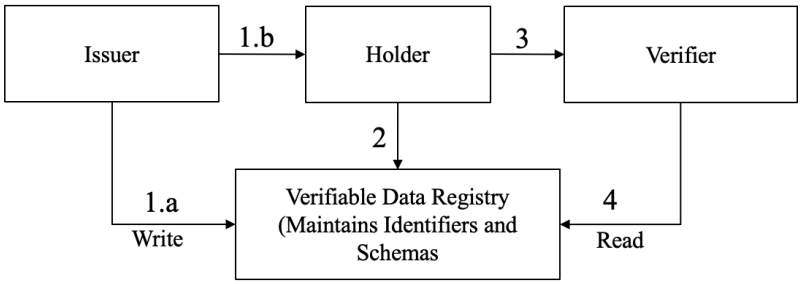

Figure 3: High-level View of Use Case

The driver then acquires, stores, and creates verifiable presentations of the VC by creating a profile in the blockchain network. When the *Holder* buys an airline ticket, he or she presents the VC to the *Verifier*, the travel agency in this case (see arrow 3 in Figure 3). The *Verifier* then verifies the VC of the driver, particularly the name and the age (driving credentials are irrelevant in this transaction), against that of the blockchain enabled VDR. If the verification process checks out right, the e-Certification process is completed. In this manner, VCs are written in the blockchain and issued to the *Holder* by the *Issuer*, which then the *Verifier* verifies the VCs presented by the *Holder* against what is in the blockchain. This is attainable only if all three entities are members of the same blockchain business network.

2.17 Zero-Knowledge Proofs (ZKP)

Zero-Knowledge Proof (ZKP) is a complex scheme that uses encryption techniques to enable a prover to certify the truthfulness of a statement to a verifier without disclosing any additional specifics than the statement itself (Goldreich & Oren, 1994).

3. RESEARCH METHODOLOGY

The methodology used for the research is qualitative in nature as it entails reviewing the literatures to: 1) investigate blockchain technology that is DID compatible, and its inherent security features and consensus mechanisms; 2) examine systems architecture of blockchain applications in the market. The data and information are then analyzed in the context of problems stated in the study and then answers to the research questions are synthesized.

Figure 4 below shows high-level waterfall-like research stages. Firstly, extensive literature reviews were conducted to identify relevant blockchain types, and their respective security features and consensus mechanisms. Secondly, the data and information from literature reviews were analyzed, compared, and contrasted to understand in full the theories, concepts, and technology. Finally, frameworks of both systems architecture and the most effective approach to integrate security features and consensus mechanism to e-Certify academic credentials are synthesized and described in the final section of the paper.

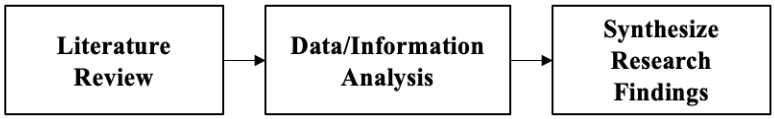

Figure 4: Research Stages

4. RESULT AND DISCUSSION

Hyperledger Indy's inherent security features and consensus mechanism are found to be highly secure. Based on the literature review, a use case and high-level systems architecture of the solution are created. Following two sections discuss the research findings in detail.

4.1 Security Features of Hyperledger Indy

One practical approach in addressing the problem of the research is using DIDs and associating academic credentials data of the students as VC in the blockchain. As stated in the introduction, Hyperledger Indy is especially created for DIDs. Therefore, it can create verifiable and decentralized digital identity of students to which their respective academic qualification data overtime are stored in its blockchain. By doing this, first: i) both provenance and immutability are achieved eliminating any possibility of fraudulent alteration of data; ii) students' privacy is achieved through use of ZKP scheme and RBFT consensus mechanism; iii) ownership of the verifiable credentials resides on the students due to SSI feature; and iv) the academic credentials of any students who are members in the network can be e-certified on a real-time basis without much inconvenience. Other specific features which make the Hyperledger Indy most plausible candidate to e-Certify online students' academic credentials are use of identify correlation resistance and VCs.

Consensus mechanism in Hyperledger Indy is called Plenum Byzantine Fault Tolerance (PBFT), or Plenum, which is a protocol inspired by RBFT (Hyperledger, 2018). The replica instances are resultantly called plenums. As in the case of RBFT, each node can host one *primary (leader) instance* (see Figure 7) and the *master instances* are used to update the ledger. The master's performance in terms of both throughput and latency is periodically compared to the average performance of other instances. If the *master instance* is found to be non-performing, then a different instance to the role of master is appointed. To detect a faulty node, represented by 'f', PBFT needs at least 3f+1 nodes to handle faulty nodes (Zhang & Lee, 2020). Figure 7 shows a network of 4 nodes, with each node with 2 BFT-like plenums running: one

master and one backup. In addition, the Figure 7 network can would have1 faulty node, based on 3f+1 = number of total nodes.

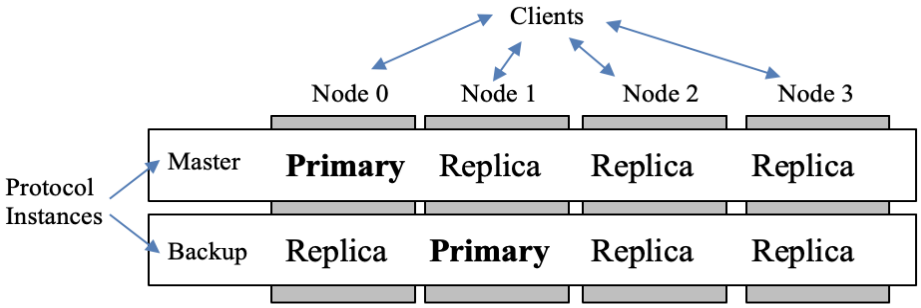

Figure 7: RBFT Overview

Figure 8 on the other hand shows a different view of PBFT. Here, the client is sending a *request* to nodes for consensus. As sending to f+1 nodes suffice, the request does not have to reach the entire nodes (Zhang & Lee, 2020). After receiving the client *request*, the nodes disseminate through a process called *propagate* in which rest of the nodes are made aware of the fact. In the following steps, each *primary instance* creates a proposal from the received *requests* called a PRE-PREPARE and sends it to all other nodes. If the nodes accept the primary's proposal, they send an acknowledgement to the proposal by a message called *prepare*. Once a node gets a *pre-prepare* proposal and 2f *prepare* messages, the process reaches a state in which sufficient information to accept the proposal is achieved, then sends a *commit* message. Once a node gets 2f+1 *commit* messages, the batch of *requests* can be ordered and added to the ledger, because needed minimum number of nodes have agreed that majority of the nodes have accepted the proposal (Zhang & Lee, 2020).

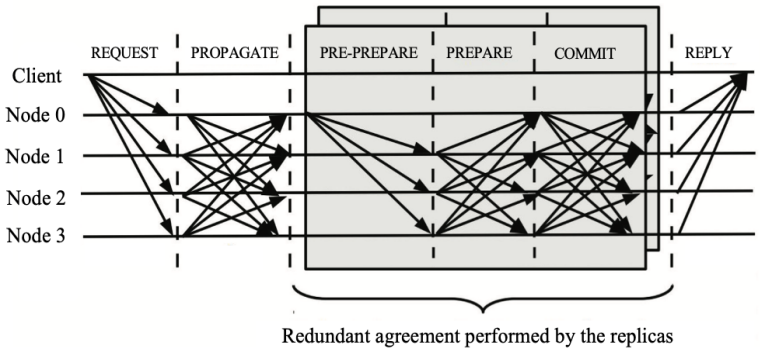

Figure 8: RBFT Protocol Commit Process

Further, Hyperledger Indy uses PBFT to handle ordering and validation, which results in a single ledger containing both ordered and validated transactions. This is unlike many blockchain networks that use a BFT protocol only for ordering. These networks leave domain-specific validation to happen after requests are ordered (Zhang & Lee, 2020).

4.2 Hyperledger Indy Architecture

Prior to discussing the Hyperledger Indy blockchain's system architecture, a use case of leveraging the technology to e-Certify academic credentials was first mapped out. The use case of Hyperledger Indy implementation is found straightforward in which initially a user provides his or her name as an identifier. This identifier is then converted into a unique key known as DID of the user. The key has an associated value with it that is called the DID descriptor object (DDO), and together they form a complete DID record of the user. Users in the blockchain network interact with each other using the public and private keys of the respective DID records.

Figure 9 illustrates the use case of a recent graduate entering the job market. As one of the members of the blockchain business network, the *Issuer* is the academic institute student graduated from. The institution executes two tasks. First, it asserts the claim about the student's academic credentials by writing VC as per its identifier and schema in the blockchain enabled VDR, or the Registry (see Figure 9). Second and last, the academic institution transmits the VC to the *Holder*, or the graduate in this case.

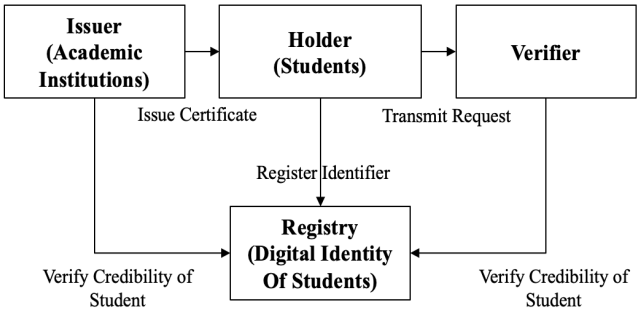

Figure 9: High-level Use Case of Blockchain Prototype

The *Holder* creates a profile with his or her name as an identifier to create a DID record that has verifiable presentations of the VC: the academic credentials. The graduate presents the VC to the *Verifier*, or the employer in this case (see Figure 9). Through the use of ZKP scheme, the *Holder* does not even need to disclose any information to the *Verifier* like in the case of showing a physical academic transcript. As long as the verification process checks out right that the *Holder* has in fact all the credentials that meet the employer's requirements, the e-Certification process is completed.

The graduate may acquire additional VCs, professional development certificate, for example, and associate them to the same DID for which identifiers and schema can be updated in the blockchain enabled Registry (see Figure 9). As all three entities are members of the same blockchain business network, and VCs can be verified rather quickly and securely, the improved efficiency and convenience is significant compared to the traditional process in which certified hardcopy of academic transcript is mailed to the employer. Additionally, the e-Certification process does not need to disclose any more information than needed to the *Verifier* because of ZKP scheme implemented in the blockchain.

Layer 4	Application
Layer 3	Governance Frameworks
Layer 2	Secure Communications
Layer 1	Distributed Ledger Access
Layer 0	DTL

Figure 10: Hyperledger Indy Architecture Overview

With the use case explained in preceding section, considered Hyperledger Indy's High-level system architecture is shown in Figure 10. As the figure depicts, the layered architecture has modular and interoperable layers with an emphasis on highly secure solution. The components of the architecture are further discussed below starting from Layer 0 to Layer 4:

i) DTL Layer is responsible for managing nodes, permissions, and PBFT protocol.
ii) Distributed Ledger Access Layer is responsible for providing the main application interface on top of the DTL. The core modules include the ledger, DIDs, and VCs. The layer also ensures that transactions are signed by the user's right DID prior to providing access to the ledger.
iii) Secure Communications Layer is responsible for providing the capability to establish secure communications with the users and exchange secure messages. Further, it has two core message protocols that enables the layer to exchange VCs and proofs in a secure fashion.
iv) Governance Frameworks Layer is responsible for managing scope of credentials, and establishing rules, and policies that all users adhere to.
v) Application layer provides frameworks that allow developer to abstract business-specific implementations using Indy-based functionality.

5. CONCLUSION

Based on the study, it is evident that the available open-source Hyperledger Indy and its codebase as well as documentation are adequate to design DID-based public permissioned blockchain solutions to e-Certify students' academic credentials. This holds merit because the PBFT is used as a consensus mechanism and in-built public key cryptography in the solution are both robust and tested in the industry. This argument is further supported by research that are already completed in this specific domain. Some of the research works are referenced in this paper.

One interesting and contrasting aspect of using Hyperledger Indy is that the platform does not use smart contracts. It is because the idea behind the DID-enabled blockchain framework is to enable users to own, control, and share the data in a way that preserves their privacy in contrast to storing the data in the distributed ledger itself and use smart contracts to access it. This further reduces the risk of personal data being compromised albeit blockchain is by design secure.

Another important aspect of the solution is the SSI which unprecedentedly shifts the ownership of the data and information related to academic credentials to the students from the traditional manner of academic institutions being the sole owner. This resultantly creates flexibility and overall efficiency as to how students' academic credentials can be verified with disclosing only relatively limited data compared to sharing the entire transcripts in the traditional process. In fact, privacy of the *Holder* can be attained 100% from the *Verifier* with the right use case and solution architecture in place. In the e-Certification process, *Verifier* can remain completely oblivious of any information that the academic transcript of the *Holder* contains, but still can proceed forward as long as the process yields favorable result. This new way of e-Certifying credentials will certainly trigger shifts in how organizations conduct businesses.

Additionally, the distinction between fully decentralized and distributed but permissioned blockchain frameworks is required to be high-lighted. As the blockchain services are available to help businesses to develop digital business models in their own networks, keeping governance and thus control in the distributed but permissioned frameworks are well justified, which otherwise are not easily feasible with fully decentralized counterparts (Alan Kernahan et al., 2021). To this effect, technology solution providers offer Blockchain-as-a-Service (BaaS), a cloud-based offerings, that enable clients to develop and utilize blockchain applications to fulfill their business needs. The benefit thereof is that the clients do not need to build and maintain their own blockchain infrastructure and applications, rather they pay the service fees periodically as agreed between the two entities. The Hyperledger Indy is also available from many cloud-based service providers. Any businesses can develop the applications they need over BaaS model to serve their respective customers by subscribing to the services and having the right degree of centralized control with its permissioned capability.

Finally, the research achieved the general objectives laid out in the study. Although a prototype can be created based on the research findings, to acquire buy-ins from network participants – academic institutions, students, and employers, for example – is a substantial change management effort. As the technology is not yet in a mature stage despite an uptick trend in adapting it, there will be laggards when it comes to adapting to new technology. Moreover, regulation and policy related efforts should be instigated to have the meaningful and sustainable future development of blockchain technologies to solve any business problems. Potential, however, is substantial.

REFERENCES

Aublin, P., Mokhtar, S.B., & Quema, V. (2013). RBFT: Redundant Byzantine Fault Tolerance. *IEEE 33rd International Conference on Distributed Computing Systems (ICDCS), 1,* 297-306. 10.1109/ICDCS.2013.53

Alan Kernahan, A., Bernskov, U., Beck, R. (2021). *Blockchain out of the Box – Where is the Blockchain in Blockchain-as-a-Service?* Proceedings of the 54th Hawaii International Conference on System Sciences. Hawaii, United States. 10.24251/HICSS.2021.520

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). *Medrec: Using blockchain for medical data access and permission management*. IEEE 2nd International Conference on Open and Big Data (OBD). Austria. 10.1109/OBD.2016.11

B. Houtan, A. S. Hafid, & D. Makrakis. (2020). A Survey on Blockchain-Based Self-Sovereign Patient Identity in Healthcare. *IEEE Access*, *8*, 90478-90494. 10.1109/ACCESS.2020.2994090

Blockchain technology overview. (2020). NIST. https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf

Costello, K., & Rimol, M. (2019, October 21). *Gartner Identifies the Top 10 Strategic Technology Trends for 2020*. Gartner.
https://www.gartner.com/en/newsroom/press-releases/2019-10-21-gartner-identifies-the-top-10-strategic-technology-trends-for-2020

Goldreich, O., & Oren, Y. (1994). Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology, 7*(1), 1-32.

Hyperledger. (2018). *Indy Plenum*. Hyperledger. https://hyperledger-indy.readthedocs.io/projects/plenum/en/latest/main.html

Hyperledger. (2020). *Hyperledger.* Hyperledger. https://www.hyperledger.org

Hyperledger Indy SDK. (2018). *DKMS (Decentralized Key Management System) Design and Architecture V3*. https://hyperledger-indy.readthedocs.io/projects/sdk/en/latest/docs/design/005-dkms/DKMS%20Design%20and%20Architecture%20V3.html?highlight=permissioned%20#ledger-architecture

IEEE Standard Association (IEEE SA). (2000). *IEEE 1471-2000 - IEEE Recommended Practice for Architectural Description for Software-Intensive Systems.* https://standards.ieee.org/standard/1471-2000.html

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing* (NIST Special Publication (SP) 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

Norman, Alan T. (2017). *Blockchain Technology Explained: The Ultimate Beginner's Guide About Blockchain Wallet, Mining, Bitcoin, Ethereum, Litecoin, Zcash, Monero, Ripple, Dash, IOTA and Smart Contract*s. Tektime.

Pourmajidi, W., Zhang, L., Steinbacher, J., Erwin, T., & Miranskyy, A. (2020). Immutable Log Storage as a Service on Private and Public Blockchains. arXiv:2009.07834v1

Jegadeesan, S., Naghulkirthik, K.S., Akash, A., & Akash, A. (2021). ESDCM: Efficient and Secure Data Communication using Multi Secret Sharing Encryption for Medical Applications. *Turkish Journal of Physiotherapy and Rehabilitation, 32*(2).

Tschorsch, F. & Scheuermann, B. (2016). Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communication Survey Tutorial, 18*, 2084-2123. https://doi.org/10.1109/COMST.2016.2535718

World Wide Web Consortium (W3C). (2021). *Decentralized Identifiers (DID) v1.0*. W3C. https://www.w3.org/TR/did-core/

World Wide Web Consortium (W3C). (2021). *Use Cases and Requirements for Decentralized Identifiers*. W3C. https://www.w3.org/TR/did-use-cases/

World Wide Web Consortium (W3C). (2019). *Verifiable Credentials Data Model 1.0*. W3C. *https://www.w3.org/TR/vc-data-model/*

Zhang, R., Xue, R., & Liu., L. (2019). Security and Privacy on Blockchain. *ACM Computer Surveys, 52*(3), 53. https://doi.org/10.1145/3316481

Zhang, S., & Lee, J-H. (2020). Analysis of the main consensus protocols of blockchain. *ICT Express, 6*(2), 93-97. https://doi.org/10.1016/j.icte.2019.08.001