potech LABS

WHITE PAPER

Blockchain in the Education Sector

WORK DONE IN COLLABORATION WITH :

LaRRIS
Laboratoire de Recherche en Réseaux, Informatique et Sécurité

Through the introduction of the Blockchain technology in the education sector, the automation of processes and services with the use of smart contracts, resource savings are considerable. This will improve as well the student experience and overall campus efficiency, and allow the education sector to remain competitive in an increasingly volatile market.

Managing students' identities goes far beyond security and falls within the realm of the users' experience. Nowadays, the Self Sovereign Identity places the foundations for a new approach, at the crossroads between cybersecurity, personal data protection, resilience and access control.

Table of Contents

Preface 1
1. The challenges in the education sector 5
2. The blockchain technology 6
2.1. Introduction to the blockchain technology 6
2.2. History 7
2.3. Main pillars of the blockchain technology 7
2.4. Benefits of blockchain-based solutions 8
2.5. Permissioned and Permissionless blockchains 9
2.6. Security perspectives 10
3. DLTs applications in education 11
4. Building a blockchain platform for the education sector using Hyperledger Fabric 13
4.1. Hyperledger Fabric Architecture Overview 14
4.2. Features 17
4.3. Advantages of leveraging Hyperledger Fabric 17
4.4. Hyperledger Fabric for certificate management 17
5. The Self- Sovereign identity 20
5.1. Verifiable credentials models 20
5.2. What is Self-Sovereign Identity? 21
5.3. What are Decentralized Identity? 22
6. Leveraging Hyperledger Indy, Ursa, Aries 26
6.1. Hyperledger Ursa 26
6.2. Hyperledger Indy 27
6.3. Hyperledger Aries 27
6.4. Use-case description 28
6.5. Implementation 28
6.6. VON Ledger 28
6.7. Credential Issuance 29
6.8. Presentation Proof Request 31
7. Conclusion 32
8. References 33

Table of Figures

Figure 1: Connected blocks. 6
Figure 2: Hyperledger Umbrella 14
Figure 3: Hyperledger Fabric consensus 15
Figure 4: Hyperledger Fabric modular architecture. 16
Figure 5: Hyperledger Composer 16
Figure 6: Hyperledger Fabric Network. 18
Figure 7: Data Entry Administrator interface. 19
Figure 8: W3C verifiable credential model 20
Figure 9: DID 22
Figure 10: Trust Over IP Technology Stack. 25
Figure 11: Hyperledger Identity History. 26
Figure 12: VON network web interface. 29
Figure 13: The university Public DID written on the blockchain 29
Figure 14: Schema written on the blockchain 30
Figure 15; University-Alice request invitation 30
Figure 16: Employer Public DID. 30
Figure 17: Proof Request/Presentation. 31
Figure 18: Alice verification response. 32

1. THE CHALLENGES IN THE EDUCATION SECTOR

Most universities, colleges and schools have different facilities as part of their computer labs, administration and student resources, managed by different applications and relying on multiple sources to identify their users. Most IT teams were struggling in managing users' identities using legacy system or open source **I**dentity and **A**ccess **M**anagement (IAM) solutions. Due to covid-19 pandemic in early 2020, educational institutes witnessed a dramatic change in teaching since they were obliged to close their premises to avoid the spreading of the virus. Classrooms moved from being physical to virtual remote classes using multiple e-learning digital platform. Because of highly sensitive data, the educational sector become a profitable target for hackers. Since most of the educational institutes were facing security crises, new challenges are added to the security system to identify and manage the users' access to these platforms.

The most relevant challenges include but not limited to, legacy identity infrastructure, student lifecycle and users access complexity and new cyber threats.

Legacy identity infrastructure

Many individual schools or larger educational institutions rely on highly distributed system based on one or multiple sources of legacy identity solutions (usually a directory such as Active Directory or LDAP, and sometimes, Human Capital Management software), or open sources IAM solutions to identify users. Besides, each facility within each institution may rely on a different source for verifying user credentials (e.g., they may rely on Student Information System to identify faculty and staff users. They might also have additional systems they may need to rely on, such as the library, athletics programs, and alumni organizations).

Student lifecycle and user access complexity

Usually, in a business environment, employees can change roles, however they have only one role at a time. In schools and higher education systems, the same person might hold multiple roles as being a student, teacher, administrator, alumni, or parent at the same time. Although, at the end of every year, students get graduated and new students enroll. The identity system also needs to support temporary guests, like substitute instructors or students who are taking online courses.

New Cyber Threats

Older identity management infrastructures and solutions are considered by malicious entities as attacked vectors beginning with weak or stolen credentials. Adding to it, with the Covid 19 pandemic, users (students, instructors, and staff) connect to the educational institution system using their personal devices (e.g., computers, tablets, and phones). Managing permissions across devices is difficult, especially as different devices have different levels of security and present different risks.

Without a next-generation identity solution, higher education institution may remain at risk of potential and harmful attacks triggered by malevolent opponents.

2. THE BLOCKCHAIN TECHNOLOGY

2.1 Introduction to the blockchain technology

Blockchain is a shared, distributed immutable ledger that facilitates the process of conducting and recording transactions to help a network of businesses keep track of his transactions. However, why named “blockchain”? This technology owns its name to the way it stores transactions data. As it is seen in Figure 1, the Blockchain is formed of blocks that are linked together to form a chain.

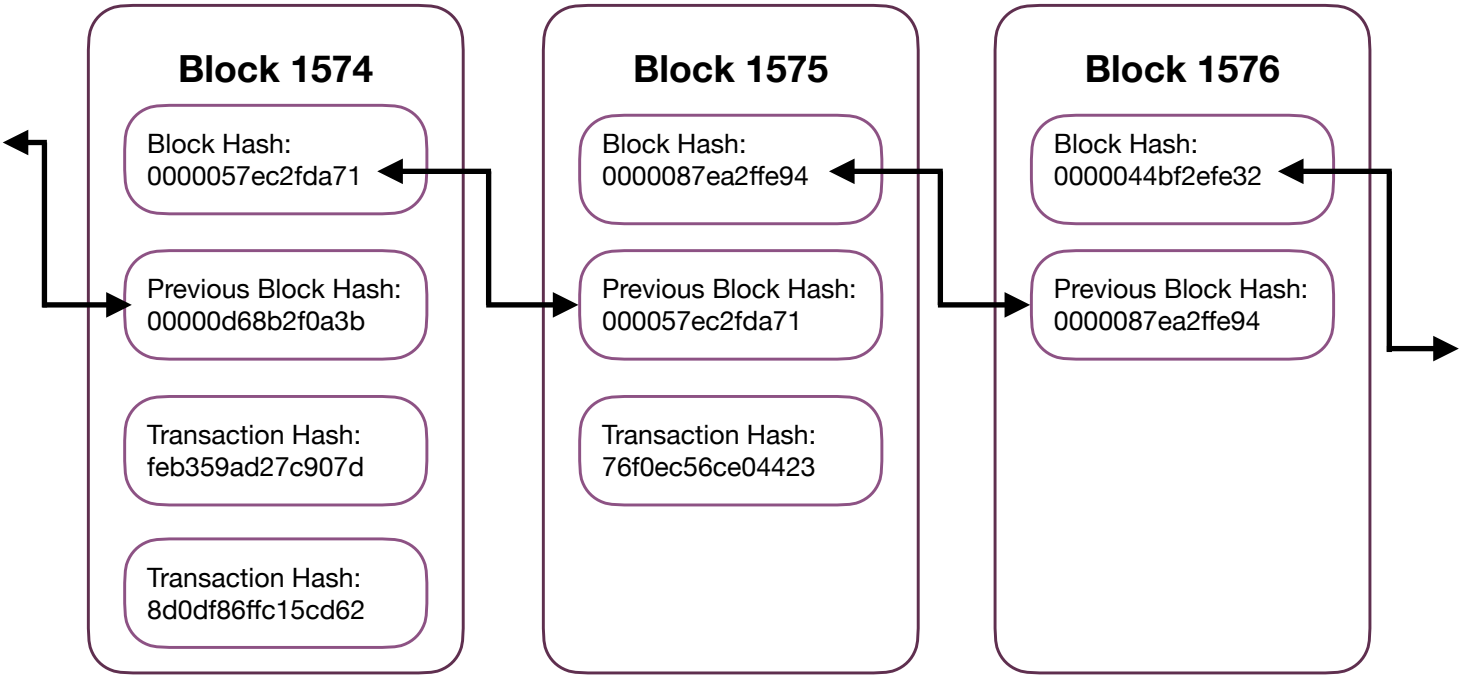


Figure 1: Connected Blocks

Each block contains a hash, timestamped batches of the recent valid transactions, and the hash of the previous block. The previous block hash links the blocks together and prevent any block from being altered or even replaced by another.
In this way, if a block has been altered it is easily discovered for example, the hash of the block 1575 will not match the record of the “Previous Block Hash” located in the block 1576. Hence, with this method the entire blockchain is immutable [1].

Similarly, to the way an accounting ledger operates, everything that happens in the blockchain system is kept in a ledger, which is then stored in every computer “called nodes” that is a part of the network.

The particularity within the blockchain lies in all the process that a given transaction has to go through to become written in that ledger. The blockchain network is decentralized, which means there is no single entity that is responsible for the whole network: only after all members of the network validate any given modification to the network, will the modification becomes a block and it is added up to the existing ledger. When that modification has been validated and added, by using complex math based cryptographic encryption, it leaves a “fingerprint” (hash), which is summed up to all the blocks before it. It only means that every single block that is part of the chain contains this fingerprint. This process highlights the consensus.

2.2 History

The main purpose of blockchain is to allow digital information to be distributed and recorded, but not edited. It was a little difficult to understand its goal until the first application of blockchain appeared in 2009.

The concept of blockchain first appeared in 1991 by two researchers Stuart Haber and W.Scott Storentta; who wanted to implement a system where document timestamps could not be tampered. However, it was not until two decades later this technology had its first real-world application with the launch of Bitcoin in January in 2009.

The Bitcoin protocol built based on the blockchain. In a research paper introducing the digital currency, Bitcoin's pseudonymous creator Satoshi Nakamoto referred to it as "a new electronic cash system that's fully peer-to-peer, with no trusted third party" [2].

In 2013, Vitalik Buterin introduced Ethereum which is an open stateful programmable blockchain that allows more functionalities due to its architecture based on the concept of accounts, smart contracts utilizing turning-complete scripting language and actual balance outlaying Ether while in the first generation, Bitcoin relied on the concept of unspent account value [3].

Other communities paved also the blockchain technology. In December 2015, the Linux foundation created the hyperledger project.
The objective is the development of the partnership between industries by advancing blockchains. Many companies are working together to support production business networks.

2.3 Main pillars of the blockchain technology

The three main properties of Blockchain technology, which have helped it spread:

- **DECENTRALIZATION**: The traditional centralized client-server model has several vulnerabilities. Centralized model means all the data is stored in one spot; this makes these spots easy targets for hackers. This centralized entity could somehow shut down for an unknown reason, nobody will be able to access the information that this entity possesses. Moreover, if that centralized entity will be corrupted or targeted then all the data that is inside will be compromised. In a decentralized network, if a person wants to interact with another then he can directly do so without the intervention of a third part.

- **TRANSPARENCY**: A person identity is hidden via a complex cryptography and represented only by their public address. So if you want to look up a person transaction history you will not see "Alice sent 2BTC" instead you will see "1Maersd9vvZjhs9oijps12 sent 2BTC". Even though the real identity of a person is secure, all the transaction history is referred to the public address.

- **IMMUTABILITY**: It means once something entered into the blockchain, it cannot be tampered. The reason why the blockchain get this property is of the cryptographic hash functions.

2.4 Benefits of blockchain-based solutions

To deploy a blockchain solution based we should have collaborating participants who are issuing transactions around a set of common assets in the network.

What are the benefits of a blockchain that we can make use to further our use case?

- **Shared ledger**: With a shared ledger, a transaction is shared (distributed) once, eliminating the duplication of effort that is typical of traditional network. Therefore, the shared ledger is the system of record, the single source of truth.
- **Provenance**: It is a complete history of all transactions related to the assets recorded on the blockchain.
- **Immutability**: A transaction that has been stored on the blockchain cannot be edited, deleted, or have transactions inserted before it.
- **Contracts**: Smart contracts hold the business logic for transactions and are executed across the network by the participants endorsing a transaction.
- **Finality**: Once a transaction is committed to the blockchain, it cannot be "rolled back" - it is considered final.
- **Consensus (agreement):** it is the process of agreeing on new transactions to put them on a ledger after those transactions are verified, and distributing them to the participants. Consensus mechanisms vary from blockchain to blockchain, but includes the following:
    - Proof of stake: this process is used it to validate transactions, validators must hold a certain percentage of the network's total value. It can be a way to provide more protection from malicious attack on the network by reducing attacks and making it very expensive to execute attacks.
    - Multi-signature: A majority of validators in a network should agree that a transaction is valid.
    - Practical Byzantine Fault tolerance (PBFT): PBFT is an algorithm designed to resolve disagreements among the network participants

These benefits help the network to be trustworthy. But it is not always a good idea to think of a blockchain as a solution for everything.

There are many reasons this technology won't be a good fit:

1) It is not suitable if only one participant is present on network.
2) It cannot be a replacement for traditional database or transaction server.
3) Because of blockchain decentralized nature is a peer-to-peer network by design and based on cryptography, with those benefits comes some nonfunctional requirements.

2.5 Permissioned and Permissionless blockchains

Blockchain technology can be permissioned or permissionless. Permissionless, like most digital currency blockchains, allow all users to write on the ledger. There is no permission needed from anyone to become a node on the network.

As for a permissioned blockchains network, the participants should first be authorized from one or several parties in order to become a node in the network. So in a permissioned network every transaction is cryptographically signed, which provides authenticity of which node sent it, offers as well nonrepudiation so the participant cannot deny sending a transaction, in addition it preserves the integrity because we can have on the network stewards that are responsible to look at the information that is written on the ledger.

Table I highlights a comparison between several **D**istributed **L**edger **T**echnologies (DLTs) [3].

	Bitcoin	**Ethereum**	**HyperLedger**
Communities	Bitcoin developers	Ethereum developers	Linux Foundation
Blockchain type	Permission-less	Permission-less	Permissioned
Currency	BTC	Ether	None
Consensus	PoW (based on SHA-256)	PoW (Ethash)	PBFT (excluding Corda)
Private transaction mode	No	No	Yes
Anonymity	No	No	No
Stimulus	Economics incentive, fees and rewards	Economics incentive, fees and rewards	Reputational Risk
Censorship resistance	No	Yes	No
Limit	7 transactions/sec	20 transactions/sec	No
State concept	No	Data	Key-value
Smart contract languages	No	Solidity, Serpent, Mutan, LLL	Chaincode
Cross-contracts	No	Yes	Yes
Scalability	No	No	Yes
Block time	10 min	15 secs	Subject to the peers involved
Variants	+ 700 variants	Olympic, Frontier, Homestead, Metropolis (Future release) and Serenity (To be announced)	Burrow, Fabric, Iroha, Corda, Sawtooth
GPU cost	Yes	Yes	No
Auditing Mechanism	No	No	Yes
Applications	Digital Registry, Crypto Currency	Digital Registry, Crypto Currency, Smart Contracts	Digital Registry, Smart Contracts
Languages	C++	Golang, C++, Python	GoLang, Java

2.6 Security perspectives

Blockchain is a technology emerging in the cyber world as a security tool despite the fact it is not its first functionality.
Blockchain is seen as a cybersecurity measure as this distributed ledger allows secure transactions and keeps the identity of the participants involved anonymous. Moreover, everything written on the ledger are validated, all peers in the network have to accept the changes made within the blockchain. In addition, all the information remains private through a complex encryption process.
This trend is powering security in different fields and is being applied in the education sector, which will be the topic for this study, transactional systems, healthcare data management and many other industries. Using the security of blockchain, organizations could save a lot of money and time by reorganizing transactions and securing them with blockchain. Although blockchain is a new technology entering our environment, so its development, changes, and improvements are still needed before mainstream [4].

Blockchain fulfils the needs of security. Identifying the main guidelines in security depending on the CIA triad we can enumerate:

- Confidentiality is maintained due to encryption and the use zero-knowledge process.
- Integrity is accomplished by the use of the consensus protocol.
- Availability - all members of the network share the records and hashes for all transactions completed and validates, and it is a distributed network. No point of failure can be found.

3. DLTs APPLICATIONS IN EDUCATION

Blockchain and DLTs provide a distributed and interoperable environment for the higher education system which focuses on a globally incorporated viewpoint for students and institutions. Potential employers can profit from the blockchain platform as well.

Students can take advantage of having their completed course history in a single and transparent view, as well as universities which have this data accessible and up to date, regardless of a student's educational origins. On the other hand, different administrations (such as employers, universities, etc.) as potential operators of the system, can certify the provided information after permissions are obtained.

Besides Blockchain platforms can be used for E-learning systems, can lower the cost and build trust and transparency in education.

Today it has become easy to forge a diploma. Until very recently, to guarantee the veracity of the diplomas obtained, it was necessary to request a true and certified copy from the issuer. Blockchain technology now makes it possible to offer a solution to guarantee the authenticity of the diploma without a tedious process.

APPLICATION
Certificates management
Competencies and learning outcomes management
Evaluating students' professional ability
Securing collaborative learning environment
Protecting learning objects
Fees and credits transfer
Obtaining digital guardianship consent
Competitions management
Enhancing students' interactions in e-learning
Examination review
Supporting lifelong learning

By integrating Blockchain technology in Education many significant benefits could be brought to education. The most relevant ones are high security, better control of data access, enhancing accountability and transparency, identity authentication and enhancing trust.
The security and privacy of the exchanged data/transactions between the intended parties were assured when using Blockchain. In fact, the order of the transactions was maintained by the consensus protocol which yields to reduce the risk of unsecured ones, while the intended parties were able to verify the ledger content [5]. The reliability of the transactions was assured by the cryptographic hashes and signatures where the blockchain was used to sign and validate learning traces and records [6]. Although an encryption algorithm was performed on the various types of data before sending it to other participants to protect these data.

Another great benefit of blockchain is the access restriction and control to the stored records. In fact, educational records include transcripts, diploma, or personal students'/teachers records. In Reference [7], a permissioned blockchain platform was used to restrict access to academic credentials and limit it to the intended participants only, such as certified institutions under specific rules [8]. Moreover, using Blockchain technology enhances the accountability and transparency of using educational records and ease the accessibility as well as the flexibility of analyzing, correlating, or distributing such information [9].

Two major benefits from the blockchain are the authenticity of the digital certificates as well as the identity of users. In Reference [10], a digital syllabus was stored in a blockchain. Every created block will be signed by the authorized university using a private key. To avoid tampering with the content, a cryptographic hash of the course syllabus will be issued. The university verifies the authenticity of these data through the hash and the key that belongs to the original institution and only trusted parties can add blocks to the network or gain access to it [11] which makes the trust another benefit to add to the blockchain technology.

BENEFITS

- Enhancing learner's activity
- Supporting learner's career decisions
- Improving the management of student's records
- Enhancing trust
- Identity authentication
- Enhancing accountability and transparency
- Better control of data access
- Enhancing student's assessments
- Low cost
- High security

What Are the Challenges of Adopting Blockchain Technology in Education?

Despite the many benefits of using technology blockchain in an education context, multiple challenges are to be considered when employing it in the education field. One of the most crucial challenges is the risks of malicious attacks despite the main feature of security in blockchain technology. In fact, to ensure privacy, many blockchain frameworks rely on the use of public and private keys. However, since the public key is publicly visible, user's transactions can be linked to reveal user's information. Private keys storage and protection is another security issue that should be considered. Frequent updates of data can cause data leakage which is a security concern that should be considered [12]. Continuous transactions and record growth used to keep track of students moving from school to school will generate big size blocks that cause slow speed blockchain transactions [13]. This big challenge might hinder blockchain development in education.

Although, in order to verify certificates, all institutes should agree to share their data. A big concern arise on how can authorized organizations take the risk to share their student's credentials. If they don't agree to provide these data, more complications will be created in the authorization process [14].

Another key challenge is the ease of use of traditional systems against blockchain technology. Good designs and easy terminology can help blockchain adaptation in the education sector.
The immutability nature of blockchain makes it more difficult to edit the data unless everyone agrees to change the content of the ledger.
Another key issue is in the decentralized blockchain technology which will affect the centralized nature of the process in educational system. As in blockchain, the availability of a continuously aggregating ledger can affect the value of the traditional school credential [15].

4. BUILDING A BLOCKCHAIN PLATFORM FOR THE EDUCATION SECTOR USING HYPERLEDGER FABRIC

Hyperledger is a collaborative effort to adapt blockchain technologies to the needs of businesses responding to major challenges.

Blockchain is a very promising technology. IBM therefore invested in Ethereum in 2014 with several projects. However, Ethereum is thought of as a public blockchain and does not respond to business challenges: confidentiality, management, governance, PoW consensus in its early release, etc.

IBM therefore collaborated with 90 other companies to develop a new technology that can be widely adopted by industry. Despite the number of players, IBM remains the major contributor with 44,000 lines of code submitted to the Linux Foundation.

IBM provides a complete set of means to exploit the blockchain: Architecture, cloud, development. The Linux Foundation incubates and promotes numerous projects derived from Hyperledger: Hyperledger Fabric, Hyperledger Composer, Hyperledger Sawtooth, Hyperledger Iroha, Hyperledger Indy, Hyperledger Burrow, etc [16]. In this section, we will discuss and present our implementation using Hyperledger Fabric.

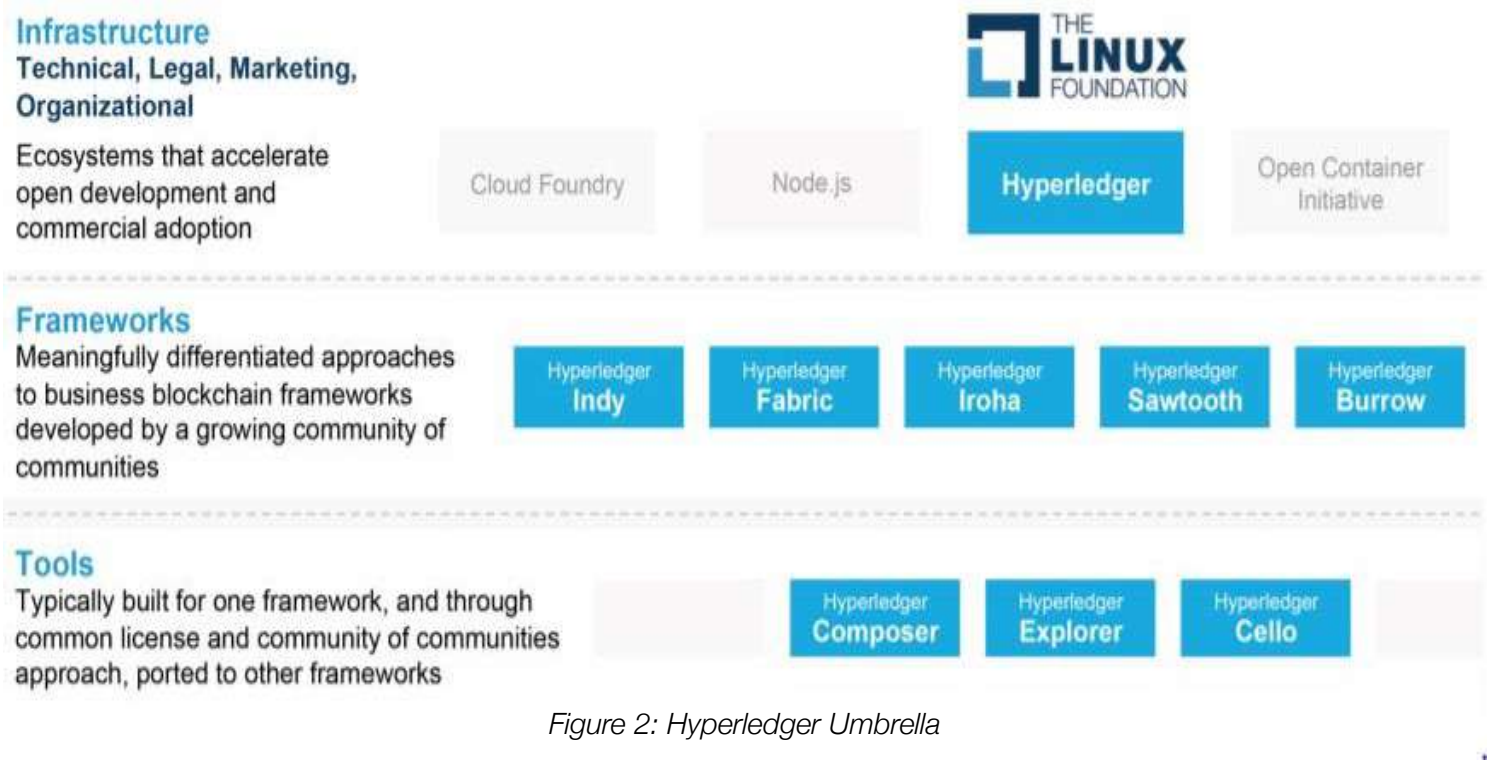


Figure 2: Hyperledger Umbrella

IBM is also joining forces with other IT giants to provide more services using blockchain:

- **Docker Hub:** Docker Hub and IBM have joined forces to offer a cross-platform distributed network integration solution. In others terms, allow a distributed network to be integrated into business infrastructures. The Docker Hub is designed to deploy and support decentralized applications in our own environment.

  The package also includes the Docker Trusted Registry which provides image storage.

- **Secure Key:** IBM has partnered with Secure Key to develop identity-related solutions. They seem to be ahead of identity issues, particularly in banking (KYC, etc.).

4.1 Hyperledger Fabric Architecture Overview

IBM provides a turnkey solution based on Hyperledger and Fabric to build, run and manage its blockchain live.

Hyperledger Fabric provides a service that facilitates the development, the deployment and the management of decentralized networks, while ensuring high levels of security, confidentiality and performance. It is also a service that allows you to fully own your blockchain network and control its governance: network access, permissions, rules and policy.

Hyperledger Fabric projects are completed on a private network and operate on what are commonly called "permissioned blockchains". Membership in the network is therefore controlled, and members may need additional authorization for certain sensitive actions.

Unlike public blockchains, where all transactions are transparent and accessible to everyone, transactions on Hyperledger Fabric are confidential and are visible only to the participants involved.

Hyperledger does not include cryptocurrency. On the other hand, it is possible to develop smart contracts called "chaincodes", in Go or Java, or even in JavaScript.

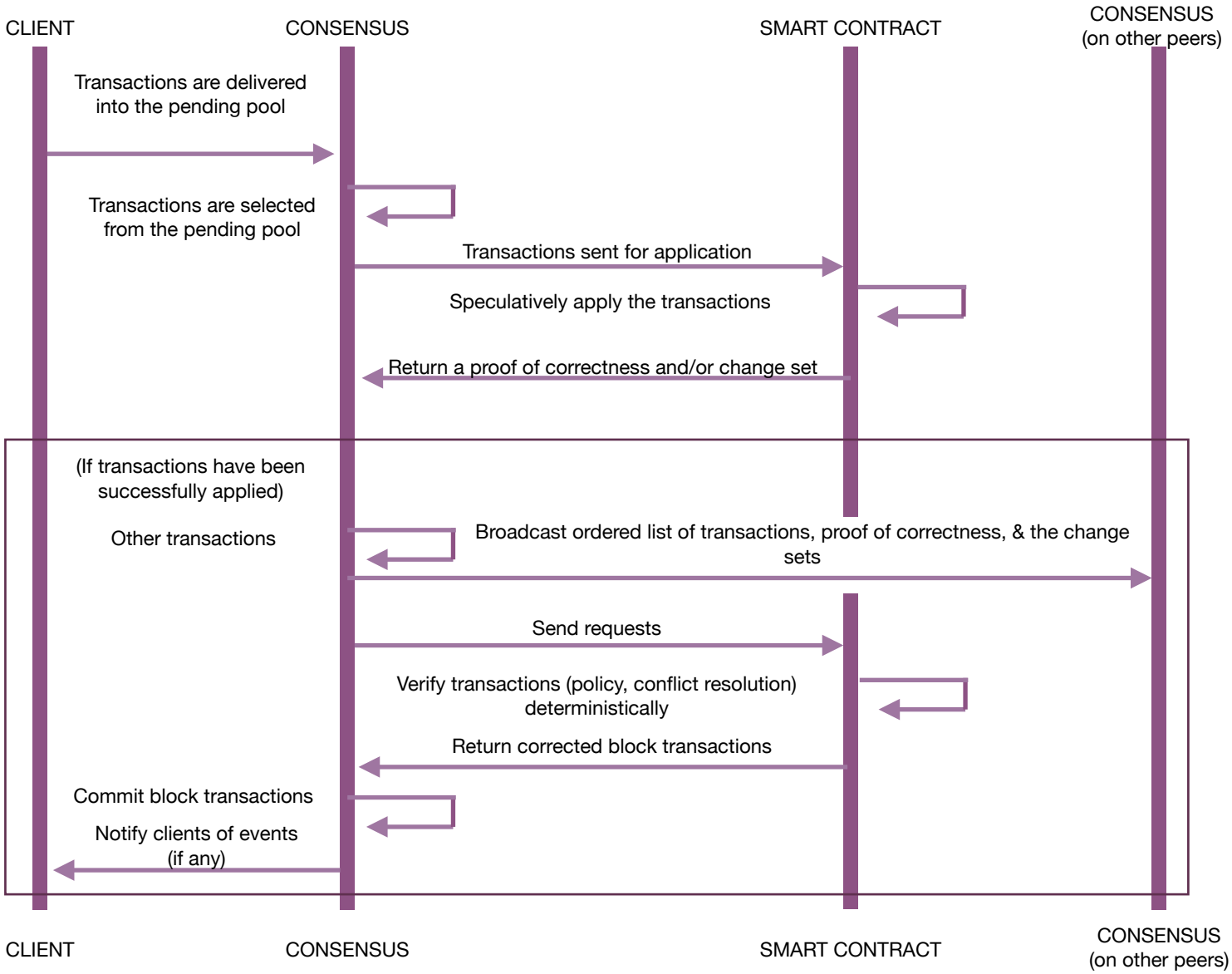


Figure 3: Hyperledger Fabric Consensus

The basic workflow of a transaction in a Hyperledger Fabric blockchain begins by two or more participant, such as a company or organization that creates and joins a channel.

Initially, the members in the channel agree on the terms of the chaincode that will govern the transaction as well as the policies governing the channel membership. When a consensus is reached on the proposal to deploy a given chaincode, it is committed to the ledger.

The Hyperledger Fabric follows multiple business blockchain components:

- Consensus Layer - Endorsing the correctness of the set of transactions that constitute a block.
- Smart Contract Layer - Processing transaction requests and executing business logic.
- Communication Layer - Peer-to-peer message transportation.
- Data Store Abstraction – Enables other modules to use different data-stores.
- Identity Services - Providing a root of trust, authentication and authorization.
- Policy Services - Policy management in the system namely endorsement policy, consensus policy, or group management policy.
- APIs - Enables clients and applications to interface to blockchains.
- Interoperation - Interaction between different blockchain instances.

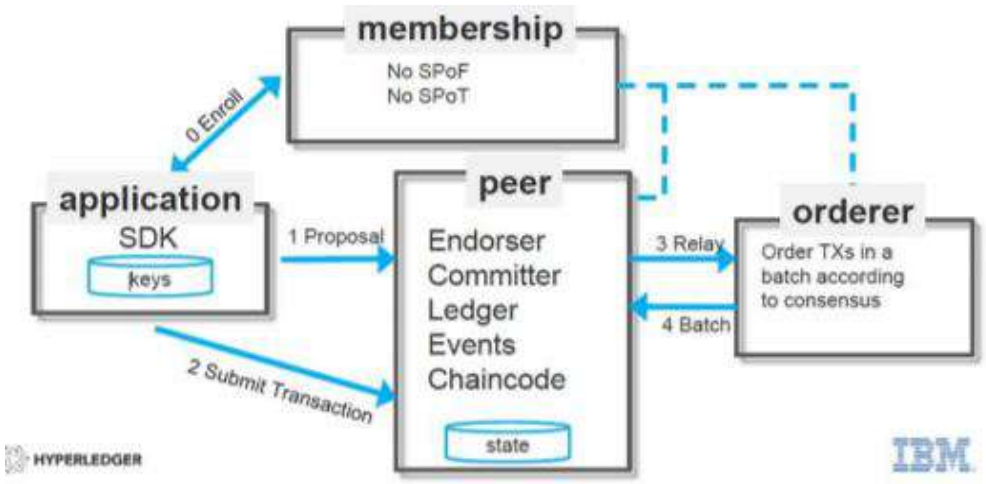


Figure 4: Hyperledger Fabric Modular Architecture

The Hyperledger Fabric architecture is comprised of the following components: peer nodes, ordering nodes and clients.

Peers are a fundamental element of the network because they host ledgers and smart contracts. Recall that a ledger immutably records all the transactions generated by smart contracts. Smart contracts and ledgers are used to encapsulate the shared processes and shared information in a network, respectively. Ordering nodes order transactions according to the consensus.

The client represents the entity that acts on behalf of an end-user. It must connect to a peer for communicating with the blockchain. The client may connect to any peer of its choice. Clients create and thereby invoke transactions.

These components have identities derived from the certificate authorities. Hyperledger Fabric uses a modular architecture -the code is broken down into several modules, which are themselves a set of functions. Building an application then boils down to nesting these components, the modules, among themselves (like object-oriented programming).

End users with the right privileges can propose transaction to endorsers in the channel that executes the chaincode. The endorsing peer verify the signature of the proposal and determines if the client is authorized to perform the proposed operation.

Endorsers use the transaction proposal as input and execute them against the current state database to produce a transaction result. The transaction result includes a response value, read set, and write set. No updates are made to the ledger at this point. The transaction result along with the endorsing peer's signature and a YES/NO endorsement statement are passed back to the client. The client ensures that the results from the endorser are consistent and signed, and send the transaction, comprised of the result, endorsement, and the channel id, to the ordering service. The ordering service does not read the transaction details; it simply orders transactions by channels as First-Come-First-Served basis into a block which is then sent to the peers to be committed to the ledger. The committer nodes validate the transactions within the block to ensure endorsement policy is fulfilled and to ensure no changes has been made to the read set variables in the ledger state. Each transaction in the block are tagged as being valid or invalid and when the peer appends the block to the channel's blockchain; only the write sets of valid transaction are committed to the current state database

This architecture has many benefits [17]:

- Easier to use, because the "services" are already produced.
- Less chance of bug because the modules are constantly used and reviewed.
- Flexibility. It is for example possible to make unit tests, important for the progressive improvement apps.
- The organization of the code in different logical units.
- Easier code sharing.
- Possibility of building libraries on it. (cf. Hyperledger Composer below)
- Thanks to the encapsulation, Fabric respects the fundamental principles of Hyperledger: that is, it delivers high levels of confidentiality, scalability and flexibility.
- Fabric makes it possible to embed assets (tangible or non-tangible goods, services, properties), related transactions and the participants.

If Hyperledger Fabric is the foundation, **Hyperledger Composer** [18] is the framework that makes it possible to create the decentralized applications. Hyperledger Composer is a set of tools, an API, a language for developers facilitating the creation of a full-stack blockchain solution, and allowing participants to carry out transactions.

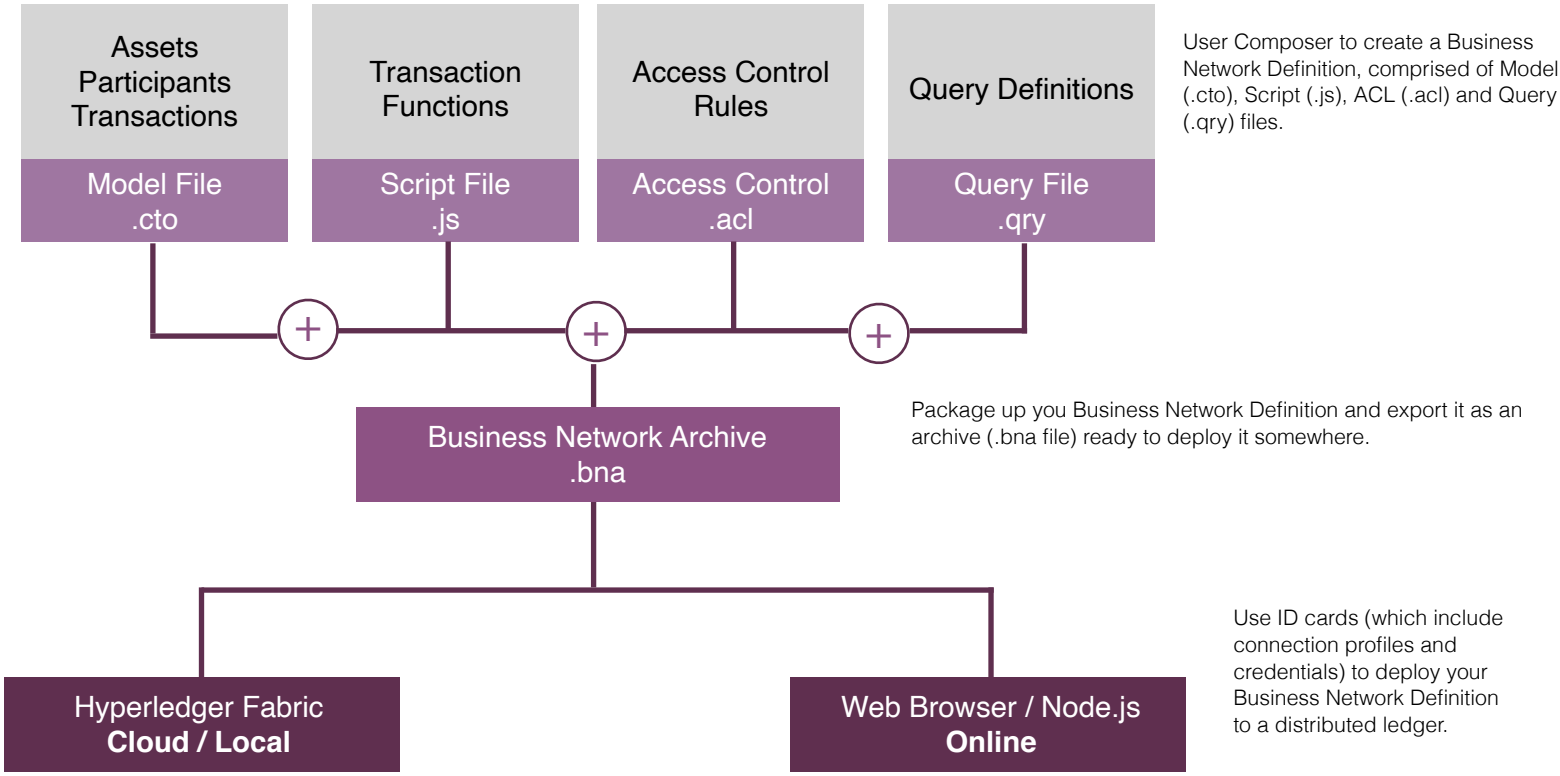


Figure 5: Hyperledger Composer

4.2 Features

The main features of Hyperledger Fabric include the following:

- Creates a dynamic distributed ledger.
- Organizes permissions across different nodes. A certification authority determines the role of each participant, the modalities (read or write) or the information to which he has access.
- Defines the rules and procedures that administer the network.
- Provides resources (storage for example).
- Creates private communication channels with selected members, for confidential transactions, for example. We are talking about information segregation.
- Defines assets and transaction instructions.

4.3 Advantages of Leveraging Hyperledger Fabric

Hyperledger Fabric is a general-purpose business platform and an open source project that can support smart contracts via chaincode and data partitioning via channels. We have considered other platforms, but eliminated them from the running over concerns about security aspects like data access control as well as concerns about being ready to withstand actual operation. Hyperledger Fabric was also selected because it allows for more flexible data models and business logic compared to other blockchain solutions.

As the endorser nodes responsible for particular chaincode are orthogonal to the orderers, the system may scale better than if these functions were done by the same nodes. In particular, this results when different chaincodes specify disjoint endorsers, which introduces a partitioning of chaincodes between endorsers and allows parallel chaincode execution (endorsement). Besides, chaincode execution, which can potentially be costly, is removed from the critical path of the ordering service [19].

Furthermore, the Hyperledger Fabric architecture facilitates the deployment of chaincodes that have confidentiality requirements with respect to the content and state updates of its transactions.

4.4 Hyperledger Fabric for Certificate Management

For the education institutions that issue diplomas, digital certifications allow real savings to be made by eliminating the management and storage of qualifications in paper format. It also becomes more efficient by automating the issuance of diplomas: students receive a unique and lasting url and the university no longer issues certificates on request.

The Blockchain technology guarantees the legitimacy of the diplomas and the storage of the diploma data on the Hyperledger Fabric blockchain. The diploma is thus secure and tamper-proof.

The graduate is assured of the permanence of his diploma, the authenticity of which he can directly assert.

For that purpose, an application was built to function as a university management system to issue academic certificate in a completely decentralized way.

On the other hand, Hyperledger Fabric provides a private blockchain therefore student records are not publicly available.

Besides, the university can customize access levels as per the requirements. In order to achieve a peer-to-peer network secure enough to store personal information a system of smart contracts was developed in the GO programming language in combination with the Fabric permissioned blockchain architecture.

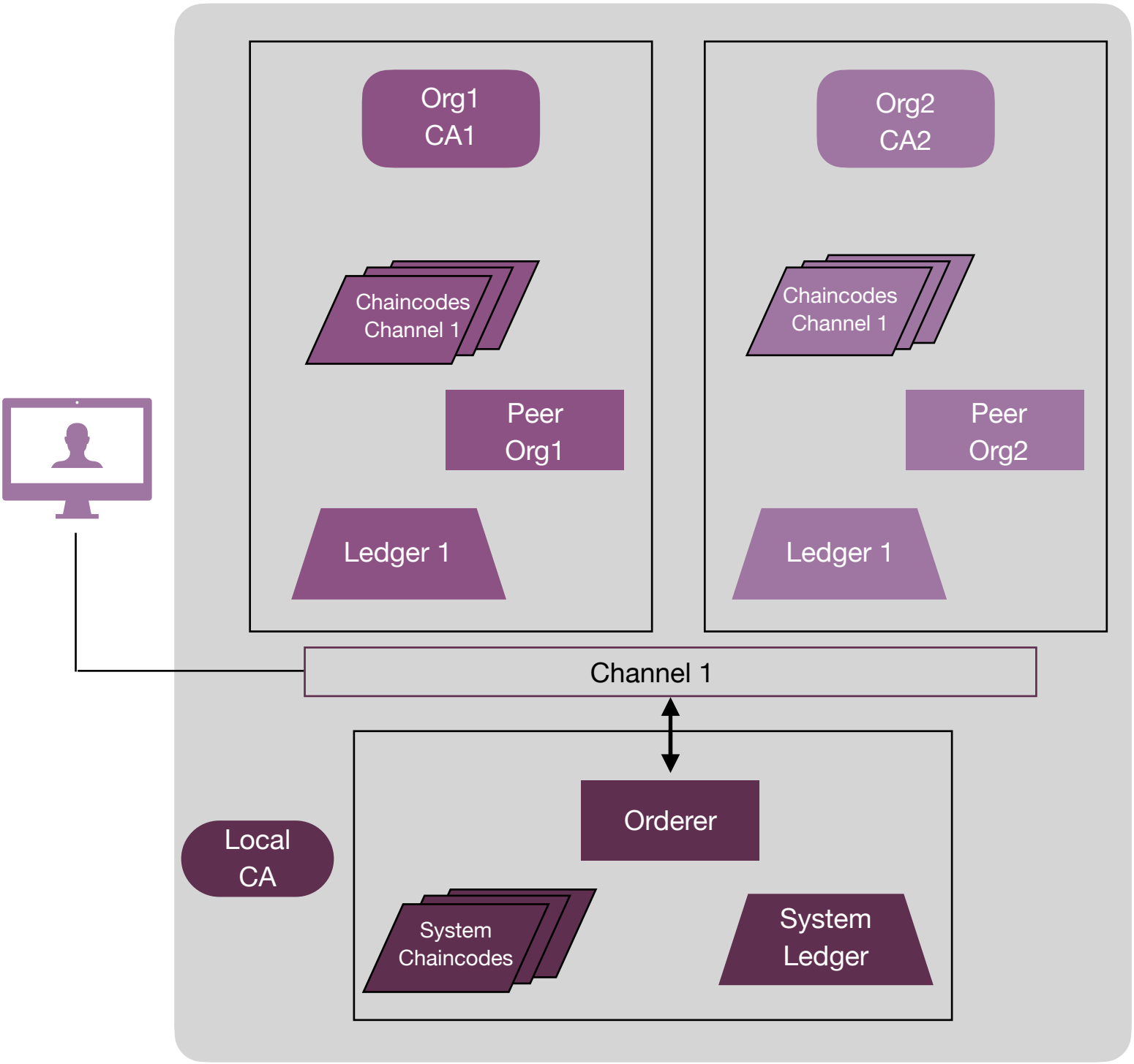


Figure 6: Hyperledger Fabric Network

After installing the Hyperledger network, the following steps are undergone:

1. Create Channel 1;
2. Make Peer 0 and Peer 1 join the channel.
3. Upon successful joining of the channel, we update the anchor peer.
4. Install the Chaincode on both peers.
5. The creation of a Data Entry Administrator which will be responsible of adding the student data and certifications to the blockchain.

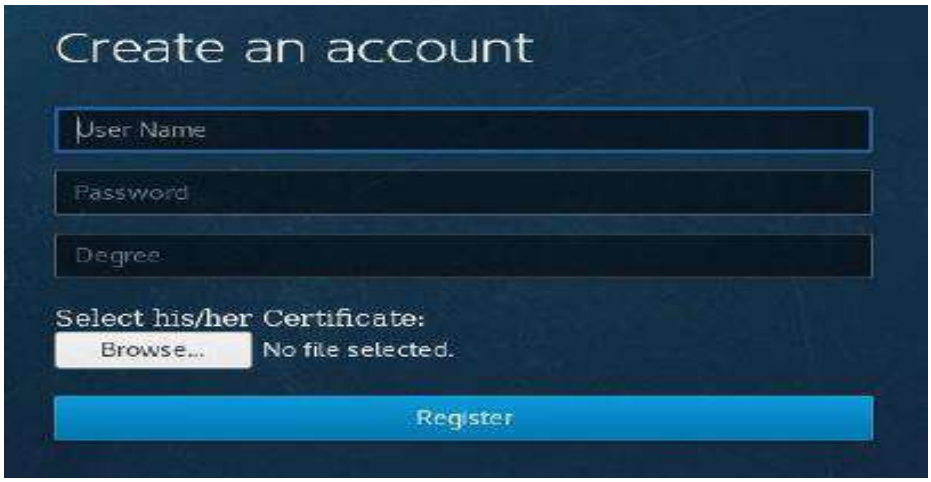


Figure 7: Data Entry Administrator Interface

1. The creation of an Administrator of data entered that will verify if data entered is correct.
2. The student connected to the same network can invoke and query his certificate from the user interface and see its results.

Through this scenario, we build a private blockchain network by using Hyperledger Fabric platform between two hosts and leveraged this network as a prototype to a blockchain in a university that allow the students to access their own official documents directly in a safe, tamperproof and reliable means.

In the following, we explore the advantages of Self-Sovereign Identity.

5. THE SELF-SOVEREIGN IDENTITY

5.1 Verifiable Credential Models

"*A credential is the abilities and experience that make someone suitable for a particular job or activity, or proof of someone's abilities and experience*" –Dictionary of Cambridge

We can conduct that a credential is an attestation of eligibility to an entity by a third party with a relevant and supposed competence to do so. Examples of paper versions of credentials are university degree, driver's license, etc.

So how can we implement a trusted digital verifiable credential model; what if it is something we can trust much more than just a scan of the paper credential?

The Hyperledger projects Indy, Aries and Ursa, which I will talk about in the next section, are tools and libraries that allow us to develop digital identities rooted on the blockchains. They help building applications that enables the verifiable credentials models.

The verifiable credential has the same concept as the paper credential model. Where we can find an authority (university/ government…) that decides if an agent (your organization /you) is eligible to receive a credential and issues one. The agent (you) hold that credential in a digital wallet. Then in some point, the agent is asked to prove that the claims are legit. So the agent must provide a verifiable presentation to the verifier. Verifiable credential and presentations are not simple to generate, created and used by anyone. They are cryptographically constructed so that a presentation proves the four key attributes of all credentials:

- Who issued the credential
- If the credential was issued to the entity presenting it
- If the claims were no tampered
- And finally if the credential has not been revoked

Here we can see the difference between the paper and verifiable credential where those four attributes are not evaluated based on a person looking at the paper credential, but rather by using complicated online cryptographic algorithms difficult to forge. Once a verifier receives a presentation from an older, they use information from a blockchain to perform the cryptographic calculations necessary to prove the attributes.

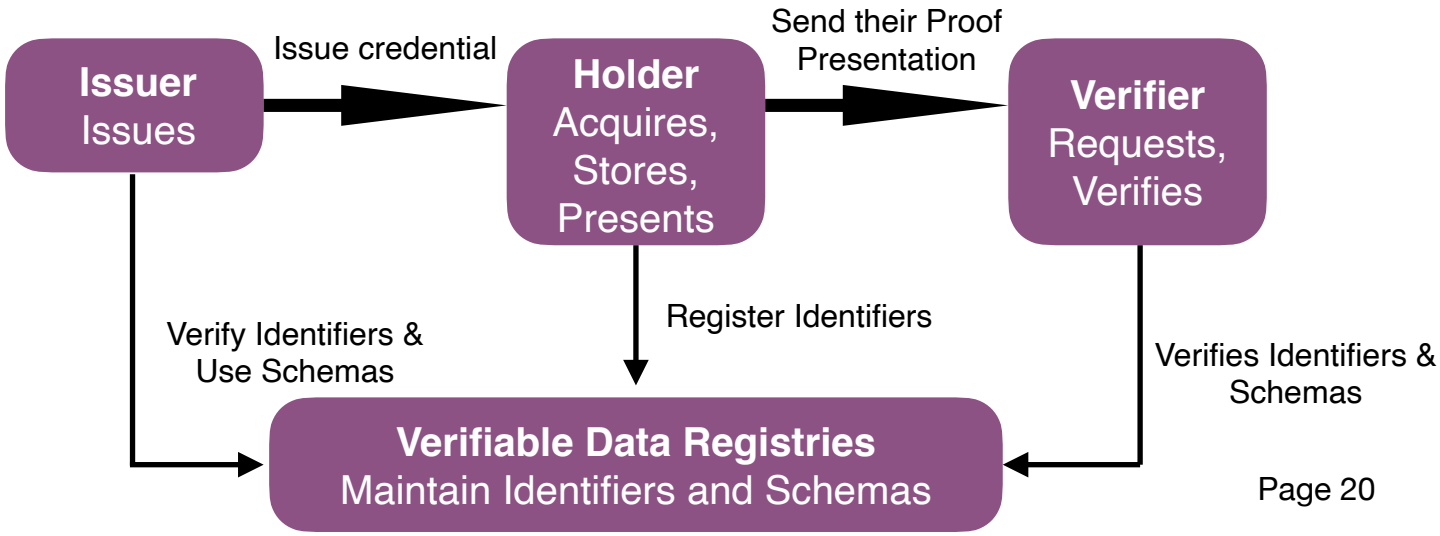


Figure 8: W3C Verifiable Credential Model

We can define the core actors of this verifiable credential [20]:

- Holder: is an entity that might possess one or more verifiable credentials and generate verifiable presentation to prove that credential are trusted. Example of holders' employees, customers' students…
- Issuer: is an entity declares the claims about a subject. Creates a verifiable credential from these claims, and it sends the VCs to the holder. Example of issuers NGO, organizations, governments, individuals...
- Subject: an entity which claims are made. Example: in many cases the holder is the subject of a credential but in certain cases it is not. A parent could be holding a credential of his child…
- Verifier: an entity verifies if a credential is trusted by processing optionally a verifiable presentation. Example verifiers include employers, websites…
- Verifiable data registry: is the system that perform by facilitating the creation and verification of identifiers, keys and other relevant data, such as verifiable credential schemas, revocation registries…

Verifiable credentials are far more secure compared with paper credential. Why so? Well the verifier is not a person trying to identify a forged document but instead the credential is cryptographically verified. When an issuer sends the credentials to the holder, the data is sent in a data structure set called credentials and they are digitally signed. When a verifier asks the holder to send him credentials, the holder send a set of credentials into a data structure called a profile and it is also sent digitally signed.

5.2 What is Self-Sovereign Identity

The new term of **S**elf-**S**overeign **I**dentity (SSI) first appeared into the internet in 2016. The SSI is a set of principles on how identity and personal data should be addressed across the digital world. In another perspective, the SSI is a package of technologies combined together, to build the new identity upon core concepts in identity management, decentralized computing, blockchain and cryptography.

The self-Sovereign identity is a new architectural design to help people reclaim trust and authority in the digital identity systems. Today's digital identities are controlled by centralized identities. People in general are being managed by these organizations. Based on the risks of centralization, a new concept was defined. With SSI there is no central authority holding your data that passes it on to others upon requests. The main idea, is that you control your own data, when it is shared and with whom. You can present claims about your identity and others can verify it with cryptographic certainty.

Ten principals of SSI include the following [21]:

- Users must have an independent existence.
- Users must control their identities.
- Users must have access to their own data.
- Systems and algorithms must be transparent.
- Identities must be long-lived.
- Information and services about identity must be transportable.
- Identities should be as widely used as possible.
- Users must agree to the use of their identity.
- Disclosure of claims must be minimized.
- The rights of users must be protected.

Drummond Reed, identity guru and Founding Trustee of the Sovrin Foundation, describes self-sovereign identity as:
"*Lifetime portable identity for any person, organization, or thing that does not depend on any centralized authority and can never be taken away.*"

SSI helps the owner of the data control his own identity but he needs someone to help issue claims and credentials. The holder of the data does not have the total control on his identity. However, the SSI concept limits the issuer to only issue the identifiers.

5.3 What are Decentralized Identity?

The identifier need to prove his identity and need to be verifiable in some way but knowing only the IP address is not enough. For several years, the use of public and private key cryptography is used to create digital proofs. In centralized systems the owner of a private key uses it as a tool to sign its message, and anyone else can verify this message by requesting the public key from a trusted third party. The signature verifies that the owner of the private key and the message has not been tampered. Thus, the validation of a content of the message depends on having the correct public key. For decentralization messaging and communications between digital agents, a new notion should be adapted. The needed requirements a strong and secure way to prove the ownership of the public keys related to the identity owner.

A new identification system is needed to bring all the requirements together. **D**ecentralized **I**dentity or DID for short is a key enabler for verifiable credentials. Based on the self-Sovereign model. They are a special kind of identifier that are created by their owner, independent of any central authority. DIDs are a new type of identifier that is in the process of becoming a World Wide Web Consortium. Because the new type of identifiers needs to be extremely secure several properties are followed:

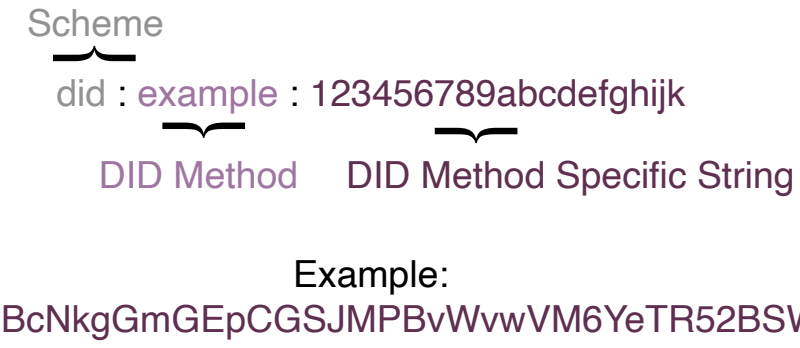


Figure 9: DID

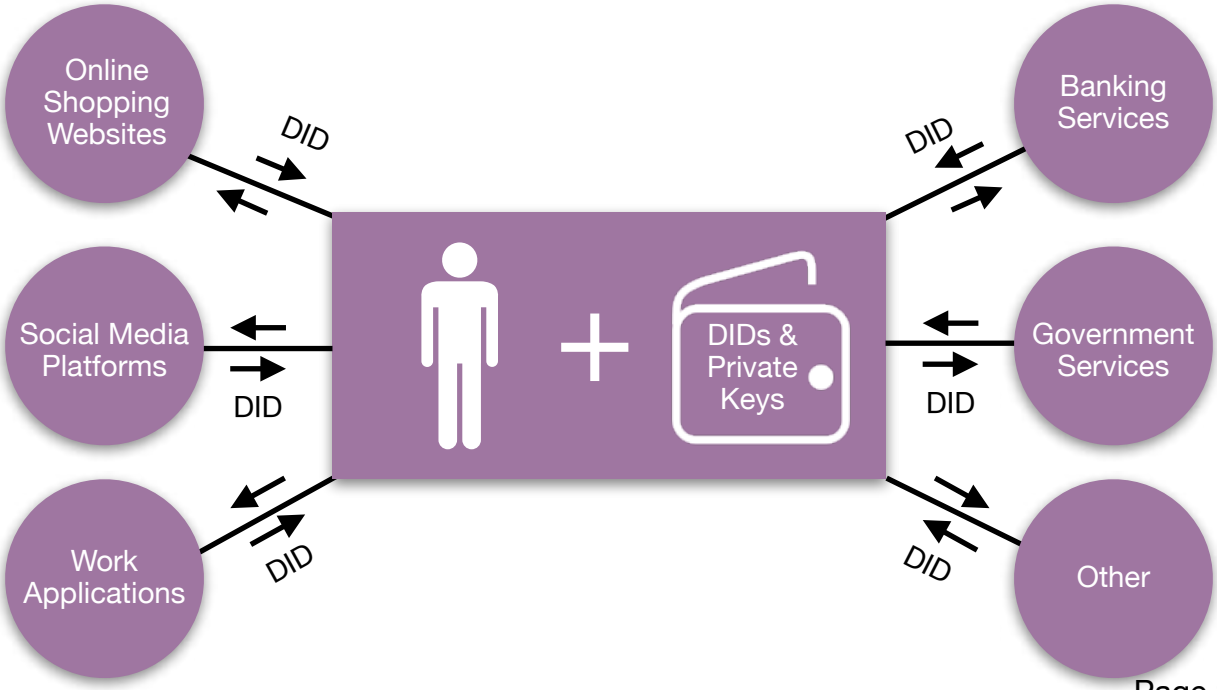
DIDs are [22]:

- A new type of uniform resource location (URL): when we pass a valid DID to a software called DID resolver, the resolver will return a DID document (DIDDoc: a JSON document format). Working like a URL, a DIDDoc states the:
    - Public keys for verification
    - Authentication protocols
    - Services endpoints necessary to initiate trustworthy interactions with the identified entity.
    - Timestamp for audit history
    - Signature for insuring integrity
    - Through the DID Document (you can find an example below, an entity should understand how to use that DID). The public Key could be published on the ledger or it can be exchanged between the agents. As for the Private Key it is held by the DID controller.

- Created by anyone at any time.
- Globally unique: taken care by the Hyperledger Indy. Following the rules for creating a specific type of DID it will be globally unique.
- The identifiers are permanent no matter where the identity owner location is. Even if the holder changes devices.
- Highly available: even if some servers are down. Since DIDs are often resolved by reading from a Blockchain, (many ledgers are available across the globe). One can conclude that we do not have a single point of failure.
- Cryptographically verifiable: means that the owner of a DID should prove its authenticity by proving their private key is related to the public key.
- Distinct from the X.509 certificates tree where we rely on centralized registries manipulated by a single entity. The new identifiers are able to avoid single points of failure using blockchain, peer to peer networks…

Importance of DIDs

- User Control: you create, control and share your DIDs. (Pillar of the SSI decentralization). In our days approximately all the identifiers are central authorities. These authorities can remove and control your ability to use your identifiers. As you are the owner of the DID and you are the one who holds the private key of your credentials. Only you can delete the DID. By “deleting” I mean the DID will no longer respond to requests for proof of control. Because blockchains are immutable.
- Provable: The association of public/private key pairs with a DID makes ongoing proof of control unimportant. (DIDs define who is the issuer and their authority to issue credential)
- Non-Correlatable: One of the main concerns that was discussed in the introduction that lead us to rethink a new concept of digital identifiers was the correlation and the lack of privacy. Since we create our own DIDs and choose how they are shared, we do not need to have just one DID, but instead of one DID for all services to which we connect. One DID for every type of service (one for every relation) is to be created. In that way, we can eliminate the user id/password concept. Both sides need to provide a DID.



- Secure Communication: An entity can encrypt a message for the controller of the DID (using the public key in the DIDDoc) and send it to the designated service endpoint. Upon receipt, the DID's controller can decrypt using the corresponding private key and process the message. A DID is not just for authentication, but it is used also to exchange verifiable digital credentials.

Public vs Private/Pairwise DID

A public DID is intended to be widely used. Since anyone can both resolve the DID and contact the owner, if the DID is used many times, it is correlatable. In this case, correlation is not a bad thing. Public DIDs are put on blockchains so that they can be globally resolved.

In case of private DIDs, instead of publishing the DIDs on the blockchain for anyone in the world to see, the entities create the DIDs and DIDDocs and then send both directly to the other party(ies) to hold in their wallets. If they need to update the DID, such as to change the endpoint, the update is sent directly to the other party(ies) to update their copy of the data. No one other than the parties involved can see or resolve the DIDs.

Zero-Knowledge Proof (ZKP):

A ZKP is a strong cryptographic technique. ZKPs can be used to ensure the validity of transactions, even though sender, recipient and other transaction data is unknown. The main idea of this algorithm is to prove an entity attributes without exposing a correlatable identifier about the entity. Here, the prover could prove that he knows a value "z" to the verifier without giving him any information other than the fact that he knows the value "z" [23].

Combined, ZKPs and selective disclosure enable a massive reduction in the data exposed during an identifying process. By not revealing a correlating identifier, and using selective disclosure to only expose the data needed for a transaction, including being able to prove possession of a credential without disclosing any information from the credential at all, is an important step in being able to keep private data private. ZKPs are not a solution for privacy. For any business transaction, the verifier must collect enough information from the prover as to mitigate their risk for the given transaction.

Experts believe that the core strategy of zero knowledge proof to be a very special case where there is no chance to convey any secret information.

Agents and wallets:

Using the credentials of the offline world, we typically stock our information in a wallet. In order to keep them protected by keeping them close to us; and easily have access to them. The job of a digital wallet does not differ as much as the role of the real world wallet. The main purpose of the wallet is to store the credentials, protect the information from being theft and keep them available for the holder to use them. However, some key differences are included, an SSI wallet should implement an open standards concerning self-sovereign verifiable credentials and sensitive data. In addition, the wallet should have an agent, which is the software that the users manipulate to process verifiable credentials and DIDs.

It is a third key component of SSI, along with verifiable credentials and DIDs, is the software that you use to process verifiable credentials and DIDs.

Trust over IP (ToIP):

ToIP is a set of protocols being developed to enable a layer of trust on the Internet, protocols embodied in Indy, Aries and Ursa. It includes self-sovereign identity in that covers identity, but goes beyond that to cover any type of authentic data.

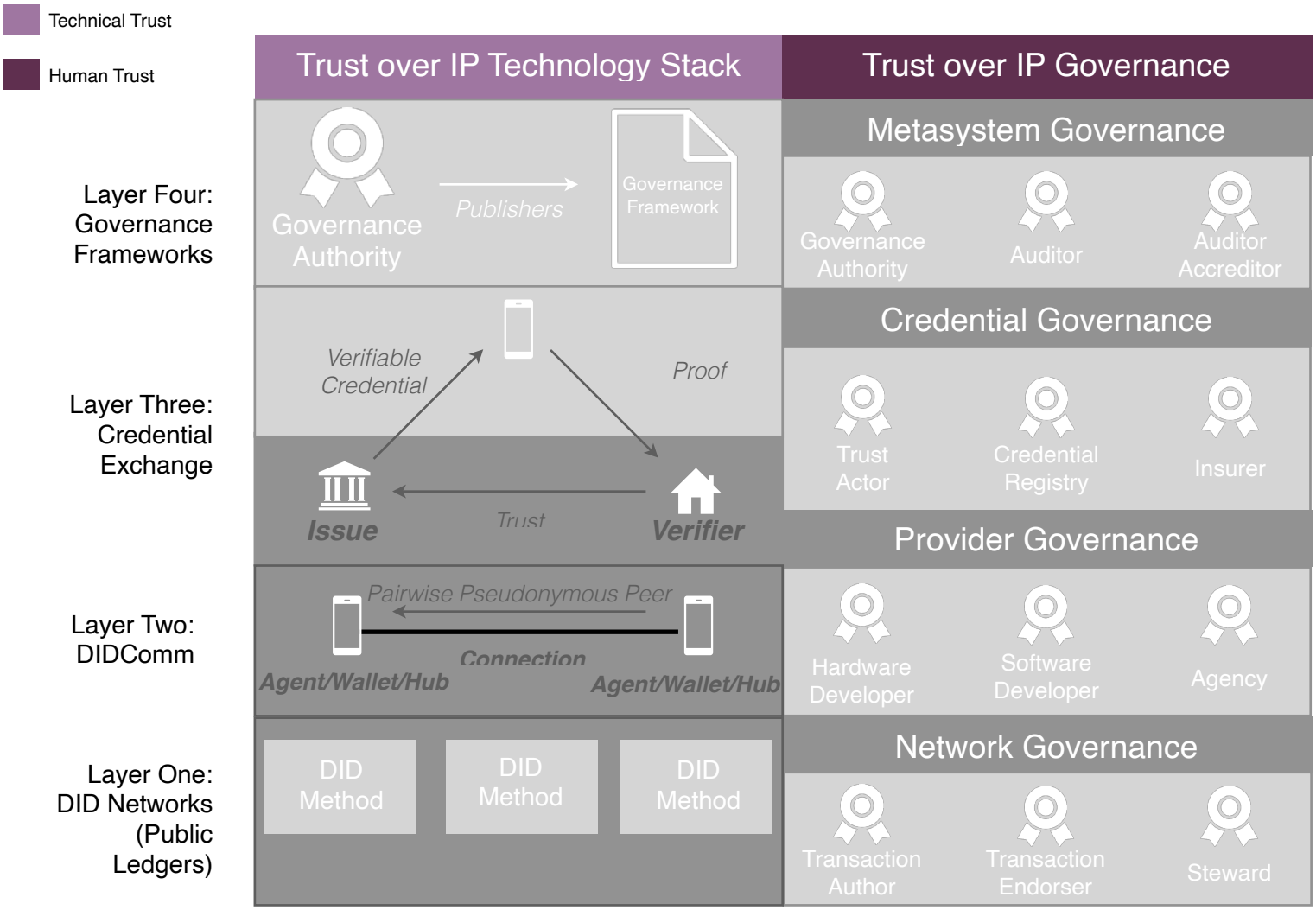


Figure 10: Trust Over !P Technology Stack

At the bottom left, the foundation of the stack are public blockchains that store decentralized identifiers (DIDs) and (in some cases) other data necessary for the higher layers. Next are agents and the protocols that enable agents to establish connections and exchange messages (information). The next layer up is the verifiable credential layer that enables trusted information flow (data cryptographically signed by the participants during issuing and verified during presentation) [24].

To summarize, Verifiable credentials, DIDs and agents are at the core of self-sovereign identity. They can solve many problems that were addressed in the first section.

6. LEVERAGING HYPERLEDGER INDY, URSA, ARIES

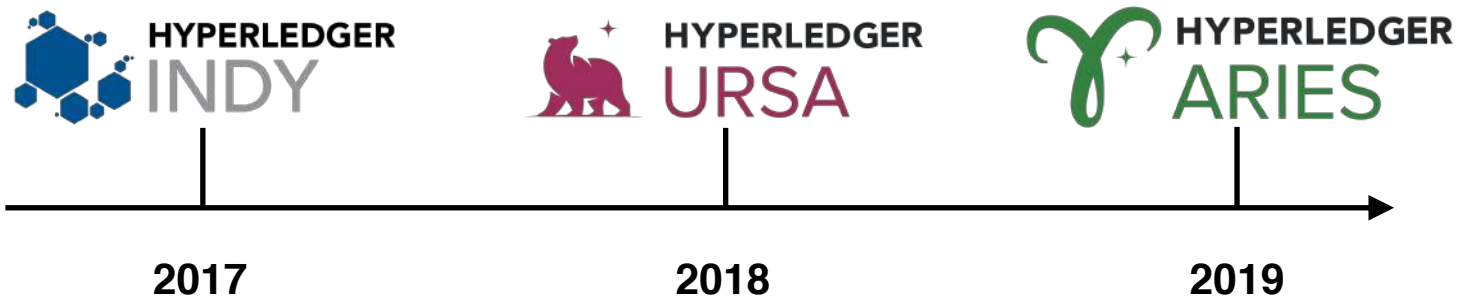


Figure 11: Hyperledger Identity History

- Hyperledger Indy joined the Hyperledger community in 2017. It was the first Hyperledger identity-focused blockchain framework. Hyperledger Indy includes verifiable credentials based on zero-knowledge proof (ZKP) technology, decentralized identifiers, permissioned distributed ledger, and a software development kit (SDK) for building agents.
- Next to the Hyperledger Indy, the Hyperledger Ursa was created in 2018. This project was to mitigate the indy-crypto code repository out if Indy into a new project “Ursa”. This new project gives cryptologists a place to work together to build and package cryptographic primitives avoiding duplicating cryptographic work and increasing security [25].
- In mid-2019, a variety of organizations met to demonstrate interoperability across a set of independently developed agent implementations. Hyperledger Aries presented a toolkit designed for initiatives and solutions focused on creating, transmitting, storing and using verifiable digital credentials. At its core are protocols enabling connectivity between agents using secure messaging to exchange information. Aries is all about peer-to-peer interactions between agents controlled by different entities—people, organizations and things. Using the standardized messaging channel, verifiable credentials can be exchanged based on DIDs rooted in different ledgers (based on Hypeledger Indy or other technology) using a range of verifiable credentials implementations.

6.1 Hyperledger Ursa

Hyperledger Ursa project was created so that all the Hyperledger platforms can benefit from a single powerful cryptographic library [25]. Before this library was developed, every Hyperledger platform used a different library.

The packages Ursa which are used by Indy and Aries for all of the uses of cryptography, include:

a) Generation of public/private key pairs

b) Data encryption and decryption

c) Data signing and verifying

d) Data hash generation and verification

e) Zero-knowledge proof (ZKP) technology, including issuing ZKP credentials and generating and verifying ZKPs.

With the Ursa packages rooted in Indy and Aries and all other Hyperledger projects, cryptographic features are easy to implement. When a write transaction is sent to an Indy ledger, it is signed via a call to an Ursa function by the transaction author and the signature is verified also by an Ursa function by the nodes. In addition, messages are encrypted from one Aries agent to another and packaged via Ursa. In this report I will only cover this paragraph about Ursa, since we are building applications on top of Aries and using an Indy ledger and all the functions of Ursa are hidden from the development, only calls from the library will be made.

6.2 Hyperledger Indy

Indy is the first Hyperledger Identity system relying on decentralization and it is the core component of this technology. Indy provides code that implements the public distributed ledger technology (DLT). Indy architecture is built on the concept of self-sovereign identity.

It is consisted of:

- Indy-sdk: a software development kit that enables Indy clients/agents to interact with the Indy blockchain and to keep track of the keys and other identity-related data. Some components are being moved to Aries.
- Indy-node: the DLT component of Indy

The indy blockchain plays a big role on the internet to develop trust. Since Indy is focused only on identity, it does not support the concept of assets exchange nor the smart contract concept. Instead, the nodes work together to agree on what transaction should be written on the ledger. In addition, agents that need to write transaction on the ledger must prove they are authorized to write on that ledger and write that specific transaction.

The Indy is a permissioned and public blockchain. It means that everyone can see the contents of the blockchain, but only pre-approved participants are permitted to participant in the validation process called stewards. In some instances, Indy could be run as a private network it depends on the way of the implementation and the business needs.

Indy uses a special BFT called Redundant Byzantine Fault Tolerance (RBFT).

Furthermore, according to the GDPR no private data should be exposed. The elements placed the blockchain are:

- **Public DIDs**: when the issuer wants to issue a credential, they must have a DID on the blockchain which will allow the verifier to find out who they are, and whether or not their credential should be trusted or not.
- **Schemas**: the issuer can put his schema on the blockchain. The attributes names and types for example.
- **Credential definitions**: Before an issuer can produce a credential using a schema, they must put a credential definition on the blockchain. A credential definition declares, that the issuer is using a DID and this DID is going to use a specific schema defined by the issuer to issue credentials. And that specifics public keys are used to sign the claims and issue credential.
- **Revocation Registry**: the issuer may want to revoke a credential; and if they do, they must write a revocation registry to the ledger before issuing credentials. The revocation registry links back to the credential definition, and allows the issuer to revoke his own credentials, independent of the holders. Nobody can have a look at the revocation registry to determine if a particular credential has been revoked.

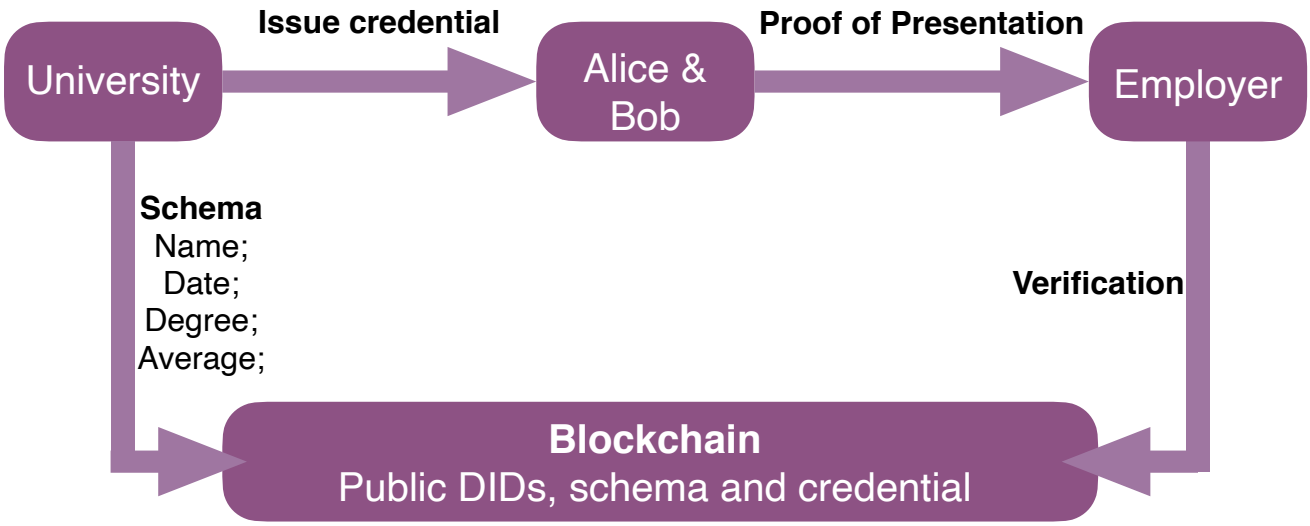
6.3 Hyperledger Aries

Hyperledger Aries consists of a decentralized kit designed for creating, transmitting and storing verifiable credentials. It Implements the decentralized key management incubated by Indy. In addition, Aries implements ZKP (zero knowledge proof) using Ursa.

The Aries architecture defines how different implementations are pluggable into an agent to support Indy and other DID and verifiable credential implementations at the same time.

One of purposes of developing Aries is to support portability. The person using this technology will have to use a digital agent to collect valuable data. Since those agents are dedicated software, they will be at some point an obsolete or better software will be discovered. Part of the Aries is about data storage and the export/import data from one agent to another without losing any information.

6.4 Use-case description



The use-case consists of four agents. Two of these agents are organizations, one is a University and the other is an Employer. In addition, two graduates from the University named Alice and Bob. The Employer have a job opening requiring a learning certificate, where the candidate should have a specific average.
Alice and Bob needs credentials from the University, so they can prove and send presentation to the employer that they meet the education requirements. To issue a credential the University needs to have a public DID registered to the Ledger, because it should be a trusted entity and the employer can then prove that it is really the specific University that issued that credential. Also, the university creates a Schema named degree schema and a credential definition based on the schema registered to the Ledger. The Schema is composed of four attributes: name, date, major and the average. The Employer has a public DID registered into the Ledger.

Alice and Bob do not have a public DID. However, private DIDs are generated between Alice/Bob and their University. Furthermore, private DIDs are generated between Alice/Bob and the employer.

6.5 Implementation

The environment consists of an Indy test network; we used the Von Organization Network (VON) as an Indy sandbox. Agents are deployed into Docker and Docker composer in order to have for each one, different virtual location. All the agents were built on top of the Aries Cloud Agent-Python (ACA-Py) framework. The University controller was built using the C# language. As for Alice and Bob controllers, we used Python, and finally OpenApi which is a manual built-in controller to manipulate the employer entity.

6.6 VON Ledger

On the VON ledger, there is one trustee, which will give authorization to write on the ledger. The transactions written on the blockchain are:

- The public DID of the University which is signed by the trustee of the ledger. All the DID written on the Indy ledger are globally unique, this feature is handled by the Indy ledger.

- The endpoint of the Employer and the University are also registered.
- The issuer schema, signed by the University. The attributes include: average, name, degree and date.
- The Credential definition related to the issuer schema.
- The Employer public DID signed by the trustee.

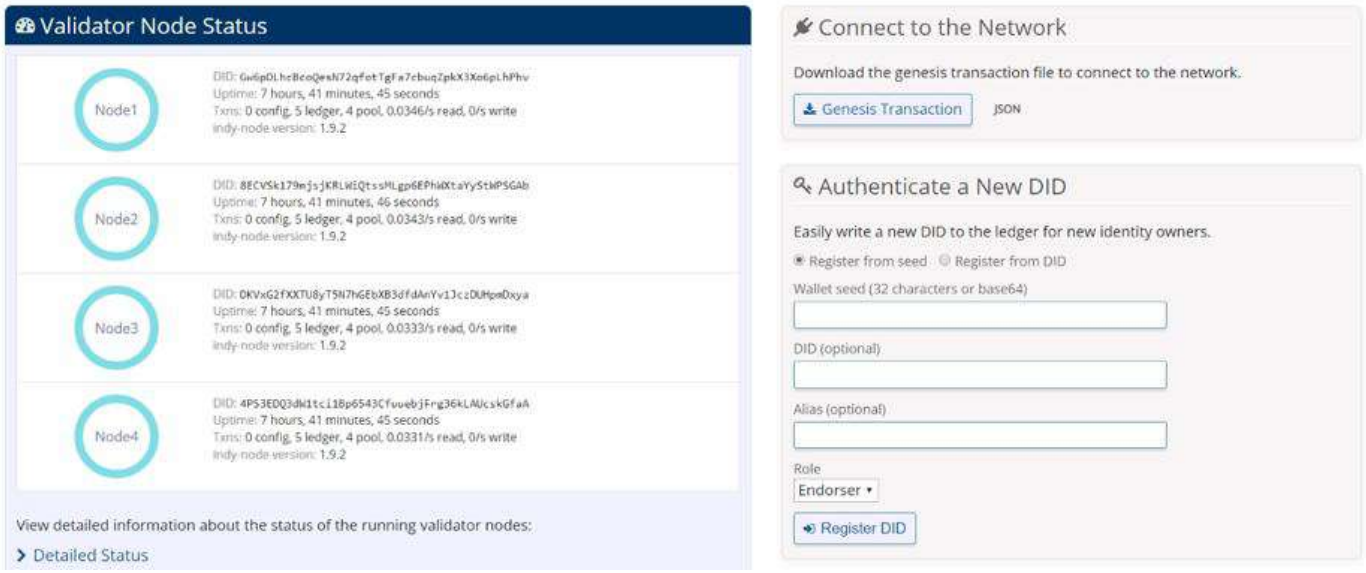


Figure 12: VON Network Web Interface

6.7 Credential Issuance

Before issuing a credential, it is mandatory to establish a secure connection between the entities.
We have two roles: the inviter and the invitee. The inviter is the agent that initiate the protocol in our case it is going to be the University where it will generate an invitation message. The first message to be an out-of-band communication, because we do not know the endpoint of the second entity. The data can be exchanged by mail, or pre-shared. To generate this invitation, we can use the public resolvable DID or the service endpoint, recipient keys and its label. All this information is used to create a provisional connection to the inviter. This connection will be made finale after the connection_response message. In the invitation we have the type of the protocol used, the id of this invitation, a recipient key so the other agent can send the response in an encrypted way and the endpoint of our agent. From the invitation response, we will get for each participant a private DID and a connection id.

Figure 13: The University Public DID Written on The Blockchain

Figure 14: Schema Written on The Blockchain

The University will now issue the credentials of its student. “This is the basis of interoperability between Issuers and Holders”. In order to issue credentials we need the connection_id, schema, schema credential and the credentials.

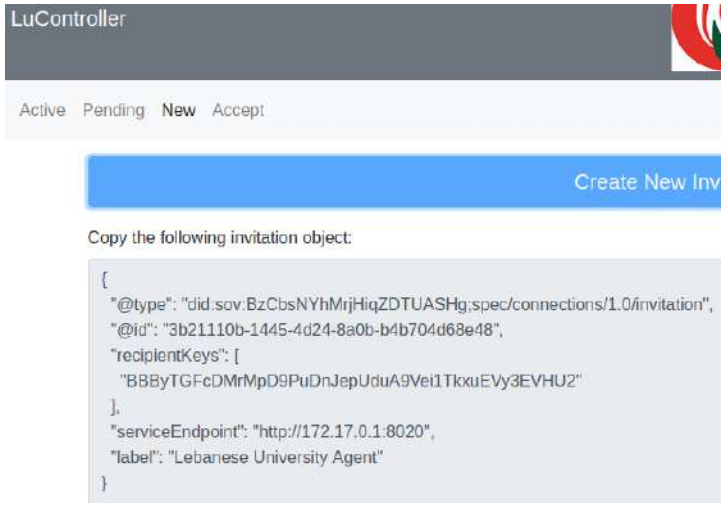


Figure 15: University-Alice request invitation

The students will store these credentials in their wallets, and a metadata is generated to keep the information secret.

Figure 16: Employer Public DID

6.8 Presentation Proof of Request

Several phases are required to complete the Proof presentation.
First, a connection between Alice and the Employer is needed, also a connection between Bob and the Employer is required. Same steps as above to connect between the agents.
Alice and Bob sent their resume to the Employer.
The Employer decides to ask Bob and Alice for their degree from the University. The business refers to the blockchain, in order read the credential definition of the university and the schema attributes used.

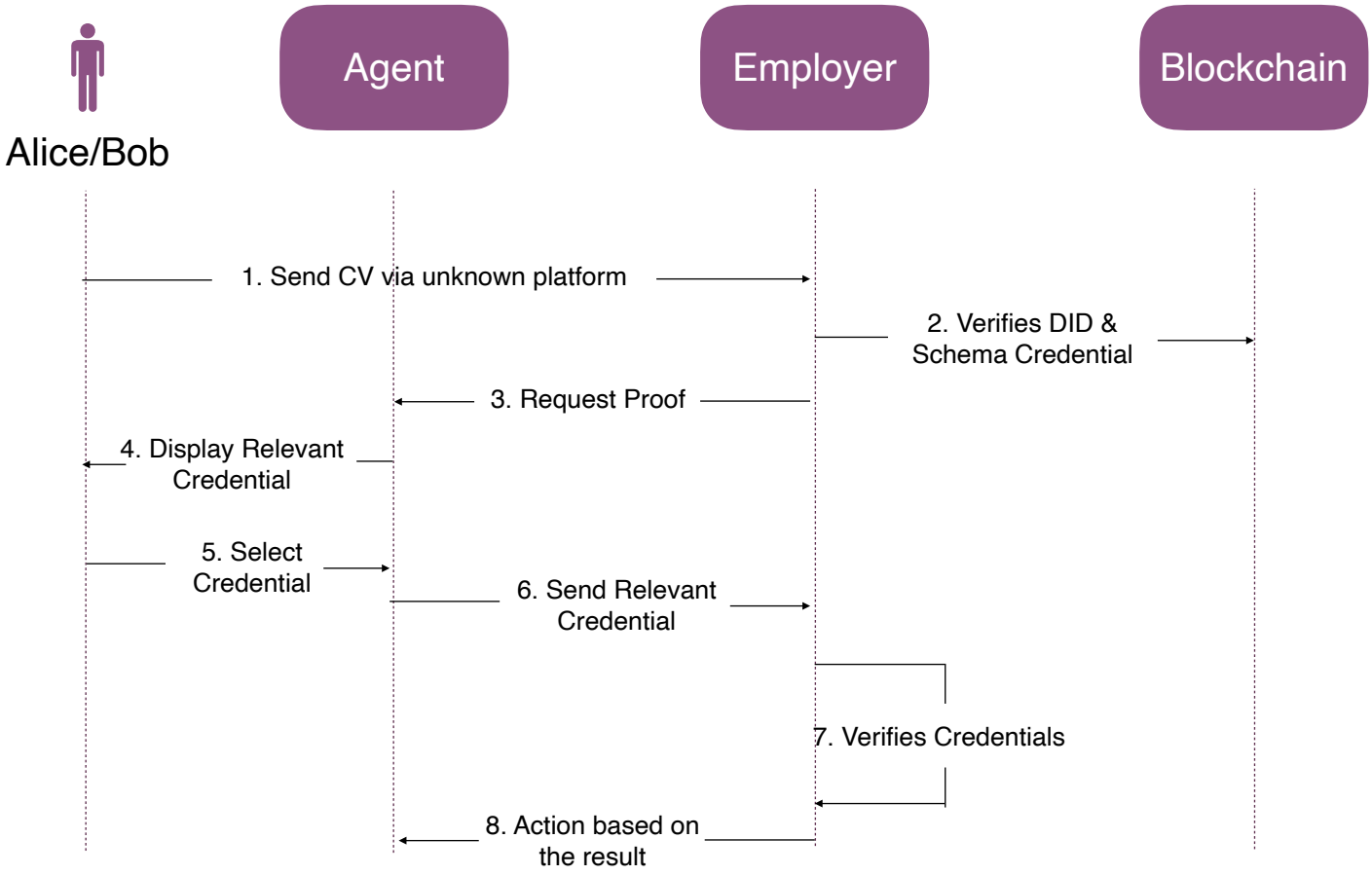


Figure 17: Proof of Request/Presentation

The Employer will then send a JSON request (all the request and messages are send in a JSON form) asking for the name, the date, the degree …
The Employer sends a request of proof containing the required attributes and restrictions. The first restriction is that all the attributes should be generated with the credential_definition_id of the corresponding university. Second, the average attribute should have a value greater or equal to needed.
After sending the requests, the controller of the students check the wallet to find if there are any credentials that satisfies the proof request. Then generate and send the proof to the verifier.
The Employer verifies the results and finds that Alice conforms all the restrictions, without disclosing any information about her average. In addition, the verification of Bob proof will return a null response; also, Bob's average will not be visible to the Employer.

Figure 18: Alice Verification Response

To summarize, on the ledger sufficient information is present, containing the public DIDs, in order to establish trust between same ledger entities. Personal data are not published; they are only transferred over a secure end-to-end communication. The holders can get and store their personal data. The issuers and verifiers are not directly connected. It means in the same ledger we can build trust between nodes. Using the zero knowledge proof, some attributes are proved without disclosing any information about its value.

7. Conclusion

The Blockchain unifies the user experience related to the authentication process through its use for personal and professional services ensuring better understanding and control on the user side. The information distributed is recorded on a medium which the user controls sharing with the various services. Users thus retain complete control over the use of their data. And administering their own identity and their preferences in terms of consent in a portable manner across all platforms that would interest them.

The modalities for setting up a robust and transparent system, in agreement with the trusted authorities and regulations still need to be standardized from a technical point of view. Ultimately, however, SSI represents an unlimited opportunity for any sector. Without question, this unified design represents the future in terms of identity management and personal data governance.

This personal sovereignty of identity with consent based on a clear vision of the usages of information and enabled by a simple and fluid user experience, will lead to a revolution in several models and sectors.

8. References

[1] M. Gupta, Blockchain for dummies, John Wiley & Sons Inc, 2020. .

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[3] J. Moubarak, E. Filiol and M. Chamoun, "Comparative analysis of blockchain technologies and tor network: Two faces of the same reality?," in *2017 1st Cyber Security in Networking Conference (CSNet)*, 2017.

[4] J. Moubarak, E. Filiol and M. Chamoun, "On blockchain security and relevant attacks," in *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2018.

[5] R. Bdiwi, C. De Runz, S. Faiz and A. A. Cherif, "A Blockchain Based Decentralized Platform for Ubiquitous Learning Environment," in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, 2018.

[6] J. C. Farah, . A. Vozniuk, M. J. Rodríguez-Triana and D. Gillet, "A blueprint for a blockchain-based architecture to power a distributed network of tamper-evident learning trace repositories.," in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, 2028.

[7] R. Arenas and P. Fernandez, "CredenceLedger: a permissioned blockchain for verifiable academic credentials," in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018.

[8] M. Han, J. H. Zhigang Li, Y. X. Dalei Wu and A. Baba, "A Novel Blockchain-based Education Records Verification Solution," in *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, 2018.

[9] N. Bore, S. Karumba, J. Mutahi, S. Solomon Darnell, C. Wayua and K. Weldemariam, "Towards blockchain-enabled school information hub.," in *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development*, 2017.

[10] I. B. Bandara, F. Ioras and M. P. Arraiza, "The emerging trend of blockchain for validating degree apprenticeship certification in cybersecurity education," 2018.

[11] M. Turkanović, M. Hölbl, K. Košič, M. Heričko and A. Kamišalić, "EduCTX: A blockchain-based higher education credit platform," 2018.

[12] S. Gilda and M. Mehrotra, "Blockchain for Student Data Privacy and Consent," in *2018 International Conference on Computer Communication and Informatics (ICCCI)*, 2018.

[13] Y. Xu, S. Zhao, L. Kong, Y. Zheng, S. Zhang and Q. Li, "ECBC: A high performance educational certificate blockchain with efficient query.," in *International Colloquium on Theoretical Aspects of Computing*, 2017.

[14] M. Sharples and J. Domingue, "The blockchain and kudos: Adistributed system for educational record, reputation and reward," in *European conference on technology enhanced learning*, 2016.

[15] J. Nespor, "Cyber schooling and the accumulation of school time," 2019.

[16] "Hyperledger Architecture, Volume 1 - Introduction to Hyperledger Business Blockchain," Hyperledger Architecture Working Group (WG).

[17] A. Chebelaine, "IBM Hyperledger".

[18] [Online]. Available: https://hyperledger.github.io/composer/v0.19/introduction/introduction.

[19] "Architecture explained," [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.3/arch-deep-dive.html.

[20] W3C, "Verifiable Credentials Data Model 1.0," 19 November 2019. [Online]. Available: https://www.w3.org/TR/vc-data-model/#presentations.

[21] C. Allen, 25 April 2016. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[22] W3C, "Sovrin DID Method Specification," 19 August 2020. [Online]. Available: https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html#sovrin-did-method.

[23] H. Anwar, November 2018. [Online]. Available: https://101blockchains.com/zero-knowledge-proof/.

[24] g. community. [Online]. Available: https://github.com/hyperledger/aries-rfcs/tree/master/concepts/0289-toip-stack.

[25] 2020. [Online]. Available: https://www.hyperledger.org/use/ursa.

potech
LABS

ABOUT US

Part of P.O.TECH - Paths of Technology, Potech Labs is the Research & Development laboratory arm of the group.

Potech Labs focuses on improving the value and use of evidence for learning strategies and avenues of innovation, as well as academic research, exchange programs, and capacity building projects.

Potech Labs engages in the promotion of scientific research through the creation of businesses-universities collaborations and contractual research activities.

Internships, research projects, PhD/Master thesis supervision, and recruitment opportunities in partner companies is possible through the establishment of framework agreements and the participation in integrated collaboration initiatives in the Middle East and Europe.

Potech Labs scientific publications have been recognized and accepted in leading journals and international conferences.

www.potech.global
info@potech.global
Berytech Technological Pole
Dekwaneh - Lebanon
+961 4 533040 Ext: 4009-4010

@potechglobal
@potechglobal
company/potechglobal
P.O.TECH - Paths of Technology
@potechglobal