People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Tebessa University


Faculty of Exact Sciences and
Natural and Life Sciences

Department of Mathematics and informatics

Thesis
Presented with a view to obtaining the MASTER's degree

**Field**: (Mathematics/Informatics)

**Speciality**: network and security

**By**

Taleb Manel

# Preserving Privacy of Users Data Using Blockchain and Respecting the Algerian Regulation of Data Protection
## Case Study: University of Tebessa

Date: 15 June 2022

**Jury:**

| | | | |
|---|---|---|---|
| Hakim Bendjenna | Professor | University of Laarbi Tebessi | President |
| Nait Med Cherif | MCB | University of Laarbi Tebessi | Examinator |
| Djedai Ala | MCB | University of Laarbi Tebessi | Reporter |

**2021/2022**

# ACKNOWLEDGMENTS

*First of all, thanks for God*

*I thank my supervisor Mr. JEDDAI ALLA for guiding me throughout my work, with hoim effort and valuable advice.*

*I thank the members of the jury, the President Pr. BENDJANNA H. and the examiner Mr. NAIT C. for agreeing to discuss our end of study project to benefit from their experience and advice.*

*Finally, I thank all the teachers of the Department of Mathematics and Computer Science*

# DEDICATION

*To*
*my husband,*
*my children,*
*my family,*
*my beautiful family,*
*and my neighbors.*

# Summary

# Liste of figures

# Liste of tables

# Abstract

In recent years, there is an increasing use of various types of applications in many fields such as government, smart cities, health, social networks...etc. In this situation, the amount of user data has been increased and hence new protocols and strategies need to be put in place to manage and process this big data. One of the biggest issues that data can face is the privacy and security of its use by apps and other parties. Thus, maintaining the confidentiality and security of data and using it reliably is the main goal of many research directions. Blockchain technology, which is a secure and decentralized database, can be used to solve this problem to keep user data private.

In this thesis we present an approach based on the use of blockchain to maintain confidentiality and data with respect to the Algerian Regulation of Data Protection based on Bloackchain.

**Keywords: Blockchain, Smart contract, Privacy, Security, Services.**

# Résumé

Ces dernières années, il y a une utilisation croissante de divers types d'applications dans de nombreux domaines tels que le gouvernement, les villes intelligentes, la santé, les réseaux sociaux...etc. Dans cette situation, la quantité de données utilisateur a été augmentée et il faut donc mettre en place de nouveaux protocoles et stratégies pour gérer et traiter ces mégadonnées. L'un des problèmes les plus importants auxquels les données peuvent être confrontées est la confidentialité et la sécurité de leur utilisation par les applications et les autres parties. Ainsi, maintenir la confidentialité et la sécurité des données et les utiliser de manière fiable est l'objectif principal de nombreuses directions de recherche. La technologie blockchain, qui est une base de données sécurisée et décentralisée, peut être utilisée pour résoudre ce problème afin de préserver la confidentialité des données des utilisateurs.

Dans cette thèse nous présentons une approche basée sur l'utilisation de la blockchain pour maintenir la confidentialité et la sécurité des données en respectant la loi algérienne pour la protection des données personnelles.

**Mots clés :** Blockchain, Smart contract , Confidentialité , Sécurité , Services.

# الملخص

في السنوات الأخيرة ، هناك استخدام متزايد لأنواع مختلفة من التطبيقات في العديد من المجالات مثل الحكومة والمدن الذكية والصحة والشبكات الاجتماعية ... إلخ. في هذه الحالة ، تمت زيادة كمية بيانات المستخدم ، وبالتالي يجب وضع بروتوكولات واستراتيجيات جديدة لإدارة ومعالجة هذه البيانات الضخمة. من أكبر المشكلات التي يمكن أن تواجهها البيانات هي خصوصية وأمان استخدامها من قبل التطبيقات والأطراف الأخرى. وبالتالي ، فإن الحفاظ على سرية وأمن البيانات واستخدامها بشكل موثوق هو الهدف الرئيسي للعديد من اتجاهات البحث. يمكن استخدام تقنية Blockchain ، وهي قاعدة بيانات آمنة ولا مركزية ، لحل هذه المشكلة للحفاظ على خصوصية بيانات المستخدم.

في هذه الأطروحة نقدم نهجًا يعتمد على استخدام blockchain للحفاظ على السرية وأمن البيانات.

**الكلمات الرئيسية**: Blockchain ، عقد ذكي ، خصوصية ، أمن ، خدمات.

# General Introduction

# 1. Introduction

In recent years, there is an increasing usage of various types of applications in many domains such as government, smart cities, healthcare, social networks...etc. In this situation, the amount of user data has been increased and therefore it must put new protocols and strategies to manage and process this big data. One of the most problems that can face the data is the privacy and security of its utilization by the applications and other parties. Thus, keeping the data privacy and security and using it in trust manner is the main goal of many research directions.

There is a growing public concern about user privacy. Centralized organizations - both public and private, amass large quantities of personal and sensitive information. Individuals have little or no control over the data that is stored about them and how it is used.

Blockchain provides new mechanisms, such as decentralized identities and zero-knowledge proofs that enable data to be shared in ways that maintain the privacy of the individual and allow users to maintain control over their own data. These advances can provide both increased cyber security and more ethical use of personal data. Blockchain participants can realize these outcomes through careful development of governance frameworks and mechanisms.

In this thesis, we focus of the regulation of data protection of our country Algeria in order to support the government direction for protection of the personal data.

# 2. Objectives

The main objective of this master thesis is using the blockchain for keeping the privacy and security of the data with respecting the Algerian regulation of data protection. Using blockchain technology to protect the personal data and its privacy

- ☑ Respecting the Algerian regulation of data protection.

# 3. Thesis structure

This thesis is organized as follow:

- ☑ **Chapter 1:** Present a State of the art about the blockchain technology and the related works
- ☑ **Chapter 2:** Describe the Algerian regulation of data protection
- ☑ **Chapter 3:** Present our contribution which is the main model that use blockchain technology to keep data privacy.
- ☑ **Chapter 4:** Describe the Hyperledger fabric and our Implementation & Evaluation

Finally, we conclude with a general conclusion and perspectives


# STATE OF THE ART

# 1. Introduction

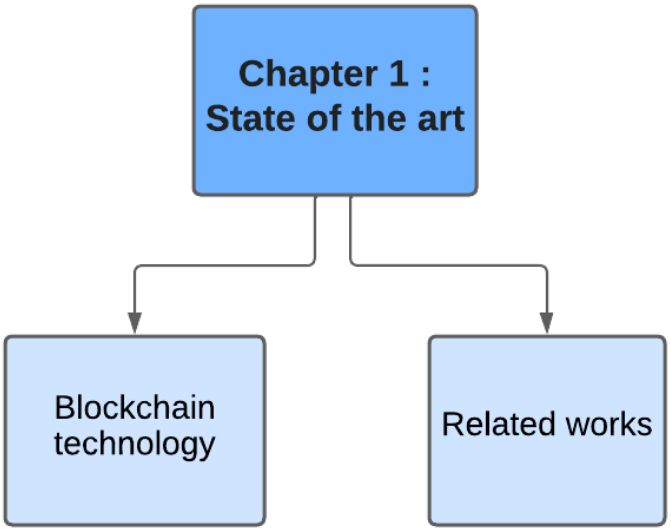
The security of data on the Internet has always been a contentious issue: since its inception, there hasn't been a day when the media hasn't reported on the hacking of bank records or massive e-commerce sites. This is Why the researchers to concentrate today on thetechnology of encryption and of securing from data like the blockchain.

The reputation of chain of blocks grows up of day in day and attracted of more in more careful at the scaleworld. Some people compare blockchain to the early internet in the 1970s and some call it the web 2.0 revolution, and it is talked about by many people in different disciplines: economists, programmers and others.

This chapter is devides into two parts:

In the first part we lets go introduce the principles and the concepts of based of the technology blockchain, their architecture and their mechanism and finally the applications of the technology to deferent scales.

In the second part, we present the related work about the blockchain application in the field of data privacy.



# Part 1: Blockchain technology

## 2. History

Before we get into the blockchain technology, let's take a look at how it came to be:

In 1991, Stuart Haber and W. Scott Stornetta envisioned what has become known as blockchain. Their first project was to create a cryptographically safe chain of blocks that would prevent anyone from tampering with document timestamps. They modified their system in 1992 to include Merkle trees, which increased efficiency and allowed them to collect more papers on a single block. However, because to the activity of one person or group known as Satoshi Nakamoto, Blockchain History begins to take prominence in 2008.

Blockchain technology is credited to Satoshi Nakamoto as the brains behind it. Persons believe Nakamoto could be a person or a group of people that worked on Bitcoin, the first application of digital ledger technology, but very little is known about him.

In 2008, Nakamoto created the original blockchain, from which the technology has evolved and found its way into a wide range of uses outside of cryptocurrencies.

In 2009, Satoshi Nakamoto published the first whitepaper on the technology. He explained in the whitepaper how the technology was perfectly suited to enhancing digital trust because to the decentralization component, which meant that no one would ever be in control of anything.

The digital ledger technology has grown since Satoshi Nakamoto left the scene and turned over Bitcoin development to other core developers, resulting in new apps that make up the blockchain History.

## 3. Definition

A blockchain is a computer science technique "open source », of digital data storage and transmission based on peer-to-peer (P2P) exchanges, in a chronological order, horizontally, transparently, decentralizedly, without middlemen [1], and secured by consensus methods. By extension, it is a public database that includes the history of all transactions conducted between its users since its decentralized formation, is dependable, inviolable, and decentralized, and is arranged into known sub-registers under the name "block [4].

As mathematician Jean-Paul Delahaye [2] described it, it can be compared to a distributed ledger (DLT) of anonymous accounts "a very enormous notebook, which everyone can read for free and write about, but which is impossible to write about in the East.

## 4. Architecture of blockchain

The Blockchain is a series of blocks that contain a comprehensive list of transaction records, such as the big delivered public classic. As shown in figure 2:

- A block has just one parent block (hash) if the block header contains a preceding block hash.
- A "Genesis block" is the initial block of a blockchain that does not have a parent block [5].

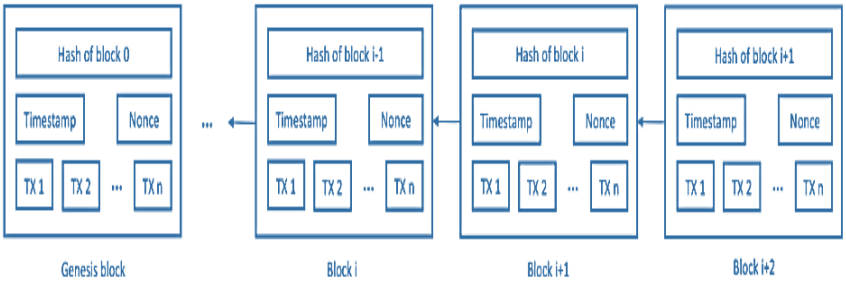



**Figure 1:Structure from blockchain [5]**

## 5.1. Architecture of block

The blocks record the transactions that take place in the blockchain system. Figure 3 shows a block with a header and a body.

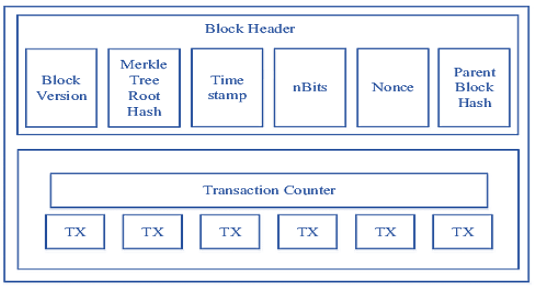



**Figure 2:The structure of the block [6]**

The metadata for the block is contained in the header (see Table 1). The details from transactions in the structure of the merkel tree are mostly included in the body.

| Name | Description |
|---|---|
| On your mind of block | Includes the fields of the header of block describe in the picture following. |
| Counter | The field contains the number total of transactions in the block. |
| Transactions | All the transactions in the block. |

**Table 2:The structure of one block [7]**

| Name | Description |
|---|---|
| Version (4 bytes) | The number version from block who dictate them rules ofvalidation of block to be continued. |
| Hash previous (32 bytes) | This is a double SHA256 hash of the block headerprevious. |
| Root of merkle (32 bytes) | This is a double SHA256 hash of the tree of selection of all the transactions included in the block. |
| Timestamp (4 bytes) | this is when the miner started hashing the en-head |
| Difficulty (4 bytes) | This is the target of difficulty from block. |
| Nonce (4bytes) | He is of one number arbitrary than the minors modify atrepeatedly in order to produce a hash that fulfills the threshold of difficulty. |

**Table 3:The structure from body of block [7]**

## 5. Process of Blockchain

Every blockchain operation necessitates token (ex :bitcoin ) .Blocks are made up of transactions conducted between network members(Nodes). Each block is validated by "miners," or network nodes, using processes that vary depending on the type of blockchain. These techniques are known as consensus algorithms, and they include problem-solving algorithms such as "Proof-of-Work" and "Proof-of-Stake."

The block is timestamped and posted to the blockchain once it has been confirmed.

The transaction is then accessible to both the sender and everyone else on the network.

The following diagram depicted the several processes taken by technology to allow a user A to conduct a transaction with a user B: [8]

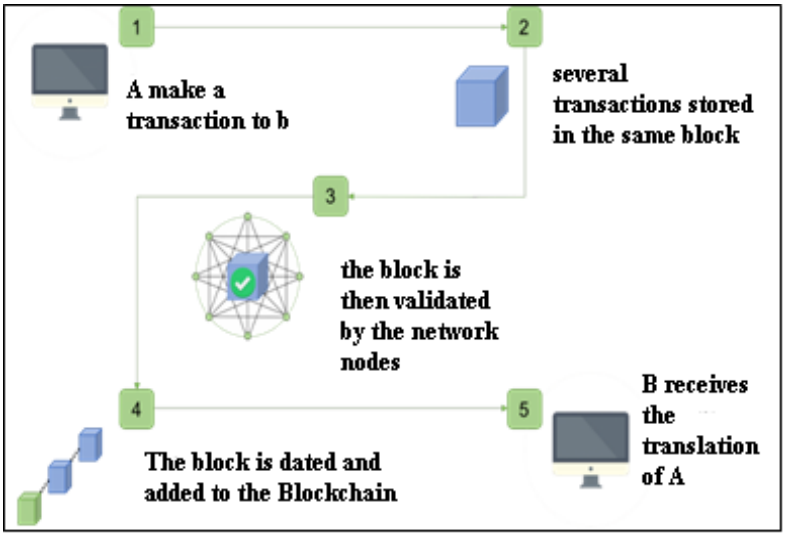



Figure 3:Functioning general of the blockchain

1. The peer (Node) : A blockchain node is a computer that is connected to the network. As a result, each node represents a user. This one preserves a copy of the blockchain ledger at all times and may be disseminated anywhere on the planet.
2. The P2P network: The blockchain is built on a peer-to-peer network that consists of a single set of interconnected nodes. Because there is no central authority in this network, it is completely decentralized. The registration blockchain is accessible to everyone in this network.
3. The block: is a record of the network's most recent transactions. When a transaction is completed, a new block East is produced to save the new transaction, and the network validates the completed block.
4. The register: is a chain of blocks (which include all transactions) shared by all network users. To put it another way, the registry is a database that categorizes all transactions and makes them available to all network users.

# 6. Consensus methods

A consensus is described as a broad and unanimity of opinion among people on a set of instructions. It is an IT consensus where people agree on a procedure for confirming transactions and updating data based on blockchain technology.

## 6.1.Proof of work

Proof of Word (PoW) is a technology designed to combat cyberattacks such as distributed denial of service assaults (DDoS).

On PoW, each node in the network uses one puzzle mathematical to compute the hash value of the block header. This brain teaser is mathematical in nature and focuses on symmetry. The nonce minors in the block header would vary often to provide different hash values. Consensus does not require that the estimated result be equal to or lower than a certain value.

When a node reaches the desired value, the block is broadcast to all other nodes, who must all validate that the hash value is correct. Other miners will add this new block to their own blockchain if the block is validated. Minors are the knots that calculate the hash values, and the PoW East operation is known as extraction in Bitcoin. When numerous nodes obtain the necessary nonce at almost the same moment in a decentralized network, legitimate blocks can be generated concurrently. As a result, it is possible to produce branches. [10]

## 6.2.Proof of stake

Proof of Stake (POS) is a method of validating blocks and registering them in a blockchain that uses far less energy than Proof of Work (POW). It's a "virtual mining" because no powerful hardware is required. To take part in a Proof-of-Stake, you'll need:

For the construction and validation of new blocks, you must own a particular amount of cryptocurrency (tokens). Invest in the purchase of bitcoin and then keep it in the cryptocurrency's official wallet to become a part of the network.

The system chooses a " validator" at random based on the previous block on the blockchain. This person will have the authority to generate and validate the following block.

The higher this number is, the more likely the user is to validate the block. In this situation, the phrase mining is replaced by minting. If the block isn't made within a certain amount of time, a second individual is chosen, and so on. When the block is genuine, you win the reward matches at the expense of the transactions in the block [10]

# 7. Characteristics of Blockchain

The following 6 attributes keys are presented on the blockchain:

1. **Decentralization.** There is no third party to rely on. Two persons can make a transaction and rely on the system to confirm the transaction. Furthermore, the network as a whole allows access to the data base and the operations that are carried out there.
2. **Data security.** The Blockchain is built to store data in an immutable and unalterable manner. The decentralized nature of blockchain and encryption techniques makes it far more difficult for malicious users to shoot their way out of the system.
3. **Transparency.** Even if the participants use pseudonyms, their transactions may be tracked. The history of transactions may be viewed by all members networks at any moment, making the system transparent.
4. **Automated.** The blockchain's members' pre-determined regulations are carried out by computers with pre-programmed instructions. Contracts that are "smart" will be self-executing.
5. **Immutability.** It is nearly impossible to fabricate because each transaction broadcast over the network must be confirmed and recorded in blocks dispersed around the network. In addition, other nodes would validate each broadcast block, and the transactions would be verified. As a result, any falsification would be clearly detectable.
6. **Authenticity.** Since the blockchain saves each transfer of an object, asset, document, property, or contract in the database, it is authentic. In addition, he can add the time, day, year, and owner to each transfer.

# 8. The categories of Blockchain

## 8.1. Public Blockchain

This is the most well-known model (Permissionless), which is an open registry accessible to anyone in the world with no authorisation or authentication requirements. Where can I participate in a consensus process, such as Ethereum.

## 8.2. Private Blockchain

There are totally closed (permissioned) blockchains in which write access is allowed by a centralized institution (for example, a central bank), but read permissions can be public or private [1]. In general, network nodes are authenticated and permitted based on predetermined criteria, such as the Hyperledger.

## 8.3. Hybrid Blockchain

Is a blockchain that combines public and private blockchains. Although she has the same scalability and privacy protection as private blockchain, the primary distinction is that instead of a single entity, a group of nodes called leader nodes is chosen to verify transaction processes. This enables a decentralized design in which leader nodes can award authorization to other users [3].

# 9. Smart contract

Nick Szabo proposed the concept of smart contracts in 1994, but it took another 20 years for the actual potential and benefits of them to be realized.

"A smart contract is a protocol of transaction computerized who performed the conditions of one contract," Szabo says of smart contracts. General objectives include meeting contractual obligations, minimizing harmful exceptions, and limiting the use of trusted intermediaries. The minimization of losses associated with fraud, the expenses of arbitration and application, as well as other transaction costs, are all economic objectives.

## 9.1. Smart contract characteristics

Smart contracts are self-verifying, self-executable, inviolable, secure, and unstoppable, and they can track performance in real time.

## 9.2. Smart contract process

Smart Contracts add to the usefulness of the blockchain by simulating real-world circumstances with complete Turing high-level programming languages. These contracts can be written in the Ethereum blockchain's own set of programming languages. Smart Contracts increase the blockchain's usefulness by simulating real-world concepts in the blockchain. The smart contract code cannot be changed once it has been deployed.

A smart contract's execution is also considered a transaction. To run a smart contract, a gas unit default is necessary. It is determined by counting the number of bytes in the Smart contract. As a result, the more complicated the contract (the more bytes), the more gases are required to complete the transaction.

The status of the blockchain is modified by execution, i.e. data is added to the chain of blocks. Where data is deleted from the inside. To make things more easier, the activities of writing in the blockchain are free. If you're interested in reading, the read operation is free. Because the Blockchain state change must be handled like any other transaction, write operations are slower. Reading else go actions are performed in a copy of the global state where data is based in the local node that gets the read request, making them relatively faster. Figure 5 is a nice representation of a Smart Contract diagram. [12]

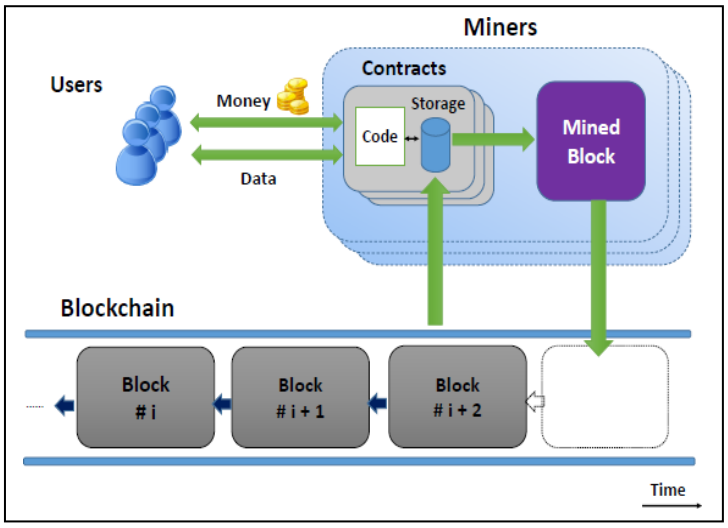



Figure 4: The structure of smart contract [12]

# 10. The reforms

The blockchain is a scalable technology, it is possible to modify the consensus rules.Those changes are called a forked (fork). In practice, that given location at a soft forking Wherea hard fork.

## 10.1. Forking temporal

In a distributed network, some systems in the network will be late in information or will have alternative information. It depends on the network latency between the nodes that gives rise to a conflict in the network. Resolving data conflicts is essentialfor agree on the state of the network chains of blocks [5].

If two blocks are validated at the same time, then two parallel chains develop, after adding some blocks, only the chain the more long subsists

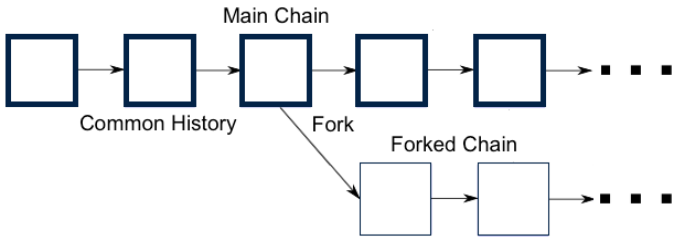



Figure 5:The fork [5]

# 11. Advantages of blockchain

☑ Blockchain establishes the authority and legality of its own transaction in order to appeal to a central administrator [13].

☑ The blockchain, like all other bdd, needs to be run on physical equipment.

☑ To all knots, the information held in a Blockchain is transparent [13].

☑ Data is not kept in a single location. As a result, there is no security officer [14].

☑ The chances of the system failing are extremely slim. Because she is executed across various systems and locations, Blockchain's resiliency is significantly superior to that of a standard database system. Yes, if a node fails, the other knots will take over the relay immediately [14].

## 12. disadvantages of blockchain

☒ A traditional database system is faster than the Blockchain. This is more expensive since it requires more energy, materials, and infrastructure capability.

☒ Each new link, on the other hand, requires proof of the validity and source integrity. This is done by a digital signature, which implies it will take more time and computing power [15].

☒ A transaction will only be approved if at least half of the nodes have validated it. Because each node must connect with the other knots, this procedure takes time [15].

☒ Because it encrypts all of the information, blockchain must validate and allow each transaction, corn for each transaction, the computations are hard.

☒ He found it extremely difficult to increase the capacity of an existing blockchain [14].

## 13. Blockchain applications

The mostly from apps his class in apps financial and no financial, inrevenge, blockchain is adopted in of numerous areas:

1. **Financial applications:** Blockchain by the financial sector will eventually lead to savings of costs in from areas such than the reports financial central.
2. **Integrity check:** Blockchain-enabled strings enabled to automate various process in the sector of insurance.
3. **Governance:** Accountability, Automation, and Security Offered by Blockchain could ultimately hamper corruption. For example, certification, identification, contracts of marriage, the taxes and the vote.
4. **Internet of Things:** The main idea is to provide secure data exchange and verifiable in heterogeneous context-aware scenarios with many devices smart interconnected.
5. **Supply Chain Management:** Blockchain Increase Transparency and Accountabilityin the networks of supply chain , allowing so chains of more flexible values. She improves in particular the visibility, the optimization and the demand.
6. **Education:** Blockchain can solve vulnerability, security and privacy issues. confidentiality in the case of environments learning like the management from certificateseducational and in the case of scholarly publishing, blockchain can be used to better deal with submissions of manuscripts.

7. **Privacy and security:** The organizations centralized - public and private - collect of large quantities information personal and sensitive. The blockchain East considered like a occasion to improve the aspects of security of the data.

# Part 2: Related works

As was indicated in the previous part, the uses of the Blockchain are many and varied and include all areas of education, government, medicine, agriculture and others.

All fields intersect in one important point, which is data protection. Blockchain technology is the first to achieve both data protection and integrity.

In this part, we present some related works that apply blockchain technology to protect personal data of users in different fields.

## Mobile Data protection

The authors address consumers' privacy concerns when using third-party services in [21]. The writers focus on mobile platforms, where providers distribute apps for customers to download. These applications are constantly collecting high-resolution personal data over which the user has no control.

The key issue is that the same method might be used for other data privacy concerns, such as patients sharing their medical data for scientific research, with the opportunity to monitor how it is utilized and opt-out immediately.

The topic of personal and sensitive data is explored in this article. They used mobile data as a case study for their contribution.

These forms of data should not be entrusted to third parties, as they are vulnerable to hacking and exploitation. Instead, people should own and control their data without jeopardizing security or limiting the ability of businesses and governments to provide individualized services.

They presented a framework that accomplishes this by combining a blockchain that has been repurposed as an access-control moderator with an off-chain storage solution. Users are not obligated to trust third parties and are always informed of the data collected about them and how it is used. Furthermore, the blockchain acknowledges users as the owners of their personal information. As a result, businesses can concentrate on using data rather than worrying about properly safeguarding and compartmentalizing it.

## Medical Data Protection

In [22] ,the authors are interested in preventing theft, exploitation, manipulation, and destruction of personal health data kept on unencrypted servers.

The authors suggest a platform that gives patients back control of their personal information. The goal of this project is to keep sensitive health data on the Blockchain in order to achieve accountability, integrity, and security. Patients will have complete discretion over the blocks in which their data is stored. Patients' pseudonymity is currently lacking in healthcare systems, but the suggested platform provides it. Patients' interest will be rekindled in this vulnerable situation of EHR systems, and accountability, integrity, pseudonymity, security, and privacy will be preserved, which are currently being lost in electronic systems.

### Education Data protection

The authors of [23] examine the difficult, error-prone, and insecure procedure of distributing students' credentials. They intend to use blockchain technology to address the existing security concerns around the sharing of students' credentials. As a result, the study presents a blockchain-based architecture for secure sharing of student credentials that is tamper-proof, immutable, authentic, non-repudiable, privacy protected, and easy to distribute. A secure off-chain storage method is also used in the proposed system.

## 14. Conclusion

In this chapter, we start by introduced Blockchain technology. This IT innovation thus makes it possible to organize the exchanges of data on a distributed network, ensuring security data by encryption, and involving the nodes of the network for the creation of new blocks of the chain.

The basic principle of a blockchain is based on the notion of proof of work, and uses techniques of the cryptography for to verify the holders distinct of one system recording collectives.

Then, we present some related works in different fields that based in the blockchain technology for enhancing the private data protection.


# ALGERIAN REGULATION OF DATA PROTECTION

# 1. Introduction

Algeria issued Law No. 18-07 on the protection of natural persons in the field of processing personal data in order to frame the legal protection of the private life of individuals and preserve their reputation, honor and the dignity of their families by protecting their personal data, which requires the express consent of the person concerned in order to process his personal data.

In this chapter, we present how the Algerian legislator deals with the issue of protecting the personal data of natural persons, and to know the most important legal mechanisms that it has developed in this field.

# 2. Framework and principles for personal data protection

## 2.1. Conceptual framework

Unusually, the Algerian legislator gave definitions and controls for all terms in the field of protection of natural persons in the field of data processing of a personal nature, and this is due to the fact that this law is a precedent in Algerian laws and in order that these terms do not remain the subject of different interpretations among which rights are lost or violations are committed in their name, and because the definition The legislature is superior to the rest of the other definitions. We have decided to adjust these definitions as mentioned in this law, with reference to some other laws.

**1. Data of a personal nature:**

is “every information, regardless of its support, related to an identified or identifiable person.. directly or indirectly, especially by reference to the identification number or an element or several elements of his physical, physiological, genetic, biometric, psychological, economic, cultural or social identity.”

The Moroccan legislator defined it as “personal data, any information of any kind regardless of its support, including sound and image, and related to a self-identified or recognizable person named after him as “the person concerned”.

The Tunisian legislator defined it as “...all data, regardless of its source or form, which makes a natural person identifiable or identifiable directly or indirectly, with the exception of information related to public life or considered as such by law.”

That is, any information that indicates a person directly or through processing or analysis, whether it is on a paper, electronic or other support, except for that information related to public life.

**2. The concerned person:**

is every natural person for whom data of a personal nature is the subject of processing.

**3. Processing of personal data:**

The Algerian legislator defined it as “every process or group of operations performed by means or

by automated means or without them on data of a personal nature, such as collection, registration, organization, preservation, fit, change, extraction, access, use or receipt by transmission, publication or any other Other form of availability, approximation, interconnection, locking, encrypting, erasing, or destroying

That is, every operation carried out by a natural person or entity that leads to the introduction of modifications and changes or the exploitation of these data for a specific purpose or without a purpose and by any means, mechanical, manual or otherwise.

The Tunisian legislator defined it as "the operations carried out, whether automatically or manually, by a natural or legal person, and which aim in particular to collect, record, preserve, organize, change, exploit, use, send, distribute, publish, destroy or access personal data, as well as all operations related to By exploiting databases, indexes, records, cards, or by interconnecting."

In the Bahraini legislation, "Any operation or group of operations performed on personal data by an automated or non-automated means, including the collection, recording, organization, categorization, storage, modification, modification, retrieval, use or disclosure of such data, from through broadcasting, publishing, transmitting, making it available to third parties, merging it, blocking it, erasing it or destroying it."

**4. Sensitive data:**

"data of a personal nature that shows racial or ethnic origin, political opinions, religious or philosophical convictions, or union affiliation of the person concerned, or related to his health, including his genetic data," as all data indicate a person's gender, race or affiliation Political, trade union, or other means may be used to harm the person concerned, his family, or public order as a whole

The Moroccan legislation defines sensitive data as: "data of a personal nature indicating racial or ethnic origin, political opinions, religious or philosophical convictions, or union affiliation of the person concerned, or related to his health, including genetic data."

**5. Responsible for processing:**

The Algerian legislator did not limit the person responsible for processing only to the natural person, but also included the legal person, such as companies, associations, public or private bodies, embassies and others. third parties by specifying the purposes and means of data processing," which is the same as what the Tunisian legislator went by saying, "It is every natural or legal person who determines the goals and methods of processing personal data."

**6. Sub-processor:**

defined by the Algerian legislator as "every natural or legal person, public or private, or any other entity that processes data of a personal nature for the account of the person responsible for the processing," that is, every processor working for another processor through an agency, contract, authorization, request or Jealous.

And in Moroccan legislation, "subprocessor": a personal or legal person, a public authority, a department, or any other body that processes data of a personal nature for the account of the person responsible for the processing.

## 2.2. Basic principles of personal data protection

**1. Prior consent and quality of data**:

Article (07) of this law stipulates the need to give the express consent of the person concerned in order to allow the processing of his personal data. His consent is necessary when it comes to respecting a legal obligation to which the subject is subject, to protect his life, to perform a contract to which he is a party, or to preserve his vital interests when he is physically or legally unable to express his consent, or in the matter of public interest or the functions of a public authority (corresponding to Article 04 of Moroccan legislation, with the addition of another case, which is excluded from the consent of the person concerned with the expression data of a personal nature obtained in application of a special legislative text). Finally, to achieve a legitimate interest by the person responsible for the processing, taking into account the interest, rights and freedoms of the person concerned, we find that the legislator, in order to preserve the limits of the person's freedom and rights, mentioned these cases by way of example, not for example, so that any case outside what was mentioned in this article is an explicit violation and an infringement on personal data which have become protected under this law. As for data relating to children, according to Article (08) of this law, their treatment is contingent upon the approval of their legal representative or with a license from the competent judge when necessary, and the latter can authorize even without the consent of the legal representative whenever the best interests of the child so require.

As for the method of data processing, the legislator stipulated that it be carried out in a legitimate and impartial manner, for specific purposes and not exaggerated, and that it be correct, complete, updated if necessary, and preserved in a way that allows identification of people within an appropriate period to achieve the goal of processing. (Corresponding to Article 03 of Moroccan legislation).

**2. Pre-treatment:**

The law requires that a license or authorization from the authority be obtained for any processing of personal data.

**A. Declaration:** The permit request shall be filed in accordance with the provisions of this law with the National Authority in return for obtaining a receipt within a maximum period of 48 hours. Conservation and others, also select the same.

The law is cases that are not subject to mandatory authorization.

**B. Licensing:** The National Authority, after examining the permit deposited with it, subjected any treatment that involves obvious dangers to respecting and protecting private life to a prior authorization by means of a reasoned decision, which is communicated to the person

responsible for the treatment within 10 days from the date of depositing the permit. With regard to the public interest and necessary to ensure the exercise of the legal or statutory functions of the person responsible for the processing or after the approval of the person concerned as stated in Article 18 of the same law, it also provided for other cases mentioned by the legislator exclusively in which a license can be obtained to process sensitive data, and the article stipulates (20) of it on the information that must be included in the license and the legal deadlines for responding to the license request.

## 3. National Authority for the Protection of Personal Data

It is an independent administrative authority that enjoys legal personality and financial and administrative independence. It is established by the President of the Republic. Its headquarters is in Algiers. It consists of (13) members appointed by presidential decree for a term of (05) years, renewable, including: (03) members, including the president, appointed by a president

The Republic is competent and (03) judges proposed by the Supreme Judicial Council and a member of each chamber of Parliament and one representative for each of: the National Council for Human Rights, the Minister of National Defense, the Minister of Foreign Affairs, the Minister of Interior, the Minister of Justice, the Minister of Post and Communications, the Minister of Health The Minister of Labour, Employment and Social Security, and the National Authority may seek the assistance of any qualified person who would assist it in its work.

The National Authority, as specified in Article (25) of the same law, undertakes a number of tasks, including ensuring compliance and processing of personal data with the provisions of the law, ensuring that the use of information and communication technologies does not pose any dangers to the rights of persons and public and private freedoms, granting licenses and providing consultations to persons and entities. that resort to processing data of a personal nature and authorizing the transfer of data abroad if this country guarantees a sufficient level of protection for the private life and fundamental freedoms and rights of persons in relation to the processing to which these data are placed as stipulated in Article (44), and to submit any proposal that would simplify Improving the legislative and regulatory framework for data processing, developing cooperation relations with similar foreign authorities, issuing administrative penalties and many other tasks. The National Authority prepares a detailed annual report on all its activities and submits it to the President of the Republic, obligating the head of the authority and its members according to Article (26) to maintain confidentiality. They are given the data and all the information they have access to, even after the end of their duties, and in return they benefit from the protection of the state against any infringement Punishments, insults, or attacks of any nature whatsoever on the occasion or during the performance of their duties. The legislator also prohibited the head of the authority and its members from acquiring any direct or indirect interests in the institutions that exercise their activities in the field of data processing of a personal nature.

On the administrative side of the National Authority, the law stipulates that the latter shall be provided with an executive secretariat to be run by an executive secretary and assisted in his duties

by employees after they have taken the text of the oath mentioned in Article (27) thereof before the Algiers Judicial Council, provided that the organization determines the conditions and modalities for creating this secretariat. The Secretary-General, under the supervision of the President, manages the administrative and technical structures of the Authority. He assumes this capacity, in addition to the powers entrusted to him and the powers that may be entrusted to him by the president. The Secretary-General takes all necessary measures to prepare and organize the work of the Authority, and ensures the preservation and preservation of its files and archives.

The National Authority is also instructed to establish and maintain a national registry for the protection of personal data in which all files processed by public and private bodies, as well as the issued declarations and licenses, as well as the identities of the persons responsible for the processing, and all the data and information provided for by the special regulation determining the conditions and modalities of maintaining the national registry, are registered.

The national authority can issue regulations in which conditions and guarantees are defined for the person concerned when it comes to freedom of expression, health, employment, historical, statistical and scientific research, remote monitoring, and the use of information and communication technologies, in coordination with the concerned sectors. To the National Authority, and it can also carry out transmission security operations through encryption whenever the quality and importance of the data necessitate this matter, especially if it is sent through the network.

## 4. Person rights obligations responsible for the processing

Through two different chapters, the Algerian legislator dealt with the rights of the person concerned in Chapter Four, which he limited to: the right to information, the right to access, the right to correction, the right to object, and the prevention of direct exploration, and in Chapter Five

The legislator defined the obligations of the person responsible for processing through 4 chapters.

### I. Rights of the person concerned

#### 1. The right to be informed:

The legislator obliged the person responsible for the processing to inform each person contacted for his personal data about the identity of the person responsible or represented, the purpose of the processing and every other useful information even if

This collection is indirectly and without contact with him, and if the data is collected through open networks unless he was previously aware of it, he must be alerted and informed of the existence of his data on the networks and can be exploited without his permission. Impossibility, as the Moroccan legislator made limits .

The right to information is restricted by Article (06) in four cases

#### 2. The right to access and the right to rectification:

The person concerned has the right to know whether his data has been processed or not, the purposes of the processing and the recipients, and he has the right to obtain his personal data that is subject to processing and to know the sources for obtaining them. The person responsible for the processing can object to the national authority about access requests when they are arbitrary or repetitive, as he has a request Determining the deadlines for answering when the ability to respond immediately is lost. The Moroccan legislator added another item of a technical nature, which is knowledge of the logic that governs every automated processing of personal data related to it.

As for the right to correction, Article (35) stipulates that a person has the right to obtain free of charge from the processing official to update, correct, erase or close personal data in certain cases and within a maximum period of 10 days from his notification. An answer within the aforementioned period or in the event of receiving a refusal by the person responsible for processing his personal data. He also has the right to inform the third party who received the personal data of any improvement, correction, deletion or closure of the data. In the event of the death of the person, the right in the two previous cases passes to the heirs.

**3. Right to object and prevent direct exploration:**

Among the most important rights established by law is the right of a person to object to the processing of his personal data, especially if it is related to advertising or commercial purposes, and he also has the right to prevent the use of his data for direct exploration by any means and without his consent, which is the important matter that protected all persons, especially customers The mobile phone, which they receive daily promotional messages

## II. Obligations of the person responsible for processing

**1. Confidentiality and integrity of processing:**

The person responsible for processing according to this law is obligated to take all necessary technical measures and precautions in order to protect and secure personal data from piracy, damage and any illegal use, especially if it is sent over a specific network. These measures increase as the value and importance of these data increases.

If the person responsible for the processing employs another official (sub-official) who works for his account, the latter must provide sufficient guarantees for the safety and security of the data of a personal nature, and this authorization must be by contract or legal document in writing or can be preserved (for evidence-gathering purposes), In particular, it states that the sub-processor shall act only in accordance with the directions and instructions of the first person responsible for the processing, in order to determine the legal responsibilities and so that the rights of persons are not lost between the person responsible for the processing and the person responsible for the sub-processing. They are also obligated, in accordance with the rules of common law and in accordance with the provisions of this law, to maintain professional secrecy even after the end of their duties.

**2. Processing personal data in the field of electronic certification and signature and in the field of electronic communication:**

The electronic certification service providers are obligated to process personal data in order to deliver and preserve the certificates associated with the electronic signature without any other purposes, except in the case of the express consent of their owners. The service providers in the field of electronic communications are also obliged, after they have taken all the necessary guarantees to protect the data, to inform the national authority and the person concerned if there is Infringement of his private life in cases of destruction, loss, disclosure or unauthorized access. They are also obligated to make an inventory of all violations of personal data and the measures taken in this regard.

**3. Transfer of data to a foreign country:**

Law 18-07 grants the National Authority the right to authorize those responsible for processing the transfer of data to a foreign country whenever the authority deems that this country guarantees an adequate level of protection for the private life, freedoms and fundamental rights of people and appropriate security measures and when it considers that the transfer of this data does not pose a threat to public security and interests The vitality of the state, and thus this law gives the necessary protection to the national data that was accessible to foreign companies operating in Algeria, especially telecom companies, Internet providers and embassies that receive thousands of visa applications and the personal data they carry that can be easily transferred to other countries in the absence of a legislative text prohibiting that. The same law also specified, through Article 45 of it, the exceptions with which data can be transferred abroad despite the foreign country's failure to meet the aforementioned necessary conditions, including: the express consent of the person concerned or if the transfer is necessary for the life of this person or to preserve the public interest or Respect is an obligation that allows to ensure that a right is proven, exercised, or defended in court and in other exceptional and limited cases.

## 5. Administrative and penal provisions

The Algerian legislator, through Law 18-07, allocated Chapter VI to Administrative and Penal Provisions, which includes (29) articles divided into two chapters as follows:

**1 Administrative procedure:** The legislator made each of the warnings, warnings, temporary withdrawal of the license or final withdrawal, and the fine penalties for breaching the provisions of this law, these penalties are issued by the National Authority, and its decisions are subject to appeal before the State Council. The National Authority can also impose a fine of 500,000 Algerian dinars against each person responsible for processing that occurred in certain cases specified by Article (47) of the same law. It can also withdraw the permit or license receipt without Subject to any deadline, when it comes to compromising national security, morals or public morals, and it has the right to inspect the shops and places where the treatment takes place, unless they are residential premises. It also has the right to access any processed data and all information and documents, whatever their support.

The National Authority can also seek the assistance of censorship agents to search and inspect

crimes of assault on personal data under the supervision of the Public Prosecutor, and this is, of course, in addition to judicial police officers and agents according to Article (50).

Any person whose rights have been violated in this field may apply to the competent judicial authority to take any precautionary measures to put an end to the infringement or to claim compensation (52).

As for Article (53), we find that the legislator has referred to Article (588) of the Code of Criminal Procedure, which stipulates the rules of jurisdiction that must be observed by the Algerian judicial authorities when pursuing the crimes stipulated in this law, as the legislator concluded and through Article ( 53) Jurisdiction "for the Algerian judicial authorities to follow up crimes of assault on personal data .. committed outside the territory of the Republic by an Algerian, a foreign person residing in Algeria, or a legal person subject to Algerian law."

**2. Penal provisions:** The legislator, through Law 18-07 in the chapter entitled "Penal Provisions," tightened the penalties for violating the provisions of this law, in which imprisonment ranges from two months to five years, and fines range from 20,000 DZD to 1,000,000, and the penalties vary according to the different violations committed by the treating person, the sub-processor, or any other person whose behavior led to a violation of the provisions of this law, and among the cases: Processing data of a personal nature despite the objection of its owner or when making false statements or continuing to work Despite withdrawing the license or authorization, processing data for purposes other than those authorized, collecting data in a fraudulent and dishonest manner, allowing unqualified persons to access the data or obstructing the work of the National Authority, as well as setting penalties for anyone who enters without being responsible to the National Registry or every processor Rejects without reason the draft rights of media, access, correction or objection, and anyone who informs the National Authority of violations of personal data, and the penalties are given the maximum in the case of transferring data of a personal nature to a foreign country in violation of the law, or anyone who retains data of a personal nature regarding crimes, convictions or measures security .

As for the legal person who violates the law, the legislator has referred to the Penal Code. It also decided, through Article (73), for any attempt to commit one of the aforementioned misdemeanours, the same penalty for the complete crime, and the penalties are doubled in case of recurrence.[16]

# 6. Conclusion

In this chapter, we discussed Law 18-07 that strengthens the legislative system in Algeria and aims to protect freedoms and rights by stipulating the necessity of prior and explicit consent of the person concerned with the data to be processed before starting any processing process, even if it is authorized. In order to put an end to the chaos prevailing in this field, especially since people daily put their data at the disposal of public or private bodies, telecommunications companies or foreign embassies in Algeria, and then do not know their fate. To apply this law and to grant licenses and

licenses to officials wishing to process these data, carry out investigation tasks and impose penalties on all those who chose this law.


# CONTRIBUTION

# 1. Introduction

In this chapter, we give details about how securing and keeping the privacy of the personal data using blockchain technology with respect to the Algerian regulation. We present our detailed architecture to secure the data. As well as, the design use access control, logs and rewards mechanisms.

# 2. The proposed Architecture and its main Functionalities

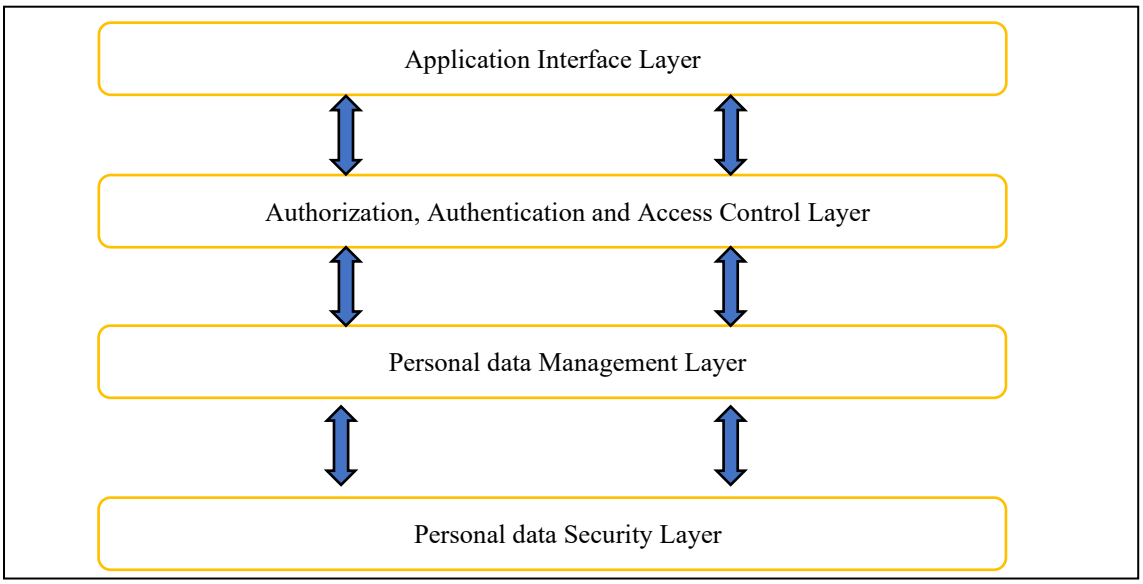



Figure 6:Layered Architecture of Our Contribution

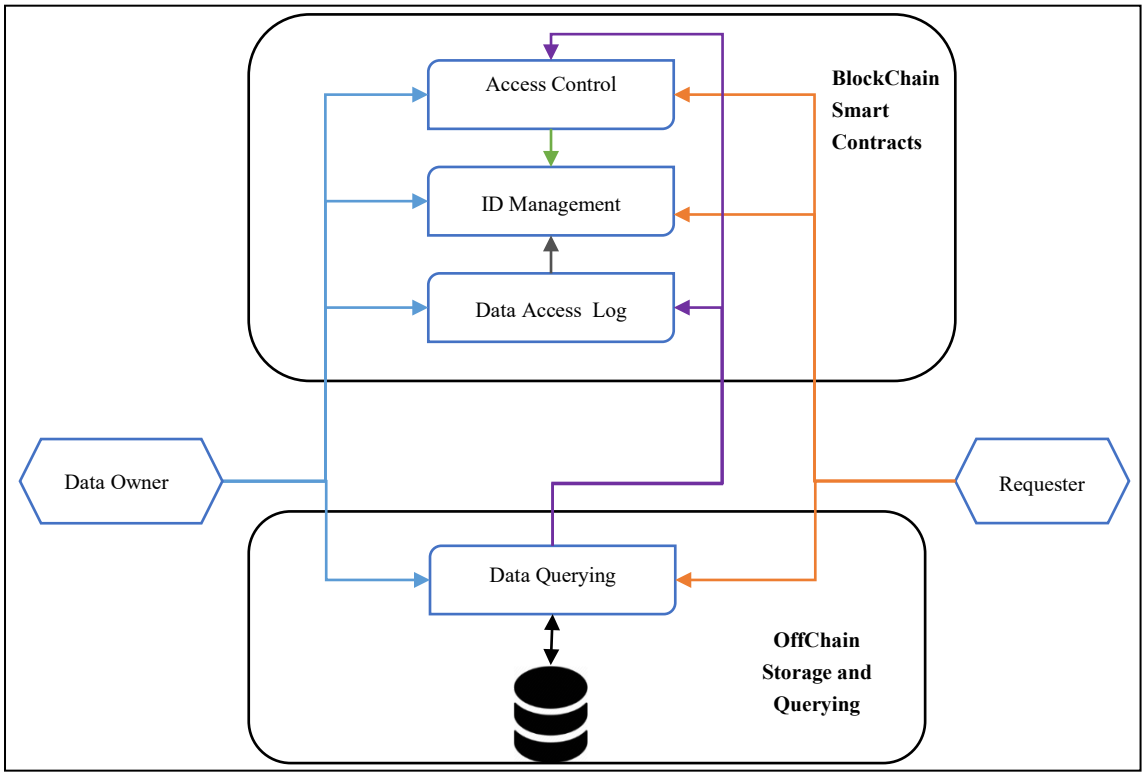



Figure 7:The Main Components of the proposed Architecture along with theirs Interactions

## 2.1. Users Roles and Responsibilities

**The Data Owner:** Is the current owner of the data. It can be humans, enterprises, universities, clinics, laboratories, etc. It represents not only one owner but multiple ones. The Algerian data regulations insists that the data owners must have the full control over its data. Thus, they put an access controls and policies in order to restrict the access and the manipulation of their data. They put two types of policies: default and non-default where the first indicates that the target operations are granted without back to the data owners in contrast to the latter which it must request directly the data owners.

**Requester:** It acts as data stockholders by requesting data access in order to use the requested the personal data data in its processing. Therefore, every request is saved on the blockchain for future verification to detect illegal data manipulations. Thus, the requester must accept the rules of data processing before access the data. Every requester must register and enrolled by the blockchain in order to manage its authorization and access control. The data about the requester registration is under its responsibility and therefore every illegal access by another entity with this critical information is not the responsibility of the blockchain network.

## 2.2. On-chain components

**Access Control:** It verifies if the data requester has the right to access to the data in order to make an operation such as: read and update the personal data data. The process is achieved by requesting the blockchain ledger to get the access policies of the requested data. If the needed operation is in the default policy, then the access is granted, if it is not then request is forwarded directly to the data owner. In other situation, the requester can request access after a permission that has been given to him. Therefore, the access control verifies if the requester had an existed acceptation from the data owner. In the end, the AC component sends his report to the coordinator in order to complete the access protocol.

| Information | Description |
|---|---|
| **Data owner identifier** | The owner identity that has been generated by the IDM. |
| **Data identifier** | A unique identity of the knowledge graph subject to access. |
| **Requester identifier** | The requester unique identity that has been generated by the IDM |
| **Data identifiers** | The requested data identities that have been used by the off-chain management. |
| **Permissions** | The permissions that have been given to the requester for every requested data. |

| | |
|---|---|
| **Validities** | The permissions validities for every requested data. |

*Table 4: list of the attributes that are stored on the blockchain for the access control*

**Audit:** its main objective is making an advanced verification to detect inconsistencies in the history of authorization and access control components. It merges data from all ledgers and performs advanced checking. For example, if the AC component gives access permission without identity checking by the authorization, then this illegal access can be detected using the ledgers of these two components. The process of auditing is started only in demand of the KG owner.

| Information | Description |
|---|---|
| **Requester identifier** | The requester unique identity that have been generated by the IDM |
| **Data identifier** | The requested data identifier that have been used by the off-chain management. |
| **Operations** | The permissions that have been given to the requester for every requested data. |
| **Time** | The permissions validities for every requested data. |

*Table 4: list of the attributes that are stored on the blockchain for the audit*

**Identity Management:** it has two main tasks: the first is creating identities and registering the new owners and requesters (along with its attributes) which requests critical and sensitive KG data, the second is verifying if a given identity is valid or not using the blockchain. All identity information is stored in a distributed ledger in order to protect its integrity. The IDM component returns a registration certificate to every accepted demand of registration where there latter contains critical information about enrolling PrivyKG users with different roles.

## 2.3. Off-chain Components

**Data Querying:** It has several specific functionalities for different requester needs. It starts its work after receiving message from the AC component via the coordinator. It has a blockchain access and offchain access in get the target data.

## 2.4. Main Scenarios in PrivyKG and their Smart Contracts

In this section, a set of the main scenarios that can be controlled by the PrivyKG smart contracts is illustrated. The first one which is presented by the figure (9) turns about how the permission is requested directly from the BC using two smart contracts: Access Control and Identity Management. These latter interacted with each other in order to achieve the request permission process. After a succeeding the identity verification, the AC verifies if the permission is listed in

the access policy of the data requested and returned a positive response by updating the BC data.

The second interaction presents how requesting the permission in case if the BC cannot give it directly because the data owner must be contacted. In this situation, the data owner does not put the permission in the default access policy and he is the only one who gives it to the requester. The process uses the same smart contracts as the first interaction.

The last interaction illustrates how the requester gets the data after he got the permission. Firstly, the data querying component interacted with two smart contracts (AC and IDM) in order to verify the requester identity and the given access grant. After that, it interacted with off-chain server to get the data using its information that has been given from the AC ledger and it transfers the data to the requester. In the end, the audit smart contract is invoked in order to save the information about the operation.

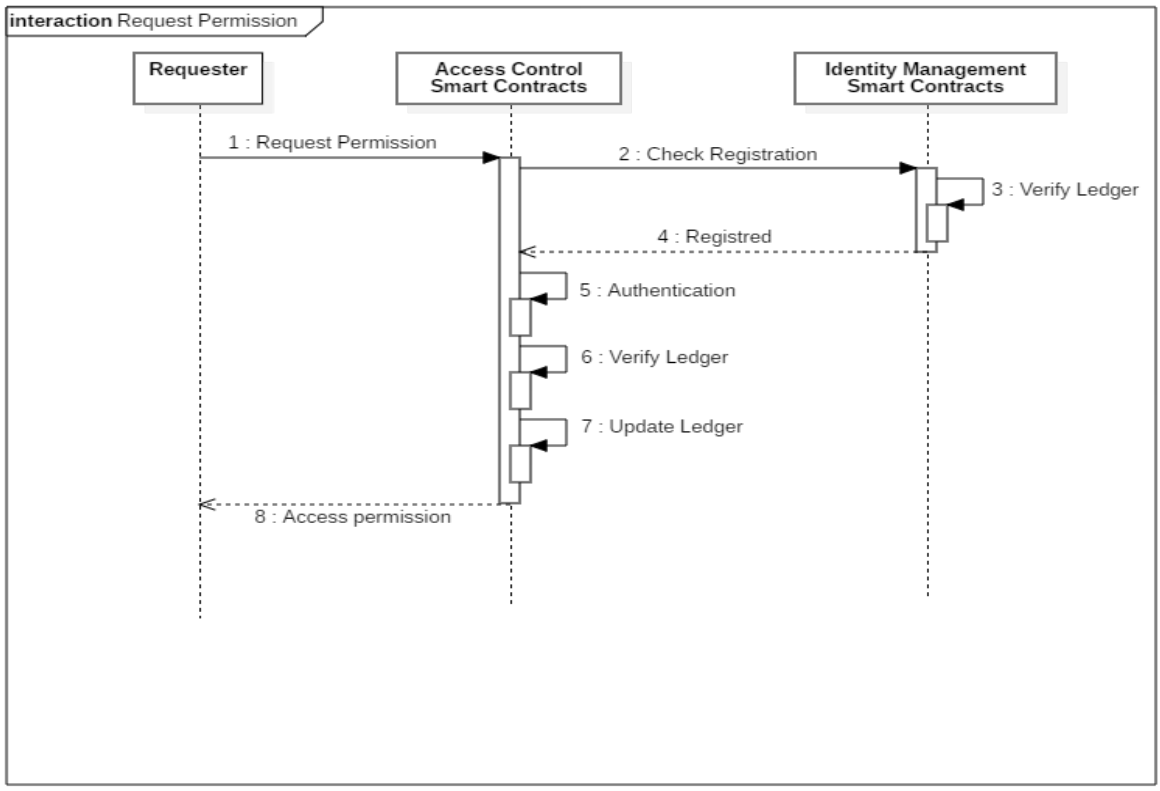



**Figure 8:Direct Request Permission from Blockchain Interaction**

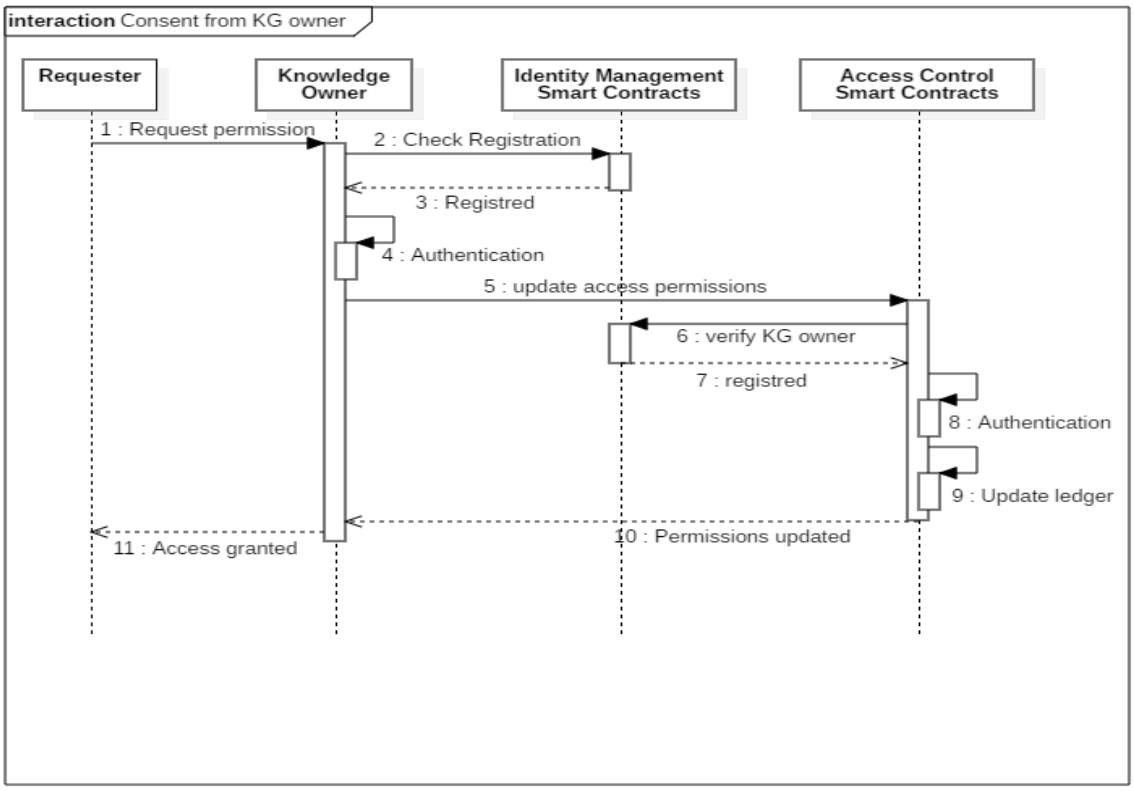



**Figure 9:Request Permission from data Owner Interaction**

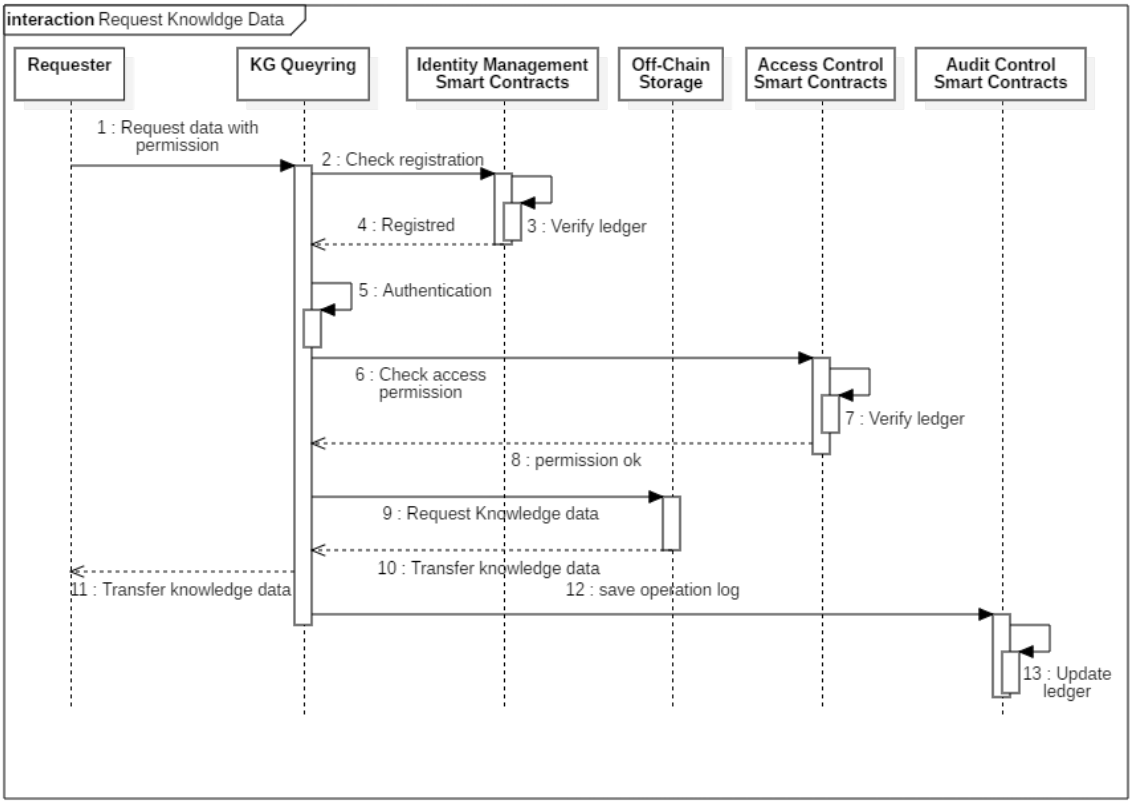



**Figure 10:Request Personal Data Interaction**

## 3. Conclusion

In this chapter, we presented our contribution to securing and maintaining the privacy of personal data using blockchain technology represented in the detailed structure of data security, access control mechanisms, records and rewards.

In the next chapter, we present the steps for building our proposed contribution.


# IMPLEMENTATION & EVALUATION

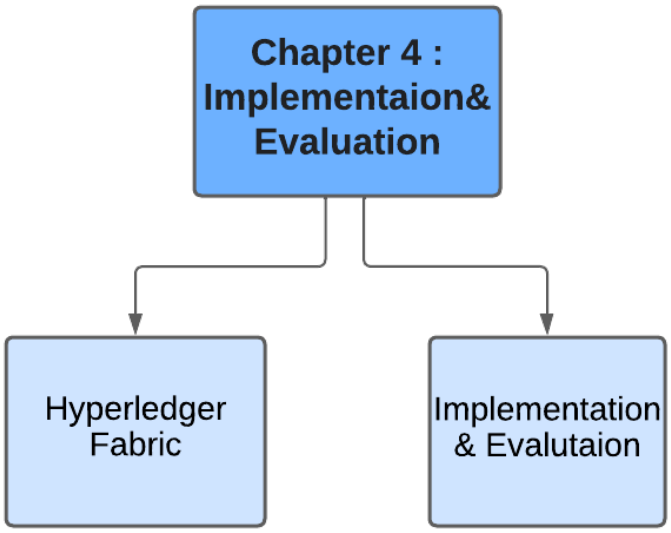
# 1. Introduction

Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture. Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Its modular and versatile design satisfies a broad range of industry use cases. It offers a unique approach to consensus that enables performance at scale while preserving privacy.

This chapter is divided in two parts:

In the firs part we introduce all the concepts of this the Hyperledger fabric.

In the second part we present our steps of implementation and their evaluation.



# Part 1: Hyperledger Fabric

## 2. Hyperledger fabric

The Linux Foundation founded the Hyperledger project in 2015 to advance cross-industry blockchain technologies. Rather than declaring a single blockchain standard, it encourages a collaborative approach to developing blockchain technologies via a community process, with intellectual property rights that encourage open development and the adoption of key standards over time.

Hyperledger Fabric is one of the blockchain projects within Hyperledger. Like other blockchain technologies, it has a ledger, uses smart contracts, and is a system by which participants manage their transactions.

Where Hyperledger Fabric breaks from some other blockchain systems is that it is private and permissioned. Rather than an open permissionless system that allows unknown identities to participate in the network (requiring protocols like «proof of work» to validate transactions and secure the network), the members of a Hyperledger Fabric network enroll through a trusted Membership Service Provider (MSP).

Hyperledger Fabric also offers several pluggable options. Ledger data can be stored in multiple formats, consensus mechanisms can be swapped in and out, and different MSPs are supported.

Hyperledger Fabric also offers the ability to create channels, allowing a group of participants to create a separate ledger of transactions. This is an especially important option for networks where some participants might be competitors and not want every transaction they make — a special price they're offering to some participants and not others, for example — known to every participant. If two participants form a channel, then those participants — and no others — have copies of the ledger for that channel [18].

### 2.1.Shared Ledger

Hyperledger Fabric has a ledger subsystem comprising two components: the world state and the transaction log. Each participant has a copy of the ledger to every Hyperledger Fabric network they belong to.

The world state component describes the state of the ledger at a given point in time. It's the database of the ledger. The transaction log component records all transactions which have resulted in the current value of the world state; it's the update history for the world state. The ledger, then, is a combination of the world state database and the transaction log history.

The ledger has a replaceable data store for the world state. By default, this is a LevelDB key-value store database. The transaction log does not need to be pluggable. It simply records the before and after values of the ledger database being used by the blockchain network.

### 2.2.Smart Contracts

Hyperledger Fabric smart contracts are written in chaincode and are invoked by an application external to the blockchain when that application needs to interact with the ledger. In most cases, chaincode interacts only with the database component of the ledger, the world state (querying it, for example), and not the transaction log.

Chaincode can be implemented in several programming languages. Currently, Go and Node are supported.

### 2.3.Privacy

Depending on the needs of a network, participants in a Business-to-Business (B2B) network might be extremely sensitive about how much information they share. For other networks, privacy will not be a top concern.

Hyperledger Fabric supports networks where privacy (using channels) is a key operational requirement as well as networks that are comparatively open.

### 2.4.Consensus

Transactions must be written to the ledger in the order in which they occur, even though they might be between different sets of participants within the network. For this to happen, the order of transactions must be established and a method for rejecting bad transactions that have been inserted into the ledger in error (or maliciously) must be put into place.

This is a thoroughly researched area of computer science, and there are many ways to achieve it, each with different trade-offs. For example, PBFT (Practical Byzantine Fault Tolerance) can provide a mechanism for file replicas to communicate with each other to keep each copy consistent, even in the event of corruption. Alternatively, in Bitcoin, ordering happens through a process called mining where competing computers race to solve a cryptographic puzzle which defines the order that all processes subsequently build upon.

Hyperledger Fabric has been designed to allow network starters to choose a consensus mechanism that best represents the relationships that exist between participants. As with privacy, there is a spectrum of needs; from networks that are highly structured in their relationships to those that are more peer-to-peer [18].

## 3. Hyperledger fabric Actors

Like any application development, we need an Architect, Developer, Network Admin, and an End User who interacts with the application. In the Blockchain, with these primary actors, we also need regulators who act as an Auditor, the Certificate Authorities for issuing certificates for access and the Data Sources for accessing data. Every actor in the blockchain ecosystem has different roles to play which enables the system to have demarcated processes.

in the next section we precisely discuss some of the important actors who plays the major role in the Blockchain ecosystem [19].

***1) Blockchain Architect***

An Architect / Business Analyst who uses the composer modeling language to create the business network model. The architect is responsible for the architecture and the design of the blockchain solution.

***2) Blockchain Developer***

Who writes the smart contract/business logic which interacts with the Blockchain.

***3) Blockchain Network Operator***

Who manages and monitors the Blockchain network. His/Her functions would vary from deployment to issuing of cards, upgrading to updating the newer versions of files etc.

# 4. Hyperledger fabric Architecture

In Hyperledger Fabric, there is a concept of channels which allows participating organizations to join and communicate with one another. Channel might be considered as a tunnel for one organization to secretly communicate with other participating organizations joining the same channel. Any others who do not take part in the channel in question do not ever have access to any transaction or information associated with that channel. One organization can take part in multiple channels at the same time. The next Figure depicts the simplest Hyperledger Fabric network with two organizations (i.e., Org1 and Org2) joining the same channel. In the next section we introduce Fabric components one by one including Peer, Orderer, CA, and Client 20].

1) **First, Peer** is a blockchain node that stores all transactions on a joining channel. Each peer can join one or more channels as required. However, the storage for different channels on the same peer would be separate. Therefore, an organization can ensure that confidential information would be shared to only permitted participants on a certain channel.
2) **Second**, **Orderer** is one of the most important components used in the Fabric consensus mechanism. Orderer is a service responsible for ordering transactions, creating a new block of ordered transactions, and distributing a newly created block to all peers on a relevant channel.
3) **Third**, **Certificate Authority or CA** is responsible for managing user certificates such as user registration, user enrollment, user revocation, and etc. More specifically, Hyperledger Fabric is a permissioned blockchain network. This means that only permitted users can query (access to information) or invoke (create a new transaction) a transaction on a granted channel. Hyperledger Fabric uses a standard certificate to represent permissions, roles, and attributes to each user. In other words, a user is able to query or invoke any transaction on any channel based on permissions, roles, and attributes he/she possesses.
4) **Fourth**, **Client** is considered to be an application that interacts with Fabric blockchain

network. That is, Client can interact with Fabric network according to its permissions, roles, and attributes as specified on its certificate derived from its associated organization's CA server.

Notice that each component was painted a different color to distinguish a different organization. The components inside the big blue box are on-chain Fabric network entities whereas the components outside the blue box are considered to be off-chain entities [20].

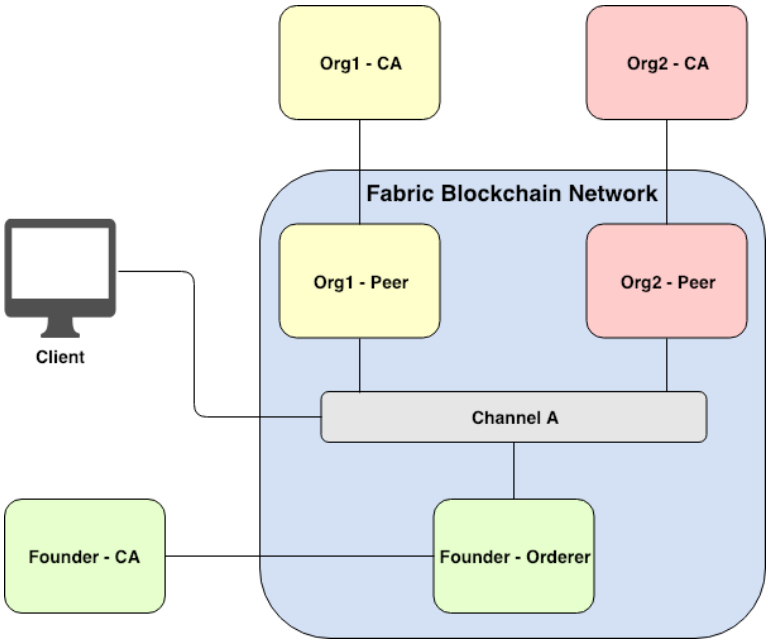



**Figure 11:Simplest Fabric network with two organizations joining the same channel [20]**

There is a concept of Smart Contract called **Chaincode** in Fabric. Currently, three languages can program Fabric chaincode including Golang, Node.js, and Java. To deploy a chaincode, a network admin must install the chaincode onto target peers and then invoke an orderer to instantiate the chaincode onto a specific channel. While instantiating the chaincode, an admin can define an endorsement policy to the chaincode. Endorsement policy defines which peers need to agree on the results of a transaction before the transaction can be added onto ledgers of all peers on the channel.

A peer specified in the endorsement policy is called an endorsing peer which consists of an installed chaincode and a local ledger on it whereas a committing peer would have only a local ledger on it. The next figure distinguishes between an endorsing peer and a committing peer.

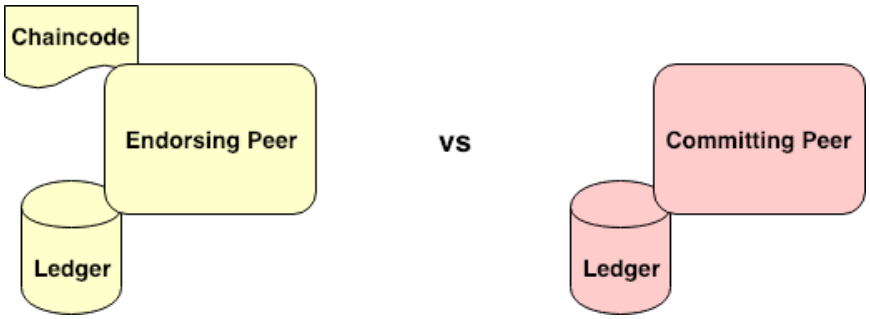



Figure 12:Endorsing Peer vs Committing Peer [20]

As illustrated in the next figure , the interior components inside the Peer's ledger include Blockchain and World State. Blockchain holds the history of all transactions for every chaincode on a particular channel. World State maintains the current state of variables for each specific chaincode.

Two types of World State database currently supported in Fabric include LevelDB and CouchDB. LevelDB is a default key-value database built on Fabric Peer, whereas CouchDB is a JSON-based database supporting rich querying operations based on JSON objects. For instance, CouchDB allows us to set an asset with a specific key and query filtered assets using JSON querying syntax. A chaincode developer must opt to use either LevelDB or CouchDB when developing a chaincode.

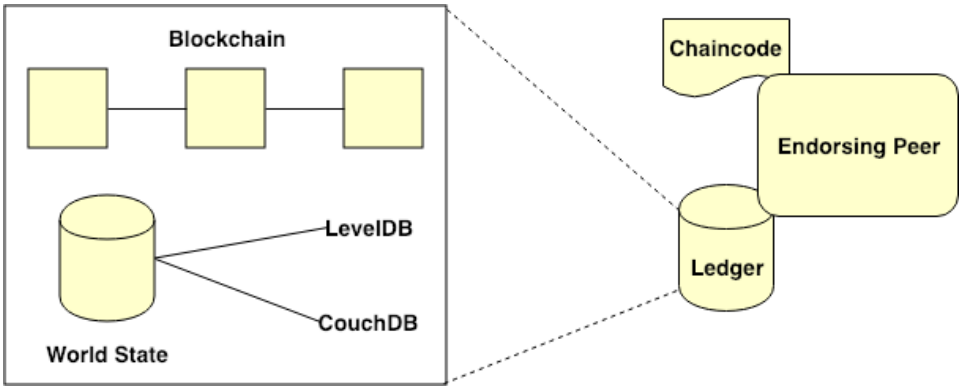



Figure 13:Interior components inside the Peer's ledger [20]

The previous Fabric network can be decorated with chaincodes and ledgers as shown in the next figure. As you can see, Org1's Peer and Org2's Peer mutually join the same channel. There can be multiple chaincodes instantiated on the same channel. Furthermore, the instantiated chaincode can be upgraded if needed. This makes a chaincode to be updatable or patchable.

Let's pay attention to Orderer. There are special system chaincodes and ledgers for Orderer. System chaincodes collect network, channel, and underlying system configurations for Fabric virtual machine to work properly; they are opposed to user chaincodes which run in separate docker containers.

In fact, system chaincodes are also registered and deployed on peers at bootstrap but they were not

put into the figure  for the sake of simplicity. System chaincodes include but are not limited to:

- QSCC (Query System Chaincode) for ledger and Fabric-related queries
- CSCC (Configuration System Chaincode) which helps regulate access control
- LSCC (Lifecycle System Chaincode) which defines the rules for the channel
- ESCC (Endorsement System Chaincode) for endorsing transactions
- VSCC (Validation System Chaincode) for validating transactions

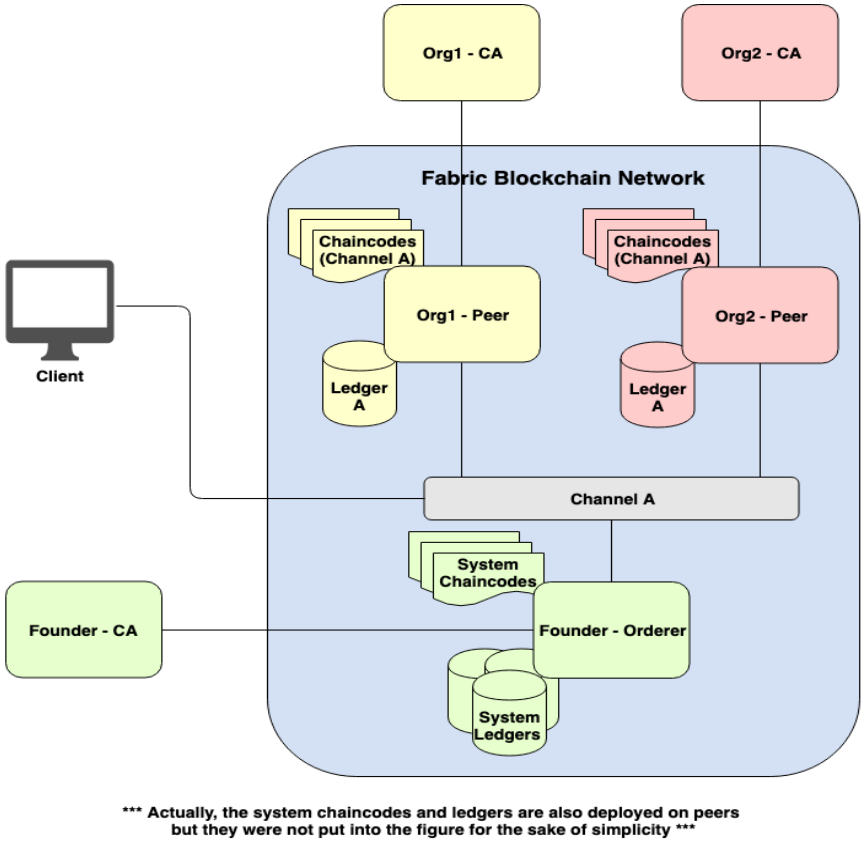



**Figure 14: Fabric network with chaincodes and ledgers attached [20]**

the next figure illustrates a more complex Fabric network with multiple channels. Org1's Peer and Org2's Peer join Channel A together, while Org2's Peer and Org3's Peer mutually join Channel B. With a separate channel, organizations that join the same channel can secretly share business transaction or information together with confidence.

In addition, one chaincode can call another chaincode on the same channel. Moreover, a chaincode can also call another chaincode on a different channel in case a calling chaincode is executed on an endorsing peer joining those two related channels like what Org2's Peer is doing.

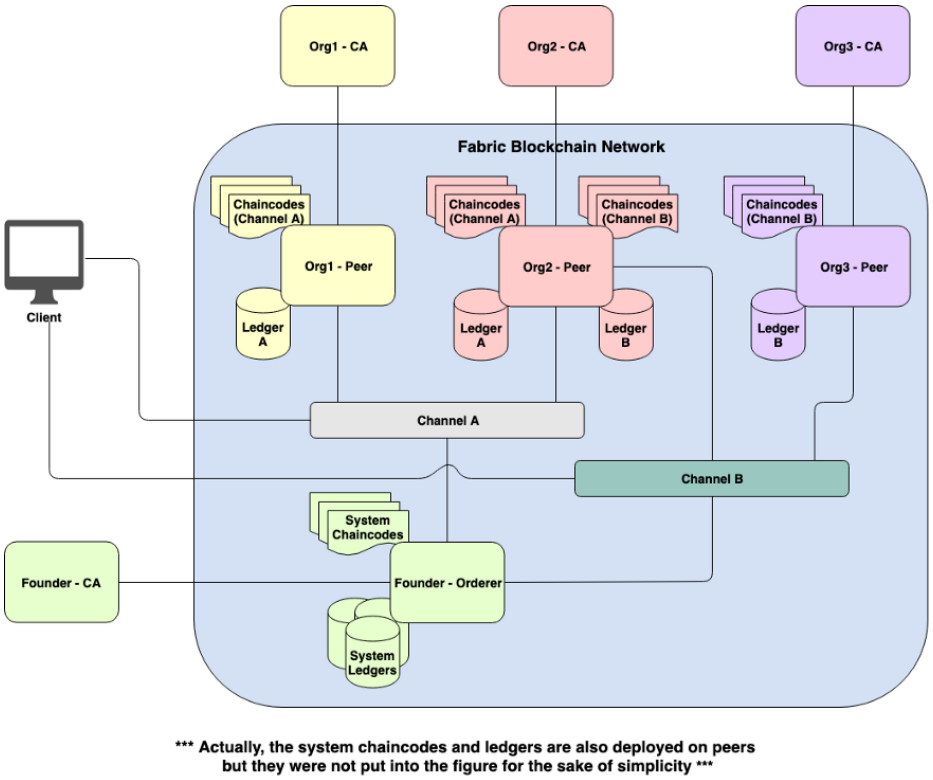



**Figure 15:More complex Fabric network with multiple channels [20]**

# 5. Hyperledger Fabric Consensus

Fabric consensus has an abundance of multi-stage and multi-hierarchy of endorsement, validity, and versioning checks. There are multiple phases to ensure the permission, endorsement, data synchronization across all participants, transaction order, and correctness of changes before writing a block of transactions onto the ledger.

Hyperledger Fabric uses a permissioned voting-based consensus which assumes that all participants in the network are partially trusted. The consensus can be divided into three phases as follow.

- ☑ *Endorsement* (Steps 1–3 in Figure 6 below)
- ☑ *Ordering* (Steps 4–5 in Figure 6 below)
- ☑ *Validation and Commitment* (Step 6 in Figure 6 below)

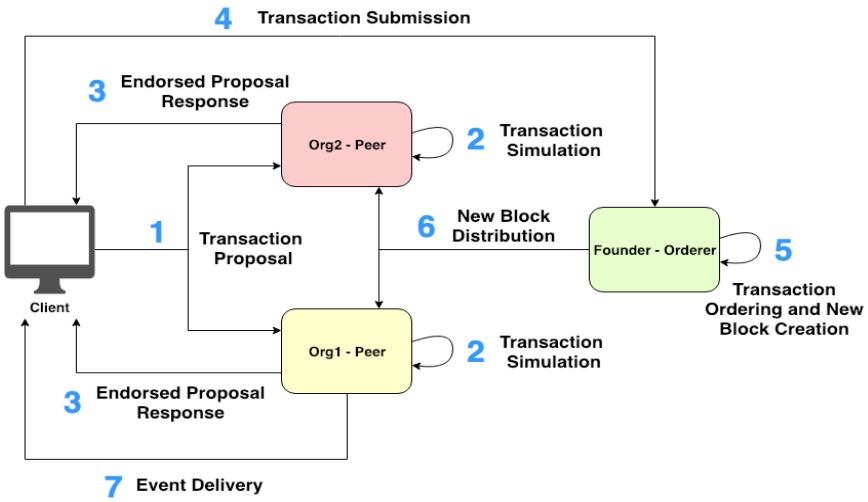



**Figure 16:Fabric transaction invocation workflow [20]**

Figure 6 describes the step-by-step workflow of Fabric transaction invocation:

1) *Client* makes a transaction proposal, signs the proposal with *User*'s certificate, and sends the transaction proposal to a set of pre-determined *Endorsing Peers* on a specific channel.
2) Each *Endorsing Peer* verifies *User*'s identity and authorization from the proposal payload. If the verification check passes, *Endorsing Peer* simulates the transaction, generates a response together with a read-write set, and endorses the generated response using its certificate.
3) *Client* accumulates and checks the endorsed proposal responses from *Endorsing Peers*.
4) *Client* sends the transaction attached with the endorsed proposal responses out to *Orderer*.
5) *Orderer* orders the received transactions, generates a new block of ordered transactions, and signs the generated block with its certificate.
6) *Orderer* broadcasts the generated block to all *Peers* (to both *Endorsing Peers* and *Committing Peers*) on the relevant channel. Then, each *Peer* ensures that each transaction in the received block was signed by the appropriate *Endorsing Peers* (i.e., determining from the *invoked chaincode*'s *endorsement policy*) and enough endorsements are present. After that, a versioning check (called the *multi-version concurrency control (MVCC)* check) will take place to validate the correctness of each transaction in the received block. That is, each *Peer* will compare each transaction's readset with its *ledger*'s *world state*. If the verification check passes, the transaction is marked as valid and each *Peer*'s *world state* is updated. Otherwise, the transaction is marked as invalid without updating the *world state*. Finally, the received block is appended into each *Peer*'s *local blockchain* regardless of whether or not the block contains any invalid transactions.
7) *Client* receives any subscribed events from *EventHub service*.

# 6. Ordering Service in Hyperledger Fabric

Orderer might be one of the most important components in Hyperledger Fabric. It acts as a hub for distributing blocks of transactions to all peers on a relevant channel. For this reason, Orderer might be considered to be the weakest point in the Fabric network.

The current implementation of Fabric Orderer supports two types of ordering service, namely Solo and Kafka. Solo-based ordering service is recommended for use in the development environment only since this kind of service composes of a single process which serves all clients. This is prone to be a single point of failure in a production environment.

In production, Kafka-based ordering service is intended to be used. With Kafka, we can set up a Kafka cluster and a ZooKeeper ensemble to provide a crash fault-tolerant ordering service.

Even though Kafka would provide Crash Fault Tolerance (CFT) consensus to Orderer, there can still be only one organization to fully control the ordering service. It is, however, insufficient for one organization to mastermind Orderer because that organization may not be trustworthy.

Fortunately, Fabric ordering service was designed to be pluggable. Currently, Byzantine Fault Tolerant (BFT) consensus is under development. The BFT-based consensus would enable the network's participating organizations to jointly control the ordering service, resisting the system from reaching agreement in the case of malicious actors or faulty nodes.

# 7. Hyperledger Fabric in Production

In production, there can still be several Fabric-related components to be collaborating with. The next figure summarizes a deployment model for Fabric network in a production environment.

***Client application*** can interact with Fabric blockchain network in two ways: via *Fabric SDK* or *Fabric CLI (command line interface)*. ***Fabric SDK*** provides a set of rich functions, which is appropriate for use in production. Typically, ***Client application*** (*Client no. 1* in Figure 7) interacts with Fabric network by way of connecting to ***RESTful API Server*** which uses ***Fabric SDK*** as a library to communicate with the blockchain network. ***Fabric SDK*** currently supports *Node.js* and *Java* languages. In addition, *Python*, *Golang*, and *REST* SDK versions are under development. ***Fabric CLI*** is appropriate for use in a development or maintenance mode (*Client no. 2* in Figure 7).

In Fabric, ***CA*** is used for user management and certificate issuance tasks. There are two ways to deploy ***Fabric CA***. First, setting up ***Fabric CA*** without extending ***LDAP Server***. With this configuration, ***Fabric CA*** would be used for registering users, authenticating users, and issuing user certificates (i.e., user enrollment). Second, setting up ***Fabric CA*** with extending ***LDAP Server***. With this configuration, ***Fabric CA*** would be used for issuing user certificates only. Whereas, ***Fabric CA*** would delegate ***LDAP Server*** to manage other tasks instead such as registering users, authenticating users, revoking users, and etc. The second option is suitable for connecting ***Fabric CA*** with an organization's existing AD, LDAP or Radius server.

***CouchDB*** might be the best option to use as a *world state database* for ***Peer***'s *ledger* in production since it supports several rich features, such as JSON querying operations, database indexing, data replication, ACID properties, and etc. Whereas, ***LevelDB*** supports only limited operations.

To support *Crash Fault Tolerance (CFT) consensus* for Fabric ordering service, extending ***Orderer*** with ***a cluster of Kafka brokers*** is a choice in production. In order for ***Kafka cluster*** to work properly, a ***ZooKeeper cluster*** is required for coordinating local tasks across *distributed Kafka brokers*.

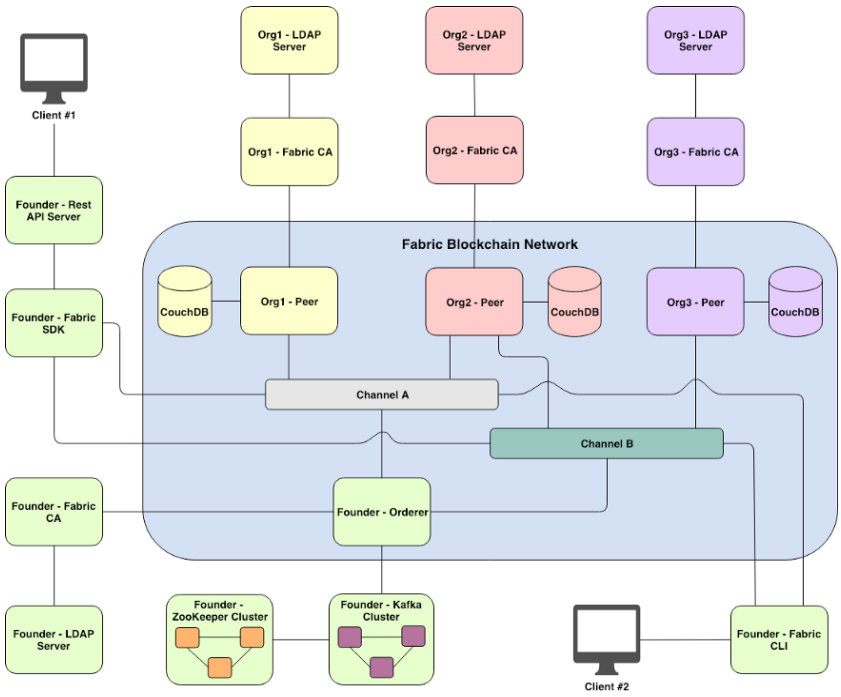



**Figure 17:Fabric network in a production environment [20]**

# 8. Hyperledger Fabric Benefits

1) **Open Source:** Hyperledger fabric is an open-source blockchain framework hosted by the Linux foundation. It has an active community of developers The code is designed to be publicly accessible. Anyone in the community can see, modify, and distribute the code as they see fit. People across the world can come and help to develop the source code.

2) **Private and Confidential:** In a public blockchain network each and every node in the network is receiving a copy of the whole ledger. Thus keeping privacy becomes a much bigger concern as everything is open to everyone. In addition to this one, the identities of all the participating members are not known and authenticated. Anyone can participate as it is a public blockchain. But in the case of Hyperledger fabric, the identities of all participating members are authenticated. And the ledger is only exposed to the authenticated members. This benefit is the most useful in industry-level cases, like banking, insurance, etc where customer data should be kept private.

3) **Access Control:** In the Hyperledger fabric, there is a virtual blockchain network on top of the physical blockchain network. It has its own access rules. It employs its own mechanism for transaction ordering and provides an additional layer of access control. It is especially useful when members want to limit the exposure of data and make it private. Such that it can be viewed by the related parties only. As an example when two competitors are on the same network. The fabric also offers private data collection and accessibility, where one competitor can control the access to its own data such that the data do not get exposed to the other competitor.

4) **Chaincode Functionality:** It includes a container technology to host smart contracts called chain code that defines the business rules of the system. And it's designed to support various pluggable components and to accommodate the complexity that exists across the entire economy. This is useful for some of the specific types of transactions like asset ownership change.

5) **Performance:** As the Hyperledger fabric is a private blockchain network, There is no need to validate the transactions on this network so the transaction speed is faster, resulting in a better performance [20].

# Part 2: Implementation & Evaluation

## 1. Implementation

PrivyKG is implemented under Eclipse using various Java APIs such as: JGraphT, and JSON, Fabric SDK. The implementation architecture is illustrated in figure"19" where the main components are as follow:

***The Fabric Hyperledger Blockchain*** is used with the configuration of two organizations and one peer node for each one. The fabric network uses CouchDB as world state database and one ordering service. It was built with one certificate authority for each organization. Six channels are created for access control, access logs, knowledge completion, rewards, entities and relations named respectively "Access Control", "Access Log", "Knowledge Completion", "Reward", "Entities" and "Relations". Six Fabric smart contracts are deployed using Go language (one for each channel). Rather than using LevelDB, CouchDB is selected as world-state storage for the Fabric nodes for several reasons. One of them is that LevelDB does not have rich query expressions like CouchDB which has more expressive power of expressing queries to the ledgers. In contrast, with CouchDB we can use the query selectors to get any data from the world states. The Hyperledger Fabric network used by the proposed method is given by the figure 19 where every channel is associated to its ledger and smart contract. The figure 20 illustrates the steps of executing a transaction in Fabric network for the chain-code associated with entities channel.

***MangoDB*:** is used for the off-chain storage where every data is stored as JSON objects. Queries are specified using the NoSQL to interrogate the database in order to get or modify the triples.

***Client Application*:** a component that need to use the decentralized KG in its processing.

All these latter are interacted with the main program using their specific Java API.

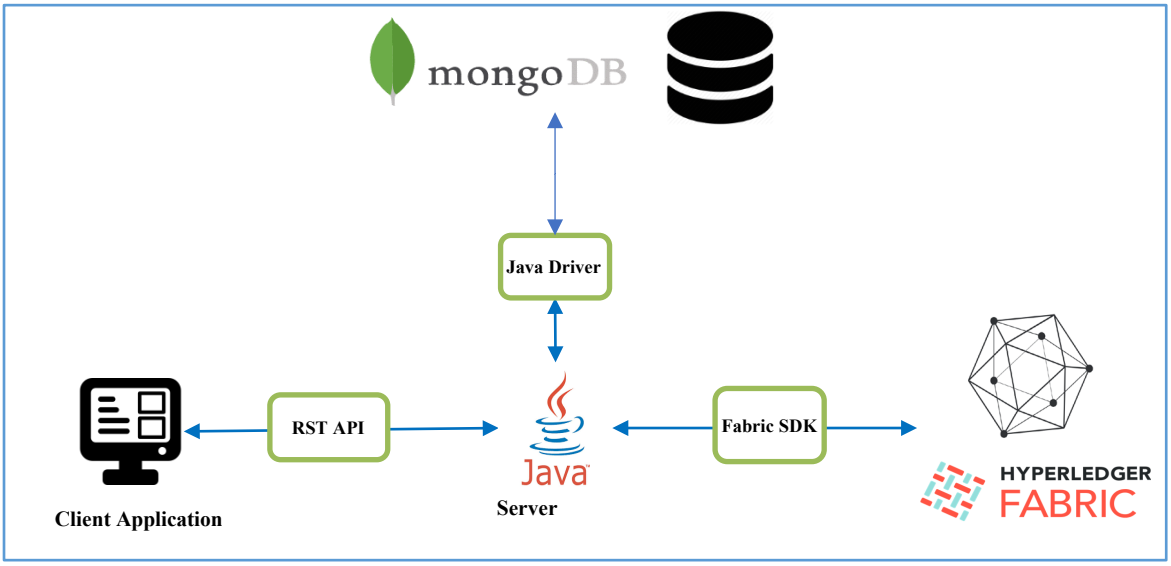



Figure 18:Implementation Architecture of our contribution.

## 1.1.Fabric Chaincodes and Distributed Ledgers in PrivyKG

Every peer in HLF has its local database (ledger) with contains all transactions executed by the network via HLF chaincodes. Thus, every peer can have several installed chaincodes for one HLF channel. The distributed ledgers in HLF are updated using smart contracts in demand by the blockchain external users. We propose to use 3 distributed ledgers where each one is associated with one smart contract and several peers. These ledgers store critical data about system functionalities such as: access control, access logs and on-chain user data.

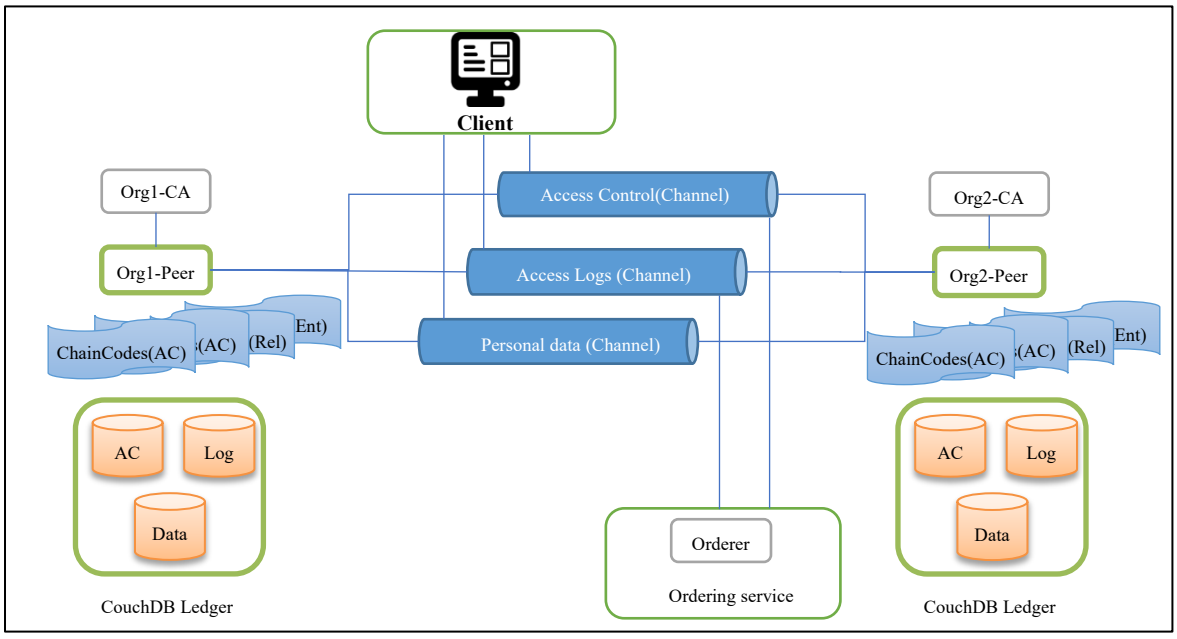



Figure 19:The Hyperledger Fabric Network used by Our Implementation

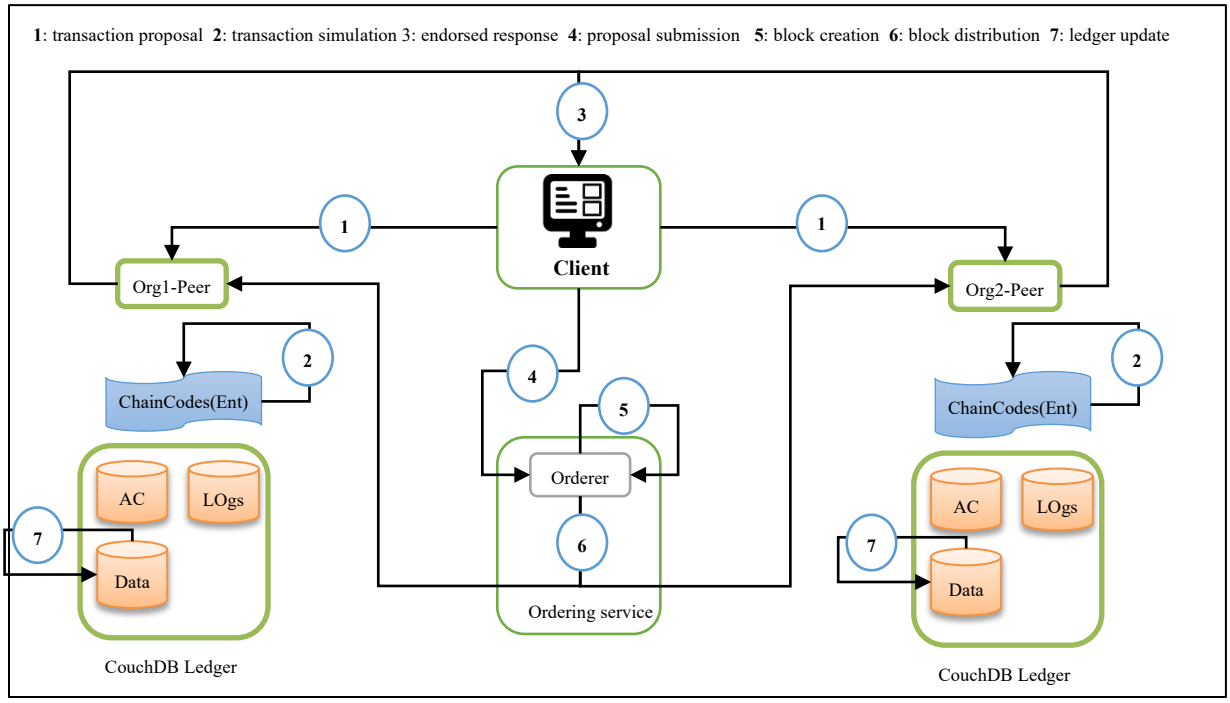



**Figure 20:Fabric Hyperledger Steps to Execute a New Transaction of the Chain-code Associated with Data Channel**

### 1.1.1. Access Control Chaincode

The AC chaincode define functions that executed by HLF peers for managing the KG access control and defining the permissions required to execute KG operations. This chaincode is installed on a channel identified by the same name “Access control” and it is associated with a local ledger that saves information about the access control on KG data. The AC chaincode uses the Golang structure which is illustrated by the figure 22 whereat the AC ledger uses the JSON key value format given by the figure 23 as a representation of the same GO structure. Thus, AC chaincode functions like creating new permission must use a JSON key value passed in the invocation call by the HLF users. The Go functions use the marshal and unmarshal methods to manipulate the JSON strings and store it on the AC ledger. All information included in the JSON string are already explained by the table 3 in the previous chapter.

```
type AccessControl struct {
        OwnerID string `json:"OwnerId"`
        PolicyData [] PolicyTypeData `json:"PolicyData"`
}
type PolicyTypeData struct {
        DataID string `json:"DataID"`
        DataPermissions [] DataPermissionType `json:"DataPermissions"`
}
type DataPermissionType struct {
        DataID string `json:"DataID"`
        DefaultPolicy [] string `json:"DefaultPolicy"`
        Permissions [] PermissionsType `json:"Permissions"`
}
type PermissionsType struct {
```

```
        Target string `json:"Target"`
        Op [] string `json:"Op"`
}
```

Figure 21:The Golang Structure used by the Access Control Chaincode

```
{
"OwnerId":"3",
"PolicyData":
[{"DataID":"2",
"DataPermissions":
[{"DataID":"2","DefaultPolicy":["Read"],
  "Permissions":[{"Target":"11",
  "Op":["Write","Read"]}]}]
}
```

Figure 22:The JSON key value Structure used by the Access Control Ledger

| Function | Description |
|---|---|
| **addPermission** | Create new permission for a given requester for accessing a given user data. |
| **checkPermission** | Check if a given requester has already registered with a given permission. |
| **updatePermission** | Change a given permission by removing or extending or restricting it. |
| **getAllPermission** | Get the set of all permissions stored in the access control ledger. |

Table 5: Some smart contract functions that are implemented by the access control chaincode

### 1.1.2. Access Log Chaincode

The Access Log chaincode define functions that executed by HLF peers for managing the transactions logs about requester access to user data along with executed permission. This chaincode is installed on a channel identified by the same name “Access Log” and it is associated with a local ledger that saves information about user rewards. The AL chaincode uses the Golang structure which is illustrated by the figure 24 whereat the AL ledger uses the JSON key value format given by the figure 25 as a representation of the same GO structure.

Thus, AL chaincode functions like saving new log must use a JSON key value passed in the invocation call by the HLF users. The Go functions use the marshal and unmarshal methods to manipulate the JSON strings and store it on the AL ledger.

All information included in the JSON string are already explained by the table 4 in the previous chapter.

```
type Logs struct {
        RequesterID   string `json:"RequesterID"`
        Tasks [] TaskType `json:"Tasks"`
}
type TaskType struct {
        OwnerID string `json:"OwnerId"`
        DataID string `json:"DataID"`
        DataID string `json:"DataID"`
        Operations [] string `json:"Operations"`
        Timestamp     time.Time `json:"timestamp"`
}
```

**Figure 23:The Golang Structure used by the Log Chaincode**

```
{
"RequesterID":"3",
"Tasks":  [{"OwnerID":"2","DataID":"2","DataID":"2","Operations":["Write","Read"]}]
}
```

**Figure 24:The JSON key value Structure used by the Log Ledger**

| Function | Description |
|---|---|
| **createLog** | Create new access log for a given requester. |
| **readLogReq** | Get the set of logs for a given requester. |
| **readLogOw** | Get the set of logs for a given data owner. |
| **getAllLogs** | Get the set of all access logs stored in the log ledger. |

**Table 6: Some smart contract functions that are implemented by the Reward chaincode**

### 1.1.3. Personal Data Chaincodes

The chaincode is created for managing critical information associated to the user data in the off-chain. The content of this ledger is already discussed in the table [] in the section []. The personal data chaincode uses the Golang structures that are illustrated by the figure 26 whereat this ledger uses the JSON strings given by the figure 27.

| Function | Description |
|---|---|
| **createData** | Create new critical data about a new personal data. |
| **readPersonalData** | Get the information stored in the data  ledger about a given KG entity. |

| | |
|---|---|
| **updateData** | Update the blockchain data about a given data such as: the hash and the off-chain ID. |
| **getAllData** | Get all data that are saved in the personal data ledger. |

**Table 7:Some smart contract functions that are implemented by the Entity chaincode**

```
type Data struct {
        DataNum int `json:"entity_num"`
        DataID string `json:"data_id"`
        ObjectID string `json:"object_id"`
        DataHash string `json:"data_hash"`
        }
```

**Figure 25:The Golang Structure used by the Entity Chaincode**

```
{
"DataNum":"10","DataId":"20","Object_id":"2","data_hash": "7221jkgf"
}
```

**Figure 26:The JSON key value Structure used by the personal data Ledger**

# 2. Evaluation

## 2.1.Experiment Configuration

To validate the functionality and test the performance of *PrivyKG*, a number of experiments have been performed. Experiments are performed on a machine with an Intel Core i7 processor running with a 1.8 GHz clock speed, 16 GB memory, 128 GB SSD and 1 TB for storage. The components of the fabric network are deployed as Docker 2.3 images (Organizations, certificate authorities, peers, CouchDB. etc.).

## 2.2.Security analysis

In this subsection, we provide a security analysis regarding authentication, authorization, confidentiality and, integrity. Table 1 illustrates the security objectives along with theirs proposed solutions.

| Security objectives | The proposed solution |
|---|---|
| **Confidentiality** | Encrypt data with secret key |
| **Authentication and Authorization** | Certificate (blockchain) + User name and password (off-chain) |

| | |
|---|---|
| **Integrity** | Hashing of the triples sets |

**Table 8:The Security requirement and the proposed solution**

We assume that an external adversary tries to make data modifications. In order to achieve this goal, the attacker must have the secret key and public key which has been used respectively to encrypt and decrypt the entity and relation names, the hashing strategy, the certificate to access the blockchain and the smart contact code, the user-name and password to access the off-chain storage. Therefore, it is very difficult to obtain all these information together. Any changing in the personal data can be detected using hash checking of triple sets.

# 9. Conclusion

In this article, we started by introducing the Hyperledger Fabric's architecture, how Fabric consensus and ordering service work, and how to deploy Fabric network in a production environment.

After that we presented the steps of implementation of our contribution to preserve privacy of Users Data Using Blockchain and Respecting the Algerian Regulation of Data Protection by apply that in the Case of University of Tebessa. Finally, we concluded the chapter by presenting the evaluations

# General Conclusion

## 1. Conclusion

The main objectif of this master thesis is to propose a new approach for protecting and keeping the privacy of the personal data with respect to the Algerian regulation of data protection. This is a new direction toward supporting the Algerian government for the goal of the application of its law of the data protection. Our work is based on the new technology called BlockChain which has been used in the crypto-currencies like Bitcoin. The use of this technology in the data protection is a new direction of the researchers in this field. Our work is implemented using Hyperledger fabric which is a permissioned blockchain designed for the business applications.

## 2. Future works

Our future works comprise several directions such as:

- Evaluate the system with real personal data from a specific domain like student personal data.

- Testing the proposed design with other blockchain systems like Ethereum which is public blockchain.
- Adding rewarding mechanism in order to reward the users for publishing their data.

# References

[1] L. Leloup, *Blockchain: La revolution de la confiance*. Editions Eyrolles, 2017.
[2] J.-P. Delahaye, “Les blockchains, clefs d’un nouveau monde’,” *Pour Sci.*, vol. 449, pp. 80–85, 2015.
[3] F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: Current status, classification and open issues,” *Telemat. Inform.*, vol. 36, pp. 55–81, 2019.
[4] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” *Decentralized Bus. Rev.*, p. 21260, 2008.
[5] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” 2017, pp. 557–564.
[6] B. Wang, S. Chen, L. Yao, B. Liu, X. Xu, and L. Zhu, “A simulation approach for studying behavior and quality of blockchain networks,” 2018, pp. 18–31.
[7] I. Bashir, *Mastering blockchain*. Packt Publishing Ltd, 2017.
[8] A. Rotsart de Hertaing, T. Hannesse, and O. de Broqueville, “" Les banques doivent-elles craindre les blocktechs et leur technologie blockchain?”.
[9] A. M. Antonopoulos, *Mastering Bitcoin: Programming the open blockchain*. O’Reilly Media, Inc., 2017.
[10] A. Kibet and S. M. Karume, “A Synopsis of Blockchain Technology,” *Int. J. Adv. Res. Comput. Eng. Technol. IJARCET*, vol. 7, no. 11, 2018.
[11] M. Mukhopadhyay, *Ethereum Smart Contract Development: Build blockchain-based decentralized applications using solidity*. Packt Publishing Ltd, 2018.
[12] M. Lamichhane, “A smart waste management system using IoT and blockchain technology,” 2017.
[13] M. Swan, “Blockchain: Blueprint for a New Economy.–O’Reill Media,” *Inc Sebastopol CA*, 2015.
[14] S. Ølnes, J. Ubacht, and M. Janssen, “Blockchain in government: Benefits and implications of distributed ledger technology for information sharing,” *Gov. Inf. Q.*, vol. 34, no. 3, pp. 355–364, 2017.
[15] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, “Blockchain-Based Database to Ensure Data Integrity in Cloud Computing Environments.,” 2017, pp. 146–155.

[16] العيدانـــي محمد , زروق يوسف, “حماية المعطيات الشخصية في الجزائر على ضوء القانون رقم 18-07 (المتعلق بحماية الأشخاص الطبيعيين في مجال معالجة المعطيات ذات الطابع الشخصي )” vol. 2, pp. 115–130. [Online]. Available: https://www.asjp.cerist.dz/en/article/73171
[17] “blockchain.html.” Accessed: May 24, 2022. [Online]. Available: https://hyperledger-fabric.readthedocs.io/it/latest/blockchain.html
[18] “actors-architecture-transaction-flow-in-hyperledger-fabric.” Accessed: May 24, 2022. [Online]. Available: https://walkingtree.tech/actors-architecture-transaction-flow-in-hyperledger-fabric/
[19] “demystifying-hyperledger-fabric-1-3-fabric-architecture-a2fdb587f6cb.” Accessed: May 24, 2022. [Online]. Available: https://medium.com/coinmonks/demystifying-hyperledger-fabric-1-3-fabric-architecture-a2fdb587f6cb
[20] “hyperledger-fabric-in-blockchain.” Accessed: May 24, 2022. [Online]. Available: https://www.geeksforgeeks.org/hyperledger-fabric-in-blockchain/
[21] Zyskind, G., Nathan, O., Pentland, A. (2015). Decentralizing Privacy: Using Blockchain

to Protect Personal Data. 2015 IEEE Security and Privacy Workshops, 180-184.

[22] Al Omar, A., Rahman, M.S., Basu, A., Kiyomoto, S. (2017). MediBchain: A Blockchain Based Privacy Preserving Platform for Healthcare Data. In: Wang, G., Atiquzzaman, M., Yan, Z., Choo, KK. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2017. Lecture Notes in Computer Science(), vol 10658. Springer, Cham. https://doi.org/10.1007/978-3-319-72395-2_49

[23] Mishra, R.A., Kalla, A., Braeken, A., & Liyanage, M. (2021). Privacy Protected Blockchain Based Architecture and Implementation for Sharing of Students' Credentials. Inf. Process. Manag., 58, 102512.

# Appendix A

# Access Control chaincode

```
package main

import (
	//"strconv"
	"encoding/json"
	"fmt"
	"log"
	"time"
  //"math"
	"github.com/golang/protobuf/ptypes"
	"github.com/hyperledger/fabric-chaincode-go/shim"
	"github.com/hyperledger/fabric-contract-api-go/contractapi"
   "github.com/hyperledger/fabric-chaincode-go/pkg/cid"
)

const Threshold = 8

// SimpleChaincode implements the fabric-contract-api-go programming
model
type SimpleChaincode struct {
	contractapi.Contract
}

type AccessControl struct {
	OwnerID string `json:"OwnerId"`
	PolicyKG [] PolicyTypeKG `json:"PolicyKG"`
}
type PolicyTypeKG struct {
	KgID string `json:"KgID"`
	DataPermissions [] DataPermissionType `json:"DataPermissions"`
}
type DataPermissionType struct {
	DataID string `json:"DataID"`
	DefaultPolicy [] string `json:"DefaultPolicy"`
	Permissions [] PermissionsType `json:"Permissions"`
}
type PermissionsType struct{
	Target string `json:"Target"`
	Op [] string `json:"Op"`
}

// QueryResult structure used for handling result of query
type QueryResult struct {
	Record              *AccessControl
	TxId                string    `json:"txID"`
	Timestamp           time.Time `json:"timestamp"`
	FetchedRecordsCount int       `json:"fetchedRecordsCount"`
	Bookmark            string    `json:"bookmark"`
}
func (s *SimpleChaincode) GetAllAssets(ctx
contractapi.TransactionContextInterface) ([]*AccessControl, error) {
	// range query with empty string for startKey and endKey does
an
```

```
	// open-ended query of all assets in the chaincode namespace.
	resultsIterator, err := ctx.GetStub().GetStateByRange("", "")
	if err != nil {
		return nil, err
	}
	defer resultsIterator.Close()

	var assets []*AccessControl
	for resultsIterator.HasNext() {
		queryResponse, err := resultsIterator.Next()
		if err != nil {
			return nil, err
		}

		var asset AccessControl
		err = json.Unmarshal(queryResponse.Value, &asset)
		if err != nil {
			return nil, err
		}
		assets = append(assets, &asset)
	}

	return assets, nil
}
// UpdateAsset updates an existing asset in the world state with
provided parameters.
func (s *SimpleChaincode) UpdateAsset(ctx
contractapi.TransactionContextInterface, id string, description
string) error {
	//assetBytes, err := ctx.GetStub().GetState(id)
	b := []byte(description)
	var asset AccessControl
	var err = json.Unmarshal(b, &asset)
	assetJSON, err := json.Marshal(asset)
	if err != nil {
		return err
	}

	return ctx.GetStub().PutState(id, assetJSON)
}
// DeleteAsset deletes an given asset from the world state.
func (s *SimpleChaincode) DeleteAsset(ctx
contractapi.TransactionContextInterface, id string) error {
	return ctx.GetStub().DelState(id)
}
func (s *SimpleChaincode) MultipleAssets(ctx
contractapi.TransactionContextInterface, description string) error {
val, ok, err := cid.GetAttributeValue(ctx.GetStub(), "role")
if err != nil {
	return err
	 // There was an error trying to retrieve the attribute
}
if !ok {
	 // The client idAccessControl does not possess the attribute
	 return fmt.Errorf("The client idAccessControl does not possess
```

```
the Role attribute")
}
      if val != "client" {
            return fmt.Errorf("The client must have the role as
Client to complete the chaincode invokation")

      }
var rel []AccessControl
json.Unmarshal([]byte(description), &rel)
for k := range rel{
      assetJSON, err := json.Marshal(rel[k])
      if err != nil {
            return nil
      }

      ctx.GetStub().PutState(rel[k].OwnerID, assetJSON)
}
return nil
}
// ReadAsset retrieves an asset from the ledger
func (t *SimpleChaincode) ReadAsset(ctx
contractapi.TransactionContextInterface, assetID string)
(*AccessControl, error) {
      assetBytes, err := ctx.GetStub().GetState(assetID)
      if err != nil {
            return nil, fmt.Errorf("failed to get asset %s: %v",
assetID, err)
      }
      if assetBytes == nil {
            return nil, fmt.Errorf("asset %s does not exist",
assetID)
      }

      var asset *AccessControl
      err = json.Unmarshal(assetBytes, &asset)
      if err != nil {
            return nil, err
      }

      return asset, nil
}

func (t *SimpleChaincode) ReadAssetHistory(ctx
contractapi.TransactionContextInterface, assetID string)
([]AccessControl, error) {
      var history []AccessControl
            resultsIterator, err :=
ctx.GetStub().GetHistoryForKey(assetID)
            if err != nil {
                  return nil, err
            }
            defer resultsIterator.Close()
            for resultsIterator.HasNext() {
                  var asset AccessControl
            historyData, err := resultsIterator.Next()
```

```
            err = json.Unmarshal(historyData.Value, &asset)
    history = append(history, asset)
            if err != nil {
                  return nil, err
            }
      }
            return history, nil
      }

func (t *SimpleChaincode) CreateAsset(ctx
contractapi.TransactionContextInterface, assetID string, description
string) error {
      exists, err := t.AssetExists(ctx, assetID)
      if err != nil {
            return fmt.Errorf("failed to get asset: %v", err)
      }
      if exists {
            return fmt.Errorf("asset already exists: %s", assetID)
      }

      b := []byte(description)
      var asset AccessControl
      err = json.Unmarshal(b, &asset)
      assetBytes, err := json.Marshal(asset)
      if err != nil {
            return err
      }

      return ctx.GetStub().PutState(assetID, assetBytes)
      }
// constructQueryResponseFromIterator constructs a slice of assets
from the resultsIterator
func constructQueryResponseFromIterator(resultsIterator
shim.StateQueryIteratorInterface) ([]*AccessControl, error) {
      var assets []*AccessControl
      for resultsIterator.HasNext() {
            queryResult, err := resultsIterator.Next()
            if err != nil {
                  return nil, err
            }
            var asset *AccessControl
            err = json.Unmarshal(queryResult.Value, &asset)
            if err != nil {
                  return nil, err
            }
            assets = append(assets, asset)
      }
      return assets, nil
}

// GetAssetHistory returns the chain of custody for an asset since
issuance.
func (t *SimpleChaincode) GetAssetHistory(ctx
contractapi.TransactionContextInterface, assetID string)
([]QueryResult, error) {
```

```
	resultsIterator, err := ctx.GetStub().GetHistoryForKey(assetID)
	if err != nil {
		return nil, err
	}
	defer resultsIterator.Close()

	var records []QueryResult
	for resultsIterator.HasNext() {
		response, err := resultsIterator.Next()
		if err != nil {
			return nil, err
		}

		var asset *AccessControl
		err = json.Unmarshal(response.Value, &asset)
		if err != nil {
			return nil, err
		}

		timestamp, err := ptypes.Timestamp(response.Timestamp)
		if err != nil {
			return nil, err
		}
		record := QueryResult{
			TxId:      response.TxId,
			Timestamp: timestamp,
			Record:    asset,
		}
		records = append(records, record)
	}

	return records, nil
}

// AssetExists returns true when asset with given ID exists in the
ledger.
func (t *SimpleChaincode) AssetExists(ctx
contractapi.TransactionContextInterface, assetID string) (bool,
error) {
	assetBytes, err := ctx.GetStub().GetState(assetID)
	if err != nil {
		return false, fmt.Errorf("failed to read asset %s from
world state. %v", assetID, err)
	}

	return assetBytes != nil, nil
}
func main() {
	chaincode, err := contractapi.NewChaincode(&SimpleChaincode{})
	if err != nil {
		log.Panicf("Error creating asset chaincode: %v", err)
	}

	if err := chaincode.Start(); err != nil {
		log.Panicf("Error starting asset chaincode: %v", err)
```

```
        }
}
```