THE PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University Mohamed khieder – BISKRA

Faculty of Exact Sciences and Natural and Life Sciences Computer Science department

Thesis

Presented for the diploma of

Master in Computer Science

Option: **Information and communication networks and technologies.**

Blockchain for The Drug Supply Chain Management

Presented in / /
By **Sirine HAMLAOUI**

Board of Examiners :

Mr. Guerrouf Fayçal	**Supervisor**	
Mr.	**Examiner**	
Mr.	**Examiner**	BISKRA

September 2020

Abstract

Algerian industrial companies face a host of changes and challenges that would affect their various activities and functions, especially in the field of health care. The pharmaceutical supply chain is one of the most prominent sectors when it comes to areas affected by the health supply chain. Where drug companies that manufacture, ship and supply products encounter difficulties in tracking their products, because the distribution processes are not transparent and users do not have any access to the data flow, given the use of traditional systems through data control in one authority; Therefore, the data is subject to change and there is no guarantee that the system administration does not change the data to achieve the desired result. In this thesis, the main objective was to create a new simple system to ensure the transparency of the product distribution structure and to see all the information that was recorded over a new technology called blockchain to overcome the problems and challenges mentioned above. The ability of blockchain systems to identify the origin of data makes them particularly suitable for pharmaceutical supply chain applications. The app has a blockchain ethereum on top and a front end that allows users to interact with the system. In the system, all information related to sales and purchases of products is recorded and all transactions that occurred in the system are saved on the blockchain starting from the manufacturers to the pharmacies and hospitals. This system allows companies to track their trade by enhancing transparency in the supply chain, as well as reducing management costs by automatically recording distribution details in the blockchain network and managing information more securely.

Résumé

Les entreprises industrielles algériennes sont confrontées à une multitude de changements et de défis qui affecteraient leurs diverses activités et fonctions, notamment dans le domaine de la santé. La chaîne d'approvisionnement pharmaceutique est l'un des secteurs les plus importants en ce qui concerne les zones touchées par la chaîne d'approvisionnement en santé. Lorsque les sociétés pharmaceutiques qui fabriquent, expédient et fournissent des produits rencontrent des difficultés pour suivre leurs produits, parce que les processus de distribution ne sont pas transparents et que les utilisateurs n'ont aucun accès au flux de données, étant donné l'utilisation de systèmes traditionnels par le biais du contrôle des données dans une seule autorité; Par conséquent, les données sont susceptibles d'être modifiées et il n'y a aucune garantie que l'administration du système ne modifie pas les données pour obtenir le résultat souhaité. Dans cette thèse, l'objectif principal était de créer un nouveau système simple pour assurer la transparence de la structure de distribution des produits et de voir toutes les informations qui ont été enregistrées sur une nouvelle technologie appelée blockchain pour surmonter les problèmes et défis mentionnés ci-dessus. La capacité des systèmes blockchain à identifier l'origine des données les rend particulièrement adaptés aux applications de la chaîne d'approvisionnement pharmaceutique. L'application a un Ethereum blockchain sur le dessus et un frontal qui permet aux utilisateurs d'interagir avec le système. Dans le système, toutes les informations relatives aux ventes et aux achats de produits sont enregistrées et toutes les transactions qui ont eu lieu dans le système sont enregistrées sur la blockchain à partir des fabricants jusqu'aux pharmacies et hôpitaux. Ce système permet aux entreprises de suivre leur commerce en améliorant la transparence dans la chaîne d'approvisionnement, ainsi qu'en réduisant les coûts de gestion en enregistrant automatiquement les détails de distribution dans le réseau blockchain et en gérant les informations de manière plus sécurisée.

DEDICATION

First, I dedicate my dissertation work to the sake of Allah, my Creator and my Master. My great teacher and messenger, Mohammed (May Allah bless and grant him), who taught us the purpose of life.

I dedicate my dissertation to all my family. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. My sisters and my brother have never left my side and are very special.

I also dedicate this dissertation to all family HAMLAOUI and my many friends who have supported me throughout the process. I will always appreciate all they have done.

May ALLAH (SWT) grant them Jannah Firdaus.
Ameen

ACKNOWLEDGEMENTS

Prima facie, I am grateful to ***ALLAH*** *for the guidance, good health, wellness and willpower that were necessary to complete this thesis.*

I would like to thank ***my Parents****, who always believed in me. It is thanks to their support and prayers that I accomplished this work, they already know how much I owe them.*

I also would like to thank my supervisor, Professor ***Guerrouf Fayçal****, whose expertise was invaluable in formulating the research methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.*

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

Finally, I also want to thank the jurors for agreeing to review and judge my work.

Contents

List of Figures

List of Tables

General Introduction

The pharmaceutical industry is the bit of the healthcare sector that deals with medications. The industry includes various sub-fields related to drug development, production and marketing. The primary goal of the pharmaceutical industry is to provide drugs that prevent infection, maintain health and treat diseases [69].

The supply chain of the pharmaceutical industry is like a supply chain for any other industry in the manufacturing sector. The activities of a pharmaceutical supply chain involve the flow and transformation of medicines from raw materials through to the end users. In addition, the associated information flows through the relationships in the supply chain to achieve a sustainable competitive advantage. Pharmaceutical supply chain management is more difficult than typical applications within industrial companies, as medicines and surgical supplies must be available for use at all times [85].

Supply chain management can be defined as the management of the flow of products and services, which begins with the origin of the products and ends when the product is consumed [42]. It also includes the movement and storage of raw materials that are used in the work in operation as most of the products are subject to manufacturing and shipped from one place to another.

The supply chain represents a network of relationships within the company and business organizations consisting of raw material suppliers, manufacturers, shippers, third party logistics, retailers and related parties involved, which facilitates the reverse production of materials, services and information from the original product to the final customer, while increasing efficiencies, and achieving customer satisfaction. As It is difficult to obtain a comprehensive picture of all the operations that takes place in the network and integrate all parties involved. What this system currently suffers is a lack of efficiency and transparency and tracking errors resulting from low quality or in the presence of a specific problem in parts of the product, time, cost and the negative effects that it has on the relationship of supply chain companies [43].

However, the current model makes it difficult to maintain a consistent and efficient supply chain system. Now, there is a need for an effective and reliable system to conduct, record and save transactions, to create a state of improvement and a tremen-

dous shift in the way of producing, marketing, buying and consuming products and developing the industry and increasing its efficiency.

Problem

Every day, billions of products are being manufactured and delivered all over the world. Meanwhile, the products raw materials are not produced in a single company, and commonly, the components are originated from different manufacturers. The supply chain in some pharmaceutical industries in many countries Especially Algeria, has faced many difficulties in tracking products and commodities, which affects the growth of the industry, its reputation and financial affairs [89], and this may be due to information that is inaccurate and not always available, or the inability to work transparently and track errors caused by low quality or the presence of a specific problem in the distribution stages, as well as the lack of reliable technology through which information can be combined in a safe and fast manner. In the supply chain, many parties, including distributors, retailers and the insurance company, are involved, through a network, in completing the chain. All parties in the network are controlled by a main or central system. However, some drug supply chains may be concerned about the source of materials and products, and the need to replace traditional client-server architectures has been raised for the following reasons:

- Documentation of records causes errors, delays, and conflicts of information to track down shipment delays, theft, or some unexpected issues that can occur at some point in the supply chain.
- Only one central server accepts all arriving requests resulting in low performance and high server maintenance cost.
- The system is weak and easily compromised because the central authority is the only point prone to failure, attack, and penetration.
- A critical issues with traditional networks is the transparency and traceability of source of the products because the users do not own or control any data.
- The information presented by the central authorities cannot be trusted because there is no way to confirm that the data has not been tampered with.

Solution

There are different studies which have tried to overcome these problems and increase transparency and visibility of supply chain [76][127], in addition to eliminate the need

of central authority through using a new technologies like Blockchain, that can make a big shift in the supply chain operations more efficient, transparent, and secure, and because it is a distributed ledger technology which guarantees trust, transparency and security.

Blockchain technology was introduced in 2008 under the pseudonym Satoshi Nakamoto to solve the problem of double spending and trust without third-party dependency "through Bitcoin", the first app to support simple transactions. It's use in cryptocurrencies did not stop there, but can be used for more than that, in its essence, it represents a general record of any type of transaction that can be distributed to all parties of the network instead of one person controlling everything. It has been used widely in various fields. Specifically, in the supply chain, visualization ensures product integrity and helps participants have a better view of the products' life cycle. It also allows participants to discover relationships.

In this thesis, we have attempted to adopt blockchain technology as a decentralized distributed network to provide participants record price, date of distributing and delivering, distributed quantities, and other relevant information to more effectively manage the supply chain. We suggested a simple system that contains three main entity types; i) Users, ii) the Smart Contract, and iii) the Blockchain Network. We tried to increase visibility, trace-ability of supply chain and lower losses. The blockchain-based smart contracts, which are automated software, are used in the proposed system to be executed on every node on the network. Therefore, blockchain technology is used to give organizations the ability to exchange distributed data and transportation without any centralized authority. Therefore, the involved parties in the supply chain may have direct interactions with each other, and trust is not required anymore. These secure direct communications lead to greater transparency, clarity, and efficiency while also reducing the cost and risk of failure in the shipment tracking process.

Organization of the thesis

This thesis is organized as follow :

- **Introduction**: We will start our thesis with an introduction to the context of this work, the targeted problem, and the solution we propose.
- **Chapter One: Blockchain Technology**

  In this chapter, we will first introduce the definitions, characteristics, the main components and types of blockchain technology. Next, we will indicate how the blockchain working with its advantages and disadvantages. We will also present the most challenges and limitations that faces of blockchain technology, and its

different uses in our lives. Finally, we will present the most implementations of blockchain.

- **Chapter Two: Drug Supply Chain**

  In this chapter, we will first present the definition of supply chain management and its processes. In addition, to present the importance and benefits of supply chain. Then, we will present the drugs supply chain and distribution processes and how its work in Algeria. Finally, we will offer the main challenges and problems that faces the drug supply chain in Algeria.

- **Chapter Three: Design and Implementation**

  In this chapter, we will present the proposed system, component model that define the drug supply chain and implementation of this system.

- **Conclusion**: We will finish our thesis with a general conclusion and some perspectives and future directions.

Chapter 1

Blockchain Technology

1.1 Introduction

The world is attending a transitional phase from the industrial economy to the economy defined by a new set of technologies, ranging from digital technology to precision technology. Among the latest digitization waves is the blockchain technology, which guarantees the safety standards for cash transactions in a completely decentralized system and provides a platform for storing data in a secure and difficult to penetrate using the encryption techniques and the consensus [109].

Tens of financial institutions and many companies from various sectors have been hard at work demonstrating the universal applications of blockchain technology to reduce transaction costs, accelerate their operations, reduce fraud risk and eliminate intermediary services. In this chapter, we will be learning about all the major sides of blockchain technology.

1.2 Definition Blockchain Technology

1.2.1 Definition

Blockchain is known from the beginning of its creation as a series of blocks used to store information on them, and has generally been known as "a technology for storing and transmitting information, transparent, secure, and operates without a centralized control body". Jean-Paul define blockchain as :

*"**The idea of a large computer notebook, shared, unfalsifiable and indestructible by the very fact of its conception is at the heart of a new revolution, that of the blockchain** "* [109].

With other definition : *" **The blockchain is an incorruptible digital ledger of economic transaction that can be programmed to record no just finan-***

cial transactions but virtually everything of value ". This statement is one of the most popular definition of the blockchain, which is developed by Don and Alex Tapscott [67].

Blockchain is essentially a distributed database of transactions that contains the history of all exchanges that have occurred between its users since its inception. It is also known as the general ledger for all digital transactions or events that have been executed and shared between the parties involved. Each transaction in the public ledger is verified by consensus of a majority of system participants, everyone can access it but cannot edit it, this is the main feature.

1.2.2 Blockchain Features

Several characteristics are associated with the blockchain : peer-to-peer network, decentralized, transparency, distributed consensus, ineffaceable, distributed structure, resilience ,autonomy, security and trust [109]. All these characteristics constitute the innovative potential and modern possibilities of the blockchain. We mention the most important characteristics of the Blockchain are :

1. ***Peer-to-Peer Network*** : The peer-to-peer is a group of independent computers called nodes which are interconnected with each other to share data without use of the centralized computer [8]. In blockchain network, transactions take place directly between two nodes in the network, meaning no third party is required. It is verified by all other nodes in the blockchain. This is known as the Peer-to-Peer network.

2. ***Decentralization*** : Blockchain system is decentralized and distributed. That is, there is no central entity that controls and manages the blockchain, but network node can be access and check all the transactions because each node has a copy of the public ledger. This is advantage makes blockchain more secured.

3. ***Security*** : The highest level of data storage security. With decentralize, the blockchain is using cryptography for protection of users. Each informations on the blocks are hashed cryptographically using mathematical algorithms.

4. ***Consensus*** : Blockchain decentralization is a core power, as a copy of the database file belongs by all actors. In order to secure the integrity of each copy, a consensus algorithm is necessary. The consensus algorithm allows the community to secure that each added block is legitimate. It also prevents attackers from compromising and forking the chain [107]. Nakamoto suggested using a proof-of-work approach, in which a hard cryptographic puzzle must be solved by

miners [99]. With other consensus models such as proof-of-stake, proof-of-burn, proof-of-elapsed-time. We will explain in more details in section 1.4.3.

5. ***Autonomy*** : The computing strength and hosting space is supplied by network nodes, i.e. users themselves. So there are no need for central infrastructure. Within the blockchain, the infrastructure is no longer concentrated in the hands of an enterprise, on the contrary, it is dispersed at all points of the network [109].

6. ***Transparency*** : The blockchain is called transparent because anyone can download it in its entirety and check its honesty at any time [111]. All blockchain users can thus view actual and past transactions[109]. If transparency is ensured for transactions, user anonymization calls into question this characteristic. In fact, the possible anonymity on the blockchain can be used for deceitful activities, hard or even impossible to disclose and regulate.

1.3 History of Blockchain

Blockchain technology has to be one of the greatest inventions of the 21st century due to its emerging impact on different sectors, from the financial sector to industrialization as well as education. Unknown many, is that Blockchain history dates back to the early 1990's [79].

In 1991 ,Stuart Haber and W. Scott Stornetta introduced the concept of a chain of blocks and working on a cryptographically secured chain whereby no one could tamper with timestamps of documents.

In 2008, the Blockchain History starts to gain importance, due to the work one person or group unknown under the pseudonym of Satoshi Nakamoto who worked on the first Blockchain from where the technology has evolved and found its way into many applications beyond cryptocurrencies .

In 2009, Satoshi Nakamoto introduced a Bitcoin white paper detailed it as an electronic peer-to-peer system, the first application of the digital ledger technology.

In 2013, Vitalik Buterin is a programmer and co-founder of the Bitcoin magazine stated and as one of the first contributors to Bitcoin codebase.

Concerned by Bitcoin's limitations, Buterin started working on a flexible blockchain that can perform different functions in addition to being a peer-to-peer network. Ethereum was born out as a new public blockchain with added functionalities compared to Bitcoin such as to record other assets such as logos and contracts. The new feature expanded Ethereum functionalities from being a cryptocurrency to being a platform for improving decentralized applications as well. It was officially launched in 2015.

Blockchain History and evolution does not stop with Ethereum and Bitcoin. In recent years, a number of projects have developed all the potential blockchain technology. New projects have sought to address some of the deficiencies of Bitcoin and Ethereum in addition to coming up with new features leveraging blockchain capabilities.

Some of the new blockchain applications include NEO, announced as the first open source, decentralized and blockchain platform launched in China. Even though the country has banned cryptocurrencies, it remains active when it comes to blockchain innovations. NEO casts itself as the Chinese Ethereum .

In the race to accelerate development of the Internet of Things, some developers, so it fit, to leverage blockchain technology and in the process came up with IOTA.

In addition to IOTA and NEO, Monero Zcash and Dash blockchains have emerged as a way to addressing some of the security and scalability issues associated with the early blockchain applications. Called as privacy Altcoins, the three blockchain platform request to provide high levels of privacy and security when it comes to transactions.

1.4 Blockchain Building

A blockchain is a way to store information. On the other hand, it is a protocol for transferring financial transactions and almost everything of value. A protocol is a set of rules that define the criteria for how participants communicate. To understand how blockchain technology is used, it is necessary to comprehend the components that go into the blockchain ecosystem and what each component does [59]. There are major components for any blockchain ecosystem as follows:

1.4.1 Transaction

Transaction means a series of information exchange and related work such as database updating, that is addressed as a unit for the objective of satisfying a request and for ensuring database integrity.

In blockchain, a transaction is a transfer value that is broadcast to the network and collected into blocks. Transactions contain one-or-more inputs and one-or-more outputs.

- An **input** is a reference to an output from a previous transaction.
- An **output** specifies an amount and a address.

A transaction usually references past transaction outputs as new transaction inputs and devotes all input values to new outputs. Transactions are not encrypted, so it is possible to scan and view every transaction ever collected into a block. With just enough confirmations of transactions, they can be considered irreversible.

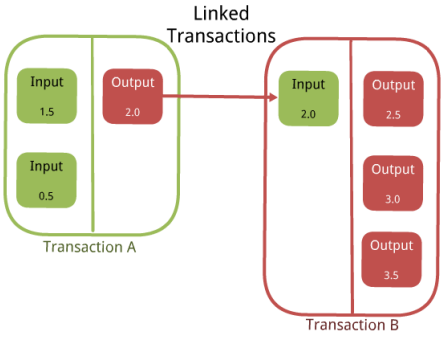


Figure 1.1: Linked Transactions

1.4.2 The Blocks

Block is a term used to describe anything that is bounded. It is a close set of bits or bytes that make up a specific a data unit. The term is used in database management, word processing, and network communication.

As we know, blockchain is made up of a series of interconnected blocks. Each block has several components which we will discuss in this section, figure 1.2 shows this components :

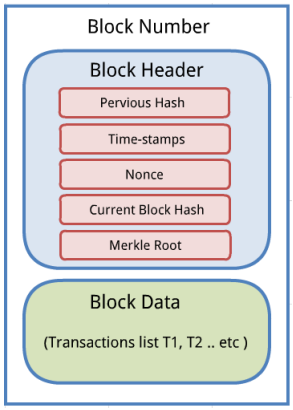


Figure 1.2: Components of Block

A one block in the blockchain has a Block header, Block number and Block data.

1. **Block Header**

   The block header is an important part as it has all the essential that contains metadata for this block.These data are as follows :

   - **Previous block hash :** is hash of the preceding block which use to link it with the next block.This hash of block creates using cryptographic methods of SHA256 algorithm, its size is 32 bytes.
   - **Timestamp :** is a time of the creation of the block and time of records a transactions, its size 4 bytes.

- **Nonce :** is a random number needed for consensus process ,its size 4 bytes.
- **Current block hash :** is hash of the actual block and its size is 32 bytes.
- **Merkle Root :** is made up of all of the hashed transaction hashes within the transaction. we will define it in section Bitcoin Protocol.

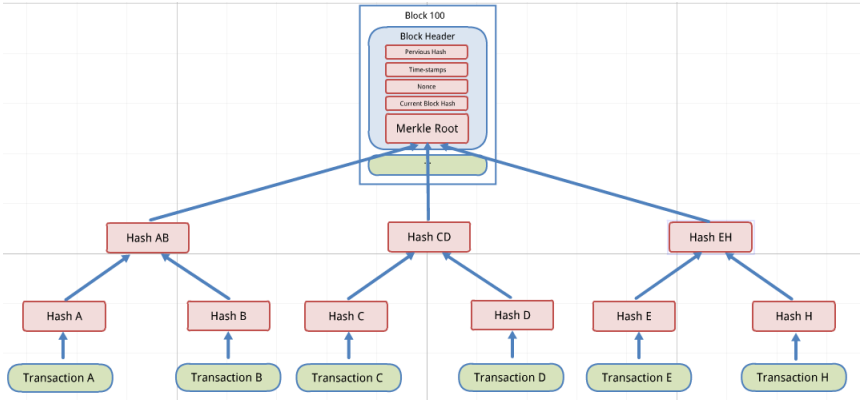


Figure 1.3: Merkle Tree

2. **Block Data**

   The Block Data is a list of transactions and ledger events included within the block.

Genesis Block: There is one special case to this, the first block in any blockchain is named the Genesis Block. It is the only and solely block which differs lightly from all the other blocks such as it's the single block with no precedent block . Every node in the network can identify the genesis block's hash and structure, the fixed time of creation, and the single transactions within.

1.4.3 Consensus Process

Blockchains are peer-to-peer networks with no central manager or authority. It is definitive to assure that the network entrants arrive consensus on the state of the ledger, the verification and order of records. By use consensus algorithms that apply different methods to ensure that the right order of transactions has been fixed and validated by due users to be added to the ledger [80]. We will describe some of the methods below [86].

1.4.3.1 Proof of Work (PoW)

Proof of Work was the first consensus algorithm to be created and used by Bitcoin and other cryptocurrencies, it is an essential algorithm in the mining process. Where

proof of work depends on the blockchain members to reach a consensus. Transactions through retail jobs, which miners perform, are resolved to add new additions to the chain. It creates an incentive by rewarding them with Bitcoin for every transaction they confirm. But there is only problem with using proof of work to confirm blockchain consensus protocols is that the mining act requires very high computational power and electricity. This makes the systems used to extract bitcoin very expensive. Figure 1.4 shown process of proof of work.

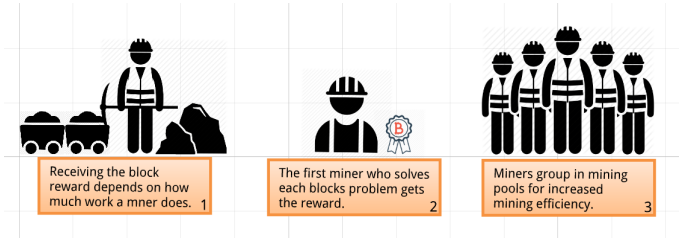


Figure 1.4: Proof of Work.
[104]

1.4.3.2 Proof of Stake (PoS)

The PoS Consensus algorithm was developed in 2011 as an alternative to PoW. Although PoS and PoW share similar goals, they do offer some basic differences and characteristics. Especially during checking new blocks. It is currently being developed in Ethereum Blockchain. Proof of Stake consensus algorithm replaces PoW mining with a mechanism by which the blocks are verified according to the share of the participants. The validator for each block is determined by development the cryptocurrency itself and not by the amount of computational power specified. Each PoS system may execute the algorithm in several ways, but in general, an individual can either mining or agree to a transaction based on the number of currencies he holds by voting. This means that the more Bitcoin or altcoin used by the wallet, the more voting power the user will has. Figure 1.5 shown process of proof of stake.

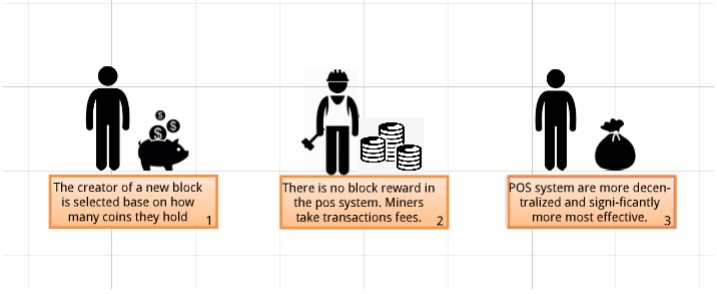


Figure 1.5: Proof of Stake.
[104]

1.4.3.3 Proof of Authority (PoA)

The PoA consensus algorithm differs somewhat from the rest of the algorithms because it does not require any mining, unlike PoW or PoS. In the PoAuthority, all transactions and blocks are vetted with approved accounts also known as validators. Transactions and block creation are executed automatically, using only the validators computing power.

1.4.4 The Hash

In simple terms, hashing means taking an input chain of any length and giving out an output of a fixed-length. In the state of cryptocurrencies like bitcoin, the transactions are taken as input and execute through a hashing algorithm (Bitcoin uses SHA-256) which gives an output of a fixed length [58]. We will explain with details in next chapter. The miners has to determine in order to ' find ' a block and it is done as follows [56] :

1. A random number is guessed 'Nonce'.
2. The Nonce is added at the end of all the data in the block.
3. This is all hashed according to SHA256 method.
4. If this hash starts with a predetermined number of zero's a new block is found. If not, the miner has to start again at step 1 with guessing another Nonce.

1.4.5 A Miners or Nodes

the CPU that tries to solve a hard math problem in order to be able to find a new block is called a miner or a node [56]. It has the ability to create and submit a new blocks to the chain. Which miner is allowed to produce a specific block may be be predetermined, or miners may simultaneously compete to add the next block to the chain, e.g. In the case of the Bitcoin network, miners execute works similar to bank teller, checking that a particular transfer of bitcoins is between two valid accounts, validating that the sender's signatures are original, and the sender owns the currency that are being transmitted.

1.4.6 The Chain

When the new block is full, it is attached to the previous block via a smart hashing process that makes blocks more and more sure the longer they have been a part of the chain. Once the new block is added to the chain, it is not possible to change any of the

previous data. This makes the data immutable, because transactions and blocks are regrouping in the order they're received. The blockchain builds forms a chronological register of activities. Much like a register in accounting, it shapes a ledger. Figure 1.6 shows a generic chain of blocks [130].

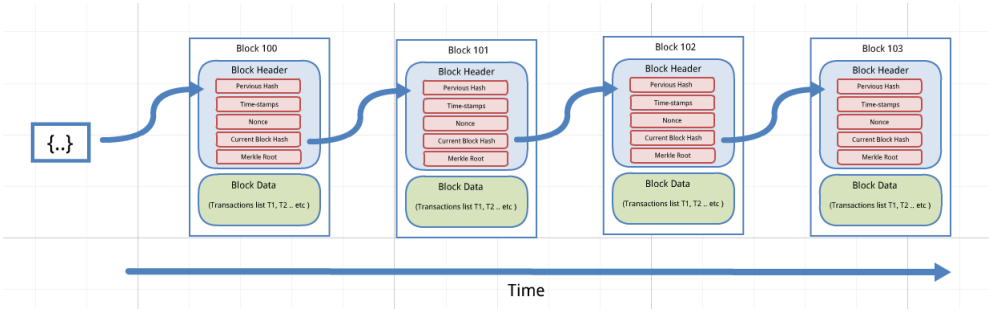


Figure 1.6: Generic Chain of Blocks

1.4.7 Smart Contracts

1.4.7.1 Definition

The notion of smart contract has been in existence since 1996. It was proposed by Szabo, which stated: ***"A smart contract is a set of promises, specified in digital form, including protocols within which the parties perform on these promises"*** [119].

Smart contract (or cryptocontract) is computer software that works on a blockchain contract and is executed directly and automatically in transferring digital assets between the parties under certain conditions. The first successful blockchain-based smart contract application was Bitcoin Script [13], an intentionally unfinished language with a set of simple pre-defined commands. As simple forms of smart contracts, standard types of Bitcoin transactions, such as pay-to-public-key-hash (P2PKH) and pay-to-script-hash (P2SH), are all defined using Bitcoin Script [60].

Additionally, there also exist platforms that offer more complex contractual functionalities and flexibilities, e.g., Ethereum [129], which adopts the full language of smart contracts. Latest blockchain platforms such as Neo [12] and Hyperledger Fabric [11] allow smart contracts to be written in different high-level languages. The following figure 1.7 shown the evolution of smart contracts [83].

1.4.7.2 Smart Contract Working

Smart contract development is linked to blockchain technology. Which is just a digital node with blockchain security encryption. Smart Contract contains details and per-

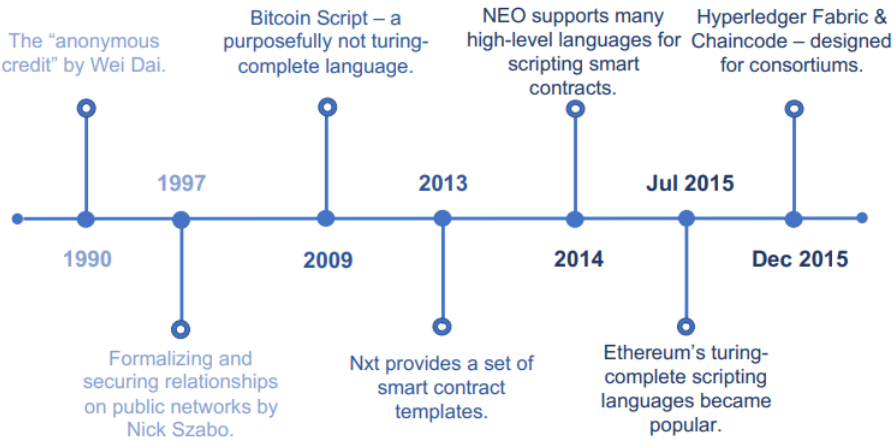


Figure 1.7: Evolution of Smart Contracts.

missions written in code with time constraints that can provide deadlines for contract. The smart contract system is illustrated in figure 1.8.

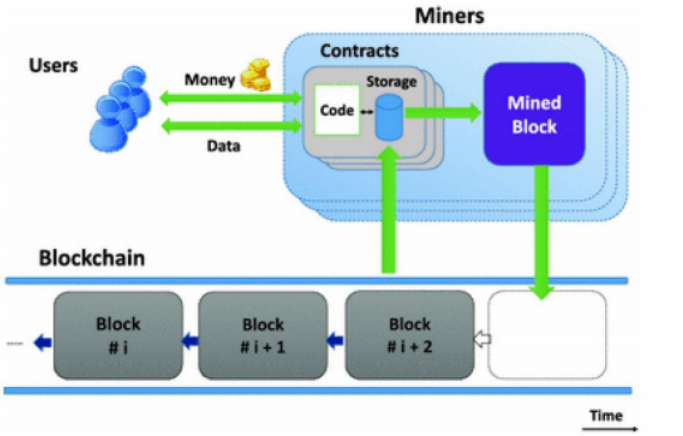


Figure 1.8: Smart Contract System [74].

Bitcoin was the first to use smart node by using it to transfer values from one person to another. In order for the contract to work, it is necessary for the parties to apply electronic signature techniques and also implement major specific conditions such as checking if the amount of value to transfer is actually available in the sender account, otherwise it will be impossible to automate the process.

At another time, Ethereum emerged which was considered more powerful, that given the ability of developers and programmers to implement custom contracts in a Turing-complete language [124], on the contrary in the case of Bitcoin it was written in an Turing-incomplete language - which limits contracts implementation in the Bitcoin network [116] [82].

To publish a smart contract in Ethereum [115], a special creation transactions are

performed. As the contract is assigned a unique address and its code is loaded to the blockchain. Accordingly, the smart contract is determined by the contract address [128]. The Ethereum address is assigned an identity for each individual involved in the transactions. Each contract holds some virtual currency and is linked to a predefined executable code using mechanism of encryption. The initiator pushes a fee for its implement. Smart contracts automatically implement the specified contract terms. The parties reach agreement on the contents of the contract and implement the contracts according to the terms written in certain computer algorithms. Through which smart contract can validate the deal, if not, the transaction is rejected [117]. Smart contracts are self-executable and self-verifiable factors that cannot be changed once published in a blockchain.

1.4.7.3 Features Smart Contract

Blockchain-based smart contract has become a growing field in the blockchain technology. Where these contracts already possess multiple advantages over traditional arrangements. We will present some advantages [114] :

- **Autonomy** : The technical architecture of smart contracts offers possibilities ranging from automatic self-help to the enforcement of legally unenforceable agreements. Incidentally, this also knocks the danger of manipulation by a 3rd party, since execution is managed mechanically by the network.
- **Trust** : Smart contracts generate absolute trust in their implementation by encrypting documents on the shared ledger. The transparent, autonomous and secure of the contract takes off any possibility of manipulation, alteration or error.
- **Security** : Smart contracts use the highest level of data encryption actually available, as are cryptocurrencies. Where their level of protection is among the best and most secure on the network.
- **Savings** : Smart Contract can save money since they eliminate the need for having a vast chain of middlemen. There's no need for advocate, witnesses, banks and other intermediaries. This is one of most important benefits of a Smart Contract.
- **Speed** : Smart contracts use software code to automate tasks online. As a result, they can perform transactions very speedily. This speed can provide many hours when compared to traditional contracts processes.

- **Backup** : Smart contracts are used to save necessary details of each transaction, they are permanently stored for future reference. Therefore, when someone lose their savings account they can be easily restored.

1.5 Cryptography in Blockchain

Encryption is a method of concealment and disclosure, usually known to encrypt and decrypt data or message content via mathematical rules. It builds and uses rules that prevent outside parties or the public from reading encrypted messages for the sake of information security [80]. In the blockchain, cryptography is used to achieve the following two goals:

- Secure the identity of the sender of the transactions.
- Ensure that previous records cannot be tampered with.

Information security uses cryptography in Blockchain on many levels. Data security, which the data cannot be scan while not a key to rewriting it. The data maintains its integrity throughout transit and whereas being held on. Blockchain Cryptography additionally aids in non-repudiation. This implies that the sender and also the delivery of a message is verified, including confidentiality and privacy [57].

1.5.1 Types of Blockchain Cryptography

There are three types of commonly used blockchain cryptography technologies namely, symmetric-key cryptography, public-key cryptography, and hash [3] [57].

1.5.1.1 Symmetric-key Cryptography

Both the sender and the receiver share a single key. This shared key is used for both encryption, which the sender uses to write plain text and send the encrypted text to the recipient as well as the recipient uses an equal key to decrypt the message and recover the plain text.

1.5.1.2 Public-Key Cryptography

This encryption method uses a pair of keys, an encryption key named public key, and a decryption key named private key. The key pair generated by this algorithm consists of a private key and a unique public key that is generated using the same algorithm.

1.5.1.3 Hash Functions

This type of encryption does not use keys. Uses cryptography to create a fixed-length hash value from plain text. It is almost impossible to retrieve the contents of the plain text from the encrypted text.

1.6 Types of Blockchains

Like many other types of databases, Blockchains can be classified as public, private, consortium or hybrid variants, depending on their application [120] [101]. Before we get into details of the various types of blockchain ,let us first learn what resemblances do they share. Each blockchain consists of a collect of nodes working on a peer-to-peer (P2P) network system. Each node in a network has a copy of the shared ledger which gets updated timely. Each node can check transactions, read ,write or receive transactions and create new blocks.

1.6.1 Public Blockchains

A public blockchain is permission-less distributed ledger system. Anyone can join the network can sign in on a blockchain platform to become a licensed node and be a part of the blockchain network because public blockchain is decentralized and does not have a only entity which monitors the network. a node or user which is a part of this blockchain can thus view actual and past registers , check transactions do proof-of-work for an incoming block .the data on blockchain are secure as can not edit it or change it when they are validated on the blockchain.the most common examples of a public blockchains are Bitcoin and Ethereum blockchains .

1.6.2 Private Blockchains

A private blockchain is permissioned blockchain working only in a locked network. Private blockchains are generally used within an enterprises where only selected members are participants of the blockchain network based on access controls.There are one or more entities which monitoring the network and this leads to reliance on third-parties to transact. In a private blockchain, only the entities participating in a transaction will have knowledge about it, while the others will not be able to access it. Hyperledger Fabric of Linux Foundation is a most popular example of a private blockchain.

1.6.3 Consortium Blockchains

This type of blockchain is known as a semi-decentra-lized blockchain where more than one organization controls the blockchain. Unlike what's happening in private blockchain which a single organization controls. Moreover, consortium blockchain is a set of pre-defined nodes on the network .Thus, consortium blockchains provide security that is inherited from public blockchains. Most oftentimes, consortium blockchains are associated with enterprise utilize, where a set of organizations collaborating to authority blockchain technology to ameliorate businesses.Examples of consortium blockchain are Energy Web Foundation, R3, etc.

1.6.4 Hybrid Blockchains

A Hybrid Blockchain was built to take feature of the best of Private and Public Blockchain. It is made up of public and private blockchain. When we say private, it means that it's a locked network. The users can enter only based on invitation, and but , the hash generated in the system is verified in the public system, thus taking the public blockchain feature. Hybrid blockchain puts the limit on who can enter the network at the same time provide transparency, security, and integrity of the transactions.In a hybrid blockchain, the transaction can be made verifiable when required. It can retain the transactions private, therefore ensuring immutability. The members of the hybrid blockchain can decide which of the transaction should keep private and which should be made public. Even if the transaction is private, it can not be altered or modified by members. There are a little hybrid blockchains that have been started. XinFin is a Hybrid blockchain that uses Ethereum and Quorum blockchain solutions. Dragon Chain is another example of a Hybrid Blockchain.

1.7 Work of Blockchain

Blockchain is a new software technology that first appeared in 2009, and was a major reason for the emergence of the first cryptocurrency like Bitcoin. It is a technology that works on an electronic record system and communicates using a peer-to-peer (p2p) network without the need third party, to process transactions and user data and register them in a safe manner and highly efficient. Types of Blockchain differ public or private , each with different techniques for work and use. But in general it has the same process. In a simplified example, we will explain how a public blockchain works with the individual steps shown in figure 1.9 following [121] :

In other term :

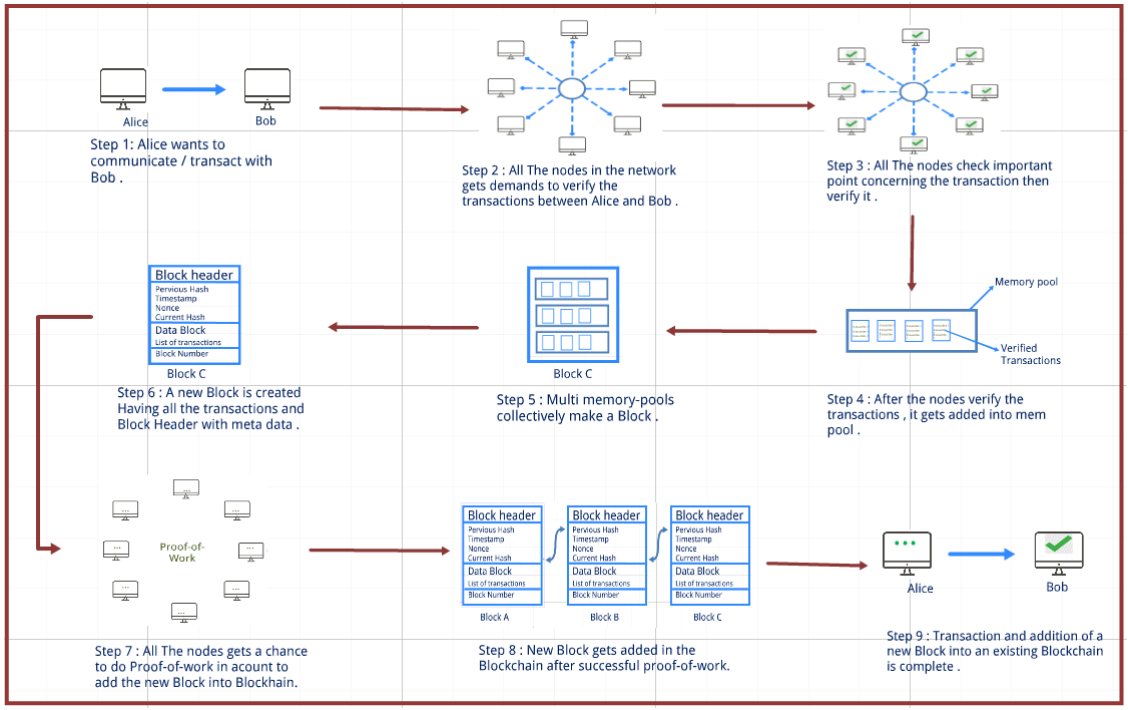


Figure 1.9: Blockchain Working

Step 1 : Suppose, A and B two nodes in public blockchain network , A wants to transmit or communicate with B.

Step 2 : This transaction can just execute if all the other entrant nodes in the network check it as a valid transaction. Therefore, each node will receive the demand to check the transaction to take place between Alice and Bob.

Step 3 : Every node will verify certain points concerning the transaction such as the validity of the two nodes, does A have enough funds to do this transaction, etc.

Step 4 : Once all the nodes check and verify all the mentioned points, the transaction is available to execute. Then that transaction gets added into a **Memory pool** or **Mempool** (a compound of two words, 'Memory' and 'Pool') Which is a so important part of the Bitcoin Network (the network of computers and devices which are connected to the internet and are working the Bitcoin Core software). As its name proposes, it is a pool of memorized, held data. The data that is being stocked on the Mempool are uncertain transactions that have been received by a node.

Step 5 : Many of these verified transactions are grouped into mempools and multiple mempools are grouped together to create a block. Each block has a specific memory limit for storing transactions.

Step 6 : Each new block will have a block header, that consists of timestamp, hash code of the previous block and its current hash, Each block has its unique hash code which acts like its fingerprint, Nonce a random number, it is used for the proof of work, and a block Data, that contains of transactions data list, the last one, is a block

number.

Step 7 : For add a new block into the present blockchain, nodes in the network require to do proof-of-work. As we know, every block has its unique hash function which is an identification code created using method SHA256. nodes in this network need to do proof-of-work ,which is decrypting this code and finding the correct answer to this hash puzzle, that's take 10 minutes to crack the code automatically by specialized computers.

Step 8 : The block is checked every time the node completes a proof-of-work and finds the correct answer to the hash puzzle for that block. More and more nodes must verify or finish the proof-of-work of the same block until it is finally added to the blockchain. Each block has a unique series of transaction records. To create a new block and add it to the blockchain, you must have a unique series of transactions.

Step 9 : However, Finally a new block is added and the deal is completed between the two points Alice and Bob.

This operation repeats itself and continues to add new blocks to the blockchain continuously.

1.8 Advantages and Disadvantages of Blockchain

1.8.1 Advantages of Public Blockchain

The most notable advantages that come from using a public blockchain are [7]:

- **Trustless**: The greatest advantages of public blockchain is that there is no need for trust each other for the network for transactions to be processed and secured. Everything is recorded, public, and cannot be changed. Since everyone is incentivize to do the right thing for the improvement of the network. There is no need for intermediaries.

- **Secure**: In a public blockchain network, anyone can participate by being a entire node or miner and contribute to the security of the system. It follows that the greater the decentralization and active participation, the more secure a blockchain will be. With more nodes in the network, it will be much difficult for any bad actors to attack the system .It is practically impossible to control the consensus network .

- **Open and Transparent**: Anyone can participate in a public blockchain network to view and validate transactions .Additionally ,The transparency of a public blockchain is a major feature that attracts a wide array of use-cases,such as decentralized identity.

1.8.2 Disadvantages of Public Blockchain

The disadvantages of using a public blockchain are [7]:

- **Slow**: The greater problems with public blockchain is speed. Public blockchains like Bitcoin are very slow, it can process 7 transaction per second (TPS) while Ethereum can do 15 transaction per second (TPS). Public blockchains are slow because it takes time for the network to reach a consensus. Additionally, the time needed to process a single block takes a long time compared to a private blockchain.

- **Scalability Concerns** : Public blockchains also face concerns scalability. Currently, public blockchains can't compete with traditional systems that can process big amounts of transactions. In fact, the more a public blockchain is utilized, the slower it gets because more transactions fill and block the network. However, steps are existence taken to treatment this problem . An example is Bitcoin's Lightning Network.

- **Energy Consumption**: Energy consumption has been a worry when it comes to public blockchain. Bitcoin's consensus algorithm relies on Proof-of-Work, which uses a significant amount of electrical resources to function . However, there are other consensus mechanisms and algorithms such as Proof-of-Stake which use far less electricity.

1.8.3 Advantages of Private Blockchain

The most notable advantages that come from using a private blockchain are [7]:

- **Faster** : Private blockchains can process much higher transactions amounts per second (TPS), as compared to public blockchains. This is due to the presence of a few participants with authorization results in lesser times in earning a network consensus. Consequently, this allows for the processing of more transactions for each block, that can process thousands or even hundreds of thousands of transactions per second (TPS), compared to Bitcoin's 7 TPS.

- **Scalable** : Only a few nodes are responsible for data management. So, the network can support and process comparatively higher transactions. Unlike a decentralized system where achieving consensus could take time a private network's decision-making process is more centralized. Thus, it is much faster.

1.8.4 Disadvantages of Private Blockchain

The disadvantages of using a private blockchain are [7]:

- **Trust is Needed** : The sincerity of the private network depends heavily on the credibility of authorized nodes. They are responsible for checking and validating authentic transactions. In addition, the validity of records cannot be independently verified. External actors have to trust a private blockchain network without having any control over the verification. Unlike public blockchain that does not demand you to trust anyone.
- **Security** : With the presence of fewer nodes, it's a lot easier for an untrustworthy individual to gain control of the network and menace the complete network. A private network is more vulnerable to hacking and the manipulation of data.
- **Centralization** : The private network should be built and maintained by the project or business or a league of industry players which include preserving an intricate Identity and Access Management (IAM) system for the users. This often leads to centralization which is what blockchain actively tries to avoid.

1.9 Challenges and Limitations of The Blockchain

Every technology has it's limitations and challenges that need to be treated in order to make a system more solid, accessible and helpful. Blockchain has come a long way since it's beginning in Bitcoin. that it has problems, issues and many more. So a lot of effort is being made to beat the challenges posed by blockchain technology, among of this challenges [68] are presented as follows :

1.9.1 Scalability

: This problem has been the focus of media attention and exact research in recent years This is the only most important problem that could mean the difference between wider adaptability of blockchains or limited private utilize only by unions. The distributed ledger technology faces difficulty in adequately supporting a substantial number of clients on the system. As a result of great research in this field, many solutions have been proposed, about protocol-level improvements. For example, a usually mentioned solution to bitcoin scalability is to increase its block size. Other suggestions contain off-chain solutions that offload certain processing to off-chain networks, for example, off-chain state networks.

1.9.2 Regulation

The major challenge in implementing the blockchain is principals and regulations of different cities. The essence question is that blockchains and mostly cryptocurrencies are not known as a legitimate coin by any government. Although, in almost cases, it has been classified as money in the US and Germany, it is still far from being accepted as a normal currency. Further, blockchains in their actual state are not known as a platform that can be utilized by financial institutions. Interested authorities want to pay care on research on this issue and define new strategy and devising novel policies.

1.9.3 Privacy

Privacy of transactions is a much required feature of blockchains. However, because of its large nature, mostly in public blockchains, everything is transparent, therefore preventing its usage in different industries where privacy is of essential value, such as finance, health, and many others. There are different proposals made to address the privacy issue and some advance has already been made. Several techniques, such as blackout on non-discrimination, usage of Symmetric encryption, zero knowledge proofs, and ring signatures.

1.9.4 Relatively immature technology

Blockchain technology remains immature, including the lack of extensive testing, absence of a regulatory framework and the lack of clarity regarding how technology would interact with existing systems [87].

1.10 The Blockchain Technology Cases

The Blockchain technology may be used in the different ranges such as industrial ,technical. Where the biggest companies in different cities are applying the Blockchain technology to improve system's quality and work capacity. Some examples of using blockchain technology :

- **The government management** : There are different solutions into the government management. The first decision is Borderless. It is the governance platform which assures the coalition of the legal and economic services [78]. The second solution is the ID2020. This organization is provided proof of the identity for people without documents [123].
- **The electronic voting** : The Follow My Vote is the secure and transparent platform for anonymous online voting[45]. The E-Residency is the electronic

identification system for the citizens of Estonia and for the business-people there [44]

- **The medicine** : The MedRec is the project which provides secure and transparent access to the medical records of each patient in the medical institution [62].
- **The supply chains** : The Blockverify is the solution for the transparency in the supply chain. This platform has four main using cases: the pharmacy, the diamonds, the luxury items and the electronics [63]. Another example of the solution in the supply chain is the Bext360 which using the Blockchain technology for the coffee trade tracking [122]. The Maersk and IBM corporations are launching the joint venture for the more efficient and secure supply chain with using the Blockchain technology. This platform shows to each participant of the supply chain the products' location and specifications of the transportation [123].

1.11 Blockchain Frameworks

The most popular use of blockchain technology is the cryptocurrency. In fact, blockchain was already discovered by Satoshi Nakamoto to permit for the creation of the world's first cryptocurrency Bitcoin.

Satoshi came up with the blockchain principle to fix the problem that stood in the way of all digital currencies, which is the problem of "Double Spending" [95]. Blockchain technology has been separated from bitcoins for injection into many other problems. It has enabled the creation of decentralized currencies, smart contracts that can be controlled online.

1.11.1 Bitcoin

1.11.1.1 Definition

In 2008, Bitcoin was introduced through a paper called, Bitcoin: A Peer-to-Peer Electronic Cash System [95].

Bitcoin is a decentralized network and digital currency that uses a peer-to-peer system to verify and process transactions via the Internet without the need for external parties to monitor them, such as banks and card processors. The Bitcoin technology uses cryptographic proof in its computer software to process transactions, verify their legitimacy, and publish processing works on the network [100].

1.11.1.2 Bitcoin Protocol

- **Hash**

  Bitcoin generally uses the SHA-256 hashing algorithm, it is computed twice. We will explain it later with details in next chapter. Another hash that contains a shorter message digest called RIPEMD160 is also used to create Bitcoin addresses [98].

  Example of double-SHA-256 encoding of string "hello" :

  First round of Sha-256 :

  2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824.

  Second round of Sha-256 :

  9595c9df90075148eb06860365df33584b75bff782a510c6cd4883a419833d50.

  For Bitcoin addresses (RIPEMD-160), this would give:

  b6a9c8c230722b7c748331a8b450f05566dc7d0f.

- **Merkle Trees and Merkle Roots**

  In bitcoin and other cryptocurrencies, Merkle trees serve to encode blockchain data more efficiently and safely. Also referred to as "Binary Hash Trees".

  Merkle trees are an essential part of this technology. For a block, the Merkle root comes from hash and associate two hash parameters and created a top-level tree node. By doing this, it will get one hash to store for the hashes all of the underlying transactions. This one hash is called Merkle Root, where every block has Merkle Root stored in the block header [72].

  Merkle tree enables each node on the network to check individual transaction and check the entire block. If the block version in the blockchain has the same Merkle root to another, then the coefficients in that block are the same. A small change would lead to Merkle roots largely different due to hash properties [72]. The figure 1.10 shows an architecture of Merkle Tree in blockchain.

- **Signatures**

  The Bitcoin system uses the Elliptic Curve Digital Signature (ECDSA) cryptographic algorithm as a basis for safety and confidence. The reason Bitcoin uses elliptical curves instead of prime numbers is because calculations that use elliptical curves use less CPU and memory, making them more efficient [64]. Public and

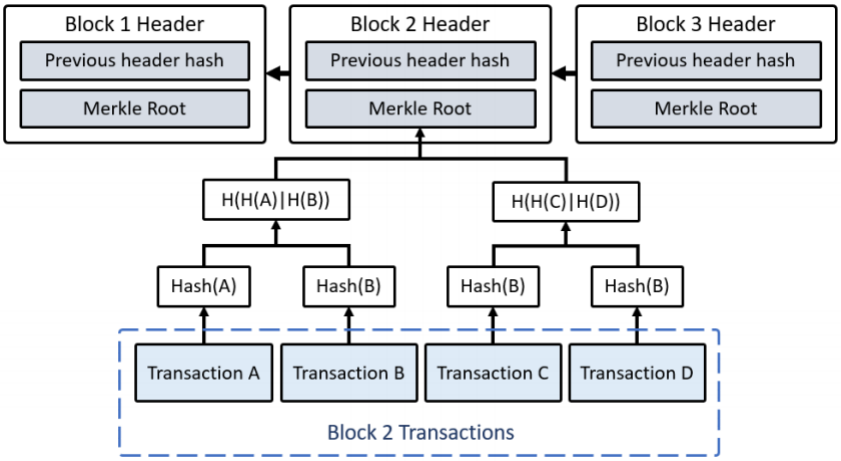


Figure 1.10: The architecture of Merkle tree in the blockchain [72].

private ECDSA keys are generated for each Bitcoin address and linked internally by Bitcoin Client Software. We will see in next chapter of cryptography how this algorithm work.

- **Bitcoin Addresses**

  Bitcoin address is a unique identifier that acts as the virtual place where the cryptocurrency can be sent. It is similar to the email address. Whereas, Bitcoin can be sent to someone by sending it to one of their Bitcoin addresses. This address consists of 27-34 alphanumeric characters begin with digit 1 or 3. There are currently two different types of Bitcoin addresses in existence, Pay-to-PubkeyHash, which is public key hashing that results in the structure named pubkeyHash and Pay-to-ScriptHash, which is a hash of the script serialized Script which results in the structure called scriptHash. They are used in conjunction with their corresponding transaction type. They are computed in the same way, but 0×00 is added to pubkeyHash in the first type and 0×05 is added to scriptHash in the second type [103] [66], figure 1.11 shows how to calculate them :

  1. **Private key and Public key Generation**

     The Elliptic Curve digital signature algorithm is applied to the private key, to obtain a 64-byte integer which is Public key. This consists prefix 0x04 and two 32-byte integers representing the point X and Y on the elliptic curve, concatenated together. The elliptical curve is a curve defined by the equation $y^2 = x^3 + ax + b$ with the chosen a and b. Bitcoin uses the secp256k1 curve. We will see with details in next chapter.

  2. **Compressed Public Key**

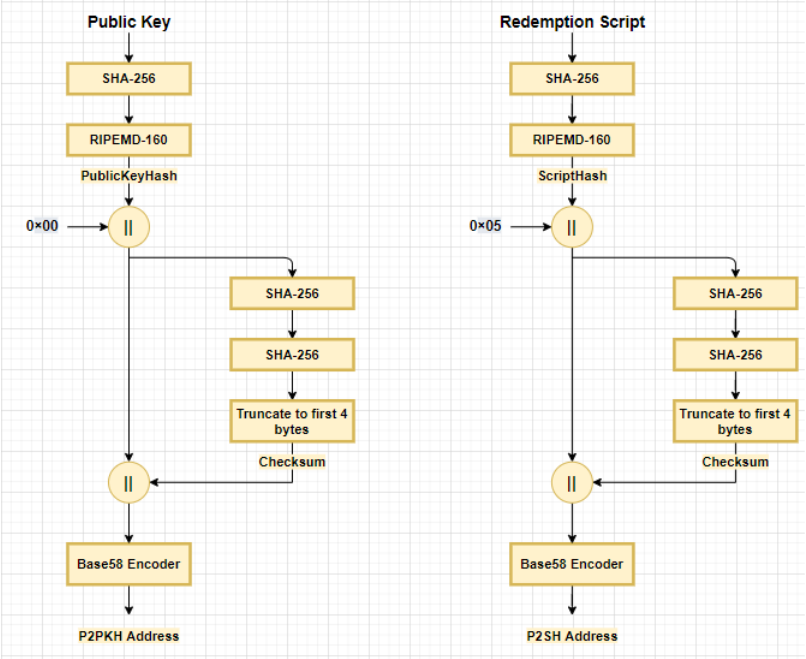


Figure 1.11: Calculate of Bitcoin Address [103].

A compressed public key is executed without using the uncompressed key because it is smaller size than it, saving blockchain space. To convert from an uncompressed public key to a compressed public key, you can omit the x value because the y value can be solved for using the equation of the elliptic curve: $y^2 = x^3 + 7$. Therefore, to distinguish between the two possible values of y, we store a compressed public key with the prefix 02 if the y is even and positive, the last binary number of the y co-ordinate is 0. and 03 if it is odd and negative and the last binary is 1.

3. **Encrypting the Public Key**

   The encrypted public key is calculated by applying the SHA256 hash to the public key, then applying the RIPEMD160 hash to the result, which results in a 160-bit (20 byte) number.

   > Encrypted public key = RIPEMD-160 (SHA-256(public key))

4. **Calculate and Append Checksum**

   A checksum is appended to the end of the string. This is calculated using the first four bytes of a double SHA256 hash process that takes the encryption public key of whatever is being validated by the checksum and this last used to ensure the address was transmitted correctly without any data corruption such as a address typing error.

C = SHA-256 (SHA-256 (Encrypted public key))
Checksum = first 4 bytes of C

5. **Base58 Encode**

   After calculating the checksum, the base58check markup is performed and is used to convert the data from binary to text using the table shows in figure 1.12. Base58check allows to display hash more accurately (using more alphabet) while avoiding characters that can be confused with each other like 0 and O where any typo might lead to losing money.

Hex Address = encrypted public key + Checksum
Address = Base 58 (Hex Address)

Value	Character	Value	Character	Value	Character	Value	Character
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

Figure 1.12: Calculate of Bitcoin Address [10].

1.11.2 Ethereum

1.11.2.1 Definition

Inventor Vitalik Buterin has done what Tim Berners-Lee had done did to the networks. The World Wide Web (WWW) put individual networks under a one umbrella. Likewise, Ethereum has merged all blockchain functionality into one network and avoided creating individual chains for every purpose [16].

Ethereum is an open source Blockchain platform which allows anyone to develop and deploy Blockchain decentralized applications, whatever the type of applications including cryptocurrency, symbols, social apps and more. In other terms, Ethereum opened the possibilities of the "Blockchain" and "distributed ledger" technology to other application areas [16].

Also, Ethereum describes as a "global computer," meaning that it can be considered a single network platform that anyone in the world can use as it can encrypt and execute many programs, and has the right to access of these programs by implementing programmable smart contracts on the blockchain [125].

Ethereum is an open source decentralized platform which is based on Blockchain Technology. There are two type of users in a Ethereum blockchain [19]. The one who issues a decentralized application (DApp) or a smart contract and others who take part in the contracts.

Each user will have an account in Ethereum, called an Externally Owned Accounts (EOA). And Other, each decentralized application DApp will have an account address in Ethereum known as Contract Accounts, where user transaction is related to these single accounts and also each user can execute transactions with both types.

DApp is the 'Decentralized Applications' running on the blockchain. They are the applications that run on blockchain without any centralized control. We can say bitcoin is a decentralized application that runs on Bitcoin blockchain. But it is the Ethereum blockchain that extended the scope of decentralized application and popularized the word DApp [16]. In simple terms, DApps can be considered as applications, tools or programs that work on the decentralized Ethereum Blockchain. It have 4 important features [18]:

- **Open source** : It means that all the codes should be accessible to all and available for scanning.
- **Decentralization** : It makes all the operations stored overtly and on a decentralized blockchain.
- **Incentivization** : It motivates validators of the blockchain.
- **Protocol** : It indicates that the application community must agree to a cryptographic algorithm to show proof of values.

1.11.2.2 Ethereum's Components

Ethereum is a single blockchain with a built-in programming language. Ethereum consists of several different components [16] that make It serves as a platform where programmers can create, use and run many different types of decentralized applications.

1. **Smart Contract**

   Software designed to run on the Ethereum platform are usually referred to as smart contracts. A smart contract is a computerized transaction protocol which

is executed automatically ,that enforces the terms of a contract. It can be considered as a computer program that organizes and automates the relationship between two or more untrusted parties without a middleman in a more secure and transparent way.

2. **Ether**

   Ether is the cryptocurrency for the Ethereum network. Ethereum website put it this way, "Ether is a form of payment made by the clients of the platform to the machines executing the requested operations." Ether is the reward for individual nodes as a result of the computational and other resources spent by nodes. As people become more interested in Ethereum, the value of ether also increases daily. Today, ether is the most requested coin after Bitcoin.

3. **Ethereum Clients**

   Ethereum Clients are the tools used to connect to the Ethereum blockchain for development or mining objectives. It has three different formally maintained implementations written in C++ [14], Go [9] and Python [15], all of which are optimally operable across the Ethereum network. More specifically:

   - Geth : Geth is an Ethereum client working in GO language. Geth has a command line interface (CLI) tool that communicates with the Ethereum Network and acts as the link between the different nodes in the network.
   - Eth : C++ Eth is a powerful Ethereum client which is more focused on miners.
   - Pyethapp : this client is useful for decentralized applications development using python. 'Pyethapp' is also an excellent choice for research and academic purpose in Ethereum blockchain.

4. **Ethereum Virtual Machine**

   Ethereum Virtual Machine (EVM) is the engine behind the entire Ethereum series. Smart contracts run on Ethereum Virtual Machine (EVM). The consensus-driven decentralized computer that distinguishes Ethereum from previous Blockchains. This Virtual Machine executes its own language of bytecode. For this reason, Contracts are typically written in higher level languages that have been developed, like Solidity, then compiled to EVM bytecode.

5. **Solidity**

   Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state [51].

Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM) [51].

Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features [51]. Below we will present some of its main features [125].

(a) **Types**

Solidity is a statically typed language, which means that the type of each variable (state and local) needs to be specified. Solidity provides several elementary types like any traditional language it supports booleans, integers and strings.

The most important data type is that of an Ethereum address. It holds the 20 byte representation of an Ethereum account address, where has internal pre-defined members to check account balances or transfer Ether via a contract, as well as to invoke functions from other contracts [20].

Solidity also supports structures and counters as well as byte arrays that can hold data of any type [20].

(b) **Event**

Events are a way Solidity provides for smart Contracts to access a transactions registers, or to write something has occurred and store the arguments passed in transactions registers, It is stored on Blockchain and can be accessed using the contract address until the contract is present on the blockchain.

Event messages are not accessible from within contracts, not even the contracts that have created them [17].

(c) **Function types**

Function types are the types of functions. Variables of function type can be set from functions, and function parameters of function type can be used to pass functions to and return functions from function requisitions. Function types come in two flavours internal and external functions.

The visibility function types are defined using one of the four visibility keywords : private, internal, external, or public, and are placed directly following the function parameter list [17].

i. **Private** : A private function is one that can only be called by the main contract itself.

ii. **Internal** : An internal function can be called by the main contract itself, plus any inherited contracts.

iii. **External** : An external function can only be called by other contracts. It cannot be called from the main contract itself or any contracts inherited from it.

iv. **Public** : A public function can be called from all potential parties. All functions are made public by default.

(d) **Function Modifiers**

Function Modifiers are structures used to easily change the behaviour of a function. They are mainly used to check a condition before to executing the function. They are inheritable properties of functions and each function can belong to various modifiers [17].

6. **Etherscripter**

Etherscripter is a visual constructor of smart-contracts for Ethereum. It provides a GUI for creating smart contracts in simple steps. Grammar shapes are represented by graphical shapes that fit together in the drag-and-drop interface where the corresponding back-end codes in Serpent, LLL, and XML will be generated automatically. Even non-programmable Etherscripter can be used to create smart contracts. A Figure 1.13 shown an example of Etherscripter Smart Contract

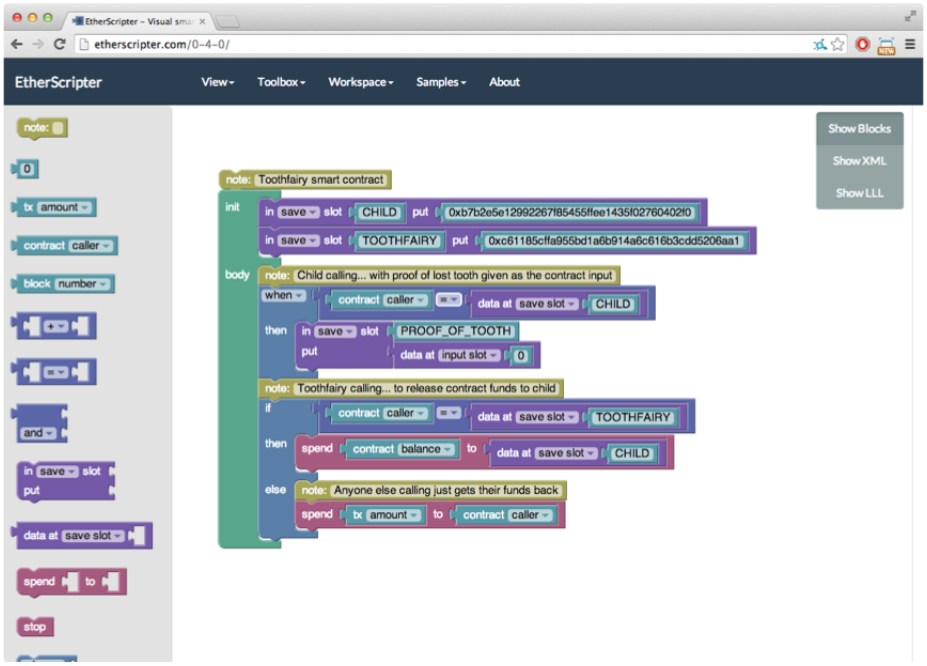


Figure 1.13: Etherscripter Smart Contract.

1.11.2.3 Advantages of Ethereum

Ethereum platform benefits from all the properties of the Blockchain technology that it runs on [21]. Below we will discuss some major benefits of Ethereum :

- **Unchanging nature** : It is immune to any third party interference, which means that no one can ever control all decentralized applications and decentralized autonomous organizations that are deployed within the network [22].
- **Debasement and sealed** : Any Blockchain network is built around the basis of consensus, meaning that all nodes within the system need to agree on all changes. This makes the network non-manipulative and removes the potentials for cheat and corruption.
- **Zero downtime** : The entire platform is decentralized. Hence, all applications will always remain online and will never shut down.
- **Secure** : It is completely immune to any third party interventions or hacking assaults and deceitful exercises due to the decentralized nature and secured utilizing cryptography.

1.11.2.4 Disadvantages of Ethereum

Regardless of achieving different advantages, and the fact that smart contracts are meant to make the network fault-proof, there are people writing the code for them, also there is always room for human error, and any error in the code can be exploited [21].

In the event of a code misuse, there is no fruitful way to stop a hacker attack, the only possible way of doing so would be to reach a consensus and rewrite the basic code. However, this goes completely contradicts the core of the Blockchain, as it is supposed to be an unchangeable and immutable ledger.

1.11.3 Hyper-Ledger Fabric

1.11.3.1 Definition

The Hyperledger Project [24] is a cooperative effort to create an enterprise-grade, open-source distributed ledger framework and code base private and permissioned business networks, where the member identities and roles are known to the other members. It aims to advance blockchain technology by identifying and realizing a cross-industry open standard platform for distributed ledgers, which can change the way business transactions are conducted globally. Determined as a project of the Linux Foundation in early 2016, the Hyperledger Project currently has more than 50 members [71].

Hyperledger Fabric is an implementation of a distributed ledger platform for executing smart contracts, leveraging usual and proven technologies, with a modular architecture allowing pluggable implementations of different functions [71] It enhances container technology and delivers enterprise-ready network safety, scalability, and confidentiality [91].

In other definition, Hyperledger Fabric is a flexible operating system for permissioned blockchains designed for business applications beyond the basic digital coin addressed by Bitcoin and other existing networks. A key property of this system is its extensibility, and in particular the support for multiple ordering services for building the blockchain [118].

1.11.3.2 Hyper-Ledger Fabric Components

Hyperledger Fabric differs from blockchains like Ethereum or Bitcoin, not only in its type or because it doesn't know coins, but also in terms of its internal structure. In a typical Hyperledger network we have the following main components [25] are shown in the following figure 1.14 :

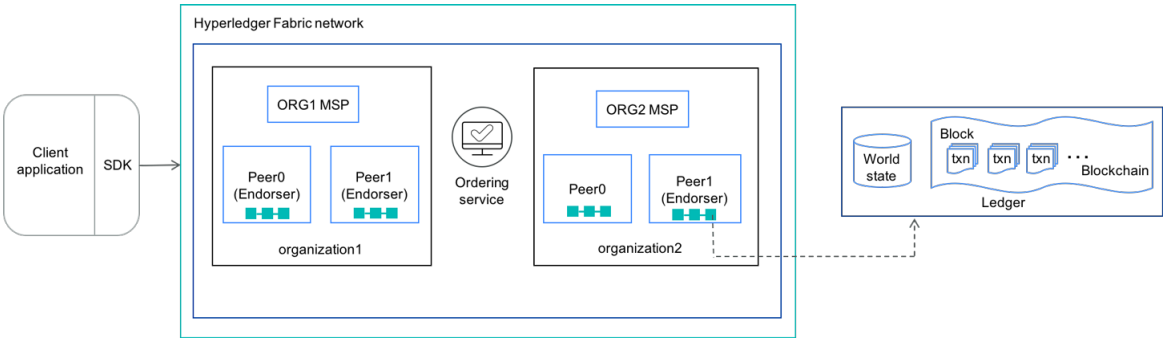


Figure 1.14: The components of a Hyperledger Fabric network [91].

- **Ledger** : The ledger is made up of two components, the world state and Blockchain. Each participant has a copy of the ledger for each Hyperledger textile network they belong to. Figure 1.15 shows the two components of ledger.

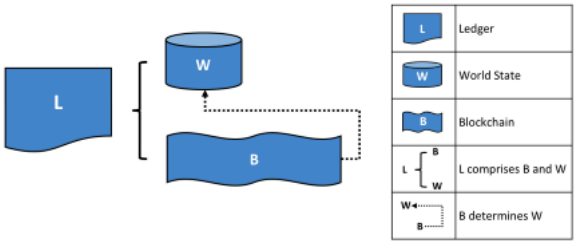


Figure 1.15: Ledger consists of W-World State and B-Blockchain [26].

  - The World State describes the state of the ledger for a specific time point. It is the ledger database.
  - Blockchain is the transaction record that records all transactions that led to the current value of the world state, it is the history of the modernization of the world state.

- **Nodes** : is only a logical function in meaning that multiple nodes of various types can run on the same physical server. What matters is how nodes are collected in "trust fields" and linked with logical entities that manages them. There are three types of nodes :

  1. **Client** : are applications that act on behalf of a user to sends transactions to the network, clients communicate with both peers and the ordering service.
  2. **Peer** : This is an entity that performs transactions and maintains the state and a copy of the ledger. Besides, some peers can be endorsing peers, or endorsers. Every chaincode may specify an assent policy, which determines the needful and enough conditions for a valid transaction assent [27].
  3. **Orderer** : This makes a shared communication channel between clients and peers, and it packages blockchain transactions into blocks and sends them to committing peers.

- **Membership Services Provider (MSP)** : The MSP is a component that provides identity validation and authentication processes by issuing and validating certificates. The default interface used for the MSP is the Fabric-CA API. It identifies which certification authorities (CAs) are trusted to define the members of a trust field, and sets the specific functions an actor may play (member, admin, and so on). There are two types of MSPs.

  - Local MSP : It sets users(Clients) and nodes(peers, orderers). It defines who has managerial or contributory rights at that level.
  - Channel MSP : It defines managerial and contributory rights at the channel level.

- **Chaincode** : Chaincode is a like concept to a smart contract in other networks such as Ethereum. It is software that defines assets and related transactions. Chaincode is a program written in a higher level language like Golang or Node.js, executing against the ledger's current state database.

- **Channels** : A channel is a private communication subnet for sharing confidential information between multiple network members. Each transaction is performed

on the channel which is only visible to the authenticated and authorized parties. The peer can maintain various ledgers, and can be linked to various channels.

- **Endorsers** : These validate transactions and invoke chaincode, sending back the endorsed transaction results to the calling applications [65].

1.11.3.3 HyperLedger Fabric Workflow

To understand how Hyperledger Fabric is various and how it works, let's look at how to validate transaction. In a typical Hyperledger network, the following figure 1.16 depicts the system flow for treatment a transaction:

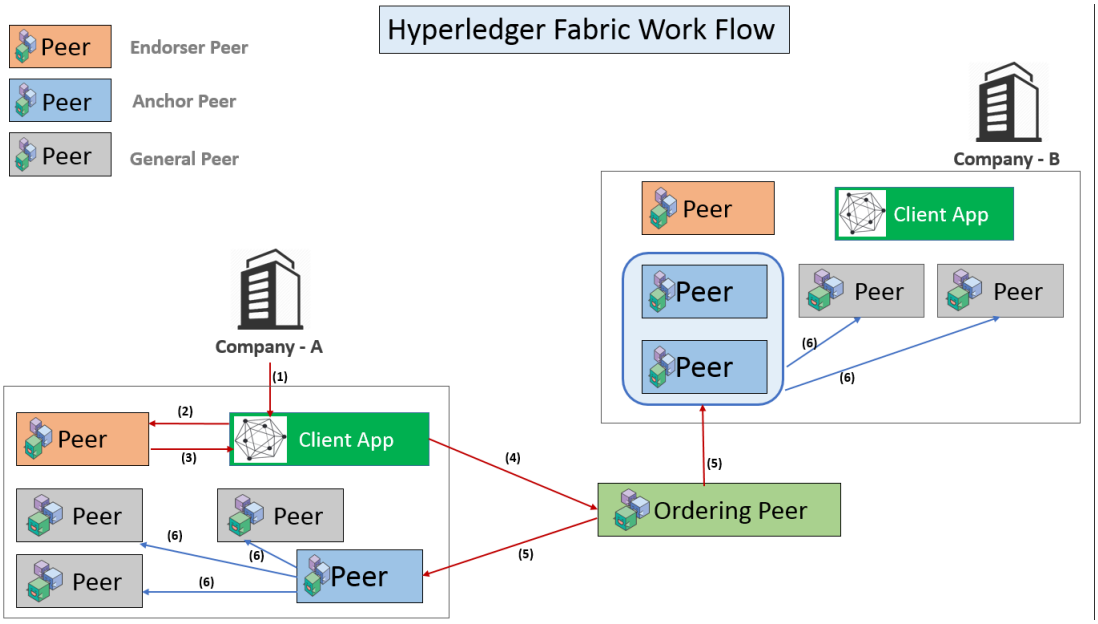


Figure 1.16: HyperLedger Fabric Transaction Flow [23].

1. A participant in the member Organization invokes a transaction demand during the client application.

2. Client application transmit the transaction invocation demand to the Endorser peer.

3. Endorser peer tests the Certificate details and others to validate the transaction. Then it performs the Chaincode and returns the Endorsement responses to the Client. Endorser peer forwards transaction assent or rejection as part of the endorsement response.

4. Client now forwards the approved transaction to the Orderer peer for this to be duly ordered and be included in a block.

5. Orderer node contains the transaction into a block and send the block to the Anchor nodes of various member Organizations of the Hyperledger Fabric network.

6. Anchor nodes then transmit the block to the other peers within their organization. These individual peers then update their local ledger with the recent block. Thus the entire network gets to synchronize the ledger.

1.12 Conclusion

Blockchain is a technological invention used for swap data on a distributed network in a secure and encrypted way and ensure that transactions can never be altered.

The Blockchain behind Bitcoin cryptocurrency since 2008 has been gaining drag as companies began to merge blockchain-based technology into their current business models [84]. Its ability to convert traditional industry has demonstrated in different sectors with its major advantages: decentralization, autonomy, security and transparency, such as business, healthcare supply chain and data management. In the next chapter, we will look at supply chain management, in particular, the pharmaceutical management, and we will provide the most challenges they face during the chain.

Chapter 2

Drug Supply Chain

2.1 Introduction

The pharmaceutical industry is a complex institution fraught with conflicting goals and many difficult restrictions. Pharmaceutical supply chains are becoming increasingly complex as the path from production to consumption is not as linear as one might think [85].

Since pharmaceutical products are vital products, their availability and accessibility are significant issues for companies and institutions. It is necessary to deliver medicines at the right time to the right person in standard conditions [85]. The incorrect distribution of medicines not only impacts corporate reputation, customers' satisfaction and corporate gains, but can also distribute healing operations to patients and have passive traces on public health.

In this chapter, we will explain the supply chain management in general and discuss the supply chain in the pharmaceutical sector. Also, we will see how distribution processes of medicines in Algeria. In addition to the challenges and difficulties it faces in its path.

2.2 Supply Chain Definition

In commerce and businesses, The Supply Chain represents a series of business processes and applied technologies that contribute to the flow of materials, information, and money from suppliers to manufacturers and distribution channels that serve consumers [1]. The supply chain includes not only the manufacturer and its suppliers, but also transport companies, warehouses, retailers and consumers themselves (depending on logistical flows). It includes also new product development, marketing, distribution operations, financing, and customer service [126].

Also known, Supply Chains are a network between a company and its suppliers to

produce and distribute a specific product to the final buyer. This network includes various activities, information and resources, people, etc. It also represents the steps taken to obtain a product or service from its original condition to the customer [31].

The supply chain starts from the extraction of raw materials. Through logistic services, these materials are taken to resources, and then are transferred to manufacturing companies, which work to process and exploit them in products.

After manufacture, distributors wholesale the finished products, and then are received it by retailers. After that, it is sold to consumers in stores, once purchased by consumers, it completes the cycle. As consumers demand more, more raw materials are paid for production, and the cycle continues [34]. Figure 2.1 depicts a generic supply chain.

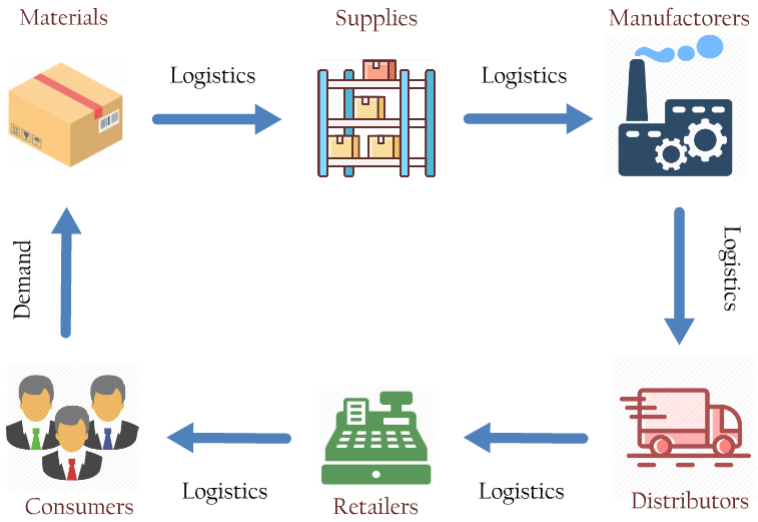


Figure 2.1: Generic Supply Chain [34].

2.2.1 Supply Chain Management

Supply Chain Management identifies all the resources, means, methods, tools and techniques intended to manage the supply chain as efficiently as possible, from the first supplier to the end customer. The most common and accepting definitions of supply chain management are :

- Simchi-Levi, Kaminski and Simchi-Levi gave the following definition:" ***Supply Chain Management is a set of approaches utilized to efficiently integrated suppliers, manufactures, warehouses, and stores, so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, in order to minimize system wide costs while satisfying service level requirements*** "[90].

- Also, Supply chain management (SCM) is defined as the management of a network of interconnected companies interested in the provision of product and service packages demanded by the end customers in a supply chain [81].
- Supply chain management is the systematic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long-term performance of the individual companies and the supply chain as a whole [92].
- Supply chain management is the integration of key business processes across the supply chain for the purpose of creating value for customers and stakeholders [88].

2.3 Supply Chain Management Processes

Effective supply chain management demands many processes relating to the flow of informations, products, and funds. These processes divided into three categories or phases, depending on many experts [73].

2.3.1 Supply Chain Strategy or Design

This stage includes designing the supply chain by the company, which defines the structure of the supply chain and the activities that will be implemented by each stage of the supply chain. It contains strategies that include choosing the production and storage site and the facility's capabilities, and making decisions about the products to be made, as well as choosing the means of transportation and the source from which the information will be collected. Supply chain design decisions are long-term projects that are costly to reverse; therefore, market uncertainty must be taken into account [73].

2.3.2 Supply Chain Planning

Supply Chain Planning (SCP) concentrates on setting policies and steps for promotional activities, inventory and production regeneration policies [35]. Essentially, it defines the parameters of the supply chain.

This stage provides strategic planning that looks to the future with a future outlook. Supply Chain Planning deals with supply, distribution, manufacturing, planning, production scheduling, demand planning, forecasting and cooperation in the supply chain, and supply chain network design [73].

Supply Chain Planning (SCP) coordinates application to improve the delivery of goods and information services, from supplier to consumer, and to achieve a balance between supply and demand commitments in real time. Typical supply chain planning program units include network design, network planning, capacity planning, demand planning, manufacturing planning, and scheduling planning, distribution, and deployment planning [36].

2.3.3 Supply Chain Execution

Supply Chain Execution (SCE) applications process the information produced by supply chain planning tools to guide inventory and production renewal policies. On the other hand, it includes activities to efficiently purchase and balance the supply of goods and materials [36].

Supply Chain Execution (SCE) concentrates on execution-oriented applications, including order management, inventory management, warehouse management, transportation management and logistical management that includes all parties [36]. The goal of supply chain Execution operations is to deal with incoming customer requests in the best possible way.

1. **Warehouse Management**

   Warehouse Management are implementations that manage the processes of a warehouse or distribution center. Which includes receiving, storage, inventory management, shipment of goods and raw materials, , order distribution, repacking, packing, work management. The use of radio frequency technology together with bar coding, RFID or other data collection technologies can help improve the effectiveness of warehouse management systems, providing accurate real-time information [36].

2. **Transportation Management**

   Transportation management are used to manage all freight activities across the organization. Also it includes planning, executing and optimizing the physical movement of goods [36]. The primary function of the Transportation management systems system includes helping the user find the best position and price for any type of shipment to assure they gets the best deals.

3. **Manufacturing Management**

   Manufacturing process includes production planning / detailed scheduling, which is process supports the process of assigning production orders to supplies in specific sequence and time frame. and manufacturing execution is process supports

the process of capturing actual production information from the shop floor to support production control and costing processes [2]. Also It can include document control, work management, quality management, process and maintenance management [36].

4. **Procurement Management**

   Procurement team is responsible for getting the best source or sources of supply. Procurement management includes three processes, which is : Firstly, Purchase Order Processing, it perform the direct procurement requirements through the sourcing, issuance, and confirmation of purchase orders. Secondly, Receipt confirmation processing informs other departments about the received and confirmed quantity of ordered goods. Thirdly, Invoice Verification process receives, enters and checks vendor's invoice for correctness [2].

5. **Order Management**

   Order management is the process of tracking sales orders and executing them effectively. It includes data collection, order processing including credit card verification. It also includes keeping a record of the customer, which can include the purchase record, payment method and order size. Sales departments notify the warehouse to fulfill the order, then the order is shipped to the customer [32].

6. **Logistics Management**

   Logistics Management is a component of supply chain management that is used to meet customer requests through effective planning, control, and execution of traffic and the storage of information, goods and services from the construction point to the final destination. Logistics management helps companies minimize costs and improve customer services [33]

2.4 Drug Supply Chain

An effective distribution system concentrates on factors that includes product safety and quality throughout the drug distribution channel. World Health Organization concentrates on supporting countries for maintaining a stable supply of medicines, keeping medicines in good condition throughout the process, reducing medicines losses due to damage and expiry, maintaining accurate stock records, rationalizing drug storage points, using efficient transportation resources available, and reducing theft and fraud [37].

Since last years, big pharmacies has been relying on a complex, costly and ineffective strategies from hundreds of suppliers to manufacture, package, transport and deliver

various products to the market. So there is a need for a new and graceful approach, aimed at solving all-in-one, to facilitate the new pharmaceutical scene and increase productivity and profit [113].

Combining the supply chain with the pharmacy system gets a wide range of benefits including, reducing risks and overheads, increasing innovation and ensuring supply and compliance, quality control and others. Cost and efficiency gains will simultaneously help the industry fulfill its social responsibilities, including the need to pilot more sustainable manufacturing processes and produce more effective and safer drugs that the entire world can afford [113].

The pharmaceutical supply chain is often a hidden component of healthcare systems. There is a detailed path between the drugs that leave the manufacturer until it is distributed to the patient. Middle men play a vital role in securing a continuous supply of high-quality drugs to the medical front-line [61].

Pharmaceutical supply chains are one of the most complex logistics operations. They depend not only on temperature control and meeting stringent border regulations across regions, but also on logistics service providers to ensure the safety and security of products throughout the process. Having a single loose tie in the chain can mean that expensive and bio-active drugs and products are left in an unusable state or even pose a potential risk, with separate problems linked to the lack of adequate biomedical stocks or even legal procedures [61].

2.5 Drug Supply Chain System

The primary objective of the distribution management is to maintain a stable supply of pharmaceuticals and supplies to the facilities in the most effective and efficient manner. The distribution cycle begins when pharmaceuticals are sent by manufacturers or resources. It ends when the drug consumption information is reported to the procurement unit [30]. Figure 2.2 illustrates the typical pharmaceutical distribution system and its interactions with both public and private sectors in different levels.

The distribution cycle in the pharmaceutical field has nine main activities that are summarized in figure 2.3. And we will explain each activity in details.

2.5.1 Pharmaceutical Procurement

Procurement process is part of the management cycle of drugs. This goes hand in hand with selection of drugs, quantification of drug needs, storage and distribution/supply [97]. Procurement is the acquisition of goods and / or services at the best possible cost of ownership, of appropriate quality and quantity, at the right time, in the right

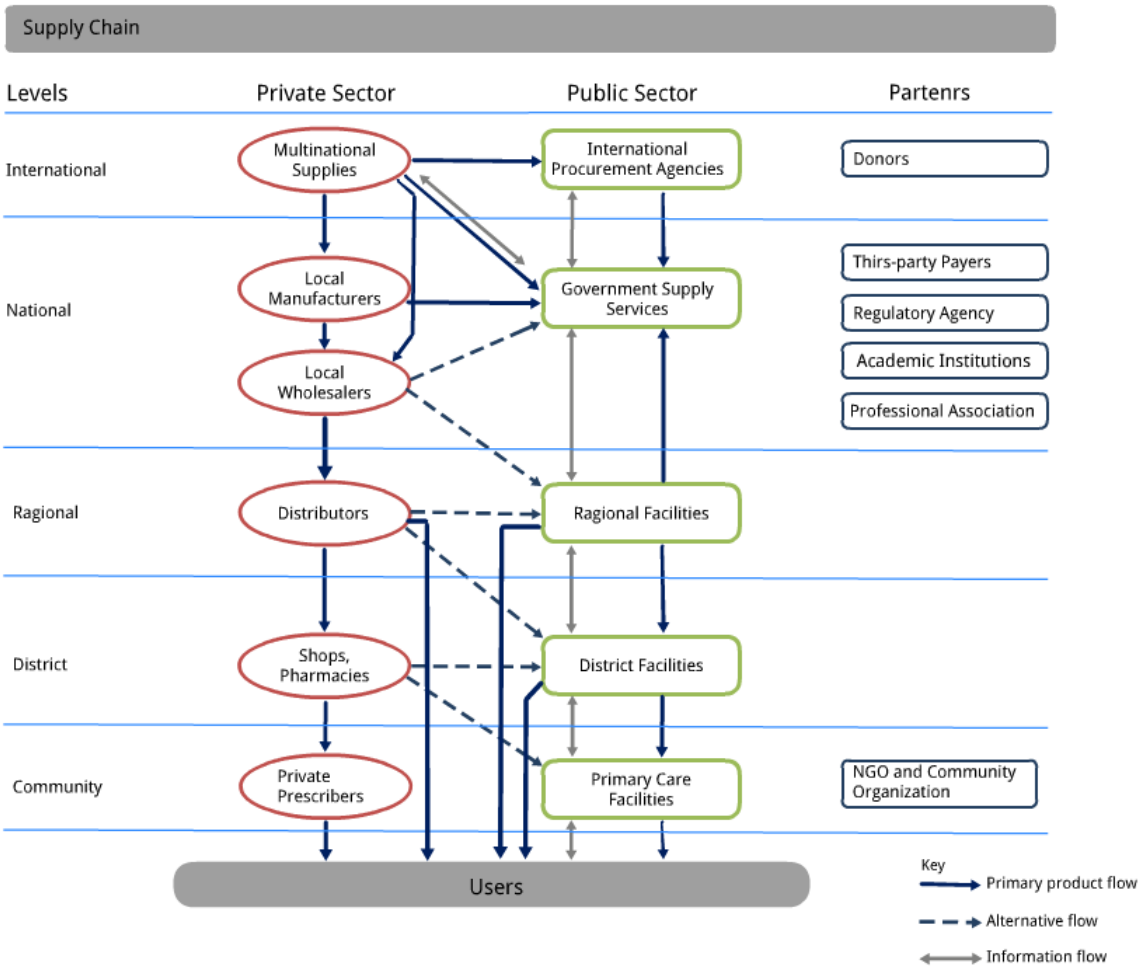


Figure 2.2: Pharmaceutical Distribution System.

place, and from the correct source of direct benefits, or the use of company personnel, or even governments [75].

Pharmaceutical procurement system is one of the major determinants of drug availability and total drug costs. Drug procurement is a complex process that involves many steps, decisions, actions, agencies, ministries, and manufacturers that determine the specific medicine quantities obtained, prices paid, and quality of medicines received [105].

2.5.2 Port Clearing

The port-clearing process is vital to the effective process of a public pharmaceutical supply programs, whether it is performed by public employees or contracted out. There is computerized system to monitor port-clearing activities which are [54]:

- Managing preshipment cases, such as documentation, which are often required

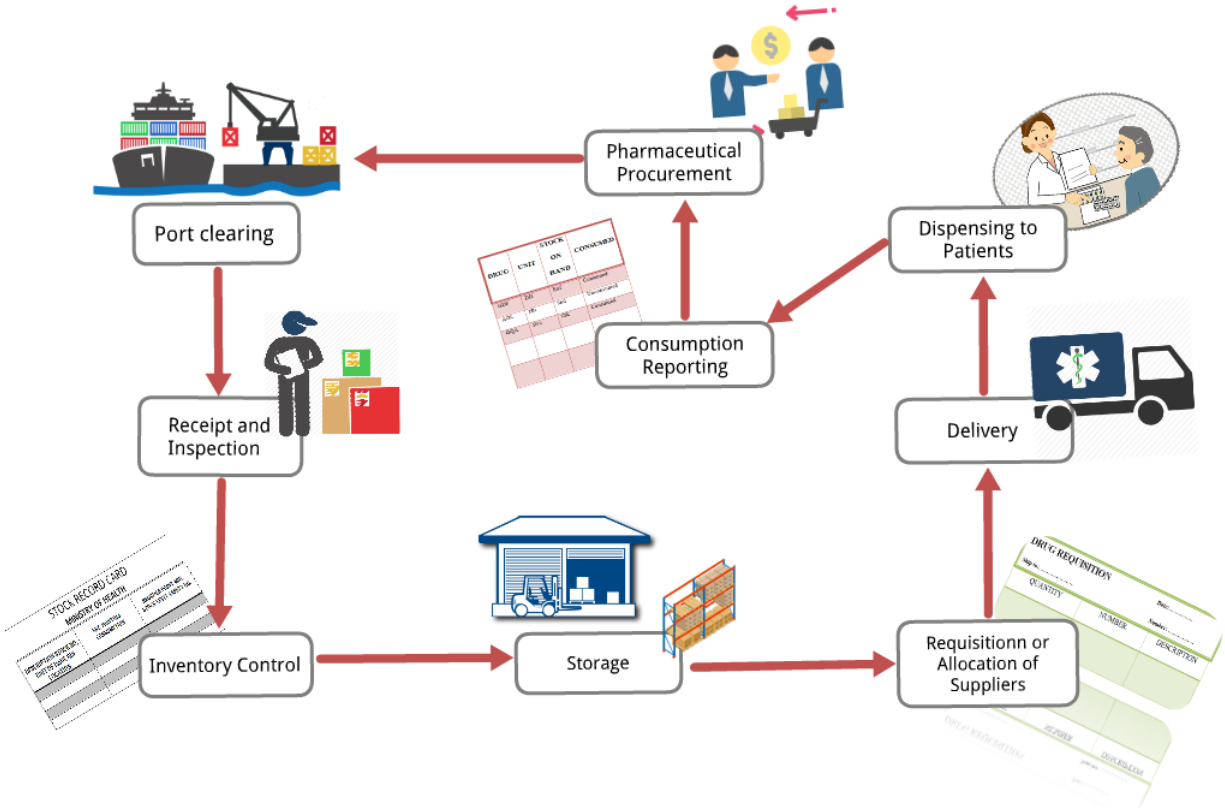


Figure 2.3: Pharmaceutical Distribution Cycle.

for clearance process.

- Identifying and anticipating the arrival of shipments as soon as they arrive in port.
- Storing medicines properly until they leave the port.
- Locating the shipments and the particular consignments.
- Obtaining the documents needed for clearing before the arrival of supplies at a port, and ensuring that the documents are in accordance with the country's port and customs requirements.
- Making timely payments relating to the clearance process.

2.5.3 Receipt and Inspection

Inbound management ensures that received goods are stored in a warehouse that includes receipt and the inspection procedures. During the inbound process, must carry out a complete inspection of each shipment as soon as it is received from the port or local supplier. Also, should check for damaged and missing items and for drug

type, quantity, packaging, presentation, labeling, etc. Quick and exact inspection of all shipments is fundamental to ensure that suppliers fulfill their contracts. The shipping and receiving processes require several other important documents that also can be electronic, including the material packing slip, the bill of lading, and the receiving discrepancy report [96].

2.5.4 Inventory Control

Inventory control is the processes used to maximize a company's use of inventory. The objective of inventory control is to produce the maximum gain from the minimal amount of inventory investment without intruding to customer satisfaction levels. Given the impact on customers and gains, inventory control is one of the main concerns of businesses that have big inventory investments, such as retailers and distributors. Inventory control is a opener to offering a cost-effective and responsive distribution system [70]. This includes [38]:

- It is a system for distribution and a primary protection against theft and corruption.
- Bar-code scanner integration.
- Reorder reports of consumption balance and adjustments.
- Product details, histories, and locations.
- Comprehensive inventory lists and counts.
- Syncing stock on hand with sales orders and purchase orders.

2.5.5 Storage

Storage in a warehouses allows time for pharmaceuticals to be quality-tested before being released to the market. Such warehouse require adequate security to deny theft or transformation of pharmaceutical shipments and also should have the capacity to store medicines at the right conditions, including appropriate temperature, to maintain medication quality, minimize theft and loss through damage [106].

2.5.6 Requisition of suppliers

The models and procedures for requisition are a main part of the inventory control system. They may differ from nation to nation. The requisition system may be manual or computerized or a both, but it should always be designed to simplify distribution by

facilitating inventory control, offering an audit path for tracing the flow of medicines, helping in financial accounting, and listing medicines issued.

2.5.7 Delivery

Medicines may be delivered by warehouse or collected by health facility staff. Transport may includes air, water, railway, etc. Cost-effective choices between public- and private- sector carries need to be made. Transport managers have to select routes of transportation neatly and schedule deliveries to offer punctual and economic service [55].

2.5.8 Dispensing to Patients

Dispensing is one of the vital keys of the rational use of medicines. The distribution process abstains its goal when medicines reach hospital wards, outpatient clinics, health centers, or community health workers, then given to patients through physician practices. Drug dispensing at the point of care has become a safe, efficient and cost-effective way of assisting patients manage their treatment programs [39].

2.5.9 Consumption Reporting

The flow of information on consumption and stock balances is a final link in the distribution cycle, to the procurement Office, for use in determining procurement needs. When sufficient inventory and requisition records are kept, consumption reports are aggregated directly.

2.6 Drug Supply Chain in Algeria

The pharmaceutical market in Algeria is seeing the same as the developing countries. And that is from the import of most of the medicines consumed. The medicines are branded and imported. Most pharmaceutical imports are from Europe, USA and Middle East countries [132]. Nevertheless, the Algerian government is trying to increase the local pharmaceutical industry. Local pharmaceutical companies can be classified into two groups:

- The first group is to import the final products and distribute them to the local market through private distribution companies or affiliated companies.
- The second group consists of local manufacturers that manufacture either for themselves or for other companies.

The pharmaceutical industry in Algeria has set its goal of changing and improving techniques to ensure the of domestic and external investment, with the goal of securing market coverage of domestic production of up to 70% in 2014 [41]. The sector in Algeria has assisted significant growth rates. In addition, the Office of Health has performed a new system of pharmaceutical supply to public institutions in order to ensure the availability of pharmaceutical. This technique adds to the measures indeed taken by the government to clean up the field of drug distribution, and to improve and redevelop the management of critical products [41].

2.6.1 Distribution System in Algeria

The pharmaceutical sector in Algeria has expanded with the healthcare system by gradually adjusting to the evolution of the level of national demand for pharmaceutical products. The institution of the facility reflects a broader trend to increase investment in the local pharmaceutical industry. In fact, the country became home to the continent's largest drug production and distribution facility in October 2018, with several complexes opening [40]. The Figure 2.4 describes the pharmaceutical distribution system in Algeria and its interactions with both public and private sectors in various levels.

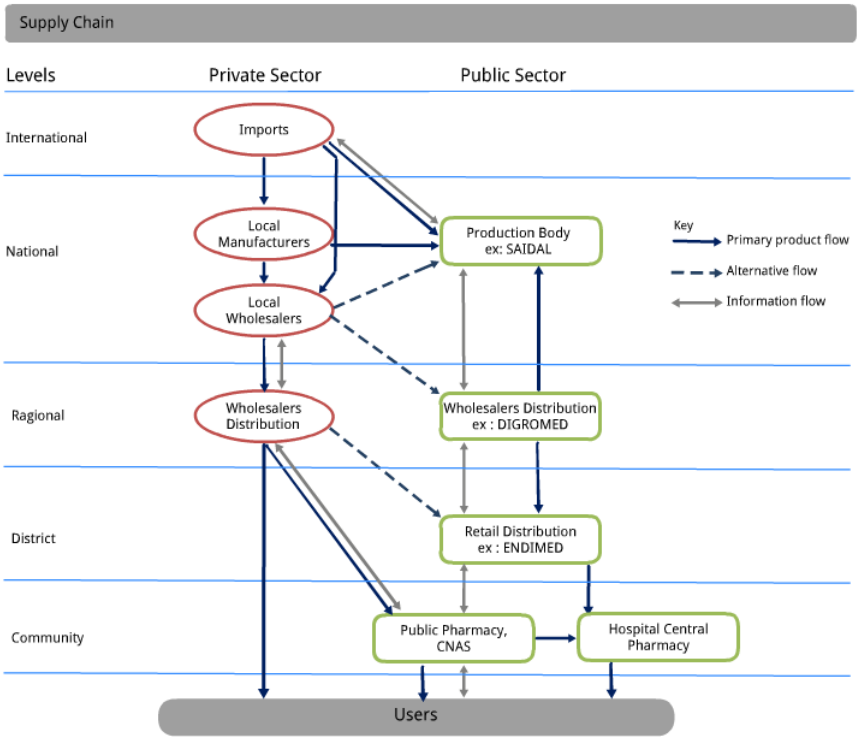


Figure 2.4: Pharmaceutical Distribution System in Algeria.

2.6.1.1 Public Sector

Actors in the general pharmaceutical sector rely on functional separation between the various bodies through production, import, wholesale and retail trade, as well as pharmacies and hospitals [77]:

1. **Production Body :** represented by SAIDAL, a public company established in 1982 and started to acquire the former central pharmacy production units in Algeria. The National Drug Company is run as a private enterprise with full administrative autonomy despite being 80% state-owned. Its double function is to consolidate its leadership position as a public local manufacturing company and consolidating the cause of the national drug policy implemented by the government as a controlling shareholder.

2. **Wholesale Distribution :** represented by DIGROMED who, since 1997, has held a network of former public import companies. And it was assigned the task of distributing in bulk and half the pharmaceutical materials. For several years, DIGROMED began diversifying its activity by manufacturing generic drugs [102] before dissolving in 2009 [108].

3. **Retail Distribution :** represented by ENDIMED, which is concerned with the operation of the retail distribution network for pharmaceutical products and the financing of public agencies. This network, which represented nearly a thousand pharmacies, was dissolved across the national territory, and pharmacies were allowed to rehabilitate pharmacists [108].

4. **Public Pharmacies :** responsible for providing a free list of drugs funded by the National Social Insurance Fund (CNAS) for disadvantaged and chronically ill people with very low returns.

5. **Hospital Central Pharmacy :** (PCH) is a public foundation of an industrial and commercial nature implemented to coordinate and rationalize public hospital supply programs.

2.6.1.2 Private Sector

Private sector actors have witnessed a rapid and significant development in the import and distribution of medicines (in large quantities). From 25 percent in 2008 to 65 percent in 2018. Algeria is allowed to reduce its dependency on imports [40]. Represented by [77]:

1. **Private Manufacturers, Importers and Private Retail Pharmacies :** From 1963 to 1990, the pharmaceutical market was completely under country

control. Since then, the suppression of the country monopoly has allowed the emergence of private sector companies in import, production and distribution.

2. **Private Wholesalers-Distributors :** They are responsible of the wholesale supply of various retail pharmacies over the national region. Their activity is controlled by by Decree No. 59 / MSP of July 20, 1995 which sets out the terms of activity for the wholesale distribution of pharmaceutical products. These wholesaler-distributers have a fundamental regulatory function and represent an important means of passing on economic information on the market, products and consumption habits [102].

2.7 Drug Supply Chain Challenges

The drug supply chain extends from the manufacturing, importing, and production companies of raw chemicals and biological components to the formation of a drug for patients and consumers. All of this is behind the scenes shipping, receiving and support an industry with customers that rarely can bear a disruption in supply.

The current supply chain approach to the transportation and distribution of medicines brings with it unique challenges, or limitations. In general, some of the main limitations are encountered in today's supply chain applications [93]. we will illustrate this below:

- **Lack of Transparency :** Lack of transparency always leaves a reputation at risk. Most often, supply chain participant mange their own data using traditional databases that don't provide data transparency by default [28], which makes a separately managed data system difficult to check how items are handled at every stage in the supply chain.

- **Lack of Traceability :** Due to the limited transparency of the supply chain stages, a major challenge with traceability is the ambiguity of product information, which results from recording ambiguous and unconfirmed product characteristics that are difficult to trace, making originality and authenticity demands of product hard or impossible.

- **Stakeholder Distrust :** Trust in any supply chain necessary to share critical information like costs, prices, etc [131]. Distrust among participants is the single biggest obstacle to improving supply chain networks [112]. By relying on the central system and third-party intermediaries as trust agents and verifying transactions and services, this greatly increases the operational cost and reduces the efficiency of the process.

- **Transport Delays :** Timing is critical to ensure that a product reaches the final consumer. Transporting products from one participant to another demands synchronization between enterprises, it may be on the production level or transport long distances. This can cause delays at each stage, leading to accumulative delays during the supply chain.
- **Data loss :** Supply chain extends to several destinations to provide products and services at the right time and place. Each participant can receive, send and manage his own data. Even with decades-old ways currently used, such as Electronic Data Interchange (EDI), which used in supply chain management to facilitate the transfer of a wide range of documents [29], or by using paper-based documentation. Some data items might not be sent along from one participant to another, resulting in data loss along the supply chain.

2.8 Conclusion

Supply chain operations are complex and require extensive management. The effective use of contractual logistics services places companies in an advantageous position in terms of chain management and agreement of their distribution operations.

Drug Supply chain challenges in Algeria constitute an obstacle in the pharmaceutical supply chain system, the risk of fraud and falsifying in traditional systems is significantly high. It needs a well-designed and reliable platform, and a way to implement and use this basic system. Drug manufacturers and distributors should consider using a platform aggregation program to address these challenges and simplify the supply chain.

Since the beginning of the digital age, organizations have been looking for improvements to their current business structure and significantly provide communication and transparency along the drug supply chain through modernisation and the emergence of new technologies. Blockchain is a digital technology that is able to support current processes and disable complex models [110]. The technology makes the process paperless that all the involved parties may interact with each other by using public and private keys. Industries can gain the advantage by adopting the technology into their business, it provides the necessary connectivity, increased security and full transparency to make the supply chain vision come true.

In next chapter, we will suggest a new simple drug supply chain system using Blockchain technology to handle secure drug supply chain records. The proposed system solves this problem by recording transactions of supply chain processes on a Blockchain basis to create an intelligent ecosystem to reduce the challenges and problems faced by the Algerian pharmaceutical supply chain.

Chapter 3

Design and Implementation

3.1 Introduction

Blockchain technology was introduced to the world, especially to our economic reality through the cryptocurrency Bitcoin, which was the first blockchain-based cryptocurrency. However, the use of blockchain is not only limited to financial transactions, rather, on everything of value. This technology extends to several areas such as Supply Chain Management, it provide data visibility for the entire flow and allows to track goods from raw material to end consumer.

Among the areas that can benefit from Blockchain solutions is the pharmaceutical supply chain. Due to the inefficiency of data sharing in existing supply chain networks, it has significantly affected the operations of manufacturers and retailers. Blockchain offers a promising future and allows the supply chain to provide better visibility, transparency and accuracy of transactions throughout the entire process.

As it is mentioned previously, in order to visualizing information, overcome challenges and improve supply chain performance, a system is needed to register history of products and services during the chain.

In this chapter, we will implement a simple system using Ethereum Blockchain to handle secure records, and for information service provided by participants in pharmaceutical supply chain. By using Ethereum smart contracts, joint supply chain business is accomplished. This proposed system solves problems by conducting drug distribution transactions on a Blockchain basis, to eliminate the need for system administration by a third party. Thus users can trust the information they see in the system as the data is not tampered with in any way.

3.2 Proposed System

In this section, we propose a blockchain system architecture that helps us address drug supply chain problems. As shown in the figure 3.1.

The great features of blockchain technology that is helpful in traceability, security, transparency. A blockchain-based system is inserted to provide a secure decentralized tracking system. System architecture relies on the Ethereum blockchain and smart contracts to remove the need for third-party system management. The application consists of a smart contract for Ethereum, which contains the processes of supply chain, in addition it have provided the ability to store and retrieve records from the blockchain ledger, which makes it easy to track the product and ensure that the data cannot be changed.

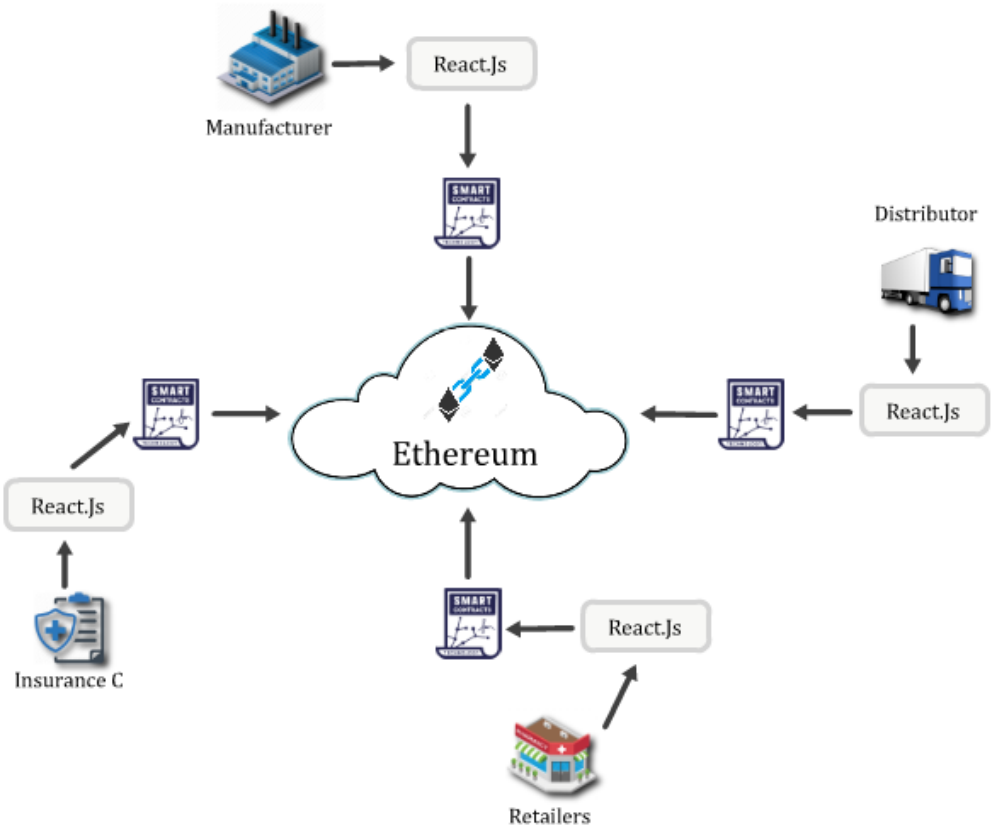


Figure 3.1: Architecture of the System

3.3 System Components

In this section, we discuss the major components of the proposed Ethereum system architecture in the pharmaceutical supply chain as shown in Figure 3.1. Figure 3.2

illustrates the system in more detail.

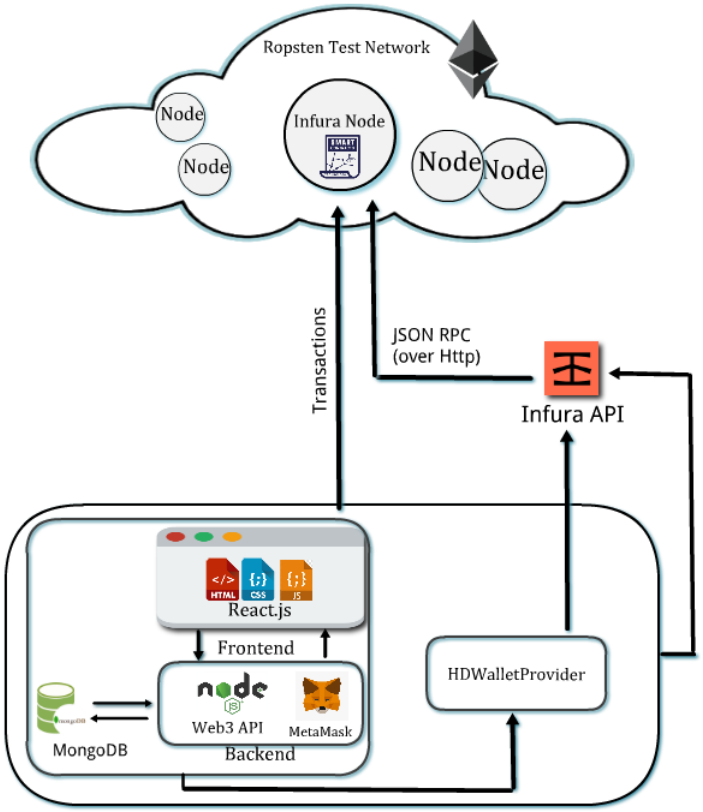


Figure 3.2: Detailed Architecture of The System

3.3.1 Blockchain

Blockchain can enable more transparent and precise tracking in the supply chain to help reduce fraud for goods such as pharmaceutical drugs., through digitize physical assets and create a decentralized immutable register of all transactions, making it possible to track assets from production to delivery or use by end user. The blockchain used as the main component in this system. By using this technology, provides all participant within supply chain with access to the same information, potentially reducing communication or transfer data errors.

The main goal of the solution is to detect details of digital assets created in the system. Wherefore, we want to save the data as it is, and once it is added to the network, no one can change or modify it until managing the system. Precisely, we chose the Ethereum platform over other blockchain platforms like Hyperledger. Hyperledger is best appropriated for creating confidential transactions within a network. Further, we have tried to detect all the minor details of the origin to the users and making it verifiable by all stakeholders in supplychain. All information concerning to accounts or

transactions that happened on the network can be checked at Etherscan.io, according to the account address of each member.

3.3.2 Smart Contracts

Smart Contracts are software programs that are used by all members of the supply chain to initiate and execute transactions and various rules of the transactions are enforced by the Smart Contract. The contract is then deployed on a test Ethereum network called Ropsten, a testing network that runs the same protocol as Ethereum does and is used to testing purposes before deploying on the main network (Mainnet). We connect to a specific node on the Ropsten network, so the contract can be published to that specific node.

The figure3.3 shows sequence diagram of the smart contract and the relationship among participants. The contract contains the functions and actions of supply chain, we will describe the function of each participant step by step.

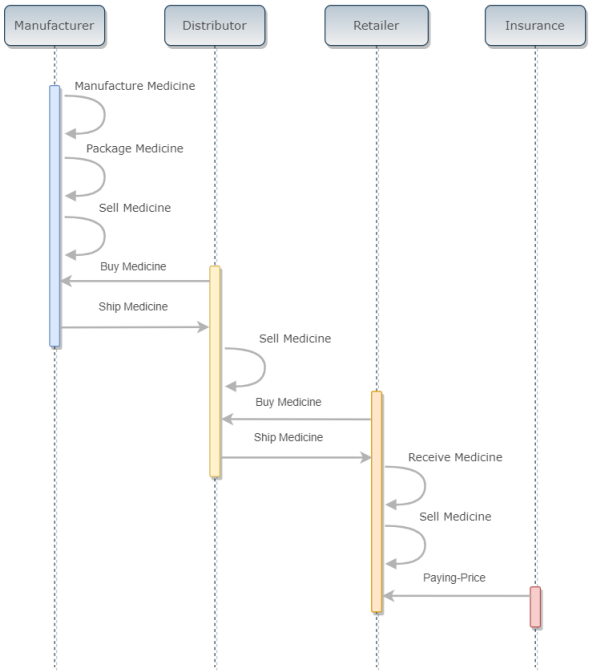


Figure 3.3: Sequence Diagram of Smart Contract Functions

Step 1: A manufacturer produces the drugs, containing essential information like product code, drug name, quantity, name and information of manufacturer. The information added by the manufacturer gets stored on the blockchain, making it possible for other stakeholders to trace the drugs' supply chain transparently. After manufacturing, Manufacturer packages the drugs and sell to distributors, they will put the money into a smart contract up front and as soon as the shipping company lets the smart contract

know that they've picked up the order the smart contract will automatically release the funds.

Step 2: Once the logistics service(manufacturer) providers deliver the drugs to distributors, they can verify the origin of medicines with the help of product code (PC) stored on the blockchain. They can trace back the information added by manufacturers such as the quantities of medicines, where it was manufactured. Distributors validate the received medicines and sign the transaction digitally which is then added to the blockchain. The signed transactions trigger the smart contracts to ship drugs to the hospitals/pharmacists. As soon as retailers indicate that they have received the shipment the smart contract will release the shipping company's payment.

Step 3: Retailers get the drugs which can be traced back to know its origination using the product code (PC) saved on the blockchain. If any illegal distributor tries to steals a quantity of the distributed medicines or the delay in delivery, the transaction is considered invalid because of the fraudulent information added about the distribution process. Therefore, retailers would immediately get to know if any anomalies are found within the transactions. Once the pharmacist approves the received medicines, the transaction between them and the distributor is added to the blockchain, ensuring the legal deal between them. Also, retailers sell the drugs to clients and is added transaction on blockchain.

Step 4: Usually, insurance companies bear the medical costs, such as paying for medicines. Insurance Company in return negotiate with the manufacturers to get discounts and can decide on which drug the patient by consumer ID receives with the help of product code (PC) stored on the blockchain. Through transactions that have been added by retailers through the sale of customers. Therefore, insurance company pay a percentage of drugs prices through the smart contract, this transaction adds to blockchain, and can anyone check it.

Smart contracts help significantly reduce losses and cost of transportation for pharmaceutical companies with their ability to custody, send and receive digital currencies, smart contracts will automate a lot of work that's being done now by hand.

3.3.3 Infura API

One simple way to succeed access to various networks without having to set up a complete nodes for each one yourself is to sign up for an Infura account. Infura maintains its own infrastructure that provides easy access to various blockchain networks as Ethereum. Infura offers secure, trustworthy, scalable, and easy to use APIs to access the Ethereum network and other decentralized platforms. It is a hosted Ethereum node cluster that let users run application without requiring them to set up their own Ethereum node or wallet. Infura accounts allow you to deploy code and interact with

mainnet, Ropsten, RinkeBy, and Kovan networks. We use Ropsten Testnet because it more like Ethereum and we can easily get fake ethers. There is no need to pay real ether to interact with the Ethereum blockchain but can get a sense of the real flow from an end-user's perspective.

3.3.4 Web3

One of the most widely used frameworks for designing DApps is Web3.js. Web3 makes it easy to interact with smart contracts on the Ethereum blockchain by applying the interface. Ethereum Blockchain provides us with web3.js, which is a useful API to make life a web developer easier. The JavaScript API enables us to communicate with an Ethereum node using the JSON RPC endpoints exposed on top of the HTTP, IPC or WebSocket transfers from the web page, through give json interface of smart contract and web3 will auto convert all function into low level ABI calls over RPC.

3.3.5 HDWallet Provider

The HDWallet provider is a appropriate and easy to configure network connection to ethereum through infura.io. The provider add some characteristics that are not available with infura like event filtering and transaction signing by using the 12-word mnemonic to unlock MetaMask account and use the account to deploy the contract.

3.3.6 Backend and Frontend

The contract controls all essential funds and functions, but in order to facilitate access to the contract for users, there must be a way for the user to interact with the contract using a website with buttons connected to the contract functions. The front end of this system is created with React which helps implement the interface to show the web page content to the user faster. We chose React for the front end development because it is very popular among front end frameworks and is not clustered compared to other libraries. It is also easy to learn, fast and scalable allowing HTML to be displayed on a web page.

We implement the backend by developing an API with NodeJS/Express, and the document database MongoDB

- NodeJS is a free, open-source server environment which uses JavaScript and runs on multiple platforms [50].
- ExpressJS is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications [5].

- MongoDB is a document database with significant scalability and flexibility. It has the following features [6]:
  - MongoDB stores data in flexible, JSON-like documents
  - As a distributed database at its core, MongoDB provides availability, horizontal scaling and geographic distribution.

  We use the mongoDB database to store the data in an orderly manner and make it easy for the user to view.

We are calling this api using React and Redux, is framework that has action and reducer which we are using to call that API. React hit our backend api and our backend call some contract function using web3 and ultimately send data to blockchain and send success message to frontend user.

3.4 Restoring Data from Blockchain

The main objective of this work is to provide better visibility, transparency and accuracy of transactions throughout the entire supply chain process, which is maintained in blockchain technology. To reach this objective, we need to restore the data that was recorded in the blockchain. Because transactions are served to an Ethereum node only through a text encryption format called JSON RPC, which is a lightweight Remote Procedure Call interface that uses JSON as a data structure to model the data that is sent to the blockchain. The participating nodes display this interface within the same process, over sockets, over HTTP, or in many various message passing environments. In this project, HTTP connections are used to send transactions over JSON RPC that plays a major role when we want to restore data from the blockchain as user accounts, send transactions, interact with smart contracts, and more. Users or applications can send JSON RPC calls directly to a node, forming the required JSON data structure and sending it to the exposed interface. However, this is heavy and time consuming. Therefore, programmers support libraries of different programming languages. This allows programmers to work in application language and create blockchain interactions, such as sending a transaction. This is then automatically translated into JSON RPC format and sent to the Ethereum node like Web3.

Web3.js is A JavaScript library to aid us evolve websites or clients that interact with blockchain, and write code that reads and writes data from the blockchain using smart contracts. Figure 3.4 shows how web3 is used in the system. In order to read data from the blockchain using web3, an example of a smart contract was implemented in JavaScript representations.

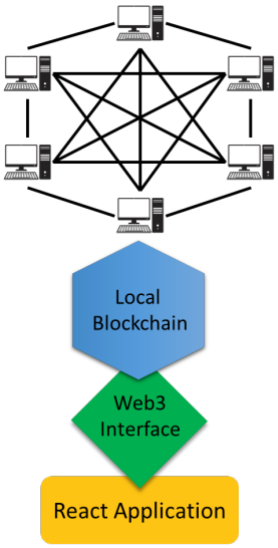


Figure 3.4: Usage of Web3 in Blockchain

3.5 System Interactions

Figure 3.5 shows the process of manufacturing, selling, buying, shipped, and viewing the history in the system. All participant of drug supply chain need to register in the system through MetaMask. After setting up a password, a twelve word account mnemonic and address. By having twelve word account mnemonic users can access their account any where with any browser they prefer. When manufacturer run the app it open the Metamask he give there private key and login to Metamask so our first step manufacturer attached with Metamask. At the main page, manufacturer can create a new product in the system by entering information, including Id, name and Information manufacturer, code, name, quantity product. After completing information on the form by clicking manufacture button and it is stored in mongodb database. Then, manufacturer pack a product by clicking Pack button. After this, manufacturer can be put up a product for sale through product code and price by clicking ForSale button. Automatically, Metamask confirms each action's transactions by deducting the stake amount from an account of manufacturer.

However, in this app, we connect with Metamask with first address of user Metamask. Later, we allow users to select any account from Metamask instead on first one. Distributor can buy a product he wants by entering name, price and ID product, quantity and distributor ID, this information will save in mongodb database. The manufacturer ship a product to distributor, through adding a transaction by clicking Ship button. After this step, distributor needs to confirms the delivery to let the core of the blockchain or the smart contract know that he received the product with required quantity. Also distributor can put a product for sale to retailer through prod-

uct code and price by clicking ForSale button, with needs to confirm a transaction or reject. when click on any button Metamask automatically confirms a transaction and records it to Blockchain. The same functions between distributor and retailer (buying, receiving and shipping).

On other side, when the retailer sell the product to consumer, he adds the information of product with the consumer ID in form by clicking sell button, a MetaMask has automatically deducting the amount of stake from the his account. Meanwhile, the insurance company can pay a percentage of the amount for consumer, by entering a name, price product, consumer ID and insurance ID by clicking Paying-Price button. The transaction is confirmed by automatically deducting the amount of stake from the account to add a transaction on the blockchain. When the insurance company confirms that he paid a percentage price the product to consumer, the smart contract releases the money that was deposited to retailer.

All information will save in an organized manner in mongodb database and and displayed to the user in an easy and clear way. Each participant is allowed to view all transactions of a supply chain.

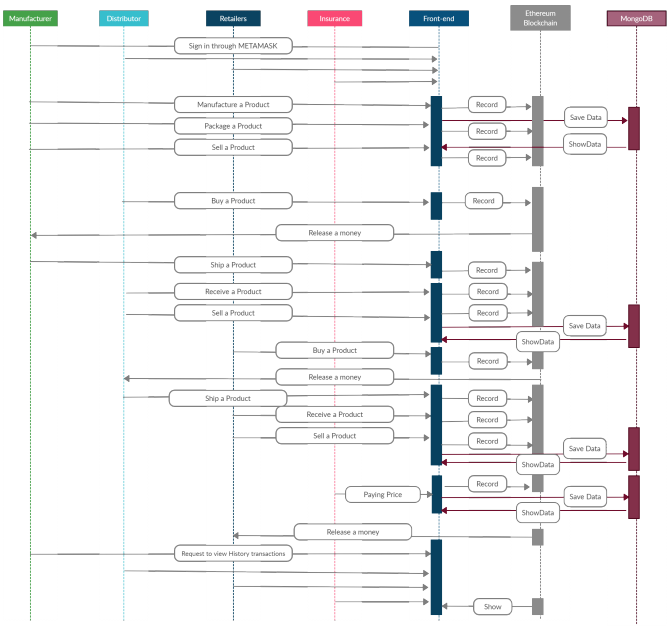


Figure 3.5: System Interactions

3.6 Development Tools

In this section, we present the system configuration, operating system, tools, language and environments that we used to implement the tool.

3.6.1 System Configuration and Operating System

The project are performed on CPU 2.30 GHz 2.40 GHz Intel Core i5, with 4 logical cores and 8 Go of memory. We implement the project using Windows 10.

3.6.2 Remix IDE

An alternative to installing an Integrated Development Environment IDE on your own computer is to use a browser based IDE. Remix is a popular IDE that you can access from any web browser. It enables you to write code in Solidity, and then deploy to a blockchain [93].

Remix IDE is a web application that can be used to write, debug, and deploy Ethereum Smart Contracts [94]. Remix also provides good tools for debugging, static analysis, and deployment all within the online environment. We use the remix IDE to create and deploy online smart contracts for drug supply chain using solidity language, accessible at `https://remix.ethereum.org`.

3.6.3 Visual Studio Code

VSCode is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript, Solidity and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity) [48]. We implement the project with VSCode editor because free and open-source, (meaning a program's code can be viewed, modified, and shared), also it has features like IDE and easy to use.

3.6.4 Truffle

Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier. It allows developers to spin up smart contract project at the click of a button and provides you with a project structure, files, and directories that make deployment and testing much easier [47].

The most characteristics that make Truffle one of the most widely used IDEs for Ethereum Blockchain are [52]:

- It has built-in support for compiling, deploying and linking Smart Contracts.
- The Truffle Console allows you to work with your compiled contracts in a hassle-free manner and using your preferred choice of RPC client.

- It has a configurable build pipeline that supports both console apps and web apps.
- It comes with built-in support for JavaScript.
- It has generators that help in the creation of new contracts and tests.
- It allows for the instant rebuilding of assets during the development stage.

To install the Truffle framework :

```
npm install -g truffle
```

3.6.5 Ganache

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. You can use Ganache across the entire development cycle; enabling you to develop, deploy, and test your distributed Applications in a safe and deterministic environment [49].

Ganache UI is desktop application supporting both Ethereum and Corda technology. In addition, an Ethereum version of ganache is available as a command-line tool: ganache-cli (formerly known as the TestRPC). All versions of Ganache are available for Windows, Mac, and Linux [49].

This will allow us to deploy smart contracts, develop applications and run tests. We chose Ganache because it provides us with 10 Ethereum accounts with a balance of 100 ether (fake ether) for each account, as well as a graphical interface that allows us to check everything that happens in this blockchain.

3.6.6 Node.JS

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices [50].

We have to configure our environment to develop smart contracts. The first dependency we will need is Node Package Manager (NPM), provided with Node.js.

3.6.7 React

React is a JavaScript library for building interactive user interfaces. It helps developers define interfaces such as functions and procedures. He designed simple views for each

situation in their app and developed many new features, and React will update and display only the correct components efficiently when the data changes. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your application and keep the case outside of the DOM [4].

3.6.8 MetaMask

MetaMask is an extension for accessing Ethereum enabled distributed applications, or "DApps" in your browser. The extension injects the Ethereum web3 API into every website's JavaScript context, so that DApps can read from the blockchain. MetaMask also allows the user to create and manage their own identities, so when DApp wants to perform a transaction and write to the blockchain, the user gets a secure interface to review the transaction, before it is approved or rejected [53].

We chose Metamask because it also allows the management of Blockchain accounts, as well as Ether funds to pay transactions. Figure 3.6 shows MetaMask interface.

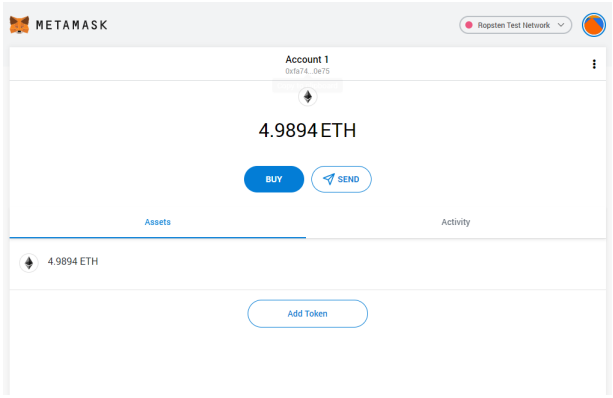


Figure 3.6: MetaMask Interface

3.7 Implementation

This section describes the implementation of the system and necessary steps to set up the drug supply chain web application for interacting with the blockchain.

3.7.1 Environment Configuration

First, we will create a directory that will contain the files of our project in Command Prompt like this :

```
$ md drug-supplychain-ethereum
```

```
$ cd drug-supplychain-ethereum
```

Now, we are starting a new Truffle project to develop our project. Truffle with version 5.0.0 can be installed using npm, using this command:

```
$ npm install -g truffle@5.0.0
```

Then, we run truffle init, this will set up the following basic structure in our directory shown in figure 3.7:

```
$ truffle init
```

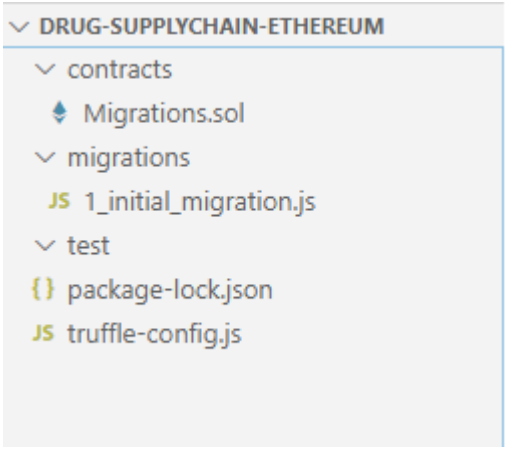


Figure 3.7: Directory structure of Drug-supplychain-ethereum

- **Contracts**: contains the code for our smart contracts.
- **Migrations**: contains the deployment instructions to our contracts.
- **Test**: contains the tests for the contracts
- **Truffle-config.js** (or truffle.js depending on your O.S.) : main configuration file, points to the Ethereum networks that we can deploy to

3.7.2 Writing Smart Contract

To build a decentralized supply chain application, we create the Ethereum smart contract first, named SupplyChain.sol file in the "Contracts / drugbase" directory. We should specify Solidity compiler version 0.4.24.

```
pragma solidity ^0.4.24;
// Define a contract 'Supplychain'
contract SupplyChain {
  address owner;
  uint  pc;
  uint  sku;
  mapping (uint => Product) products;
  mapping (uint => string[]) itemsHistory;
  enum State
  {
    Manufactured,  // 0
    Packed,      // 1
    ForSale,     // 2
    Sold,        // 3
    Shipped,     // 4
    Received,    // 5
    Purchased    // 6
    }
    }
```

Inside the contract form, we define four state variables to store the address of owner of contract, which is the account for deploying the smart contract, 'pc' for product code, 'sku' for stock keeping unit, 'Items' that maps the product code (pc) to an 'Item'. We define a public mapping 'itemsHistory' that maps the product code to an array of TxHash, that track its journey through the supply chain to be sent from DApp. In addition, we define states of supply chain processes. The item structure defines the details for each unique product described in the following code :

```
  struct Product {
    uint     sku;
    uint     pc; /
    address ownerID;
    address originManufactID;
    string  originManufactName;
    string  originManufactInformation;
    uint  quantity;
    uint     productID;
    string  productNotes;
    uint     productPrice;
    State    itemState;
    address distributorID;
    address retailerID;
    address insuranceID;
  }
```

A constructor is a special type of function that is automatically triggered when a

smart contract is deployed to a network. It cannot be called afterwards. The importance of writing constructor () as payable is that the sender's address must be obligated as owner. We set the owner to msg.sender, which is the representative of an embedded global variable of the address that calls the function.

```
1       constructor() public payable {
2         owner = msg.sender;
3         sku = 1;
4         pc = 1;
5       }
```

Modifiers are declared inside of the smart contract. They are containing a set of conditions that will enable or not the function to be executed. We specify six modifiers containing a conditions, that if '_pc' has gone through all stages of the supply chain(Manufactured, Packed, ForSale, Shipped, Received, Purchased). Modifiers can modify the state of the smart contract.

```
1       modifier manufactured(uint _pc) {
2         require(items[_pc].itemState == State.Manufactured);
3          _;
4          }
```

3.7.2.1 Supply Chain Functions

In this section, we define the functionality of our supply chain smart contract, we need to set its functions and provide the code for structure of every function.

1. The manufacturerProduct() Function

   We define a function *manufacturerProduct()* that allows a manufacturer to mark an product 'Manufactured'. This function increments the stock keeping unit (sku), creates a new product and sets its attributes to the passed values.

```
1    function manufacturerProduct(uint _pc,  address
        _originManufactID, string _originManufactName, string
        _originManufactInformation, uint  _quantity, string
        _productNotes) public {
2      products[_pc] = Product({
3        sku: sku,
4        pc: _pc,
5        ownerID: _originManufactID,
6        originManufactID: _originManufactID,
7        originManufactName: _originManufactName,
8        originManufactInformation: _originManufactInformation,
9        quantity: _quantity,
10       productID: sku + _pc,
11       productNotes: _productNotes,
```

```
12        productPrice: 0, // default value
13        itemState: State.Manufactured,
14        distributorID: address(0), // default value
15        retailerID: address(0), // default value
16        insuranceID: address(0) // default value
17      });
18      sku = sku + 1;
19      emit Manufactured(_pc);
20    }
```

2. The packProduct() Function :

   We define function *packProduct()* that allows a manufacturer to mark an Product 'Packed' and we set call modifier 'manufactured' to check if product code '_pc' passed previous supply chain phase and modifier to verify caller of this function.

```
1     function packProduct(uint _pc) public
2       manufactured(_pc)
3       verifyCaller(products[_pc].originManufactID)
4         {
5             products[_pc].itemState = State.Packed;
6             emit Packed(_pc);
7         }
```

3. The sellProduct() Function :

   We set a function *sellProduct()* that allows a manufacturer to mark an product 'ForSale'. we set a modifier to check if 'pc' passed preceding supply chain stage and modifier to verify caller of this function. In this case, the caller is manufacturer ID. Also, we update the price for sell of product. The same function between distributor retailer, but the caller is to distributor ID.

```
1         function sellProduct(uint _pc, uint _price) public
2             packed(_pc)
3             verifyCaller(items[_pc].originManufactID)
4             {
5                 products[_pc].itemState = State.ForSale;
6                 products[_pc].productPrice = _price;
7                 emit ForSale(_pc);
8             }
```

   In state between retailer consumer, the caller is retailer ID. we record the consumer ID that buy product. In other side, the insurance company pays a percentage of product price to benefit of the consumer.

```
1             products[_pc].itemState = State.ForSale;
2             products[_pc].productPrice = _price;
```

```
3             products[_pc].consumerID = _consumerID;
```

4. The buyProduct() Function:

   We define a *buyProduct()* function that allows a distributor to mark the product as "sold." We use modifiers to check whether the product is available for sale, if the buyer (distributor) has paid enough, and any excess ether sent is refunded to the buyer. The same function between distributor retailer, but the caller is to retailer ID.

```
1  function buyProduct()(uint _pc) public payable
2      forSale(_pc)
3      paidEnough(products[_pc].productPrice)
4      checkValue(_pc) {
5          products[_pc].ownerID = msg.sender;
6          products[_pc].itemState = State.Sold;
7          products[_pc].distributorID = msg.sender;
8          // Transfer money to Manufacturer
9          products[_pc].originManufactID.transfer(products[_pc].
               productPrice);
10         emit Sold(_pc);
11 }
```

   we define a modifier *paidEnough()* to check if the paid amount is sufficient to cover the price. Also, we set a modifier *checkValue()* to test the price and refunds the remaining balance.

```
1  modifier paidEnough(uint _price) {
2      require(msg.value >= _price);
3      _;
4  }
5  modifier checkValue(uint _pc) {
6      _;
7      uint _price = products[_pc].productPrice;
8      uint amountToReturn = msg.value - _price;
9      products[_pc].insuranceID.transfer(amountToReturn);
10 }
```

5. The shipProduct() Function:

   We determine a function *shipProduct()* that allows the distributor to mark an product 'Shipped' and we use modifiers to test if the product is sold. The same function between distributor retailer, but the caller is to retailer ID.

```
1  function shipProduct(uint _pc) public sold(_pc)  verifyCaller(
       items[_pc].distributorID)
2  {
```

```
3     products[_pc].itemState = State.Shipped;
4     emit Shipped(_pc);
5 }
```

6. The receiveProduct() Function:

   We determine a function *receiveProduct()* to let the retailer to mark an item 'Received'. We utilize a modifiers to check if 'pc' has passed previous supply chain stage (shipped by distributor).

```
1 function receiveProduct(uint _pc) public shipped(_pc)
2 {
3     products[_pc].ownerID = msg.sender;
4     products[_pc].retailerID = msg.sender;
5     products[_pc].itemState = State.Received;
6     emit Received(_pc);
7 }
```

7. The purchaseProduct() Function

   We set a function 'purchaseProduct' to allows the insurance company to mark an item 'Purchased', that it pays a percentage of product price and we also define modifiers to check if the item is received

```
1 function purchaseProduct(uint _pc, uint _consumerID) public
     received(_pc)
2 {
3     products[_pc].ownerID = msg.sender;
4     products[_pc].insuranceID = msg.sender;
5     products[_pc].consumerID = _consumerID;
6     products[_pc].itemState = State.Purchased;
7     emit Purchased(_pc);
8 }
```

3.7.2.2 Compiling and Deploying the Smart Contract

First, we compile the smart Contract to check it runs properly and there are no errors, using this command in console of directory of project:

```
$ truffle compile
```

When we compile our contract a new file is created, at the following location: '/build/contracts/SupplyChain.json'. This file is the ABI (Abstract Binary Interface) file that describes the specific structure of a contract, including the input function, interface functions, parameter list of functions, return value, and events.

Second, we need to write a migrating for deploy our smart contract on the development network Ganache (local blockchain). To do this, we create a new file in migration folder named "2_deploy_contracts.js" and we add the following code:

```
// migrating the appropriate contract
var SupplyChain = artifacts.require("../SupplyChain.sol");
    module.exports = function(deployer) {
        deployer.deploy(SupplyChain);
    };
```

To set up the Ganache blockchain development network, we will update truffle.config as follows:

```
module.exports = {
        networks: {
        development: {
            host: "127.0.0.1",
            port: 7545,
            network_id: "*" // Match any network id
        }
    },
    compilers: {
        solc: {
            version: "0.4.24"
        }
    }
};
```

Then, we run this command :

```
$ truffle migrate –network development
```

3.7.2.3 Testing the Smart Contract

Smart contract testing is essential in the blockchain development process. The test of smart contract will make sure that functions will have correct way. The truffle framework makes the testing process very easy.

In order to start testing, in the test folder, we create a file called TestSupplyChain.js. and we create test using JavaScript to simulate client-side interaction with our smart contract. Then, we run it to see the behavior of our smart contract using this command:

```
$ truffle test
```

Figure 3.8 shows the output of the smart contract test.

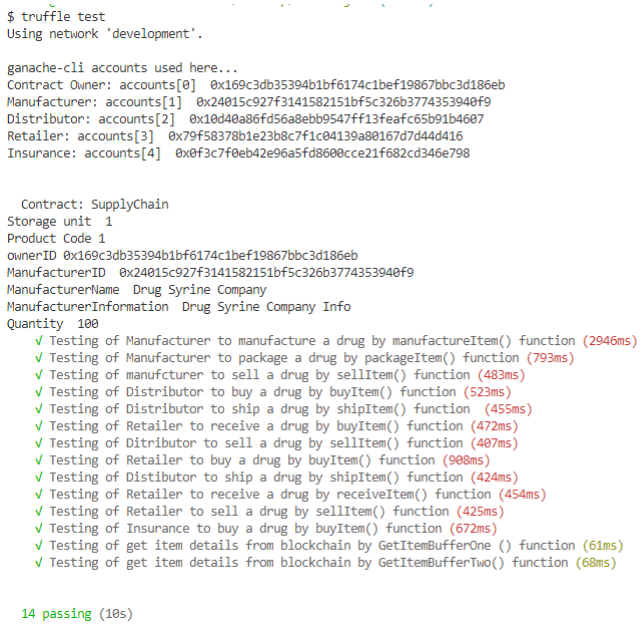


Figure 3.8: Testing The Smart Contract

3.7.2.4 Deploying Smart Contract on Testnet

In this step, we are deploying a smart contract on a real Ethereum Blockchain network that actually costs real Ether (transactions are verified and blocks are mined by real miners) using MetaMask and remix IDE. Mainnet/Testnet - These refer to two different Ethereum networks with separate chains. Mainnet is the primary live Ethereum blockchain (real “money”). Testnet is any testing environment where fake money can be used instead to test contracts [46].

We perform our test by deploying it on Testnet that are just a simulation of the real Ethereum network, currently there are three famous Testnet:

- **Rinkeby:** An alternative Blockchain test that uses a consensus algorithm called " Proof of Authority(PoA)" as we mentioned before in Chapter 1 consensus process subsection, means the need to demonstrate existence in order to retrieve ethers from a faucet, and a fixed block generation time.
- **Kovan:** An alternative test blockchain, its consensus algorithm is similar to Rinkeby but block generation time is faster.
- **Ropsten:** A blockchain test similar to the real Ethereum blockchain because it use similar "Proof of Work (PoW)" consensus algorithm, (i.e. it can be mined on) so the simulation of transaction confirmations is the most real.

We use Ropsten network to deploy and test the smart contract, we demand that network some free ether, it’s easy to getting ether from that network’s faucet. We

just need to navigate to `https://faucet.ropsten.be`, and entering account address of MetaMask. Figure 3.9 shows the Ropsten Ethereum faucet.

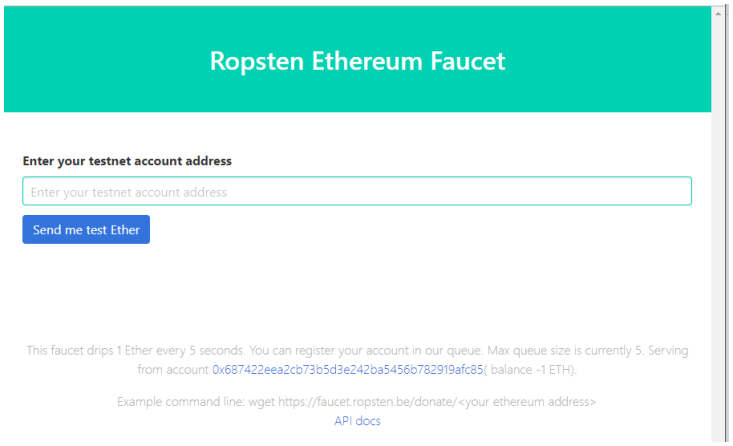


Figure 3.9: Ropsten Ethereum Faucet

Now, we copy the SupplyChain.sol and paste it into the remix IDE. As we knew it before, it is a web application that can be used to write, debug, and deploy Ethereum Smart Contracts. Then, we select Injected Web3 under environment The account in Metamask is shown here under account with balance ether.

We deploy our smart contract to the Ropsten Testnet, here's the cost required for the migration, as well as shown in figure 3.10.

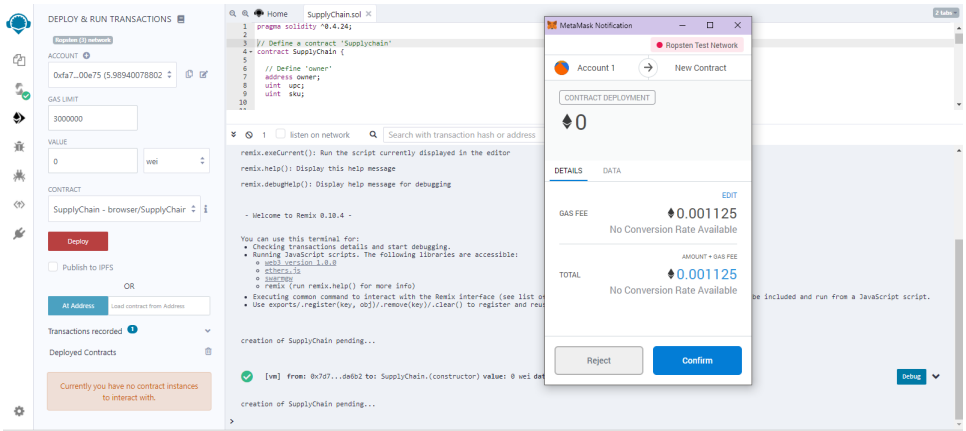


Figure 3.10: Remix IDE.

Etherscan allows to explore and search the Ethereum blockchain for transactions, addresses, etc,. Through Etherscan, we can access smart contract accounts, with all the events and transactions (failed and valid), all thing is writing in a transparent and regular way since its creation, shown in figure 3.11.

Figure 3.12 shows all the major functions of our post-release smart contract that the Remix IDE after the deployment. Under the name and address of the deployed contract,

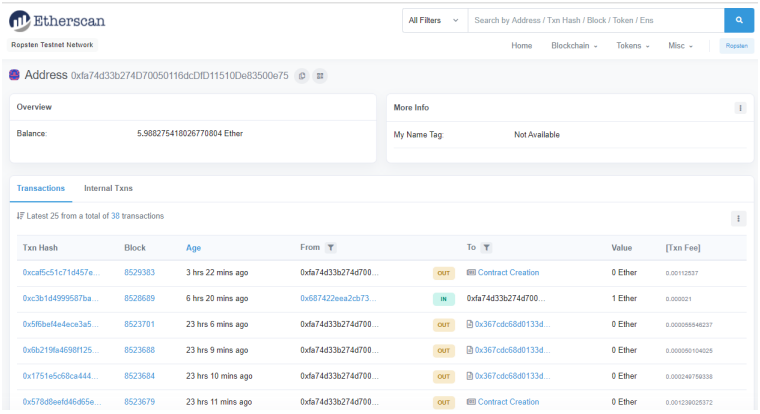


Figure 3.11: Our Smart Contract Account.

we have some buttons in the red, orange and blue colors. The red buttons indicate functions that cause a pay ethers to another participant and writing a transaction, the orange buttons refer functions that are writing to the blockchain and needing a transaction, where the blue buttons indicate reading from the blockchain.

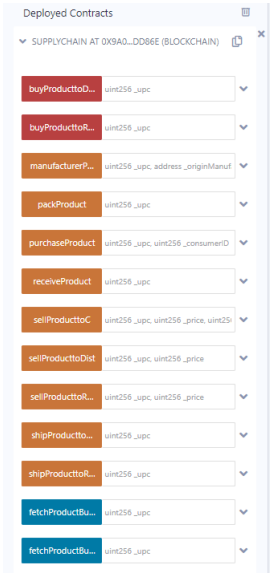


Figure 3.12: Functions of Smart Contract.

3.7.3 Web3 and HDWallet Provider

As we mentioned earlier web3 in section 4.4 system components, can be thought as communication between the web3 library and an ethereum network, has a set of methods which allow the web3 library to send a request to a local network and receive the response to that request. Also, we use HDWalletProvider that is provider to unlock account. It lets us to connect to the Rospten which is hosted by infura. Web3Js provides the web3 object that enables us to exploit the Web3 API functions in JavaScript.To

find and interact with our deployed contract blockchain, it needs to know the contract's address and its application binary interface (ABI) . The following script shows that we apply to create a web3 instance.

```
const accounts = await web3.eth.getAccounts();
const supplyChainContract = await new web3.eth.Contract(abi, address);
```

To utilize HDWalletProvider, the Truffle HDWallet must be installed, then by heading to infura.io, we need to register to get an Infura API key to use the service. By providing HDWallet with MetaMask Account Mnemonic and Infura API, we can create a enabled web3 instance of the Ropsten network. HDWalletProvider takes 2 arguments as input; The first is a mnemonic account which is used to open accounts and the second argument is the ethereum node that we want to connect. The script shows that we implement to connect to the Ethereum node and unlocking accounts with account mnemonic.

```
var HDWalletProvider = require("truffle-hdwallet-provider");
var mnemonic = "diet mistake resist blood pool process toss frequent
   zero judge crime equip"; // 12 word mnemonic
const Web3 = require("web3");
var provider = new HDWalletProvider(mnemonic, "https://ropsten.infura.
   io/v3/c67e8fb7d27f4228a1a13e7c5e23e2f1");
const web3 = new Web3(provider);
```

To interact with contract from React for system frontend implementation, a local instance of the contract was created.The contract ABI and the address at which the contract was posted to create a copy of the contract.

```
//Deployed contract ABI
const abi = [
   {constant: false,inputs: [
            { name: "_pc", type: "uint256" },
            { name: "_originManufactID", type: "address" },
            { name: "_originManufactName", type: "string" },
            { name: "_originManufactInformation", type: "string" },
            { name: "_quantity", type: "uint256" },
            { name: "_productNotes", type: "string" },
            { name: "_price", type: "uint256" },
        ],
        name: "manufacturerItem",outputs: [],payable: false,
           stateMutability: "nonpayable", type: "function",
    },
    ... ]
const address = "0xED07d16ff28B71f86a5b0F0B526bE36D84f085DA";
```

3.7.4 Backend and Frontend

Before frontend development, we implement the system backend using node.js and Express.js, which is an unknown popular web framework written in JavaScript and hosted in the Node.js runtime environment. We call each function of smart contract in backend in the following way, for example when we add a product :

```
const express = require("express");
    ...
      //First we add manufacturer product
app.post("/addProduct", async function (req, res) {

  const accounts = await web3.eth.getAccounts();
  console.log("account",accounts[0])

    const supplyChainContract = await new web3.eth.Contract(abi,
       address);
    try {
        const response = await supplyChainContract.methods
            .manufacturerItem(req.body.pc, req.body.manufactureId, req
               .body.manufactureName, req.body.manufactureInfo, req.
               body.quantity, req.body.notes, req.body.price)
            .send({
                from: accounts[0],
                gas: "3000000",
            });
        console.log("Response of function", response);

        res.send({
            data: response,
            success: true,

        });
    } catch (e) {
        res.sendStatus(500);
    }
});
```

We also define a port number and that needs to be used when starting our project.

```
app.listen(4000, function() {
    console.log("Server is listening on port", 4000);
});
```

The frontend of this system is developed by using React, it is an open-source JavaScript library used for frontend development to react with it via regular browser, and Redux, is framework that has action and reducer which we are using to call backend API, then

will interact with Ethereum blockchain via Web3 API. We invoke the function of the previous example with this way:

```
1       export const manufacturerItem = (productInfo) => async dispatch
            => {
2     dispatch({
3         type: ADD_LOADER,
4         payload: "LoadingManufacturer"
5     });
6     try {
7         const res = await axios.post(`${baseUrl}/addProduct`,
              productInfo);
8     } catch (err) {
9         dispatch({
10            type: REMOVE_LOADER,
11            payload: "Loading"
12        });
13        swal("Something went wrong", "Try again!", "error");
14    }
15 };
```

React is the main engine for developing the User Interface. It combines the HTML, CSS and Javascript to display data on screen. We insert a constructor, that initializes defaults state and define the action of each button from Redux actions. We insert a react life-cycle method called componentDidMount(), that will execute the getAllTransaction() method to get all transactions from Ethereum Blockchain, getAllProduct() method to get all products from mongoDB database after the component mounts. Last, we define the frontend components using HTML to display on our page to interact with user as shown with script:

```
1  import React, { Component } from "react";
2  import { connect } from "react-redux";
3  import Web3 from 'web3'
4  ...
5  class supplyChain extends Component {
6      constructor(props) {
7          super(props);
8          this.state = {
9              manufactureId: "",
10             manufactureName: "",
11             manufactureInfo: "",
12             pc: "",
13             quantity: "",
14             notes: "",
15             price: "",...,
16         };
```

```
17      }
18      onChange = e => {
19          this.setState({
20              [e.target.name]: e.target.value,
21              [e.target.name.concat("error")]: "",
22          });
23      };
24       manufacturerClick = async () => {
25          var productInfo = {
26              manufactureId: this.state.manufactureId,
27              manufactureInfo: this.state.manufactureInfo,
28              manufactureName: this.state.manufactureName,
29              notes: this.state.notes,
30              quantity: this.state.quantity,
31              pc: this.state.pc,
32          };
33          await this.props.manufacturerItem(productInfo);
34      };
35      componentDidMount = async ()=>{
36        this.props.getAllTransaction()
37        this.props.getAllProduct()
38      };
39      render() {
40      ...
41  <div class="container">
42      <form class="serviceBox" >
43          <span class="text-center">Manufacturer Details </span>
44              <div class="input-container">
45                  <label> Manufacturer ID </label>
46                  <input type="text" name="manufactureId" onChange={this
                        .onChange} value={this.state.manufactureId}
                        placeholder="0
                        x99c289eb2aacec289631a5ddf62cf27a63d4494f" size
                        ={50} id=""/>
47              </div>
48              ..
49          </div> ..
50      }
51  }
```

Figure 3.13 shows the main page of our drug supply chain system, creating, selling, buying a drug and viewing the history of transactions. All participant on this network can check all supply chain stage.

In Manufacturer part, can register a new drug in the system by entering its information including name and information of manufacturer, product code, name and

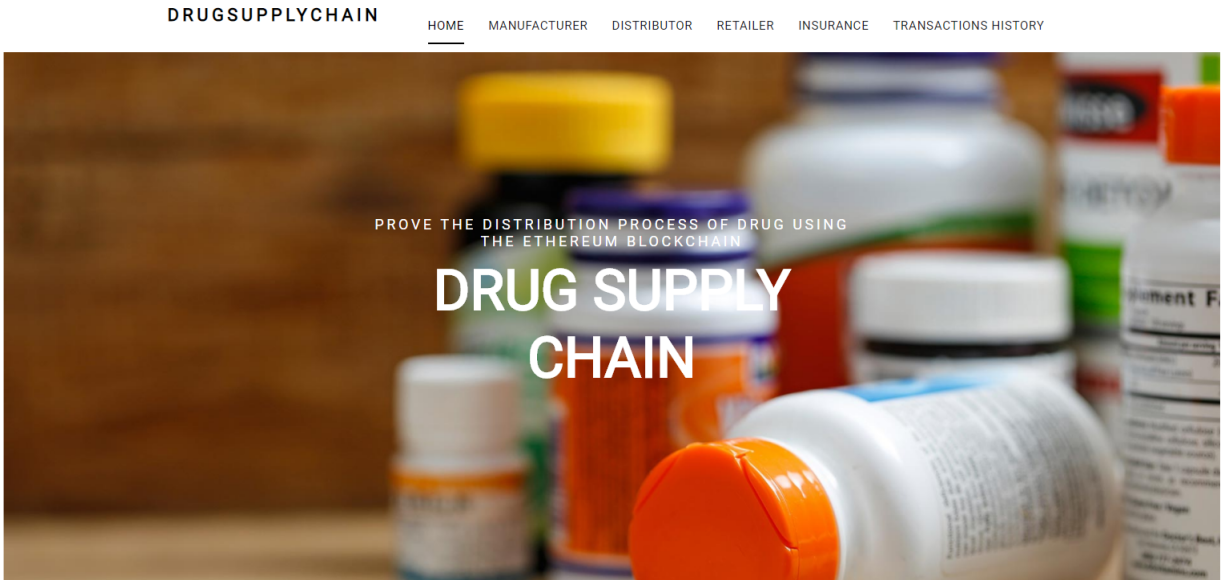


Figure 3.13: Home Page

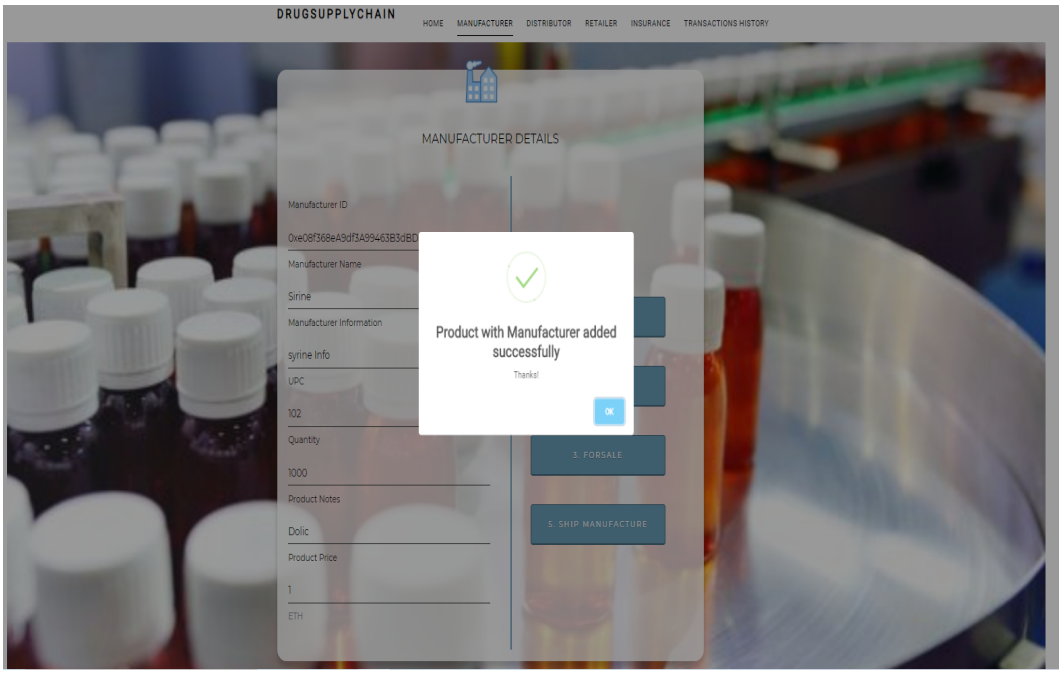
quantity drug. By clicking on the "Add Product" button, it will perform a transaction containing all the information about the drug for safekeeping in the blockchain, and stored in the orderly manner on mongoDB as shown in Figure 3.14. Also it can packing, selling, shipping a drugs.

In other side, distributor can buy a product based on entering the name, price product, in addition distributor Id. the same for retailer, can buying a product from distributor and offer it for sale to the customer by entering a name, price product, consumer Id and retailer Id. As for the insurance company, it can pay a percentage of price product to consumer by entering a name, price product, consumer Id and insurance Id. Each process, it will perform a transaction containing all the information for safekeeping it in the blockchain, and stored in the orderly manner on mongoDB, if there an error by entering the data it will show errors message and reject the transaction. Some of the supply chain processes are shown in Figure 3.15.

Users are shown all information and transaction log stages of the supply chain as shown in Figure 3.16.

3.8 Ethereum benefits for the Drug Supply Chain

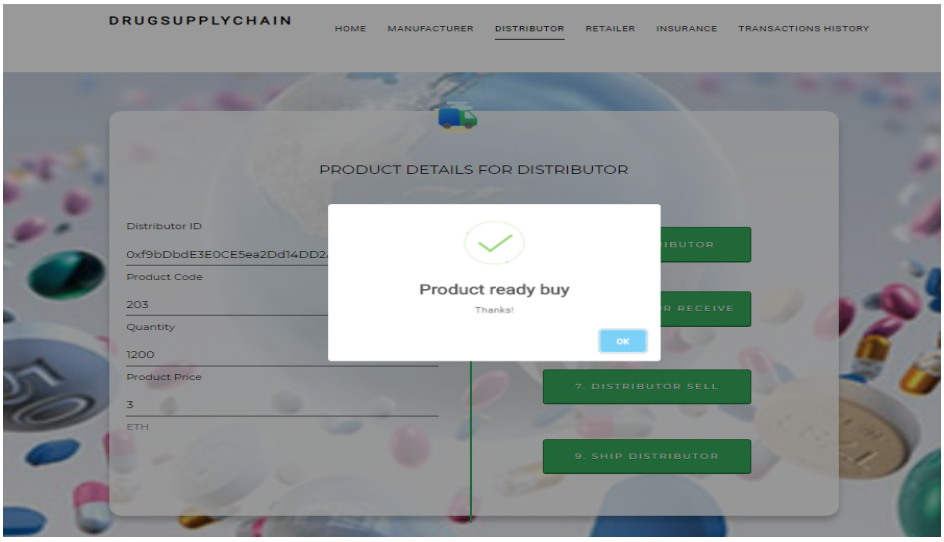
Ethereum Blockchain can help address many of the supply chain limitation and challenges, also can help resolve the majority of the deficiencies in today's supply chain applications. We will list the limitation from drug supply chain challenges section, and how our system can help resolve each one.



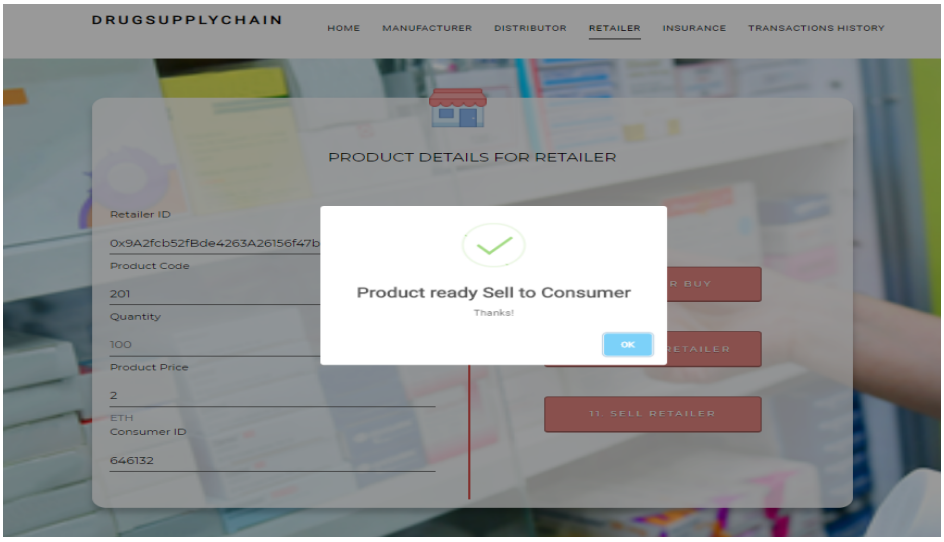
DRUGS OVERVIEW

PRODUCT CODE	NAME	QUANTITY	MANUFACTURER NAME
2001	tX ACID	10000	syrine
102	Dolic	1000	Sirine
98269514	Asperine	1000	Saidel

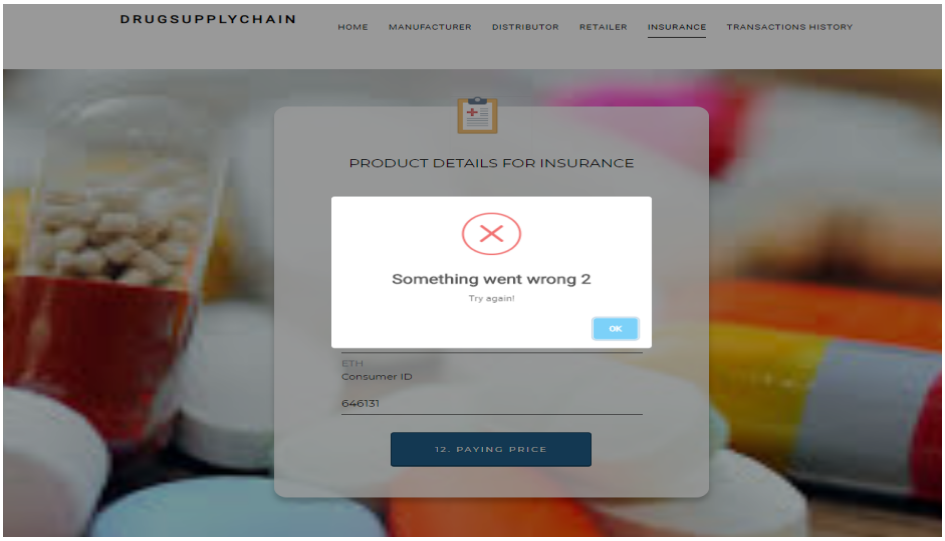
Figure 3.14: Add Product by Manufacturer



- Buying a Drugs from Manufacturer by Distributor



- Selling a Drug to Consumer by Retailer



- Paying a Percentage of Price by Insurance.

Figure 3.15: Buying, Selling Processes

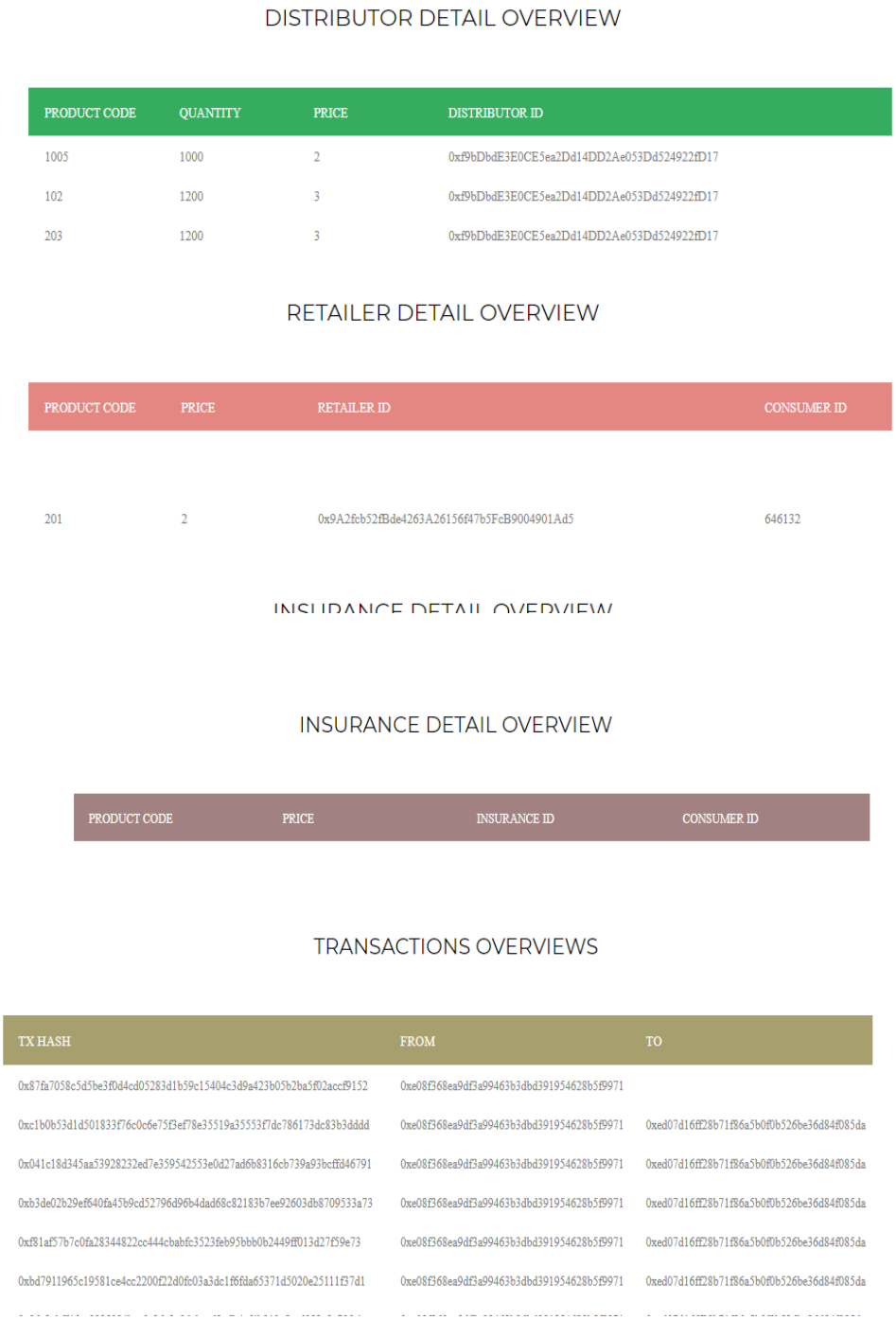


Figure 3.16: Data View and Transactions History

- **Lack of Transparency**

  Transparency structures confidence by capturing key data points, and provides open access to this data generally. Blockchain technology does not have a central authority. All transaction are published on the Ethereum blockchain, any participating can check and verify all transactions in real-time.

- **Lack of Traceability**

  Traceability improves operational supply chains through having access to all Ethereum blockchain transactions. Smart contracts are used to enforce the product tracking processes on the Ethereum blockchain. Anyone can view the provenance and journey of an product in real-time.

- **Promoted Trust**

  Through the use of the Ethereum blockchain, which provides transparency and traceability with any transaction of products, services, data and financial resources, all data is synchronized with all stakeholders in real time, which promote trust among stakeholders within the drug supply chain.

- **Transport Delays**

  Smart contracts provide the ability to evaluate the current state of the blockchain and make decisions on demand, old solutions often require human interaction, which depends on specific business hours. Blockchain presents an opportunity for smart contracts, where the life cycle and reliability of drug transit conditions can be tracked with immutable and variable data.

- **Data loss**

  Ethereum smart contracts define the data required for each transaction and ensure that all participants provide the same input. This means that each node uses the same principles, which do not change from participant to participant as they move along the supply chain.

3.9 Conclusion

In previous chapter, we have approached the challenges of drug supply chain and analyzed the needs of the a decentralized application. In this chapter, we have proposed our solution to have a maximum transparency, integrity, immutability and tranceability of data of supply chain without third-party. We implemented our solution, which is Smart Contract and a local client-side application using Web3, we deployed our smart contract in test network using Infura API, which is a platform as a service to connect

to Ropsten Network which is the exact replica of Ethereum network to better test its behavior.

After the tests taken, we have proven that our smart contract is able to manage and control the drug supply chain, it is self-executing when predetermined conditions are met. It reduce complexity in a supply chain through automated verification, execution of the multiple business transactions involved. The immutable, decentralized record also ensures that all participants have equal access to information, helps build trust and ensures complete transparency, integrity, traceability at great cost without worrying about infrastructure.

General Conclusion

The use of data collected from traditional systems cannot be relied upon as there is no guarantee that the data will not be tampered with in some way. The data is entirely under the control of a single authority that can be considered as the door lock that can be opened easily. Lack of control over data has become a serious concern in the drug supply basket because all parties involved deal with the data constantly.

Hence the need to rely on innovative new technologies to meet the needs of the supply chain like Blockchain, it has proven its effectiveness in the field of security and decentralization in different application sectors around the world it has brought many new concepts and ideas into the field of research, thus proposing a new way that can bring various benefits to create a meaningful conception:

- Decentralized network without intermediary authority, by relying on transactions and cryptography through all network nodes that have a copy of the blockchain and can connection with each other.
- Immutability of data that can be only entered, it cannot be modified or deleted by the participants in the network.
- Transparency of network where all records stored in the blockchain are available to view which increases visibility and keeps the whole system consistent and secure.

In this thesis, a review has been done on blockchain technology and we discussed the drug supply chain and its problems between the chain and the potential benefits of adopting blockchain for supply chain management. A solution has been proposed to visualize the source of products created in the system. The detailed structure is discussed and implemented.

In this work, we aimed to increase transparency and traceability by providing source products. We focus primarily on transparency as it brings multiple benefits to all entities in the supply chain. By increasing transparency and traceability, the manufacturer is confident that the products are obtained faithfully without falsification of data. In addition, other entities have this option that enables them to browse the product history since the data is not private and is available to everyone.

3.10 Future Work

Current work has primarily focused on visualizing product distribution processes and leveraging transparent tracking and timely controls in the pharmaceutical supply chain. We tried to suggest a potential solution using the Ethereum platform as a distributed network. There are multiple parts of his work that can be added or improved for future work and which we have mentioned below:

- **Re-implement the application on other platforms**, we implemented our application on the Ethereum Blockchain. There are many different platforms that can be replaced with some changes including Hyperledger and Corda. Creating the same application over other platforms allows us to compare the efficiency of these platforms.
- **Use Google Map API** to increase supply chain visibility by tracking changes in product life cycle and visualization from origin to end-destination.
- **Use of Internet of Things (IoT) devices** to improve controls and product quality traceability. By the possibility of adding functions that can use the data received directly from the product, without doubting its safety or reliability, the input of information will be automatic and more trustworthy.

Bibliography

[1] Supply chain 4.0. white paper. http://www3.weforum.org/docs/WEF_Supply_Chain_4.0_2019_Report.pdf, (Accessed 19 February, 2020). Global Practices and Lessons Learned for Latin America and the Caribbean January 2019.

[2] What is supply chain management process? https://www.predictiveanalyticstoday.com/supply-chain-management-process/, (Accessed 20 February, 2020).

[3] Cryptography in blockchain: Types & applications [2020]. https://www.upgrad.com/blog/cryptography-in-blockchain/, (Accessed August 1. 2020).

[4] React, "react a javascript library for building user interfaces.". https://reactjs.org/, (Accessed August 12, 2020).

[5] Express4.17.1 fast, unopinionated, minimalist web framework for node.js. https://expressjs.com/, (Accessed August 14, 2020).

[6] Mongodb. https://www.mongodb.com/what-is-mongodb, (Accessed August 14, 2020).

[7] Understanding public vs. private blockchain. https://selfkey.org/understanding-public-vs-private-blockchain/, (Accessed December 10, 2019).

[8] P2p network. https://www.vocal.com/video/p2p-network/, (Accessed December 23, 2019).

[9] The go programming language. https://golang.org, (Accessed February 04, 2020).

[10] How to generate a bitcoin address - step by step. https://medium.com/coinmonks/how-to-generate-a-bitcoin-address-step-by-step-9d7fcbf1ad0b, (Accessed February 04, 2020).

[11] Hyperledger, advancing business blockchain adoption through global open source collaboration. https://www.hyperledger.org, (Accessed February 04, 2020).

[12] Neo smart economy. https://neo.org, (Accessed February 04, 2020).

[13] Script. https://en.bitcoin.it/wiki/Script, (Accessed February 04, 2020).

[14] Standard c++. [online]Availableat:https://isocpp.org, (Accessed February 04, 2020).

[15] Welcome to python. https://www.python.org, (Accessed February 04, 2020).

[16] Blockchain, e-book. https://www.blockchainexpert.uk/book/blockchain-book.pdf, (Accessed February 05, 2020).

[17] Contracts — solidity 0.2.0 documentation (2016). https://solidity.readthedocs.io/en/latest/contracts.html, (Accessed February 06, 2020).

[18] Ethereum eth - what are dapps? https://support.bitkub.com/hc/en-us/articles/360030572492-What-are-DApps-, (Accessed February 06, 2020).

[19] Ethereum eth - what are the different types of "account" in ethereum? https://support.bitkub.com/hc/en-us/articles/360004415452-What-are-the-different-types-of-Account-in-Ethereum-, (Accessed February 06, 2020).

[20] Types — solidity 0.2.0 documentation (2016). https://solidity.readthedocs.io/en/latest/types.html, (Accessed February 06, 2020).

[21] What is ethereum blockchain – learn eth in 5 min. https://data-flair.training/blogs/ethereum-blockchain, (Accessed February 07, 2020).

[22] What is ethereum. guide for beginners. https://cointelegraph.com/ethereum-for-beginners/what-is-ethereum, (Accessed February 07, 2020).

[23] How does hyperledger fabric works? https://medium.com/@techgeek628/how-does-hyperledger-fabric-works-d5a4d4ff6b07, (Accessed February 10, 2020).

[24] Hyperledger , advancing business blockchain adoption through global open source collaboration. https://www.hyperledger.org/, (Accessed February 10, 2020).

[25] Hyperledger fabric — part 1 — components and architecture. https://blog.clairvoyantsoft.com/hyperledger-fabric-components-and-architecture-b874b36c4af5, (Accessed February 10, 2020).

[26] Hyperledger fabric, a blockchain platform for the enterprise. https://hyperledger-fabric.readthedocs.io/en/release-2.0/, (Accessed February 10, 2020).

[27] Introduction - hyperledger fabric docs. https://hyperledger-fabric.readthedocs.io/en/release-2.0/blockchain.html, (Accessed February 10, 2020).

[28] Blockchain vs database: Understanding the difference between the two. https://101blockchains.com/blockchain-vs-database-the-difference/, (Accessed February 13, 2020).

[29] Edi is critical to successful supply chain management. https://www.optiproerp.com/blog/why-is-edi-critical-for-successful-supply-chain-management/, (Accessed February 13, 2020).

[30] Procurement and logistics. https://www.msh.org/our-work/health-systems/pharmaceutical-management/procurement-and-logistics, (Accessed February 14, 2020).

[31] Supply chain. investopedia. https://www.investopedia.com/terms/s/supplychain.asp, (Accessed February 17, 2020).

[32] All you need to know about order management and how to speed up your procurement timeline. https://blog.procurify.com/2018/03/13/need-know-order-management-speed-procurement-timeline/, (Accessed February 20, 2020).

[33] Logistics management. https://www.techopedia.com/definition/13984/logistics-management, (Accessed February 20, 2020).

[34] Supply chain. https://corporatefinanceinstitute.com/resources/knowledge/strategy/supply-chain/, (Accessed February 20, 2020).

[35] What is supply chain management (scm)? https://erpblog.iqms.com/what-is-supply-chain-management/, (Accessed February 20, 2020).

[36] What is supply chain planning and supply chain execution? https://www.predictiveanalyticstoday.com/supply-chain-planning-supply-chain-execution/, (Accessed February 20, 2020).

[37] Essential medicines and health products. https://www.who.int/medicines/areas/access/supply/en/index5.html, (Accessed February 22, 2020).

[38] Inventory control: definition, systems, and management_2020. https://www.tradegecko.com/inventory-management/inventory-control, (Accessed February 25, 2020).

[39] Medication dispensing. https://www.mckesson.com/Resources/Medication-Dispensing/, (Accessed February 25, 2020).

[40] First oncology pharmaceutical plant to be constructed in algeria .isly-holding 2018. https://isly-holding.com/en/2018/12/23/first-oncology-pharmaceutical-plant-to-be-constructed-in-algeria/, (Accessed February 27, 2020).

[41] Health 2020. http://www.andi.dz/index.php/en/secteur-de-sante, (Accessed February 27, 2020).

[42] Supply chain management - introduction. https://www.tutorialspoint.com/supply_chain_management/supply_chain_management_introduction.htm, (Accessed January 23, 2020).

[43] Report on blockchain in trade finance and supply chain. https://www.eublockchainforum.eu/sites/default/files/report_supply_chain_v1.pdf?width=1024&height=800&iframe=true, (Accessed January 24, 2020).

[44] The new digital nation. https://e-resident.gov.ee/, (Accessed January 30, 2020).

[45] Why online voting. https://followmyvote.com, (Accessed January 30, 2020).

[46] Introduction to smart contracts and web3.js. https://hackmd.io/@cryptoknight/rJ-Nr1B47?type=view, (Accessed July 10, 2020).

[47] Blockchain technologies & tools | truffle - geniusee. https://geniusee.com/tools/truffle, (Accessed June 15, 2020).

[48] Documentation for visual studio code. https://code.visualstudio.com/docs, (Accessed June 15, 2020).

[49] Ganache. https://www.trufflesuite.com/docs/ganache/overview/, (Accessed June 15, 2020).

[50] Node.js - introduction. https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm, (Accessed June 15, 2020).

[51] Solidity. https://solidity.readthedocs.io/en/v0.7.0/, (Accessed June 15, 2020).

[52] What is truffle suite? features, how to install, how to run smart contracts. https://www.upgrad.com/blog/what-is-truffle-suite/, (Accessed June 15, 2020).

[53] Metamask. https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn, (Accessed June 16, 2020).

[54] Importation and port clearing, chapter 24. https://www.msh.org/sites/msh.org/files/mds3-ch24-importation-mar2012.pdf, (Accessed March 02, 2020).

[55] Transport management, chapter 25. https://www.msh.org/sites/msh.org/files/mds3-ch25-transportmgmt-mar2012.pdf, (Accessed March 02, 2020).

[56] Blockchain an introduction. research paper. https://beta.vu.nl/nl/Images/werkstuk-bruyn_tcm235-862258.pdf, (Accessed November 26. 2019).

[57] Blockchain cryptography – history | cryptosystem. https://data-flair.training/blogs/blockchain-cryptography/, (Accessed November 26. 2019).

[58] Hashing in blockchain explained. https://www.onlinehashcrack.com/how-to-hashing-in-blockchain-explained.php, (Accessed November 26. 2019).

[59] Anwaar Ali, Siddique Latif, Junaid Qadir, Salil Kanhere, Jatinder Singh, Jon Crowcroft, et al. Blockchain and the future of the internet: A comprehensive review. *arXiv preprint arXiv:1904.00733*, 2019.

[60] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc.", 2014.

[61] RPh Avinash Verma. Role of pharmacist in supply chain, logistics & storage of medicine. www.linkedin.com/pulse/role-pharmacist-supply-chain-logistics-storage-avinash-verma-rph/, (Accessed February 22, 2020).

[62] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30. IEEE, 2016.

[63] Rita Azzi, Rima Kilany Chamoun, and Maria Sokhn. The power of a blockchain-based supply chain. *Computers & industrial engineering*, 135:582–592, 2019.

[64] N. Mistry B121555. An introduction to bitcoin, elliptic curves and the mathematics of ecdsa. `https://raw.githubusercontent.com/bellaj/Bitcoin_Ethereum_docs/6bffb47afae6a2a70903a26d215484cf8ff03859/ecdsa_bitcoin.pdf`, (Accessed February 11, 2020).

[65] Bellaj Badr, Richard Horrocks, and Xun Brian Wu. *Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger*. Packt Publishing Ltd, 2018.

[66] Timur Badretdinov. How to create a bitcoin wallet address from a private key. `https://www.freecodecamp.org/news/how-to-create-a-bitcoin-wallet-address-from-a-private-key-eca3ddd9c05f/`, (Accessed February 11, 2020).

[67] Arshdeep Bahga and Vijay K Madisetti. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, 9(10):533–546, 2016.

[68] Imran Bashir. *Mastering blockchain*. Packt Publishing Ltd, 2017.

[69] Mike Benson. An easier way to understand the pharma industry, 2015.

[70] Steven Bragg. Inventory control — accountingtools. `https://www.accountingtools.com/articles/what-is-inventory-control.html?rq=Inventory%20Control`, (Accessed February 25, 2020).

[71] Christian Cachin et al. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, page 4, 2016.

[72] Yi-Cheng Chen, Yueh-Peng Chou, and Yung-Chen Chou. An image authentication scheme using merkle tree mechanisms. *Future Internet*, 11(7):149, 2019.

[73] Sunil Chopra and Peter Meindl. Supply chain management. strategy, planning & operation. In *Das summa summarum des management*, pages 265–275. Springer, 2007.

[74] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International conference on financial cryptography and data security*, pages 79–94. Springer, 2016.

[75] D.W. Dobler and D.N. Burt. *Purchasing and Supply Management: Text and Cases*. Management Series. McGraw-Hill, 1996.

[76] Sami Farooq and Chris O'Brien. A technology selection framework for integrating manufacturing within a supply chain. *International Journal of Production Research*, 50(11):2987–3010, 2012.

[77] MOHAMED WADIE ZERHOUNI L. ASMA EL ALAMI EL FELLOUSSE. Moving towards a north african pharmaceutical market. `http://www.ipemed.coop/fr/publications-r17/collection-construire-la-mediterranee-c49/moving-towards-a-north-african-pharmaceutical-market-a2517.html`, (Accessed February 12, 2020).

[78] Christopher Franko. Borderless : A governance platform and charity for a global society.

[79] Swati Goyal. The history of blockchain technology: Must know timeline, 2018.

[80] Kirill Grigorchuk. Blockchain technology primer. `https://digiforest.io/en/blog/blockchain-consensus-algorithms`, (Accessed November 26. 2019).

[81] Christine M Harland. Supply chain management, purchasing and supply management, logistics, vertical integration, materials management and supply chain dynamics. *Blackwell Encyclopedic Dictionary of Operations Management. UK: Blackwell*, 1996.

[82] Parikshit Hooda. Blockchain | smart contracts. `https://www.geeksforgeeks.org/smart-contracts/`, (Accessed February 05, 2020).

[83] Yining Hu, Madhusanka Liyanage, Ahsan Mansoor, Kanchana Thilakarathna, Guillaume Jourjon, and Aruna Seneviratne. Blockchain-based smart contracts-applications and challenges. *arXiv preprint arXiv:1810.04699*, 2018.

[84] CB Insights. Banking is only the beginning: 30 big industries blockchain could transform, 2017.

[85] D Kapoor, RB Vyas, and D Dadarwal. An overview on pharmaceutical supply chain: A next step towards good manufacturing practice. drug des int prop int j 1 (2)-2018. *DDIPIJ. MS. ID*, 107.

[86] IAB ThCH LAB. Overview of 9 blockchain consensus algorithms. `https://iabtechlab.com/wp-content/uploads/2018/07/Blockchain-Technology-Primer.pdf`, (Accessed November 26. 2019).

[87] M'elodie Lamarque. *The blockchain revolution: new opportunities in equity markets*. PhD thesis, Massachusetts Institute of Technology, 2016.

[88] D.M. Lambert. *Supply Chain Management: Processes, Partnerships, Performance*. Supply Chain Management Institute, 2008.

[89] Djamel Eddine Laouisset. *The Algerian Pharmaceutical Industry: Evolution & Challenges*. Berrett-Koehler Publishers, May 11, 2020.

[90] David Simchi Levi, Philip Kaminsky, and Edith Simchi Levi. *Designing and managing the supply chain: Concepts, strategies, and case studies*. McGraw-Hill, 2003.

[91] Shikha Maheshwari. Blockchain basics: Hyperledger fabric. `https://developer.ibm.com/articles/blockchain-basics-hyperledger-fabric/`, (Accessed February 10, 2020).

[92] John T Mentzer, William DeWitt, James S Keebler, Soonhong Min, Nancy W Nix, Carlo D Smith, and Zach G Zacharia. Defining supply chain management. *Journal of Business logistics*, 22(2):1–25, 2001.

[93] Solomon Michael. Ethereum fro dummies. 2019.

[94] Vince Millora. Compile and deploy using remix ide. `https://medium.com/openberry/compile-and-deploy-using-remix-ide-f58fcc662ed0`, 2019. (Accessed June 15, 2020).

[95] Waseem Akram Mir. Necessity and implementation of blockchain technology. *International Journal of Scientific Research and Management (IJSRM)*, 5(07):6279–6280, 2017.

[96] Robert M Monczka, Robert B Handfield, Larry C Giunipero, and James L Patterson. *Purchasing and supply chain management*. Cengage Learning, 2015.

[97] J Muhia, L Waithera, and R Songole. Factors affecting the procurement of pharmaceutical drugs: A case study of narok county referral hospital, kenya. *Med Clin Rev*, 3(4):20, 2017.

[98] Rahul P Naik and Nicolas T Courtois. Optimising the sha256 hashing algorithm for faster and more efficient bitcoin mining. *MSc Information Security Department of Computer Science UCL*, pages 1–65, 2013.

[99] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.

[100] Lam Pak Nian and David Lee Kuo Chuen. Introduction to bitcoin. In *Handbook of digital currency*, pages 5–30. Elsevier, 2015.

[101] M Niranjanamurthy, BN Nithya, and S Jagannatha. Analysis of blockchain technology: pros, cons and swot. *Cluster Computing*, 22(6):14743–14757, 2019.

[102] National Union of Pharmaceutical Operators. « l'organisation du marché national des médicaments : Diffi cultés et perspectives annoncées faceaux échéances de l'application de l'accord d'association avec l'union européenne et à l'entrée de l'algérie à l'omc », (Accessed February 27, 2020).

[103] Krzysztof Okupski. Bitcoin developer reference. In *Eindhoven*. 2014.

[104] Oladayo Oladipupo. Mining without rig – understanding how proof of stake (pos) works. `availableathttps://iost.watch/mining-without-rig-understanding-how-proof-of-stake-pos-works`, 2019.

[105] World Health Organization et al. Operational principles for good pharmaceutical procurement. Technical report, World Health Organization, 1999.

[106] World Health Organization et al. Annex 9: Guide to good storage practices for pharmaceuticals. *WHO technical report series*, 908, 2003.

[107] Hadja F Ouattara, Daouda Ahmat, Fr'ederic T Ou'edraogo, Tegawend'e F Bissyand'e, and Oumarou Si'e. Blockchain consensus protocols. In *International Conference on e-Infrastructure and e-Services for Developing Countries*, pages 304–314. Springer, 2017.

[108] Mohamed Yassine FERFERA Ouerdia BELLAHCENE. Les effets contrastÉs de l'intervention des laboratoires pharmaceutiques etrangers dans le secteur algerien de l'industrie pharmaceutique. `https://www.ajol.info/index.php/cread/article/viewFile/125577/115114`, (Accessed February 27, 2020).

[109] Marion PIGNEL and Denis STOKKINK. La technologie blockchain une opportunit'e pour l''economie sociale?

[110] Marc Pilkington. Blockchain technology: principles and applications. In *Research handbook on digital transformations*. Edward Elgar Publishing, 2016.

[111] Claire F'en'eron Plisson. La blockchain, un bouleversement 'economique, juridique voire soci'etal. *I2D Information, donnees documents*, 54(3):20–22, 2017.

[112] Charles C Poirier. *Advanced supply chain management: How to build a sustained competitive advantage*. Berrett-Koehler Publishers, 1999.

[113] RICH QUELCH. Origin: The pharma supply chain of the future. `https://www.supplychaindigital.com/technology/origin-pharma-supply-chain-future`, (Accessed February 22, 2020).

[114] Ameer Rosic. Smart contracts: The blockchain technology that will replace lawyers. `https://blockgeeks.com/guides/smart-contracts/`, (Accessed February 04, 2020).

[115] Matthew NO Sadiku, K Eze, and Sarhan M Musa. Smart contracts: A primer. *Journal of Scientific and Engineering Research*, 5(5), 2018.

[116] Pablo Lamela Seijas, Simon J Thompson, and Darryl McAdams. Scripting smart contracts for distributed ledger technology. *IACR Cryptology ePrint Archive*, 2016:1156, 2016.

[117] Voshmgir Shermin. Disrupting governance with blockchains and smart contracts. *Strategic Change*, 26(5):499–509, 2017.

[118] Joao Sousa, Alysson Bessani, and Marko Vukolic. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 51–58. IEEE, 2018.

[119] N. Szabo. "smart contracts: Building blocks for digital markets". `availableathttp://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html`, (Accessed March 04, 2020).

[120] DATAFLAIR TEAM. Types of blockchains – decide which one is better for your investment needs. `availableathttps://data-flair.training/blogs/types-of-blockchain`, (Accessed 23 December, 2019).

[121] DATAFLAIR TEAM. Blockchain tutorial – learn blockchain technology from scratch. `https://data-flair.training/blogs/blockchain-tutorial`, (Accessed December 23, 2019).

[122] Vinesh Thiruchelvam, Alexandre Shaka Mughisha, Maryam Shahpasand, and Mervat Bamiah. Blockchain-based technology in the coffee supply chain trade: Case of burundi coffee. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(3-2):121–125, 2018.

[123] Edvard Tijan, Saša Aksentijevi'c, Katarina Ivani'c, and Mladen Jardas. Blockchain technology implementation in logistics. *Sustainability*, 11(4):1185, 2019.

[124] Sergei Tikhomirov, Ekaterina Voskresenskaya, Ivan Ivanitskiy, Ramil Takhaviev, Evgeny Marchenko, and Yaroslav Alexandrov. Smartcheck: Static analysis of ethereum smart contracts. In *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, WETSEB '18, page 9–16, New York, NY, USA, 2018. Association for Computing Machinery.

[125] Nikolaos Petros Triantafyllidis and TNO Oskar van Deventer. *Developing an Ethereum blockchain application.* PhD thesis, Ph. D. Thesis, University of Amsterdam, Amsterdam, The Netherlands, 2016.

[126] JGAJ Van der Vorst. Supply chain management: theory and practices. In *Bridging Theory and Practice*, pages 105–128. Reed Business, 2004.

[127] Elizabeth A Williamson, David K Harrison, and Mike Jordan. Information systems development within supply chain management. *International Journal of Information Management*, 24(5):375–385, 2004.

[128] Maximilian Wohrer and Uwe Zdun. Smart contracts: security patterns in the ethereum ecosystem and solidity. In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 2–8. IEEE, 2018.

[129] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[130] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain technology overview. *arXiv preprint arXiv:1906.11078*, 2019.

[131] Nazila Yousefi and Ahmad Alibabaei. Information flow in the pharmaceutical supply chain. *Iranian journal of pharmaceutical research: IJPR*, 14(4):1299, 2015.

[132] Zouied Zohra. Moving from «partnership for manufacturing» to «partnership for innovation» in algerian pharmaceutical industry: case of saidal group. `http://ipco-co.com/IJBES/Papers/1.pdf`, (Accessed February 27, 2020).