REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : RTIC06/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Réseaux et Technologies de l'Information et de la Communication (RTIC)**

Titre

Blockchain pour suivi et sécurisation des produits de santé

Par :

AZIZI MIYADA

Soutenu le ../../…. devant le jury composé de :

Nom Prénom	grade	Président
Aloui Ahmed	**MCB**	**Rapporteur**
Nom Prénom	grade	Examinateur

Année universitaire 2020-2021

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 diffcultés 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 conséquence, les données sont susceptibles d'être modiffiées et il n'y a aucune garantie que l'administration du système ne modifie pas les données pour obtenir le résultat souhaité. Dans cette mémoire, 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 dés mentionnés ci-dessus. La capacité des systèmes blockchain à identier l'origine des données les rend particulièrement adaptés aux applications de la chaîne d'approvisionnement pharmaceutique. Le système a un Ethéreum 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.

Remerciements

En premier lieu, je remercie le bon dieu de m'avoir donné la force et la patience nécessaire pour achever ce travail de mémoire.

Je tiens à remercier ALOUI Ahmed Docteur à l'université de Biskra, d'avoir Co-encadré mon travail, de son suivi et ses conseils. Ainsi que pour sa disponibilité et son soutien.

Je remercie les membres du jury de m'avoir fait l'honneur d'accepter de participer à mon jury de mémoire.

Je tiens aussi à saluer toute ma promotion de Mastre et tous mes amis.

Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail de recherche.

TABLE DE MATRICE

TABLE DE FIGURE

TABLE

Introduction génerale

La Blockchain est une nouvelle technologie qui a été mise en avant ces dernières années grâce à la popularité du Bitcoin. Le bitcoin n'est pourtant qu'une des utilisations de cette technologie. En effet, il existe beaucoup d'autres usages possibles et les entreprises qui travaillent sur son développement sont de plus en plus nombreuses. L'objet de cette mémoire sera d'étudier les mariages possibles entre la blockchain et le secteur pharmaceutique.

La technologie blockchain est une nouvelle façon de concevoir le stockage d'information en abolissant la nécessité d'un tiers de confiance. En alliant plusieurs technologies, la blockchain permet à plusieurs entités non seulement de partager des données mais aussi de les modifier, et tout simplement d'y accéder de manière collaborative et surtout sécurisée. En réussissant le tour de force d'ouvrir et partager des données tout en les sécurisants, la technologie blockchain permet de créer de la confiance entre les différents utilisateurs de cette donnée. [9]

Quel potentiel pour la blockchain dans le monde pharmaceutique ? De par son universalité, la technologie blockchain va très probablement impacter tous les pharmaciens, quel que soit leur domaine : à l'officine, à l'hôpital, en industrie, sans oublier la recherche, la blockchain a le potentiel de pouvoir faciliter ou améliorer de nombreux services et de réaliser les interconnections entre ces différents domaines.[9]

L'industrie pharmaceutique est la partie du secteur de la santé qui s'occupe des médicaments. L'industrie comprend divers sous-domaines liés au développement de médicaments, à la production et commercialisation. L'objectif principal de l'industrie pharmaceutique est de fournir des médicaments qui préviennent les infections, maintiennent la santé et traitent les maladies.

La chaîne d'approvisionnement de l'industrie pharmaceutique est comme une chaîne d'approvisionnement pour toute autre industrie du secteur manufacturier. Les activités de la chaîne d'approvisionnement pharmaceutique impliquent le flux et la transformation des médicaments des matières premières jusqu'aux les utilisateurs finaux. De plus, les informations associées proviennent des relations dans la chaîne d'approvisionnement pour obtenir un avantage concurrentiel durable. La gestion de la chaîne d'approvisionnement Pharmaceutique est plus difficile que les applications typiques au sein d'entreprises de

l'industrie, car les médicaments et les fournitures chirurgicales doivent être disponibles à tout moment. [27]

La chaîne d'approvisionnement représente un réseau de relations au sein de l'entreprise et organisations commerciales composées de fournisseurs de matières premières, de fabricants, d'expéditeurs, la logistique de tiers, les détaillants et les parties liées impliquées, ce qui facilite l'inverse production de matériaux, de services et d'informations depuis le produit original jusqu'au produit final client, tout en augmentant l'efficacité et en obtenant la satisfaction du client. Tel quel difficile d'obtenir une image complète de toutes les opérations qui se déroulent dans le réseau et intégrer toutes les parties concernées. Ce que ce système subit actuellement, c'est une manque d'efficacité et de transparence et erreurs de suivi résultant d'une mauvaise qualité ou la présence d'un problème spécifique dans certaines parties du produit, le temps, le coût et le négatif effets qu'il a sur la relation des entreprises de la chaîne d'approvisionnement.

Cependant, le modèle actuel rend difficile le maintien d'un système de chaîne d'approvisionnement cohérent et efficace. Aujourd'hui, il faut un système efficace et fiable pour effectuer, enregistrer et sauvegarder des transactions, pour créer un état d'amélioration et un changement considérable dans la façon de produire, de commercialiser, d'acheter et de consommer des produits et développer l'industrie et accroître son efficacité.

Problimatique

Chaque jour, des milliards de produits sont fabriqués et livrés partout dans le monde. Pendant ce temps, les matières premières des produits ne sont pas produites dans une seule entreprise, et généralement, les composants proviennent de fabricants différents. La chaîne d'approvisionnement de certaines industries pharmaceutiques dans de nombreux pays, en particulier l'Algérie, a confronté à de nombreuses difficultés dans le suivi des produits et des matières premières, ce qui affecte la croissance de l'industrie, sa réputation et ses affairs financiers, et cela peut être dû à des informations inexactes et pas toujours disponibles, ou à l'incapacité de travailler de manière transparente et suivre les erreurs causées par une mauvaise qualité ou la présence d'un problème spécifique dans les étapes de distribution, ainsi que le manque de technologie fiable par laquelle l'information peuvent être combinés de manière sûre et rapide. Dans la chaîne d'approvisionnement, de nombreuses parties, y compris les distributeurs, les pharmaciens et les patients, sont impliquées, à travers un réseau, en complétant la chaîne. Toutes les parties du réseau sont contrôlées par un système. Cependant,

certaines chaînes d'approvisionnement en médicaments peuvent être préoccupées par la source des matériaux et des produits, et la nécessité de remplacer les architectures client-serveur traditionnelles a été soulevée pour les raisons suivantes :

- La documentation des enregistrements entraîne des erreurs, des retards et des conflits d'informations pour traquer les retards d'expédition, le vol ou certains problèmes inattendus qui peuvent survenir à un certain point dans la chaîne d'approvisionnement.
- Un seul serveur central accepte toutes les demandes entrantes, ce qui entraîne de faibles performances et des coûts élevés de maintenance du serveur.
- Le système est faible et facilement compromis parce que l'autorité centrale est le seul point sujet à l'échec, à l'attaque et à la pénétration.
- Un problème critique avec les réseaux traditionnels est la transparence et la traçabilit des sources des produits car les utilisateurs ne possèdent ni ne contrôlent aucune donnée.
- Les informations présentées par les autorités centrales ne sont pas fiables car il n'y a aucun moyen de confirmer que les données n'ont pas été falsifiées. [26]

Solution

Il existe différentes études qui ont tenté de surmonter ces problèmes et d'augmenter transparence et visibilité de la chaîne d'approvisionnement, en plus d'éliminer le besoin d'une autorité centrale grâce à l'utilisation de nouvelles technologies comme Blockchain, qui peuvent rendre un grand changement dans les opérations de la chaîne d'approvisionnement plus efficaces, transparentes et sécurisées, et car il s'agit d'une technologie de grand livre distribué qui garantit la confiance, la transparence et Sécurité.

La technologie Blockchain a été introduite en 2008 sous le pseudonyme de Satoshi Nakamoto pour résoudre le problème de la double dépense et de la confiance sans dépendance envers des tiers "via Bitcoin", la première application à prendre en charge des transactions simples. Son utilisation dans les crypto-monnaies ne s'est pas arrêtée là, mais peut être utilisée pour plus que cela, dans son essence, il représente un enregistrement général de tout type de transaction qui peut être distribué à tous parties du réseau au lieu d'une seule personne contrôlant tout. Il a été utilisé largement dans divers domaines. Plus précisément, dans la chaîne d'approvisionnement, la visualisation garantit que le produit l'intégrité et aide

les participants à avoir une meilleure vision du cycle de vie des produits. Ça aussi permet aux participants de découvrir des relations.

Dans cette mémoire, nous avons tenté d'adopter la technologie blockchain comme un réseau distribué pour fournir aux participants le prix record, la date de distribution et de livraison, les quantités distribuées et d'autres informations pertinentes pour gérer plus efficacement la chaîne d'approvisionnement. Nous avons suggéré un système simple qui contient trois types d'entités principaux ;

1-les utilisateurs, 2-le contrat intelligent et 3-le réseau Blockchain. Nous avons essayé d'augmenter visibilité, traçabilité de la chaîne d'approvisionnement et réduction des pertes. Le smart contrat basée sur la blockchain , qui sont des logiciels automatisés, sont utilisés dans le système proposé pour être exécutés sur chaque nœud du réseau. Par conséquent, la technologie blockchain est utilisée pour donner organisations la capacité d'échanger des données distribuées et le transport sans aucune autorité centralisée. Par conséquent, les parties impliquées dans la chaîne d'approvisionnement peuvent avoir interactions directes les uns avec les autres, et la confiance n'est plus requise. Ces sécuriser les communications directes conduisent à une plus grande transparence, clarté et efficacité tout en réduire le coût et le risque d'échec dans le processus de suivi des expéditions.

Organisation de la mémoire

Cette mémoire est organisée comme suit :

- **Introduction** : Nous commencerons notre mémoire par une introduction au contexte de ce travail, le problème ciblé et la solution que nous proposons.
- **Chapitre 01** : **Technologie Blockchain**

Dans ce chapitre, nous allons d'abord introduire les définitions, les caractéristiques, les principaux composants et types de technologie blockchain. Ensuite, nous indiquerons comment le blockchain fonctionnant avec ses avantages et ses inconvénients. Nous présenterons également la plupart des défis et des limites auxquels la technologie blockchain est confrontée, et ses différents usages dans nos vies. Enfin, nous présenterons la plupart des implémentations de blockchain.

- **Chapitre 02 : Chaîne d'approvisionnement en médicaments**

Dans ce chapitre, nous présenterons la définition du gestion de la chaine d'approvisionnement et ses processus. De plus, pour présenter l'importance et les avantages de

la chaîne d'approvisionnement. Ensuite, nous présenterons la chaîne d'approvisionnement et les processus de distribution des médicaments et comment son travail en Algérie. Enfin, nous vous proposerons les principaux défis et problèmes auxquels est confrontée la chaîne d'approvisionnement du médicament en Algérie.

- **Chapitre 03 : Conception et mise en œuvre**

Dans ce chapitre, nous présenterons le système proposé, modèle de composants qui définir la chaîne d'approvisionnement en médicaments et la mise en œuvre de ce système.

- **Conclusion :** Nous terminerons notre mémoire par une conclusion générale et quelques perspectives et orientations futures.

1 Chapire I : TECKNOLOGIE DE BLOCKCHAIN

1.1 Introduction

Le 31 octobre 2008, un inconnu utilisant le pseudonyme « Satoshi Nakamoto » a écrit dans une liste de diffusion d'e-mails réservée aux cypherpunks (un mouvement de personnes utilisant la cryptographie pour protéger la vie privée). "Je travaille sur un nouveau système de monnaie électronique entièrement de pair-à-pair, sans tiers de confiance". Ce texte est accompagné d'un lien qui amène vers Bitcoin.org et sur lequel est hébergé le livre blanc du Bitcoin [3], rédigé dans un anglais impeccable, résumant le fonctionnement du nouveau protocole. Le premier concept de Blockchain a été appliqué le 03 Janvier 2009 dans le cadre de Bitcoin.

La technologie à la base de Bitcoin et d'autres crypto-monnaies, est une base de données de grand livre distribuée pour l'enregistrement des transactions, permettant ainsi aux utilisateurs de partager leur grand livre de transactions.

Dans ce chapitre, On présente et on explique la technologie Blockchain avec ses fonctionnalités les plus importantes et ses concepts associés.

1.2 Historique

L'histoire de la blockchain peut être scindée en deux grandes périodes séparées l'une de l'autre par la création du Bitcoin en 2009 dont elle est indissociable.

Les années 1970 à 2009 marquent l'émergence non seulement des différentes technologies qui permettront plus tard de créer la blockchain Bitcoin, mais aussi, la volonté de certains acteurs de commencer à assembler ces technologies entre elles afin de créer des systèmes ressemblant de plus en plus à un modèle blockchain moderne.

La naissance du Bitcoin marque un tournant pour la blockchain. En effet, c'est le premier projet à mettre en œuvre à grande échelle cette technologie. Le 3 janvier 2009 Satoshi Nakamoto crée le premier bloc de la blockchain Bitcoin, donnant ainsi naissance à la première monnaie digitale supportée par la technologie blockchain. [8] La preuve de concept

Bitcoin va par la suite inspirer foule de programmeurs et entrepreneurs qui vont élargir l'utilisation de la blockchain à d'autres fins que les cryptomonnaies. Aujourd'hui, les premiers projets dans le monde de la santé commencent à voir le jour. [9]

A. Le Bitcoin, première application exploitant la blockchain

Le 19 août 2008, le mystérieux Satoshi Nakamoto réserve le nom de domaine bitcoin.org. On ne sait pratiquement rien de Satoshi Nakamoto. Nul ne sait avec certitude si derrière ce pseudonyme se cache une personne physique unique ou bien un groupe de travail. Le 31 octobre de la même année Satoshi Nakamoto publie un message sur une liste de diffusion de cryptographie (medium qui rappelle celui utilisé par les Cypherpunks). [9]

"J'ai travaillé sur un nouveau système de paiement électronique qui est entièrement pair à pair, sans tiers de confiance.

Les principales propriétés :

- Les doubles dépenses sont évitées grâce à un réseau pair à pair.
- Pas de tiers éditant la monnaie ou autre personne de confiance.
- Les participants peuvent être anonymes.
- Les nouvelles pièces sont fabriquées à partir de preuves de travail de type Hashcash.
- La preuve de travail pour la génération de nouvelles pièces alimente aussi le réseau pour éviter les doubles dépenses."

Le document proposé dans ce message, livre blanc du bitcoin, expose très précisément les objectifs de la création de cette monnaie électronique mais surtout l'ensemble des moyens techniques pour y parvenir. Pour la première fois, une monnaie électronique complètement décentralisée sans tiers de confiance ayant un niveau de sécurité suffisant voit le jour. Le Bitcoin est considéré comme le père des cryptomonnaies. Bitcoin étant un projet open source d'innombrables autres cryptomonnaies verront le jour par la suite, chacune ayant une propriété technique et/ou un objectif différent. Au 19 août 2018, il existait plus de 1600 cryptomonnaies. Malgré leur nombre, toutes ont une propriété en commun: leur fonctionnement est sous-tendu par une technologie que l'on appelle désormais blockchain. [9]

Le 3 janvier 2009, un premier bloc est créé. Neuf jours plus tard, une première transaction de 10 Bitcoins a lieu. Le 5 octobre 2009, une première estimation de la valeur du Bitcoin est faite à partir de son coût de production (0,00071€), que l'on peut définir comme le coût en électricité nécessaire à entretenir le réseau. Un peu plus d'un an plus tard, en novembre

2010, le Bitcoin vaut 40 centimes d'euros. Le 12 décembre 2010, Satoshi Nakamoto annonce qu'il quitte le projet Bitcoin. À la fin de l'année 2013, la valeur du Bitcoin dépasse 800 euros, et au cours de cette année, la cryptomonnaie a largement franchi la frontière des cercles d'initiés auxquels elle s'était limitée jusque-là. Des premiers organismes acceptent de se faire payer en Bitcoins, les autorités: états et banques centrales commencent à se saisir du sujet dans de nombreux pays. L'Allemagne, par exemple, donne le statut de monnaie privée au Bitcoin. [9]

L'année 2017 marquera un engouement massif du grand public pour le Bitcoin qui atteindra un seuil historiquement haut avec une valeur de 16323€ et une capitalisation boursière totale de plus de 281 milliards d'euros. Depuis, la valeur du Bitcoin chute de manière continue. Il est aux alentours des 5600 euros en septembre 2018. [9]

Satoshi Nakamoto a su tirer profit d'un gain en maturité des technologies (cryptographie, preuve de travail, …) constituantes du Bitcoin. Il a aussi bénéficié des innovations incrémentales des différents projets qu'ont proposés ses prédécesseurs (E-cash, B-Money, …). En voulant mettre à disposition une monnaie électronique et décentralisée, le Bitcoin, il a mis au point une technologie, la Blockchain, qui marque un avant et un après dans la quête de la mise au point d'une technologie qui permet de créer de la confiance entre des entités ne se connaissant pas. Avec le Bitcoin, des personnes du monde entier s'échangent des milliards d'euros chaque jour, le tout sans qu'aucun organisme central ne régule cette monnaie. Pour la première fois, la confiance nécessaire à deux partis pour échanger de la valeur ne vient pas du fait qu'un organisme tiers, comme une banque, valide l'échange. La confiance entre les deux partis est directement issue du système utilisé: la blockchain Bitcoin.

Il y a donc un lien étroit entre la première cryptomonnaie, le Bitcoin, et la technologie permettant son fonctionnement, la blockchain, toute deux émanant du même objectif et du même inventeur. Cependant, il est crucial de distinguer les deux. Le Bitcoin est donc la première preuve de concept et cas d'usage de la technologie blockchain. Il ouvre la voie à une diversification dans l'utilisation et la montée en maturité de cette technologie après avoir mise cette dernière sous le feu des projecteurs. [9]

B. D'autres cryptomonnaies

Gardons en tête que la blockchain Bitcoin a été codée en open source autorisant de fait son appropriation par un tiers afin de la modifier pour en éditer une version modifiée. Ainsi, des centaines et des centaines de cryptomonnaies ont vu le jour depuis le Bitcoin, chacune implémentant une caractéristique particulière. Nous en décrirons uniquement quelques-unes sur un total supérieur à 1600. La succession de ces cryptomonnaies nommées altcoins (dans le sens "alternative coins") en opposition à la monnaie mère Bitcoin proposent des améliorations que l'on peut qualifier d'incrémentales et rarement de rupture.

1. Litecoin

Litecoin est une des premières cryptomonnaies à émerger après le Bitcoin, en octobre 2011. Sa capitalisation boursière est de 3 milliards de dollars mi 2018. La principale innovation de Litecoin par rapport au Bitcoin est de confirmer les transactions dans un délai oscillant autour de deux minutes et demie quand Bitcoin les confirme en une dizaine de minutes. Litecoin espère gagner la faveur des utilisateurs en proposant un service plus réactif que Bitcoin. [10]

2. Peercoin

Le 19 août 2012, est publié le livre Blanc du Peercoin qui propose un nouveau concept innovant: "la preuve d'enjeu". Peercoin introduit un nouveau mode de consensus entre les nœuds du réseau extrêmement moins énergivore que celui du Bitcoin. Ce mode de consensus est appelé "preuve d'enjeu" (souvent mentionné en anglais "proof of stake"). Nous pouvons considérer l'invention de ce mode de consensus comme une innovation majeure de rupture pour la blockchain. Pour être exact, Peercoin utilise un modèle de consensus hybride utilisant cette nouvelle preuve d'enjeu et la preuve de travail, modèle historique utilisé par le Bitcoin. [11]

3. Monero ZCash

Une caractéristique de la blockchain est de publier chaque transaction de manière ouverte. Ainsi, n'importe qui peut (via le site www.blockchain.info, par exemple) voir chacune des transactions réalisées sur la blockchain. Monero et ZCash, respectivement créés en 2014 et 2016, abolissent ce système de livre ouvert en ne permettant plus aux tiers d'accéder à la liste des transactions. [9]

4. BitcoinCash

En 2017, la blockchain Bitcoin est soumise à un nombre accru de transactions, or cette dernière ne peut pas créer des blocs de transactions supérieurs à 1Mo. En résulte des transactions en attente de validation et une flambée des frais de transaction. BitcoinCash, directement dérivé de Bitcoin en août 2017, passe la taille des transactions de 1 à 8Mo, apportant ainsi une réponse à ce problème de scalabilité rencontré par Bitcoin. [9]

1.3 Définition de Blockchain

Blockchain désigne une chaine de blocs sur lesquels sont stockées des informations de toute nature. La blockchain est définie généralement comme une « technologie de stockage et de transmission d'informations, transparente, sécurisée, et fonctionnant sans organe central de contrôle ». La blockchain est une technologie des registres distribués, ou DLT (Distributed Ledger Technology), qui regroupe les systèmes numériques qui enregistrent des transactions d'actifs et leurs détails dans plusieurs emplacements à la fois. La blockchain est la technologie DLT la plus connue. [3]

La technologie Blockchain se caractérise principalement de six éléments majeurs: décentralisé, transparente, sécurisé et immuable, autonome, open source et anonyme. Comme décrit cidessus. [3]

Elle est décentralisé: La blockchain contient Un système de bases de données décentralisé avec un contrôle en libre accès pour tous ceux qui sont connectés au réseau. Les données peuvent être consultées, surveillées, stockées et mises à jour sur plusieurs systèmes.

Ces données ne sont pas toutes regroupées dans le serveur d'un intermédiaire central, mais au contraire « distribuées », c'est-à-dire hébergées chez chaque participant ; il n'y a donc pas d'autorité unique pouvant approuver les transactions ou définir des règles spécifiques pour que les transactions soient acceptées. Cela signifie que la confiance est énorme, car tous les participants du réseau doivent parvenir à un consensus pour accepter les transactions.

Elle est transparente: C'est l'avantage le plus important. Tous les participants peuvent voir les blocs et les transactions qui y sont stockés dedans. Les données enregistrées et stockées dans la blockchain sont transparentes pour les utilisateurs potentiels et peuvent être mises à jour facilement. Cela ne signifie toutefois pas que tout le monde peut voir le contenu réel des transactions, qui sont protégés par une clé privée.

Comme dans le réseau Bitcoin, toutes les transactions sont publiques et vérifiables par tous en effectuant un mécanisme consensus, ce qui va permettre à chacun de s'assurer que chaque participant possède bien les Bitcoins qu'il dépense et qu'il ne les dépense qu'une seule fois. La nature transparente des blockchains pourrait certainement empêcher la modification ou le vol de ces données. [3]

le consensus : la blockchain correspond à un historique de transactions sur lequel tout le monde s'accorde, ce consensus sur le séquencement des transactions permet de résoudre le problème dit de la "double dépense" : un Bitcoin dépensé dans une transaction ne peut pas être dépensé une deuxième fois dans une transaction qui serait diffusée ultérieurement sur le réseau. La deuxième transaction serait rejetée par le réseau. [3]

Elle est sécurisé: La base de données peut uniquement être étendue et les enregistrements précédents ne peuvent pas être modifiés (au moins, le coût est très élevé si quelqu'un souhaite modifier les enregistrements précédents).

Ces enregistrements sont dits Immuables, une fois stockés, deviennent réservés pour toujours et ne peuvent pas être modifiés facilement sans le contrôle simultané de plus de 51% des nœuds du réseau.

Le système cryptographique de validation garantit qu'il est quasiment impossible de réécrire une transaction une fois son bloc validé (personne n'a réussi à le faire depuis la création du Bitcoin). [3]

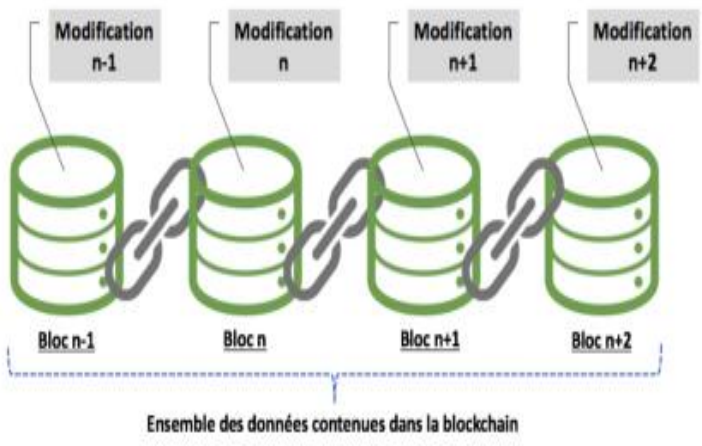


Figure 1.1 : mode de stockage des données dans une blockchain [9]

Autonome : Le système blockchain est indépendant et autonome, ce qui signifie que chaque nœud du système blockchain peut accéder aux données, les transférer, les stocker et les mettre à jour en toute sécurité, ce qui les rend fiables et exemptes de toute intervention externe. [3]

Open source : La technologie de la blockchain est formulée de manière à fournir un accès open source à toutes les personnes connectées au réseau. Cette polyvalence inimitable permet à quiconque non seulement de vérifier publiquement les enregistrements, mais également de développer diverses applications imminentes. [3]

Anonyme : Lorsque le transfert de données à lieu entre nœuds, l'identité de l'individu reste anonyme, ce qui en fait un système plus sécurisé et fiable.

Une personne faisant partie de ce réseau doit vérifier chaque nouvelle transaction effectuée. Une transaction de recherche dans un bloc d'une blockchain est vérifiée par tous les nœuds du réseau, elle devient de plus en plus immuable. [3]

1.4 Fonctionnalités de la blockchain

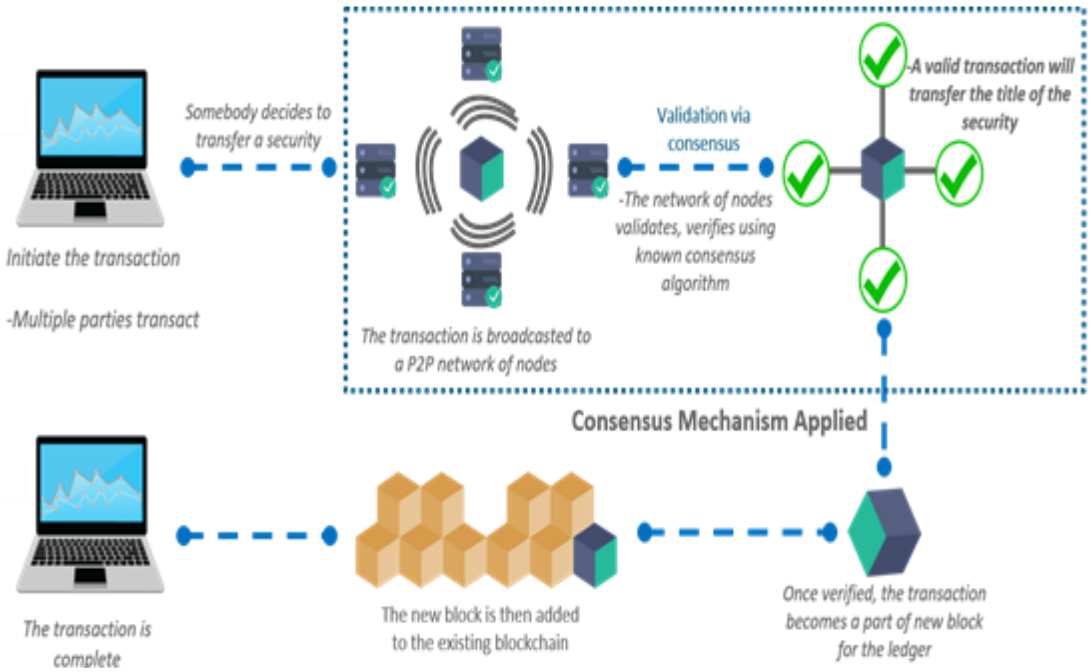


Figure 1.2 : Les étapes sur un réseau Blockchain [3]

Source : euderka.co/blockchain

La figure 1.2 illustre le mécanisme de fonctionnement des transactions dans le réseau Blockchain.

Les étapes de ce mécanisme sont les suivantes :

- Quelqu'un demande une transaction.
- La transaction est diffusée sur un réseau P2P public (réseau Blockchain) composé de plusieurs nœuds.
- Le réseau de nœuds valide la transaction en utilisant les algorithmes de hachage.
- Une fois vérifiée, la transaction est combinée avec d'autres transactions pour créer un nouveau bloc de données pour le grand livre.
- Le nouveau bloc est ajouté à la chaîne de blocs existante, sous une forme qui est permanente et inaltérable.
- Enfin la transaction sera effectuée avec succès.

Pour comprendre vraiment comment fonctionne une blockchain, le plus simple est de suivre l'itinéraire d'une transaction (même si l'immense majorité de ceux qui utiliseront un service sur la blockchain passeront par des intermédiaires qui leur simplifieront la tâche et se chargeront de l'enregistrement de l'opération sur le réseau.

Prenons le cas d'une opération réalisée sur la plus ancienne et la plus commune des blockchains, Bitcoin, qui a servi de modèle à l'ensemble de celles qui ont suivi.

Oussama souhaite envoyer 1 Bitcoin à ***Ilham*** sur le réseau Bitcoin, chacun possède une adresse publique, appelée « clé publique » (l'équivalent d'un RIB) : une suite d'environ 34 caractères contenant des chiffres et des lettres en minuscules ou majuscules (*par exemple : 13o7TCoNWbaqYp9g89w1gHrZ7GvvKftRsd, pour l'un et 16Xgsii16x4icN7yjQgXX648Lf4LxRsd19 pour l'autre).*

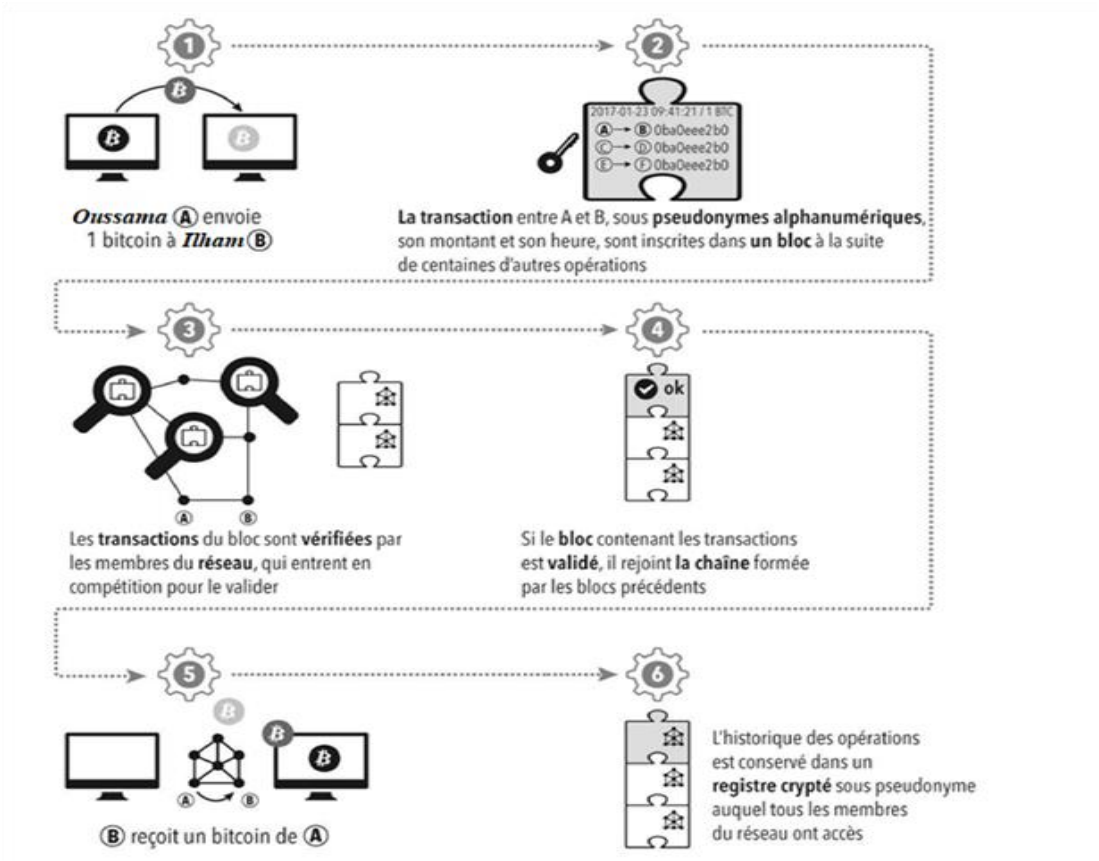


Figure 1.3 : Exemple de fonctionnement Blockchain (Bitcoin) [3]

1. Oussama signe la transaction avec sa clé privée : une autre suite de chiffres et de lettres, cette fois-ci confidentielle (c'est l'équivalent numérique de la clé de votre coffre-fort, que vous n'avez donc intérêt ni à donner ni à perdre), qui autorise le versement de l'argent (points 1 et 2 correspondants à l'étape 1 du schéma de la figure 1.3.
2. La transaction est alors écrite dans la blockchain. Elle est entrée, à la suite d'autres transactions, dans ce qui est appelé un bloc (une grappe de plusieurs centaines, voire milliers, de transactions).

   Pour chaque transaction, différentes informations apparaissent et seront donc consultables par tous les membres du réseau :

   – les clés publiques de ***Oussama*** et de ***Ilham*** : les transactions ne sont donc pas anonymes, mais réalisées sous les pseudonymes que constituent ces clés publiques ;

   – le nombre de Bitcoins transférés de Oussama à Ilham : 1 ;

– l'heure précise, à la seconde près, et la date à laquelle l'opération a eu lieu (2019-07-12, 09 : 41 : 21).

3. Chaque bloc, en plus de toutes ces transactions, contient un résumé cryptographique du bloc précédent : un nombre, qui correspond à l'unique résultat possible que l'on obtient quand on entre les informations du bloc précédent dans une fonction appelée SHA256 (appelé le hash, soit le résultat d'un algorithme de hachage). Cela permet de noter dans chaque bloc l'ADN cryptographique du bloc précédent. Ainsi, si un pirate veut modifier a posteriori les informations d'un bloc, il devra corriger tous les blocs qui suivent, et ce sur chaque exemplaire de la blockchain hébergé par chaque ordinateur du réseau (ce qui est quasiment impossible). Par ce système, les blocs sont donc enchaînés les uns aux autres (d'où le nom « blockchain » – points 3 et 4 correspondants à l'étape 2 du schéma de la figure 1.3

4. Les membres du réseau qui le souhaitent vont alors entrer en compétition pour valider le bloc. On les appelle les « mineurs ». Ils y sont incités par une récompense : celui qui aura le privilège d'effectuer cette validation gagnera automatiquement 12,5 nouveaux Bitcoins, qui seront générés pour l'occasion par le système. Pour gagner ce droit, tous les participants vont vérifier la véracité des informations contenues dans toutes les transactions du bloc : est-ce que ***Oussama*** possède bien 1 Bitcoin au vu de toutes les transactions qu'il a déjà réalisées ? N'utilise-t-il bien ce Bitcoin qu'une seule fois ? Est-ce que les autres transactions sont également justes ? Mais ils vont aussi devoir résoudre un problème mathématique complexe, en utilisant les capacités de calcul de leur ordinateur (c'est le système de « **preuve de travail** », ou ***proof of work***). Chaque ordinateur va tester des solutions jusqu'à trouver la bonne. Le premier à réussir pourra valider le bloc et gagner la récompense. Le système a été conçu pour qu'une solution soit trouvée au bout de 10 minutes en moyenne (ce qui impose un léger délai à la sécurisation de la transaction – point correspondant à l'étape 3 du schéma de la figure 1.3

5. Le bloc est alors validé. Il est ajouté à la blockchain qui est mise à jour sur les ordinateurs de chaque participant (point correspondant à l'étape 4 du schéma de la figure 1.3. Les participants n'acceptent ce nouveau bloc que si les transactions qu'il contient sont valides (si l'argent de chaque transactionneur n'y est bien dépensé qu'une seule fois). Si le bloc est valide, ils expriment leur accord en travaillant sur le bloc suivant selon le même processus, et y inscrivant le hash du bloc validé. C'est pourquoi on parle de système de consensus.

6. *Ilham* a désormais reçu le Bitcoin envoyé par *Oussama*. En récompense du travail fourni, le mineur reçoit une certaine quantité de cryptomonnaie créée pour l'occasion (12,5 Bitcoins à ce jour), mais ce rendement décroît au fil du temps – points 7 et 8 correspondants aux étapes 5 et 6 du schéma de la figure 1.3 [3]

1.5 Structure d'une Blockchain

1.5.1 Transaction

Sont stockées dans les fichiers appelés blocs. Ils sont cryptés et sont généralement liés à des transactions précédentes, formant ainsi une chaîne. Un propriétaire d'une valeur (devise numérique dans BC Bitcoin) signe numériquement la transaction précédente avec sa clé publique et crée un hachage. Le propriétaire de la transaction précédente signe alors le hachage avec sa clé privée.

La figure 1.4 illustre une version simplifiée de la chaîne de propriété. Dans des cas plus complexes, le nombre d'entrées et des sorties peuvent être multiples. Une transaction contient un certain nombre de champs, comme indiqué dans le tableau 1.1 [3]

Size	Field	Description
4 bytes	Version	Specifies which rules this transaction follows
1-9 bytes (VarInt)	Input Counter	How many inputs are included
Variable	Inputs	One or more transaction inputs
1-9 bytes (VarInt)	Output Counter	How many outputs are included
Variable	Outputs	One or more transaction outputs
4 bytes	Locktime	A Unix timestamp or block number

Tableau 1.1 : Champs dans une transaction

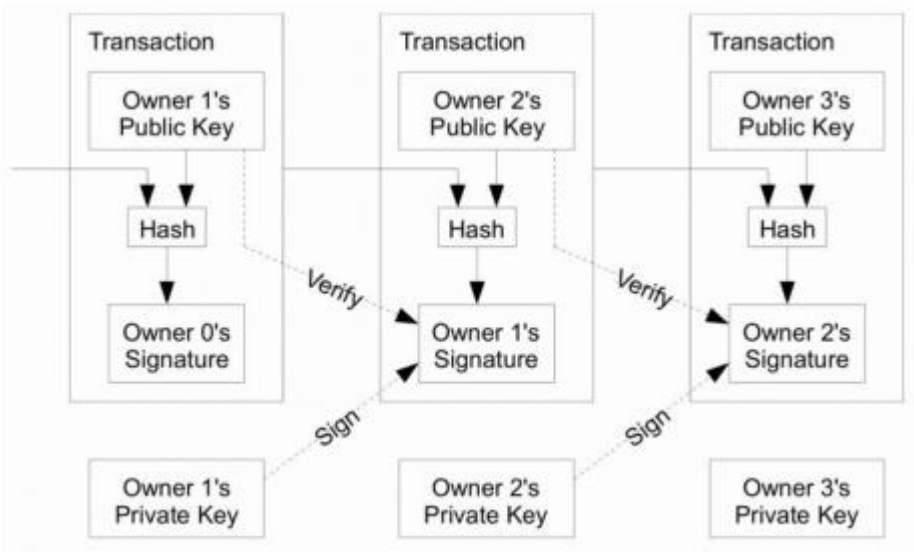


Figure 1.4 : Chaîne de propriété de transaction [3]

1.5.2 Les blocs

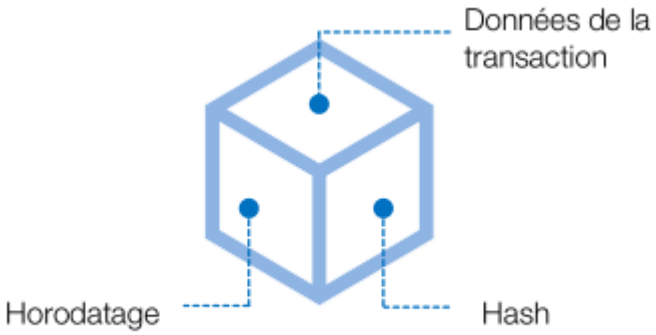


Figure 1.5 : Contenu d'un bloc de la blockchain

Un bloc constitue un ensemble de données, contenant toutes les informations d'une ou de plusieurs transactions entre deux personnes, à un moment donné. Il contient entre autres :

- **Transaction**

Chaque transaction a recours à la cryptographie : l'origine et la confidentialité des données est assurée. Cela fonctionne par un système de pair de clés publique et privée, l'une privée servant au chiffrage, l'autre publique servant à l'identification de l'acteur. Ce système permet à la fois de signer une transaction et donc de chiffrer son contenu.

• **Horodatage**

Les blocs, constitués de plusieurs transactions signées par clés publiques, sont ensuite horodatés par leur auteur. L'horodatage est essentiel car il permet la datation relative des blocs, permettant la classification chronologique de ces derniers, et donc la traçabilité des différentes transactions ayant eu lieu au fur et à mesure du temps.

• **Hash** (du bloc précédent, et du bloc actuel)

Le « hash » est obtenu par une fonction dite de « hachage ». Cette fonction permet de condenser toutes les informations d'un bloc en une suite de 64 caractères. Le hash résultant de cette condensation d'informations peut être comparé à une empreinte digitale, identifiant de manière précise et unique un bloc.

Il n'y a aucune ressemblance avec le précédent hash, cette caractéristique rend toute modification du contenu d'un bloc immédiatement visible, puisque le hash d'un bloc est similaire à son empreinte digitale.

Un bloc validé est un bloc reprenant correctement le hash du bloc précédent. Si ce précédent hash est contenu dans le bloc actuel, cela signifie que les données présentent dans ce bloc sont en accord avec celles du bloc précédent, qu'il n'y a donc pas eu de modification des données précédentes.

Si une entité malicieuse cherche à modifier un bloc précédent, elle ne pourra enregistrer ces modifications puisque le hash généré sera différent et le lien avec le bloc suivant ne pourra donc pas se faire.

Figure 1.6 : Liaison des blocs par leur hash

En conséquence, modifier le contenu d'un bloc suppose de recalculer les hashs de tous les blocs qui le suivent, ce qui nécessiterait une puissance de calcul d'un ordinateur supérieure à tous les ordinateurs du réseau, rendant donc cette tâche impossible.

C'est là un des forts intérêts de la blockchain : elle ne perd aucune information dans son évolution. Nous pouvons être certains que le registre des transactions n'a pas subi de traitement malveillant, et que l'état du registre à un instant donné reflète la réalité. [12]

1.5.3 Processus de consensus

Pour qu'un bloc puisse être lié à un autre, il doit être validé. Cette validation est faite par certains utilisateurs, appelés des « mineurs », qui donnent la puissance de calcul nécessaire pour établir un bloc. Chaque bloc validé est transmis aux nœuds du réseau, aux détenteurs de la blockchain qui s'actualise en permanence.

L'identité d'une personne derrière un nœud peut différer selon le type de blockchain, ainsi, dans une blockchain dite :

- « **Ouverte** », un nœud peut être n'importe quel utilisateur d'internet,
- « **Fermée** », un nœud sera un utilisateur prédéterminé.

Chaque utilisateur, chaque nœud, peut télécharger la blockchain d'un nœud existant, chaque nœud est connecté à un ou plusieurs autres appelés pairs, ayant eux-mêmes leurs propres pairs.

Nous avons, de ce fait, ce qu'on appelle un réseau pair à pair.

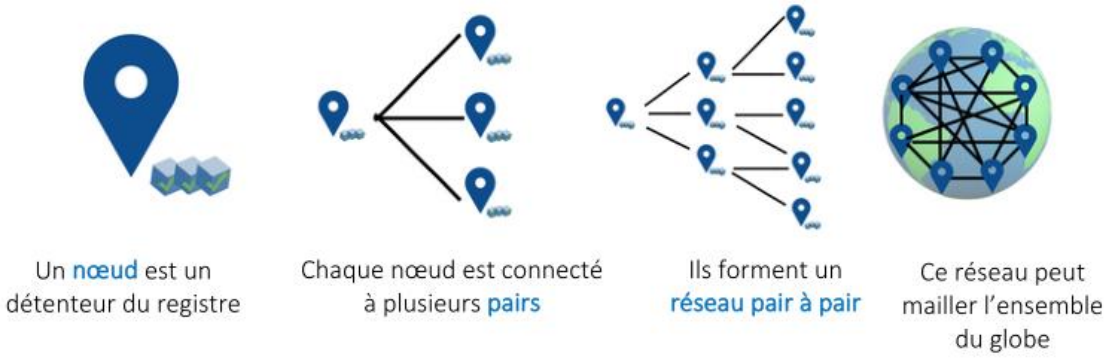


Figure1.7 : Notion de réseau pair à pair

Source : Office Parlementaire d'Evaluation des Choix Scientifiques

Une fois qu'un nouveau bloc est créé, et qu'il est transmis à un nœud, le nœud l'ajoute à sa copie du registre et le transmet à ses nœuds pairs. Les nœuds pairs vérifient à nouveau que ce bloc soit bien valide, et, si tel est le cas, l'intègrent à leur registre, et le transmettent à nouveau à leurs pairs.

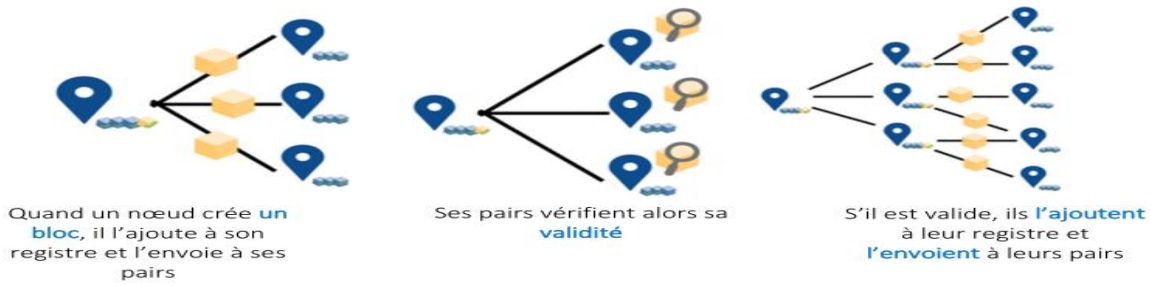


Figure 1.8 : Introduction d'un bloc invalide

Source : Office Parlementaire d'Evaluation des Choix Scientifiques

Le processus de validation expliqué ci-dessus permet de se prémunir du risque d'attaque malveillante, le tout sans autorité centrale de contrôle.

C'est ici un aspect essentiel de la blockchain, la sécurité, l'intégrité et la confiance en éliminant la nécessité d'un acteur central, d'une tierce partie.

La puissance de la blockchain et de son partage fait que, plus celle-ci est distribuée, plus elle est sécurisée. La multiplication des nœuds (et donc des acteurs de la blockchain) entraine une vérification accrue des blocs, rendant la blockchain plus fiable face aux manipulations malveillantes des données. [12]

1.6 Nœuds du réseau, « mineurs » et consensus

Chaque bloc est validé par certains utilisateurs baptisés « **mineurs** » (en référence aux chercheurs d'or), et sont transmis aux « nœuds » du réseau, c'est-à-dire aux détenteurs du registre, qui l'actualisent en permanence. La validation des blocs permet de se prémunir du risque d'attaques malveillantes. Aucune autorité centrale ne s'en occupe, puisque les utilisateurs s'en chargent en surveillant le système et en se contrôlant mutuellement. Cette sécurité, source de confiance, est l'un des aspects essentiels de la *blockchain*. Le fait que des centaines de copies du registre soient mises à jour simultanément et régulièrement, au terme d'une compétition cryptographique, rend les *blockchains* quasiment indestructibles. Une « **méthode de consensus** » permet de décider qui validera le prochain bloc à ajouter à la chaîne.

Dans le cas du bitcoin, elle est appelée « **preuve de travail** » (*proof of work*) car elle suppose la réussite à une épreuve cryptographique dénommée « **minage** », qui se répète en moyenne toutes les dix minutes. Elle consiste en la résolution par les mineurs de problèmes cryptographiques complexes. Ils consistent à obtenir un *hash*, commençant par un certain nombre de zéros, du bloc que le mineur souhaite intégrer. Cette opération, très coûteuse en puissance de calcul informatique, est motivée par l'obtention d'une récompense en bitcoins par le mineur gagnant. Le bloc validé par ce dernier est transmis de pair à pair à chaque nœud qui ajoute à sa propre *blockchain* le bloc ainsi validé. Si deux blocs sont validés au même moment, les mineurs utilisent l'un ou l'autre et **deux chaînes parallèles se développent**. Le protocole prévoit alors que, rapidement, seule **la plus longue subsiste**, c'est-à-dire en pratique celle que la majorité des nœuds aura adoptée. [5]

La rémunération des mineurs est complétée par des **frais** prélevés sur les transactions qu'ils intègrent à chaque nouveau bloc. Leur montant est en théorie déterminé librement par les utilisateurs, mais les mineurs sélectionnant en priorité les plus élevés, ces frais varient de fait en fonction du nombre de transactions en attente. L'**organisation des mineurs en groupements** ou « *pools* » induit le risque qu'une majorité organisée oriente la validation des blocs. La confiance des utilisateurs dans le système étant en théorie un objectif partagé par les mineurs, celui-ci est censé suffire à garantir le respect des règles, dans une logique de « main invisible » protégeant les intérêts privés. Il faut cependant souligner que quatre pools dont trois chinois, appuyés sur des « fermes de minage », assurent aujourd'hui plus de 60 % de la puissance de calcul nécessaire à la *blockchain* du bitcoin et pourraient utiliser cette position dominante contre l'intérêt des autres utilisateurs. [5]

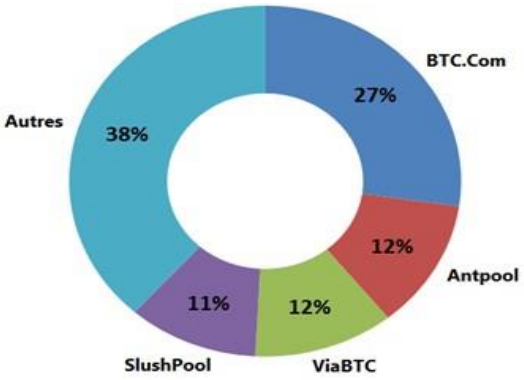


Figure 1.9 : Les « pools » de mineurs du bitcoin [5]

D'autres méthodes de consensus que la « preuve de travail » (proof of work) existent et sont souvent plus centralisées: la principale alternative, qui présenterait un risque plus grand d'utilisation malveillante, est la « preuve d'enjeu », appelée aussi « preuve de participation » (proof of stake), basée sur la possession de cryptomonnaies mises en gage, qui se décline à son tour en « preuve de possession » (proof of hold), fondée sur la durée de possession, « preuve d'utilisation » (proof of use), fonction du volume de transactions, ou encore « preuve d'importance » (proof of importance), reposant sur la « réputation ». Deux autres méthodes moins usitées peuvent aussi être évoquées : la « preuve de capacité » (proof of space) qui consiste à mettre en gage de l'espace disque disponible, ou encore la « preuve de destruction » (proof of burn) qui revient à détruire des cryptoactifs, pour obtenir la confiance du réseau. [5]

1.7 Contrats intelligents

Dans la blockchain, les contrats intelligents vont au-delà de simples transactions de bitcoins et comportent des instructions plus détaillées (traitement) intégrées. Formellement, un contrat intelligent est une méthode d'utilisation de la blockchain (ou transactions bitcoin) pour former des accords entre agents. En général, un contrat c'est une promesse faite à deux agents ou plus de travailler (ou de ne pas faire) de travail en échange de quelque chose d'autre. Chaque agent doit faire confiance à l'autre agent pour remplir sa part de l'engagement. Le contrat intelligent comporte le même type de règlement pour agir ou ne pas agir, mais il élimine la nécessité pour un agent de faire confiance aux autres agents. En effet, un contrat intelligent est un code logiciel exécuté sur une blockchain sans aucune discrétion. En fait, deux éléments des contrats intelligents qui les distinguent sont l'autoapplicabilité et la décentralisation. L'auto-applicabilité signifie qu'après son lancement, les agents engagés dans le contrat intelligent n'ont plus besoin d'être en contact. Décentralisé signifie que les contrats intelligents ne subsistent pas sur un seul serveur centralisé ; ils sont distribués et s'exécutent automatiquement sur le réseau de chaînes de blocs. L'illustration classique des contrats intelligents dans la vie quotidienne est un distributeur automatique. Contrairement à une personne, le distributeur automatique se comporte de manière algorithmique ; le même jeu d'instructions sera exécuté à chaque fois dans tous les cas.

Un exemple de contrat intelligent de base, avec des instructions plus détaillées par rapport aux bitcoins, est *un cadeau d'héritage disponible dès le dix-huitième anniversaire*.

Vous pouvez créer une transaction qui repose sur la chaîne de blocs et reste non initiée jusqu'à ce que les deux conditions suivantes soient déclenchées.

1. Le programme définit la date (18ème anniversaire) à laquelle lancer la transaction, ce qui inclut la vérification si la transaction a déjà été exécutée.
2. Le programme analyse une base de données de registre de décès en ligne pour certifier que l'entité héréditaire (parent ou grand-parent) est décédée. Lorsque le contrat intelligent confirme le décès, il peut automatiquement transférer l'héritage (par exemple, des fonds).

1.8 Types de cryptographie dans Blockchain

1.8.1 Définition de la cryptographie

La cryptographie est l'**art de chiffrer**, coder les messages est devenue aujourd'hui une science à part entière. Au croisement des **mathématiques**, de l'**informatique**, et parfois même de la **physique**, elle permet ce dont les civilisations ont besoin depuis qu'elles existent : le maintien du secret. Pour éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses. [6]

1.8.2 L'usage de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur **intégrité** et leur **authenticité.**

- **La confidentialité** : consiste à rendre l'information intelligible à d'autres personnes que les acteurs de la transaction.
- **L'intégrité** : vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication.
- **L'authentification :** consiste à assurer l'identité d'un utilisateur, c.-à-d de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.
- **La non répudiation :** de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction. [6]

1.8.3 Mécanisme de la cryptographie

Un algorithme de cryptographie ou un chiffrement est une fonction mathématique utilisée lors du processus de cryptage et de décryptage. Cet algorithme est associé à une clé (un mot, un nombre ou une phrase), afin de crypter une donnée. Avec des clés différentes, le résultat du cryptage variera également. La sécurité des données cryptées repose entièrement sur deux éléments : l'invulnérabilité de l'algorithme de cryptographie et la confidentialité de la clé**. Qu'entend-on par clé ?**

On appelle clé une valeur utilisée dans un algorithme de cryptographie, afin de chiffrer une donnée. Il s'agit en fait d'un nombre complexe dont la taille se mesure en bits. On peut imaginer que la valeur correspondant à 1024 bits est absolument gigantesque. Voir aussi bits bytes. Plus la clé est grande, plus elle contribue à élever la sécurité à la solution. Toutefois, c'est la combinaison d'algorithme complexe et de clés importantes qui seront la garantie d'une solution bien sécurisée.

Les clés doivent être stockées de manière sécurisée et de manière à ce que seul leur propriétaire soit en mesure de les atteindre et de les utiliser. [4]

1.8.4 Confidentialité et algorithmes de chiffrement

La confidentialité est le premier problème posé à la cryptographie. Il se résout par la notion de chiffrement. Il existe deux grandes familles d'algorithmes cryptographiques à base de clef :

Les algorithmes à clef secrète ou algorithmes symétriques, et les algorithmes à clef publique ou algorithmes asymétriques. [4]

1.8.5 Cryptographie symétrique

Chiffrement symétrique ou clef secrète : dans la cryptographie conventionnelle, les clefs de chiffrement et de déchiffrement sont identiques : c'est la clef secrète, qui doit être connue des tiers communiquant et d'eux seuls. Le procédé de chiffrement est dit symétrique.

Les algorithmes symétriques sont de deux types :

- Les algorithmes de chiffrement en continu, qui agissent sur le message en clair un bit à la fois ;
- Les algorithmes de chiffrement par bloc, qui opèrent sur le message en clair par groupes de bits appelés blocs. Les principaux algorithmes à clé privée sont : Blowfish ; DES/3DES ; IDEA.

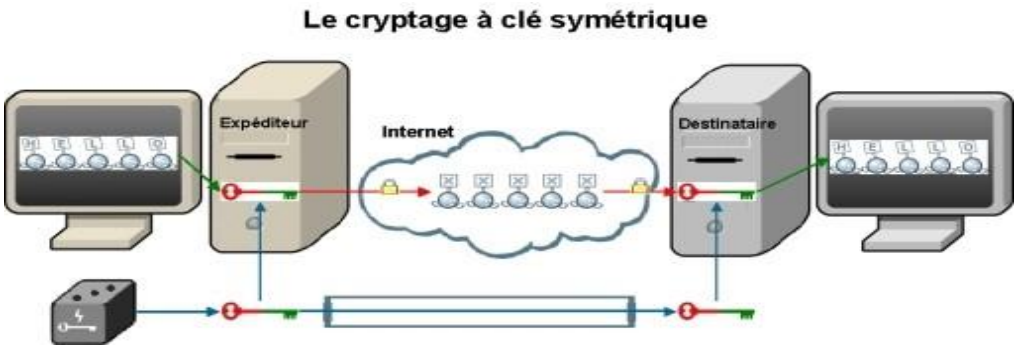


Figure 1.10 : cryptage à clé secrète [4]

1.8.6 Cryptographie Asymétrique

Chiffrement asymétrique ou à clef public : avec les algorithmes asymétriques, les clefs de chiffrement et de déchiffrement sont distinctes et ne peuvent se déduire l'une de l'autre. On peut donc rendre l'une des deux publique tandis que l'autre reste privée. C'est pourquoi on parle de chiffrement à clef publique. Si la clef publique sert au chiffrement, tout le monde peut chiffrer un message, que seul le propriétaire de clef privé pourra déchiffrer. On assure ainsi la confidentialité. Certains algorithmes permettent d'utiliser la clef privée pour chiffrer. Dans ce cas, n'importe qui pourra déchiffrer, mais seul le possesseur de clef privée peut chiffrer. [4]

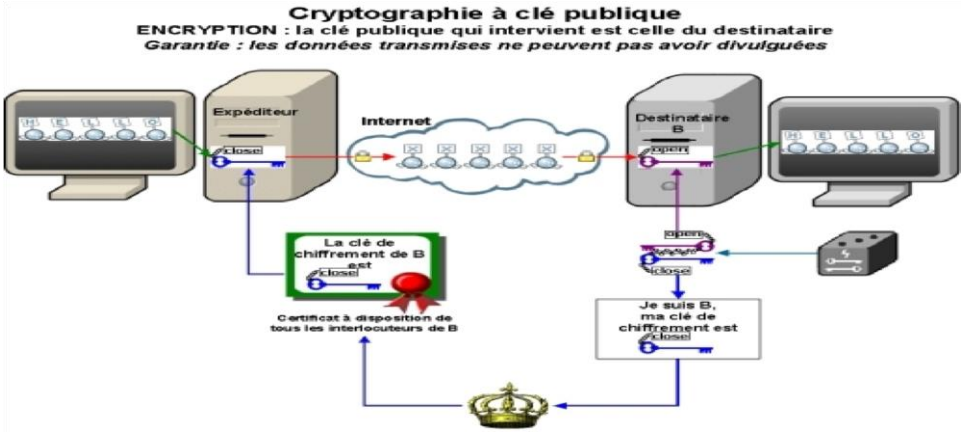


Figure 1.11 : cryptage à clé publique [4]

1.8.7 La signature numérique

La signature électronique est un ensemble de mesures techniques qui visent à garantir l'intégrité d'un document et d'en authentifier son auteur. Cette dernière est donc obligatoirement liée à un document mais également à la personne qui l'appose. Ainsi, la

signature électronique a pour objectif de démontrer à un tiers que le document signé a été approuvé par une personne identifiée. [7]

La blockchain quant à elle fonctionne comme un vaste registre public intégrant l'ensemble des transactions effectuées par ses utilisateurs depuis sa création. Ces transactions sont regroupées à l'intérieur de blocs qui sont ordonnés du plus ancien au plus récent. Chaque bloc contient des informations relatives au bloc précèdent de sorte qu'il est impossible de modifier un bloc sans avoir à modifier toute la blockchain en aval. Les utilisateurs peuvent télécharger l'intégralité de la blockchain et vérifier à tout moment son intégrité. Le contrôle de la Blockchain est donc décentralisé. Prenons l'exemple de blockchain Bitcoin, ainsi lorsqu'un utilisateur souhaite transférer une valeur à un autre utilisateur, il va signer une transaction avec une clé privée qu'il est le seul à connaitre et renseigner l'adresse Bitcoin de l'utilisateur bénéficiaire. Des mineurs possédant la copie complète de la blockchain vont alors vérifier la validité de la transaction et sa conformité vis-à-vis de l'historique de la blockchain. Si la blockchain confirme ensuite que l'utilisateur possède le solde de crypto-actif nécessaire à sa transaction (pour rémunérer les mineurs), la transaction sera donc rajoutée au nouveau bloc de la chaine. [7]

En outre, il convient de préciser que les informations objet des transactions sont « ancrées » dans la blockchain au moyen de différentes mesures de sécurité, sachant que techniquement ce ne sont ni les informations, ni les documents qui sont stockés en tant que tels dans la blockchain mais uniquement leur empreinte numérique (dite « hash ») inscrite de façon irréversible, immuable, intangible.

Dès lors, la signature électronique sur la blockchain est un ensemble de mesures techniques qui vise à sceller une transaction permettant ainsi d'authentifier la signature.

Eu égard aux garanties précitées fournies par la blockchain, on peut s'interroger sur la force probante de la signature électronique sur la blockchain comme mode de preuve. [7]

La signature électronique est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur, par analogie avec la signature manuscrite d'un document papier.

1.9 Hachage

le « **hachage** » permet de convertir n'importe quel ensemble de données numériques en un *hash*, c'est-à-dire en une courte suite binaire qui lui est propre. L'algorithme de chiffrement

Utilisé à cet effet est appelé « fonction de hachage cryptographique ». Le *hash* d'un ensemble de données peut ainsi être comparé à une empreinte digitale, bien moins complexe que l'individu entier, mais l'identifiant de manière précise et unique. Une fonction de hachage est dite « à sens unique » : elle est conçue de telle sorte que le *hash* produit, à savoir une image ou empreinte de taille fixe créée à partir d'une donnée de taille variable, fournie en entrée, est impossible à inverser. Celle utilisée pour le bitcoin est parmi les plus répandues : il s'agit de la fonction *Secure Hash Algorithm-256* (SHA-256), ainsi dénommée car elle produit des *hashs* d'une taille de 256 bits**.** [5]

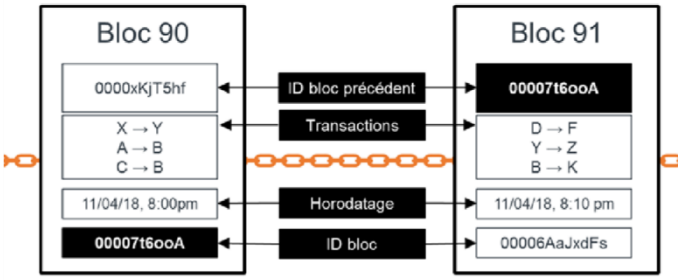


Figure 1.12 : La structure d'une blockchain [5]

1.10 Fonction de Hachage

Les fonctions de hachage cryptographique permettent de garantir l'intégrité d'un document de par leurs propriétés d'irréversibilité et d'unicité (dite de « résistance aux collisions »). En effet, une fonction de hachage produit un résultat (ou hash) de taille fixe (quelle que soit la taille du document fourni, la fonction retourne toujours un résultat de même taille). Aussi, pour le même exact document, le hash calculé sera toujours exactement le même

• L'irréversibilité de cette fonction signifie qu'il est impossible, compte tenu des outils algorithmiques et informatiques actuels, de réussir à trouver le document fourni à partir du résultat de cette fonction ;

• La résistance aux collisions signifie qu'il est impossible de trouver deux documents qui aboutissent au même hash;

• De facto, si deux documents donnent le même hash, c'est qu'ils sont identiques;

• Horodater un hash dans une blockchain correspond à horodater le document. Signer le hash correspond à signer le document.

Les fonctions de hachage sont très utilisées dans les protocoles blockchain. Elles servent ainsi à générer des signatures pour authentifier chaque transaction, à garantir un lien entre l'adresse d'un utilisateur de la blockchain et sa clé publique, à identifier une transaction ou un bloc ou encore à lier les blocs de la blockchain entre eux de manière à garantir l'intégrité de cette blockchain.

Dans un contexte de notarisation de documents sur une blockchain, il est très intéressant d'utiliser un arbre de Merkle pour pouvoir stocker au sein de cet arbre le résultat du hachage d'un nombre important de documents et d'enregistrer uniquement sur la blockchain la valeur de la racine de l'arbre. Cela présente l'avantage de réduire considérablement le volume de données à stocker sur la blockchain ainsi que le nombre de transactions à réaliser pour enregistrer ces données sur la blockchain, tout en garantissant l'intégrité de chaque document. [7]

1.11 Types de la blockchains

La blockchain peut être avec permission (privée) ou sans permission (publique). La première catégorie impose des restrictions aux contributeurs du consensus. Seul ceux de confiances et choisis qui ont le droit de valider des transactions. Elle ne nécessite pas beaucoup de calcul pour atteindre un consensus, ainsi, elle est économique en termes de temps d'exécution et en énergie. Généralement les transactions sont privées et ne sont accessibles que par les objets autorisés. La deuxième catégorie (blockchain publique) utilise un nombre illimité d'objets anonymes. En se basant sur la cryptographie, chaque acteur peut communiquer d'une manière sécurisée. Chaque objet est représenté par une paire de clés (publique/privée), et a le droit de lire, d'écrire et de valider des transactions dans la blockchain. La blockchain est sûre si 51% des objets (ou plus) sont honnêtes et lorsque le consensus du réseau est atteint. Généralement, les blockchains sans permission consomment beaucoup d'énergie et de temps, car elles exigent un montant de calcul pour renforcer la sécurité du système (ex. en utilisant la PoW).

Il existe trois types de Blockchain, selon leur mode de fonctionnement: [3]

1.11.1 Blockchain public

Tout le monde peut lire ou écrire des données et la seule condition est de disposer d'un ordinateur et d'une connexion Internet. Une partie de ce type de réseau restreint l'accès

uniquement en lecture ou en écriture. Ethreum et Bitcoin sont des exemples qui utilisent une approche où tout le monde peut écrire. [3]

1.11.2 Blockchain privée

N'est pas ouvert au public, mais est accessible uniquement sur invitation et tous les membres participants se connaissent et se font confiance. Ceci est très utile lorsque la Blockchain est utilisée entre entreprises appartenant à la même branche. Parmi les plus célèbres, citons Hyperledger (de Linux Foundation) et Ripple (protocole permettant les transferts internationaux). [3]

1.11.3 Blockchain permissionnée

Aussi connu sous le nom de Consortium Blockchain, est un hybride entre Blockchain publique et privée. Dans ce type, seuls quelques nœuds sélectionnés sont prédéterminés et les nœuds participants sont invités, mais toutes les transactions sont publiques. Cela signifie que les nœuds participent à la maintenance et à la sécurité de ce réseau, mais que toutes les transactions sont visibles pour les utilisateurs du monde entier. Le droit de lecture peut être public ou limité aux participants. Les Blockchains du consortium préservent la confidentialité des données, comme les Blockchains privés. BigchainDB est un exemple de consortium Blockchain. [3]

1.12 Acteurs de Blockchain

Une solution de blockchain d'entreprise nécessite que de nombreux acteurs jouant différents rôles soient pleinement fonctionnels : [3]

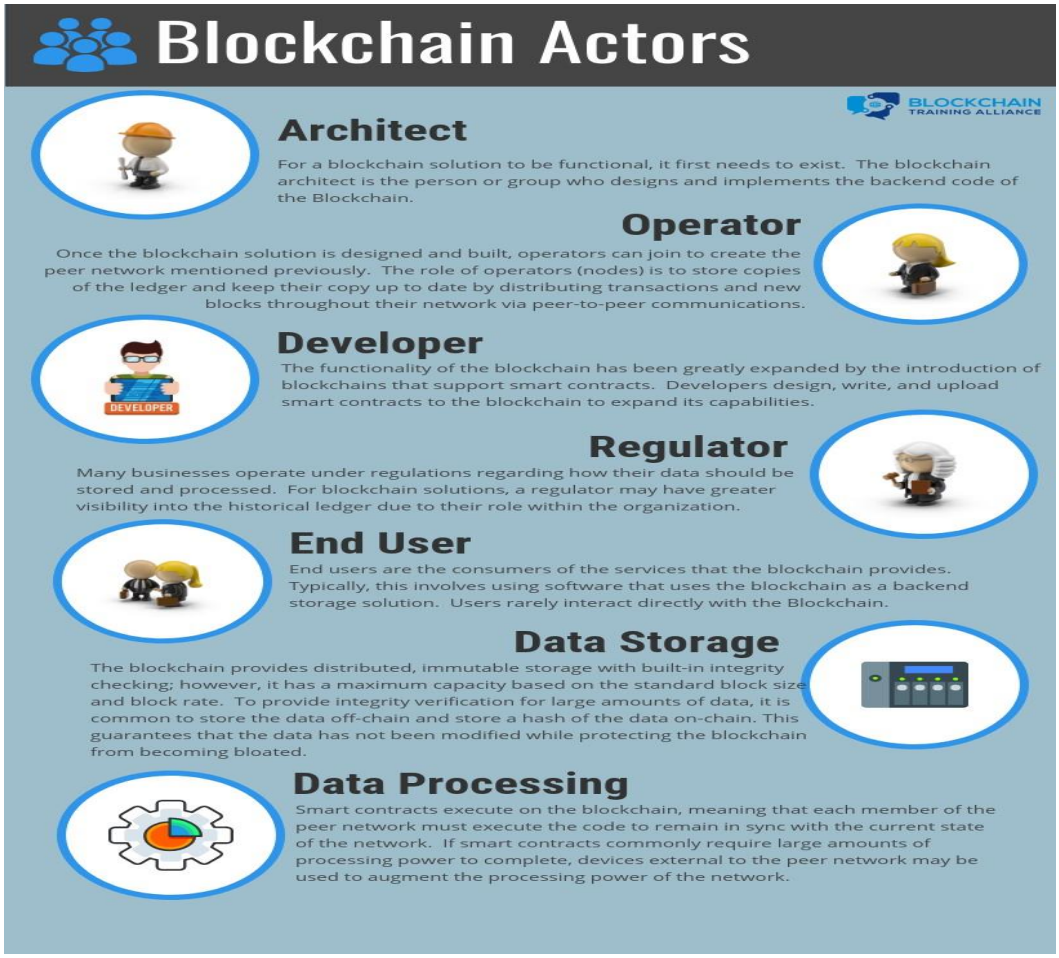


Figure 1.13 : Acteurs de Blockchain

Un architecte blockchain: est le designer de la solution blockchain. Pour qu'une solution blockchain soit fonctionnelle, elle doit d'abord exister. L'architecte blockchain est la personne ou le groupe qui a conçu la blockchain.

L'opérateur de la chaîne de blocs: stocke, tient à jour et met à jour le livre des chaînes de bloc. Une fois que la solution blockchain est conçue et réalisée, un opérateur peut s'associer pour créer le réseau homologue mentionné précédemment. Le rôle de l'opérateur est de configurer et de maintenir des pairs au sein du réseau.

Le développeur Blockchain: crée des contrats intelligents à exécuter sur la blockchain. La fonctionnalité de la blockchain a été considérablement étendue par l'introduction de blockchains prenant en charge les contrats intelligents. Les développeurs conçoivent et téléchargent des contrats intelligents dans la blockchain pour étendre ses capacités. Outre la mise en œuvre des contrats intelligents, des développeurs front-end peuvent également

implémenter des applications qui accèdent à la blockchain (c'est-à-dire que les applications initient les transactions sur la blockchain).

Le régulateur Blockchain: de nombreuses entreprises sont soumises à des réglementations concernant la manière dont leurs données doivent être stockées et traitées. Pour les solutions de type blockchain, un régulateur peut avoir une plus grande visibilité dans le grand livre historique en raison de son rôle au sein de l'organisation.

L'utilisateur final: est le consommateur de services construits autour de la blockchain. En règle générale, cela implique l'utilisation d'un logiciel qui utilise la blockchain comme solution de stockage principale. Les utilisateurs interagissent rarement directement avec la blockchain.

Le stockage de données: est représenté par les bases de données traditionnelles pour stocker les données hors chaîne. La blockchain fournit un stockage distribué immuable avec un contrôle d'intégrité intégré; Cependant, sa capacité maximale est basée sur la taille et le taux de blocs standard. Pour permettre la vérification de l'intégrité de grandes quantités de données, il est courant de stocker les données hors chaîne et de stocker un hachage des données en chaîne. Cela garantit que les données ne sont pas modifiées tout en protégeant la blockchain contre le gonflement.

Le traitement des données: est représenté par un système externe utilisé pour un traitement supplémentaire. Les contrats intelligents s'exécutent sur la blockchain, ce qui signifie que chaque membre du réseau homologue doit exécuter le code pour rester synchronisé avec l'état actuel du réseau. Si les contrats intelligents nécessitent généralement une grande quantité de puissance de traitement, des périphériques externes au réseau homologue peuvent être utilisés pour augmenter la puissance de traitement du réseau.

1.13 Avantages de la technologie Blockchain

Les chaînes de blocs peuvent renforcer la sécurité principalement sur trois aspects : le blocage du vol d'identité, la prévention de la manipulation des données et l'arrêt des attaques par déni de service.

- Blocage de vol d'identité: La structure de la preuve de travail du mineur de réseau de Blockchain et son grand livre distribué de transactions de données réduisent les risques de vol et de corruption des données. [3]

- Prévenir la manipulation et la fraude des données: Dans la technologie Blockchain, la cryptographie, le hachage et une structure décentralisée empêchent quasiment tout membre de modifier les données du grand livre. Cela empêche et détecte toute forme de manipulation et permet aux organisations de maintenir la protection des informations. Une solution importante qui a été développée pour éviter la fraude et la manipulation est KSI (Keyless Signature Infrastructure), qui assure la protection des réseaux ainsi que la sécurité et la confidentialité des données.

Avec KSI, personne ne peut manipuler les données et l'authenticité des données électroniques peut être prouvée mathématiquement. KSI stocke les signatures numériques des fichiers originaux dans une Blockchain, puis vérifie les copies en revérifiant les signatures des copies par rapport à celles stockées dans la Blockchain. Si une quelconque manipulation est effectuée, elle est détectée très rapidement car les hachages stockés dans la chaîne de caractères résident dans des milliers de nœuds. KSI Technology est utilisée activement dans les secteurs de l'aérospatiale et de la défense, ainsi que dans le secteur de la santé, afin de mieux contrôler le dossier médical du patient. [3]

- Prévention des attaques par déni de service distribué: Il existe un grand nombre d'infrastructures critiques à protéger. Blockchain peut aider avec DNS (Domain Name System) qui fournit un accès à des sites Web utilisant des noms de domaine plutôt que des adresses IP. Le système DNS est dangereusement centralisé dans quelques serveurs racine sous le contrôle de l'ICANN (Internet Corporation for Assigned Names and Numbers), qui est responsable des adresses de protocole IP, des identificateurs de protocole, des fonctions de gestion de système de domaine et de la gestion de système du serveur racine. Blockchain pourrait créer un DNS distribué, beaucoup plus transparent, rendant pratiquement impossible la manipulation des enregistrements par une seule entité.

Il existe certaines différences entre les réseaux Blockchain et le paradigme du Cloud Computing. Dans le modèle en nuage, les périphériques IoT sont identifiés, authentifiés et connectés via des serveurs en nuage, où le traitement et le stockage sont souvent effectués. Les réseaux IoT ayant des coûts élevés sont concernés par le modèle de cloud centralisé. Les appareils IoT sont vulnérables aux attaques DDoS, au vol de données, au piratage et au piratage à distance. Si un périphérique IoT connecté à un serveur fait l'objet d'une violation, toutes les personnes connectées au serveur pourraient être affectées. En outre, le modèle de

nuage centralisé est sujet à la manipulation. Les données collectées ne garantissent pas que les informations sont utilisées de manière appropriée. Blockchain peut éliminer ces problèmes de Cloud Computing. Dans Blockchain, les échanges de messages entre périphériques peuvent être traités de la même manière que les transactions financières dans un réseau bitcoin. Les appareils reposent sur des contrats intelligents qui garantissent plus de sécurité. Le fait que Blockchain vérifie de manière cryptographique les transactions signées, elle élimine la possibilité d'attaque par interférence, de rejeu ou d'autres attaques. [3]

1.14 Inconvénients De La Blockchain

- **Vitesse réduite pour les transactions numériques**

Les blockchains nécessitent d'énormes quantités de puissance de calcul, ce qui tend à réduire la vitesse des transactions numériques, bien qu'il existe des solutions de contournement, il est conseillé d'utiliser des bases de données centralisées lorsque vous avez besoin de transactions à grande vitesse en millisecondes. [39]

- **Immutabilité**

Des données L'immutabilité des données a toujours été l'un des plus gros inconvénients de la blockchain. Il est clair que plusieurs systèmes en bénéficient, notamment la chaîne d'approvisionnement, les systèmes financiers, etc. Cependant, il souffre du fait qu'une fois les données écrites, elles ne peuvent pas être supprimées. Chaque personne sur terre a droit à la vie privée. Cependant, si la même personne utilise une plate-forme numérique qui fonctionne sur la technologie blockchain, alors elle ne pourra pas supprimer sa trace du système lorsqu'elle ne le souhaite pas. En termes simples, il n'y a aucun moyen de supprimer sa trace – laissant les droits à la vie privée en morceaux. [39]

- **Nécessite une expertise Connaissance**

La mise en œuvre et la gestion d'un projet blockchain est difficile. Cela nécessite des connaissances approfondies pour passer par l'ensemble du processus. C'est pourquoi il est difficile de rencontrer des spécialistes ou des experts blockchain car il faut beaucoup de temps et d'efforts pour former un expert blockchain. Par conséquent, cet article est un bon point de départ et un bon guide si vous avez déjà commencé. [39]

- **Interopérabilité**

Plusieurs réseaux de chaînes de blocs travaillant durs pour résoudre le problème du grand livre

distribué rendent difficile de les relier ou de les intégrer les uns aux autres. Cela rend la communication entre les différentes chaînes difficile. [39]

- **Intégration**

d'applications héritées De nombreuses entreprises et applications utilisent encore des systèmes et une architecture hérités ; L'adoption de la technologie blockchain nécessite une refonte complète de ces systèmes qui, je dois le dire, n'est pas réalisable pour beaucoup d'entre eux. [39]

1.15 Usage de la blockchain (Domaine d'applications)

Dans cette section, On présente certaines des applications potentielles de la technologie Blockchain, Ces applications sont ; Chaîne d'approvisionnement, identité numérique, vote, santé et gouvernement. (3)

-Chaîne d'approvisionnement: La chaîne d'approvisionnement est un segment très complexe et il est devenu plus difficile d'avoir une visibilité transparente sur l'ensemble de la chaîne d'approvisionnement. Il est devenu plus difficile de suivre le flux de matériel et les canaux de distribution, ce qui a entraîné divers comportements contraires à l'éthique dans les entreprises, allant du commerce illégal aux produits de contrefaçon et aux dommages environnementaux. Tout au bout de la chaîne d'approvisionnement, les consommateurs ne disposent pas des informations selon lesquelles un produit final a été importé tout au long de la chaîne d'approvisionnement. De nos jours, vous pouvez perdre vos colis par la poste. En tirant parti de la convergence du paradigme IoT et des contrats intelligents, vous pourrez enregistrer la position à tout moment de vos colis grâce à la connexion de capteurs à chaque étape.

Le contrat intelligent: apporte la fiabilité tout au long de la ligne, permettant avec sécurité où trouver le paquet. Identité numérique. Cette application pourrait permettre aux consommateurs d'avoir une identité enregistrée sur un grand livre partagé et d'ajouter des appareils à leur identité.

L'identité numérique: garantit un moyen plus sûr de vérifier l'authenticité d'une personne et d'éviter et de réduire les fraudes possibles.

Vote: Blockchain peut transformer le système de vote traditionnel sur papier en un système numérisé et peut fournir une plate-forme de vote sécurisée servant de support à tout le processus ; voter, dépister et compter les votes et éviter des problèmes tels que la perte de

registres et la fraude électorale. Les électeurs pouvaient compter les votes eux-mêmes et vérifier qu'aucun vote n'avait été supprimé, manipulé ou modifié.

Gouvernement: La blockchain pourrait être utilisée pour assurer au public que les politiciens agissent correctement avec l'argent, et peut également lutter contre le crime financier. Grâce à la technologie, chaque transaction peut être enregistrée sans manipulation, ce qui rend la destination ultime transparente pour le public.

Soins de santé: Les établissements de santé doivent faire face à des problèmes de sécurité et de confidentialité lorsqu'ils partagent des données sur plusieurs plates-formes. L'amélioration de la collaboration de données entre fournisseurs signifie l'amélioration de nombreux aspects du domaine de la santé, tels que la précision des diagnostics et l'efficacité des traitements. Blockchain peut créer cet environnement sécurisé pour permettre aux établissements de santé, aux payeurs et aux autres acteurs de ce domaine de partager l'accès à leur réseau avec des garanties d'intégrité des données.

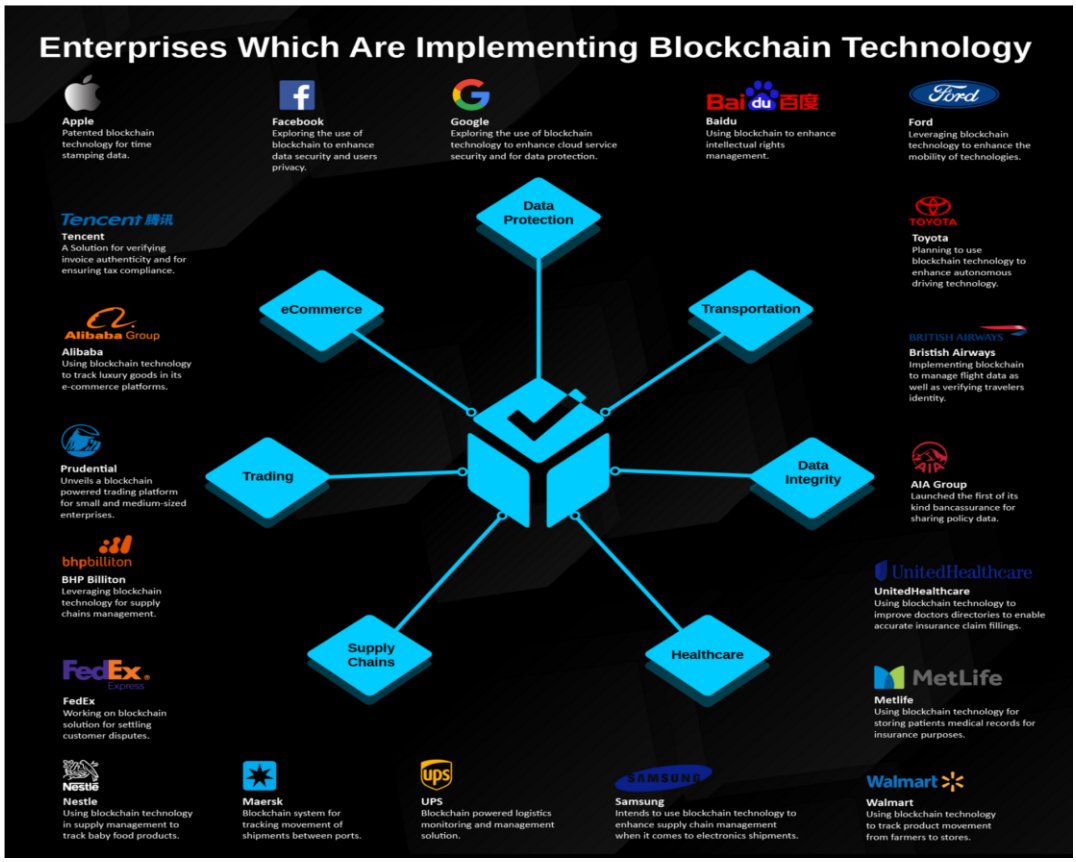


Figure 1.14 : Entreprises implémentant la Blockchain

Source: blockchain101.com

1.16 Blockchain aujourd'hui

1.16.1 Bitcoin

Il a été introduit pour la première fois sur une liste de diffusion cryptographique le 31 octobre 2008 et a été publié en tant que logiciel à code source ouvert en 2009. L'idée de construire cette monnaie numérique ou virtuelle est de construire un système décentralisé qui permettrait à la monnaie de pouvoir être transférée électroniquement avec des frais de transaction négligeables ou nuls. Il s'agit du premier réseau peer-peer qui permet à ses utilisateurs de ne disposer d'aucune autorité centrale ni de banques. Ces bitcoins sont construits à partir d'un protocole bitcoin. Selon ce protocole, il y aurait un nombre défini de bitcoins pouvant être produits (extraits), soit 21 millions. Cependant, chaque bitcoin peut être divisé en parties plus petites pouvant aller jusqu'à un centième de bitcoin. Cette plus petite division du bitcoin est appelée "Satoshi".

Bitcoin a des utilisateurs qui sont répandus à travers le monde et il n'est pas contrôlé par une seule personne. Du point de vue de l'utilisateur, bitcoin est une application qui fournit un portefeuille et permet aux utilisateurs d'effectuer des transactions entre eux. Pour effectuer des transactions, les utilisateurs sont libres d'utiliser le logiciel de leur choix, à condition que le logiciel soit compatible et respecte les règles du protocole Bitcoin. Ce système de paiement électronique repose sur une preuve cryptographique plutôt que sur la confiance, raison pour laquelle le besoin d'un tiers est éliminé. Les transactions effectuées sont pratiquement impossibles à inverser ou à détruire. Le réseau bitcoin partage un grand livre public appelé blockchain. [3]

1.16.2 Ethereum

Ethereum est un projet open source construit par les développeurs du monde entier, semblable au protocole Bitcoin, mais beaucoup plus adaptable et flexible, car il permet aux utilisateurs de créer et d'utiliser les applications décentralisées exécutées sur la technologie de blockchain sous-jacente. Généralement, il existe deux approches pour créer une application blockchain: démarrer un réseau indépendant ou établir un protocole sur Bitcoin. Ethereum a introduit une blockchain avec un langage de programmation intégré complet de Turing qui permet à quiconque d'écrire des contrats intelligents et des applications décentralisées avec des règles et des paramètres personnalisés.

Contrairement aux états dans Bitcoin, Ethereum a des comptes. Les deux types de comptes sont les suivants: (1) comptes contrôlés de manière externe (ou) comptes d'utilisateur, et (2) contrats, c.-à-d. Extraits de code. Les transactions peuvent être initiées à partir des deux types de comptes, mais les contrats ne peuvent démarrer une transaction qu'à la suite d'autres transactions qu'ils ont reçues. Les contrats sont écrits dans un langage de programmation de haut niveau (Solidity, par exemple), qui est ensuite converti en bytecode Ethereum Virtual Machine (EVM). La devise intégrée pour le réseau Ethereum, Ether, peut être utilisée pour échanger des actifs numériques et fournit également un mécanisme pour payer les frais de transaction. La plus petite dénomination d'Ether est Wei (1018 Wei = Ether). Dans Ethereum, il n'y a pas de limite de taille de bloc comme dans Bitcoin, mais il existe un concept appelé «Gas». Dans Ethereum, tous les calculs programmables, y compris la création de contacts, l'exécution d'opérations et la réalisation d'appels de messages, ont un coût convenu universellement, mesuré en termes de gaz. Au lieu d'une limite de taille de bloc, il existe une limite de gaz (définie par l'expéditeur de la transaction) pour chaque transaction, ce qui signifie que la validation de cette transaction ne doit pas utiliser plus de gaz que la limite mentionnée. Le gaz restant non utilisé à la fin de la transaction est remboursé sur le compte de l'expéditeur. De plus, le temps d'exploitation de blocs dans Ethereum est considérablement réduit à 15 secondes en moyenne par rapport aux 10 minutes de Bitcoin. Cela se fait par la mise en œuvre du protocole GHOST, qui est une politique de sélection de la chaîne principale dans l'arborescence de blocs. Toutefois, ce temps peut être réduit davantage en fonction de la taille des transactions et de la difficulté de calcul liée à la validation d'un bloc. (3)

Les sujets	Bitcoin	Ethereum
Concept	Monnaie numérique	Contrats intelligents
Fondateur	Satoshi Nakamoto	Vitalik Buterin
Méthode de libération	Genesis Block Mined	Prévente
Crypto-monnaie utilisée	Bitcoin (Satoshi)	Éther
Algorithme	SHA-256	Ethash
Minage des Blocs	10 minutes	12-14 secondes
Évolutif	Pas encore	Oui

Tableau 1.2 : Comparaison entre Bitcoin et Ethereum

1.16.3 Hyperledger Fabric

Hyperledger Fabric est une blockchain open-source privée, créée par la Linux Foundation, plus précisément par IBM. Contrairement à Bitcoin et Ethereum, Hyperledger Fabric ne fournit pas une monnaie virtuelle. Selon la nature des informations stockées, les transactions peuvent être publiques ou confidentielles. Hyperledger utilise le Practical Byzantine Fault Tolerant (PBFT) comme mécanisme de consensus. Comme expliqué dans [123], PBFT est un mécanisme utilisé dans les réseaux distribués, et qui tolère un certain taux de fautes afin de permettre la continuité des opérations du système. Tous les objets participants se connaissent et sont de confiances, et les objets validateurs sont choisis aléatoirement. Hyperledger Fabric permet également le développement des smart contracts appelé chain codes. [3]

1.17 Défis de la Blockchain

La blockchain est une technologie émergente qui se répand dans divers secteurs et qui présente un grand nombre d'avantages et d'opportunités. Cependant, cette technologie présente son propre ensemble de défis à relever (voir figure 15). Quelques-uns de ces défis majeurs sont abordés dans cette section.

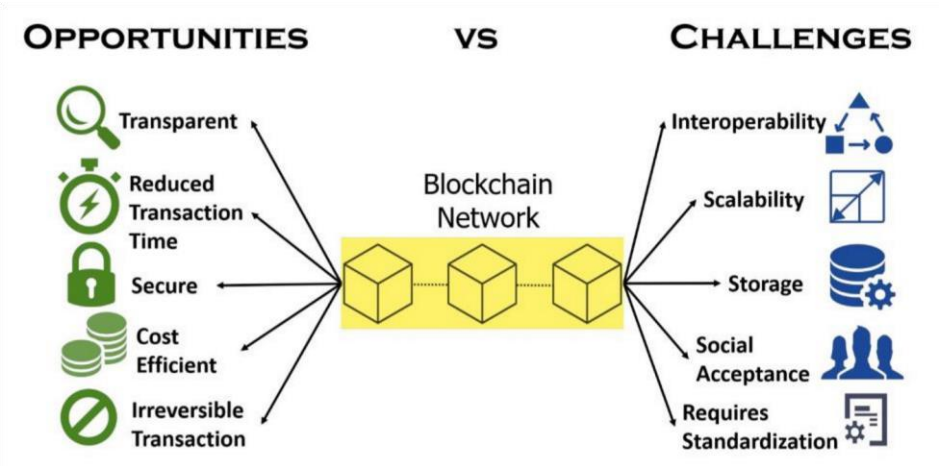


Figure 1.15 : Opportunités et défis des blockchains.

Source: Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives

1.17.1 Sécurité et confidentialité des données

Le premier et le plus important défi concerne la sécurité et la confidentialité des données. Avec la mise en œuvre d'applications basées sur la technologie de la blockchain, la nécessité pour un tiers d'effectuer une transaction est éliminée. Étant donné que le mécanisme de blockchain permet à l'ensemble de la communauté, plutôt qu'à un seul tiers de confiance, de vérifier les enregistrements dans une architecture de blockchain, les données sont exposées à des risques potentiels en matière de sécurité et de confidentialité. Étant donné que tous les nœuds peuvent accéder aux données transmises par un nœud, la confidentialité des données ne sera pas active. En cas d'absence d'une tierce partie pour autorisation, le patient doit sélectionner un ou plusieurs représentants qui peuvent accéder à ses informations et / ou à ses antécédents médicaux en son nom, en cas d'urgence. Désormais, ce représentant peut également autoriser un ensemble de personnes à accéder aux enregistrements du même patient, ce qui peut créer une menace énorme pour la sécurité et la confidentialité des données. L'implication de mécanismes de haute sécurité dans les données entraînera à son tour des obstacles pour le transfert des données d'un bloc à un autre et, par conséquent, les destinataires auront accès à des données limitées ou incomplètes. En outre, les réseaux blockchain sont sujets à une sorte de violation de la sécurité connu sous le nom d'attaque 51%. Cette attaque implique une équipe de mineurs qui possèdent plus de 50% des blocs d'un réseau blockchain.

Les mineurs obtiennent une autorité du réseau et pourraient empêcher toute nouvelle transaction en ne leur donnant pas leur consentement. Cinq cryptomonnaies ont récemment été victimes de cette attaque. En outre, un dossier patient peut contenir des données sensibles qui ne conviennent pas pour figurer dans la chaîne de blocs. [3]

1.17.2 Gestion de la capacité de stockage

Un autre défi qui apparaît sur ce front est la gestion de la capacité de stockage. La Blockchain a été conçue pour enregistrer et traiter les données de transaction, qui ont une portée limitée, de sorte qu'elle n'a pas besoin de beaucoup de stockage. Avec le temps, au fur et à mesure de son expansion dans le domaine de la santé, les défis du stockage devinrent évidents. Le secteur de la santé contient une grande quantité de données qui doivent être traitées quotidiennement. Des dossiers des patients aux antécédents médicaux, en passant par les rapports de test, en passant par les analyses IRM, les rayons X et autres images médicales, toutes les données du scénario de la blockchain seront disponibles pour tous les nœuds de la chaîne, ce qui nécessite un espace de stockage considérable. De plus, les applications de la blockchain étant basées sur des transactions, les bases de données utilisées pour cette technologie ont tendance à se développer rapidement. En raison de la taille croissante des bases de données, la vitesse de recherche et d'accès à l'enregistrement devient lente, ce qui est tout à fait inadéquate pour les types de transactions pour lesquels la rapidité est essentielle. Par conséquent, une solution de chaîne de blocs doit être évolutive et résiliente. [3]

1.17.3 Problèmes d'interopérabilité

La blockchain souffre également du problème de l'interopérabilité, c'est-à-dire que les chaînes de blocs de divers fournisseurs et services de communication communiquent entre elles de manière transparente et appropriée. Ce défi crée des obstacles au partage efficace des données.

1.17.4 Défis de la normalisation

La technologie de la blockchain en est encore à ses balbutiements et elle sera donc certainement confrontée à des problèmes de standardisation en vue de son application pratique en médecine et en soins de santé. Un certain nombre de normes bien authentifiées et certifiées seraient exigées des autorités internationales de normalisation. Ces normes prédéfinies seraient utiles pour évaluer la taille, la nature des données et le format des informations échangées dans

les applications blockchain. Ces normes examineront non seulement les données partagées, mais devront également servir de mesures de sécurité préventives.

1.17.5 Défis sociaux

La technologie des chaînes de blocs évolue toujours et fait donc face à des défis sociaux, tels que le changement de culture, en plus des défis techniques susmentionnés. Accepter et adopter une technologie complètement différente des méthodes de travail traditionnelles n'est jamais chose facile. Bien que l'industrie médicale s'achemine lentement vers la numérisation, il lui reste encore beaucoup à faire pour passer complètement à cette technologie, en particulier celle comme la blockchain, qui n'a pas encore été validée sur le plan clinique. Il faudra du temps et des efforts pour convaincre les médecins de passer de la paperasserie à la technologie. En raison de son faible taux d'adoption dans le secteur de la santé, la technologie et les politiques proposées sont relativement peu fiables. En raison de tous ces défis et menaces, nous ne pouvons pas, à ce jour, le qualifier de solution viable et universelle pour tous les problèmes de santé. [3]

Afin de mieux comprendre, examiner et identifier les forces, faiblesses, opportunités et menaces de la technologie de la chaîne de blocs dans le domaine de la santé, nous avons mené une approche d'analyse SWOT (comme illustré à la figure 1.16).

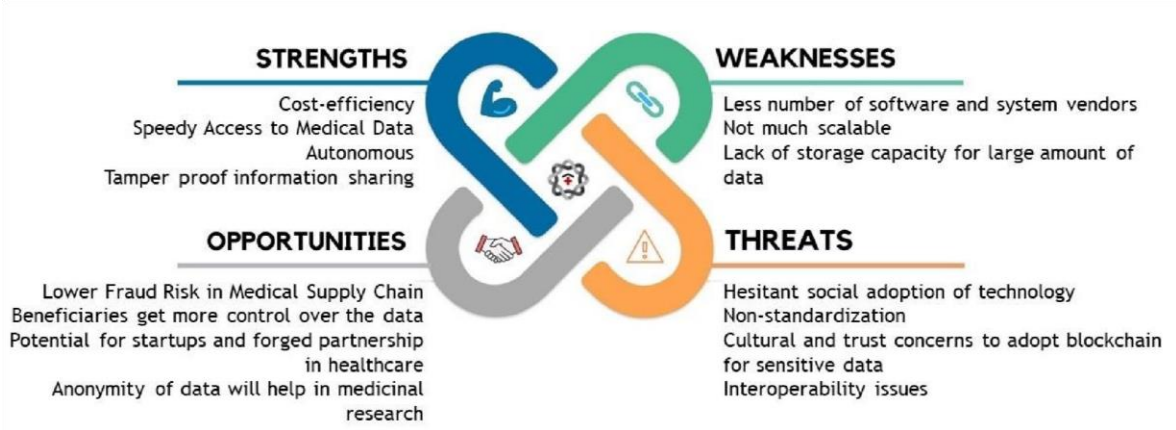


Figure 1.16 : Analyse SWOT pour les blockchains dans les soins de santé.

Source: Applications of Blockchain Technology in Medicine and Healthcare: Challenges and Future Perspectives

1.18 Conclusion

La blockchain présente indéniablement de fortes qualités, et ouvre des perspectives intéressantes à la fois sur le partage de données ainsi que sur la sécurisation de celles-ci. Ses capacités de conservation de l'information (intégrité) paraissent essentielles dans un contexte d'échange croissant des données, tout en étant en contradiction profonde avec le droit à l'oubli et à la rectification, provenant du RGPD.

Elle présente encore des limites technologiques qui nécessitent la recherche active de solutions, permettant de répondre aux réglementations actuelles, et aux défis identifiés. Les principales priorités pour la recherche sont d'améliorer la capacité à monter en charge (scalabilité), la sécurité des systèmes et la fiabilité des applications. Également, un point majeur de discussion et celui de la contradiction entre la protection des données personnelles, supposant un certain anonymat, et la lutte contre les fraudes, nécessitant une forme de transparence.

2 Chapitre II : La chaine d'approvisionnement en medicaments

2.1 Introduction

Souvent décrite comme un outil permettant d'instaurer la confiance entre des acteurs aux intérêts divergents, la blockchain intéresse de nombreux secteurs depuis sa première utilisation en 2008. Initialement destinée à répertorier des transactions entre particuliers, ses applications ont largement évolué au gré des avancées technologies et à l'intérêt grandissant des entreprises internationales. Dans le secteur de la santé, la blockchain intéresse par plusieurs de ses fonctionnalités : son immutabilité qui en fait un excellent support pour authentifier des données sensibles comme des consentements d'essais cliniques, la possibilité d'éditer des smart contracts qui automatisent et facilitent de nombreux processus ou encore la constitution d'un réseau qui se met d'accord sur l'état de l'information. Beaucoup plébiscitée, la blockchain doit néanmoins faire ses preuves dans les conditions réelles d'utilisation et s'inscrire dans un contexte réglementaire et économique particulièrement complexe dans le secteur de la santé. [3]

2.2 Applications des blockchains en santé

Afin de comprendre l'intérêt de la technologie blockchain dans le domaine de la santé, il est important de saisir les caractéristiques qui la différencient des outils actuels de partage et de gestion des données.

Tout d'abord, la blockchain est une technologie décentralisée. Contrairement aux outils classiques qui sont gérés de manière centralisée par un intermédiaire unique, la blockchain est une base de données distribuée entre tous les nœuds du réseau. Ainsi, chaque mineur dispose d'une copie du registre. Cette technologie se prête ainsi au partage d'informations entre plusieurs parties prenantes comme les hôpitaux, les centres de recherche ou les laboratoires pharmaceutiques.

La blockchain permet également de tracer l'origine et le devenir de chacune des données enregistrées. Contrairement à un système centralisé dans lequel un administrateur contrôle ces informations, dans une blockchain, seul le propriétaire peut les enregistrer et les transférer aux autres parties prenantes. De plus, bien que le registre créé par la blockchain soit mis à disposition de tous les membres du réseau, les données qu'il contient sont chiffrées, ce qui permet d'assurer la confidentialité et la sécurité des données sensibles. [3]

Blockchains dans la santé peut être envisagé dans cinq endroits primaires :

- Caractéristiques électroniques de management (EMR) de dossier médical
- Protection des caractéristiques de santé
- Gestion des données personnelle de dossier santé
- management de génomique de Remarque-de-soins
- Gestion des données de dossiers santé de l'électronique [16]

2.2.1 Les blockchains en recherche Clinique

Encore appelé essai thérapeutique ou étude clinique, un essai clinique est une recherche biomédicale organisée et pratiquée sur l'Homme en vue du développement des connaissances biologiques et médicales. Les essais cliniques portant sur les médicaments ont pour objectif, selon le cas, d'établir ou de vérifier un certain nombre de données.

Les caractéristiques de la technologie blockchain font d'elle une technologie ayant un rôle on ne peut plus important dans la certification des essais cliniques. En effet, la blockchain pourrait être utilisée pour faire en sorte que les données soient recueillies et échangées lorsque cela est nécessaire, tout en respectant la vie privée des patients ou les informations exclusives. Les enregistrements immuables appliqués aux essais cliniques, aux protocoles et aux résultats susceptibles d'aboutir à l'horodatage entre autres pourraient solutionner les problèmes de changement de résultat, de « snooping » de données et de rapports sélectifs, réduisant ainsi l'incidence de la fraude et de l'erreur dans les dossiers d'essais cliniques. La blockchain apporte la transparence dans les essais cliniques. L'industrie pharmaceutique pourrait utiliser la blockchain pour authentifier les résultats des essais cliniques. [18]

L'utilisation de la blockchain dans le domaine médical serait d'un apport bénéfique pour les consommateurs que nous sommes indépendamment des continents mais surtout pour les populations africaines en proie au chaos du fait de l'utilisation de médicaments nuisibles pour la santé. En tout état de cause, convient-il de faire cas du fait qu'à l'intérieur de la blockchain sont traitées des données à caractère personnel de santé lesquelles sont qualifiées de données sensibles. Ces données qui sont la source de revenus des grands collecteurs de données personnelles en l'occurrence les GAFAM , pourraient être protégées efficacement via la blockchain. [18]

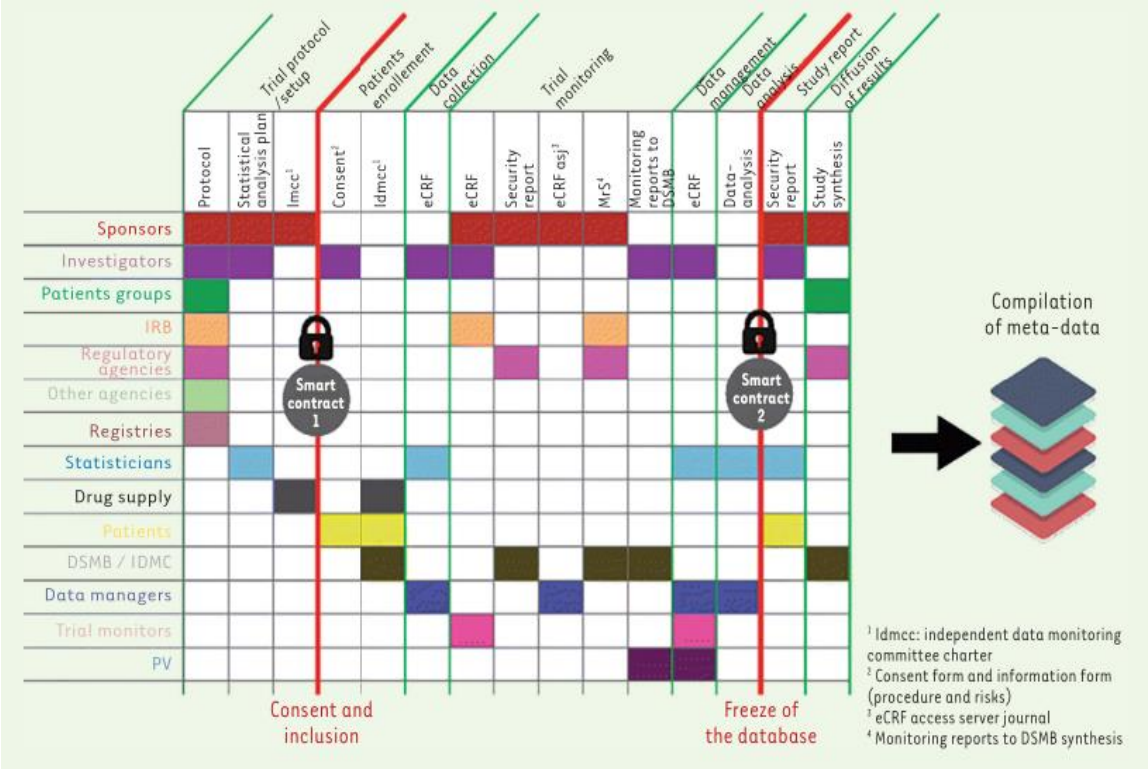


Figure 2.1 Technologie blockchain pour la qualité de la recherche clinique

Figure 2.1. Technologie blockchain pour améliorer la qualité de la recherche clinique. Cette figure indique les sources de données dans un essai clinique ainsi que les parties prenantes qui y ont accès. Deux points sensibles sont identifiés : l'inclusion des patients dépendante du consentement et les rapports fournis suite aux études qui nécessitent que les informations ne soient plus modifiées a posteriori. Ces deux points critiques sont des opportunités d'implémentation de la blockchain. [20]

2.2.2 Blockchains dans la chaîne d'approvisionnement

La blockchain, la technologie sur laquelle reposent les cryptomonnaies, est en voie de simplifier la chaîne d'approvisionnement enrayée par la paperasserie. En suivant digitalement chaque transaction sous une forme qui, en théorie, ne peut être ni falsifiée ni modifiée, la blockchain pourrait supprimer la nécessité d'une chaîne d'approvisionnement sécurisée pour toutes les holdings financières, des banques aux services comptables des entreprises.

La blockchain, une solution qui a considérablement amélioré la visibilité et réduit le temps et les efforts nécessaires.

La blockchain est une base de données distribuée sur les ordinateurs de tous ses membres ou tous les participants de la chaîne d'approvisionnement, si bien que le registre se développe à mesure que le nombre d'utilisateurs augmente.

L'application de la chaîne de blocage à certaines chaînes d'approvisionnement peut profiter au consommateur final, au producteur primaire et à toute autre partie intermédiaire, grâce à une meilleure compréhension de la genèse de tout produit. Les chaînes d'approvisionnement de l'avenir sont prêtes à continuer d'être plus réactives, efficaces et flexibles que leurs contemporaines. Avec les réseaux commerciaux mondiaux en jeu, Blockchain aidera à concrétiser les bonnes voies de transport des marchandises d'un continent à l'autre et d'un producteur à un client final. [23]

2.2.3 Blockchains dans l'industrie pharmaceutique et la recherche

L'industrie pharmaceutique est l'un des secteurs en forte croissance et un secteur de premier plan à la pointe de la prestation des soins de santé. Le secteur pharmaceutique contribue non seulement à l'introduction de médicaments nouveaux et potentiels sur le marché, mais aide également à renforcer la sécurité et la validité des produits médicaux et des médicaments vendus au consommateur final. En outre, le secteur pharmaceutique contribue également à l'évaluation et au traitement de médicaments sûrs, ce qui contribue finalement au rétablissement plus rapide du patient. Dans les cas habituels, les sociétés pharmaceutiques sont confrontées au défi de suivre leurs produits au bon moment, ce qui pose parfois des risques graves en permettant aux contrefacteurs de compromettre la production ou d'invasion de faux médicaments dans le système. En conséquence, la production et la distribution de médicaments contrefaits sont devenues l'un des principaux risques pour la santé au niveau mondial, en particulier dans les pays en développement. Au cours de la production, de la recherche et du développement (R &D) de ces médicaments, la blockchain pourrait être une technologie optimale, qui peut être utilisée pour évaluer, surveiller et garantir les processus de production de médicaments potentiels. Récemment, Hyperledger, a lancé un projet de médicament contrefait utilisant la technologie de la blockchain comme principal outil d'inspection et de lutte contre la production de médicaments contrefaits. En ce qui concerne la fourniture efficace de médicament saut hentiques et fiables aux patients, il est absolument nécessaire de surveiller, évaluer et assurer le processus général de développement et de fourniture de médicaments par le biais de l'utilisation des technologies numériques dans le

monde, et en particulier dans les pays en développement. À cet égard, un système de contrôle numérique des drogues (DDCS) pourrait constituer une solution durable pour la prévention des médicaments contrefaits. En utilisant un DDCS basé sur la blockchain, les grandes industries pharmaceutiques (Sanofi, Pfizer et Amgen) ont lancé un projet pilote commun d'inspection et d'évaluation de nouveaux médicaments. En utilisant la blockchain comme approche, il serait possible non seulement de suivre la production et l'emplacement du médicament, mais également d'améliorer la traçabilité des médicaments, de sécuriser le système d'approvisionnement en médicaments et de garantir la qualité des médicaments fournis aux consommateurs ou aux utilisateurs finaux. [3]

2.2.4 Blockchain et le Dossier de Santé Electronique

Les DSE sont des renseignements centrés sur le patient, en temps réel, qui mettent les informations à la disposition des utilisateurs autorisés de manière instantanée et sécurisée. Si bien le DSE contient les antécédents médicaux et de traitement des patients, un système de DSE est conçu pour aller au-delà de la donnée clinique standard recueillie dans le cabinet médical et peut inclure une vision plus large des soins d'un patient.

Les DSE sont conçus pour partager des informations avec d'autres fournisseurs de soins de santé - laboratoires, spécialistes, centres d'imagerie médicale, pharmacies, établissements d'urgence et cliniques scolaires et professionnelles - afin qu'ils contiennent des informations provenant de tous les cliniciens impliqués dans les soins.

Avec cet outil, nous cherchons à améliorer la qualité des soins, la sécurité des patients et la communication entre les acteurs du système. [17]

Certains de ses principaux avantages:

- Améliorer l'accès à l'information,
- Entrée de données standardisée,
- Systèmes d'alerte en cas de contre-indications, d'interactions ou de surdoses,
- Systèmes d'aide à la décision clinique,
- Faciliter le travail scientifique et statistique,
- Automatiser et rationaliser le flux de travail du fournisseur
- Enregistrement et partage de données sécurisés

2.3 Définition de la contrefaçon et de la falsification de médicaments

La contrefaçon d'un produit (film, produit de luxe, logiciel…) est définie comme une atteinte aux droits de propriété intellectuelle : le trafiquant lèse un créateur, un propriétaire d'un brevet et trompe le client en imitant une marque.

La notion de contrefaçon en tant qu'atteinte à un droit de propriété intellectuelle se distingue de la notion de falsification, qui ne se limite pas à une contrefaçon et qui intègre une problématique de santé publique. Dans ce cas, est évoqué le terme de « faux médicament ».

Dans le cas du médicament, la volonté de protéger la santé de la population et l'évolution rapide du risque ont ainsi entraîné une grande variété de définitions. Cette multiplicité des définitions ne facilite pas toujours la compréhension du problème.

L'Organisation mondiale de la santé (OMS) définit le faux médicament comme une imitation : « Un médicament contrefait est un médicament qui est délibérément et frauduleusement muni d'une étiquette n'indiquant pas son identité et/ou sa source véritable. Il peut s'agir d'une spécialité ou d'un produit générique et, parmi les produits contrefaits, il en est qui contiennent les bons ingrédients ou de mauvais ingrédients, ou bien encore pas de principe actif, et il en est d'autres où le principe actif est en quantité insuffisante ou dont le conditionnement a été falsifié. » [22]

2.4 Définition de la chaîne d'approvisionnement

La chaîne d'approvisionnement, connue également sous le nom de chaîne logistique ou Supply Chain (terme anglais), est constituée d'un ensemble d'activités, méthodes, processus et d'acteurs associés à la transformation des biens depuis les matières premières jusqu'au processus de vente d'un produit fini. L'objectif principal de la chaîne d'approvisionnement est l'optimisation des processus de commande, de production et livraison afin de créer un avantage concurrentiel et répondre au mieux à l'attente ses clients. [13]

La chaîne d'approvisionnement est un segment très complexe, et il est devenu plus difficile d'avoir une visibilité transparente sur l'ensemble de la chaîne d'approvisionnement. Il est devenu plus difficile de suivre le flux de matériel et les canaux de distribution, ce qui a entraîné divers comportements contraires à l'éthique dans les entreprises, allant du commerce illégal aux produits de contrefaçon et aux dommages environnementaux. Tout au bout de la chaîne d'approvisionnement, les consommateurs ne disposent pas des informations selon lesquelles un produit final a été importé tout au long de la chaîne d'approvisionnement. De nos

jours, nous pouvons perdre nos colis par la poste. En tirant parti de la convergence du paradigme IoT et des contrats intelligents, nous pourrons enregistrer la position à tout moment de nos colis grâce à la connexion de capteurs à chaque étape. [3]

2.5 Processus de gestion de la chaîne d'approvisionnement

À la base, **la gestion de la chaîne d'approvisionnement** (GCA) est la gestion du flux des marchandises, des données et des finances qui sont liées à un produit ou à un service. La portée de cette gestion va de l'approvisionnement en matières premières à la livraison du produit, jusqu'à la destination.

Bien que de nombreuses personnes soient responsables de la logistique de la chaîne d'approvisionnement, cette activité n'est qu'un des constituants de la chaîne. Aujourd'hui, les systèmes de gestion de chaîne d'approvisionnement numériques comprennent la manutention des matières, les logiciels pour toutes les parties impliquées dans la création de produits ou de services, l'exécution des commandes et le suivi des informations. Ils font intervenir les fournisseurs, les fabricants, les grossistes, les fournisseurs de service de transport et de logistique et les détaillants.

Les activités de la chaîne d'approvisionnement englobent l'approvisionnement, la gestion du cycle de vie des produits, la planification de la chaîne d'approvisionnement (y compris la planification des stocks et la maintenance des actifs et des chaînes de production de l'entreprise), la logistique (y compris le transport et la gestion de parc) et la gestion des commandes. La GCA peut également s'étendre aux activités liées au commerce mondial, comme la gestion des fournisseurs mondiaux et les processus de production multinationaux. [15]

Une gestion efficace de la chaîne d'approvisionnement exige de nombreux processus liés au flux d'informations, produits et fonds. Ces processus se divisent en trois catégories ouphases, selon de nombreux experts [23].

2.5.1 Stratégie ou conception de la chaîne d'approvisionnement

Cette étape comprend la conception de la chaîne d'approvisionnement par l'entreprise, qui définit la structure de la chaîne d'approvisionnement et des activités qui seront mises en œuvre à chaque étape de la chaîne d'approvisionnement. Il contient des stratégies qui incluent le choix de la production et site de stockage et les capacités de l'installation, et prendre des décisions sur les produits à être faite, ainsi que le choix du moyen de transport et de la source à

partir de laquelle les informations seront collectées. Les décisions de conception de la chaîne d'approvisionnement sont des projets à long terme qui sont coûteux à inverser ; par conséquent, l'incertitude du marché doit être prise en compte. [23]

2.5.2 Planification de la chaîne d'approvisionnement

La planification de la chaîne d'approvisionnement se concentre sur l'établissement de politiques et d'étapes pour les activités promotionnelles, l'inventaire et les politiques de régénération de la production. [24] Essentiellement, il définit les paramètres de la chaîne d'approvisionnement. Cette étape fournit une planification stratégique tournée vers l'avenir avec des perspectives d'avenir. La planification de la chaîne d'approvisionnement traite de l'approvisionnement, de la distribution, de la fabrication, de la planification, de l'ordonnancement de la production, de la planification de la demande, de la prévision et de la coopération dans la chaîne d'approvisionnement, et la conception du réseau de la chaîne d'approvisionnement. [23]

La planification de la chaîne d'approvisionnement coordonne l'application pour améliorer la livraison debiens et services d'information, du fournisseur au consommateur, et pour parvenir à un équilibre entre les engagements d'offre et de demande en temps réel. Planification typique de la chaîne d'approvisionnement les unités de programme comprennent la conception du réseau, la planification du réseau, la planification de la capacité, la demande planification, planification de la fabrication et planification de la planification, de la distribution et du déploiement. [25]

2.5.3 Exécution de la chaîne d'approvisionnement

Les applications d'exécution de la chaîne d'approvisionnement traitent les informations produites par les outils de planification de la chaîne d'approvisionnement pour guider les politiques de renouvellement des stocks et de la production. Sur d'autre part, il comprend des activités visant à acheter et à équilibrer efficacement l'offre de biens et matériaux. [25]

L'exécution de la chaine d'approvisionnement se concentre sur les applications orientées, y compris la gestion des commandes, la gestion des stocks, la gestion des entrepôts, la gestion des transports et la gestion logistique qui inclut toutes les parties. [25] L'objectif de la chaîne d'approvisionnement et que les opérations d'exécution sont de traiter les demandes des clients entrants de la meilleure façon possible.

2.5.3.1 Gestion des entrepôts

La gestion d'entrepôt est des implémentations qui gèrent les processus d'un entrepôt ou centre de distribution. Qui comprend la réception, le stockage, l'inventaire, expédition des marchandises et matières premières, distribution des commandes, reconditionnement, emballage, gestion des travaux. L'utilisation conjointe de la technologie des radio fréquences avec le codage à barres, la RFID ou d'autres technologies de collecte de données peuvent aider à améliorer l'efficacité des systèmes de gestion des entrepôts, fournissant des informations précises en temps réel. [25]

2.5.3.2 Gestion des transports

La gestion du transport est utilisée pour gérer toutes les activités de fret à travers l'organisation. Cela comprend également la planification, l'exécution et l'optimisation de la circulation des marchandises. [25] La fonction principale du système de gestion du transport consiste à aider l'utilisateur à trouver la meilleure position et le meilleur prix pour tout type d'expédition afin de s'assurer qu'ils obtiennent les meilleures offres.

2.5.3.3 Gestion de la fabrication

Le processus de fabrication comprend la planification de la production / l'ordonnancement détaillé, qui ce processus prend en charge le processus d'affectation des ordres de fabrication aux approvisionnements dans un ordre et dans un délai précis. Et l'exécution de la fabrication est prise en charge par les processus. Le processus de capture des informations de production réelles de la boutique à soutenir le contrôle de la production et les processus d'établissement des coûts. [26] Il peut également inclure le contrôle des documents, la gestion du travail, la gestion de la qualité, les processus et la maintenance gestion. [25]

2.5.3.4 Gestion des achats

L'équipe d'approvisionnement est chargée d'obtenir la ou les meilleures sources d'approvisionnement. La gestion des achats comprend trois processus, à savoir: Premièrement, achat traitement des commandes, il exécute les exigences d'approvisionnement direct via l'approvisionnement, l'émission et la confirmation des bons de commande. Deuxièmement, la confirmation de réception le traitement de la formation informe les autres départements des informations reçues et confirmées quantité de marchandises commandées. Troisièmement, le processus de vérification des factures reçoit, entre et vérifie l'exactitude de la facture du fournisseur. [26]

2.5.3.5 Gestion des commandes

La gestion des commandes est le processus de suivi des commandes clients et de leur exécution effectivement. Il comprend la collecte de données, le traitement des commandes, y compris la carte de crédit vérification. Cela comprend également la tenue d'un enregistrement du client, ce qui peut inclure le dossier d'achat, le mode de paiement et la taille de la commande. Les services commerciaux notifient l'entrepôt pour exécuter la commande, puis la commande est expédiée au client. [27]

2.5.3.6 Gestion de la logistique

La gestion de la logistique est une composante de la gestion de la chaîne d'approvisionnement qui est utilisée pour répondre aux demandes des clients grâce à une planification, un contrôle et une exécution efficaces du trafique et le stockage des informations, des biens et des services de la construction pointez sur la destination finale. La gestion logistique aide les entreprises à minimiser coûts et améliorer les services à la clientèle. [28]

2.6 Chaîne d'approvisionnement en médicaments.

La chaîne logistique pharmaceutique doit permettre de mettre à disposition des patients le plus efficacement possible les produits pharmaceutiques qui leur seront administrés, dans des conditions garantissant sécurité et traçabilité tout en respectant les nombreuses réglementations entourant les produits pharmaceutiques et leur dispensation. La figure 2.2 illustre cette chaîne [BERETZ, 2002] et permet de mettre en évidence ces spécificités. [19]

2.6.1 Les fournisseurs

Le marché des produits pharmaceutiques est dominé par quelques grands groupes fournissant une grande variété de références. L'attribution des marchés pour la fourniture des médicaments est réglementée par le code des marchés publics, ce qui restreint la liberté de négociations avec ceux-ci. [19]

2.6.2 La pharmacie générale et ses stocks

Les activités et responsabilités de la pharmacie hospitalière sont également définies par la loi ainsi que les conditions de délivrance et de remboursement des produits pharmaceutiques. En outre, la gestion des stocks de la pharmacie est rendue complexe par le nombre de produits et l'hétérogénéité des données logistiques, les volumes et conditionnement

divers, les conditions de stockage spécifiques (frigo, endroits sécurisés pour les narcotiques, espaces stériles,…), la gestion des dates de péremption,….

2.6.3 Les stocks avancés

Chaque unité de soins et unité médico-technique dispose d'un stock de produits pharmaceutiques, géré localement par les infirmières. La gestion de ces stocks pour un même produit est différente dans les unités de soins ou dans les services médico-techniques (remplissage, facturation,…).

2.6.4 Le processus de soins

Tout comme la gestion des stocks, le processus de soins est lui aussi différent en fonction des unités de soins et des unités médico-techniques. Par exemple, la délivrance de bandage facturable sera effectuée dans une unité de soins sur base d'une ordonnance et sera facturé à priori, tandis que dans une unité médico-technique, l'ordonnance et la facturation seront effectuées à posteriori. En outre, le processus de soins qui conditionne la demande de produits pharmaceutiques est fortement marqué par l'aléatoire et la prépondérance du facteur humain.

2.6.5 Les flux d'information

Les flux d'information qui entourent la chaîne logistique pharmaceutique sont nombreux et complexes. Ils doivent indiquer quel médicament prescrire à quel patient, doivent assurer la traçabilité des produits pharmaceutiques administrés et leur facturation, ils doivent également permettre le remboursement auprès des organismes assureurs, assurer un retour des informations pour une assistance pharmaceutique sur l'administration des médicaments. [19]

2.6.6 Les acteurs

Les acteurs intervenant tout au long de la chaîne sont nombreux et doivent avoir une double compétence (technique et médicale), ce qui a comme conséquence la gestion indépendante des flux pharmaceutiques par rapport aux autres flux logistiques de l'hôpital. [19]

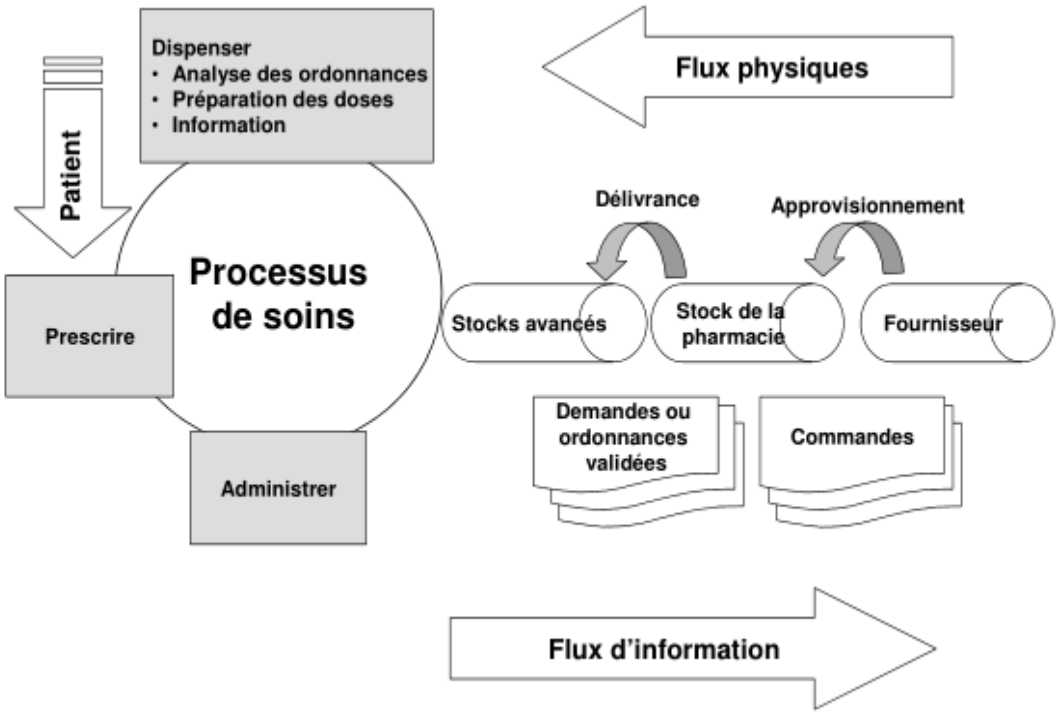


Figure2. 2. Description de la chaîne logistique pharmaceutique [BERETZ, 2002]

2.7 Système de chaîne d'approvisionnement en médicaments

L'objectif principal de la gestion de la distribution est de maintenir un approvisionnement stable des produits pharmaceutiques et des fournitures aux établissements de la manière la plus efficace. Le cycle de distribution commence lorsque les produits pharmaceutiques sont envoyés par les fabricants ou ressources. Il se termine lorsque les informations sur la consommation de médicaments sont communiquées à l'unité d'achat [29].

La figure 2.3 illustre le système de distribution pharmaceutique typique et ses interactions avec les secteurs public et privé à différents niveaux. Le cycle de distribution dans le domaine pharmaceutique comporte neuf activités principales qui sont résumé dans la figure 2.4 et nous expliquerons chaque activité en détail.

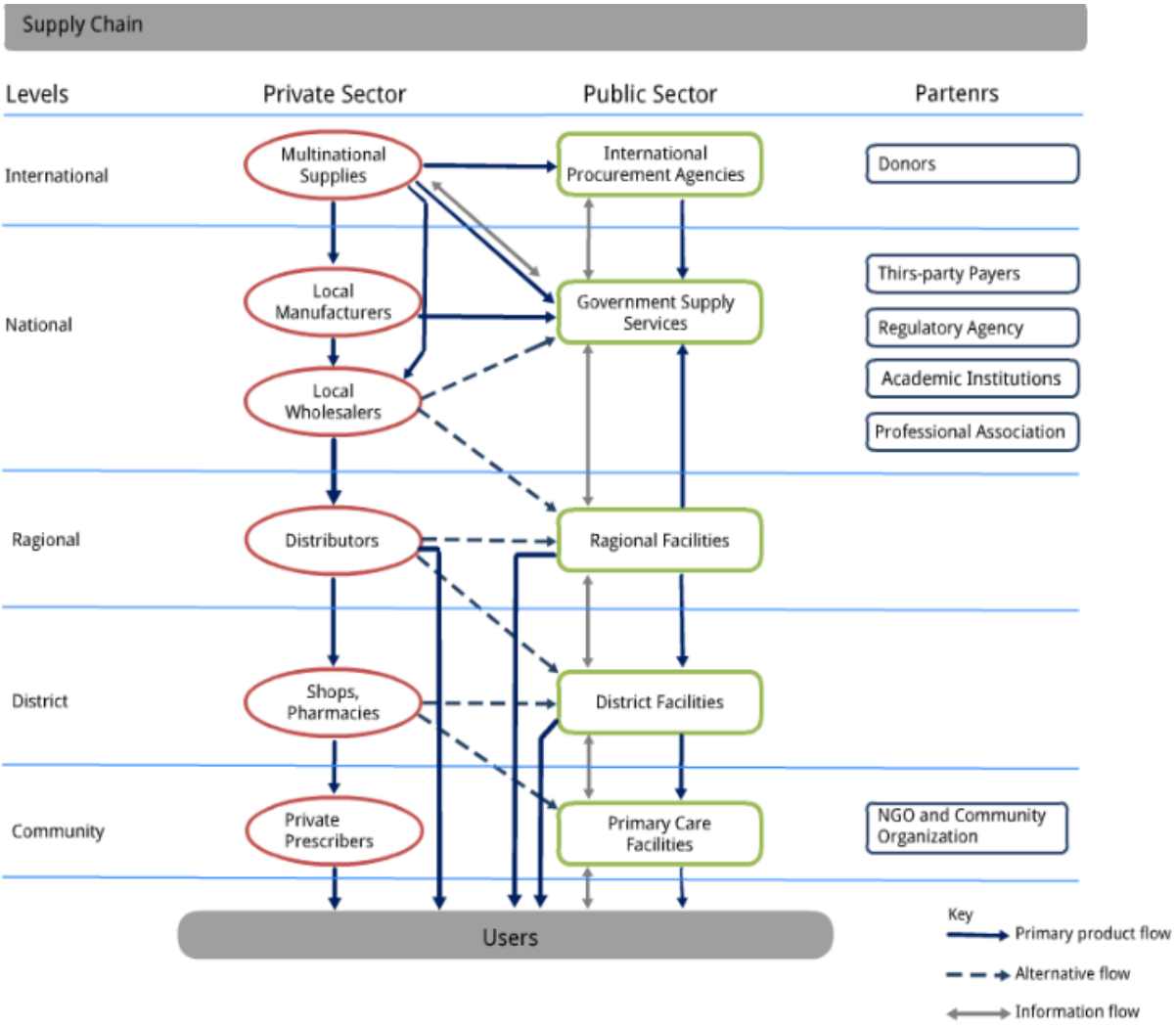


Figure 2.3 : Système de distribution pharmaceutique

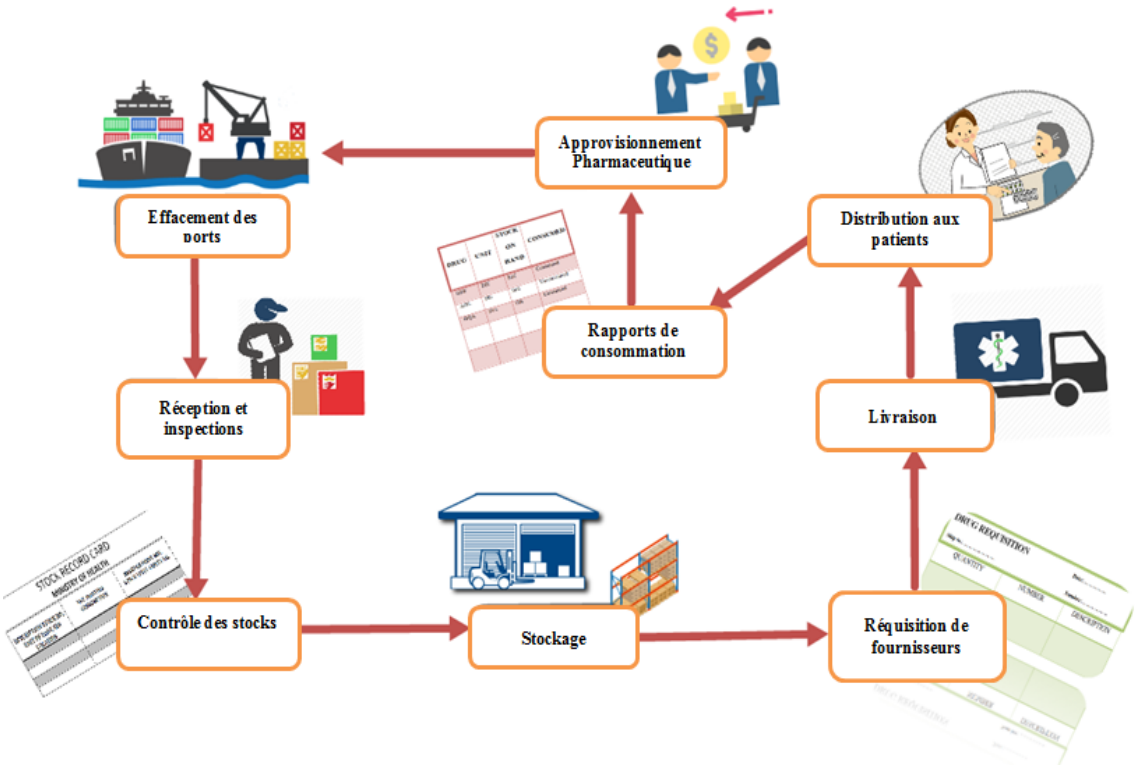


Figure 2.4 : Cycle de distribution pharmaceutique.

2.7.1 Approvisionnement pharmaceutique

Le processus d'achat fait partie du cycle de gestion des médicaments. Cela va de pair avec sélection des médicaments, quantification des besoins en médicaments, stockage et distribution / approvisionnement.

L'approvisionnement est l'acquisition de biens et / ou de services au mieux le coût de possession, de qualité et quantité appropriées, au bon moment, au bon endroit et à partir de la source correcte des avantages directs, ou de l'utilisation du personnel de l'entreprise, ou même les gouvernements.

Le système d'achat de produits pharmaceutiques est l'un des principaux déterminants de la disponibilité des médicaments et du coût total des médicaments. L'achat de médicaments est un processus complexe qui implique de nombreuses étapes, décisions, actions, agences, ministères et fabricants qui déterminent les quantités spécifiques de médicaments obtenues, les prix payés et la qualité des médicaments reçus. [29]

2.7.2 Effacement des ports

Le processus de compensation portuaire est vital pour le processus efficace d'une entreprise pharmaceutique publique programmes d'approvisionnement, qu'ils soient exécutés par des fonctionnaires ou sous-traités.

Là est un système informatisé de surveillance des activités de dédouanement qui sont : [29]

- Gérer les cas avant expédition, tels que la documentation, qui sont souvent nécessaires pour le processus de dédouanement.
- Identifier et anticiper l'arrivée des envois dès leur arrivée Port.
- Stocker correctement les médicaments jusqu'à ce qu'ils quittent le port.
- Localisation des envois et des envois particuliers.
- Obtenir les documents nécessaires au dédouanement avant l'arrivée des fournitures à un port, et en veillant à ce que les documents soient conformes au port du pays et les exigences douanières.
- Effectuer des paiements en temps opportun dans le cadre du processus de dédouanement.

2.7.3 Reception et inspections

La gestion des arrivées garantit que les marchandises reçues sont stockées dans un entrepôt qui comprend la réception et les procédures d'inspection. Pendant le processus entrant, doit porter une inspection complète de chaque envoi dès sa réception du port ou fournisseur local. Vérifier également les articles endommagés et manquants et le type de médicament, la quantité, l'emballage, la présentation, l'étiquetage, etc. Inspection rapide et exacte des toutes les expéditions sont fondamentales pour garantir que les fournisseurs remplissent leurs contrats. Les processus d'expédition et de réception nécessitent plusieurs autres documents importants qui peuvent également être électronique, y compris le bon de livraison du matériel, le connaissement et la réception rapport d'anomalie. [29]

2.7.4 Contrôle des stocks

Le contrôle des stocks est le processus utilisé pour maximiser l'utilisation des stocks par une entreprise.

L'objectif du contrôle des stocks est de produire le maximum de gain à partir du montant minimal d'investissement en stocks sans empiéter sur les niveaux de satisfaction des clients.

Compte tenu de l'impact sur les clients et les gains, le contrôle des stocks est l'une des principales préoccupations des entreprises qui ont de gros investissements en stocks, comme les détaillants et les distributeurs. Le contrôle des stocks ouvre la voie à une distribution rentable et réactive système. [29]

Cela comprend :

- C'est un système de distribution et une protection primaire contre le vol et la corruption.
- Intégration du lecteur de codes-barres.
- Réorganiser les rapports du solde de consommation et des ajustements.
- Détails du produit, historiques et emplacements.
- Listes et dénombrements complets des inventaires.
- Synchronisation du stock disponible avec les bons de commande et les bons de commande.

2.7.5 Stockage

Le stockage dans un entrepôt laisse le temps aux produits pharmaceutiques d'être testés avant être mis sur le marché. Un tel entrepôt nécessite une sécurité adéquate pour refuser le vol ou la transformation des expéditions pharmaceutiques et devrait également avoir la capacité de conserver les médicaments dans les bonnes conditions, y compris la température appropriée, pour la qualité des médicaments, minimiser les vols et les pertes par dommages. [29]

2.7.6 Réquisition de fournisseurs

Les modèles et procédures de réquisition constituent une partie essentielle du contrôle des stocks système. Ils peuvent varier d'une nation à l'autre. Le système de réquisition peut être manuel ou informatisé ou les deux, mais il doit toujours être conçu pour simplifier la distribution en facilitant le contrôle des stocks, en offrant un parcours d'audit pour suivre le flux des médicaments, aider à la comptabilité financière et répertorier les médicaments émis. [29]

2.7.7 Livraison

Les médicaments peuvent être livrés par entrepôt ou collectés par le personnel de l'établissement de santé. Le transport peut inclure les transports aériens, fluviaux, ferroviaires, etc. Des choix rentables entre les transports publics et privés doivent être faits. Les

gestionnaires de transport doivent choisir soigneusement les itinéraires de transport et planifier les livraisons pour offrir un service ponctuel et économique. [29]

2.7.8 Distribution aux patients

La délivrance est l'une des clés vitales de l'usage rationnel des médicaments. Le processus de distribution s'abstient de son objectif lorsque les médicaments arrivent dans les services hospitaliers, les cliniques ambulatoires, les centres de santé ou les agents de santé communautaires, puis sont donnés aux patients par le biais de cabinets médicaux. La distribution de médicaments au point de service est devenue un moyen sûr, efficace et rentable d'aider les patients à gérer leurs programmes de traitement. [29]

2.7.9 Rapports de consommation

Le flux d'informations sur la consommation et les soldes des stocks est un dernier maillon du cycle de distribution, vers le bureau des achats, à utiliser pour déterminer les besoins en matière d'achats.

Lorsque des registres d'inventaire et de demande suffisants sont conservés, les rapports de consommation sont directement agrégés. [29]

2.8 Chaîne d'approvisionnement en médicaments en Algérie :

Le marché pharmaceutique en Algérie connaît la même chose que les pays en développement.

Et cela provient de l'importation de la plupart des médicaments consommés. Les médicaments sont de marque et importés. La plupart des importations de produits pharmaceutiques proviennent d'Europe, des États-Unis et des pays du Moyen-Orient. Néanmoins, le gouvernement algérien tente de développer l'industrie pharmaceutique locale. Les entreprises pharmaceutiques locales peuvent être classées en deux groupes :

- Le premier groupe consiste à importer les produits finaux et à les distribuer au local marché via des sociétés de distribution privées ou des sociétés affiliées.
- Le deuxième groupe se compose de fabricants locaux qui fabriquent soit pour eux-mêmes ou pour d'autres entreprises.

L'industrie pharmaceutique algérienne s'est fixé comme objectif de changer et d'améliorer les techniques pour garantir l'investissement intérieur et extérieur, dans le but de garantir une couverture du marché de la production nationale jusqu'à 70% en 2014. [29]

Le secteur en Algérie a contribué à des taux de croissance significatifs. En outre, le Bureau de la santé a mis en place un nouveau système d'approvisionnement en produits pharmaceutiques aux institutions publiques afin de garantir la disponibilité des produits pharmaceutiques. Cette technique s'ajoute aux mesures effectivement prises par le gouvernement pour assainir le champ de la distribution des médicaments, et pour améliorer et redévelopper la gestion des produits critiques. [29]

2.8.1 Système de distribution en Algérie (Secteur public et Secteur privé)

Le secteur pharmaceutique en Algérie s'est développé avec le système de santé en s'adapter progressivement à l'évolution du niveau de la demande nationale de produits pharmaceutiques des produits.

L'institution de l'installation reflète une tendance plus large à accroître les investissements dans l'industrie pharmaceutique locale. En fait, le pays est devenu le foyer du la plus grande installation de production et de distribution de médicaments du continent en octobre 2018, avec l'ouverture de plusieurs complexes. La figure 2.5 décrit la distribution pharmaceutique système en Algérie et ses interactions avec les secteurs public et privé dans divers les niveaux.

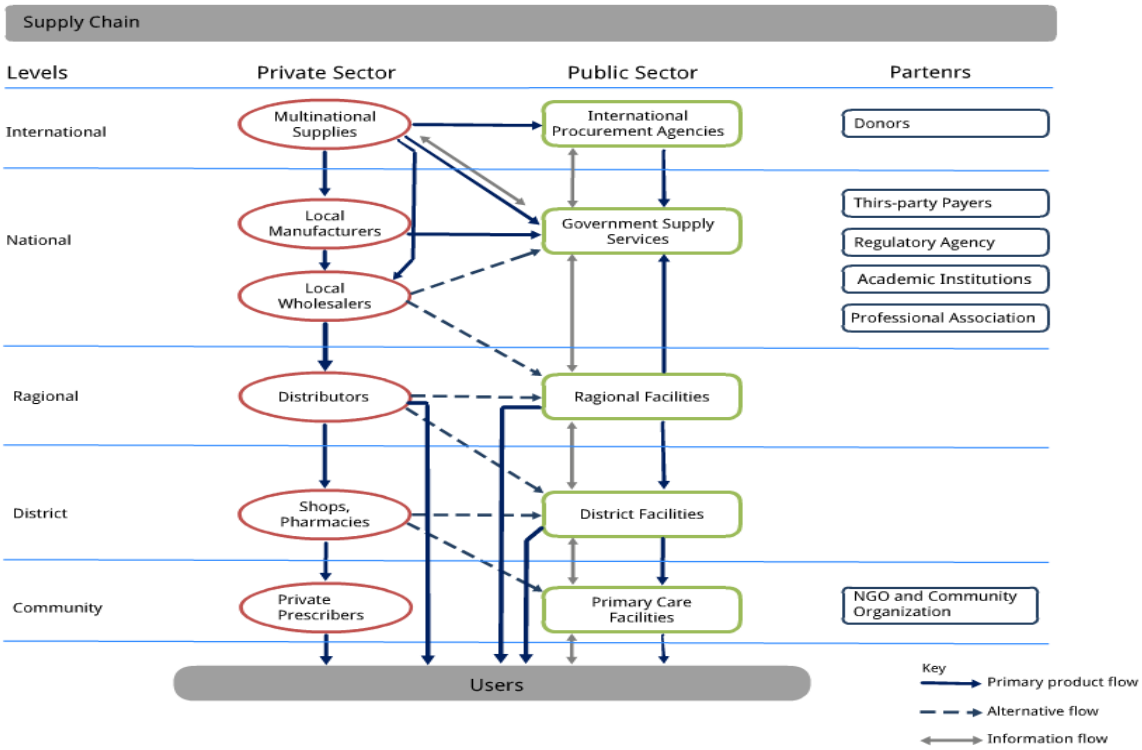


Figure 2.5: Système de distribution pharmaceutique.

2.8.1.1 Secteur public

Les acteurs du secteur pharmaceutique général s'appuient sur une séparation fonctionnelle entre les divers organismes à travers la production, l'importation, le commerce de gros et de détail, ainsi que les pharmacies et les hôpitaux.

1. Organisme de production

Représenté par **SAIDAL**, société publique créée en 1982 et a commencé à acquérir les anciennes unités de production de la pharmacie centrale en Algérie.

La compagnie nationale des médicaments est gérée comme une entreprise privée avec une autonomie administrative bien qu'elle soit détenue à 80% par l'État. Sa double fonction est consolider sa position de leader en tant qu'entreprise publique de fabrication locale et consolider la cause de la politique pharmaceutique nationale mise en œuvre par le gouvernement en tant qu'actionnaire majoritaire.

2. Distribution en gros

Représentée par **DIGROMED** qui, depuis 1997, détenait un réseau d'anciennes sociétés d'importation publiques. Et il a été assigné la tâche de distribuer en vrac et la moitié du matériel pharmaceutique. Pour plusieurs années, **DIGROMED** a commencé à diversifier son activité en fabriquant des médicaments génériques [31] avant de se dissoudre en 2009. [32]

3. Distribution au détail

Représentée par **ENDIMED**, qui s'occupe de la l'exploitation du réseau de distribution au détail de produits pharmaceutiques et le nancement des agences publiques. Ce réseau, qui représentait près d'un millier pharmacies, a été dissoute sur le territoire national et les pharmacies ont été autorisé à réhabiliter les pharmaciens [32] Distribution au détail: représentée par ENDIMED, qui s'occupe de la l'exploitation du réseau de distribution au détail de produits pharmaceutiques et le nancement des agences publiques. Ce réseau, qui représentait près d'un millier pharmacies, a été dissoute sur le territoire national et les pharmacies ont été autorisées à réhabiliter les pharmaciens [32].

4. Pharmacies publiques

Chargée de fournir une liste gratuite des médicaments financés par la Caisse nationale d'assurance sociale (CNAS) pour les personnes défavorisées et chroniques les personnes malades avec des rendements très faibles.

5. Pharmacie Centrale Hôpital (PCH)

Est une fondation publique d'une entreprise et caractère commercial mis en œuvre pour coordonner et rationaliser l'hôpital public programmes d'approvisionnement.

2.8.1.2 Secteur privé

Les acteurs du secteur privé ont assisté à un développement rapide et significatif de l'importation et distribution de médicaments (en grandes quantités). De 25 pour cent en 2008 à 65 pour cent en 2018. L'Algérie est autorisée à réduire sa dépendance à l'égard des importations. [29] Représentée par :

1. Fabricants privés, importateurs et pharmacies de détail privées

De 1963 à 1990, le marché pharmaceutique était complètement sous contrôler. Depuis, la suppression du monopole national a permis à l'émergence d'entreprises du secteur privé dans l'importation, la production et la distribution.

2. Grossistes-distributeurs privés

Ils sont responsables de la vente en gros fourniture de diverses pharmacies de détail dans la région nationale. Leur activité est contrôlée par le décret n ° 59 / MSP du 20 juillet 1995 qui fixe les conditions d'activité pour la distribution en gros de produits pharmaceutiques. Celles-ci les grossistes-distributeurs ont une fonction réglementaire fondamentale et représentent des moyens importants de transmission d'informations économiques sur le marché, produits et les habitudes de consommation. [32]

2.9 Défis de la chaîne d'approvisionnement en médicaments

La chaîne d'approvisionnement en médicaments s'étend des sociétés de fabrication, d'importation et de production de produits chimiques bruts et de composants biologiques à la formation d'un médicament pour les patients et les consommateurs. Tout cela est dans les coulisses de l'expédition, de la réception et du support une industrie avec des clients qui peuvent rarement supporter une interruption de l'approvisionnement.

L'approche actuelle de la chaîne d'approvisionnement pour le transport et la distribution des médicaments comporte des défis ou des limites uniques. En général, certaines des principales limitations sont rencontrées dans les applications actuelles de la chaîne d'approvisionnement. [29] nous allons illustrer cela au-dessous de :

- **Manque de transparence** : le manque de transparence met toujours une réputation en péril. Le plus souvent, les participants à la chaîne d'approvisionnement gèrent leurs

propres données en utilisant des bases de données traditionnelles qui ne fournissent pas de transparence des données par défaut, ce qui rend un système de données géré séparément difficile pour vérifier comment les articles sont traités à chaque fois. Étape de la chaîne d'approvisionnement.

- **Manque de traçabilité**: en raison de la transparence limitée des étapes de la chaîne d'approvisionnement, un défi majeur de la traçabilité est l'ambiguïté des informations sur les produits, qui résulte de l'enregistrement de caractéristiques de produit ambiguës et non confirmées qui sont difficiles à retracer, ce qui rend l'originalité et les exigences d'authenticité du produit sont difficile ou impossibles.
- **Méfiance des parties prenantes** : Faites confiance à toute chaîne d'approvisionnement nécessaire pour partager des informations critiques telles que les coûts, les prix, etc. La méfiance des participants est le principal obstacle à l'amélioration des réseaux de la chaîne d'approvisionnement. En s'appuyant sur le système central et des intermédiaires tiers comme agents de confiance et en vérifiant les transactions et les services, cela augmente considérablement le coût opérationnel et réduit l'efficacité du processus. [30]

2.10 Chaîne d'approvisionnement en médicaments dans blockchain.

La chaine d'approvisionnement pharmaceutique est aujourd'hui en proie à de nombreux défis de traçabilité des médicaments et de lutte contre la fraude. L'Organisation Mondiale de la Santé (OMS) estime que 10 à 30% des médicaments en circulation dans les pays en développement sont en réalité de faux médicaments, ce qui entraînerait la mort de près de **700 000** personnes chaque année.

L'utilisation d'une blockchain, registre transparent et inaltérable, pourrait aider à lutter contre ce fléau, en enregistrant les empreintes de chaque étape de la chaîne de fabrication et distribution d'un médicament. Tous les acteurs de la supply chain pharmaceutique, ainsi que les patients, pourraient alors directement vérifier la provenance et l'intégrité des médicaments.

Des startups comme Chronicled (installée à San Francisco) ou BlockPharma (en France) ont déjà commencé des expérimentations sur ce sujet. Chornicled a par exemple lancé ennovembre 2016 le projet CryptoSeal, un projet alliant la technologie NFC (qui fait le lien entre monde physique et monde numérique)et la blockchain. L'idée ici est d'utiliser une puce NFC pour contenir des données d'authentification au préalable enregistrées sur une

blockchain. Cette puce, apposée sur les boîtes de médicaments, doivent permettre un suivi particulièrement fiable. [9]

2.10.1 Traçabilité et lutte contre la fraude

Outre ses applications pour les données patientes, la blockchain, grâce à sa transparence et son inaltérabilité, peut également être utilisée en tant qu'outil de traçabilité et de vérification d'authenticité pour les médicaments, les ordonnances médicales ou encore les brevets. Les laboratoires pharmaceutiques pourraient ainsi en bénéficier dans leurs problématiques de contrefaçons de médicaments L'utilisation d'une blockchain pourrait aider à lutter contre ce fléau, en enregistrant les empreintes de chaque action liée à un médicament, lors des différentes phases du processus de fabrication et distribution. Tous les acteurs de la supply chain pharmaceutique pourraient alors vérifier la provenance et l'intégrité des médicaments.

De la même façon, il est envisageable d'utiliser une blockchain pour stocker les preuves d'existence de documents telles que les ordonnances médicales, qui sont aujourd'hui sujettes à des fraudes. Grâce à la blockchain, les pharmaciens pourraient ainsi vérifier l'authenticité des ordonnances qui leur sont présentées. [14]

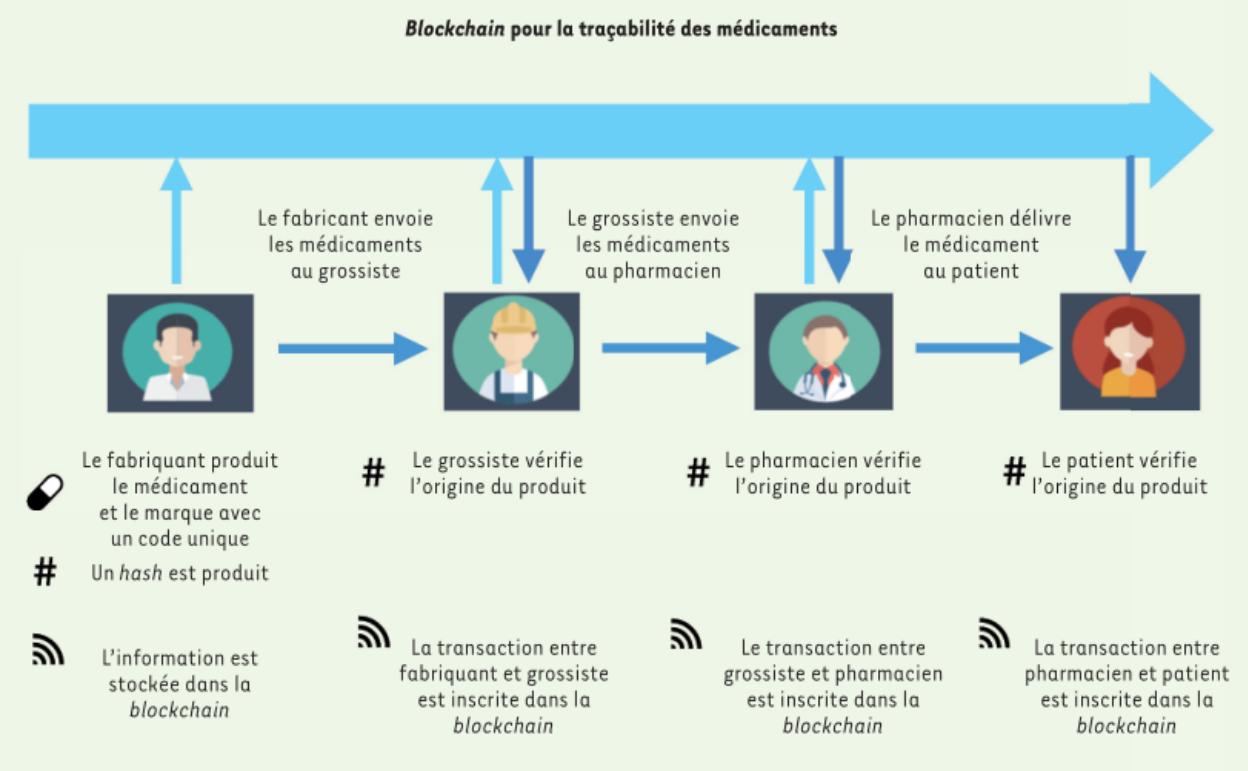


Figure 2.6 Technologie blockchain pour la traçabilité des médicaments.

Cette figure retrace le parcours du médicament depuis le fabricant jusqu'au patient. Chaque interaction d'un maillon de la chaîne de distribution avec le produit génère une empreinte numérique (appelée hash) qui est stockée dans la blockchain. [20]

2.10.2 Gestion de la supply chain

Outre une plus grande transparence, la blockchain peut également apporter de la fluidité à la supply chain du secteur pharmaceutique. Aujourd'hui celle-ci est contrainte par un nombre important de procédures administratives. La blockchain pourrait réduire ces contraintes en rassemblant les participants de la supply chain au sein d'un registre distribué. Chaque étape logistique pourrait être rendue visible en temps réel à l'ensemble des participants, de façon à permettre une gestion plus fluide, plus transparente et en temps réel de la supply chain.

Il est également envisageable que certains paiements voire certaines tâches administratives soient automatisées grâce à des smart contracts. [14]

2.11 Système de chaîne d'approvisionnement en médicaments dans Blockchain:

La blockchain permet à tous les acteurs de suivre l'avancée du médicament dans la chaîne de distribution, il permet donc à ces mêmes acteurs d'avoir des informations sur les ventes en temps réelle. Le laboratoire peut avoir une vision sur les stocks de ces produits au niveau régional, national et même mondial, et par conséquent anticiper les phases de production. La visibilité sur les ventes en temps réel permet à tous les acteurs, que ce soit les laboratoires, les fournisseurs de matières premières ou les grossistes, d'anticiper sur les commandes, la fabrication, les besoins de leurs partenaires. Devant autant de points importants pour lutter contre les ruptures, cette vision, jusqu'au client final, permet d'ajuster plus finement le circuit de distribution, et donc de l'optimiser au maximum. On peut même automatiser certaines actions en utilisant des smart contracts. Pour cela, on fixe des conditions, comme par exemple un certain stock disponible en Europe pour lancer une nouvelle phase de production. La blockchain permet de partager toutes les informations avec des détails masqués, d'autres visibles, des visions macroscopiques et d'autres microscopiques. C'est un outil qui s'adapte aux besoins de chacun des acteurs et dont on peut tirer de nombreux avantages. [21]

Les médicaments n'étant pas des produits comme les autres, la chaine logistique qui les entoure revêt une importance critique. Deux choses doivent à tout prix être respectées. D'une

part, s'assurer qu'aucun médicament non légitime ne puisse s'immiscer dans la chaine logistique se faisant passer pour un produit approuvé. D'autre part, pouvoir contrôler, pendant tout le cheminement du produit les standards qualité, comme la température ou l'humidité. Il est primordial que depuis l'acheminement des matières premières au fabricant jusqu'à la délivrance du médicament au patient les produits n'aient subi aucun dommage durant leur transport. [9]

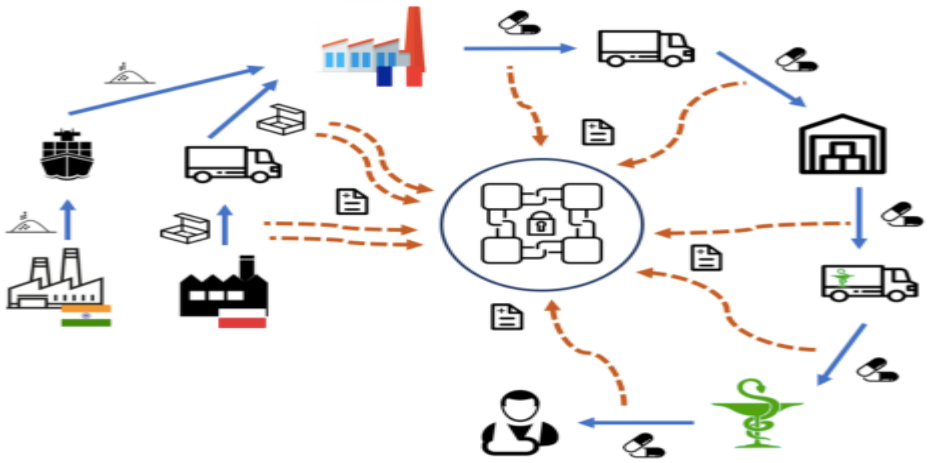


Figure 2.7 : chaîne logistique appuyée par une blockchain

Ici les informations sont stockées sur une blockchain partagée par tous les acteurs de la chaîne logistique. Ceci permet de disposer d'une base de données complète et à jour sur un produit ayant traversé cette cascade d'évènements. En ouvrant l'accès à une entité d'audit, la vérification de toutes les étapes est rapide et facile. [9]

2.12 Comment la blockchain peut participer à l'intégrité de la chaîne Logistique

Historiquement chaque acteur de la chaine logistique stocke ses informations et donc sa version des faits sur l'historique des produits. Or il serait bien plus fructueux de stocker l'information en un lieu unique, source de vérité commune sur l'historique du cheminement des médicaments ou de leurs précurseurs. [9]

Jusqu'ici, disposer d'une base de données commune entre plusieurs entreprises ne se faisant pas nécessairement confiance était presque impensable, cependant ce n'est plus vrai avec la technologie blockchain. Grâce à cette dernière il devient possible de stocker des données sensibles sans crainte qu'elles ne soient accessibles par quelqu'un d'autre.

Ce sont les blockchain de type consortium qui est le plus à même de pouvoir aider à gagner en transparence dans la chaine logistique. Pour rappel, dans ce type de blockchain, uniquement des acteurs approuvés peuvent apporter des changements aux informations contenues dans la blockchain.

Pour ce faire, chaque acteur de la chaîne logistique dispose d'une clé privée qui lui permettra de s'authentifier comme entité apte à pouvoir proposer des informations. Ce modèle est représenté sur la figure 2.7. Au lieu que chaque acteur stocke les informations relatives au produit dans son système d'information, il va la communiquer dans la blockchain. Une fois les données dans la blockchain il est possible de décider l'étendue que l'on veut partager avec les autres acteurs du consortium.

Pour pouvoir suivre informatiquement des produits dans un système blockchain ou non il faut tout d'abord les digitaliser. Il est donc impératif de pouvoir faire un lien entre le monde digital, à savoir la trace informatique et le monde physique, à savoir les produits que l'on veut tracer. [9]

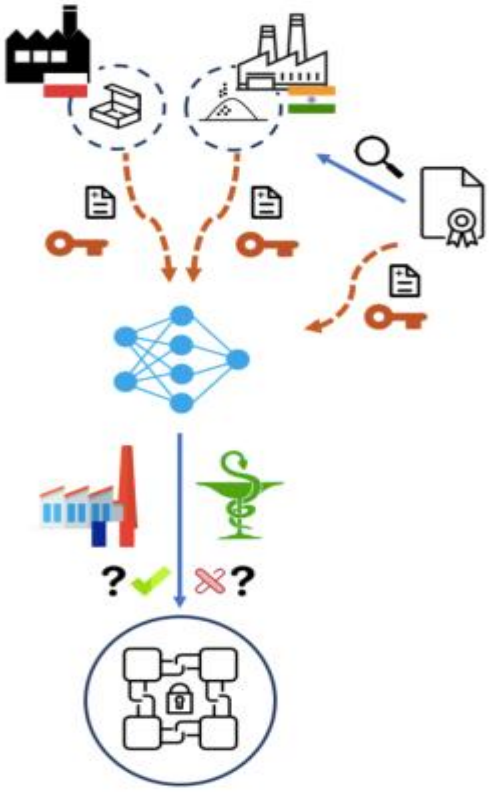


Figure 2.8 : ajout d'un actif dans la blockchain soumis à un consensus [9]

Lors de la première étape les actifs sont créés de toute pièce dans la blockchain. Cette étape étant critique, il est possible de demander à un tiers externe de la valider. Ces actifs sont créés sous la forme d'un contrat intelligent qui régira les règles de transformation du produit par la suite. [9]

2.12.1 Création du produit dans la blockchain

Pour tracer un produit il faut que le premier maillon de la chaîne crée à partir de rien les premières données dans la blockchain. Dans notre exemple ce sera le laboratoire Indien qui déclarera disposer des précurseurs nécessaires à la production des matières premières et l'usine Polonaise qui inscrira les produits nécessaires à la fabrication des articles de conditionnement.

Cette étape est critique car le premier maillon de la chaîne crée un actif à partir de rien. Il est possible de pouvoir demander à une entité externe de certifier les quantités qui sont ajoutées par exemple. [9]

La figure 2.8 représente la création des actifs dans la blockchain. Chacune des deux entités va envoyer les informations relatives aux produits signées avec leur clé privée. On peut imaginer une grande variété d'informations comme la quantité, la péremption, l'identification des contenants par un data matrix, les références des matières utilisées pour la production, Dans le cas des matières premières produites en Inde, un tiers certificateur va venir approuver les informations comme la qualité des précurseurs par exemple

Ensuite deux des membres de la chaîne logistique différents de ceux impliqués dans la transaction sont tirés au hasard vont venir contrôler la validité de toutes les informations avant de les ajouter de manière immuable. Ils vont vérifier que les informations sont bien issues de membres autorisés en vérifiant leur signature, ils vont aussi vérifier que l'entité tiers a bien certifié les matières premières.

La création du produit dans la blockchain correspond à la création d'un contrat intelligent dont les règles ont été définies au préalable. Notamment les relations de transformation du produit tout au long de son parcours. [9]

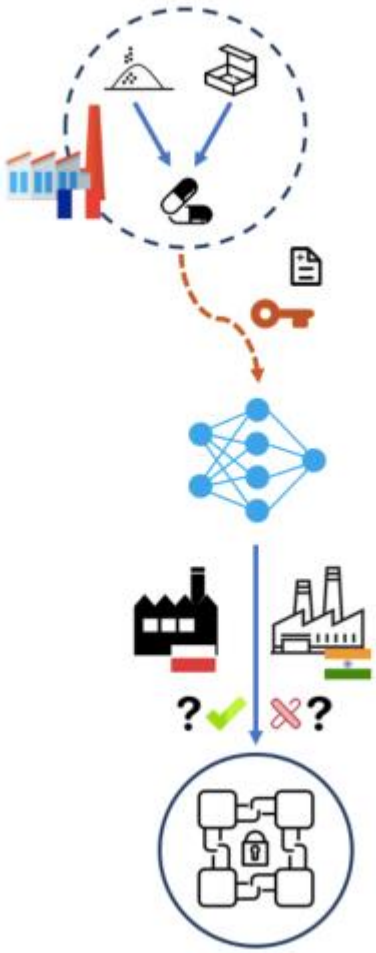


Figure 2.9 : déclaration de la production dans la blockchain

Les matières premières sont transformées en produits finis. Lors du consensus, deux membres du consortium tirés au hasard vérifient que toutes les données de transformation sont conformes. [9]

Nous avons donc à ce stade des matières premières enregistrées dans la blockchain, nous disposons de toutes les informations intrinsèques à ces matières ainsi que de l'identification de leur contenant qui peut être physiquement associée à un mécanisme d'inviolabilité pour garantir que le produit est bien conforme à ce qui est décrit informatiquement. [9]

2.12.2 Étapes de transformation

Que ce soit la fabrication des matières premières à partir des précurseurs, celle des articles de conditionnement à partir de carton et d'aluminium ou bien la production du produit fini cela revient à appliquer un changement au contrat intelligent.

Dans le contrat intelligent est défini une quantité maximale qu'il est possible de produire à partir des matières premières. Chaque produit intermédiaire est donc lié à son prédécesseur et son successeur dans la chaîne de production. Il est donc impossible informatiquement de créer des produits qui ne seraient pas issus de ces matières identifiées et contrôlées et donc d'introduire des produits illégitimes dans la chaîne logistique.

L'usine Française va donc déclarer sa production comme nous pouvons le voir sur la figure 2.10. Seront mentionnés la référence des produits utilisés ainsi que celle des produits fabriqués. Il faudra évidemment que tous les produits utilisés soient bien pré-identifiés dans la blockchain pour être utilisés, et que la production respecte les règles du contrat intelligent notamment que les quantités produites ne dépassent pas la quantité théoriquement faisable. Si l'entreprise le souhaite elle peut également ajouter des informations de production, comme la date de production, les équipements de production utilisés, …. [9]

Il est aussi possible en cas de problème de production de déclarer celle-ci inapte à la commercialisation. Ceci aura pour effet de bloquer toutes les étapes ultérieures et de prévenir l'arrivée du produit jusqu'au client. Comme pour l'étape précédente deux autres membres du consortium sont tirés au hasard pour valider la transaction, ici les entreprises Indienne et Polonaise. Si toutes les données sont conformes alors les produits fabriqués sont ajoutés dans la blockchain.

2.12.3 Étape de transport

Pour garantir la qualité et la sécurité des produits pendant le transport il faut d'une part disposer de l'identifiant de chaque unité transportée ainsi que chaque unité soit équipée d'un dispositif d'inviolabilité. L'inviolabilité d'un contenant associé à son identification permet de pouvoir scanner à tout moment le contenant, et ainsi récupérer les informations associées au produit et donc d'avoir confiance dans les produits que l'on transporte.

L'entité qui cède les produits et le transporteur qui les prend en charge vont le déclarer dans la blockchain. De la même manière un consensus va avoir lieu pour vérifier, que les entités de cession et réception sont légitimes et que l'identité des produits cédés (via leur data matrix) correspond bien à ceux enregistrés plus tôt dans la blockchain. [9]

Il est important de noter que pour chaque étape de transport et de stockage il est possible de placer des capteurs de température ou d'humidité dans les contenants qui peuvent enregistrer ces variables. Il est possible par la suite, via un oracle de blockchain de transférer

ces informations dans la blockchain. Le produit étant inscrit sous forme de contrat intelligent il peut être prévu que les données des capteurs puissent automatiquement invalider le produit en cas de non-respect des conditions de transport ou d'entreposage.

Si ce sont les matières premières qui sont concernées il ne sera plus possible de les transformer en produits finis dans la blockchain. Si ce sont les produits finis il ne sera pas possible de les délivrer. [9]

2.12.4 Délivrance au patient

Lors de la délivrance au patient le pharmacien va scanner le code data matrix présent sur la boîte. Si une étape de la chaine logistique concernant le médicament n'est pas en règle, il est possible de prévoir que le contrat intelligent refuse d'exécuter la délivrance. Si la délivrance est autorisée, l'officine va alors proposer au consensus d'enregistrer dans la blockchain le numéro unique de la boite délivrée ainsi que le patient à qui a été délivré la boite.

Le patient peut lui aussi scanner le code data matrix à l'aide d'une application connectée à la blockchain par exemple pour vérifier l'authenticité de son médicament.

Il existe déjà de nombreuses start-up qui proposent des services de traçabilité sous-tendu par une blockchain avec des applications permettant de vérifier l'authenticité des médicaments. [9]

2.13 Les avantages d'une logistique qui s'appuie sur la blockchain

2.13.1 Traçabilité exhaustive, confiance partagée, maitrise des données

De la production des matières premières jusqu'à la délivrance au patient, toutes les données concernant le produit sont contenues au sein d'une seule et même base de données. Chaque information a été ajoutée par une entité authentifiée, puis vérifiées par d'autres acteurs de la chaîne.

Le statut du produit est mis à jour dans la blockchain à chaque étape de réception, cession ou transformation avec un lien direct entre les matières premières et le produit fini le tout en temps réel. Il est impossible d'introduire des produits non légitimes au milieu de la chaîne logistique ceux-ci seraient aussitôt détectés. [9]

La base de données est physiquement répliquée chez tous les acteurs de la chaine logistique ainsi il est presque impossible de pouvoir la pirater. Il est aussi impossible qu'un acteur de la chaine logistique puisse modifier des informations à son avantage.

Enfin il est possible d'entrer des données dans la blockchain sans que les autres acteurs ne puissent y accéder. Lors de la délivrance par exemple il est possible lors du consensus de s'assurer que le pharmacien ait bien renseigné l'identité du patient auquel il a délivré le médicament, le tout sans divulguer cette information aux deux membres du consortium qui valident l'information. Une fois stockée l'identité du patient ne sera pas visible par les membres du consortium, excepté le pharmacien. Chaque entreprise peut stocker dans cette blockchain des informations qu'elle ne veut pas partager comme ses prix d'achat et de vente par exemple. Il est de ce fait possible de voir encore plus loin et d'automatiser des paiements entre les acteurs en incluant des pénalités de retard lors des livraisons par exemple grâce à des contrats intelligents. [9]

2.13.2 Une traçabilité à double sens

Grâce à l'unification du stockage des données il est possible de descendre ou remonter la chaine logistique très rapidement et à très faible cout. En effet si a posteriori un lot de matière première est déclaré non conforme par exemple il est possible de descendre toute la chaine logistique très rapidement pour informer tous les acteurs et retirer les produits jusque chez le patient. [9]

À l'inverse un patient peut à l'aide d'une application scanner le data matrix sur sa boite de médicament et être certain de la conformité de son médicament du fait de la conformité de toutes les étapes amont.

Selon la quantité d'information que veulent partager les entreprises le patient peut seulement savoir que le produit est conforme mais il est possible de partager plus d'informations comme les lieux et dates de production si les entreprises du consortium le désirent.

La blockchain peut être un excellent moyen de faire preuve de transparence et contribuer à maintenir la confiance avec les patients.

Enfin un suivi logistique appuyé par la technologie blockchain est plus facilement audit able par les autorités sanitaires en cas de besoin. Si besoin, les entités de la chaine peuvent ouvrir leurs données à une autorité régulatrice qui en quelques instants peut disposer de toutes les informations disponibles sur le produit. [9]

2.13.3 La sécurité disponible aux personnes en ayant le plus besoin

Si les mesures mises en place dans les pays à hauts revenus sont relativement efficaces pour lutter contre la falsification, les améliorations comme la directive européenne sur les médicaments falsifiés ou le Drug Supply Chain Security Act peinent à améliorer le problème des médicaments falsifiés dans les pays en développement. [9]

L'avantage d'un système blockchain est d'être plus universel. Si un laboratoire met en place un système de traçabilité blockchain pour ses médicaments ce dernier pourra s'appliquer quel que soit le pays de vente. Contrairement à la base de données pour les médicaments falsifiés en Europe qui par définition n'a d'utilité que pour les pays européens. Ainsi toute personne avec un téléphone portable pourrait être en mesure de vérifier l'authenticité des médicaments qu'elle achète en scannant le code data matrix sur la boite.

De plus, il est important de noter que les téléphones mobiles sont de plus en plus utilisés dans les pays en développement. En Afrique sub-saharienne par exemple 80% de la population dispose d'un téléphone mobile, et 90 % de la population bénéficie d'une couverture réseau. (159,160) Il est donc tout à fait raisonnable penser pouvoir combattre les médicaments falsifiés en proposant des services basés sur la téléphonie et appuyés en fond par la blockchain.

La blockchain associée à la sérialisation peut ainsi révolutionner la logistique pour les produits médicaux. Cela permet de garantir la continuité de l'information, son accessibilité, sa transparence et son auditabilité le tout en ayant des infrastructures moins lourdes à mettre en place que les solutions disponibles jusqu'ici. [9]

2.14 Conclusion

Les opérations de la chaîne d'approvisionnement sont complexes et nécessitent une gestion approfondie. L'utilisation efficace des services logistiques contractuels place les entreprises dans une position avantageuse en termes de gestion de la chaîne et d'accord de leurs opérations de distribution.

Les défis de la chaîne d'approvisionnement en médicaments en Algérie constituent un obstacle dans le système de la chaîne d'approvisionnement pharmaceutique, le risque de fraude et de falsification dans les systèmes traditionnels est significativement élevé. Il a besoin d'une plate-forme bien conçue et fiable, ainsi que d'un moyen de mettre en œuvre et d'utiliser ce système de base. Les fabricants et distributeurs de médicaments devraient envisager en

utilisant un programme d'agrégation de plates-formes pour relever ces défis et simplifier la chaîne d'approvisionnement.

Depuis le début de l'ère numérique, les organisations recherchent des améliorations à leur structure commerciale actuelle et assurent de manière significative la communication et la transparence tout au long de la chaîne d'approvisionnement en médicaments grâce à la modernisation et à l'émergence de nouvelles technologies. La blockchain est une technologie numérique capable de prendre en charge les processus actuels et de désactiver des modèles complexes. La technologie rend le processus sans papier que toutes les parties impliquées peuvent interagir les unes avec les autres en utilisant clés. Les industries peuvent gagner l'avantage en adoptant la technologie dans leurs activités, elle fournit la connectivité nécessaire, une sécurité accrue et une transparence totale pour concrétiser la vision de la chaîne d'approvisionnement.

Dans le prochain chapitre, nous proposerons un nouveau système simple de chaîne d'approvisionnement en médicaments utilisant la technologie Blockchain pour gérer les enregistrements sécurisés de la chaîne d'approvisionnement en médicaments. Le système proposé résout ce problème en enregistrant les transactions des processus de la chaîne d'approvisionnement sur une base de blockchain pour créer un écosystème intelligent afin de réduire les défis et les problèmes rencontrés par la chaîne d'approvisionnement pharmaceutique algérienne

3 Chapitre III : Conception et implémentation

3.1 Introduction

Dans ce chapitre, nous allons implémenter un système simple utilisant Ethereum Blockchain pour gérer des enregistrements sécurisés et pour les services d'information fournis par les participants à l'industrie pharmaceutique (chaîne d'approvisionnement pharmaceutique). En utilisant des contrats intelligents Ethereum, une entreprise conjointe de la chaîne d'approvisionnement est accompli. Ce système proposé résout les problèmes en conduisant des distribu-transactions sur la base de la blockchain, pour éliminer le besoin d'administration du système par un tiers. Ainsi, les utilisateurs peuvent faire confiance aux informations qu'ils voient dans le système en tant que les données ne sont en aucun cas altérées.

3.2 Système proposé

Dans ce chapitre, nous allons proposer une architecture système blockchain qui nous aidé à lutter la contrefaçon des drogues. Comme le montre la figure 3.1. Les grandes fonctionnalités de la technologie blockchain qui sont utiles pour la traçabilité, la sécurité et la transparence. Un système basé sur la blockchain est inséré pour fournir un système décentralisé sécurisé système de pistage. L'architecture du système repose sur la blockchain Ethereum et la smart contracts pour supprimer le besoin de gestion du système par un tiers. L'application consiste en un contrat intelligent pour Ethereum, qui contient les processus de la chaine d'approvisionnement, en outre, il a fourni la possibilité de stocker et de récupérer des enregistrements à partir du ledger blockchain, ce qui facilite le suivi du produit et garantit que les données ne peuvent pas être changées.

Nous allons créer :

1. Contrats intelligents Ethereum avec le langage de programmation Solidity
2. Nous écrirons des tests pour les contrats intelligents en JavaScript
3. Nous allons déployer les contrats intelligents vers une blockchain
4. Nous allons créer un site Web côté client avec Web3.js et React.js afin que les utilisateurs puissent parler aux contrats intelligents

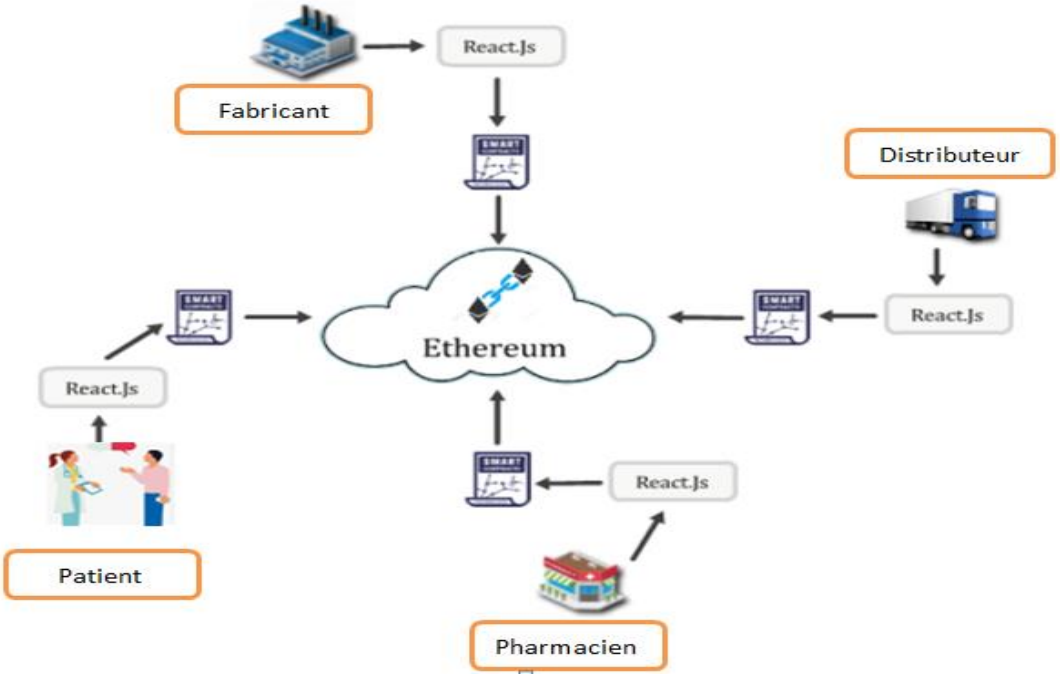


Figure 3.1: Architecture du système

3.3 Composants du système

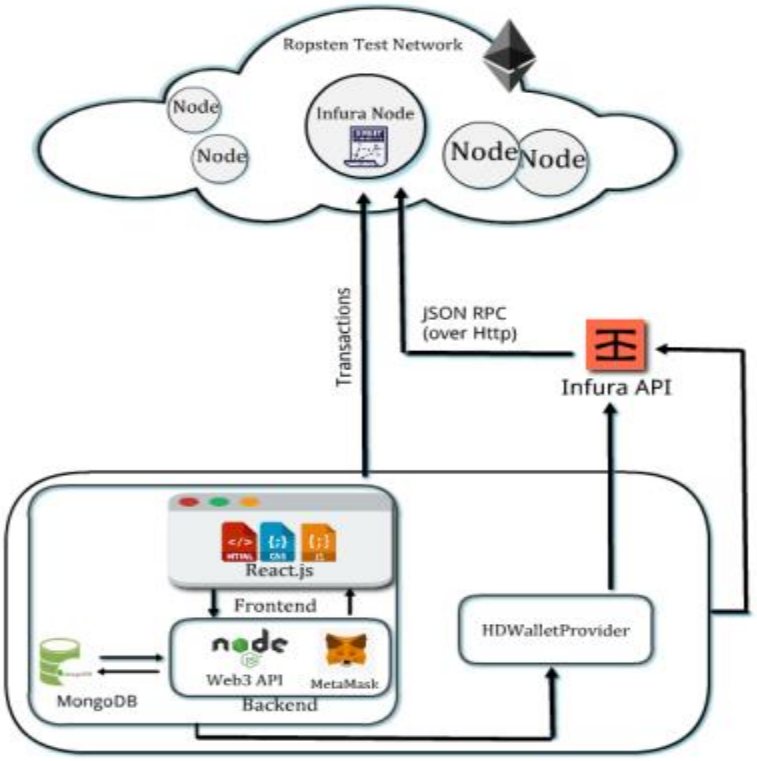


Figure 3.2: Architecture détaillée du système

3.3.1 Blockchain

La blockchain peut permettre un suivi plus transparent et précis dans la chaîne d'approvisionnement pour aider à réduire la fraude pour des produits tels que les médicaments pharmaceutiques, grâce à la numérisation physique actifs et créer un registre immuable décentralisé de toutes les transactions, permettant de suivre les actifs de la production à la livraison ou à l'utilisation par l'utilisateur final. La blockchain utilisé comme composant principal de ce système. En utilisant cette technologie, fournit à tous les participants de la chaîne d'approvisionnement un accès aux mêmes informations, réduisant potentiellement erreurs de communication ou de transfert de données.

L'objectif principal de la solution est de détecter les détails des actifs numériques créés dans le système. Par conséquent, nous voulons enregistrer les données telles quelles, et une fois qu'elles sont ajoutées au réseau, personne ne peut le changer ou le modifier avant de gérer le système. Nous avons choisi la plate-forme Ethereum sur d'autres plates-formes blockchain comme Hyperledger, nous avons essayé de détecter tous les détails mineurs de l'origine aux utilisateurs et de le faire vérifiable par toutes les parties prenantes de la chaîne d'approvisionnement. Toutes les informations concernant les comptes ou les transactions effectuées sur le réseau peuvent être vérifiées sur Etherscan.io, selon à l'adresse de compte de chaque membre.

3.3.2 Contrats intelligents

Les contrats intelligents sont des logiciels utilisés par tous les membres de la chaine d'approvisionnement pour initier et exécuter des transactions et diverses règles des transactions sont mises en œuvre par le Smart Contract. Le contrat est ensuite déployé sur un test Ethereum réseau appelé **Ropsten**, un réseau de test qui exécute le même protocole qu'Ethereum fait et est utilisé à des fins de test avant le déploiement sur le réseau principal (Mainnet).

Nous nous connectons à un nœud spécifique sur le réseau Ropsten, afin que le contrat puisse être publié à ce noeud spécifique

Le figure3.3 montre le diagramme de séquence du contrat intelligent et la relation parmi les participants. Le contrat contient les fonctions et les actions de la chaîne d'approvisionnement, nous décrirons la fonction de chaque participante étape par étape.

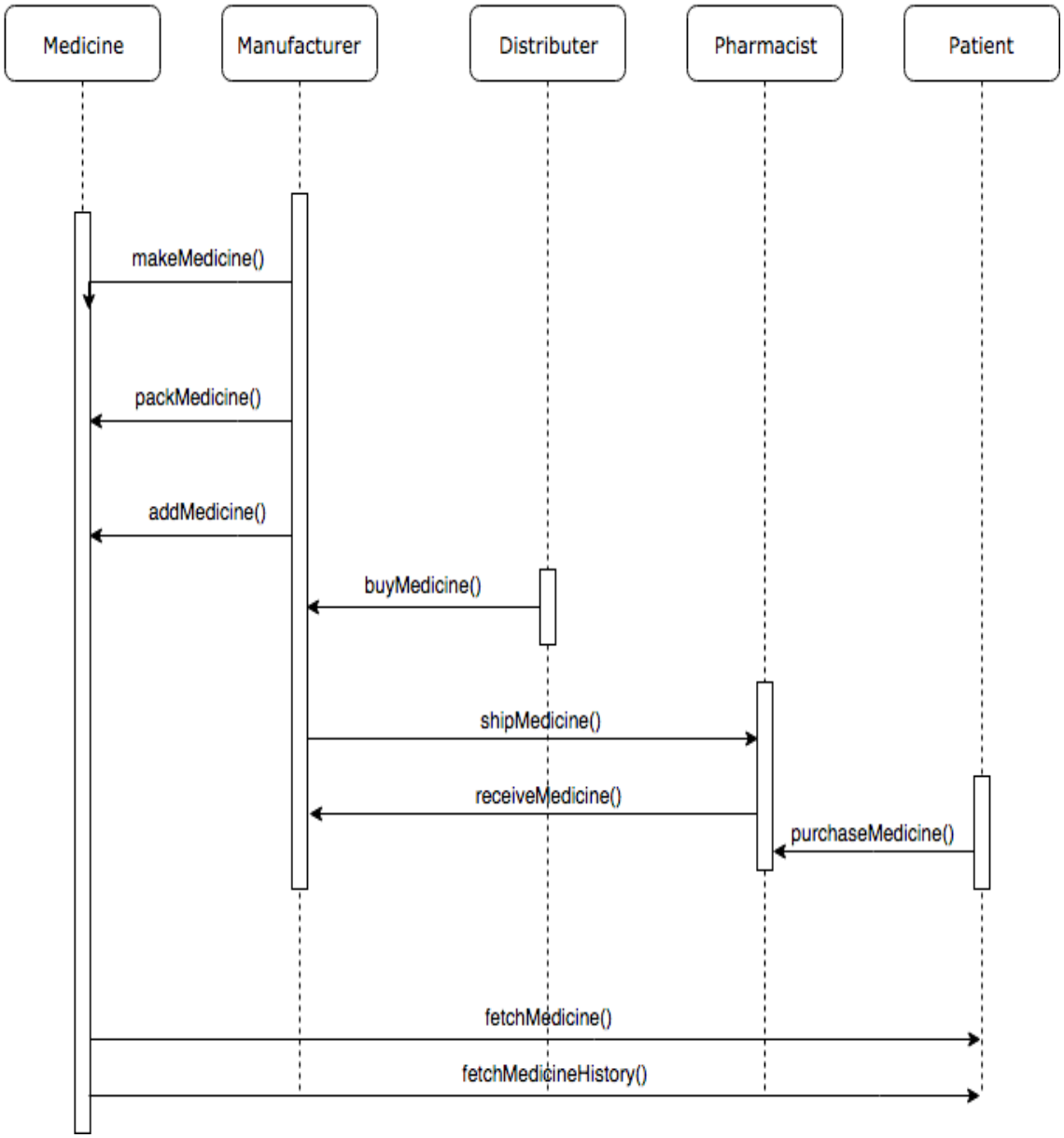


Figure 3.3: Diagramme de séquence des fonctions de contrat intelligent

Le figure3.4 montre le diagramme d'état des fonctions de contrat intelligent.

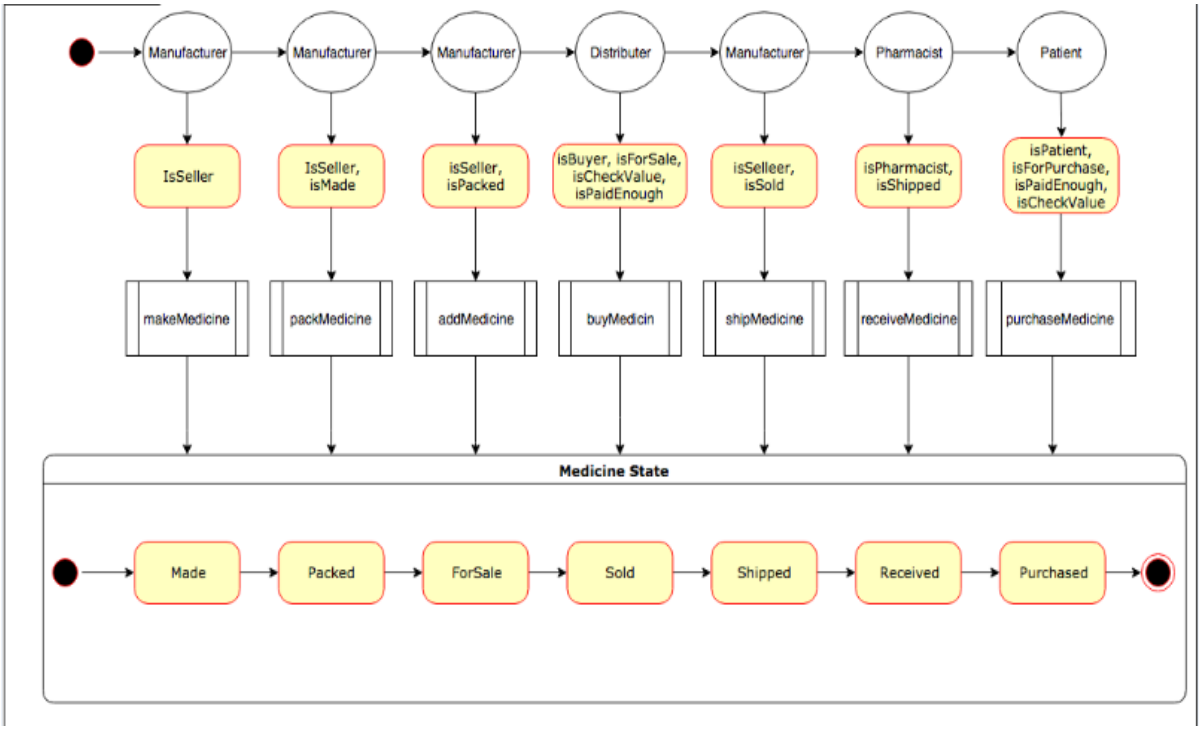


Figure 3.4: Diagramme d'État des fonctions de contrat intelligent

Le figure3.5 montre le diagramme d'activité des fonctions de contrat intelligent.

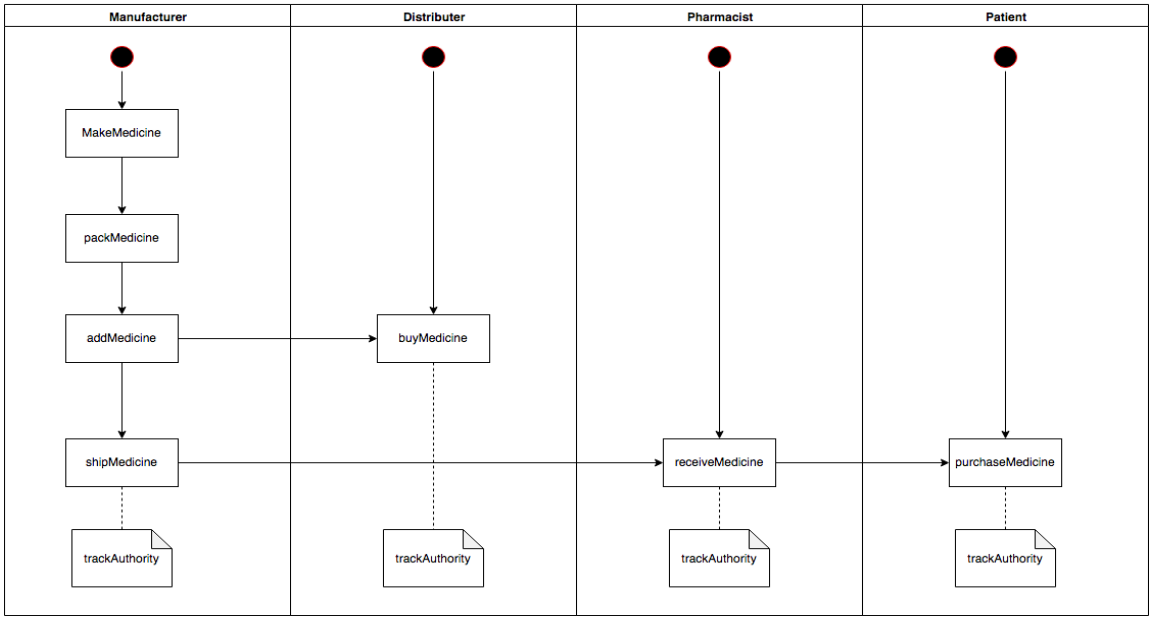


Figure 3.5: Diagramme d'Activité des fonctions de contrat intelligent

Étape 1: Un fabricant produit les médicaments, contenant des informations essentielles telles que code du produit, nom du médicament, quantité, nom et informations du fabricant. Les informations ajoutées par le fabricant sont stockées sur la blockchain, ce qui permet d'autres parties prenantes pour retracer la chaîne d'approvisionnement des médicaments de manière transparente. Après fabrication, Le fabricant emballe les médicaments, et les vend aux distributeurs, ils mettront l'argent dans un contrat intelligent dès le départ et dès que la compagnie maritime informe le contrat intelligent qu'il a récupéré la commande, le contrat intelligent sera automatiquement libéré les fonds.

Étape 2: Une fois que les fournisseurs de services logistiques (fabricants) livrent les médicaments aux distributeurs, ils peuvent vérifier l'origine des médicaments à l'aide du code produit (UPC) stocké sur la blockchain. Ils peuvent retracer les informations ajoutées par les fabricants telles que les quantités de médicaments, où ils ont été fabriqués. Distributeurs valider les médicaments reçus et signer la transaction numériquement qui est ensuite ajoutée à la blockchain. Les transactions signées déclenchent les contrats intelligents pour expédier les médicaments aux pharmaciens, le contrat intelligent libérera le paiement de la compagnie maritime.

Étape 3: Les pharmaciens obtiennent les médicaments qui peuvent être retracés pour connaître leur origine en utilisant le code produit (UPC) enregistré sur la blockchain. Si un distributeur illégal tente de vole une quantité de médicaments distribués ou le retard de livraison, la transaction est considérée comme invalide en raison des informations frauduleuses ajoutées sur la distribution traité. Par conséquent, les pharmaciens sauraient immédiatement si des anomalies sont trouvées dans les transactions. Une fois que le pharmacien a approuvé les médicaments reçus, la transaction entre eux et le distributeur est ajoutée à la blockchain, assurant l'accord juridique entre eux. En outre, les pharmaciens vendent les médicaments aux clients (patient) et sont ajoutés transaction sur blockchain.

3.3.3 API Infura

Un moyen simple de réussir l'accès à divers réseaux sans avoir à mettre en place un nœud complet pour chacun d'eux-mêmes consiste à créer un compte Infura. Infura maintient sa propre infrastructure qui offre un accès facile à divers réseaux blockchain comme Ethereum. Infura propose des API d'acceès sécurisées, fiables, évolutives et faciles à utiliser pour accéder le réseau Ethereum et d'autres plates-formes décentralisées. C'est un Ethereum

hébergé cluster de nœuds qui permet aux utilisateurs d'exécuter l'application sans les obliger à configurer leur propre Nœud ou portefeuille Ethereum. Les comptes Infura vous permettent de déployer du code et d'interagir avec les réseaux Mainnet, Ropsten, RinkeBy et Kovan. Nous utilisons Ropsten Testnet car il plus comme Ethereum et nous pouvons facilement obtenir de faux éthers. Il n'y a pas besoin de payer réel éther pour interagir avec la blockchain Ethereum mais peut avoir une idée de la vraie ow du point de vue de l'utilisateur final.

3.3.4 Web3

L'un des frameworks les plus utilisés pour la conception de DApps est Web3.js. Web3 fait il est facile d'interagir avec les contrats intelligents sur la blockchain Ethereum en appliquant l'interface. Ethereum Blockchain nous fournit web3.js, qui est une API utile à faire la vie d'un développeur Web plus facile. L'API JavaScript nous permet de communiquer avec un Nœud Ethereum utilisant les points de terminaison JSON RPC exposés au-dessus de HTTP, IPC ou Transferts WebSocket à partir de la page Web, grâce à donner l'interface json du contrat intelligent et web3 convertira automatiquement toutes les fonctions en appels ABI de bas niveau via RPC.

3.3.5 Fournisseur HDWallet

Le fournisseur HDWallet est une connexion réseau appropriée et facile à configurer à ethereum via infura.io. Le fournisseur ajoute des caractéristiques qui ne sont pas disponible avec infura comme le filtrage d'événements et la signature de transaction en utilisant les 12 mots mnémonique pour déverrouiller le compte MetaMask et utiliser le compte pour déployer l contrat

3.3.6 Backend ET Front-end

Le contrat contrôle tous les fonds et fonctions essentiels, mais afin de faciliter l'accès au contrat pour les utilisateurs, il doit y avoir un moyen pour l'utilisateur d'interagir avec le contrat en utilisant un site Web avec des boutons connectés aux fonctions du contrat. L'extrémité avant de ce système est créée avec React qui aide à implémenter l'interface pour afficher le web contenu de la page à l'utilisateur plus rapidement. Nous avons choisi React pour le développement front-end car il est très populaire parmi les frameworks frontaux et n'est pas groupé par rapport aux autres bibliothèques. Il est également facile à apprendre, rapide et évolutif permettant d'afficher du HTML sur une page Web.

Nous implémentons le backend en développant une API avec NodeJS / Express, et la base de données de documents MongoDB

- NodeJS est un environnement de serveur gratuit et open-source qui utilise JavaScript et exécute sur plusieurs plates-formes.
- ExpressJS est un framework d'application Web Node.js minimal et flexible qui fournit un ensemble robuste de fonctionnalités pour les applications Web et mobiles.
- MongoDB est une base de données de documents avec une évolutivité et une flexibilité importantes. Il présente les caractéristiques suivantes:
    - MongoDB stocke les données dans des documents exibles de type JSON
    - En tant que base de données distribuée à sa base, MongoDB fournit la disponibilité, la mise à l'échelle horizontale et la distribution géographique.

Nous utilisons la base de données mongoDB pour stocker les données de manière ordonné et faire il est facile pour l'utilisateur de le visualiser.

Nous appelons cet API en utilisant **React** et **Redux**, c'est un framework qui a une action et un réducteur que nous utilisons pour appeler cette API. React a frappé notre API backend et notre appel backend certaines fonctions de contrat en utilisant web3 et en fin de compte envoyer des données à la blockchain et envoye message de réussite à l'utilisateur du frontend.

3.4 Restauration des données à partir de la blockchain

L'objectif principal de ce travail est de fournir une meilleure visibilité, transparence et précision des transactions tout au long du processus de la chaîne d'approvisionnement, qui est maintenue dans la technologie blockchain. Pour atteindre cet objectif, nous devons restaurer les données qui ont été enregistré dans la blockchain. Parce que les transactions ne sont servies qu'à un nœud Ethereum via un format de chiffrement de texte appelé JSON RPC, qui est une télécommande légère Interface d'appel de procédure qui utilise JSON comme structure de données pour modéliser les données envoyé à la blockchain. Les nœuds participants affichent cette interface dans le même processus, via des sockets, via HTTP ou dans de nombreux environnements de transmission de messages. Dans ce projet, les connexions HTTP sont utilisées pour envoyer des transactions via JSON RPC qui joue un rôle majeur lorsque l'on souhaite restaurer les données de la blockchain en tant que comptes utilisateurs, envoyer des transactions, interagir avec des contrats intelligents, etc. Les utilisateurs ou les applications peuvent envoyer des appels JSON RPC directement à un nœud, formant la structure de

données JSON requise et l'envoyer à l'interface exposée. Cependant, cela est lourd et prend du temps. Par conséquent, les programmeurs prennent en charge des bibliothèques de différents langages de programmation. Cela permet aux programmeurs de travailler dans le langage d'application et de créer des interactions blockchain, comme l'envoi d'une transaction. Ceci est ensuite automatiquement traduit en JSON RPC format et envoyé au nœud Ethereum comme Web3.

Web3.js est une bibliothèque JavaScript pour nous aider à faire évoluer les sites Web ou les clients qui interagissent avec blockchain, et écrivez du code qui lit et écrit les données de la blockchain en utilisant contrats intelligents. La figure 3.4 montre comment web3 est utilisé dans le système. Afin de lire données de la blockchain utilisant web3, un exemple de contrat intelligent a été implémenté dans les représentations JavaScript.

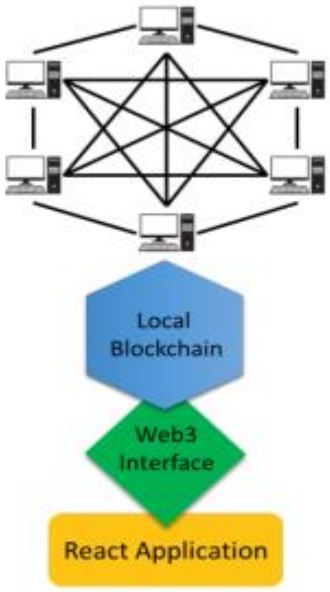


Figure 3.6: Utilisation de Web3 dans Blockchain

3.5 Interactions avec le système

Le processus de fabrication, de vente, d'achat, d'expédition et de visualisation l'histoire du système. Tous les participants à la chaîne d'approvisionnement en médicaments doivent s'inscrire le système via MetaMask. Après avoir configuré un mot de passe, un compte de douze mots mnémonique et adresse. En permettant aux utilisateurs d'accéder au mnémonique de compte de douze mots leur compte n'importe où avec n'importe quel navigateur qu'ils

préfèrent. Lorsque le fabricant exécute l'application il ouvre le Metamask, il lui donne la clé privée et se connecte à Metamask donc notre première étape fabricant attaché avec Metamask. Sur la page principale, le fabricant peut créer un nouveau produit dans le système en saisissant des informations, y compris l'identifiant, le nom et les informations fabricant, code, nom, quantité produit. Après avoir rempli les informations sur le formulaire en cliquant sur le bouton de fabrication et il est stocké dans la base de données mongodb. Puis, le fabricant emballe un produit en cliquant sur le bouton Pack. Après cela, le fabricant peut être mettre en vente un produit via le code produit et le prix en cliquant sur le bouton ForSale. Automatiquement, Metamask confirme les transactions de chaque action en déduisant la mise montant d'un compte du fabricant.

Cependant, dans cette application, nous nous connectons à Metamask avec la première adresse de l'utilisateur Metamask. Plus tard, nous permettons aux utilisateurs de sélectionner n'importe quel compte de Metamask à la place le premier une. Le distributeur peut acheter un produit qu'il souhaite en entrant le nom, le prix et l'identifiant du produit, quantité et identifiant du distributeur, ces informations seront sauvegardées dans la base de données mongodb. Le fabricant expédie un produit au distributeur, en ajoutant une transaction en cliquant sur Bouton de livraison. Après cette étape, le distributeur doit confirmer la livraison pour laisser le cœur de la blockchain ou du smart contract sait qu'il a reçu le produit avec quantité requise. Le distributeur peut également mettre un produit en vente au pharmacien via le code produit et le prix en cliquant sur le bouton ForSale, avec les besoins de confirmer une transaction ou rejeter. Lorsque vous cliquez sur n'importe quel bouton, Metamask confirme automatiquement une transaction et l'enregistre dans Blockchain. Les mêmes fonctions entre distributeur et revendeur (achat, réception et expédition).

De l'autre côté, lorsque le pharmacien vend le produit au patient, il ajoute les informations du produit avec l'identifiant du patient sous forme en cliquant sur le bouton de vente, un MetaMask a automatiquement déduit le montant de la mise de son compte. La transaction est confirmée en déduisant automatiquement le montant de la mise du compte pour ajouter une transaction sur la blockchain.

Toutes les informations seront sauvegardées de manière organisée dans la base de données mongodb et et affiché à l'utilisateur de manière simple et claire. Chaque participant est autorisé à voir toutes transactions d'une chaîne d'approvisionnement.

3.6 Outils de développement

3.6.1 Configuration du système et système d'exploitation

Le projet est réalisé sur CPU 2.40 GHz 2.40 GHz Intel Core i5-6300U, et 8 Go de RAM. Nous implémentons le projet sous Windows 8 64 bits.

3.6.2 Remix IDE

Remix est un IDE qui est accessible par navigateur à l'adresse suivante https://remix.ethereum.org (le tout premier chargement peut prendre quelques secondes selon la vitesse de votre connexion)

Remix est très pratique et très pertinent pour apprendre à coder sur Solidity pour plusieurs raisons:

– On y accède juste par navigateur et il n'y a rien à installer.

– On dispose automatiquement des dernières versions de Solidity.

– Il y a un débugueur intégré.

– Il permet de compiler et d'exécuter les smart contracts instantanément, dans toutes sortes de blockchains, c'est à dire qu'on peut déployer dans la vrai blockchain Ethereum un smart contract directement depuis Remix, mais il peut aussi se connecter à la blockchain de test (qui s'appelle Ropsten), ou à une blockchain locale comme Ganache. Il est donc très souple et flexible. Pour l'apprentissage, il y a même une VM Javascript qui permet de simuler une blockchain pour faire des tests, avec directement des adresses Ethereum qui on chacun de l'ETHER.

La somme d'outils à installer (node, git, ide etc) peut être repoussante pour débuter à développer des smart contracts, Solidity règle ce problème puisqu'il suffit de se rendre sur une URL pour démarrer sur un environnement réel et non pas dégradé comme c'est parfois le cas sur les outils en ligne. [34]

3.6.3 Code Visual Studio

Visual Studio Code est un éditeur de code multiplateforme édité par Microsoft. Ce outil destiné aux développeurs supporte plusieurs dizaines de langages de programmation comme le HTML, C++, PHP, Javascript, Markdown, CSS, etc. Visual Studio Code intègre plusieurs outils facilitant la saisie de code par les développeurs comme la coloration syntaxique ou encore le système d'auto-complétion IntelliSense. En outre, l'outil permet aux développeurs de

corriger leur code et de gérer les différentes versions de leurs fichiers de travail puisqu'un module de débogage est aussi de la partie. [33]

3.6.4 Truffle

Truffle est un framework de développement Ethereum (créé par ConsenSys, l'entreprise co-fondée par Vitalik Buterin). Il permet d'interfacer des smart contracts avec du code JavaScript et l'ensemble de l'écosystème NodeJS. Cela ouvre donc la voie à l'utilisation des outils d'industrialisation du monde JavaScript pour la chaîne de blocks Ethereum.

Truffle apporte les fonctionnalités suivantes : [35]

- Gestion des dépendances (au travers de NPM ou EthPM - le package manager Ethereum de smart contracts)
- Compilation des contrats
- Migrations
- Tests (2 modes possibles)

3.6.5 Ganache

Ganache est une blockchain personnelle pour une application distribuée rapide Ethereum et Corda développement. Vous pouvez utiliser Ganache tout au long du cycle de développement; vous permettant pour développer, déployer et tester vos applications distribuées de manière sûre et déterministe environnement.

Ganache UI est une application de bureau prenant en charge les technologies Ethereum et Corda. De plus, une version Ethereum de ganache est disponible en tant qu'outil de ligne de commande: ganache-cli (anciennement connu sous le nom de TestRPC). Toutes les versions de Ganache sont disponibles pour Windows, Mac et Linux.

Cela nous permettra de déployer des contrats intelligents, de développer des applications et d'exécuter des tests. Nous avons choisi Ganache car il nous fournit 10 comptes Ethereum avec un solde de 100 éther (faux éther) pour chaque compte, ainsi qu'une interface graphique qui permet nous pour vérifier tout ce qui se passe dans cette blockchain.

3.6.6 Node.JS

Node.JS est une plateforme de développement JavaScript intégrant un serveur HTTP. Son fonctionnement se base sur une boucle événementielle qui lui permet le support de fortes montées en charge. Caractérisée comme étant une bibliothèque de ce langage, elle permet la

réalisation d'actions comme créer un fichier ou bien ouvrir et fermer des connections réseau. Un point important qu'il faut noter est qu'elle n'est ni un framework ni un serveur !

Node JS est caractérisé comme :

- un logiciel libre sous licence MIT (licence pour logiciels libres et open source)
- un système non bloquant
- une performance du moteur de JavaScript V8 de Google qui est aussi focalisée sur la sécurité.

Cette plateforme logicielle libre conçoit des applications réseau rapides et évolutives. Elle doit élaborer ces applications en temps réel pendant que le serveur doit être capable de donner des informations au client. Elle a été conçu pour le développement d'applications côté serveur tout en disposant d'un environnement d'exécution ainsi que d'une bibliothèque JavaScript. Elle est aussi très souvent utilisé pour écrire des services côté serveur que l'on appelle API qui signifie Application Programming Interface. [36]

Remarque : Pour d´evelopper des smart contracts, nous devons configurer notre environnement par l'installation de Node Package Manager(NPM), fourni avec Node.js.

3.6.7 React

React (aussiappelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité. [37]

3.6.8 MetaMask

MetaMask est une extension ou un plugin pour les navigateurs Web qui permet aux utilisateurs d'interagir facilement avec les DApps sur la blockchain Ethereum. Cela est possible, car MetaMask agit comme un pont entre les DApp et les navigateurs Web, facilitant leur utilisation et leur utilisation. [38]

3.7 Implémentation

3.7.1 Configuration de l'environnement

Tout d'abord, nous allons créer un répertoire qui contiendra les fichiers de notre projet dans l'invite Commandes comme ceci:

```
$ md Medical-Blockchain-medicament
```

```
$ cd Medical-Blockchain-medicament
```

Avant de commencer, il faut installer Truffle dans votre ordinateur (assurez vous d'avoir NodeJs déjà installé). Pour ce faire, ouvrez un terminal et exécutez cette commande:

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

Ensuite créons notre projet backend en utilisant Truffle, nous exécutons truffle init, cela mettra en place la structure de base suivante dans notre répertoire montré dans la figure 3.7:

```
$ truffle init
```

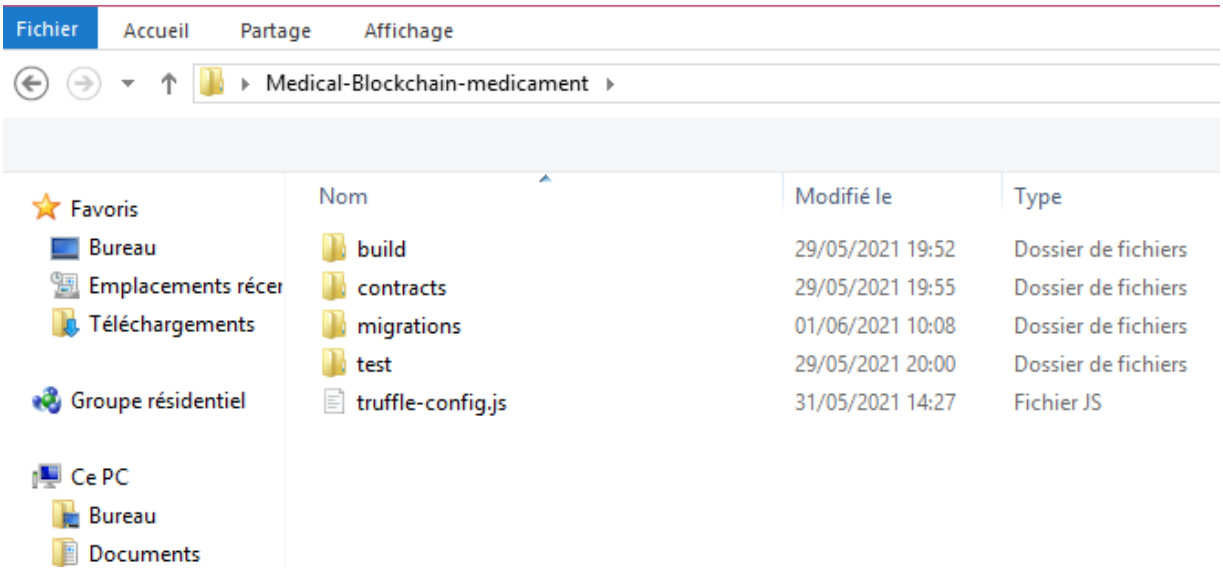


Figure 3.7: Structure du répertoire de Medical-Blockchain-medicament

- **contracts**: ce dossier contient les smarts contracts du projet
- **migrations**: ici on retrouvera tous les scripts pour les déploiements des contrats
- **tests**: ce dossier contient les fichiers pour tester l'application et les smarts contratcts
- **truffle-config.js**: c'est le fichier de configurer de Truffle. Par exemple ici on mettra le network que l'on utilisera pour déployer notre application (Localhost, Testnet, Mainnet)

3.7.2 Rédaction d'un contrat intelligent

Pour construire une application de chaîne d'approvisionnement décentralisée, nous créons d'abord le contrat intelligent Ethereum, nommé fichier SupplyChain.sol dans le répertoire "Contracts / medicinebase ". Nous devrait spécifier la version 0.4.24 du compilateur Solidity.

```
pragma solidity ^0.4.24;
import "../medicineaccesscontrol/Roles.sol";
import "../medicineaccesscontrol/DistributorRole.sol";
import "../medicineaccesscontrol/ManufacturerRole.sol";
import "../medicineaccesscontrol/PatientRole.sol";
import "../medicineaccesscontrol/PharmacistRole.sol";
import "../medicinecore/Ownable.sol";
// Define a contract 'Supplychain'
contract SupplyChain is PharmacistRole, PatientRole, ManufacturerRole, DistributorRole {
  // Define 'owner'
  address owner;
  // Define a variable called 'upc' for Universal Product Code (UPC)
  uint  upc;
  // Define a variable called 'sku' for Stock Keeping Unit (SKU)
  uint  sku;
  // Define a public mapping 'medicines' that maps the UPC to an Medicine.
  mapping (uint => Medicine) medicines;
  // Define a public mapping 'medicinesHistory' that maps the UPC to an array of TxHash,
  // that track its journey through the supply chain -- to be sent from DApp.
  mapping (uint => string[]) medicinesHistory;
  // Define enum 'State' with the following values:
  enum State
  {
    Made,       // 0
    Packed,     // 1
    ForSale,    // 2
    Sold,       // 3
    Shipped,    // 4
    Received,   // 5
    Purchased   // 6
  }
```

À l'intérieur du formulaire de contrat, nous définissons quatre variables d'état pour stocker l'adresse du propriétaire du contrat, qui est le compte de déploiement du contrat intelligent, 'UPC' pour le code produit, 'SKU' pour l'unité de gestion des stocks, 'medicines' qui

mappe le code produit (UPC) à un 'medicines'. Nous définir un mappage public 'medicineHistory' qui mappe le code produit à un tableau de TxHash, qui suit son parcours dans la chaîne d'approvisionnement à envoyer depuis DApp. Dans De plus, nous définissons les états des processus de la chaîne d'approvisionnement.

La structure de l'article définit les détails pour chaque produit unique décrit dans le code suivant:

```
// Define a struct 'Medicine' with the following fields:
struct Medicine {
  uint    sku;  // Stock Keeping Unit (SKU)
  uint    upc; // Universal Product Code (UPC), generated by the Manufacturer, goes on the package, can be ver
  address ownerID;  // Metamask-Ethereum address of the current owner as the medicine moves through 8 stages
  address originManufacturerID; // Metamask-Ethereum address of the Manufacturer
  string  originFactoryName; // Manufacturer Name
  string  originFactoryInformation;  // Manufacturer Information
  string  originFactoryLatitude; // Factory Latitude
  string  originFactoryLongitude;  // Factory Longitude
  uint    medicineID;  // Product ID potentially a combination of upc + sku
  string  medicineNotes; // Product Notes
  uint    medicinePrice; // Product Price
  State   medicineState;  // Product State as represented in the enum above
  address distributorID;  // Metamask-Ethereum address of the Distributor
  address pharmacistID; // Metamask-Ethereum address of the Pharmacist
  address patientID; // Metamask-Ethereum address of the Patient
}
```

Un constructeur est un type spécial de fonction qui est automatiquement déclenché lorsqu'un contrat intelligent est déployé sur un réseau. Il ne peut pas être appelé par la suite. L'importance d'écrire constructor () comme payable est que l'adresse de l'expéditeur doit être obligée en tant que propriétaire. Nous définissons le propriétaire sur msg.sender, qui est le représentant d'un serveur intégré variable globale de l'adresse qui appelle la fonction.

```
// In the constructor set 'owner' to the address that instantiated the contract
// and set 'sku' to 1
// and set 'upc' to 1
constructor() public payable {
  owner = msg.sender;
  sku = 0;
  upc = 0;
}
```

Les modificateurs sont déclarés à l'intérieur du contrat intelligent. Ils contiennent un ensemble de conditions qui permettront ou non l'exécution de la fonction. Nous spécifions six modificateurs contenant une condition, que si '_upc' est passé par toutes les étapes de la chaîne d'approvisionnement (made, packed, forsal, sold, shipped, receiver, purchased). Les modificateurs peuvent modifier l'état du contrat intelligent.

```
89    // Define a modifier that checks if an medicine.state of a upc is Made
90    modifier made(uint _upc) {
91      require(medicines[_upc].medicineState == State.Made);
92      _;
93    }
```

3.7.2.1 Fonctions de la chaîne d'approvisionnement

Nous définissons la fonctionnalité de notre contrat intelligent de chaîne d'approvisionnement, nous avons besoin pour définir ses fonctions et fournir le code de la structure de chaque fonction.

1. **La Fonction makemedecine()**

Nous définissons une fonction ***makeMedicine()*** qui permet à un fabricant de marquer un produit «made». Cette fonction incrémente l'unité de gestion des stocks (sku), crée un nouveau produit et définit ses attributs sur les valeurs transmises.

```
153   function makeMedicine(uint _upc, address _originManufacturerID, string _originFactoryName, string _originFacto
154     onlyManufacturer
155   {
156     // Add the new medicine as part of medicines
157     Medicine memory temp_medicine = Medicine({
158       sku:sku + 1,
159       upc:_upc,
160       ownerID:_originManufacturerID,
161       originManufacturerID:_originManufacturerID,
162       originFactoryName:_originFactoryName,
163       originFactoryInformation:_originFactoryInformation,
164       originFactoryLatitude:_originFactoryLatitude,
165       originFactoryLongitude:_originFactoryLongitude,
166       medicineID:sku+_upc,
167       medicineNotes:_medicineNotes,
168       medicineState:State.Made,
169       medicinePrice:0,
170       distributorID:0,
171       pharmacistID:0,
172       patientID:0
173       });
174     medicines[_upc] = temp_medicine;
175     medicines[_upc].medicineState = State.Made;
176     sku = sku + 1;
177     // Emit the appropriate event
178     emit Made(_upc);
179   }
```

2. **La Fonction packMédecine()**

Nous définissons une fonctions ***packMédecine()*** qui permet à un fabricant de marquer un produit 'Packed[upc]' et nous définissons le modificateur d'appel 'Made[upc]' pour vérifier si le code produit «_upc» passé la phase précédente de la chaîne d'approvisionnement .

```
208     function packMedicine(uint _upc) public
209     // Call modifier to check if upc has passed previous supply chain stage
210     made(_upc)
211     // Call modifier to verify caller of this function
212     onlyManufacturer
213     {
214       // Update the appropriate fields
215       medicines[_upc].medicineState = State.Packed;
216
217       // Emit the appropriate event
218       emit Packed(_upc);
219     }
```

3. **La fonction sellMédecine():**

Nous définissons une fonction ***sellMedicine()*** qui permet à un fabricant de marquer un produit 'ForSal[upc]'. Nous définissons un modificateur pour vérifier si 'upc' a passé l'étape précédente de la chaîne d'approvisionnement 'Packed[upc]'. De plus, nous mettons à jour le prix de vente du produit.

```
222     function sellMedicine(uint _upc, uint _price) public
223     // Call modifier to check if upc has passed previous supply chain stage
224     packed(_upc)
225     // Call modifier to verify caller of this function
226     onlyManufacturer
227
228     {
229       // Update the appropriate fields
230       medicines[_upc].medicineState = State.ForSale;
231       medicines[_upc].medicinePrice = _price;
232       // Emit the appropriate event
233       emit ForSale(_upc);
234     }
```

4. **La fonction buyMédecine()**

Nous définissons une fonction ***buyMedicine()*** qui permet à un distributeur de marquer le produit comme «Sold[upc]». Nous utilisons des modificateurs pour vérifier si le produit est disponible à la vente'ForSal[upc]', si le distributeur a payé suffisamment, et tout excédent d'éther envoyé est remboursé à l'acheteur.

```
239     function buyMedicine(uint _upc) public payable
240       // Call modifier to check if upc has passed previous supply chain stage
241       forSale(_upc)
242       // Call modifer to check if buyer has paid enough
243       paidEnough(medicines[_upc].medicinePrice)
244       // Call modifer to send any excess ether back to buyer
245       checkValue(_upc)
246       //limit to distributers , no end consumers are allowed to buy from factory.
247       onlyDistributor
248       {
249
250         // Update the appropriate fields - ownerID, distributorID, medicineState
251         medicines[_upc].medicineState = State.Sold;
252         // Transfer money to manufacturer
253         medicines[_upc].originManufacturerID.transfer(medicines[_upc].medicinePrice);
254         // emit the appropriate event
255         emit Sold(_upc);
256       }
```

On définit un modificateur ***paidEnough()*** pour vérifier si le montant payé est suffisant pour couvrir le prix. De plus, nous définissons un modificateur ***checkValue()*** pour tester le prix et rembourser le solde restant.

5. **La fonction shipMédecine ()**

Nous déterminons une fonction ***shipMedicine()*** qui permet au fabricant de marquer un produit Shipped[upc] et nous utilisons des modificateurs pour tester si le produit est vendu'Sold[upc]'.

```
260     function shipMedicine(uint _upc) public
261       // Call modifier to check if upc has passed previous supply chain stage
262       sold(_upc)
263       // Call modifier to verify caller of this function
264       onlyManufacturer
265       {
266         //check if the factory is the one making this medicine.
267         require(medicines[_upc].originManufacturerID == msg.sender,"Manufacturers can ship only medicines by them"
268         // Update the appropriate fields
269         medicines[_upc].medicineState = State.Shipped;
270         // Emit the appropriate event
271         emit Shipped(_upc);
272       }
```

6. **La fonction receiveMédecine()**

Nous déterminons une fonction ***receiveMedicine()*** pour permettre de marquer un article 'Received[upc]'. Nous utilisons un modificateur pour vérifier si 'upc' a passé l'offre précédente étape de la chaîne (Shipped[upc]).

```
276     function receiveMedicine(uint _upc) public
277       // Call modifier to check if upc has passed previous supply chain stage
278       shipped(_upc)
279       // Access Control List enforced by calling Smart Contract / DApp
280       {
281       // Update the appropriate fields - ownerID, pharmacistID, medicineState
282       medicines[_upc].medicineState = State.Received;
283
284       // Emit the appropriate event
285       emit Received(_upc);
286     }
287
```

7. **La fonction purchaseMédecine ()**

Nous définissons une fonction ***purchaseMedicine()*** pour permettre à la patient de marquer un article 'purchase[upc]' et nous définissons également des modificateurs pour vérifier si l'article est reçu 'Received[upc]'.

```
290     function purchaseMedicine(uint _upc) public
291       // Call modifier to check if upc has passed previous supply chain stage
292       received(_upc)
293       // Access Control List enforced by calling Smart Contract / DApp
294       onlyPatient
295       {
296         // Up    medicines[_upc].medicineState = State.Shipped;
297         medicines[_upc].medicineState = State.Purchased;
298
299         // Emit the appropriate event
300         emit Purchased(_upc);
301       }
302
```

8. **La fonction fatchMédecinebufferOne ()**

Nous déterminons une fonction fatchMedicinebufferOne() pour permettre de rechercher et afficher les information concernant un '[upc]'.

```
303     // Define a function 'fetchMedicineBufferOne' that fetches the data
304     function fetchMedicineBufferOne(uint _upc) public view returns
305     (
306       uint    medicineSKU,
307       uint    medicineUPC,
308       address ownerID,
309       address originManufacturerID,
310       string  originFactoryName,
311       string  originFactoryInformation,
312       string  originFactoryLatitude
313       // string  originFactoryLongitude
314       )
315     {
316     // Assign values to the 7 parameters
317       return
318     (
319       medicines[_upc].sku,
320       medicines[_upc].upc,
321       medicines[_upc].ownerID,
322       medicines[_upc].originManufacturerID,
323       medicines[_upc].originFactoryName,
324       medicines[_upc].originFactoryInformation,
325       medicines[_upc].originFactoryLatitude
326       // medicines[_upc].originFactoryLongitude
327       );
328   }
```

3.7.2.2 Les contrats intelligent importer dans le contrat principale

Afin de pouvoir mener à bien un projet qui gère correctement la chaîne d'approvisionnement, et pour la garentir des controls d'acces nous devons prendre en compte les comptes des participants tel que les fabricants, les distrubiteurs, les pharmaciens, les patients, pour cette object on va crée un contrat intelligent pour chaque participant :

DistributorRole.sol : pour gérer les comptes des distribiteur.

ManufacturerRole.sol : pour gérer les comptes des fabricants.

PatientRole.sol : pour gérer les comptes des patient .

PharmacistRole.sol : pour gérer les comptes des pharmaciens.

Apartir de chaque contrat intelligent on peut ajouter des comptes et modifier des comptes et supprimer des comptes et peut tester le compte si appartient l'ensemble de participant.

On prend comme exemple **DistributorRole.sol**

```
pragma solidity ^0.4.24;
import "./Roles.sol";
// Define a contract 'DistributorRole' to manage this role - add, remove, check
contract DistributorRole {
  using Roles for Roles.Role;
  event DistributorAdded(address indexed account);
  event DistributorRemoved(address indexed account);
  // Define a struct 'distributors' by inheriting from 'Roles' library, struct Role
  Roles.Role private distributors;
  // In the constructor make the address that deploys this contract the 1st distributor
  constructor() public {
    _addDistributor(msg.sender); }
  modifier onlyDistributor() {
    require(isDistributor(msg.sender));
    _; }
  // Define a function 'isDistributor' to check this role
  function isDistributor(address account) public view returns (bool) {
    return Roles.has(distributors, account); }
  // Define a function 'addDistributor' that adds this role
  function addDistributor(address account) public onlyDistributor {
    _addDistributor(account); }
  function renounceDistributor() public {
    _removeDistributor(msg.sender); }
  // Define an internal function '_addDistributor' to add this role, called by 'addDistributor'
  function _addDistributor(address account) internal {
    Roles.add(distributors, account);
    emit DistributorAdded(account); }
  // Define an internal function '_removeDistributor' to remove this role, called by 'removeDistributor'
  function _removeDistributor(address account) internal {
    Roles.remove(distributors, account);
    emit DistributorRemoved(account);
  }
}
```

3.7.2.3 Compilation et déploiement du contrat intelligent

Tout d'abord, nous compilons le contrat intelligent pour vérifier qu'il fonctionne correctement et qu'il n'y a pas d'erreurs, en utilisant cette commande dans la console du répertoire du projet:

```
$ truffle compile
```

Lorsque nous compilons notre contrat, un nouveau fichier est créé, à l'emplacement suivant: '/ build / contrats / SupplyChain.json '. Ce fichier est l'ABI (Abstract Binary Interface) fichier décrivant la structure spécifique d'un contrat, y compris la fonction d'entrée, fonctions d'interface, liste de paramètres des fonctions, valeur de retour et événements.

Deuxièmement, nous devons rédiger une migration pour déployer notre contrat intelligent sur le réseau de développement Ganache (blockchain locale). Pour ce faire, nous créons un nouveau fichier en migration dossier nommé 2_deploy_contracts.js et nous ajoutons le code suivant:

```
// migrating the appropriate contracts
var ManufacturerRole = artifacts.require("./ManufacturerRole.sol");
var DistributorRole = artifacts.require("./DistributorRole.sol");
var PharmacistRole = artifacts.require("./PharmacistRole.sol");
var PatientRole = artifacts.require("./PatientRole.sol");
var SupplyChain = artifacts.require("./SupplyChain.sol");

module.exports = function(deployer) {
  deployer.deploy(ManufacturerRole);
  deployer.deploy(DistributorRole);
  deployer.deploy(PharmacistRole);
  deployer.deploy(PatientRole);
  deployer.deploy(SupplyChain);
};
```

Après l'exécution de la commande, nous pouvons voir la sortie suivante:

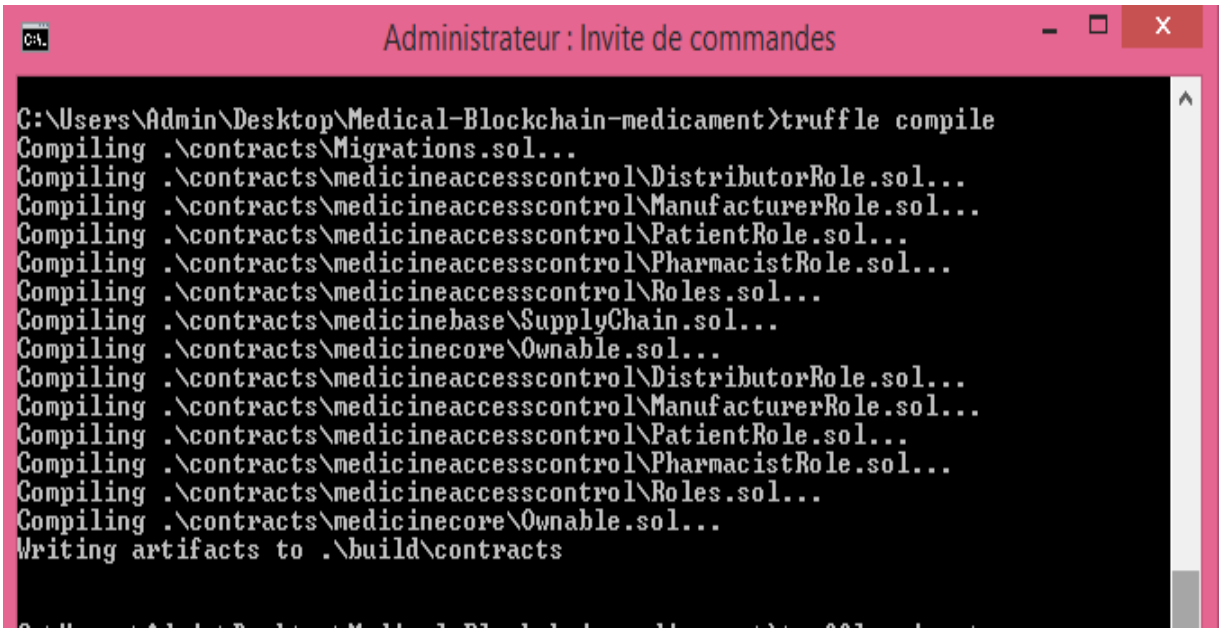


Figure 3.8: Compiler les Smart Contract

Nous pouvons ensuite déployer notre contrat. Nous aurons besoin de se connecter à la blockchain pour le déploiement. Truffle fournit une blockchain personnelle pour des besoins de tests. Nous allons utiliser cette blockchain locale.

Pour créer cette blockchain et interagir avec, ouvrez un nouveau terminal, nous placons vous dans le dossier du projet et exécutez la commande suivante:

```
$truffle develop
```

Cette commande crée et lance la blockchain locale, comme le montre la sortie suivante:

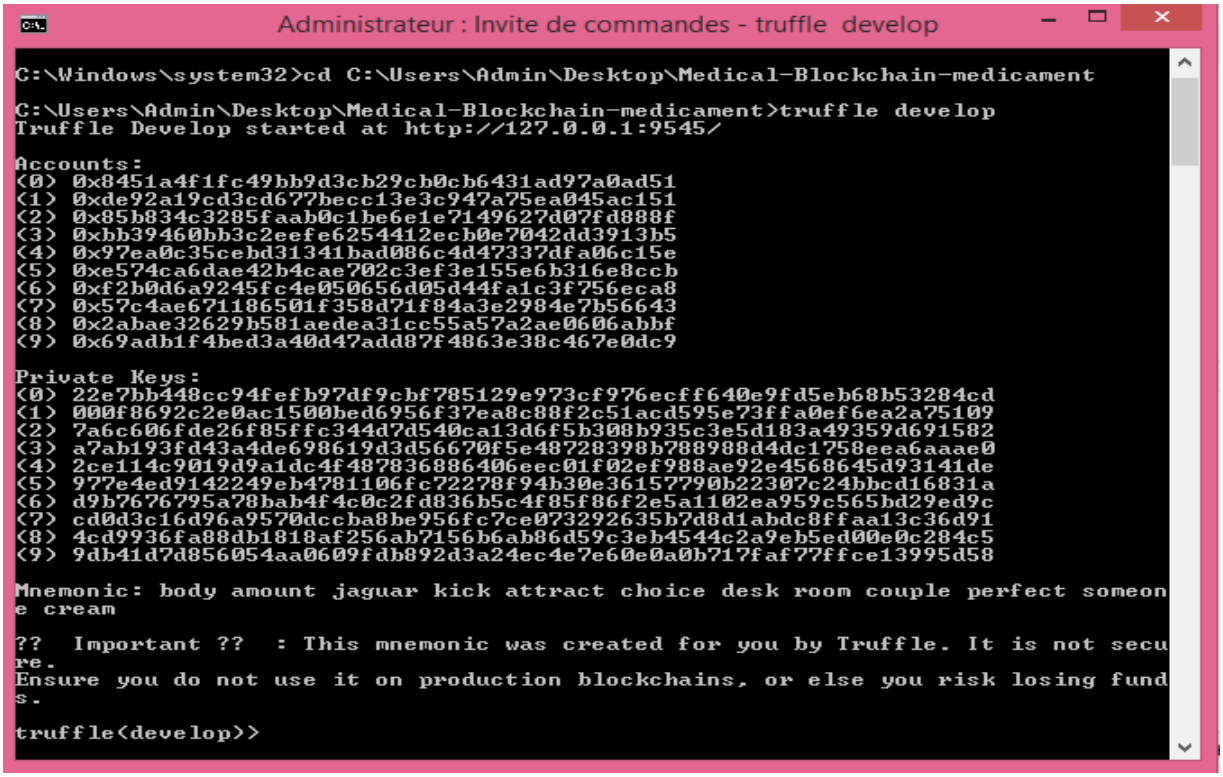


Figure 3.9: Créer un reseau blockchain avec truffle

La sortie montre également que dix comptes (avec les clés privées associées) ont été créés que nous pouvons utiliser lors de l'interaction avec la blockchain.

Ensuite, nous exécutons cette commande:

```
$ truffle  migrate - network development
```

3.7.2.4 Test du contrat intelligent

Les tests de contrats intelligents sont essentiels dans le processus de développement de la blockchain. Le test du contrat intelligent s'assurera que les fonctions auront la bonne manière. La truffle framework rend le processus de test très facile.

Afin de commencer les tests, dans le dossier de test, nous créons un fichier appelé TestSupplyChain.js. et nous créons des tests à l'aide de JavaScript pour simuler l'interaction côté client avec notre smart Contrat. Ensuite, nous l'exécutons pour voir le comportement de notre contrat intelligent à l'aide de cette commande:

```
$ truffle test
```

La figure 3.10 montre la sortie du test de contrat intelligent.

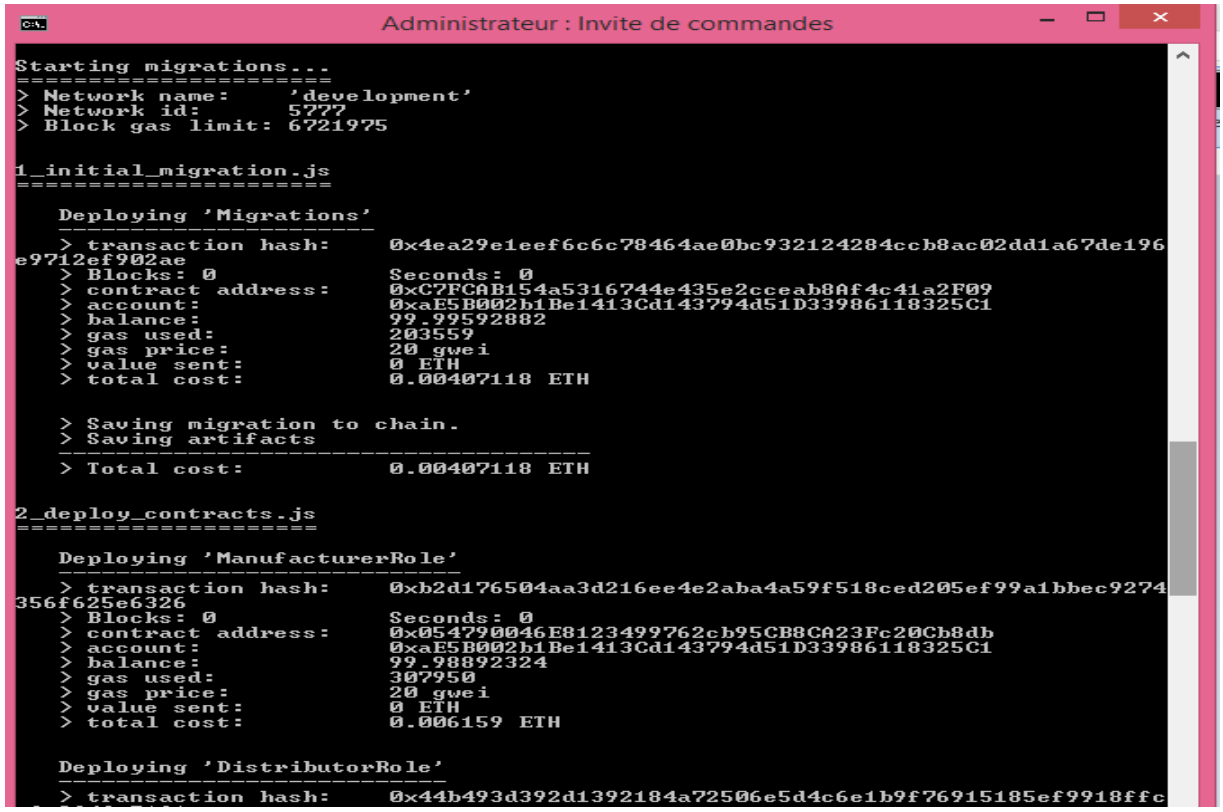
```
C:\Users\Admin\Desktop\Medical-Blockchain-medicament>truffle test
Using network 'development'.

Compiling .\contracts\medicineaccesscontrol\DistributorRole.sol...
Compiling .\contracts\medicineaccesscontrol\ManufacturerRole.sol...
Compiling .\contracts\medicineaccesscontrol\PatientRole.sol...
Compiling .\contracts\medicineaccesscontrol\PharmacistRole.sol...
Compiling .\contracts\medicineaccesscontrol\Roles.sol...
Compiling .\contracts\medicinecore\Ownable.sol...
ganache-cli accounts used here...
Contract Owner: accounts[0]  0xaE5B002b1Be1413Cd143794d51D33986118325C1
Manufacturer: accounts[1]  0x9e7e6E0662E2D52C419c23e179aCf1511A074B04
Distributor: accounts[2]  0x7faF8Cf40C899abb6013702087BE566cB77982b8
Pharmacist: accounts[3]  0x6F19d17Be1a8183705FBcDAfCB3bF5fD8217f8F4
Patient: accounts[4]  0xe6f19468A3aA7e035363cAEb6563e63c883C3e2C


  Contract: SupplyChain
    √ Testing smart contract function makeMedicine() that allows a manufacturer
to make medicine (634ms)
    √ Testing smart contract function packMedicine() that allows a manufacturer
to pack medicine (253ms)
    √ Testing smart contract function sellMedicine() that allows a manufacturer
to sell medicine (245ms)
    √ Testing smart contract function buyMedicine() that allows a distributor to
 buy medicine (256ms)
    √ Testing smart contract function shipMedicine() that allows a distributor t
o ship medicine (429ms)
    √ Testing smart contract function receiveMedicine() that allows a pharmacist
 to mark medicine received (196ms)
    √ Testing smart contract function purchaseMedicine() that allows a patient t
o purchase medicine (250ms)
    √ Testing smart contract function fetchMedicineBufferOne() that allows anyon
e to fetch medicine details from blockchain
    √ Testing smart contract function fetchMedicineBufferTwo() that allows anyon
e to fetch medicine details from blockchain
```

Figure 3.10: Test du contrat intelligent

3.7.2.5 Déploiement de Smart Contract sur Ganache (reseaux local blockchaine)

Pour mettre en place le réseau de développement de la blockchain Ganache, nous mettrons à jour truffle.config comme suit:

```
module.exports = {

  networks: {
      development: {
    host: "127.0.0.1",     // Localhost (default: none)
    port: 8545,            // Standard Ethereum port (default: none)
    network_id: "*",       // Any network (default: none)
    },

     },

  compilers: {
    solc: {
      version: "0.4.24",    // Fetch exact version from solc-bin (default: truffle's version)

      }

  }
}
```



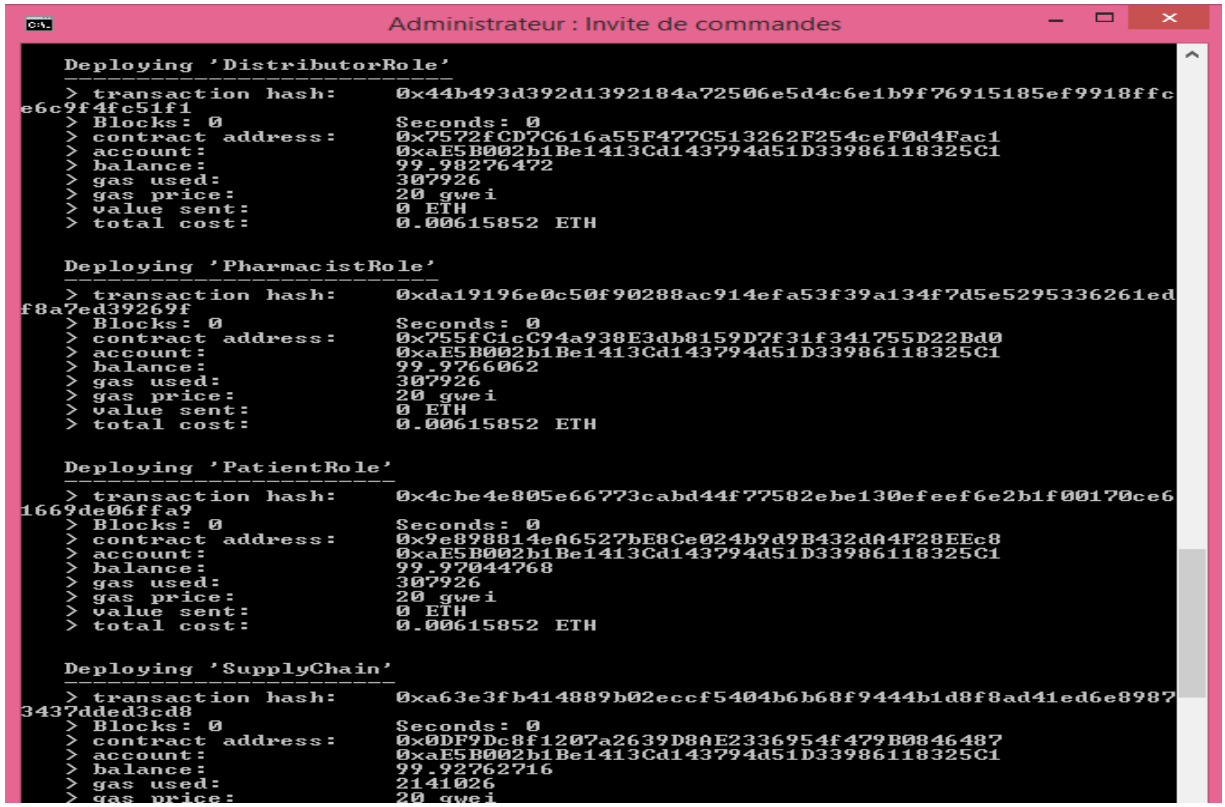


Figure 3.11: Migration smart contract sur GANACHE

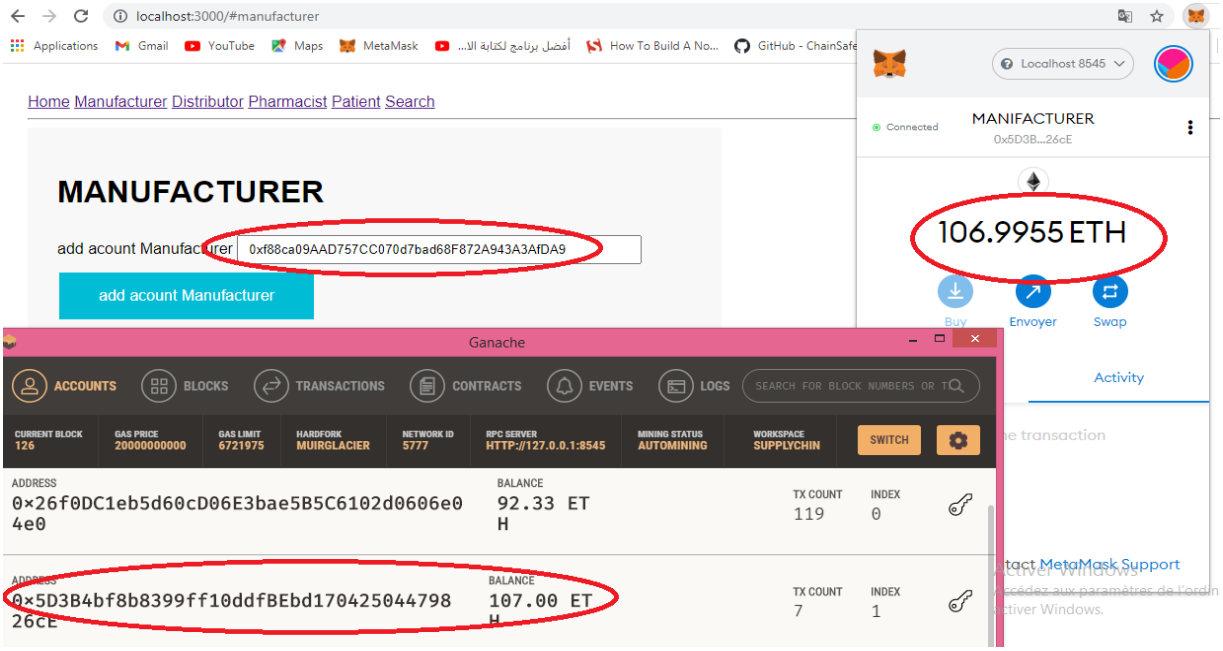


Figure 3.12: Ganache avec Meta mask

3.7.2.6 Déploiement de Smart Contract sur Testnet

Dans cette étape, nous déployons un contrat intelligent sur un véritable réseau Ethereum Blockchain qui coûte réellement de l'éther réel (les transactions sont vérifiées et les blocs sont extraits par de mineurs) en utilisant MetaMask et remix IDE. Mainnet / Testnet - Ceux-ci font référence à deux réseaux Ethereum différents avec des chaînes séparées. Mainnet est le principal Ethereum en direct blockchain (argent réel). Testnet est n'importe quel environnement de test où la fausse monnaie peut être utilisée à la place pour tester les contrats.

Nous effectuons notre test en le déployant sur Testnet qui ne sont qu'une simulation du véritable réseau Ethereum, il existe actuellement trois célèbres Testnet:

Rinkeby: Un test alternatif de Blockchain qui utilise un algorithme de consensus appelé " Preuve d'autorité (PoA) "comme nous l'avons mentionné précédemment dans le processus de consensus du chapitre 1, signifie la nécessité de démontrer l'existence afin de récupérer les éthers à partir d'un robinet, et un temps de génération de bloc fixe.

Kovan: Une blockchain de test alternative, son algorithme de consensus est similaire à Rinkeby mais le temps de génération des blocs est plus rapide.

Ropsten: un test de blockchain similaire à la vraie blockchain Ethereum car il utiliser un algorithme de consensus similaire «Proof of Work (PoW)» (c'est-à-dire qu'il peut être exploité on) donc la simulation des confirmations de transaction est la plus réelle.

Nous utilisons le réseau Ropsten pour déployer et tester le contrat intelligent, nous exigeons que réseau de l'éther gratuit, il est facile d'obtenir de l'éther à partir du robinet de ce réseau. Il suffit de naviguer vers https://faucet.ropsten.be et d'entrer l'adresse du compte de MetaMask. La figure 3.13 montre le robinet Ropsten Ethereum.

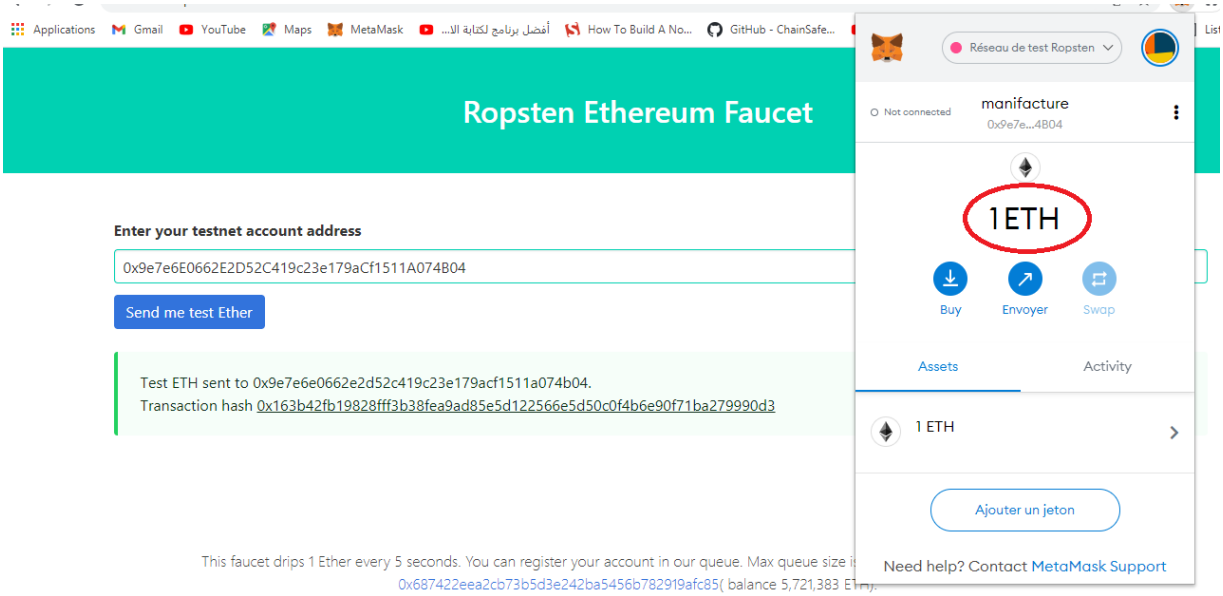


Figure 3.13: Robinet Ropsten Ethereum

Maintenant, nous copions le SupplyChain.sol et le collons dans remix IDE. Comme nous le savions avant, c'est une application Web qui peut être utilisée pour écrire, déboguer et déployer les contrats intelligents (la figure 3.14). Ensuite, nous sélectionnons Injected Web3 sous environnement le compte en Metamask est montré ici sous le compte avec balance éther. Nous déployons notre contrat intelligent sur le Ropsten Testnet, voici le coût requis pour la migration, ainsi que le montre la figure 3.15.

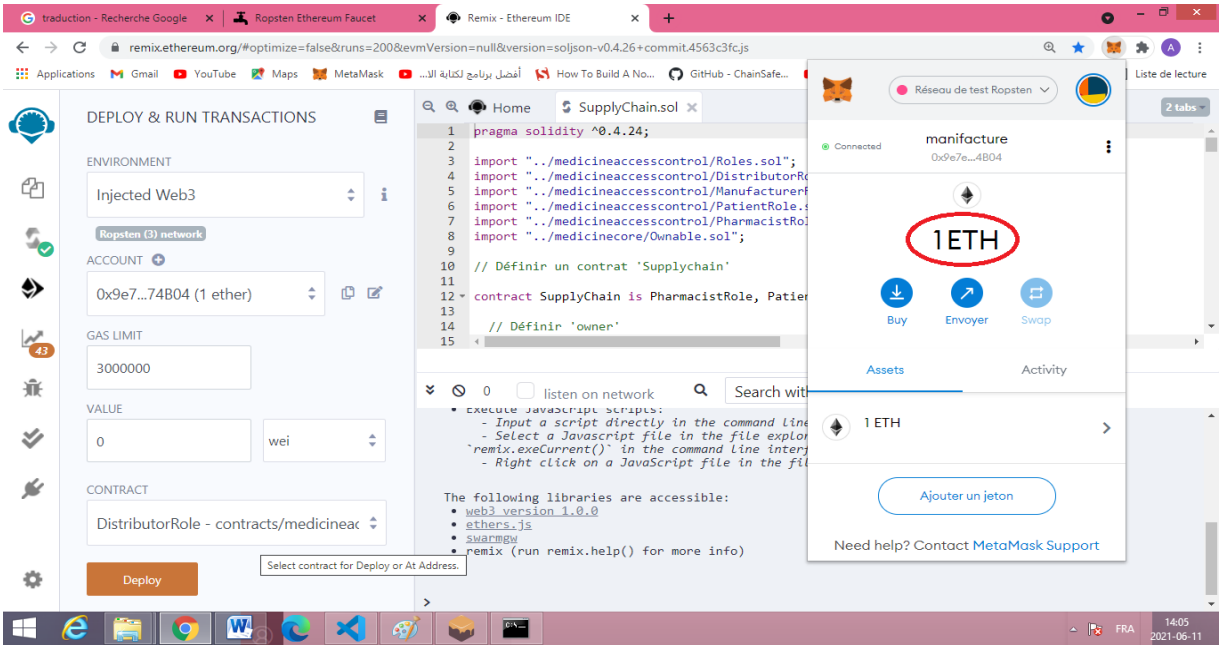


Figure 3.14: Remix IDE Avant Deploy

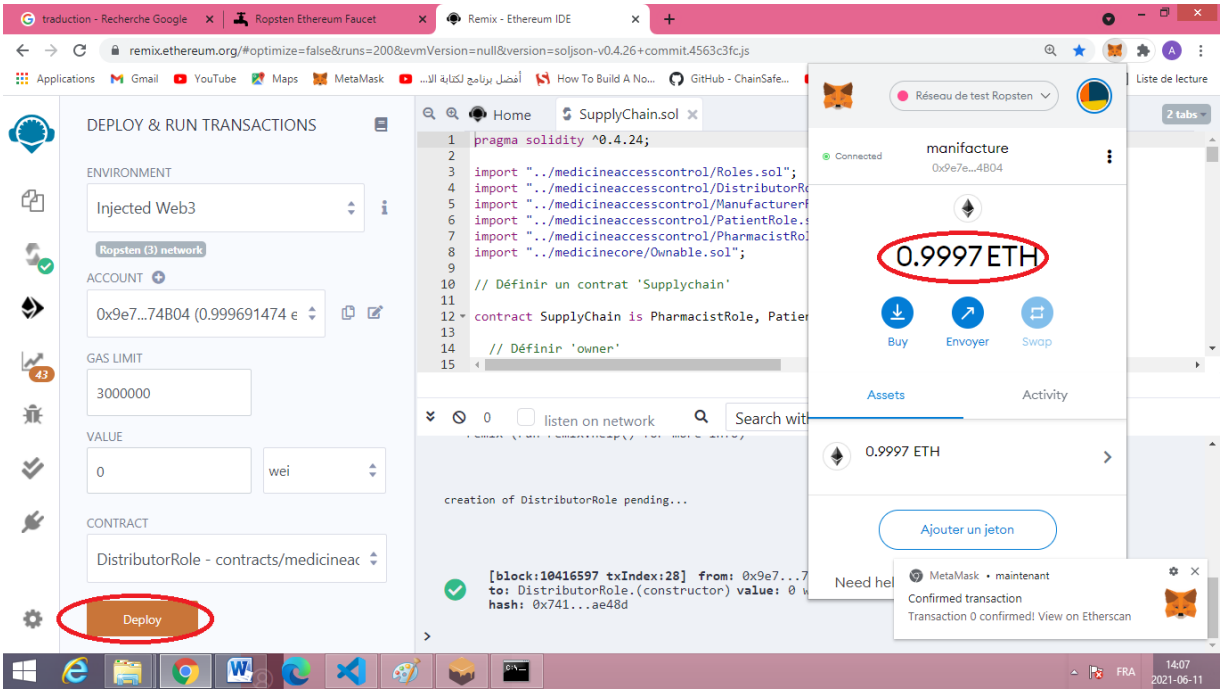


Figure 3.15: Remix IDE Après Deploy

Etherscan permet d'explorer et de rechercher la blockchain Ethereum pour les transactions, adresses, etc. Grâce à Etherscan, nous pouvons accéder à des comptes de contrats intelligents, avec tous Les événements et les transactions (échoués et valides), tout est écrit de manière transparente et de manière régulière depuis sa création, illustrée sur la figure 3.16.

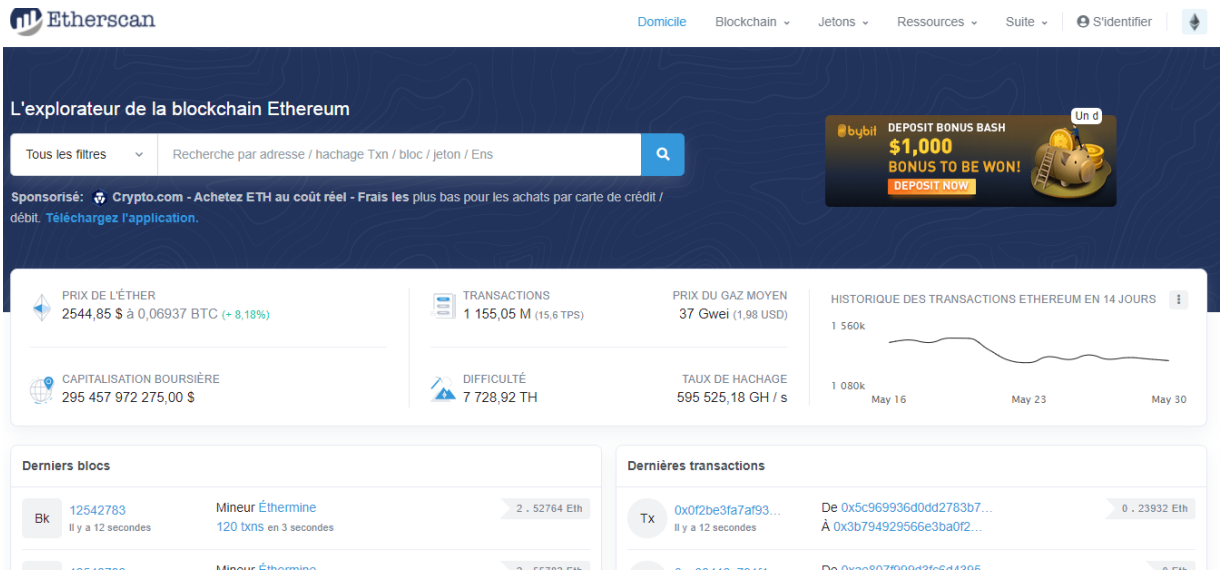


Figure 3.16: https://etherscan.io/.

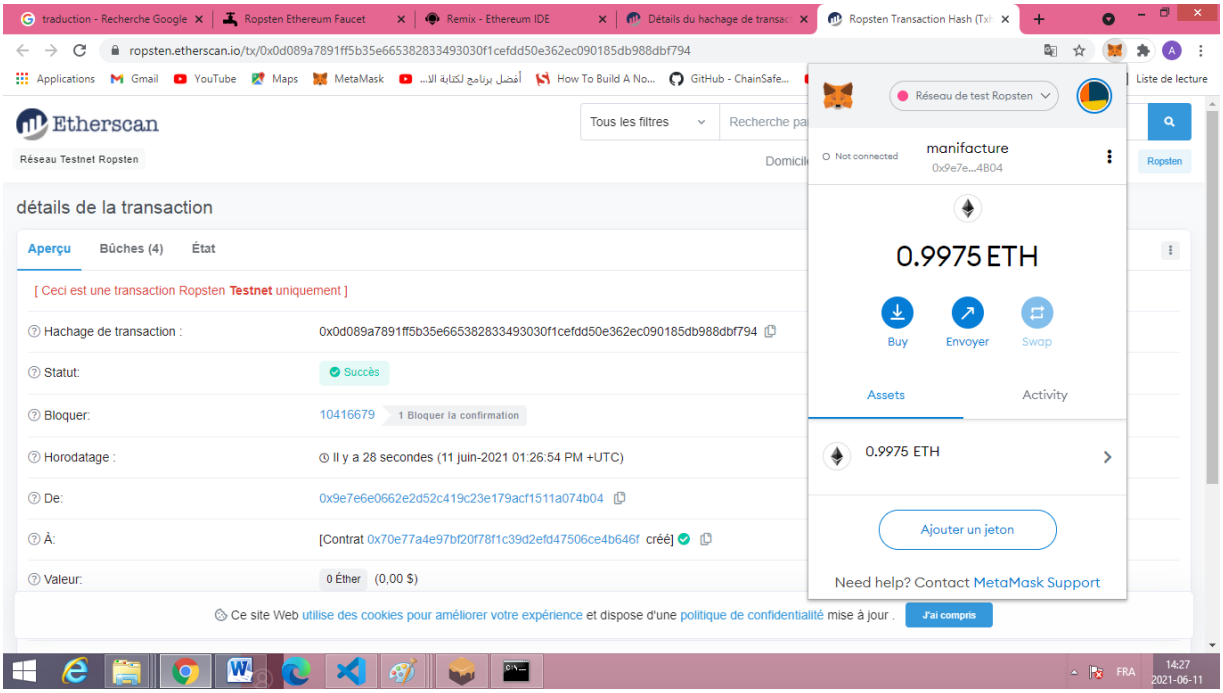


Figure 3.17: Notre compte Smart Contract.

La figure 3.18 montre toutes les principales fonctions de notre contrat intelligent post-lancement que le Remix IDE après le déploiement. Sous le nom et l'adresse du contrat déployé,

Nous avons des boutons dans les couleurs rouge, orange et bleu. Les boutons rouges indiquent les fonctions qui provoquent un pay ethers à un autre participant et l'écriture d'une

transaction, les boutons orange font référence aux fonctions qui écrivent dans la blockchain et qui nécessitent une transaction, où les boutons bleus indiquent la lecture de la blockchain.

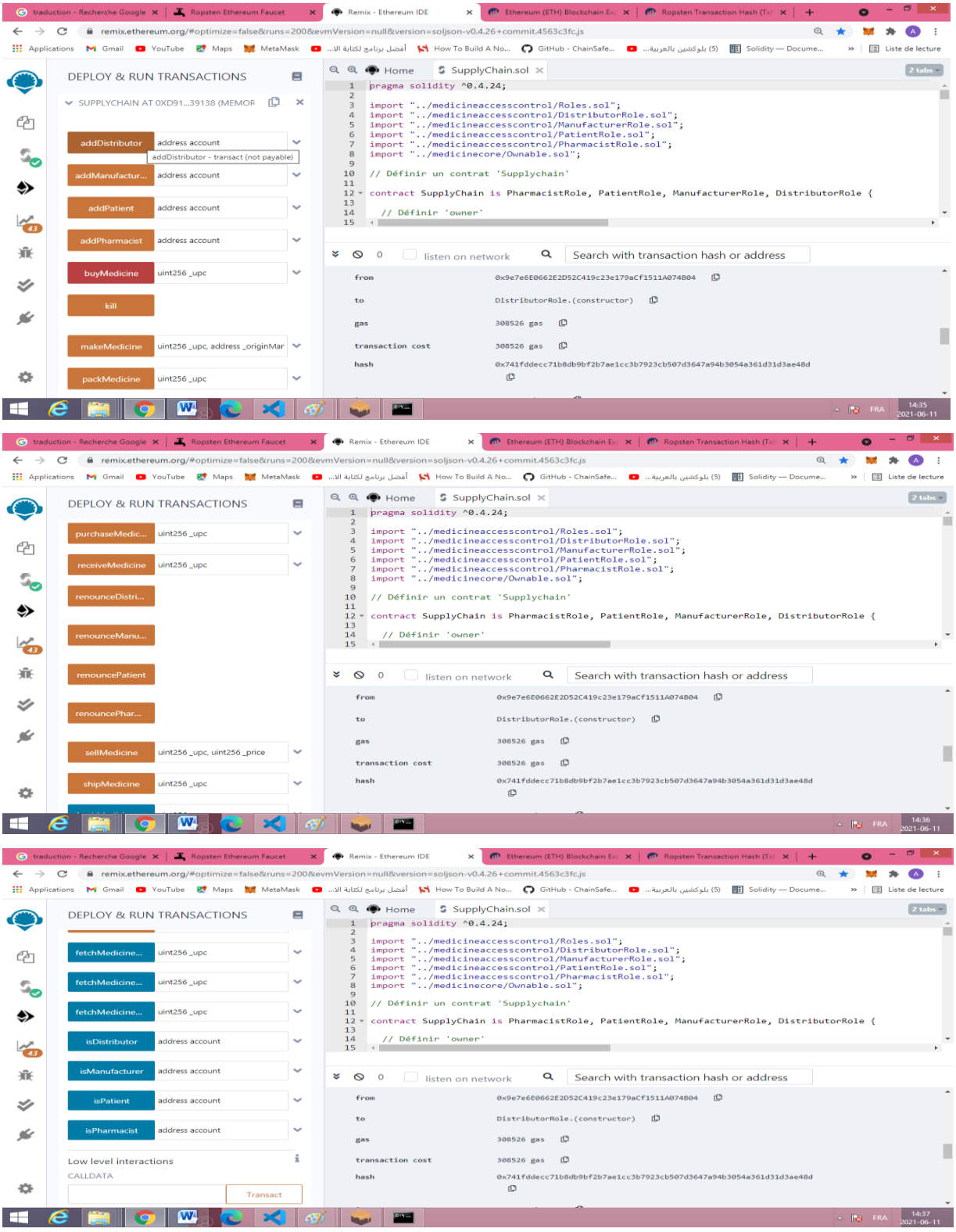


Figure 3.18: Fonctions du contrat intelligent.

3.7.2.7 Des Exemples de test les fonctions de contrat intelligent depuis IDE Remix

1- Fonction Addmanifacturer

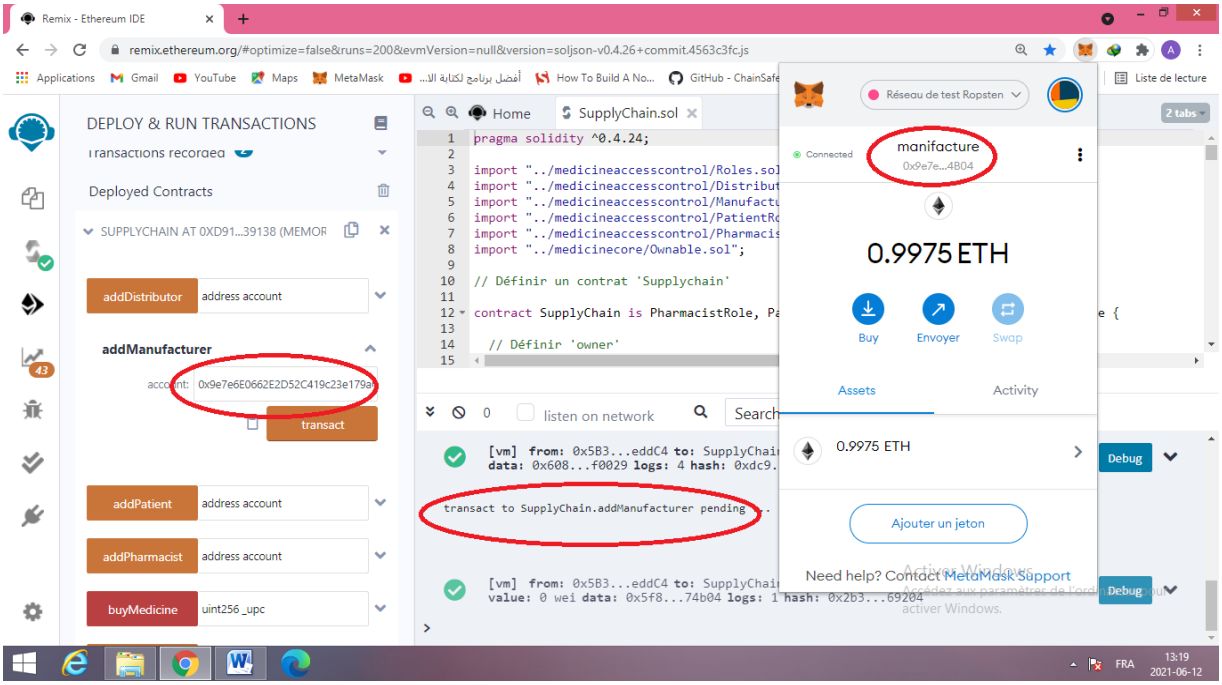


Figure 3.19: Exemple fonctions du contrat intelligent addmanifacturer.

2- Fonction makeMedicine

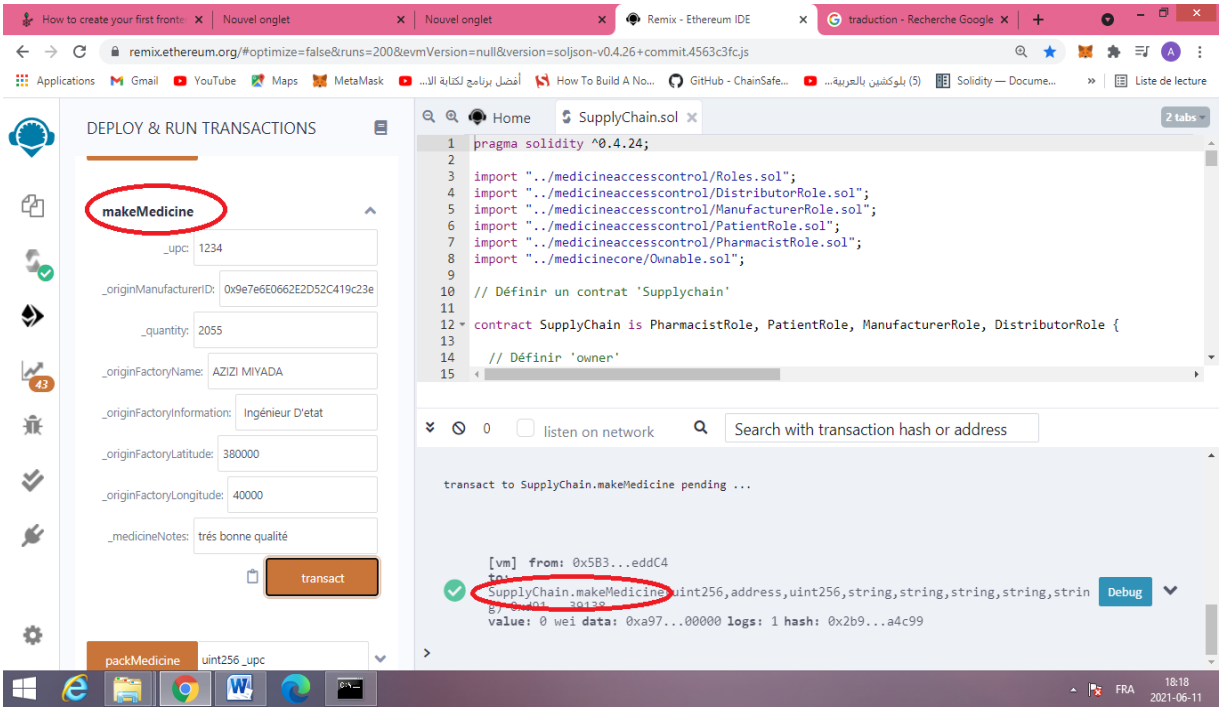


Figure 3.20: Exemple fonctions du contrat intelligent makeMedicine.

3- Fonction fatchMedicineBufferOne

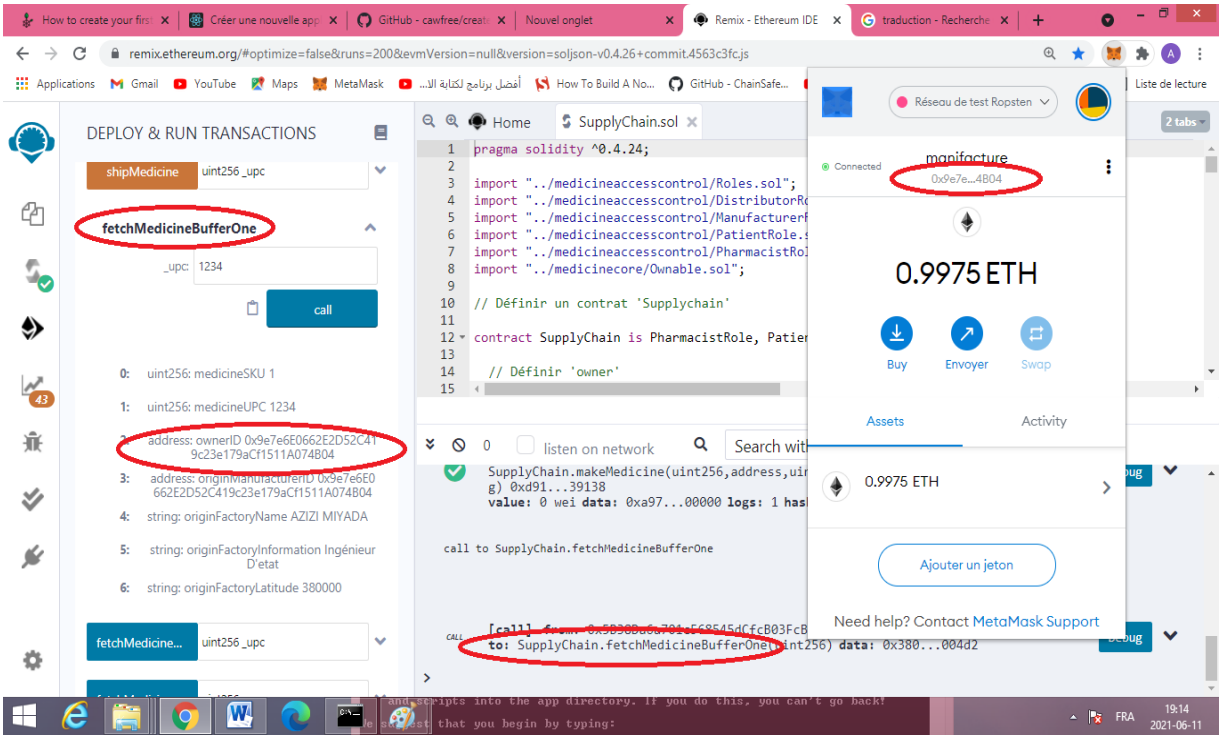


Figure 3.21: Exemple fonctions du contrat intelligent fatchMedicineBufferOne.

4- Fonction packMedicine

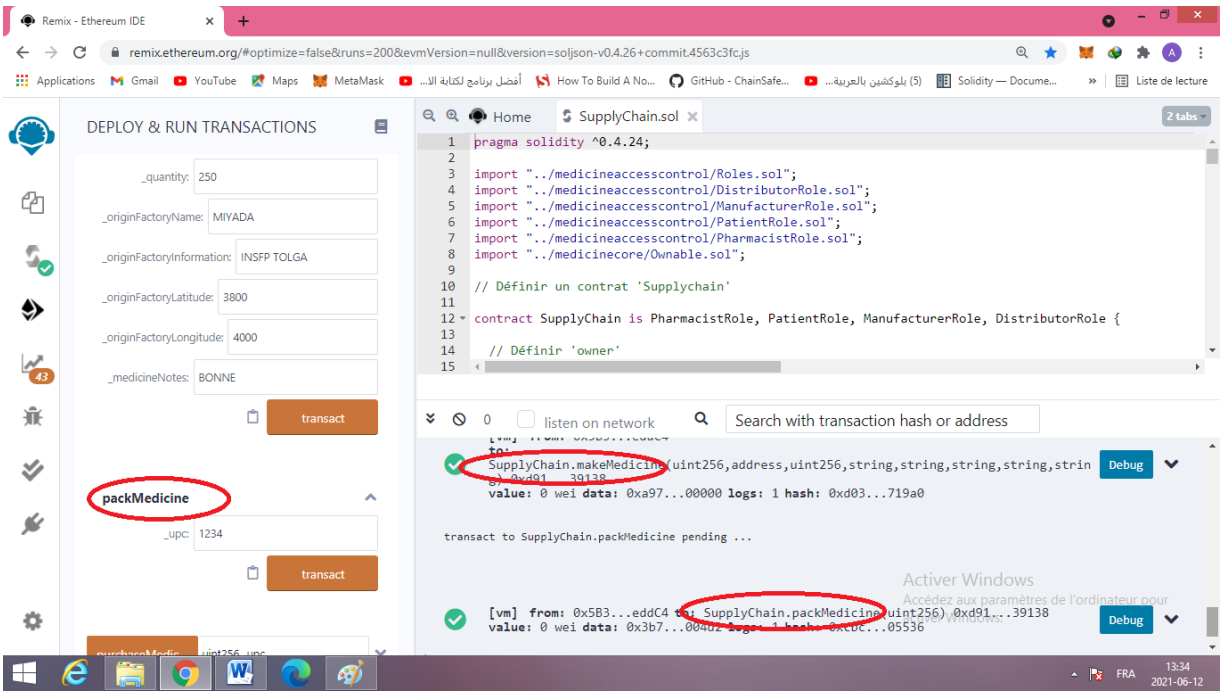


Figure 3.22: Exemple fonctions du contrat intelligent packMedicine.

5- Fonction sellMedicine

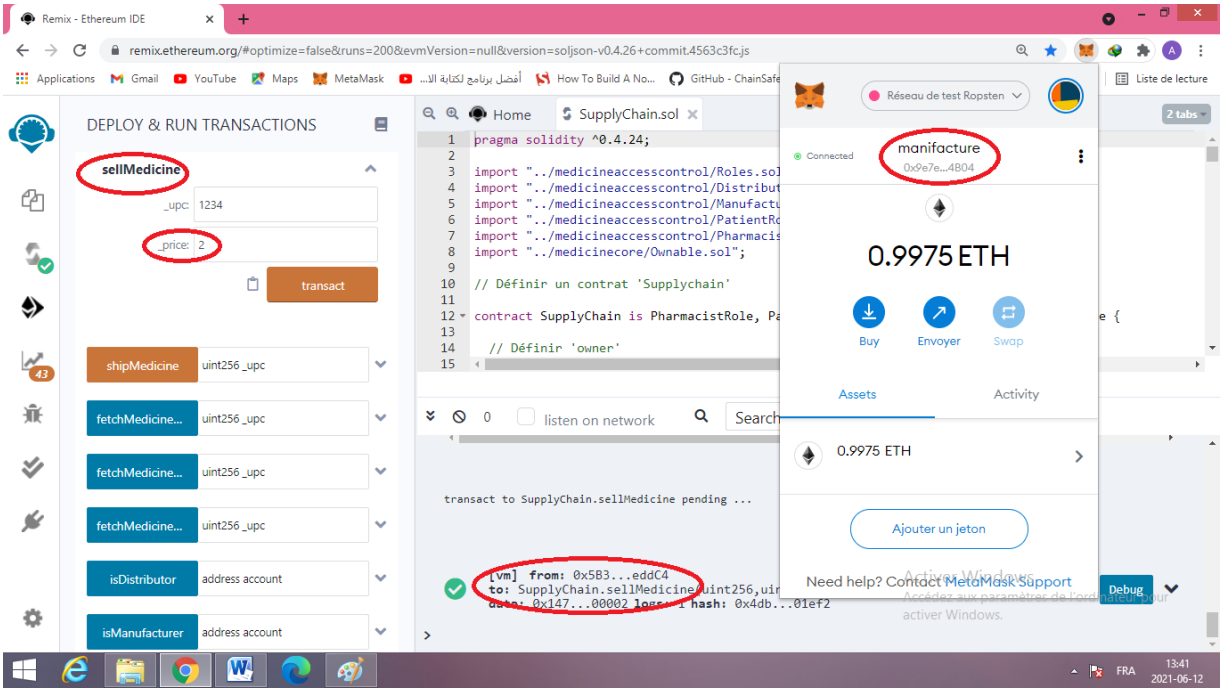


Figure 3.23: Exemple fonctions du contrat intelligent sellMedicine.

3.7.2.1 Web3 et HDWallet Provider

Comme nous l'avons mentionné précédemment, web3 dans les composants systèmes, peut être considéré comme communication entre la bibliothèque web3 et un réseau ethereum, dispose d'un ensemble de méthodes qui permettent à la bibliothèque web3 d'envoyer une requête à un réseau local et de recevoir le réponse à cette demande. En outre, nous utilisons HDWalletProvider qui est un fournisseur pour déverrouiller le compte. Cela nous permet de nous connecter au Rospten qui est hébergé par infura. Web3Js fournit l'objet web3 qui nous permet d'exploiter les fonctions de l'API Web3 en JavaScript.Pour trouver et interagir avec notre blockchain de contrat déployée, il a besoin de connaître les adresse et son interface binaire d'application (ABI).

Le script suivant montre que nous appliquer pour créer une instance web3.

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

Remarque : Pour installer Web3.js taper la commande suivante : **npm install web3**.

Remarque : Pour installer HDWallet Provider taper la commande suivante : **npm i hd-wallet.**

Pour utiliser HDWalletProvider, le Truffle HDWallet doit être installé, puis en cliquant sur à infura.io, nous devons nous enregistrer pour obtenir une clé d'API Infura pour utiliser le service. Par HDWalletProvider avec MetaMask Account Mnemonic et l'API Infura, nous pouvons créer une instance web3 activée du réseau Ropsten. HDWalletProvider prend 2 arguments comme entrée; Le premier est un compte mnémotechnique qui sert à ouvrir des comptes et le second L'argument est le nœud Ethereum que nous voulons connecter.

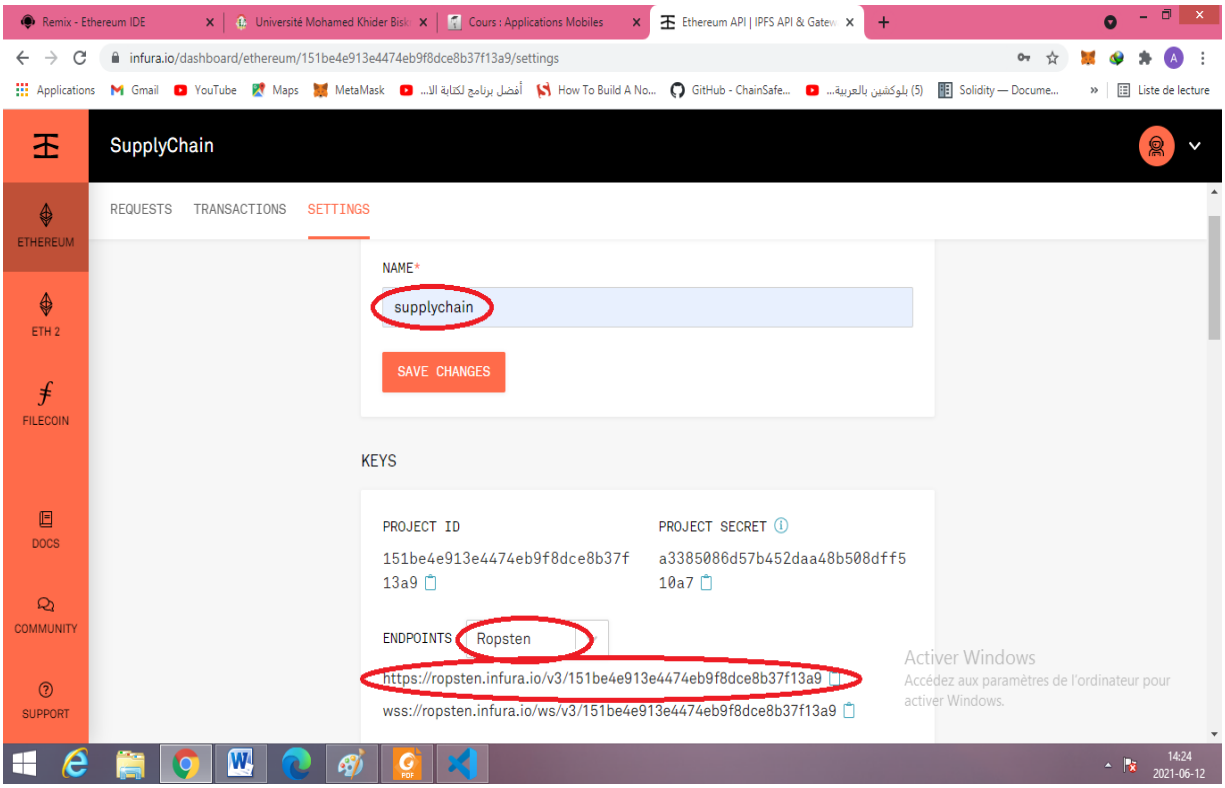


Figure 3.24: Créer nouveau proget infura Ropsten.

Le script montre que nous mettre en œuvre pour se connecter au nœud Ethereum et déverrouiller des comptes avec un compte mnémonique.

```
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/ 151be4e913e4474eb9f8dce8b37f13a9");
const web3 = new Web3 ( provider );
```

Pour interagir avec le contrat de React pour l'implémentation de l'interface système, un instance du contrat a été créée.L'ABI du contrat et l'adresse à laquelle le contrat a été affiché pour créer une copie du contrat.

Remarque : Pour installer une application react dans notre proget taper la commande suivante :

1- **npm i –save-dev truffle create-react app**

2- **npx create-react app dapp**

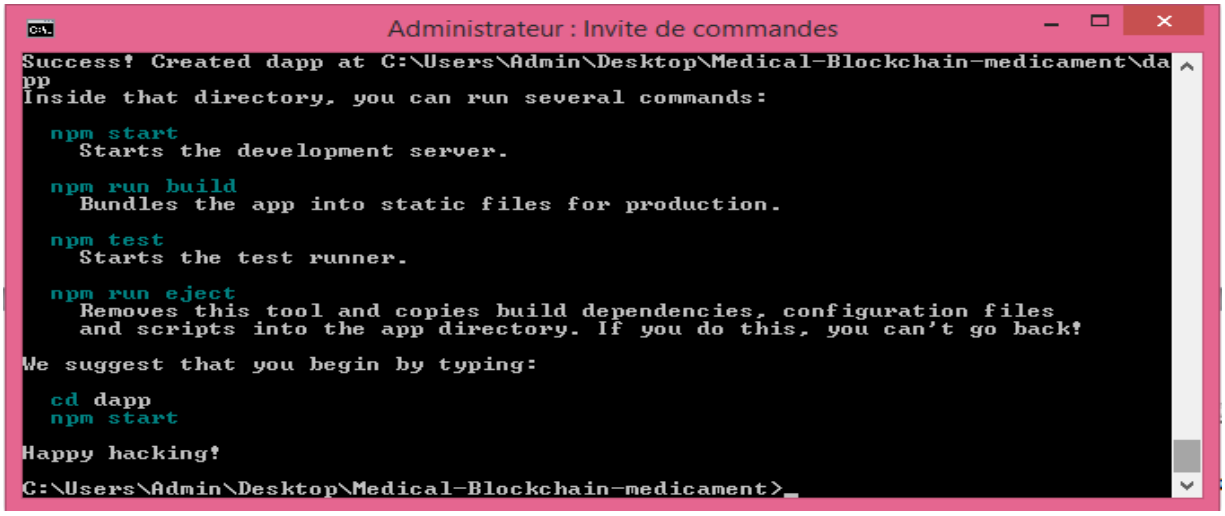


Figure 3.25: La resultat sur le tarminal

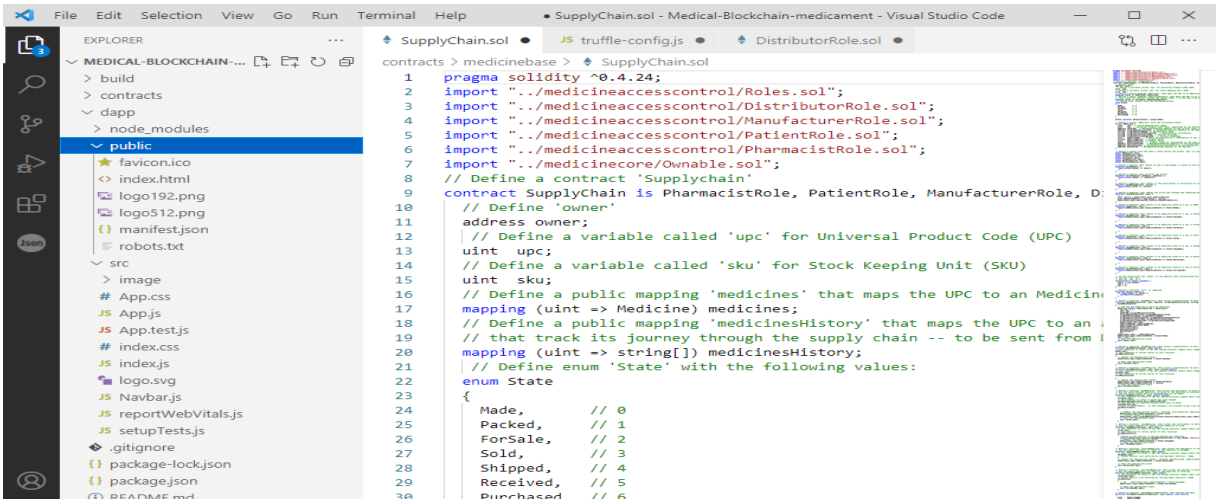


Figure 3.26: Le répertoire aprér l'installation Dapp react

```
// 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 = "0 xED07d16ff28B71f86a5b0F0B526bE36D84f085DA ";
```

3.7.3 Backend et Frontend

Avant le développement du frontend, nous implémentons le backend du système en utilisant node.js et Express.js, qui est un framework Web populaire inconnu écrit en JavaScript et hébergé dans l'environnement d'exécution Node.js. Nous appelons chaque fonction du contrat intelligent dans backend.

Le frontend de ce système est développé en utilisant React, c'est un JavaScript open-source bibliothèque utilisée pour le développement du frontend pour y réagir via un navigateur normal et Redux, est un framework qui a une action et un réducteur que nous utilisons pour appeler l'API backend, puis interagira avec la blockchain Ethereum via l'API Web3.

React est le principal moteur de développement de l'interface utilisateur. Il combine le HTML, CSS et Javascript pour afficher les données à l'écran. Nous insérons un constructeur, qui initialise l'état par défaut et définit l'action de chaque bouton à partir des actions Redux. Nous insérons une méthode de cycle de vie de réaction appelée componentDidMount (), qui exécutera la méthode getAllTransaction () pour obtenir toutes les transactions de Ethereum Blockchain, méthode getAllProduct () pour obtenir tous les produits de la base de données mongoDB. Enfin, nous définissons les composants frontend en utilisant HTML à afficher sur notre page pour interagir avec l'utilisateur.

La figure 3.27 montre la page principale de notre système de chaîne d'approvisionnement en médicaments, la création, la vente, acheter un médicament et consulter l'historique des transactions. Tous les participants de ce réseau peuvent vérifier toutes les étapes de la chaîne d'approvisionnement.

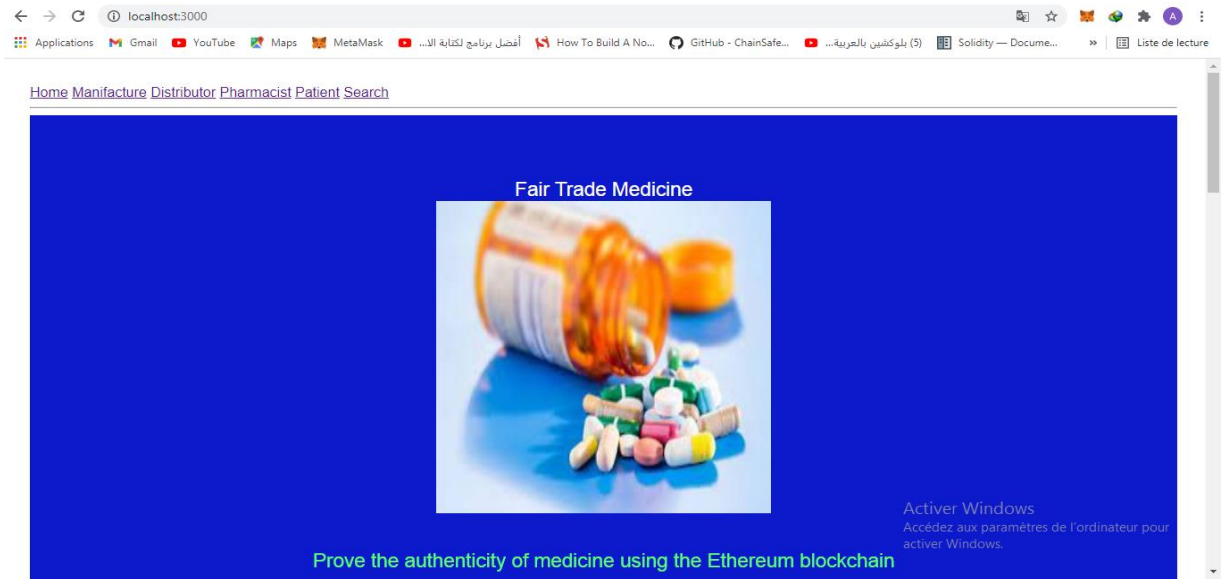


Figure 3.27: La page principal

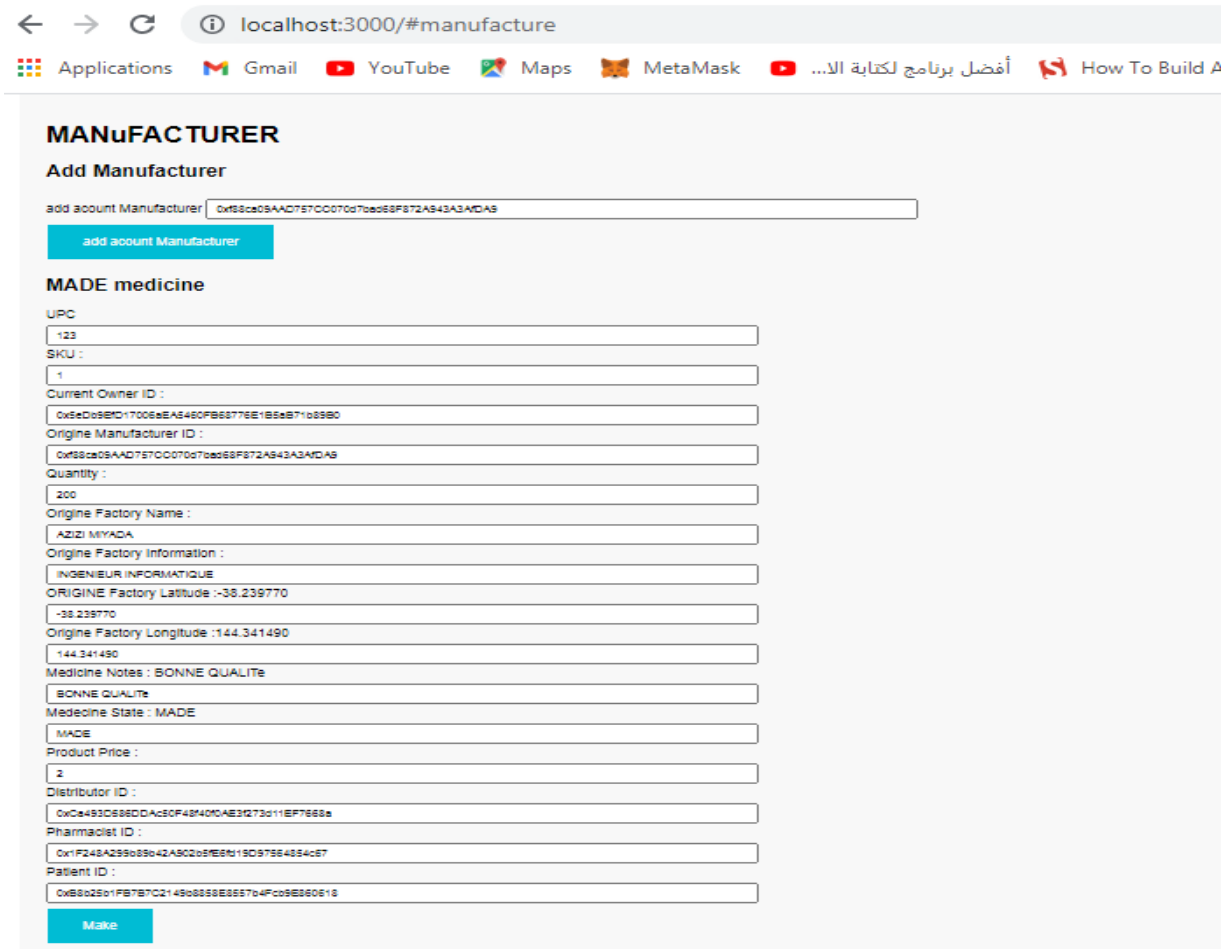


Figure 3.28: La page manufacturer « make medecine »

Dans la page Fabricant, il peut enregistrer un nouveau médicament dans le système en entrant les informations de medicament, y compris le nom et les informations du fabricant, le code du produit, le nom et la quantité du médicament. En cliquant sur le bouton «Make», il fait une transaction contenant toutes les informations sur le médicament à conserver dans la blockchain, et stockés d'une manière ordonnée sur la base de données mongoDB.

Figure 3.29: La page manufacturer « pack medecine »

Ensuite, le fabricant peut changer l'état de médicament vers l'état « packed » après la vérification si le médicament respecté la chaine d'approvisionnement.

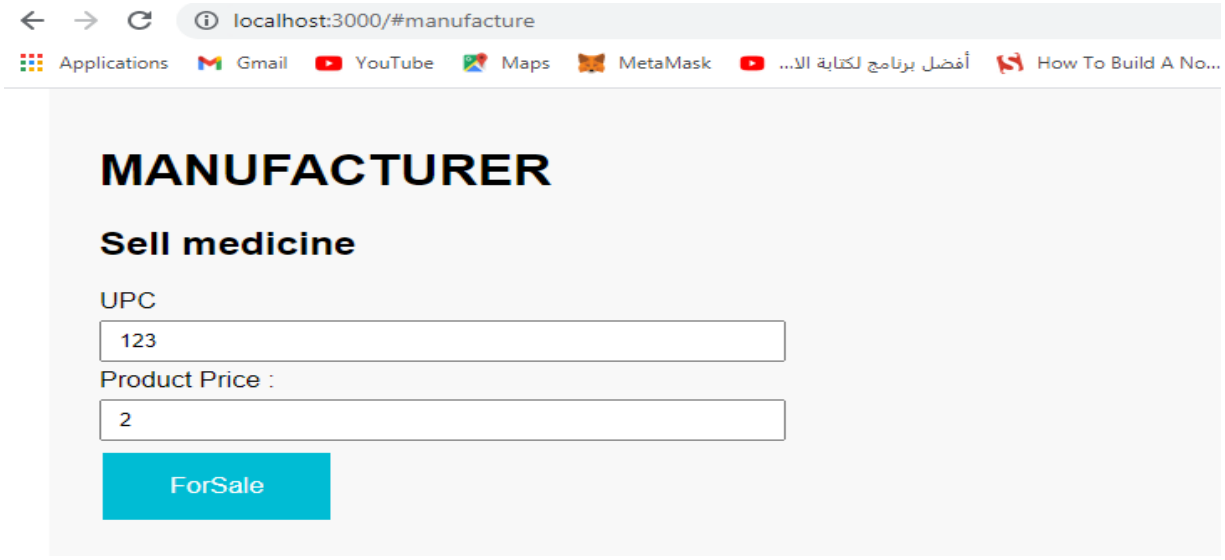


Figure 3.30: La page manufacturer « Sell medecine »

Ensuite, le fabricant peut changer l'état de médicament en état « sell », après la verification si le médicament respecte la chaine d'approvisionnement on ajoute le prix unitaire de médicament.

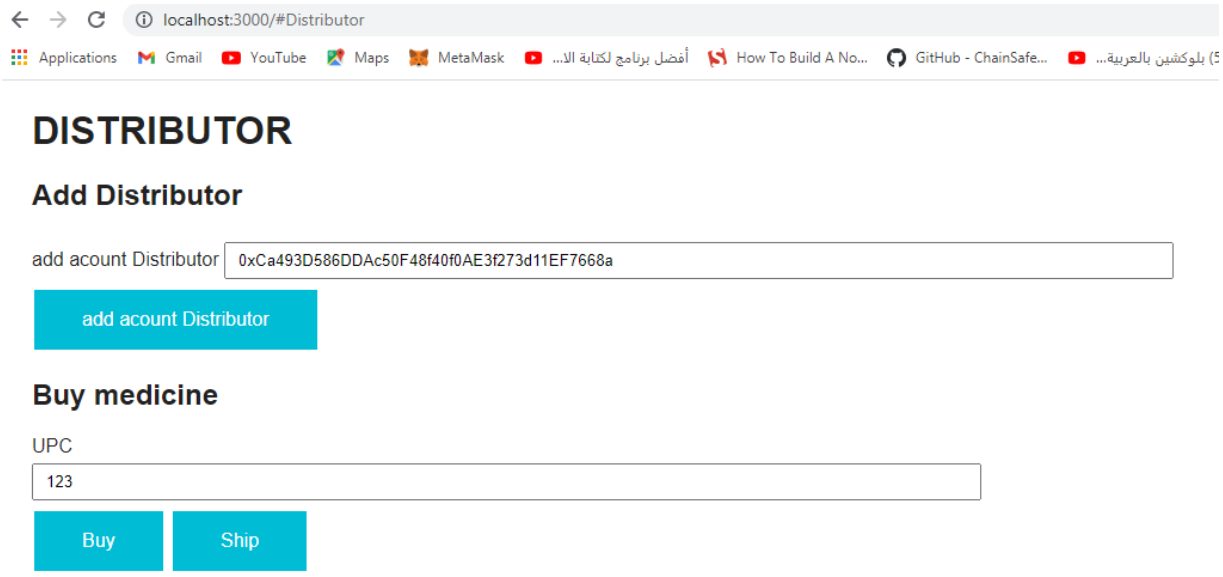


Figure 3.31: La page distributor « Buy . Ship »

Le fabricant poser une medicament en etat « For Sale », le ditributeur peut avoir cette medicament si le solde qui decouvert le prix de cet medicament, apres la verification, la Smart-Contract fait le transfaire monitaire vers le fabricant, l'etat de medicament changer vers « Buy ».

Le ditributeur à acheter le medicament et marqué l'etat de medicament « Ship »,

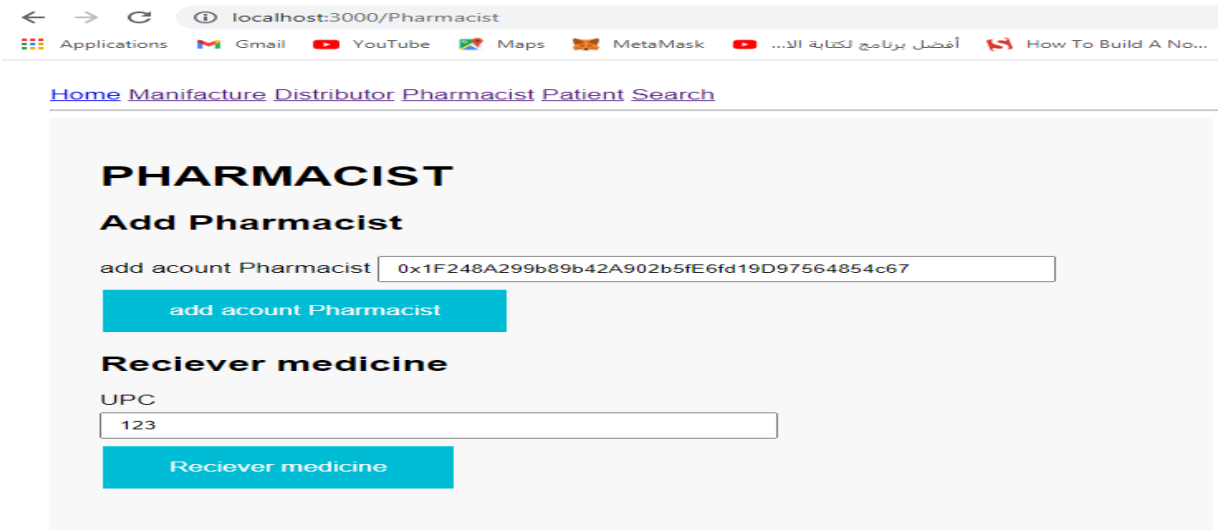


Figure 3.32: La page pharmacist « Reciever medecine »

Apres la verification la chaine d'approvisionnement de medicament depuis la pharmacie, ce dernier vas changer l'etat de medicament vers « Reciever ».

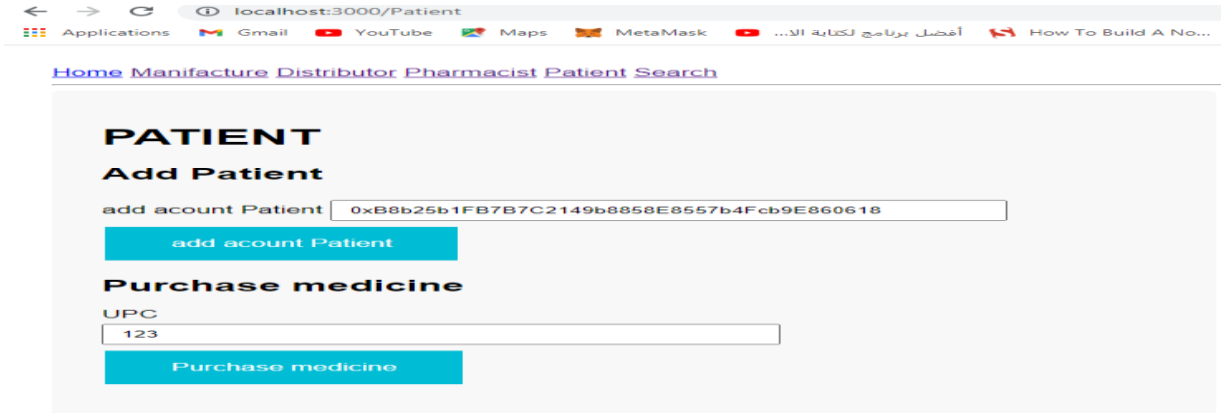


Figure 3.33: La page patient « Purchase medecine »

Apres la verification la chaine d'approvisionnement de medicament depuis le patient, ce dernier vas changer l'etat de medicament vers « Purchase ».

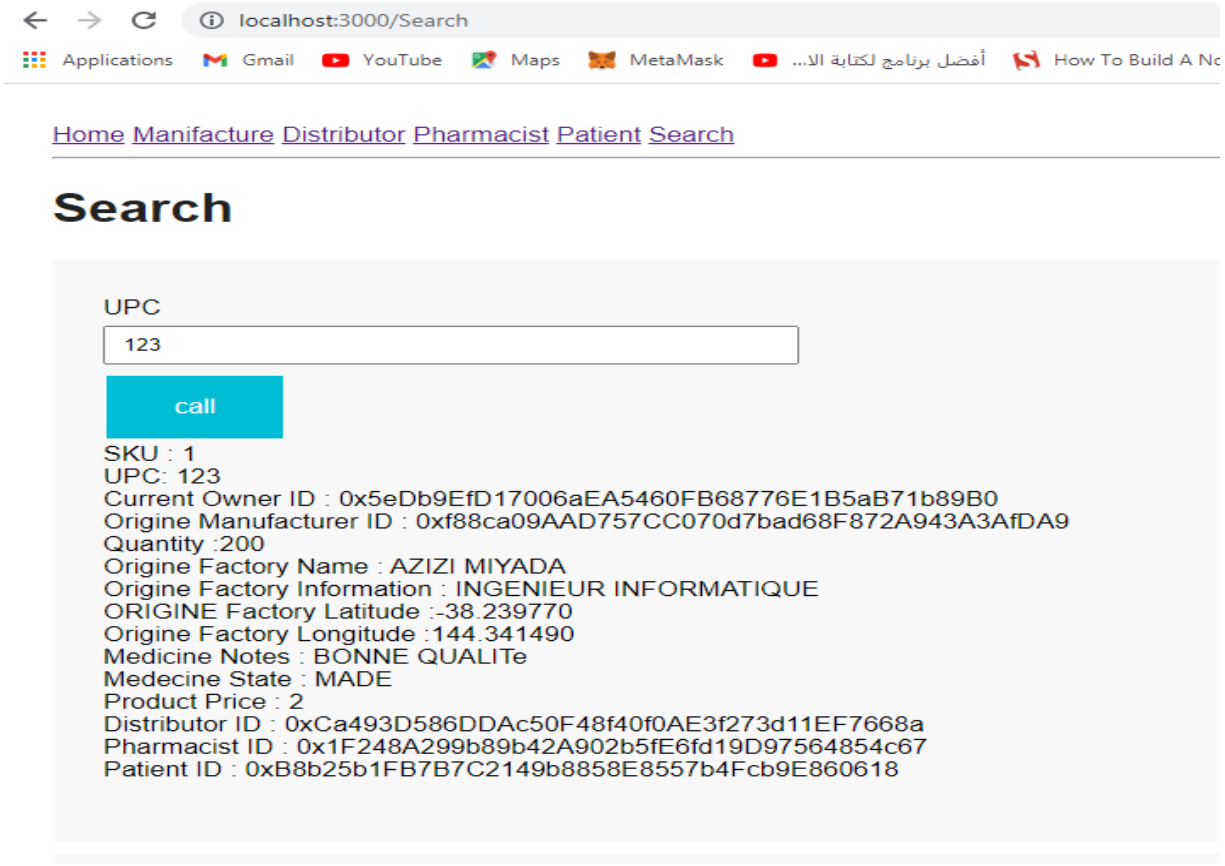


Figure 3.34: La page Search

Depuis l'UPC, en peut avoir tous les informations concernent n'import quel medicamant

3.8 Avantages d'Ethereum pour la chaîne d'approvisionnement en medicaments

Ethereum Blockchain peut aider à résoudre de nombreux problèmes et limitations de la chaîne d'approvisionnement, peut également aider à résoudre la majorité des lacunes de la chaîne d'approvisionnement d'aujourd'hui. applications. Nous énumérerons les limites de la section des défis de la chaîne d'approvisionnement en médicaments, et comment notre système peut aider à résoudre chacun d'eux.

- **Manque de transparence**

La transparence structure la confiance en capturant les points de données clés et fournit libre accès à ces données en général. La technologie blockchain n'a pas d'autorité centrale. Toutes les transactions sont publiées sur la blockchain Ethereum, tout

les participants peuvent vérifier et vérifier toutes les transactions en temps réel.

- **Manque de traçabilité**

La traçabilité améliore les chaînes d'approvisionnement opérationnelles en ayant accès à tous Transactions blockchain Ethereum. Les contrats intelligents sont utilisés pour appliquer les processus de suivi des produits sur la blockchain Ethereum. Tout le monde peut consulter la provenance et le parcours d'un produit en temps réel.

- **Promouvoir la confiance**

Grâce à l'utilisation de la blockchain Ethereum, qui offre transparence et traçabilité avec toute transaction de produits, services, données et ressources nancières, toutes les données sont synchronisées avec toutes les parties prenantes en temps réel, ce qui favorise confiance entre les parties prenantes au sein de la chaîne d'approvisionnement en médicaments.

- **Retards de transport**

Les contrats intelligents permettent d'évaluer l'état actuel de la blockchain et prendre des décisions à la demande, les anciennes solutions nécessitent souvent une interaction humaine, qui dépend des heures d'ouverture spéci ques. La blockchain présente une opportunité pour les contrats intelligents, où le cycle de vie et la fiabilité des conditions de transit de la drogue peuvent être suivis avec des données immuables et variables.

- **Perte de données**

Les contrats intelligents Ethereum définissent les données requises pour chaque transaction et garantissent que tous les participants fournissent la même entrée. Cela signifie

que chaque nœud utilise les mêmes principes, qui ne changent pas d'un participant à l'autre ils se déplacent le long de la chaîne d'approvisionnement.

3.9 Conclusion

Dans le chapitre précédent, nous avons abordé les défis de la chaîne d'approvisionnement en médicaments et analysé les besoins d'une application décentralisée. Dans ce chapitre, nous avons proposé notre solution pour avoir un maximum de transparence, d'intégrité, d'immuabilité et de transpercabilité des données de la chaîne d'approvisionnement sans tiers. Nous avons implémenté notre solution, qui est Smart Contract et une application locale côté client utilisant Web3, nous avons déployé notre contrat dans le réseau de test utilisant l'API Infura, qui est une plate-forme en tant que service pour se connecter au réseau Ropsten qui est la réplique exacte du réseau Ethereum pour mieux tester son comportement.

Après les tests effectués, nous avons prouvé que notre contrat intelligent est capable de gérer et contrôler la chaîne d'approvisionnement en médicaments, elle s'exécute automatiquement lorsque les conditions prédéterminées sont rencontré. Il réduit la complexité dans une chaîne d'approvisionnement grâce à la vérification et à l'exécution automatisées des multiples transactions commerciales impliquées. L'enregistrement immuable et décentralisé garantit également que tous les participants ont un accès égal à l'information, contribue à instaurer la confiance et assure une transparence, une intégrité, une traçabilité complète à un coût élevé sans souci sur l'infrastructure.

Conclusion general

L'utilisation de données collectées à partir de systèmes traditionnels ne peut pas être invoquée car il n'y a pas de garantir que les données ne seront pas falsifiées d'une manière ou d'une autre. Les données sont entièrement sous le contrôle d'une autorité unique qui peut être considérée comme la serrure de porte qui peut être ouvert facilement. Le manque de contrôle sur les données est devenu une préoccupation sérieuse dans le médicament panier d'approvisionnement, car toutes les parties concernées traitent les données en permanence. D'où la nécessité de s'appuyer sur de nouvelles technologies innovantes pour répondre aux besoins des supply chain comme Blockchain, elle a prouvé son efficacité dans le domaine de la sécurité et décentralisation dans différents secteurs d'application à travers le monde, il a apporté de nombreux de nouveaux concepts et idées dans le domaine de la recherche, proposant ainsi une nouvelle voie qui peut apporter divers avantages pour créer une conception significative:

- Réseau décentralisé sans autorité intermédiaire, en s'appuyant sur les transactions et cryptographie à travers tous les nœuds du réseau qui ont une copie de la blockchain et peuvent se connecter les uns aux autres. Immuabilité des données qui ne peuvent être que saisies, elles ne peuvent être ni modi ées ni supprimées par les participants au réseau.
- Transparence du réseau où tous les enregistrements stockés dans la blockchain sont disponibles pour voir ce qui augmente la visibilité et maintient l'ensemble du système cohérent et sécurise.

Dans cette mémoire, une revue a été faite sur la technologie blockchain et nous avons discuté la chaîne d'approvisionnement du médicament et ses problèmes entre la chaîne et les avantages potentiels d'adopter la blockchain pour la gestion de la chaîne d'approvisionnement. Une solution a été proposée pour visualiser la source des produits créés dans le système. La structure détaillée est discutée et mis en œuvre.

Dans ce travail, nous avons cherché à accroître la transparence et la traçabilité en fournissant des sources des produits. Nous nous concentrons principalement sur la transparence car elle apporte de multiples avantages à toutes les entités de la chaîne d'approvisionnement. En augmentant la transparence et la traçabilité, le fabricant est convaincu que les produits sont obtenus fidèlement sans falsification des données. Dans De

plus, d'autres entités ont cette option qui leur permet de parcourir l'historique du produit puisque les données ne sont pas privées et sont accessibles à tous.

Travail du Futur

Les travaux actuels se sont principalement concentrés sur la visualisation des processus de distribution des produits et tirer parti d'un suivi transparent et de contrôles en temps opportun dans la chaîne d'approvisionnement pharmaceutique. Nous avons essayé de suggérer une solution potentielle utilisant la plate-forme Ethereum comme un réseau. Il y a plusieurs parties de son travail qui peuvent être ajoutées ou améliorées pour l'avenir travail et que nous avons mentionnées ci-dessous :

- Ré-implémenter l'application sur d'autres plateformes, nous avons implémenté notre application sur la Blockchain Ethereum. Il existe de nombreuses plateformes différentes qui peuvent être remplacé par certains changements, notamment Hyperledger et Corda. Création la même application sur d'autres plates-formes nous permet de comparer l'efficacité de ces plateformes.
- Utilisez l'API Google Map pour augmenter la visibilité de la chaîne d'approvisionnement en suivant les changements dans le cycle de vie du produit et la visualisation de l'origine à la destination finale.
- Utilisation d'appareils Internet des objets (IoT) pour améliorer les contrôles et les produits traçabilité de la qualité. Par la possibilité d'ajouter des fonctions pouvant utiliser les données reçu directement du produit, sans douter de sa sécurité ou de sa fiabilité, le la saisie des informations sera automatique et plus fiable.

BIBLIOGRAPHIQUE

[1] : Marion PIGNEL , juin 2019 , La Technologie Blockchain Une opportunité pour l'économie sociale, https://www.pourlasolidarite.eu/sites/default/files/publications/files/na-2019-technologie-blockchain_0.pdf.

[2] : Bercy Infos,2019[cité 20/09/2019], Qu'est-ce que la blockchain ? Disponible sur : https://www.economie.gouv.fr/entreprises/blockchain-definition-avantage-utilisation-application

[3] : Oussama Abderraouf Ayadi, Juillet 2019, Projet : *Blockchain et IoMT, Technologie Blockchain CHAPITRE_III_Etat_de_l'art_de_la_Blockchain* , Université Constantine 2 Disponible sur : https://www.researchgate.net/publication/335174496_CHAPITRE_III_Etat_de_l'art_de_la_Blockchain,(54-96).

[4] : Hraoui Said , Faiq Gmira , A W Jarar , Khalid Satori , Fevrer 2019, *Etude-comparative-entre-la-cryptographie.pdf* https://www.researchgate.net/publication/331329705_Etude_comparative_de_deux_cryptosystemes_L'AES_versus_l'attracteur_Chaotique_1 [5] : les notes scientifiques de l'office SENAT. Ensemble National de France, Note N°04-Comprende les BlocksChains. Edition 2018

[6] : SIDI AISSA, IkramKEDDAR, Souria2018[cité 6-jui-2018], *Proposition d'un système à base de blockchain pour la gestion des opérations sur les véhicules au niveau national* ,mémoire de mastre spécialité télécommunication Université Aboubakr Belkaïd– Tlemcen, Disponible sur : http://dspace.univ-tlemcen.dz/handle/112/12896

[7] : Document Co-rédigé par le cabinet d'avocats Solegal & la société Blockchain EZ, Mai 2020, *L'utilité de la signature électronique sur blockchain publique pour les grands comptes : intérêt, faisabilité et valeur juridique*, Disponible sur : https://www.sword-group.com/wp-content/uploads/2020/06/Livre-blanc-Analyse-de-la-signature-lectronique-sur-blockchain-Solegal-Blockchain-EZ.pdf

[8] : Bitcoin. In: Wikipédia [Internet]. 2019 [cité 27 janv 2019]. Disponible sur https://fr.wikipedia.org/w/index.php?title= Bitcoin&oldid=156101655

[9] : Sylvain TESSIER, 2019[cité 27 mai 2019], *FONCTIONNEMENT DE LA BLOCKCHAIN ET SON INTÉRÊT POUR LE MONDE PHARMACEUTIQUE*, Mémoire pour L'obtention du Diplome D'etat de Docteur En Pharmacis, Disponible sur https://dumas.ccsd.cnrs.fr/dumas-02296520/document

[10] : Litecoin. In: Wikipedia [Internet]. 2018 [cité 6 sept 2018]. Disponible sur : https://en.wikipedia.org/w/index.php?title= Litecoin&oldid=857658944

[11] : Peercoin. In: Wikipedia [Internet]. 2018 [cité 6 sept 2018]. Disponible sur : https://en.wikipedia.org/w/index.php?title= Peercoin&oldid=856314793

[12] : Cedric Strub, 2020[cité juin2020], *CONTRIBUTION DE LA BLOCKCHAIN AU MANAGEMENT DES DONNEES DE SANTE* , memoire de diplome D'etat De Docteur En Pharmacie University of Strasbourg, Disponible sur : https://www.researchgate.net/publication/342871864_Contribution_de_la_Blockchain_au_management_des_donnees_de_sante_-_These_d'_Exercice_Cedric_Strub_-_final .pdf

[13] : Paola Oviedo, 2020[cité 16 julliet 2020], *Quel est le rôle de la technologie Blockchain dans la Logistique et la Supply Chain* , Haute École de Gestion de Genève (HEG-GE), Disponible sur :.https://www.latribune.fr/opinions/tribunes/la-blockchain-a-la-rescousse-de-l-industrie-pharmaceutique-760279.html

[14] : BlockchainetSantéUne étude réalisée par Blockchain Partner, *Etude-Sante-industrie-pharmaceutique-Blockchain-Partner.pdf* Disponible sur :. https://blockchainpartner.fr/wp-content/uploads/2017/05/Etude-Sante-industrie-pharmaceutique-Blockchain-Partner.pdf

[15] : *Qu'est-ce que la gestion de la chaîne d'approvisionnement?*, Disponible sur : https://www.oracle.com/ca-fr/scm/what-is-supply-chain-management/

[16] : Dr Liji Thomas, 2020 , *Applications de Blockchain dans la santé,* Disponible sur : https://www.news-medical.net/health/Blockchain-Applications-in-Healthcare-(French).aspx

[17] : *Dossier de Santé Électronique (DSE),* Disponible sur : https://gcblockchain-chainedeblocsgc.github.io/dossier-electronique-sante.html

[18] : Désiré Allechi, Juriste, 2020[cité 12 mai 2020], *l'usage de la blockchain dans le domaine de santé* **,),** Disponible sur **:** https://www.village justice.com/articles/usage-blockchain-dans-domaine-sante,35251.html

[19] : Christine Di Martinelly, A. Guinet, Fouad Riane, 2005[cité juin 2005], *Chaîne logistique en milieu hospitalier : modélisation des processus de distribution de la pharmacie,* Disponible sur https://www.researchgate.net/publication/29606443

[20**]** : Anca Petre, Nassima Haï, 2018, *Numérique et santé (9) Opportunités et enjeux de la technologie blockchain dans le secteur de la santé* ,médecine/sciences 2018 ; 34 : 852-6, Disponible sur **:** https://www.ipubli.inserm.fr/bitstream/handle/10608/9876/MS_2018_10_852.pdf?sequence=1&isAllowed=y

[21] : SANCHEZ, Thomas, 2019[cité23 mai 2019], *La blockchain et le secteur pharmaceutique* , DIPLOME D'ETAT de DOCTEUR EN PHARMACIE,univercité Bordeaux Disponible sur : https://dumas.ccsd.cnrs.fr/dumas-02373905/document

[22] : Leem les entreprise medicament,2017, *Contrefaçon de médicaments, une atteinte à la santé publique* , Disponible sur : https://www.leem.org/sites/default/files/DP-contrefacon-06-07-2017.pdf

[23] : Sunil Chopra, Peter Meindln, 2007, *Supply chain management. Strategy, planning & operation.* In Das summa summarum des managements, pages 265 275, Disponible sur : https://base-logistique-services.com/storage/app/media/Chopra_Meindl_SCM.pdf

[24] : Qu'est-ce que la gestion de la chaîne d'approvisionnement (SCM), Disponible sur :

https://erpblog.iqms.com/what-is-supply-chain-management/, (visite le 20 avril 2021).

[25] : Abby Jenkins, 2020[cité 8octobre 2020], *Qu'est-ce que l'exécution de la chaîne d'approvisionnement (SCE) ?,* Disponible sur :

https://www.netsuite.com/portal/resource/articles/erp/supply-chain-execution-sce.shtml

[26] : Qu'est- ce que le processus de la gestion de la chaine d'approvisinement?, Disponible sur **:** https://www. predictiveanalyticstoday.com/supply-chain-management-process/, (visite le 20 avril 2021).).

[27] : Dani Hao, 2020, *Tout ce que vous devez savoir sur la gestion des commandes et comment accélérer votre calendrier d'approvisionnement*, Disponible sur : https://blog.procurify.com/2018/03/13/need-know-order-management-speed-procurement-timeline/, (visite le 20 avril 2021).

[28] : Gestion de la logistique, Disponible sur : https://www.techopedia.com/definition/13984/logistics-management, (visite le 20 avril 2021).

[29] : Sirine HAMLAOUI, 2020 , *Blockchain for The Drug Supply Chain Management*, Mémoire pour L'obtention du Diplome Mastre UNIV- biskra.

[30] : Nazila Youse, Ahmad Alibabaei, 2015[cité decembre 2015], *Flux d'informations dans la chaîn d'approvisionnement pharmaceutique*, Disponible sur : https://www.researchgate.net/publication/286994116_Information_flow_in_the_pharmaceutical_supply_chain

[31] : *Accord de l'OMC sur les produits pharmaceutiques*, Disponible sur : https://www.wto.org/french/tratop_f/pharma_ag_f/pharma_agreement_f.htm, Organisation mondiale de commerce.

[32] : Mohamed Yassine Ferfera, 2014,*Les effets contrastés de l'intervention des laboratoires pharmaceutiques etrangers dans le secteur algerien de l'industrie pharmaceutique*, Disponible sur :
https://www.ajol.info/index.php/cread/article/view/125577

[33] : Visual Studio Code, Disponible sur :
https://www.01net.com/telecharger/windows/Programmation/creation/fiches/130819.html

[34] : Renaud, 2018[cité 25 février 2018] ,*Tutorial développement Solidity avec Remix*, Disponible sur : https://www.une-blockchain.fr/tutorial-developpement-solidity-remix/

[35] : Hugo Briand, *Industrialiser les smart contracts avec Truffle* , Disponible sur : https://www.ekino.com/articles/industrializing-smart-contracts-with-truffle

[36] : *Esokia réalise des applications web serveurs avec Node.JS !*
Disponible sur : https://esokia.com/fr/node-js

[37] : Wikipédia L'encyclopédie libre que chacun peut améliorer, 2021, Disponible sur :
https://fr.wikipedia.org/wiki/React , La dernière modification de cette page a été faite le 12 mars 2021 à 14:57.

[38] : Qu'est-ce que MetaMask? La façon la plus simple d'utiliser DApps, bit 2me Academy, Disponible sur :
https://academy.bit2me.com/fr/qu%27est-ce-que-metamask-le-moyen-le-plus-simple

[39] : John Agbanusi,2021[cité 19 janvier 2021] , *Comment créer une API Node.js pour Ethereum Blockchain,* Disponible sur :
https://www.smashingmagazine.com/2021/01/nodejs-api-ethereum-blockchain/

Annexe

Code source de Smart Contract

Supplychain.sol/

```
pragma solidity ^0.4.24;
import "../medicineaccesscontrol/Roles.sol";
import "../medicineaccesscontrol/DistributorRole.sol";
import "../medicineaccesscontrol/ManufacturerRole.sol";
import "../medicineaccesscontrol/PatientRole.sol";
import "../medicineaccesscontrol/PharmacistRole.sol";
import "../medicinecore/Ownable.sol";
// Define a contract 'Supplychain'
contract SupplyChain is PharmacistRole, PatientRole, ManufacturerRole, Distrib
utorRole {
  // Define 'owner'
  address owner;
   // Define a variable called 'upc' for Universal Product Code (UPC)
  uint  upc;
  // Define a variable called 'sku' for Stock Keeping Unit (SKU)
  uint  sku;
  // Define a public mapping 'medicines' that maps the UPC to an Medicine.
  mapping (uint => Medicine) medicines;
  // Define a public mapping 'medicinesHistory' that maps the UPC to an array
of TxHash,
  // that track its journey through the supply chain -- to be sent from DApp.
  mapping (uint => string[]) medicinesHistory;
   // Define enum 'State' with the following values:
  enum State
  {
    Made,       // 0
    Packed,     // 1
    ForSale,    // 2
    Sold,       // 3
    Shipped,    // 4
    Received,   // 5
    Purchased   // 6
  }
  State constant defaultState = State.Made;
  // Define a struct 'Medicine' with the following fields:
  struct Medicine {
    uint    sku;  // Stock Keeping Unit (SKU)
    uint    upc; // Universal Product Code (UPC), generated by the Manufacture
r, goes on the package, can be verified by the Patient
```

```
    address ownerID;  // Metamask-
Ethereum address of the current owner as the medicine moves through 8 stages
    address originManufacturerID; // Metamask-
Ethereum address of the Manufacturer
    string  originFactoryName; // Manufacturer Name
    string  originFactoryInformation;  // Manufacturer Information
    string  originFactoryLatitude; // Factory Latitude
    string  originFactoryLongitude;  // Factory Longitude
    uint    medicineID;  // Product ID potentially a combination of upc + sku
    string  medicineNotes; // Product Notes
    uint    medicinePrice; // Product Price
    State   medicineState;  // Product State as represented in the enum above
    address distributorID;  // Metamask-Ethereum address of the Distributor
    address pharmacistID; // Metamask-Ethereum address of the Pharmacist
    address patientID; // Metamask-Ethereum address of the Patient
  }
  // Define 8 events with the same 7 state values and accept 'upc' as input ar
gument
  event Made(uint upc);
  event Packed(uint upc);
  event ForSale(uint upc);
  event Sold(uint upc);
  event Shipped(uint upc);
  event Received(uint upc);
  event Purchased(uint upc);
  // Define a modifer that checks to see if msg.sender == owner of the contrac
t
  modifier onlyOwner() {
    require(msg.sender == owner);
    _;
  }
  // Define a modifer that verifies the Caller
  modifier verifyCaller (address _address) {
    require(msg.sender == _address);
    _;
  }
  // Define a modifier that checks if the paid amount is sufficient to cover t
he price
  modifier paidEnough(uint _price) {
    require(msg.value >= _price);
    _;
  }
    // Define a modifier that checks the price and refunds the remaining balan
ce
  modifier checkValue(uint _upc) {
```

```
    _;
    uint _price = medicines[_upc].medicinePrice;
    uint amountToReturn = msg.value - _price;
    medicines[_upc].patientID.transfer(amountToReturn);
  }
  // Define a modifier that checks if an medicine.state of a upc is Made
  modifier made(uint _upc) {
    require(medicines[_upc].medicineState == State.Made);
    _;
  }
    // Define a modifier that checks if an medicine.state of a upc is Packed
  modifier packed(uint _upc) {
    require(medicines[_upc].medicineState == State.Packed);
    _;
  }
  // Define a modifier that checks if an medicine.state of a upc is ForSale
  modifier forSale(uint _upc) {
    require(medicines[_upc].medicineState == State.ForSale);
    _;
  }
  // Define a modifier that checks if an medicine.state of a upc is Sold
  modifier sold(uint _upc) {
    require(medicines[_upc].medicineState == State.Sold);
    _;
  }
    // Define a modifier that checks if an medicine.state of a upc is Shipped
  modifier shipped(uint _upc) {
    require(medicines[_upc].medicineState == State.Shipped);
    _;
  }
  // Define a modifier that checks if an medicine.state of a upc is Received
  modifier received(uint _upc) {
    require(medicines[_upc].medicineState == State.Received);
    _;
  }
  // Define a modifier that checks if an medicine.state of a upc is Purchased
  modifier purchased(uint _upc) {
    require(medicines[_upc].medicineState == State.Purchased);
    _;
  }
  // In the constructor set 'owner' to the address that instantiated the contr
act
  // and set 'sku' to 1
  // and set 'upc' to 1
  constructor() public payable {
```

```
    owner = msg.sender;
    sku = 0;
    upc = 0;
  }
  // Define a function 'kill' if required
  function kill() public {
    if (msg.sender == owner) {
      selfdestruct(owner);
    }
  }
  // Define a function 'makeMedicine' that allows a manufacturer to mark a med
icine 'Made'
  function makeMedicine(uint _upc, address _originManufacturerID, string _orig
inFactoryName, string _originFactoryInformation, string  _originFactoryLatitud
e, string  _originFactoryLongitude, string  _medicineNotes) public
    onlyManufacturer
  {
    // Add the new medicine as part of medicines
    Medicine memory temp_medicine = Medicine({
      sku:sku + 1,
      upc:_upc,
      ownerID:_originManufacturerID,
      originManufacturerID:_originManufacturerID,
      originFactoryName:_originFactoryName,
      originFactoryInformation:_originFactoryInformation,
      originFactoryLatitude:_originFactoryLatitude,
      originFactoryLongitude:_originFactoryLongitude,
      medicineID:sku+_upc,
      medicineNotes:_medicineNotes,
      medicineState:State.Made,
      medicinePrice:0,
      distributorID:0,
      pharmacistID:0,
      patientID:0
      });
    medicines[_upc] = temp_medicine;
    medicines[_upc].medicineState = State.Made;
    sku = sku + 1;
    // Emit the appropriate event
    emit Made(_upc);
  }
  // Define a function 'packMedicine' that allows a manufacturer to mark an me
dicine 'Packed'
  function packMedicine(uint _upc) public
  // Call modifier to check if upc has passed previous supply chain stage
```

```
  made(_upc)
  // Call modifier to verify caller of this function
  onlyManufacturer
  {
    // Update the appropriate fields
    medicines[_upc].medicineState = State.Packed;
    // Emit the appropriate event
    emit Packed(_upc);
  }
  // Define a function 'sellMedicine' that allows a manufacturer to mark an me
dicine 'ForSale'
  function sellMedicine(uint _upc, uint _price) public
  // Call modifier to check if upc has passed previous supply chain stage
  packed(_upc)
  // Call modifier to verify caller of this function
  onlyManufacturer
  {
    // Update the appropriate fields
    medicines[_upc].medicineState = State.ForSale;
    medicines[_upc].medicinePrice = _price;
    // Emit the appropriate event
    emit ForSale(_upc);
  }
   // Define a function 'buyMedicine' that allows the disributor to mark an me
dicine 'Sold'
  // Use the above defined modifiers to check if the medicine is available for
 sale, if the buyer has paid enough,
  // and any excess ether sent is refunded back to the buyer
  function buyMedicine(uint _upc) public payable
    // Call modifier to check if upc has passed previous supply chain stage
    forSale(_upc)
    // Call modifer to check if buyer has paid enough
    paidEnough(medicines[_upc].medicinePrice)
    // Call modifer to send any excess ether back to buyer
    checkValue(_upc)
    //limit to distributers , no end consumers are allowed to buy from factory
    onlyDistributor
    {

      // Update the appropriate fields - ownerID, distributorID, medicineState
      medicines[_upc].medicineState = State.Sold;
      // Transfer money to manufacturer
      medicines[_upc].originManufacturerID.transfer(medicines[_upc].medicinePr
ice);
      // emit the appropriate event
```

```
      emit Sold(_upc);
    }
  // Define a function 'shipMedicine' that allows the distributor to mark an m
edicine 'Shipped'
  // Use the above modifers to check if the medicine is sold
  function shipMedicine(uint _upc) public
    // Call modifier to check if upc has passed previous supply chain stage
    sold(_upc)
    // Call modifier to verify caller of this function
    onlyManufacturer
    {
      //check if the factory is the one making this medicine.
      require(medicines[_upc].originManufacturerID == msg.sender,"Manufacturer
s can ship only medicines by them");
      // Update the appropriate fields
      medicines[_upc].medicineState = State.Shipped;
      // Emit the appropriate event
      emit Shipped(_upc);
    }
  // Define a function 'receiveMedicine' that allows the pharmacist to mark an
 medicine 'Received'
  // Use the above modifiers to check if the medicine is shipped
  function receiveMedicine(uint _upc) public
    // Call modifier to check if upc has passed previous supply chain stage
    shipped(_upc)
    // Access Control List enforced by calling Smart Contract / DApp
    {
    // Update the appropriate fields - ownerID, pharmacistID, medicineState
    medicines[_upc].medicineState = State.Received;
    // Emit the appropriate event
    emit Received(_upc);
  }
  // Define a function 'purchaseMedicine' that allows the patient to mark an m
edicine 'Purchased'
  // Use the above modifiers to check if the medicine is received
  function purchaseMedicine(uint _upc) public
    // Call modifier to check if upc has passed previous supply chain stage
    received(_upc)
    // Access Control List enforced by calling Smart Contract / DApp
    onlyPatient
    {
      // Up    medicines[_upc].medicineState = State.Shipped;
      medicines[_upc].medicineState = State.Purchased;
      // Emit the appropriate event
      emit Purchased(_upc);
```

```
    }
  // Define a function 'fetchMedicineBufferOne' that fetches the data
  function fetchMedicineBufferOne(uint _upc) public view returns
  (
    uint    medicineSKU,
    uint    medicineUPC,
    address ownerID,
    address originManufacturerID,
    string  originFactoryName,
    string  originFactoryInformation,
    string  originFactoryLatitude
    // string  originFactoryLongitude
    )
  {
  // Assign values to the 7 parameters
    return
  (
    medicines[_upc].sku,
    medicines[_upc].upc,
    medicines[_upc].ownerID,
    medicines[_upc].originManufacturerID,
    medicines[_upc].originFactoryName,
    medicines[_upc].originFactoryInformation,
    medicines[_upc].originFactoryLatitude
    // medicines[_upc].originFactoryLongitude
    );
}
  // Define a function 'fetchMedicineBufferTwo' that fetches the data
  function fetchMedicineBufferTwo(uint _upc) public view returns
  (
    string  originFactoryLongitude,
    string  medicineNotes,
    uint    medicinePrice,
    State    medicineState,
    address distributorID,
    address pharmacistID,
    address patientID
    )
  {
    // Assign values to the 7 parameters
    return
    (
      medicines[_upc].originFactoryLongitude,
      medicines[_upc].medicineNotes,
      medicines[_upc].medicinePrice,
```

```
      medicines[_upc].medicineState,
      medicines[_upc].distributorID,
      medicines[_upc].pharmacistID,
      medicines[_upc].patientID
      );
  }
  // Define a function 'fetchMedicineBufferThree' that fetches the data
  function fetchMedicineBufferThree(uint _upc) public view returns
  (
    uint    medicineSKU,
    uint    medicineUPC,
    uint    medicineID
    )
  {
    // Assign values to the 3 parameters
        return
    (
      medicines[_upc].sku,
      medicines[_upc].upc,
      medicines[_upc].medicineID
      );
  }
*************************************************************************
```

DistributorRole.sol

```
pragma solidity ^0.4.24;
import "./Roles.sol";
// Define a contract 'DistributorRole' to manage this role -
 add, remove, check
contract DistributorRole {
  using Roles for Roles.Role;
  event DistributorAdded(address indexed account);
  event DistributorRemoved(address indexed account);
  // Define a struct 'distributors' by inheriting from 'Roles' library, struct
 Role
  Roles.Role private distributors;
  // In the constructor make the address that deploys this contract the 1st di
stributor
  constructor() public {
    _addDistributor(msg.sender); }
  modifier onlyDistributor() {
    require(isDistributor(msg.sender));
    _; }
  // Define a function 'isDistributor' to check this role
  function isDistributor(address account) public view returns (bool) {
    return Roles.has(distributors, account); }
```

```
  // Define a function 'addDistributor' that adds this role
  function addDistributor(address account) public onlyDistributor {
    _addDistributor(account); }
  function renounceDistributor() public {
    _removeDistributor(msg.sender); }
  // Define an internal function '_addDistributor' to add this role, called by
 'addDistributor'
  function _addDistributor(address account) internal {
    Roles.add(distributors, account);
    emit DistributorAdded(account); }
  // Define an internal function '_removeDistributor' to remove this role, cal
led by 'removeDistributor'
  function _removeDistributor(address account) internal {
    Roles.remove(distributors, account);
    emit DistributorRemoved(account);
  }
}
**************************************************************************
```

ManufacturerRole.sol

```
pragma solidity ^0.4.24;
// Import the library 'Roles'
import "./Roles.sol";
// Define a contract 'ManufacturerRole' to manage this role -
 add, remove, check
contract ManufacturerRole{
  using Roles for Roles.Role;
  // Define 2 events, one for Adding, and other for Removing
  event ManufacturerAdded(address indexed account);
  event ManufacturerRemoved(address indexed account);
  // Define a struct 'manufacturers' by inheriting from 'Roles' library, struc
t Role
  Roles.Role private manufacturers;
  // In the constructor make the address that deploys this contract the 1st ma
nufacturer
  constructor() public {
    _addManufacturer(msg.sender);
  }
  // Define a modifier that checks to see if msg.sender has the appropriate ro
le
  modifier onlyManufacturer() {
    require(isManufacturer(msg.sender));
    _;
  }
  // Define a function 'isManufacturer' to check this role
  function isManufacturer(address account) public view returns (bool) {
```

```
    return Roles.has(manufacturers, account);
  }
  // Define a function 'addManufacturer' that adds this role
  function addManufacturer(address account) public onlyManufacturer {
    _addManufacturer(account);
  }
  // Define a function 'renounceManufacturer' to renounce this role
  function renounceManufacturer() public {
    _removeManufacturer(msg.sender);
  }
  // Define an internal function '_addManufacturer' to add this role, called b
y 'addManufacturer'
  function _addManufacturer(address account) internal {
    Roles.add(manufacturers, account);
    emit ManufacturerAdded(account);
  }
  // Define an internal function '_removeManufacturer' to remove this role, ca
lled by 'removeManufacturer'
  function _removeManufacturer(address account) internal {
    Roles.remove(manufacturers, account);
    emit ManufacturerRemoved(account);
  }
}
**************************************************************************
```

PatientRole.sol

```
pragma solidity ^0.4.24;

// Import the library 'Roles'
import "./Roles.sol";

// Define a contract 'PatientRole' to manage this role - add, remove, check
contract PatientRole {
  using Roles for Roles.Role;

  // Define 2 events, one for Adding, and other for Removing
  event PatientAdded(address indexed account);
  event PatientRemoved(address indexed account);
  // Define a struct 'patients' by inheriting from 'Roles' library, struct Rol
e
  Roles.Role private patients;
  // In the constructor make the address that deploys this contract the 1st pa
tient
  constructor() public {
    _addPatient(msg.sender);
```

```
  }
  // Define a modifier that checks to see if msg.sender has the appropriate ro
le
  modifier onlyPatient() {
    require(isPatient(msg.sender));
    _;
  }
  // Define a function 'isPatient' to check this role
  function isPatient(address account) public view returns (bool) {
    return Roles.has(patients, account);
  }
  // Define a function 'addPatient' that adds this role
  function addPatient(address account) public onlyPatient {
    _addPatient(account);
  }
  // Define a function 'renouncePatient' to renounce this role
  function renouncePatient() public {
    _removePatient(msg.sender);
  }
  // Define an internal function '_addPatient' to add this role, called by 'ad
dPatient'
  function _addPatient(address account) internal {
    Roles.add(patients, account);
    emit PatientAdded(account);
  }
  // Define an internal function '_removePatient' to remove this role, called
by 'removePatient'
  function _removePatient(address account) internal {
    Roles.remove(patients, account);
    emit PatientRemoved(account);
  }
}
```

PharmacistRole.sol

```
pragma solidity ^0.4.24;
// Import the library 'Roles'
import "./Roles.sol";
// Define a contract 'PharmacistRole' to manage this role - add, remove, check
contract PharmacistRole {
  using Roles for Roles.Role;
  // Define 2 events, one for Adding, and other for Removing
  event PharmacistAdded(address indexed account);
  event PharmacistRemoved(address indexed account);
  // Define a struct 'pharmacists' by inheriting from 'Roles' library, struct
Role
  Roles.Role private pharmacists;
```

```
  // In the constructor make the address that deploys this contract the 1st ph
armacist
  constructor() public {
    _addPharmacist(msg.sender);
  }
  // Define a modifier that checks to see if msg.sender has the appropriate ro
le
  modifier onlyPharmacist() {
    require(isPharmacist(msg.sender));
    _;
  }
  // Define a function 'isPharmacist' to check this role
  function isPharmacist(address account) public view returns (bool) {
    return Roles.has(pharmacists, account);
  }
  // Define a function 'addPharmacist' that adds this role
  function addPharmacist(address account) public onlyPharmacist {
    _addPharmacist(account);
  }
  // Define a function 'renouncePharmacist' to renounce this role
  function renouncePharmacist() public {
    _removePharmacist(msg.sender);
  }
  // Define an internal function '_addPharmacist' to add this role, called by
'addPharmacist'
  function _addPharmacist(address account) internal {
    Roles.add(pharmacists, account);
    emit PharmacistAdded(account);
  }
  // Define an internal function '_removePharmacist' to remove this role, call
ed by 'removePharmacist'
  function _removePharmacist(address account) internal {
    Roles.remove(pharmacists, account);
    emit PharmacistRemoved(account);
  }
}
```

Roles.sol

```
pragma solidity ^0.4.24;
/**
 * @title Roles
 * @dev Library for managing addresses assigned to a Role.
 */
```

```
library Roles {
  struct Role {
    mapping (address => bool) bearer;
  }
  /**
   * @dev give an account access to this role
  function add(Role storage role, address account) internal {
    require(account != address(0));
    require(!has(role, account));
    role.bearer[account] = true;
  }
  /**
   * @dev remove an account's access to this role
   */
  function remove(Role storage role, address account) internal {
    require(account != address(0));
    require(has(role, account));
    role.bearer[account] = false;
  }
  /**
   * @dev check if an account has this role
   * @return bool
   */
  function has(Role storage role, address account)
    internal
    view
    returns (bool)
  {
    require(account != address(0));
    return role.bearer[account];
  }
}
```

Ownable.sol

```
pragma solidity ^0.4.24;
/// Provides basic authorization control
contract Ownable {
    address private origOwner;
    // Define an Event
    event TransferOwnership(address indexed oldOwner, address indexed newOwner
    );
    /// Assign the contract to an owner
    constructor () internal {
        origOwner = msg.sender;
        emit TransferOwnership(address(0), origOwner);
```

```
    }
    /// Look up the address of the owner
    function owner() public view returns (address) {
        return origOwner;
    }
    /// Define a function modifier 'onlyOwner'
    modifier onlyOwner() {
        require(isOwner());
        _;
    }
    /// Check if the calling address is the owner of the contract
    function isOwner() public view returns (bool) {
        return msg.sender == origOwner;
    }
    /// Define a function to renounce ownerhip
    function renounceOwnership() public onlyOwner {
        emit TransferOwnership(origOwner, address(0));
        origOwner = address(0);
    }
    /// Define a public function to transfer ownership
    function transferOwnership(address newOwner) public onlyOwner {
        _transferOwnership(newOwner);
    }
    /// Define an internal function to transfer ownership
    function _transferOwnership(address newOwner) internal {
        require(newOwner != address(0));
        emit TransferOwnership(origOwner, newOwner);
        origOwner = newOwner;
    }
}
**************************************************************************
```

Migrations .sol

```
pragma solidity >=0.4.22 <0.9.0;
contract Migrations {
address public owner = msg.sender;
uint public last_completed_migration;
  modifier restricted() {
    require(
      msg.sender == owner,
      "This function is restricted to the contract's owner"
          );
                _;
  }
  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
```

```
  }
}
```

TEST

```
/ This script is designed to test the solidity smart contract -
 SuppyChain.sol -- and the various functions within
// Declare a variable and assign the compiled smart contract artifact
var SupplyChain = artifacts.require('SupplyChain')

contract('SupplyChain', function(accounts) {
    // Declare few constants and assign a few sample accounts generated by gan
ache-cli
    var sku = 1
    var upc = 1
    const ownerID = accounts[0]
    const originManufacturerID = accounts[1]
    const originFactoryName = "John Doe"
    const originFactoryInformation = "Yarray Valley"
    const originFactoryLatitude = "-38.239770"
    const originFactoryLongitude = "144.341490"
    var medicineID = sku + upc
    const medicineNotes = "Best beans for Espresso"
    const medicinePrice = web3.utils.toWei("1", "ether")
    var medicineState = 0
    const distributorID = accounts[2]
    const pharmacistID = accounts[3]
    const patientID = accounts[4]
    const emptyAddress = '0x00000000000000000000000000000000000000000'

    ///Available Accounts
    ///==================
    ///(0) 0x27d8d15cbc94527cadf5ec14b69519ae23288b95
    ///(1) 0x018c2dabef4904ecbd7118350a0c54dbeae3549a
    ///(2) 0xce5144391b4ab80668965f2cc4f2cc102380ef0a
    ///(3) 0x460c31107dd048e34971e57da2f99f659add4f02
    ///(4) 0xd37b7b8c62be2fdde8daa9816483aebdbd356088
    ///(5) 0x27f184bdc0e7a931b507ddd689d76dba10514bcb
    ///(6) 0xfe0df793060c49edca5ac9c104dd8e3375349978
    ///(7) 0xbd58a85c96cc6727859d853086fe8560bc137632
    ///(8) 0xe07b5ee5f738b2f87f88b99aac9c64ff1e0c7917
    ///(9) 0xbd3ff2e3aded055244d66544c9c059fa0851da44

    console.log("ganache-cli accounts used here...")
    console.log("Contract Owner: accounts[0] ", accounts[0])
    console.log("Manufacturer: accounts[1] ", accounts[1])
```

```
    console.log("Distributor: accounts[2] ", accounts[2])
    console.log("Pharmacist: accounts[3] ", accounts[3])
    console.log("Patient: accounts[4] ", accounts[4])

    // 1st Test
    it("Testing smart contract function makeMedicine() that allows a manufactu
rer to make medicine", async() => {
        const supplyChain = await SupplyChain.deployed()

        // Declare and Initialize a variable for event
        var eventEmitted = false

        // Watch the emitted event Made()
        supplyChain.Made((err, res) => {
            eventEmitted = true;
        })


        // Mark an medicine as Made by calling function makeMedicine()
        await supplyChain.makeMedicine(upc, originManufacturerID, originFactor
yName, originFactoryInformation, originFactoryLatitude, originFactoryLongitude
, medicineNotes)

        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferOne = await supplyChain.fetchMedicineBufferOne.call(
upc)
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)
        const resultBufferThree = await supplyChain.fetchMedicineBufferThree.c
all(upc)
        // Verify the result set
        // console.log(resultBufferOne);
        assert.equal(resultBufferOne[0].toNumber(), sku, 'Error: Invalid medic
ine SKU')
        assert.equal(resultBufferOne[1], upc, 'Error: Invalid medicine UPC')
        assert.equal(resultBufferOne[2], originManufacturerID, 'Error: Missing
 or Invalid ownerID')
        assert.equal(resultBufferOne[3], originManufacturerID, 'Error: Missing
 or Invalid originManufacturerID')
        assert.equal(resultBufferOne[4], originFactoryName, 'Error: Missing or
 Invalid originFactoryName')
        assert.equal(resultBufferOne[5], originFactoryInformation, 'Error: Mis
sing or Invalid originFactoryInformation')
```

```
        assert.equal(resultBufferOne[6], originFactoryLatitude, 'Error: Missin
g or Invalid originFactoryLatitude')
        assert.equal(resultBufferTwo[0], originFactoryLongitude, 'Error: Missi
ng or Invalid originFactoryLongitude')
        assert.equal(resultBufferTwo[5], 0, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')
    })

    // 2nd Test
    it("Testing smart contract function packMedicine() that allows a manufactu
rer to pack medicine", async() => {
        const supplyChain = await SupplyChain.deployed()

        // Declare and Initialize a variable for event
        var eventEmitted = false

        // Watch the emitted event Packed()
        supplyChain.Packed((err, res) => {
            eventEmitted = true;
        })


        // Mark an medicine as Packed by calling function PackMedicine()
        await supplyChain.packMedicine(upc)

        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)

        assert.equal(resultBufferTwo[3], 1, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')

    })
    // 4th Test
    it("Testing smart contract function sellMedicine() that allows a manufactu
rer to sell medicine", async() => {
        const supplyChain = await SupplyChain.deployed()
                // Declare and Initialize a variable for event
        var eventEmitted = false
                // Watch the emitted event ForSale()
        supplyChain.ForSale((err, res) => {
            eventEmitted = true;
        })
```

```
        // Mark an medicine as ForSale by calling function sellMedicine()
        await supplyChain.sellMedicine(upc,web3.utils.toWei("1", "ether"))

        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)

        assert.equal(resultBufferTwo[3], 2, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')
       })

    // 5th Test
    it("Testing smart contract function buyMedicine() that allows a distributo
r to buy medicine", async() => {
        const supplyChain = await SupplyChain.deployed()

        // Declare and Initialize a variable for event
        var eventEmitted = false

        // Watch the emitted event Sold()
        supplyChain.Sold((err, res) => {
            eventEmitted = true;
        })
        // Mark an medicine as Sold by calling function buyMedicine()
        await supplyChain.buyMedicine(upc,{value: web3.utils.toWei('1', 'ether
') })
        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)

        assert.equal(resultBufferTwo[3], 3, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')

    })
    // 6th Test
    it("Testing smart contract function shipMedicine() that allows a distribut
or to ship medicine", async() => {
        const supplyChain = await SupplyChain.deployed()
        // Declare and Initialize a variable for event
        var eventEmitted = false
        assert.equal(1, 1, )
        // Watch the emitted event Packed()
        supplyChain.Shipped((err, res) => {
```

```
            eventEmitted = true;
        })
        await supplyChain.addManufacturer(originManufacturerID)
        // Mark an medicine as Shipped by calling function shipMedicine()
        await supplyChain.shipMedicine(upc,{from: originManufacturerID})
        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)

        assert.equal(resultBufferTwo[3], 4, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')

    })
      // 7th Test
    it("Testing smart contract function receiveMedicine() that allows a pharma
cist to mark medicine received", async() => {
        const supplyChain = await SupplyChain.deployed()

        // Declare and Initialize a variable for event

        var eventEmitted = false

        // Watch the emitted event Received()
        supplyChain.Received((err, res) => {
            eventEmitted = true;
        })


        // Mark an medicine as Reiceived by calling function receiveMedicine()
        await supplyChain.receiveMedicine(upc)
        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)

        assert.equal(resultBufferTwo[3], 5, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')

    })

    // 8th Test
    it("Testing smart contract function purchaseMedicine() that allows a patie
nt to purchase medicine", async() => {
        const supplyChain = await SupplyChain.deployed()
```

```
        // Declare and Initialize a variable for event
        var eventEmitted = false
        // Watch the emitted event Purchased()
        supplyChain.Purchased((err, res) => {
            eventEmitted = true;
        })
        // Mark an medicine as Purchased by calling function purchaseMedicine(
)
        await supplyChain.purchaseMedicine(upc)

        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        const resultBufferTwo = await supplyChain.fetchMedicineBufferTwo.call(
upc)

        assert.equal(resultBufferTwo[3], 6, 'Error: Invalid medicine State')
        assert.equal(eventEmitted, true, 'Invalid event emitted')

    })
// Deleted as I have already tested them in the first case
    // 9th Test
    it("Testing smart contract function fetchMedicineBufferOne() that allows a
nyone to fetch medicine details from blockchain", async() => {
        const supplyChain = await SupplyChain.deployed()

        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()
        // Verify the result set:

    })
    // 10th Test
    it("Testing smart contract function fetchMedicineBufferTwo() that allows a
nyone to fetch medicine details from blockchain", async() => {
        const supplyChain = await SupplyChain.deployed()
        // Retrieve the just now saved medicine from blockchain by calling fun
ction fetchMedicine()

        // Verify the result set:

    })
});
```

Migration

```
// migrating the appropriate contracts
var ManufacturerRole = artifacts.require("./ManufacturerRole.sol");
var DistributorRole = artifacts.require("./DistributorRole.sol");
var PharmacistRole = artifacts.require("./PharmacistRole.sol");
var PatientRole = artifacts.require("./PatientRole.sol");
var SupplyChain = artifacts.require("./SupplyChain.sol");

module.exports = function(deployer) {
  deployer.deploy(ManufacturerRole);
  deployer.deploy(DistributorRole);
  deployer.deploy(PharmacistRole);
  deployer.deploy(PatientRole);
  deployer.deploy(SupplyChain);
};
```

Truffel.js

```
const HDWalletProvider = require("@truffle/hdwallet-provider");
const mnemonic = "mango urge flame spread govern can shield spawn keep poet so
cial foam";
module.exports = {
  networks: {
    ropsten: {
      provider: function() {
        return new HDWalletProvider(mnemonic, "https://ropsten.infura.io/v3/15
1be4e913e4474eb9f8dce8b37f13a9")
      },
      network_id: 3
    },
  },
  compilers: {
    solc: {
       version: "0.4.24",    // Fetch exact version from solc-
bin (default: truffle's version)

       }

  }
};
```