



---

# TPE : Programmation d'une Blockchain à l'aide de la plateforme Hyperledger

---

M1 IWOCS

ANNÉES 2020/2021

ROBIN GUYOMAR  
MAXENCE BOURGEAUX  
SOUHAIL KACIMI

ENCADRANTS : CLAUDE DUVALLET ET CYRILLE BERTELLE

# Table des matières

<b>1</b>	<b>Définition et propriétés de la blockchain</b>	<b>3</b>
1.1	Les caractéristiques et les propriétés de la blockchain . . . . .	3
1.2	L’aspect chaînage de la blockchain . . . . .	4
1.3	L’aspect réseau d’une blockchain . . . . .	5
1.4	Les particularités des différentes blockchains . . . . .	6
1.4.1	Les mécanismes de gestion de consensus . . . . .	6
1.4.2	Les types de blockchains . . . . .	7
1.5	L’enregistrement d’une transaction . . . . .	8
<b>2</b>	<b>Hyperledger Fabric et les différents outils utilisés</b>	<b>9</b>
2.1	La plateforme Hyperledger Fabric . . . . .	9
2.2	Les contrats intelligents . . . . .	10
2.3	Les différents outils utilisés . . . . .	11
<b>3</b>	<b>Vente de voitures sécurisée via la blockchain Hyperledger Fabric</b>	<b>14</b>
3.1	Guide d’installation de la blockchain Hyperledger Fabric . . . . .	14
3.2	Réalisation de l’application de vente de voitures . . . . .	16
3.2.1	Mise en place du réseau . . . . .	16
3.2.2	Développement du smart contract . . . . .	16
3.2.3	Déploiement du smart contract . . . . .	18
3.2.4	Interface Web . . . . .	19
<b>4</b>	<b>Conclusion et perspectives</b>	<b>29</b>

# Introduction

Le mot *blockchain* peut se traduire littéralement en français par chaîne de blocs. C'est une technologie de stockage et de transmission de données, dont la caractéristique principale est d'être infalsifiable.

La première blockchain est apparue en 2008 avec la création du Bitcoin par un (ou plusieurs) inconnu(s) sous le pseudonyme « Satoshi Nakamoto » (son identité demeure toujours inconnue mais il a affirmé être un japonais né le 5 avril 1975). Il est le premier à avoir miné du Bitcoin avec le block 0 (*genesis block of bitcoin*) en 2009, qui initialise la blockchain et qui contenait 50 bitcoins.

Il a résolu le problème de la double dépense pour les monnaies numériques, qui consiste à utiliser plusieurs fois le même token, c'est à dire le dupliquer ou le falsifier (ce qui engendre une inflation car on utilise de l'argent qui n'existe pas).

Depuis l'envolée du Bitcoin en novembre 2017 : 15 000 € (jusqu'à presque 50 000 € récemment début mars 2021), les blockchains suscitent un plus grand intérêt.

La deuxième blockchain mondialement connue (en termes de capitalisation) est Ethereum. Inventée par Vitalik Buterin (canadien d'origine russe né le 31 janvier 1994) en 2013 qui elle permet, contrairement au Bitcoin qui se limite à des applications monétaires, de créer n'importe quel type d'application. Elle peut être considérée comme un « ordinateur global » qui permet de créer un nouveau Web décentralisé.

Contrairement au Bitcoin, Ethereum permet donc à des organismes comme les banques, les assurances, les marchands d'œuvres d'art, etc., d'utiliser une blockchain pour leurs différentes activités.

Il existe bien évidemment d'autres blockchains, mais le but de ce TPE est de découvrir le principe général de la blockchain, puis d'apprendre à utiliser une blockchain précise qui est celle d'Hyperledger Fabric, qui nous est imposée par le sujet.

Les blockchains utilisent notamment ce qu'on appelle des contrats intelligents via la programmation chaincode, nous allons donc également explorer cet aspect important.

Nous avons donc deux objectifs principaux :

- Réaliser une documentation sur la forge de l'université Le Havre Normandie qui explique comment installer et utiliser une blockchain Hyperledger Fabric (disponible ici <https://www-apps.univ-lehavre.fr/forge/duvallec/tpe-blockchain-hyperledger>)
- Réaliser un cas d'étude en proposant une application de ventes de voitures qui utilise la blockchain Hyperledger Fabric

# 1 Définition et propriétés de la blockchain

Dans cette partie, nous allons commencer par nous intéresser aux caractéristiques et aspects fondamentaux de la blockchain. Puis, nous allons présenter les différents types de blockchains qui existent ainsi que les différents mécanismes mis en place pour ajouter des blocs à la chaîne.

## 1.1 Les caractéristiques et les propriétés de la blockchain

Une blockchain peut être caractérisée comme un registre distribué, immuable et infalsifiable. Lorsqu'une donnée ou une transaction est écrite dans cette chaîne de blocs par l'un des membres et qu'elle est validée par tous, il devient impossible de la modifier.

Une blockchain est une base de données généralement sans organe de contrôle, dont la particularité est d'être quasiment impossible à modifier grâce à des moyens cryptographiques. De plus, les données sont indépendantes les unes des autres.

L'utilisation de la blockchain comporte d'autres avantages comme la rapidité des transactions puisque la validation d'un bloc ne prend que quelques minutes voire même quelques secondes. Cela apporte également des gains de productivité et d'efficacité puisqu'elle confie l'organisation des transactions à un protocole informatique, ce qui réduit logiquement les coûts de transaction ou de centralisation existant dans les systèmes traditionnels.

Une blockchain possède trois propriétés majeures :

- **La transparence**

Les blockchains ont tendance à être transparentes. En effet, tout le monde peut consulter l'ensemble des données et des transactions dans la blockchain depuis sa date de création.

- **La sécurité**

Grâce à un mécanisme de hachage, il est quasiment impossible de corrompre les données. En effet, il y a une réplication des données sur un ensemble de nœuds (ordinaires et validateurs). La corruption de données devient donc quasiment impossible car il faudrait corrompre au moins 51% des nœuds.

- **L'absence d'organe de contrôle**

La blockchain est décentralisée signifiant qu'il n'y a pas d'autorité derrière elle contrairement à un système classique. Elle fonctionne uniquement grâce à une architecture pair-à-pair.

## 1.2 L'aspect chaînage de la blockchain

Le premier aspect fondamental de la blockchain est le chaînage des blocs. En effet, les données sont sauvegardées dans des blocs, ce qui permet de retracer n'importe quelle transaction.

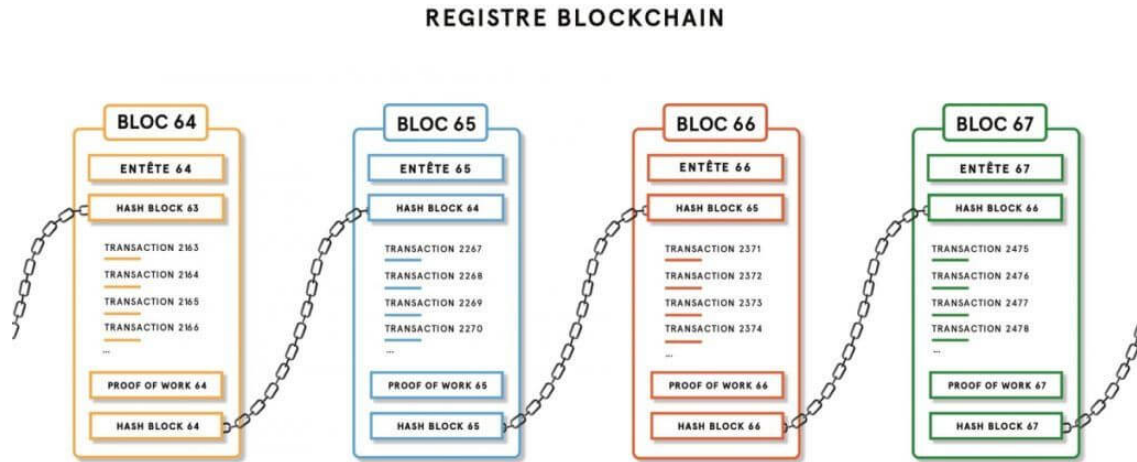


FIGURE 1 – L'aspect chaînage de la blockchain

Source : <https://coin24.fr/dictionnaire/blockchain/>

Comme on peut le voir sur l'image ci-dessus, chaque bloc contient un hash, le hash du bloc précédent, la date de création du bloc et la ou les transactions.

Un hash est une suite de nombres qui représente une information. Pour le créer, on utilise une fonction de *hachage* qui permet de crypter une donnée en la transformant en empreinte numérique. Cette empreinte est impossible à décrypter, et si elle est modifiée, le hash changera complètement et n'aura rien à voir avec l'original.

On peut prendre comme exemple la fonction de hachage SHA-256 qui est la plus connue. Développée par la NSA, elle produit un hash de 256 bits à partir d'un message d'une taille maximum égale à 264 bits (les blocs ont une taille maximum de 512 bits et les mots 32 bits).

Voici une image montrant l'utilisation de la fonction de hachage SHA-256 :

Message	Empreinte
bœuf	06d2c4c0420e18394520d8ccde3a2a937da4040e3f69a8f440077a935ab7b968
Bœuf	7abd6d70f64e3b264f657c6ba92ad752a422a7dd27ca61fa957c0b79ca36dd0b
boeuf	9c0ed395dc2af34608210254d1f7828757bd750a86d02b63db177ec404d3dd01

FIGURE 2 – Un exemple de fonction de hachage

Source : <https://cryptoast.fr/hash-hachage-bitcoin-blockchain/>

### 1.3 L'aspect réseau d'une blockchain

Les blockchains publiques utilisent généralement un réseau pair à pair, c'est à dire qu'elles ne possèdent pas d'organe de contrôle direct, contrairement aux blockchains privées qui sont centralisées.

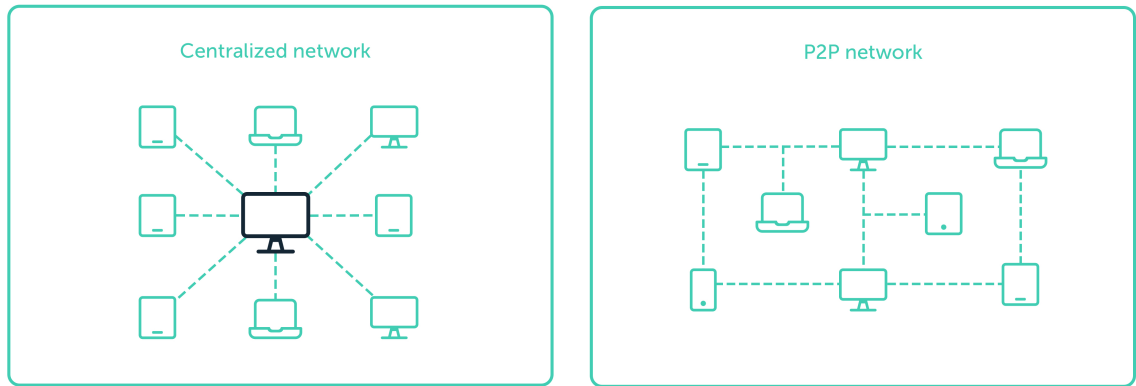


FIGURE 3 – L'aspect réseau de la blockchain

Source : <https://www.ledger.com/academy/blockchain/what-is-blockchain>

Comme nous pouvons l'observer via cette image, l'objectif d'un tel réseau est de décentraliser l'information et de la distribuer entre chaque nœud. L'information est donc disponible pour chaque utilisateur, et tous les utilisateurs sont reliés les uns aux autres, ce qui permet de retracer n'importe quelle information très facilement.

Ainsi, si l'on souhaite modifier un nœud dans le but de le falsifier, il faudrait le faire pour chaque nœud de chaque utilisateur, ce qui est impossible avec les moyens technologiques actuels puisqu'il faut résoudre un problème cryptographique de grande complexité qui demande un grand coût de calcul (cf Preuve de travail, partie 1.4.1).

## 1.4 Les particularités des différentes blockchains

### 1.4.1 Les mécanismes de gestion de consensus

Pour ajouter un nouveau bloc à la chaîne, il faut se faire valider par les autres blocs via une gestion de consensus. Cela permet de garantir que chaque bloc de la chaîne possède la même information, on appelle cela le consensus.

Il existe, comme pour les types de blockchains, plusieurs mécanismes de preuve :

- **La preuve de travail** (blockchain publique)  
C'est la méthode utilisée par le Bitcoin et Ethereum pour ne citer qu'eux. Pour qu'un bloc soit validé, il faut qu'il remplisse cette preuve de travail, qui n'est autre qu'un problème cryptographique à résoudre. Cependant, ce problème cryptographique est d'une grande difficulté et nécessite donc une grande puissance de calcul et donc un certain coût de calcul. Ce coût de calcul est asymétrique, en effet, il est difficile pour le mineur mais facilement vérifiable par l'ensemble du réseau. Le premier mineur qui aura résolu ce problème devra ensuite être validé par l'ensemble du réseau pour finalement être ajouté à la chaîne. Cette difficulté augmente au fil du temps.
- **La preuve d'autorité** (blockchain privée)  
Cette preuve se base sur la *réputation*, c'est à dire la confiance des nœuds du réseau. Seul un petit nombre de nœuds ayant la confiance de la totalité du réseau aura le droit de miner. On retrouve le principe d'administrateur des blockchains privées qui choisit qui aura le droit de miner.
- **La preuve d'enjeu** (blockchain de consortium)  
Cette méthode demande à un utilisateur (mineur) de prouver la possession d'une certaine quantité de crypto-monnaie, c'est à dire leur participation dans la chaîne, pour prétendre à pouvoir valider les prochains blocs. La sélection de l'utilisateur est aléatoire et est pondérée par la quantité de crypto-monnaies que possède l'utilisateur. Cette pondération permet d'éviter que le plus riche utilisateur ait toujours l'avantage.

### 1.4.2 Les types de blockchains

Il existe trois différents types de blockchains :

- **La blockchain publique**

C'est la blockchain la plus répandue dans le monde et la plus connue. Elle utilise un réseau pair à pair, c'est à dire sans organe de contrôle, où tout le monde peut effectuer des transactions et les vérifier (libre accès). Les transactions ne sont pas anonymes, mais utilisent un pseudonyme et une adresse publique. Par exemple, c'est cette blockchain que le Bitcoin utilise.

- **La blockchain privée**

Complètement centralisée, elle est dirigée par un organe central (une sorte d'administrateur) appelé gérant. C'est lui qui va ajouter les blocs à la chaîne et il peut la modifier à sa guise. Il n'y a pas de lien entre les différents acteurs. Il faut l'autorisation du gérant pour participer à la blockchain et les autres participants peuvent refuser cet accès suivant les mécanismes de contrôle mis en place. Utilisée principalement par des entreprises voulant garder leurs transactions privées et avoir une confidentialité élevée, comme par exemple les banques.

- **La blockchain de consortium**

Elle regroupe des acteurs qui veulent travailler ensemble. Seul certains acteurs (les plus importants) pourront prendre les décisions. C'est un système décentralisé avec des droits d'écritures modifiables. Ce sont ces décisionnaires qui choisissent quelles informations seront rendues publiques (quels blocs sont privés ou publics).



## 1.5 L'enregistrement d'une transaction

Maintenant que nous avons vu les différentes caractéristiques d'une blockchain, nous pouvons nous intéresser à l'enregistrement d'une transaction, autrement dit, l'ajout d'un nouveau bloc dans la chaîne.

Pour ajouter un nouveau bloc à la chaîne, il faut se faire valider par les autres blocs via une gestion de consensus. Nous allons prendre le cas d'un ajout de bloc dans une blockchain publique. C'est donc le mécanisme de preuve qui nous intéresse ici. Pour rappel, le mécanisme de preuve consiste à résoudre un problème cryptographique nécessitant un grand coût de calcul qui continue d'augmenter au fil du temps (plus de détails dans la partie 1.4.1).

Prenons un exemple de transaction impliquant deux personnes : une première personne nommée Alice qui souhaite vendre une voiture, et une deuxième personne nommée Bob souhaitant acheter la voiture d'Alice. On peut décomposer l'ajout de la transaction dans la blockchain en plusieurs étapes :

1. Tout d'abord, Les deux personnes prennent contact et se mettent d'accord sur les modalités de la transaction (date, prix, objet, etc.).
2. Les données de la transaction sont ensuite regroupées dans un seul et même bloc.
3. Bob va résoudre le problème cryptographique de la blockchain afin de pouvoir être ajouté. Comme vu précédemment, le coût de calcul ici est très important.
4. Les nœuds de la blockchain (les autres utilisateurs) vont vérifier le résultat obtenu par Bob, et s'il est correct, vont approuver l'ajout du nouveau bloc. Le coût de calcul étant asymétrique, les utilisateurs vont vérifier facilement et rapidement le résultat de Bob.
5. Le bloc est finalement ajouté à la chaîne, et l'ensemble des utilisateurs de la blockchain y ont accès.
6. Enfin, Alice reçoit le paiement, tandis que Bob reçoit la voiture.

## 2 Hyperledger Fabric et les différents outils utilisés

Nous allons maintenant parler d'Hyperledger Fabric et de sa blockchain, qui va être l'outil principal de notre TPE. Puis, nous allons parler des différents logiciels qui ont été nécessaires pour réaliser ce projet et le faire fonctionner.

### 2.1 La plateforme Hyperledger Fabric



Hyperledger Fabric est un framework (que l'on peut traduire en français par environnement de développement ou encore infrastructure logicielle) open source lancée en 2015 par la Fondation Linux qui propose la mise en place d'une blockchain privée.

Hyperledger Fabric se désigne comme une fondation permettant le développement d'applications avec une architecture modulable et adaptable aux différentes situations. Il s'agit d'un framework modulaire et généraliste qui offre des fonctionnalités uniques de gestions d'identités et de contrôles d'accès, ce qui le rend adapté à une variété d'applications industrielles. Il a d'ailleurs été conçu pour satisfaire et répondre aux besoins des entreprises.

Puisqu'Hyperledger Fabric propose la mise en place d'une blockchain privée, le nombre d'utilisateurs sera donc limité. Par conséquent, il offre une approche unique du consensus qui permet d'obtenir des performances à grande échelle tout en préservant la confidentialité des utilisateurs.

La confidentialité des utilisateurs est garantie et maintenue grâce à la notion de *canal* qu'Hyperledger Fabric propose. Ce système de canaux permet l'isolation totale des transactions. En effet, cela offre la possibilité de partager les hachages sur le registre comme preuve de transaction, tout en gardant les données de la transaction privées.

## 2.2 Les contrats intelligents



Un contrat intelligent ou *smart contract* est par définition un programme automatisé. Ces programmes ou contrats sont des protocoles informatiques visant à automatiser une action lorsque les conditions prédéfinies sont remplies et garantissent l'exécution d'un contrat entre deux ou plusieurs parties.

En pratique, un contrat intelligent exécute automatiquement des conditions prédéfinies dans une blockchain. Seul le code informatique décide de l'exécution totale ou partielle d'un contrat. C'est cette spécificité qui rend ces protocoles *intelligents*. Ce ne sont donc pas de simples contrats à proprement parler.

L'ensemble de ces fonctionnalités sont assurées par le caractère informatisé des contrats élégants. Ainsi, lorsque les conditions prédéfinies seront remplies, le contrat prendra forme et chaque contractant sera débiteur de son obligation.

Ce type de contrat va conduire à une transaction. En effet, un contrat élégant est rattaché à une personne ou une entité qui déclenche une transaction et peut subir les frais de la mise en place de cette transaction dans la blockchain.

Les contrats élégants permettent ainsi de traiter des transactions sans qu'un tiers de confiance ne soit nécessaire pour vérifier ou faire appliquer l'accord. Les accords sont traçables et irréversibles.

L'objectif des contrats intelligents est d'offrir une sécurité numérique supérieure par rapport aux contrats traditionnels, permettre un gain de temps considérable ou encore d'économiser des frais puisque théoriquement il n'y a plus besoin d'un juriste.

Pour notre projet de vente de voitures, le contrat élégant vérifie par exemple que la voiture est bien à vendre et que le prix d'achat correspond bien au prix de vente. Si ces conditions sont remplies, alors la transaction bancaire s'effectue et le véhicule change de propriétaire.

Dans Hyperledger Fabric, les contrats élégants se traduisent par des chaincodes que nous allons voir par la suite (dans la partie 3. Vente de voitures sécurisée par la blockchain Hyperledger).

## 2.3 Les différents outils utilisés

Nous allons maintenant lister et présenter les différents outils nécessaires à ce projet.

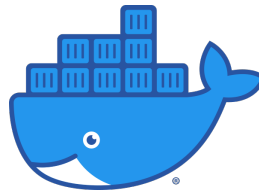
### — Hyperledger Explorer



Comme son nom l'indique, c'est une application spécialement conçue pour la blockchain Hyperledger Fabric pour visualiser les opérations effectuées sur une blockchain, en permettant notamment d'afficher les blocs, les transactions et ses données associées, les chaincodes ainsi que toutes les autres informations stockées dans le registre.

Il s'agit du tout premier explorateur de blockchains pour les registres autorisés, permettant à quiconque d'explorer de l'intérieur les projets de registres distribués créés par les membres d'Hyperledger, sans compromettre leur vie privée.

### — Docker



Docker est un logiciel permettant de lancer des applications dans des conteneurs logiciels.

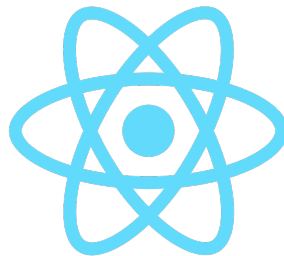
C'est un outil qui peut emballer une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur. Il ne s'agit pas de virtualisation, mais de *conteneurisation*, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement.

— **Node.js**



Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau. Elle permet concrètement l'exécution de JavaScript côté serveur. De plus, elle permet de créer des applications réseau évolutives grâce à son fonctionnement non bloquant, telles qu'un serveur web ou une API par exemple.

— **React.js**



React.js est une bibliothèque JavaScript libre dont le but principal est de faciliter la création d'application web monopage (page web unique) via la création de composants dépendant d'un état et générant une page HTML à chaque changement d'état. Cette bibliothèque est utilisée sur des sites très connus comme Facebook, Instagram, Netflix ou encore WhatsApp.

## — Express JS



Express JS est un framework qui permet de construire des applications ou des API web basées sur Node.js. Il est le framework standard pour le développement de serveur en Node.js.

## — WebSocket



WebSocket est un protocole réseau basé sur le protocole TCP permettant une communication bidirectionnelle entre un navigateur (coté client) et un serveur.

## — Bulma



Bulma est un framework CSS open source qui fournit des composants front-end directement prêts à l'emploi que l'on peut facilement combiner entre eux pour créer des interfaces web réactives et belles visuellement.

### 3 Vente de voitures sécurisée via la blockchain Hyperledger Fabric

Nous allons maintenant présenter notre travail réalisé. Nous allons voir la réalisation d'une documentation dans un premier temps puis la réalisation d'une application de ventes de voitures dans un second temps.

#### 3.1 Guide d'installation de la blockchain Hyperledger Fabric

Comme évoqué dans l'introduction, l'un des objectifs de ce TPE était de réaliser une documentation correspondant à un guide d'installation et d'utilisation. Ce guide explique notamment comment installer les différents logiciels nécessaires pour faire fonctionner Hyperledger Fabric.

Il explique aussi en détail comment faire fonctionner le TPE 2020 d'Amine Bousoulam. Cette documentation permet également de comprendre comment fonctionne Hyperledger Explorer.

Enfin, ce guide montre les fonctionnalités de notre propre application Web et comment elle fonctionne.

Voici quelques captures d'écrans de notre documentation :

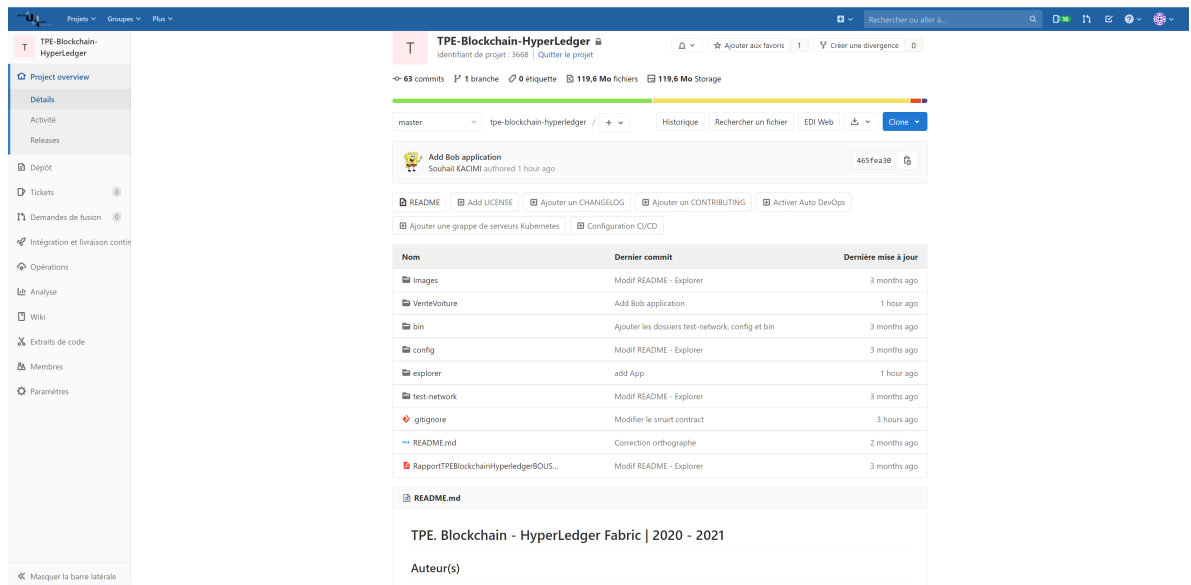


FIGURE 4 – Dépôt Git regroupant tous les fichiers nécessaires au projet

**T** TPE-Blockchain-Hyperledger

- Project overview
- Details
- Activité
- Releases

---

- Dépôt
- Tickets
- Demandes de fusion
- Intégration et livraison continue
- Opérations
- Analyse
- Wiki
- Extraits de code
- Membres
- Paramètres

## TPE. Blockchain - Hyperledger Fabric | 2020 - 2021

### Auteur(s)

Nom	Prénom	Email
Bourgeois	Marcene	marsence.bourgeois@etu.univ-lehavre.fr
Guyomar	Robin	robin.guyomar@etu.univ-lehavre.fr
Kacimi	Sauhaïl	sauhaïl.kacimi@etu.univ-lehavre.fr

Ce TPE reprend le travail effectué par Amine Boussoualim l'année précédente (2019 - 2020).

Son rapport est disponible sous format pdf et son travail est disponible à l'adresse suivante : <https://www.apps-univ-lehavre.fr/forge/ba160129/tpeblockchain>

Lien vers le rapport Overleaf : <https://www.overleaf.com/read/cgghnfrkvmmq>

### I. Installer les différents logiciels

#### Prérequis

La version actuelle du projet utilise Hyperledger fabric version 2.3.1. Il est nécessaire d'avoir une version de Node <= 14 et de Docker compose <= 1.25 pour utiliser cette version d'Hyperledger Fabric.

Nous tests ont été effectués avec ces versions :

```

marcene@marsence-VirtualBox:~$ node -v
v14.15.4
marcene@marsence-VirtualBox:~$ npm -v
6.14.10
marcene@marsence-VirtualBox:~$ docker -v
Docker version 20.10.3, build 48d8b95
marcene@marsence-VirtualBox:~$ docker-compose -v
docker-compose version 1.25.0, build 0a186604
    
```

La suite du tutoriel proposera d'installer soit la version la plus récente, soit une version précise du logiciel.

#### Se mettre à jour

```

sudo apt update
sudo apt upgrade
    
```

#### Installer git

```

sudo apt-get install git
    
```

## Auteur(s)

Nom	Prénom	Email
Bourgeois	Maxence	maxence.bourgeois@etu.univ-lehavre.fr
Guyomar	Robin	robin.guyomar@etu.univ-lehavre.fr
Kacimi	Souhaïl	souhaïl.kacimi@etu.univ-lehavre.fr

Ce TPE reprend le travail effectué par Amine Boussoulalim l'année précédente (2019 - 2020)

Son rapport est disponible sous format pdf et son travail est disponible à l'adresse suivante : <https://www-apps.univ-lehavre.fr/forge/ba160129/tpeblockchain>

Lien vers le rapport Overleaf : <https://www.overleaf.com/read/cgghnfrkvrmo>

## I. Installer les différents logiciels

## Prérequis

La version actuelle du projet utilise HyperLedger fabric version 2.3.1. Il est nécessaire d'avoir une version de Node <= 14 et de Docker compose <= 1.25 pour utiliser cette version d'HyperLedger Fabric.

Nos tests ont été effectués avec ces versions

```
maxence@maxence-VirtualBox:~$ node -v
v14.15.4
maxence@maxence-VirtualBox:~$ npm -v
6.14.10
maxence@maxence-VirtualBox:~$ docker -v
Docker version 20.10.3, build 48d30b5
maxence@maxence-VirtualBox:~$ docker-compose -v
docker-compose version 1.25.0, build 0a186664
```

La suite du tutoriel proposera d'installer soit la version la plus récente, soit une version précise du logiciel.

## Se mettre à jour

```
sudo apt update
sudo apt upgrade
```

Installer git

```
sudo apt-get install git
```

FIGURE 5 – Début du guide d'installation

Toute la documentation est disponible sur la forge de l'université Le Havre Normandie à l'adresse suivante : <https://www-apps.univ-lehavre.fr/forge/duvallec/tp-e-blockchain-hyperledger>



## 3.2 Réalisation de l'application de vente de voitures

### 3.2.1 Mise en place du réseau

Pour pouvoir utiliser Hyperledger Fabric, la première étape est de déployer un réseau de tests qui nous permettra d'utiliser Hyperledger Fabric sur nos propres machines. Ce réseau de tests nous offre la possibilité de tester les contrats intelligents ainsi que les applications.

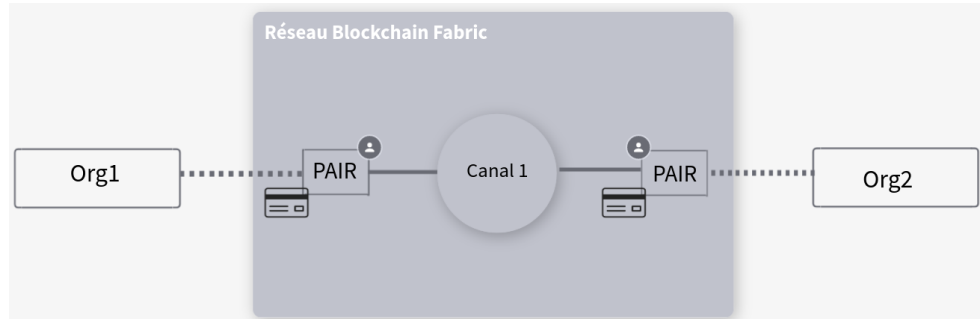


FIGURE 6 – Schéma illustrant les différents composants du réseau Hyperledger Fabric

Chaque nœud et utilisateur qui interagit avec un réseau Fabric doit appartenir à une organisation afin de participer au réseau. Notre réseau comprend deux organisations : Org1 et Org2.

Les pairs sont les composants fondamentaux de tout le réseau Fabric. Ils stockent le registre distribué de la blockchain et valident les transactions avant qu'elles ne soient validées dans le registre. Les pairs exécutent les contrats intelligents qui contiennent la logique métier utilisée pour gérer les assets dans le registre de la blockchain. Enfin, chaque pair possède une identité propre qui lui permet d'être reconnu dans le réseau Fabric.

Une autre notion très importante est celle du canal. On peut remarquer sur notre schéma qu'il y a un canal qui lie les deux pairs des deux organisations. Ce canal permet une communication privée entre les pairs qui en font partie.

Par exemple s'il y a un autre pair qui appartient à une autre organisation et qu'il ne participe pas au canal, il ne peut ni exécuter des transactions sur ce canal ni voir ce qu'il se passe dedans.

### 3.2.2 Développement du smart contract

Le smart contract permet une communication directe avec la blockchain. Les utilisateurs qui veulent exécuter des transactions invoquent ce contrat intelligent, et en fonction de l'opération qu'ils choisissent, le smart contract va soit extraire soit ajouter des données dans la blockchain.

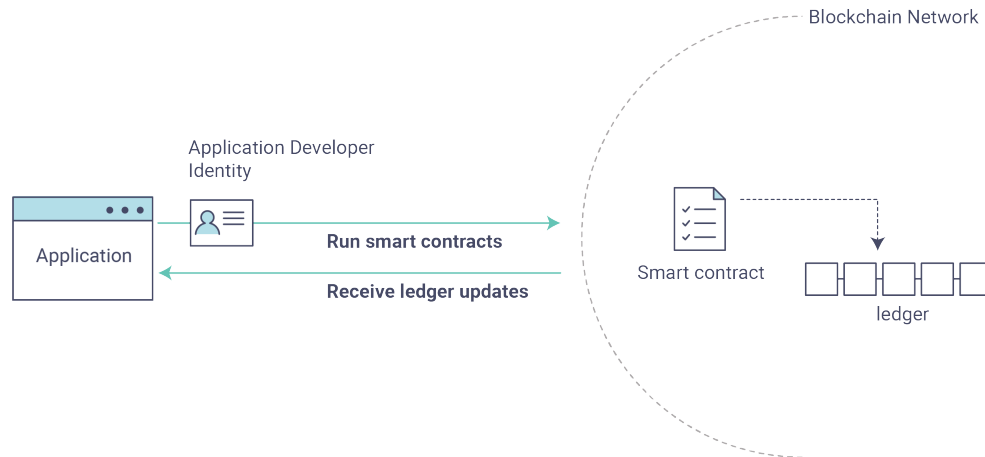


FIGURE 7 – Exécution du smart contract

Notre application est une application de vente de voitures, dont les principales fonctionnalités sont l'achat et la vente d'une voiture. Notre smart contract doit donc comporter les méthodes qui permettent de mettre une voiture à la vente, d'acheter une voiture ainsi que toutes les autres méthodes qui peuvent être utiles pour l'utilisateur.

### Les principales méthodes de notre smart contract

#### 1. **sale**

Les utilisateurs vont invoquer le smart contract avec la méthode *sale* s'ils ont besoin d'ajouter une nouvelle voiture. Une fois invoqué, le smart contract va communiquer directement avec la blockchain, et un nouveau bloc qui contient les informations de la nouvelle voiture va s'ajouter à la blockchain.

#### 2. **buy**

Si un utilisateur veut acheter une nouvelle voiture, il va invoquer le smart contract avec la méthode *buy* pour pouvoir l'acheter. Une fois invoqué, le smart contract va s'occuper de l'achat de la voiture, et un nouveau bloc va s'ajouter à la blockchain pour enregistrer la transaction effectuée.

#### 3. **update**

La modification des propriétés de la voiture se fait avec la méthode *update*. L'utilisateur va invoquer le smart contract avec cette méthode, et le smart contract va s'occuper lui-même de la modification des propriétés de la voiture. Une fois la modification effectuée, un nouveau bloc contenant les informations modifiées de la voiture va s'ajouter dans la blockchain.

#### 4. **queryOwner**

Permet d'invoquer le smart contract pour extraire les voitures appartenant à un propriétaire grâce à la blockchain.

#### 5. **carInfo**

Récupère les informations relatives à une voiture en invoquant le smart contract qui va aller chercher les informations dans la blockchain.

### 3.2.3 Déploiement du smart contract

Dans Hyperledger Fabric, les contrats intelligents sont déployés dans des packages appelés chaincode. Les organisations qui souhaitent valider des transactions ou interroger la blockchain doivent installer un chaincode sur leurs pairs.

Une fois qu'un chaincode a été installé sur les pairs, ces organisations peuvent déployer le chaincode sur le canal et utiliser les smart contract (dans le chaincode) pour créer ou mettre à jour les assets sur le registre du canal.

Pour déployer un smart contract, il faut passer par différentes étapes :

1. Empaqueter le smart contract dans un chaincode avant qu'il puisse être installé sur nos pairs.
2. Après avoir empaqueté le contrat intelligent en chaincode, nous pouvons installer le chaincode sur nos pairs. Il doit être installé sur chaque pair qui va approuver une transaction.
3. Après avoir installé le chaincode, il faut l'approuver pour chacune des deux organisations. Pour cela l'ensemble des membres du canal doit approuver le chaincode avant qu'il puisse être déployé.

Par défaut, pour qu'un chaincode soit approuvé il faut que la majorité des membres du canal l'approuve. Étant donné que nous n'avons que deux organisations sur le canal et que la majorité de 2 est bien évidemment 2 lui-même, nous devons approuver une définition de chaincode comme Org1 et Org2.

4. Une fois qu'un nombre suffisant d'organisations ont approuvé le chaincode, une organisation peut déployer le chaincode dans le canal.

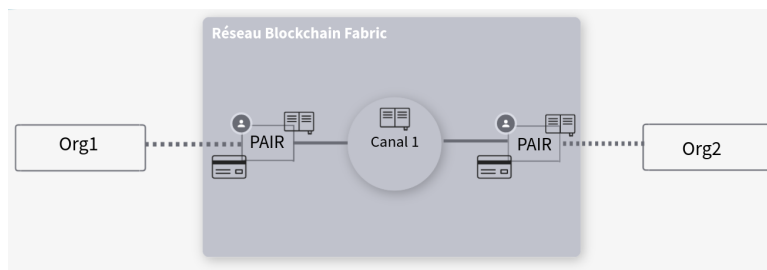


FIGURE 8 – Smart contract déployé sur l'ensemble du réseau

### 3.2.4 Interface Web

Dans notre application on aura deux utilisateurs, Alice et Bob. Ces deux utilisateurs sont des membres d'une organisation (Alice est membre de l'organisation 1 et Bob de l'organisation 2).

L'interface web permettra aux utilisateurs d'acheter ou de vendre des voitures, tout en invoquant le smart contract pour enregistrer les transactions dans la blockchain.

Les principales fonctionnalités de notre application sont :

- Visualiser ses propres voitures
- Visualiser les voitures mises en vente
- Ajouter une nouvelle voiture
- Mettre une voiture à la vente ou la retirer de la vente
- Modifier les propriétés de ses voitures
- Acheter une nouvelle voiture

#### 1. Visualiser ses propres voitures

Une fois que l'utilisateur est connecté, il va pouvoir lister les voitures qu'il possède en cliquant sur l'onglet *My cars*.



FIGURE 9 – Page d'accueil

Lorsqu'un utilisateur clique sur l'onglet *My cars*, cela signifie qu'une requête HTTP de type GET se déclenche vers le serveur pour récupérer la liste de voitures appartenant à l'utilisateur.

Une fois que la requête est reçue par le serveur, il vérifie l'identité de l'utilisateur et invoque le smart contract avec la méthode *queryOwner* qui permet d'extraire la liste des voitures appartenant à un utilisateur grâce à la blockchain.

Dès que le serveur recevra les données de la blockchain, il les transmettra directement à l'utilisateur.

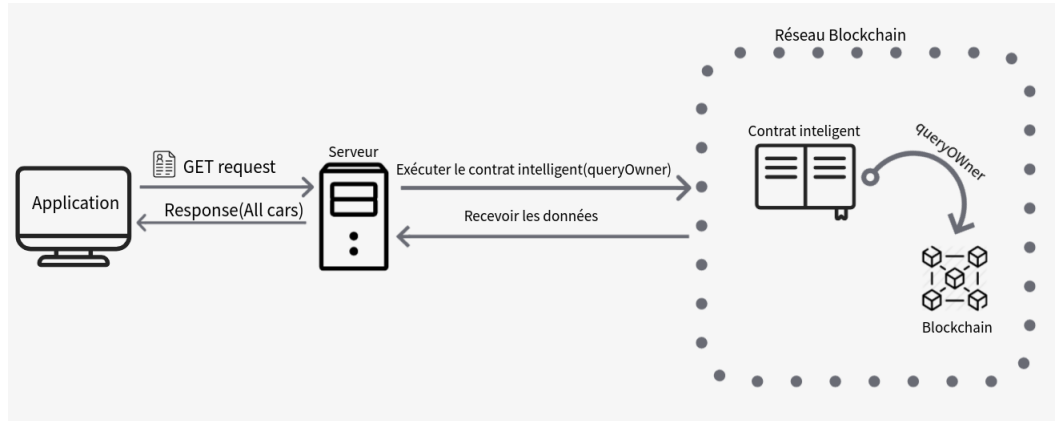


FIGURE 10 – Schéma illustrant le processus de récupération de la liste de voiture

## 2. Visualiser les voitures mises en vente

Pour visualiser les voitures mises en vente, l'utilisateur doit cliquer sur l'onglet *Cars To Buy*, ce qui déclenche une requête HTTP de type GET vers le serveur. Celui-ci va ensuite vérifier l'identité de l'utilisateur et invoquer le smart contract avec la méthode *queryOwner* qui va extraire la liste des voitures appartenant à d'autres utilisateurs.

Quand le serveur va recevoir les données à partir de la blockchain, il va seulement extraire les voitures mises en vente et les transmettre à l'utilisateur.

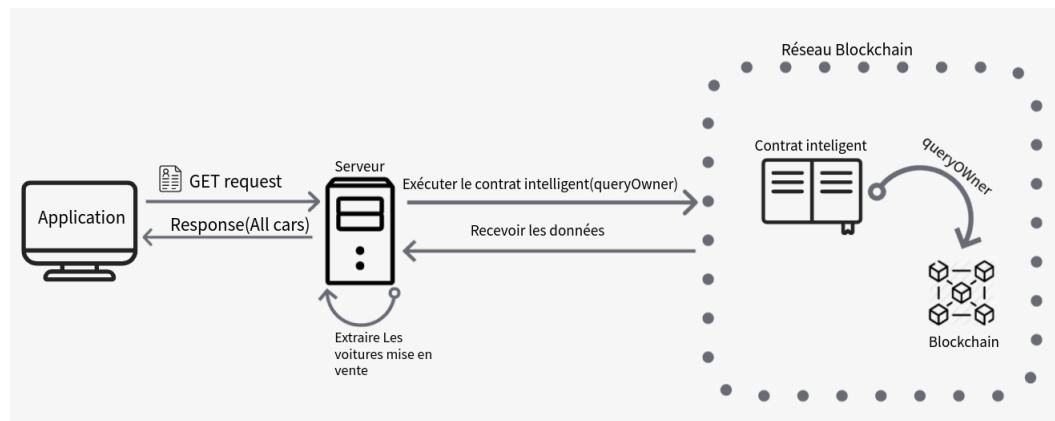


FIGURE 11 – Schéma illustrant le processus de récupération des voitures mise en vente

### 3. Ajouter une nouvelle voiture

Lorsque l'utilisateur clique sur l'onglet *My cars*, celui-ci obtient la liste de ses voitures ainsi qu'un bouton permettant l'ajout d'autres voitures.

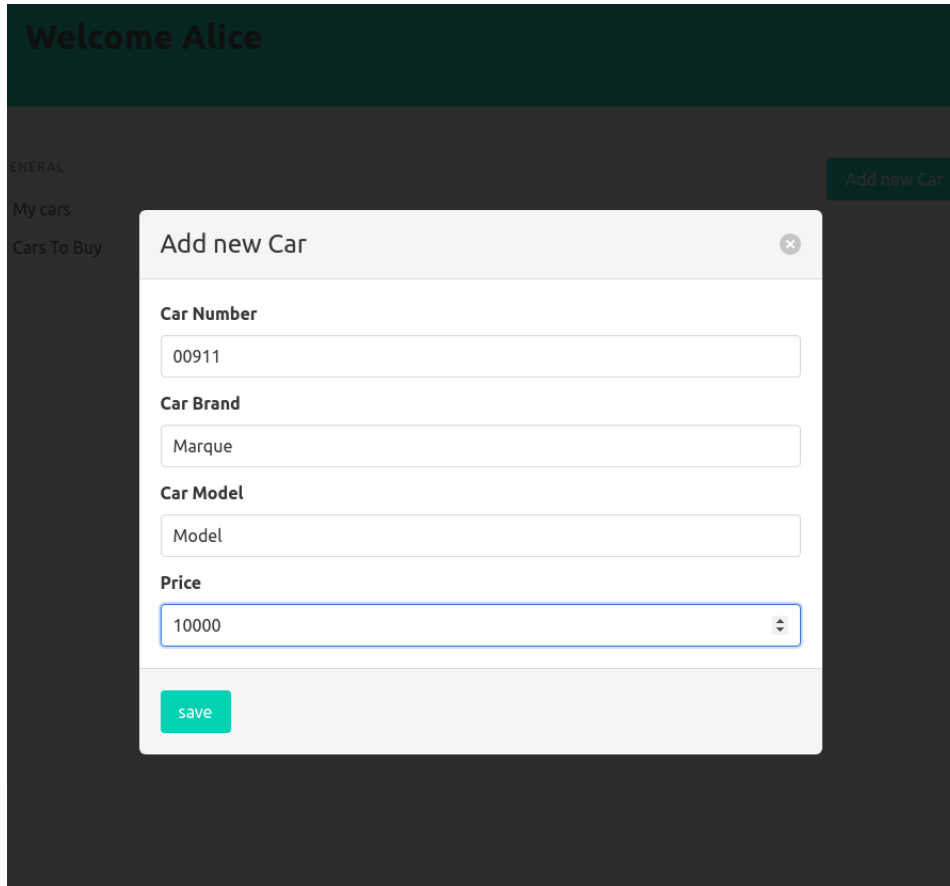
The image shows a web application interface. At the top, a dark green header displays 'Welcome Alice'. Below this, a sidebar on the left contains navigation links: 'GENERAL', 'My cars', and 'Cars To Buy'. On the right side of the main content area, there is a button labeled 'Add new Car'. A modal window is open in the center, titled 'Add new Car' with a close button (X) in the top right corner. The modal contains four input fields: 'Car Number' with the value '00911', 'Car Brand' with the value 'Marque', 'Car Model' with the value 'Model', and 'Price' with the value '10000'. At the bottom of the modal is a green button labeled 'save'.

FIGURE 12 – Formulaire pour l'ajout des voitures

Lorsque l'utilisateur clique le bouton *save* pour enregistrer la nouvelle voiture, une requête HTTP de type POST se déclenche vers le serveur. Le serveur va ensuite vérifier l'identité de l'utilisateur et invoquer le smart contract avec la méthode *sale* qui permet de communiquer avec la blockchain pour ajouter un nouveau bloc qui contient les informations de la voiture.

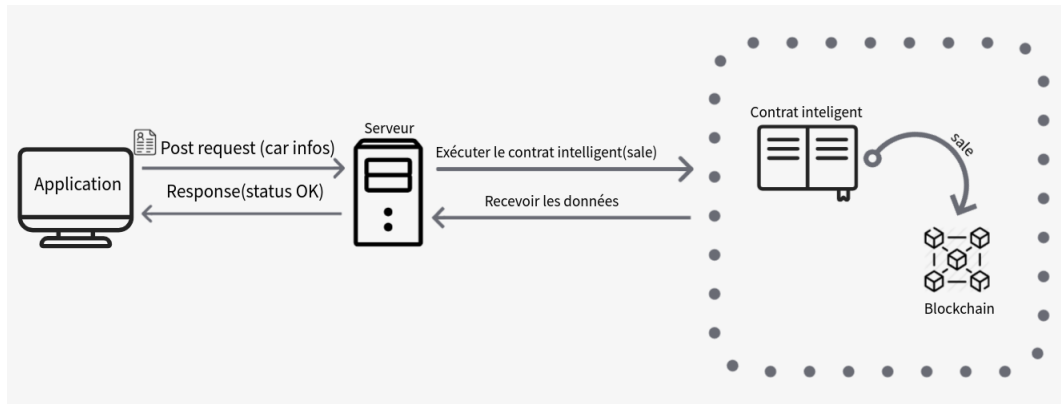


FIGURE 13 – Schéma illustrant le processus d’ajout d’une nouvelle voiture

Du côté de l’utilisateur, on remarque l’ajout de la voiture grâce à l’apparition de cette dernière dans la liste des voitures mais également grâce à un message de succès.

Au niveau d’Hyperledger Explorer on reçoit une notification qui indique qu’un nouveau bloc vient d’être ajouté dans la blockchain.

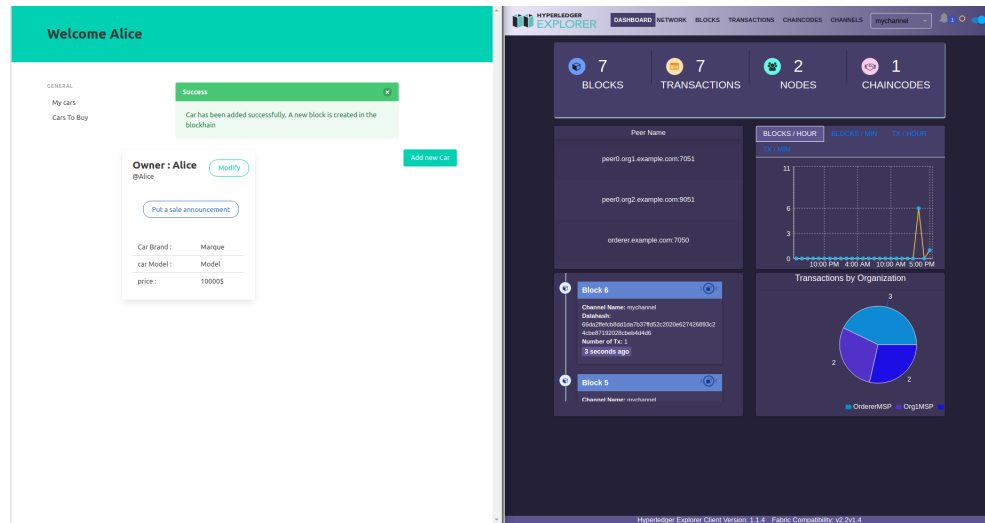


FIGURE 14 – L’ajout d’une nouvelle voiture avec succès

Explorons par la suite le contenu du bloc ajouté dans la blockchain

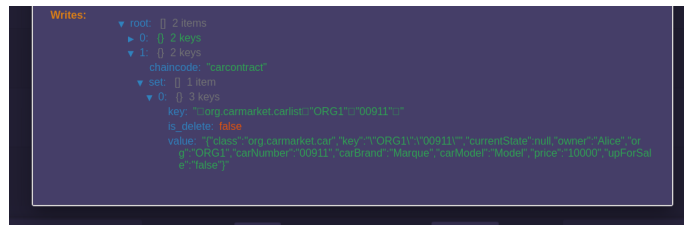


FIGURE 15 – Contenu du bloc ajouté dans la blockchain

On remarque que les données dans le bloc correspondent aux informations de la voiture ajoutée.

#### 4. Mettre une voiture à la vente ou non

Par défaut, quand un utilisateur ajoute une nouvelle voiture, elle n'est pas mise en vente. Pour réaliser cela, il suffit de cliquer sur le bouton *Put a sale announcement*

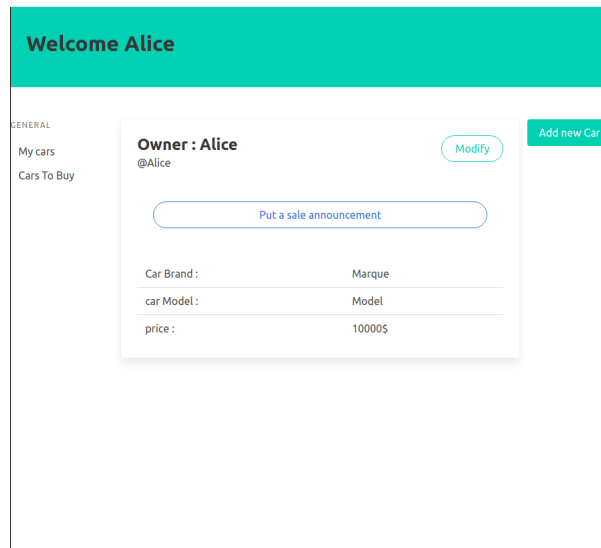


FIGURE 16 – Mettre une voiture à la vente

Une fois que l'utilisateur a cliqué sur *Put a sale announcement*, une requête HTTP de type PUT va se déclencher vers le serveur pour modifier l'état de la voiture et la mettre en vente.

Le serveur va vérifier l'identité de l'utilisateur et invoquer le smart contract avec la méthode *update* qui permet de modifier les propriétés d'une voiture. La voiture va donc changer d'état de *ne pas à vendre* vers *à vendre* et un nouveau bloc va s'ajouter dans la blockchain contenant les nouvelles informations de la voiture.



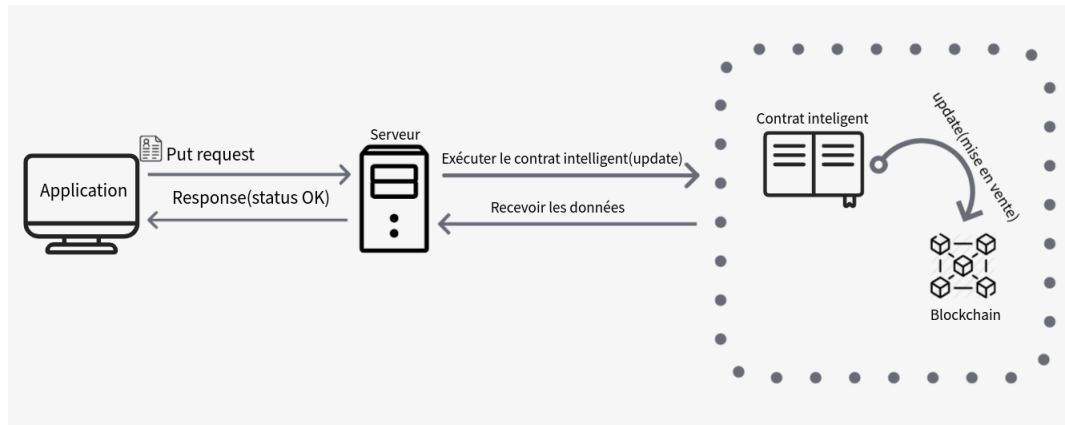


FIGURE 17 – Schéma illustrant le processus de mettre une voiture à la vente

Regardons ce qu'il se passe quand Alice met une voiture à la vente

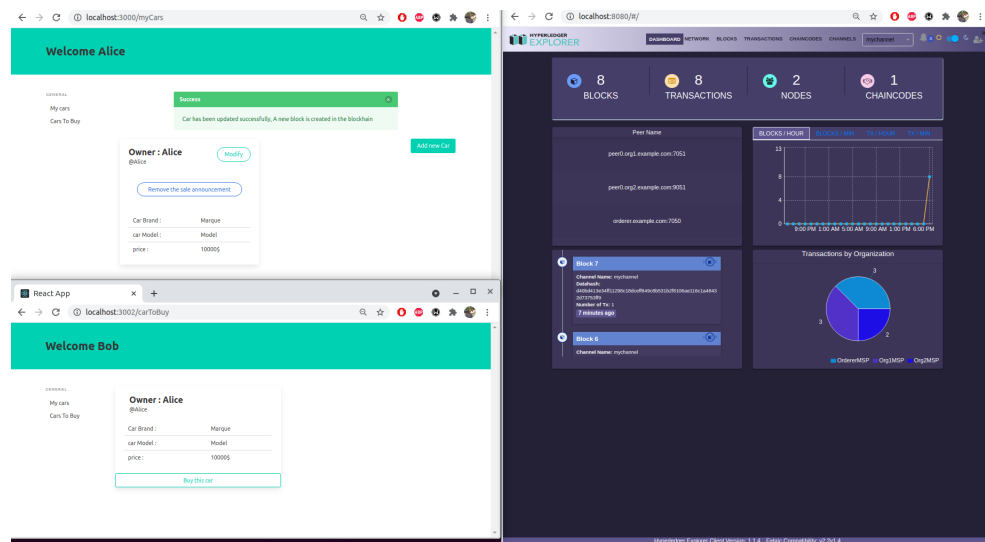


FIGURE 18 – Mise en vente d'une voiture avec succès

On constate que la voiture qu'Alice vient de mettre en vente est désormais présente dans la liste des voitures à acheter du côté de Bob. De plus, la blockchain a bien été mise à jour du côté d'Hyperledger Explorer, le tout sans avoir à actualiser la page.

Pour supprimer la voiture de la liste des voitures mises en vente, il suffit de cliquer sur le bouton *Remove the sale announcement* et le procédé que l'on a expliqué pour la mise en vente va se reproduire.

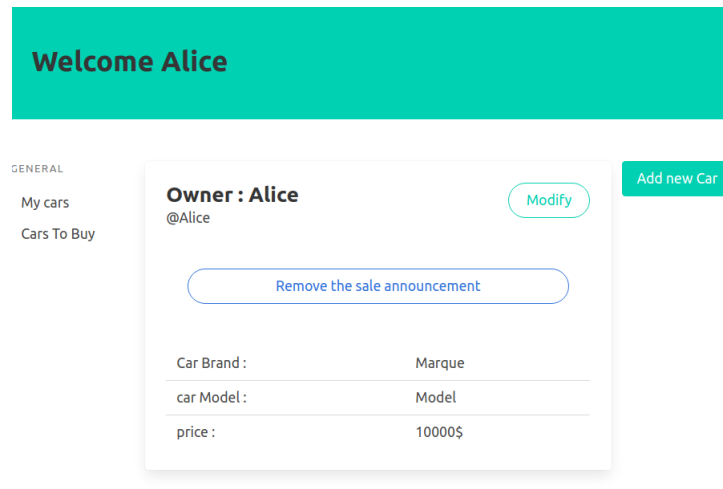


FIGURE 19 – Supprimer une voiture de la liste des voitures mises en vente

## 5. Modifier les propriétés de ses voitures

Un utilisateur peut facilement modifier les propriétés de ses voitures en cliquant sur le bouton *Modify*. Un formulaire apparaît à l'écran et permet à l'utilisateur de modifier les propriétés de la voiture (prix, etc.).

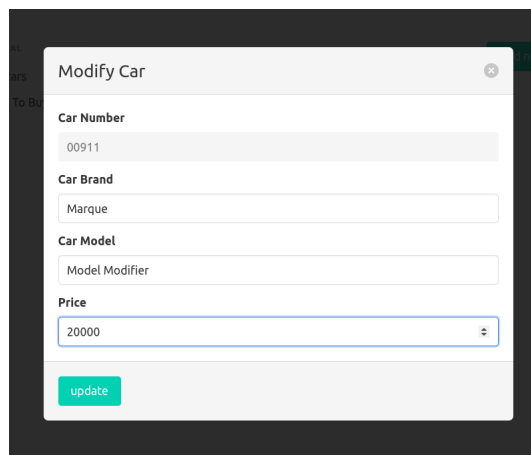


FIGURE 20 – Modifier les propriétés d'une voiture

Une fois que l'utilisateur a cliqué sur *update*, une requête HTTP de type PUT va se déclencher vers le serveur pour modifier les propriétés de la voiture.

Le serveur va vérifier l'identité de l'utilisateur et invoquer le smart contract avec la méthode *update* qui permet de modifier les propriétés d'une voiture. La voiture va changer de propriétés et un nouveau bloc va s'ajouter dans la blockchain contenant les nouvelles informations de la voiture.

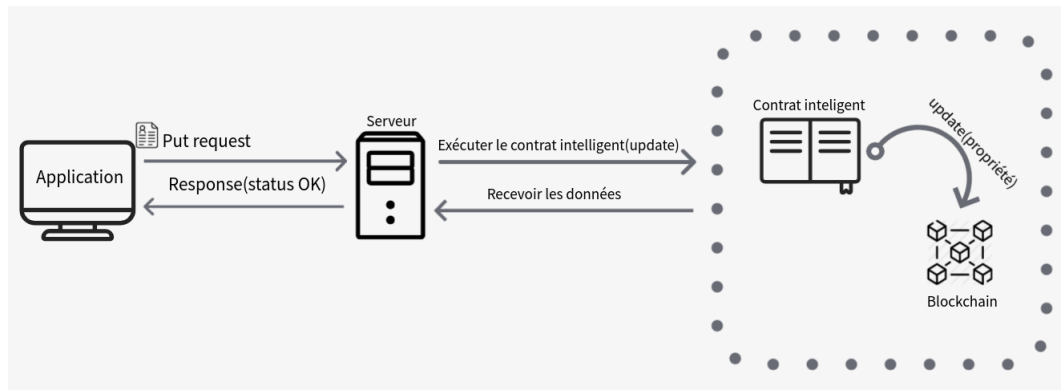


FIGURE 21 – Schéma illustrant la modification de propriétés d'un voiture

Du côté de l'utilisateur, on remarque la modification des propriétés de la voiture grâce à un message de succès, ainsi que la mise à jour d'Hyperledger Explorer sur la blockchain.

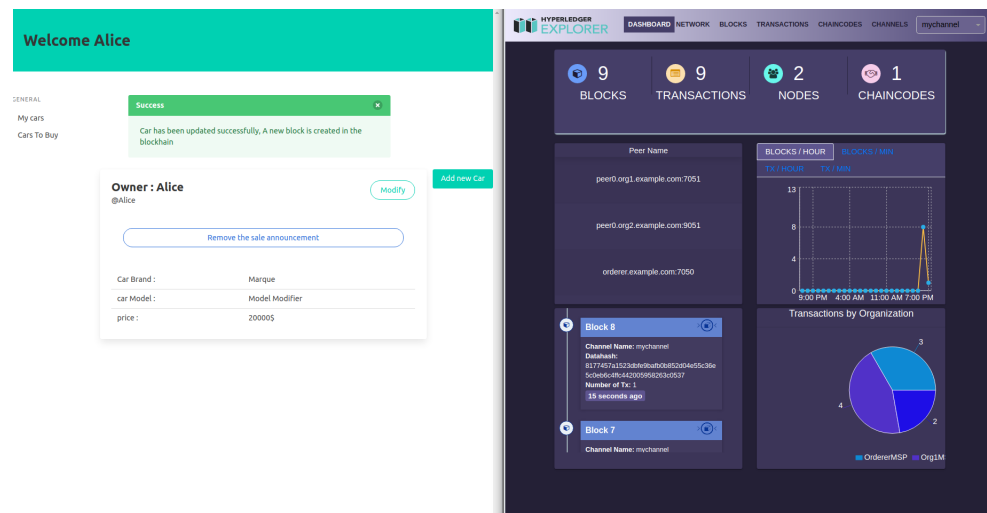


FIGURE 22 – Propriété de voiture modifier avec succès

## 6. Acheter une nouvelle voiture

Si un utilisateur veut acheter une nouvelle voiture, il doit aller sur l'onglet *Cars To Buy* qui lui permettra de voir la liste de toutes les voitures qu'il peut acheter. Pour cela il suffit de cliquer sur le bouton *Buy this car* et confirmer son achat, le processus d'achat va commencer.

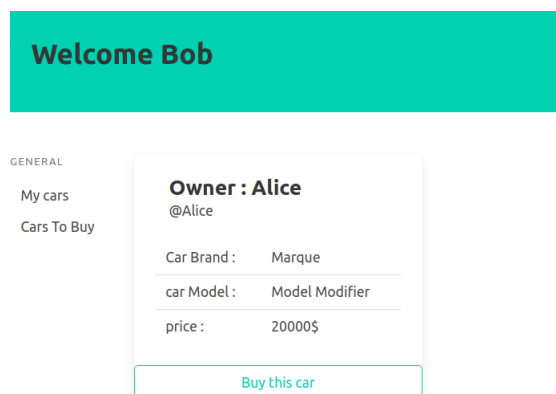


FIGURE 23 – Acheter une voiture

Lorsque l'utilisateur confirme l'achat du véhicule, une requête HTTP de type PUT se déclenche vers le serveur.

Celui-ci va vérifier l'identité de l'utilisateur et invoquer le smart contract avec la méthode *buy* qui s'occupe de l'achat de la voiture et du changement de propriétaire.

La transaction va s'enregistrer dans la blockchain sur un nouveau bloc et le serveur va informer l'utilisateur que la transaction a bien été effectuée.

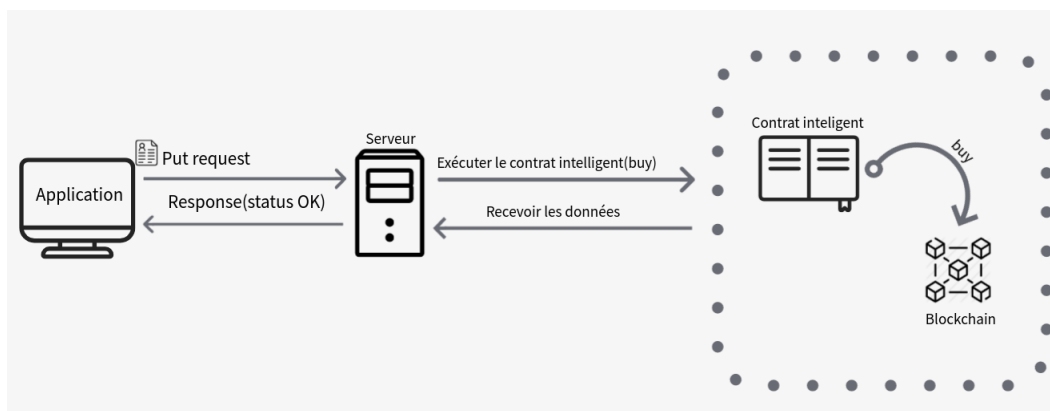


FIGURE 24 – Schéma illustrant l'achat d'une voiture

Regardons ce qu'il se passe quand Bob achète une voiture qui appartient à Alice

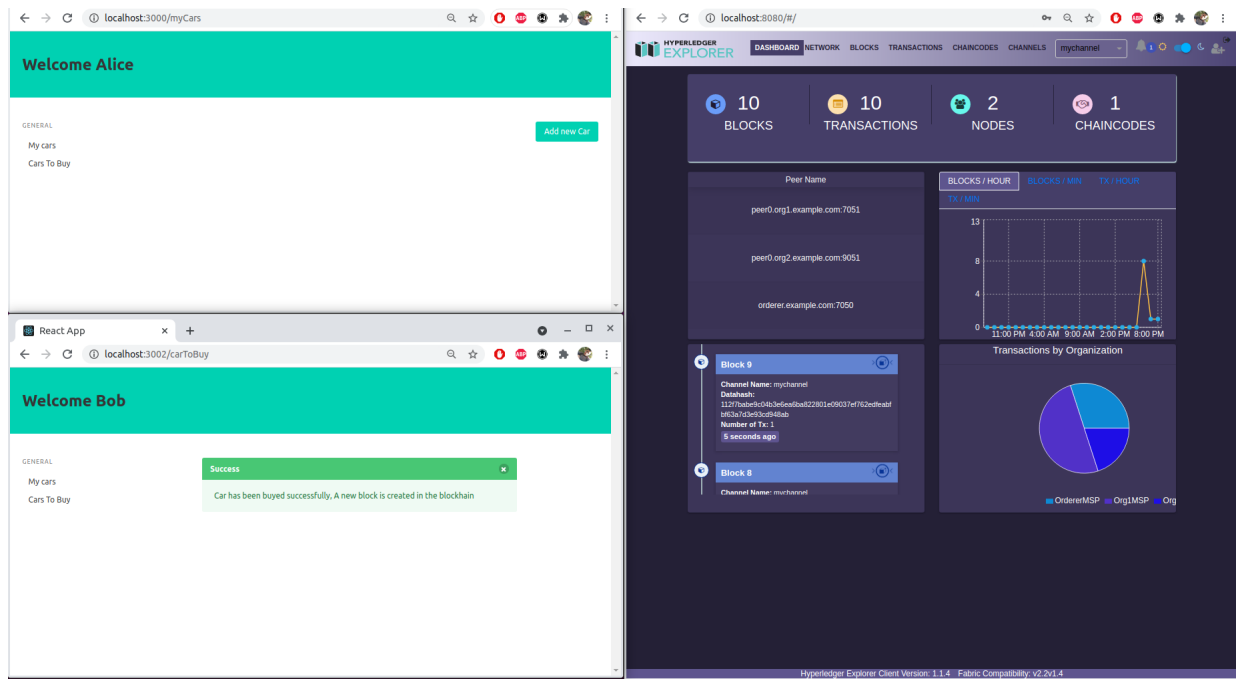


FIGURE 25 – Acheter une voiture avec succès

Bob a bien acheté la voiture, que l'on retrouvera dans son onglet *My cars*, tandis que du côté d'Alice la voiture a disparue.

Hyperledger Explorer quant à lui a bien été mis à jour.

## 4 Conclusion et perspectives

Pour conclure, ce TPE nous a permis de nous familiariser avec la notion de blockchain et plus particulièrement la manipulation d'Hyperledger Fabric.

L'utilisation de blockchain peut s'avérer très utile s'il existe un manque de confiance entre les individus grâce à son immuabilité et son infalsifiabilité sans aucun organe de contrôle.

Cette nouvelle technologie peut servir dans de nombreux domaines comme la logistique, les registres (par exemple la vente de voitures), les assurances ou encore la finance. La blockchain possède de nombreux avantages mais possède également quelques inconvénients comme le coût énergétique, le problème du stockage, le coût des cartes graphiques ou encore le faible nombre de transactions.

Hyperledger Fabric est l'un des frameworks blockchain les plus aboutis. Il se distingue de ses concurrents par le fait d'être une blockchain privée qui se concentre sur la façon de gérer le système et d'accorder des autorisations spéciales aux administrateurs pour choisir qui peut rejoindre la blockchain et quels éléments de données peuvent être lus ou ajoutés.

En ce qui concerne notre application de vente de voitures, il y a plusieurs perspectives d'amélioration :

- Mettre en place un système d'authentification
- Ajouter d'autres organisations
- Ajouter la plaque d'immatriculation du véhicule
- Mettre l'image des voitures pour avoir un aperçu de ce que l'on souhaite acheter
- Mettre en place un système de vérification de modèles qui permet de vendre uniquement des voitures qui existent
- Mettre en place un choix de devise
- Mettre en place un choix de langage.

## Références

- [1] M. Benjamin. «Que signifient hash et hachage?» *Cryptoast*[En ligne]  
Disponible sur : <https://cryptoast.fr/hash-hachage-bitcoin-blockchain/>
- [2] COIN24 «Comment fonctionne la blockchain?» *Coin24*[En ligne]  
Disponible sur : <https://coin24.fr/dictionnaire/blockchain/>
- [3] LEDGER «What is blockchain?» *Ledger*[En ligne]  
Disponible sur : <https://www.ledger.com/academy/blockchain/what-is-blockchain>
- [4] ELYAN Jean. «IBM mise sur Hyperledger Fabric pour son écosystème blockchain» *Le Monde Informatique*[En ligne]  
Disponible sur : <https://www.lemondeinformatique.fr/actualites/lire-ibm-mise-sur-hyperledger-fabric-pour-son-ecosysteme-blockchain-66731.html>
- [5] BOHIC Clément. «Hyperledger Fabric 1.0 : cap sur les blockchains de production» *ITespresso*[En ligne]  
Disponible sur : <https://www.itespresso.fr/hyperledger-fabric-blockchains-production-165298.html>
- [6] FOUCHAULT Aymeric. «Hyperledger Fabric, un premier pas vers la blockchain privée» *YouTube*[En ligne]  
Disponible sur : <https://www.youtube.com/watch?v=9Nzzqc7eM4k>
- [7] HYPERLEDGER «Getting Started - Install» *Hyperledger Fabric*[En ligne]  
Disponible sur : [https://hyperledger-fabric.readthedocs.io/en/latest/getting\\_started.html](https://hyperledger-fabric.readthedocs.io/en/latest/getting_started.html)
- [8] CHABERT Antoine. «Ethereum vs Hyperledger Fabric» *Chainhero*[En ligne]  
Disponible sur : <https://chainhero.io/fr/2018/07/ethereum-vs-hyperledger-fabric-2/>
- [9] GUYOMAR Robin, BOURGEAUX Maxence, KACIMI Souhail. «TPE. Blockchain - Hyperledger Fabric | 2020 - 2021» *Forge de l'université Le Havre Normandie*[En ligne]  
Disponible sur : <https://www-apps.univ-lehavre.fr/forge/duvallec/tpe-blockchain-hyperledger>