République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université SAAD DAHLAB de Blida 1
Faculté des Sciences
Département d'Informatique

Mémoire de Master

**Spécialité :** Ingénierie des Logiciels

**Présenté par :**

BERSALI Asma YETTOU Ikram

**THEME**

# DÉVELOPPEMENT D'UN SYSTÈME DE PAIEMENT ÉLECTRONIQUE MULTICANAL BASÉ SUR LA BLOCKCHAIN

Soutenu le **03/07/2025**, devant le jury composé de :

| | | | |
|---|---|---|---|
| M. | ALLALI Mustapha, | Chef de Projet - NAFTAL -, | Encadreur |
| Mme. | BOUDRAA Sawsen, | Maitre de conférences B - USDB -, | Promotrice |
| Mme. | TOUBALINE Nesrine, | Maitre de conférences A - USDB -, | Présidente |
| Mme. | TOBJI Rachida, | Maitre de conférences B - USDB -, | Examinatrice |

# Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude à Allah, le Miséricordieux et le Compatissant, pour nous avoir accordé la force, la patience et la sagesse nécessaires à la réalisation de ce travail.

Nous exprimons notre vive reconnaissance à **M.ALLALI Mustapha** pour son soutien précieux, qui nous a permis de nous intégrer rapidement au sein de l'entreprise dès les premiers jours de notre stage. Ses conseils avisés et sa disponibilité ont été essentiels à notre compréhension de l'organisation et au bon déroulement de notre expérience.

Nous adressons également nos remerciements à l'équipe de développement et au service sécurité de la société Naftal, qui nous ont accueillis et accompagnés tout au long de notre stage. particulièrement à **M.SI HADJ MOHAND Amine** pour avoir proposé ce projet ambitieux et pour la confiance qu'il nous a accordée afin de le mener à bien, et **M.RAFA Tarek** pour avoir répondu à toutes nos questions et pour ses explications claires, qui ont grandement contribué à l'amélioration de la qualité de notre projet. Sans oublier **M.BERSALI Mahmoud** pour sa précieuse proposition de stage.

Nos sincères remerciements s'adressent également à notre promotrice **Mme.BOUDRAA Sawsen** , ainsi à l'ensemble des enseignants et des professeurs du département d'informatique de l'Université de Blida pour leurs enseignements et leur accompagnement tout au long de notre parcours universitaire.

Nous exprimons notre gratitude aux membres du jury pour leur lecture attentive de notre mémoire ainsi que pour les remarques qu'ils nous adresseront afin d'améliorer notre travail.

Ce mémoire est dédié à nos familles, les familles **BERSALI** et **YETTOU**, ainsi qu'à toutes les personnes qui ont cru en nous, nous ont soutenus, et ont partagé notre enthousiasme et nos défis durant cette aventure. Nos remerciements s'adressent aussi à nos amis et à toutes celles et ceux qui nous ont témoigné leur gentillesse et leur soutien tout au long de ce parcours. Vos encouragements et votre confiance ont été une source de motivation inestimable.

Nous dédions ce travail aux générations futures, dans l'espoir qu'il serve de tremplin à leurs efforts et à leurs aspirations.

**Résumé**

Le paysage économique mondial est de plus en plus marqué par la digitalisation et la modernisation accélérée. L'Algérie s'inscrit activement dans l'intégration de ces innovations et poursuit sa transformation numérique dans ses différents secteurs économiques. Dans ce contexte, la société Naftal spécialisée dans la commercialisation et la distribution de produits pétroliers à l'échelle nationale, a identifié le besoin de moderniser leur système de paiement au sein de ses stations-service. Cette thèse propose une solution innovante à cette problématique, en explorant la conception et la mise en œuvre d'un système de paiement électronique sécurisé, basé sur la technologie blockchain. L'objectif est d'optimiser l'efficacité opérationnelle tout en garantissant l'intégrité, la traçabilité et la sécurité des données de transactions.

La solution repose sur une application mobile destinée aux clients, leur permettant de suivre leurs transactions et d'effectuer leurs paiements de manière simple, sécurisée et confidentielle dans les stations-service de Naftal, en interaction avec une application web qui assure la fluidité des échanges et le traitement sécurisé des paiements. Les transactions sont stockées sur une infrastructure technologique basée sur Hyperledger Fabric, une blockchain privée et permissionnée, offrant un haut niveau de sécurité, de confidentialité et de résilience.

Ce projet illustre le potentiel transformateur des technologies numériques dans la gestion des transactions commerciales .Il contribue concrètement à l'effort national de digitalisation, tout en posant les bases de futures évolutions technologiques au service de l'économie algérienne.

**Mots-Clés :** Digitalisation, Naftal, Paiement électronique, Blockchain, Sécurité des données, Hyperledger Fabric, Gestion des transactions, Communication en temps réel .

**Abstract**

The global economic landscape is increasingly shaped by rapid digitalization and modernization. Algeria is actively engaging in the integration of these innovations and is pursuing its digital transformation across various economic sectors.In this context, Naftal a company specializing in the nationwide marketing and distribution of petroleum products, has identified the need to modernize its payment system at its service stations. This thesis proposes an innovative solution to this challenge by exploring the design and implementation of a secure electronic payment system based on blockchain technology. The objective is to optimize operational efficiency while ensuring the integrity, traceability, and security of transaction data.

The proposed solution relies on a mobile application designed for customers, allowing them to track their transactions and make payments in a simple, secure, and confidential manner at Naftal service stations, in interaction with a web application that ensures seamless exchanges and secure payment processing.Transactions are stored on a technology infrastructure based on Hyperledger Fabric, a private and permissioned blockchain, providing a high level of security, confidentiality, and resilience.

This project illustrates the transformative potential of digital technologies in the management of commercial transactions.It makes a concrete contribution to the national digitalization effort, while laying the foundations for future technological developments serving the Algerian economy.

**Keywords:** Digitalization, Naftal, Electronic payment, Blockchain, Data security, Hyperledger Fabric, Transaction management, Real-Time Communication .

## ملخص

يشهد المشهد الاقتصادي العالمي تحوّلاً متسارعاً نحو الرقمنة والتحديث، حيث باتت الابتكارات الرقمية عاملاً رئيساً في إعادة تشكيل مختلف القطاعات. تنخرط الجزائر بشكل فعّال في جهود التحول الرقمي، ساعيةً إلى دمج هذه التكنولوجيات الحديثة ضمن نسيجها الاقتصادي. ومن بين الفاعلين الاقتصاديين البارزين، برزت شركة نفطال، المتخصصة في تسويق وتوزيع المنتجات البترولية على المستوى الوطني، والتي عبّرت عن حاجتها لتحديث نظام الدفع في محطاتها للخدمات.تقترح هذه المذكرة حلاً مبتكراً لهذه الإشكالية، من خلال ابتكار و تصميم نظام دفع إلكتروني آمن قائم على تكنولوجيا البلوكشين.

يهدف هذا النظام إلى تحسين الكفاءة التشغيلية مع ضمان سلامة المعاملات، قابليتها للتتبع، وسريتها.تعتمد الحلول المقترحة على تطبيق محمول مخصص للزبائن، يمكّنهم من تتبع معاملاتهم وتنفيذ عمليات الدفع بسهولة وأمان وسرية داخل محطات نفطال. ويتكامل هذا التطبيق مع منصة ويب مخصّصة لضمان سلاسة المعالجة وأمن تدفق العمليات المالية.يتم تخزين جميع المعاملات على بنية تحتية قائمة على Hyperledger Fabric، وهي شبكة بلوكشين خاصة ومُرخّصة توفّر مستويات عالية من الأمان والخصوصية والمرونة.

يجسّد هذا المشروع الإمكانيات التحويلية التي توفّرها التكنولوجيات الرقمية في مجال إدارة المعاملات التجارية، ويساهم بشكل فعلي في جهود الرقمنة الوطنية، كما يمهّد الطريق نحو تطورات تكنولوجية مستقبلية داعمة للاقتصاد الجزائري .



**الكلمات المفتاحية**:الرقمنة، نفطال، الدفع الإلكتروني، البلوكشين، أمن البيانات Hyperledger Fabric، إدارة المعاملات المالية ، التواصل في الزمن الحقيقي.

# Table des matières

**Table des figures** **i**

**Liste des tableaux** **iv**

**Liste des abréviations** **v**

**Introduction Générale** **1**

**1 La Technologie Blockchain et le Paiement Électronique dans les Entreprises** **4**

1.1 Introduction . . . . . 4
1.2 Qu'est-ce que la technologie Blockchain ? . . . . . 4
1.2.1 Aperçu sur cette Technologie . . . . . 4
1.2.2 Définition de la Blockchain . . . . . 5
1.2.3 Architecture des Couches de Blockchain . . . . . 5
1.2.4 Structure d'une blockchain . . . . . 6
1.2.5 Le Trilemme de la Blockchain . . . . . 8
1.3 Les différents types de blockchain . . . . . 8
1.3.1 Blockchain publique . . . . . 9
1.3.2 Blockchain privée . . . . . 9
1.3.3 Blockchain consortium . . . . . 9
1.3.4 Blockchain hybride . . . . . 9
1.4 Mécanismes de consensus . . . . . 9
1.4.1 Proof of Work . . . . . 10
1.4.2 Proof of Stake . . . . . 11
1.4.3 Reliable, Replicated, Redundant, And Fault-Tolerant . . . . . 12
1.4.4 Practical Byzantine Fault Tolerance . . . . . 13
1.5 Les solutions blockchain . . . . . 14
1.5.1 Bitcoin . . . . . 14
1.5.2 Ethereum . . . . . 15
1.5.3 Hyperledger Fabric . . . . . 17
1.6 Le paiement électronique en Algérie . . . . . 22

1.6.1 État de lieux de l'e-paiement en Algérie . . . 22
1.6.2 Impacts opérationnels sur les entreprises . . . 23
1.7 L'impact de la technologie de blockchain sur l'entreprise . . . 23
1.8 Conclusion . . . 24

**2 Étude de la Situation Existante et Analyse des Besoins 25**
2.1 Introduction . . . 25
2.2 Étude de la situation existante . . . 25
2.2.1 Le processus d'achat dans les stations-service . . . 26
2.2.2 Gestion des ventes et paiement dans les stations-service . . . 26
2.3 Travaux connexes . . . 27
2.4 Analyse des besoins . . . 29
2.4.1 Limitations des méthodes actuelles . . . 29
2.4.2 Intégration du paiement électronique et de la technologie blockchain dans la gestion de paiement dans les stations-service de NAFTAL . 29
2.4.3 Défis liés à l'intégration d'une nouvelle solution de E-Paiement . . . 30
2.5 Conclusion . . . 31

**3 Conception d'une Solution Blockchain Adaptée aux Besoins de l'Entreprise Naftal 32**
3.1 Introduction . . . 32
3.2 Description de l'architecture et du comportement général du Système . . . 32
3.3 Les exigences fonctionnelles et non fonctionnelles du Système . . . 35
3.3.1 Exigences fonctionnelles . . . 35
3.3.2 Exigences non fonctionnelles : . . . 36
3.4 Présentation et description des cas d'utilisation . . . 36
3.4.1 Acteurs . . . 37
3.4.2 Diagramme de cas d'utilisation global . . . 37
3.5 Présentation de la structure des données du système . . . 38
3.6 Architecture de l'écosystème applicatif . . . 38
3.7 Exploration des processus interactifs du système . . . 40
3.8 Conclusion . . . 42

**4 La Solution Blockchain : Implémentation et Déploiement 43**
4.1 Introduction . . . 43
4.2 Environnement de développement . . . 43
4.3 Implémentation de la blockchain . . . 44
4.3.1 Aperçu des choix technologiques . . . 45
4.3.2 Processus de déploiement du réseau Hyperledger Fabric . . . 47
4.4 Mise en œuvre des microservice . . . 56

4.4.1 Technologies utilisées pour l'API . . . . . 57
4.4.2 Microservice de Stockage . . . . . 57
4.4.3 Microservice pour socket . . . . . 57
4.4.4 Microservice pour Paiement . . . . . 58
4.5 Implémentation de l'application mobile . . . . . 58
4.5.1 Technologies utilisées pour l'application mobile . . . . . 58
4.5.2 Mécanismes de sécurité et contrôle d'accès . . . . . 59
4.5.3 Communication avec les microservices . . . . . 59
4.6 Quelques interfaces du produit final . . . . . 61
4.7 Conclusion . . . . . 67

**Conclusion générale** **68**

**Bibliographie** **70**

# Table des figures


1.1 Client-Serveur et Peer-to-Peer : Deux Modèles d'Architecture Réseau . . . 5
1.2 Architecture des Couches de Blockchain [3] . . . . . 6
1.3 Structure de Blockchain [5] . . . . . 7
1.4 Mécanismes de consensus dans la blockchain . . . . . 10
1.5 Algorithme proof of work [13] . . . . . 11
1.6 Algorithme proof of stake . . . . . 11
1.7 Transitions d'état des nœuds dans le protocole Raft[17] . . . . . 12
1.8 Processus de Protocole PBFT . . . . . 13
1.9 Relation entre clé privée et clé publique et adresse bitcoin . . . . . 15
1.10 Mécanisme de Validation des Transactions dans un Réseau Bitcoin . . . . . 15
1.11 Interaction entre l'utilisateur et le réseau Ethereum via le contrat intelligent [26] . . . . . 16
1.12 Intégration du MSP dans l'architecture Hyperledger Fabric [31] . . . . . 17
1.13 Architecture d'un Réseau Hyperledger Fabric : Répartition des Peers, CA et Service d'Ordonnancement autour des Channels [36] . . . . . 19
1.14 Organisation interne d'un channel Hyperledger Fabric : structuration des peers,du ledger et du chaincode [36] . . . . . 20
1.15 Hyperledger Fabric CA [37] . . . . . 20
1.16 Flux de transaction [38] . . . . . 21
1.17 Évolution de paiement par mobile en Algérie . . . . . 22

2.1 Architecture du Système d'Approvisionnement et de Gestion des Transactions en Station-Service NAFTAL . . . . . 27

3.1 Présentation de l'architecture globale du système . . . . . 33
3.2 Présentation de L'interaction entre l'application mobile client et l'application web de la station-service . . . . . 34
3.3 Architecture des Flux de Transactions entre l'application web de la station-service et le Réseau Hyperledger Fabric . . . . . 35
3.4 Diagramme de cas d'utilisation global . . . . . 37
3.5 Diagramme de Classe global . . . . . 38

3.6 Architecture de l'écosystème applicatif . . . 38
3.7 Diagramme de séquence de processus de l'authentification . . . 40
3.8 Diagramme de séquence de traitement et de stockage des transactions . . . 41

4.1 Logos des outils d'implémentation d'un réseau Hyperledger Fabric de test . 45
4.2 L'interopérabilité entre les différents outils de déploiement dans un réseau Hyperledger Fabric : Kubernetes, Istio et Docker . . . 46
4.3 Le flux d'interaction entre les différents composants du réseau Hyperledger Fabric . . . 47
4.4 les variables d'environnement . . . 49
4.5 L'autorité de certification pour l'organisation Org1 . . . 49
4.6 enregistrement d'un utilisateur de type peer . . . 49
4.7 La creation des peers . . . 50
4.8 Déploiement des Orderer . . . 51
4.9 Enregistrement de l'administrateur Orderer auprès de la CA . . . 52
4.10 Création de l'identité de signature . . . 52
4.11 Création de l'identité TLS . . . 52
4.12 Enregistrement de l'administrateur Org1 . . . 52
4.13 Création de l'identité d'administration Org1 . . . 52
4.14 Création de Certificats pour l'organisation Orderer . . . 53
4.15 La structure du Chaincode pour les transaction des clients . . . 54
4.16 Déploiement du Chaincode . . . 55
4.17 Vue des pods en cours d'exécution dans le cluster Kubernetes via l'interface utilisateur Lens . . . 55
4.18 Déploiement de réseau Hyperledger Fabric avec Kubernetes . . . 56
4.19 Logos des outils d'implémentation d'une API . . . 57
4.20 Logos des outils d'implémentation d'une application mobile . . . 58
4.21 Fonction de création d'un order paypal . . . 60
4.22 Fonction pour récupérer les prix des produits . . . 60
4.23 Interconnecté avec une API Socket . . . 60
4.24 Interface de démarrage du processus de paiement . . . 61
4.25 Interface d'interconnexion avec l'application Naftalclick via QR code . . . 61
4.26 Démarrage du processus de paiement – côté application mobile : authentification via empreinte digitale . . . 62
4.27 Interface d'authentification avec le numéro de téléphone et le code de vérification . . . 62
4.28 Interface d'ajout de produit et de génération du ticket de commande . . . 63
4.29 Interface de sélection du moyen de paiement et validation par QR code . . . 63
4.30 Interface de traitement de la transaction . . . 64

4.31 Interface de confirmation des produits sélectionnés . . . . . 64
4.32 Connexion et validation du paiement via PayPal . . . . . 65
4.33 Validation finale de la transaction – paiement effectué avec succès . . . . . 65
4.34 Message de succès du paiement – côté web . . . . . 66
4.35 Localisation des stations-service et présentation de l'application . . . . . 66
4.36 Vue globale des transactions et des statistiques sur les opérations effectuées par le client . . . . . 67

# Liste des tableaux

2.1 Études comparatives sur l'utilisation de la technologie blockchain dans le partage d'informations comptables, financières et logistiques en entreprise. 28

4.1 Caractéristiques techniques des équipements utilisés . . . . . 44
4.2 Tableau comparatif des configurations SmartBFT et etcdRaft . . . . . 53

# Liste des abréviations

- **API** : *Application Programming Interface*
- **BFT** : *Byzantine Fault Tolerance*
- **BL** : *Bon de Livraison*
- **BLF** : *Bon de Livraison Facturé*
- **BLR** : *Bon de Livraison de Retour*
- **BTC** : *Bitcoin*
- **CIB** : *Carte Interbancaire*
- **DApp** : *Application Décentralisée*
- **DCSI** : *Direction Centrale des Systèmes d'Information*
- **ECDSA** : *Elliptic Curve Digital Signature Algorithm*
- **ERP** : *Enterprise Resource Planning*
- **EVM** : *Ethereum Virtual Machine*
- **GPL** : *Gaz de Pétrole Liquéfié*
- **LDAP** : *Lightweight Directory Access Protocol*
- **MSP** : *Membership Service Provider*
- **OSN** : *Ordering Service Node*
- **P2P** : *Peer-to-Peer*
- **PBFT** : *Practical Byzantine Fault Tolerance*
- **PME** : *Petite et Moyenne Entreprise*
- **PoS** : *Proof of Stake*
- **PoW** : *Proof of Work*
- **PVA** : *Point de Vente Agréé*
- **QR Code** : *Quick Response Code*
- **RAFT** : *Reliable, Replicated, Redundant, And Fault-Tolerant*
- **RMI** : *Relevé Mobile d'Identité*
- **RPC** : *Remote Procedure Call*

- **SCF** : *Supply Chain Finance*
- **SDK** : *Software Development Kit*
- **SGBD** : *Système de Gestion de Base de Données*
- **SHA** : *Secure Hash Algorithm*
- **SPA** : *Société par Actions*
- **TIC** : *Technologies de l'Information et de la Communication*
- **TLS** : *Transport Layer Security*
- **TPE** : *Terminal de Paiement Électronique*
- **TPS** : *Transactions Par Seconde*

# Introduction Générale

## Contexte

Depuis 1982, Naftal est la première entreprise en Algérie spécialisée dans la distribution et la commercialisation des produits pétroliers dérivés sur le marché national. Au fil du temps, l'automatisation des processus financiers est devenue une nécessité pour répondre à la demande croissante du marché. À l'instar d'autres entreprises historiques, elle est confrontée au défi de moderniser son infrastructure technologique. Pour des institutions comme Naftal, la transformation numérique est devenue pressante et essentielle afin d'améliorer l'efficacité opérationnelle et de se distinguer de la concurrence. Cette transformation doit être accompagnée d'une sécurisation renforcée pour la protection des données, à une époque où les attaques et les vulnérabilités se multiplient, surtout pour les entreprises. Des technologies émergentes comme la blockchain, en plein essor, apparaissent comme une solution prometteuse grâce à leur robustesse et leur fiabilité, permettant de résoudre ces problèmes de sécurité.

## Problématique

Le système actuel repose sur une architecture centralisée basée sur un Système de Gestion de Base de Données (SGBD) traditionnel. Dans ce modèle, le pouvoir est entièrement centralisé conférant à l'administrateur la capacité de modifier ou de supprimer des données, ce qui compromet la sécurité et l'intégrité des informations. La traçabilité des opérations dépend exclusivement de la fiabilité des journaux internes (logs), lesquels peuvent être modifiés ou altérés, ce qui réduit considérablement la transparence du système. De plus, les systèmes traditionnels sont vulnérables non seulement aux pannes matérielles, mais aussi aux attaques ciblées contre le serveur central, rendant les données totalement inaccessibles en cas de défaillance. Ainsi, la confiance accordée à l'intégrité du système repose essentiellement sur l'administrateur et sur l'infrastructure sous-jacente, ce qui représente une faiblesse majeure en matière d'aide à la décision.

# Objectif

L'objectif principal de cette mémoire est de répondre aux besoins de Naftal, qui cherche à mettre en place une solution plus intelligente et performante pour la gestion des transactions et de la vente dans ses stations-service. Cette solution proposée consiste à mettre en place un système de « Self-service » au sein des stations-service Naftal. En prenant en compte les aspects confidentiels et la protection des informations des clients, une tâche complexe mais essentielle pour préserver l'image et la pérennité de l'entreprise.Pour cela, nous avons développé une application mobile dédiée aux clients, permettant de consulter l'historique de leurs transactions et d'interagir avec une application web installée dans les stations-service pour effectuer leurs paiements de manière autonome que ce soit en magasin ou à la pompe, sans avoir à passer par la caisse, et en toute sécurité.

L'intégration de la technologie blockchain dans le processus de traitement et de stockage des transactions permet d'offrir un haut niveau de sécurité, de confidentialité et de résilience. L'infrastructure technologique repose sur Hyperledger Fabric, une blockchain privée et permissionnée, garantissant le contrôle des accès et la traçabilité des opérations. Une protection renforcée contre les altérations de données et une robustesse face aux défaillances techniques ou aux attaques ciblées.

# Organisation du mémoire

La structure de notre mémoire couvre quatre composantes essentielles de notre démarche de travail :

- **« La technologie blockchain et le paiement électronique dans les entreprises »** : Dans cette première partie, nous présentons en détail la technologie blockchain et les différentes solutions existantes dans ce domaine. Nous analysons également le contexte du paiement électronique en Algérie, en mettant en lumière son impact sur les entreprises. Cette section vise à comprendre comment ces technologies innovantes peuvent être intégrées et apporter de plus au sein de l'entreprise.
- **« Étude de la situation actuelle et analyse des besoins »** : La deuxième partie est consacrée à une analyse approfondie de la gestion actuelle des ventes dans les stations-service. Nous recueillons des informations clés relatives aux besoins, aux contraintes et aux défis rencontrés par ces structures, notamment en ce qui concerne l'intégration d'une nouvelle solution de paiement. Ce travail d'analyse est fondamental pour concevoir une solution réellement adaptée et efficace.
- **« Conception d'une solution Blockchain adaptée aux besoins de l'entreprise Naftal »** : Dans cette troisième partie, nous abordons la conception théorique

de la solution sur la base des besoins identifiés, nous définissons l'architecture globale du système et les fonctionnalités à implémenter. Cette étape vise à établir une solution technologique cohérente, sécurisée et conforme aux exigences spécifiques de Naftal.

- « **Implémentation et déploiement de la solution Blockchain** » : Enfin, la quatrième partie décrit en détail la phase de mise en œuvre de la solution proposée. Nous présentons le processus de test, de validation et de déploiement progressif de la solution. Cette phase comprend le développement des modules fonctionnels, la création des interfaces utilisateur (web et mobile), l'intégration avec les systèmes existants, ainsi que la mise en place des contrats intelligents. L'objectif est de démontrer comment cette solution répond efficacement aux enjeux identifiés lors de l'analyse, tout en apportant une réelle valeur ajoutée à l'entreprise.

Nous concluons notre travail par une réflexion mettant en évidence l'importance d'utiliser les nouvelles technologies dans la gestion des systèmes technologiques au service de l'économie algérienne. Nous proposons également des pistes d'amélioration qui pourraient renforcer l'efficacité du système à l'avenir.

# Chapitre 1

# La Technologie Blockchain et le Paiement Électronique dans les Entreprises

## 1.1 Introduction

Dans un monde où la numérisation des processus financiers est devenue une nécessité incontournable pour les entreprises souhaitant rester compétitives et efficaces, deux piliers technologiques majeurs émergent au cœur de cette transformation : le paiement électronique et la technologie blockchain. Si la transition vers les paiements électroniques s'accélère, offrant aux entreprises et aux consommateurs des alternatives plus rapides, efficaces aux méthodes traditionnelles, la technologie blockchain promet quant à elle de révolutionner la manière dont les transactions sont sécurisées, vérifiées et enregistrées. Dans ce chapitre, dans un premier temps nous présenterons cette technologie, ses différents types et leurs mécanismes de consensus. Ensuite, nous expliquons l'architecture des différentes solutions blockchain emblématiques telles que Bitcoin, Ethereum et Hyperledger. Après, nous passerons au paysage du paiement électronique en Algérie. Nous étudierons l'état actuel de l'adoption des paiements numériques dans le contexte algérien, ainsi que leurs impacts opérationnels sur les entreprises. Enfin, nous conclurons ce chapitre en présentant les avantages que cette technologie peut apporter aux entreprises qui choisissent de l'adopter dans leur activité.

## 1.2 Qu'est-ce que la technologie Blockchain ?

### 1.2.1 Aperçu sur cette Technologie

En général, il existe deux grands types de réseaux informatiques : les réseaux centralisés et les réseaux décentralisés. Ces réseaux permettent aux ordinateurs d'être connectés

entre eux pour qu'ils puissent partager des ressources, comme des fichiers ou des périphériques (figure 1.1).

Dans une architecture peer-to-peer (P2P), les ordinateurs sont connectés entre eux de manière à ce que chacun puisse partager tout ou partie de ses ressources. Ainsi dans une architecture client-serveur, un ou plusieurs ordinateurs appelés serveurs sont chargés de stocker les données et de gérer les ressources. Les autres ordinateurs se connectent à ces serveurs pour accéder aux informations ou aux services dont ils ont besoin.

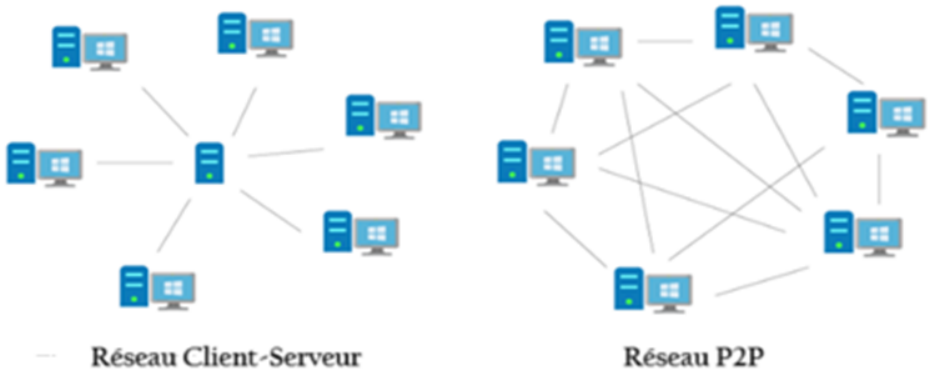


**FIGURE 1.1** – Client-Serveur et Peer-to-Peer : Deux Modèles d'Architecture Réseau

La technologie blockchain est un nouveau succès dans l'informatique sécurisée sans autorité centralisée dans un environnement P2P ouvert.

### 1.2.2 Définition de la Blockchain

D'après le mathématicien Jean Paul Delahaye [1] : « L'idée d'un grand cahier informatique, partagé, infalsifiable et indestructible du fait même de sa conception est au cœur d'une nouvelle révolution, celle de la blockchain » [2].

Le Blockchain est une technologie qui permet la sécurisation des transactions d'un écosystème en rendant les données transparentes et non modifiables via la construction d'un registre distribué, infalsifiable, pour la parfaite traçabilité des échanges d'information [3].

### 1.2.3 Architecture des Couches de Blockchain

La technologie blockchain est organisée en couches pour répondre aux défis techniques et fonctionnels de la décentralisation (la figure 1.2). La couche de données est responsable de la création et de la structuration des blocs, la couche réseau gère la communication

1. **Jean-Paul Delahaye** est l'un des premiers intellectuels francophones à avoir vulgarisé en profondeur le fonctionnement des blockchains. Dans ses écrits, il s'est attaché à expliquer le rôle fondamental des registres distribués, des algorithmes de consensus (comme le proof-of-work) [1]

et la synchronisation entre les nœuds du réseau distribué, et enfin, la couche application regroupe les différentes applications de cette technologie. Des informations détaillées sur chacune de ces couches sont présentées ci-dessous :

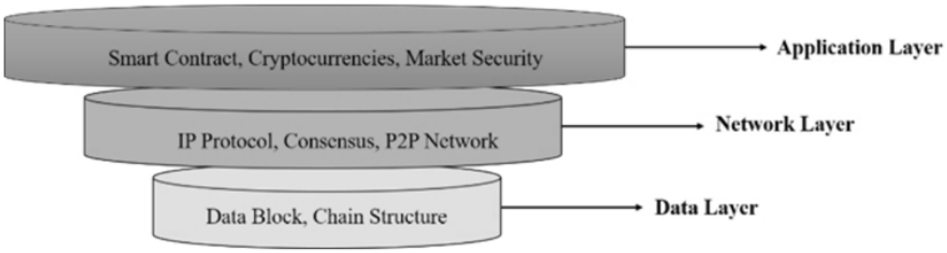


**Figure 1.2** – Architecture des Couches de Blockchain [3]

**Couche de données**

La couche de données fournit les moyens fonctionnels et procéduraux de créer des enregistrements de données (blocs) dans le grand livre de la blockchain. En utilisant les structures de données et les algorithmes tels que la signature numérique, l'arbre de Merkle, l'horodatage, le pointeur de hachage. Afin de garantir l'intégrité, la transparence et l'immutabilité des données [4].

**Couche de réseau**

La couche réseau fournit les mécanismes permettant la décentralisation du réseau via les protocoles de réseaux P2P. L'algorithme de consensus et sa mise en œuvre, qui servent à atteindre un accord distribué pour la validation des blocs, sont décrits dans cette couche afin d'assurer la résilience et la sécurité du système [4].

**Couche d'application**

La couche Application constitue la partie visible et utilisable de la blockchain, où se déploient les contrats intelligents, les cryptomonnaies et les mécanismes de sécurité des marchés. Elle permet aux utilisateurs et aux développeurs d'interagir avec la blockchain via des applications décentralisées (DApp), exploitant les propriétés de transparence, d'automatisation et de confiance offertes par les couches inférieures. Cette couche facilite l'adoption de la blockchain dans divers secteurs, allant des services financiers à la gestion d'actifs numériques.

### 1.2.4 Structure d'une blockchain

Le premier concept de blockchain (chaîne de blocs) remonte à 1991, lorsque les chercheurs Stuart Haber et W. Scott Stornetta ont publié un article décrivant un système

d'horodatage de documents numériques fondé sur une suite de blocs liés par des fonctions de hachage cryptographique, garantissant l'intégrité et l'ordre chronologique des enregistrements [5].

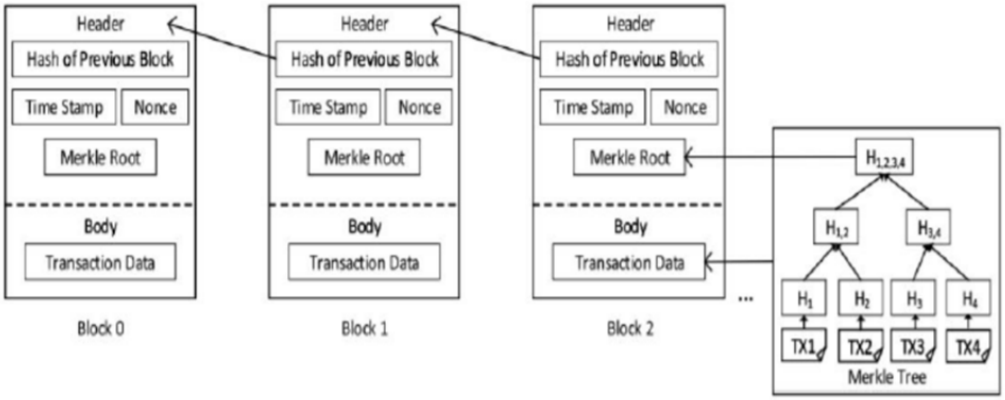


**FIGURE 1.3** – Structure de Blockchain [5]

La figure 1.3 illustre comment les transactions sont regroupées en blocs et enregistrées dans un registre immuable et transparent afin de construire une liste de blocs ordonnés chronologiquement tel que chaque bloc est construit à partir d'éléments de base et est divisé en deux parties :

**Header :**

- **Block Hash :** un identifiant unique et cryptographique qui est calculé à partir du contenu du block en appliquant une fonction de hachage (par exemple, SHA-256 dans Bitcoin).
- **Hash of Previous Block :** contient le hash cryptographique du bloc précédent, ce qui s'assure que le bloc précédent ne peut pas être modifié sans modifier l'entête du bloc actuel.
- **Time Stamp :** enregistre la date et l'heure de création du bloc.
- **Merkle Root :** le hash racine dérivé des transactions du bloc, obtenu via un arbre de Merkle. (Les transactions sont d'abord hachées individuellement. Puis, ces hash sont combinés par paires pour produire des hash intermédiaires. Enfin, les hash intermédiaires sont eux-mêmes combinés pour former un hash final appelé Merkle Root)

**Body :**

- **Transaction Data :** liste des transactions incluses dans ce bloc.

### 1.2.5 Le Trilemme de la Blockchain

La quête d'une blockchain parfaite repose sur l'équilibre dynamique entre trois impératifs indissociables : la sécurité, la décentralisation et la scalabilité. Ce trilemme, conceptualisé par Vitalik Buterin, fondateur d'Ethereum, expose l'impossibilité théorique d'optimiser simultanément ces trois dimensions sans compromis. Pourtant, les avancées technologiques récentes redéfinissent les frontières de ce dilemme, proposant des architectures hybrides et des mécanismes innovants pour concilier ces exigences apparemment contradictoires [6].

**La décentralisation : le cœur et la nature de la blockchain**

La décentralisation transcende la simple répartition géographique des nœuds. C'est une accessibilité technique et une indépendance vis-à-vis des entités centralisées. Afin d'assurer que le contrôle et la prise de décision sont répartis entre un vaste réseau de participants, empêchant toute entité d'exercer une influence excessive [6].

**La sécurité : une propriété essentielle**

Maintenir la résilience face aux attaques, préserver l'intégrité des données et garantir que le système puisse parvenir à un consensus même en présence d'acteurs malveillants [6] repose sur des mécanismes cryptographiques robustes (comme les fonctions de hachage SHA-256) et des protocoles de consensus éprouvés. À titre d'exemple, les mécanismes de consensus de type « proof of work» renforcent la sécurité en rendant la modification coûteuse de l'historique des transactions en termes de calcul, ce qui garantit une résistance aux attaques de type 51 % [7].

**La scalabilité : le principal défi**

La scalabilité mesure la capacité à augmenter le débit transactionnel (TPS), c'est-à-dire à traiter efficacement un nombre croissant de transactions, ce qui est essentiel pour une adoption généralisée et une utilisation pratique, sans augmentation proportionnelle des coûts [6]. Les solutions passent par une optimisation de la couche de base (Layer 1), notamment par l'utilisation du sharding ou de consensus améliorés, ainsi que le déploiement de protocoles secondaires(Layer 2), comme les rollups zk-SNARKs, qui compressent les données hors chaîne tout en conservant les garanties de sécurité, ou par des architectures modulaires innovantes qui séparent la couche de consensus de la couche d'exécution, permettant une spécialisation fonctionnelle et une meilleure scalabilité globale du système.

## 1.3 Les différents types de blockchain

La blockchain étant un registre distribué dans lequel chaque nœud du réseau possède une copie complète de l'ensemble des données. Certains nœuds peuvent avoir des rôles

spécifiques ou des caractéristiques particulières en fonction du type de la blockchain. En effet Il existe plusieurs types, chacune répondant à des besoins différents. Ces types sont présentés ci-dessous :

### 1.3.1 Blockchain publique

C'est un réseau ouvert à tous, dans lequel tous les nœuds de la chaîne sont accessibles. Un nœud dans la blockchain est un participant actif du réseau qui contribue à la validation, à la vérification et à la conservation des données de la blockchain [8].Toutefois, ce type de réseau présente une certaine lenteur en termes de publication des données sur la blockchain[ [9]. Comme exemples de blockchain publiques telles que : Bitcoin, Ethereum, Lisk,Litecoin ...

### 1.3.2 Blockchain privée

Ce sont des réseaux avec un nombre limité des nœuds nécessitent l'autorisation pour rejoindre le réseau [8], dont les données de la blockchain sont contrôlées par une organisation ou un groupe d'individus. Cette organisation ou ces individus peuvent modifier les règles de fonctionnement de la blockchain. Comme exemples de blockchain privée nous avons : Hyperledger Fabric, Hyperledger Sawtooth, Ripple, MultiChain...

### 1.3.3 Blockchain consortium

Également connue sous le nom de blockchain fédérée, elle désigne un système partiellement décentralisé . Le réseau est contrôlé par un groupe d'organisations, mais fonctionne dans différentes organisations. Il n'y a pas de consolidation du pouvoir de contrôle. Les nœuds ont besoin d'autorisations pour afficher, lire, écrire et vérifier des données de la blockchain [9]. Comme exemples nous citons : Quorum, R3 Corda...

### 1.3.4 Blockchain hybride

Fait référence à une combinaison de la blockchain privée et publique. La blockchain hybride est contrôlée par un consortium d'entreprises ou d'entités gouvernementales qui peuvent à la fois donner accès au public pour afficher ou ajouter des données et restreindre l'accès à ses membres [10]. Comme exemples de blockchain hybride nous avons : Dragonchain, XinFin, Balance...

## 1.4 Mécanismes de consensus

L'idée derrière les protocoles de consensus est de donner un droit à un nœud qui fournit suffisamment d'efforts et de puissance de calcul pour ajouter un nouveau bloc à

la blockchain. En général, ces algorithmes sont utilisés pour garantir La cohérence des données entre les différents nœuds du réseau, bien que de nombreuses recherches aient été menées pour proposer un protocole de consensus approprié afin d'accroître l'efficacité de la blockchain et d'encourager les exigences des applications [11]. Certains des protocoles de consensus les plus importants sont définis ici :

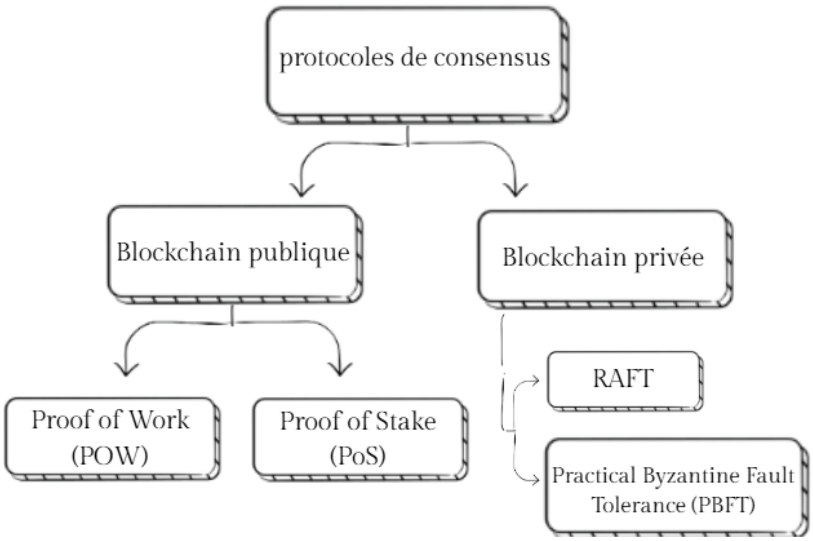


**FIGURE 1.4** – Mécanismes de consensus dans la blockchain

### 1.4.1 Proof of Work

Le Proof of Work (POW) est un protocole cryptographique et le mécanisme de consensus le plus célèbre utilisé dans les réseaux blockchain (Mining). L'objectif de ce protocole est de résoudre une énigme mathématique complexe en guise de récompense. Le mineur gagnant, généralement celui disposant de la plus grande puissance de calcul, obtient le droit d'ajouter un nouveau bloc à la séquence de blocs.

Les mineurs tentent de trouver une valeur de hachage qui respecte une condition prédéfinie par le réseau. En calculant de nombreuses fois des fonctions de hachage sur les données du bloc avec différentes valeurs (appelées "nonces") jusqu'à obtenir un hachage valide (comme le montre la figure 1.5). Le premier mineur à trouver une solution valide diffuse cette preuve à l'ensemble du réseau. Les autres nœuds doivent alors vérifier et valider cette solution avant que le bloc ne soit ajouté à la blockchain.

Bien que ce processus ne soit pas conceptuellement difficile, il nécessite d'importantes ressources informatiques et une consommation énergétique élevée, ce qui soulève des préoccupations environnementales. C'est pourquoi d'autres protocoles de consensus ont été étudiés [12].

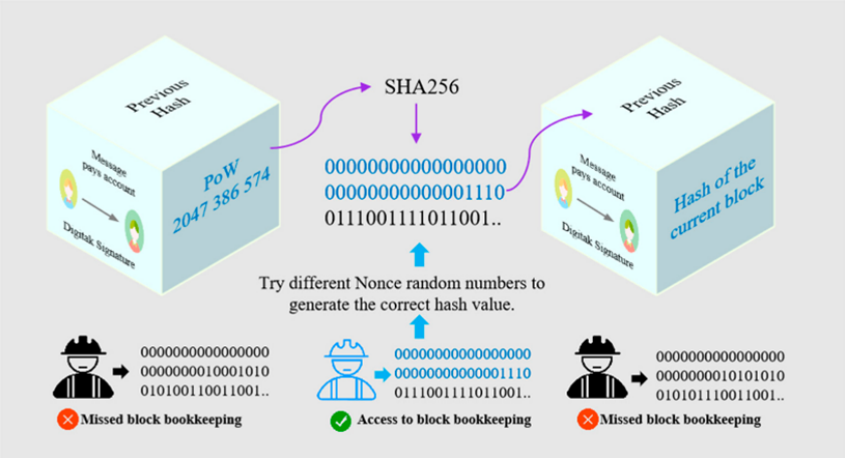


**FIGURE 1.5** – Algorithme proof of work [13]

### 1.4.2 Proof of Stake

Le Proof of Stake (PoS) ne nécessite pas une grande puissance de calcul. Il est considéré comme une alternative au PoW du point de vue de l'économie d'énergie. Dans ce protocole, il n'y a pas de mineurs. À la place, ce sont des validateurs qui en fonction de leur mise (stake) peuvent participer à la création d'un nouveau bloc [13] .

Le protocole sélectionne de façon pseudo aléatoire un validateur pour proposer le prochain bloc, , en s'appuyant principalement sur le mécanisme du coinage, c'est-à-dire l'ancienneté et la quantité des fonds détenus par chaque participant.

Cette méthode permet de réduire l'avantage des validateurs ayant uniquement un grand volume de tokens et d'éviter une centralisation excessive. Outre le coinage, d'autres critères, comme la taille du stake et la participation passée, peuvent également être pris en compte pour renforcer l'équité du processus de sélection [14]. Ensuite, si le bloc est approuvé, il collecte les frais de transaction dans le bloc (comme le montre la figure 1.6).

De cette manière, la consommation d'énergie, la latence et le débit sont réduits.

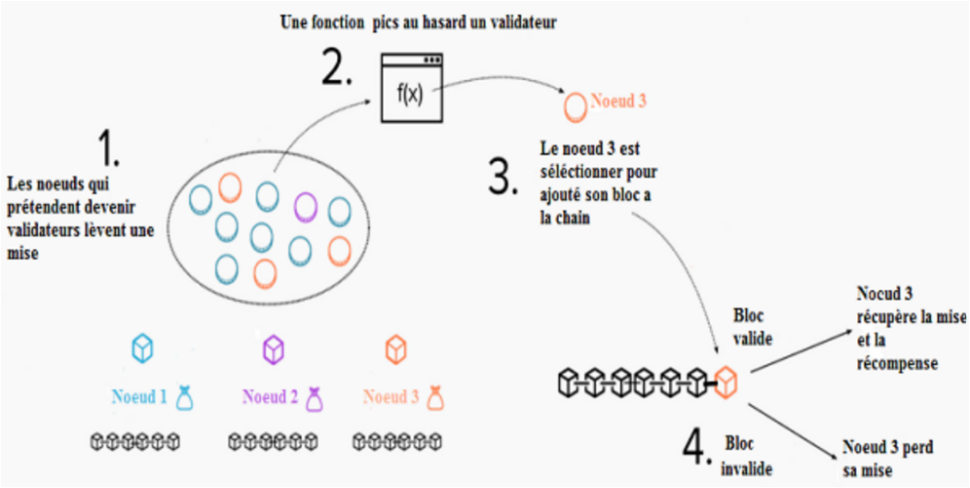


**FIGURE 1.6** – Algorithme proof of stake

### 1.4.3 Reliable, Replicated, Redundant, And Fault-Tolerant

Les blockchains privées ou permissionnées n'ont pas besoin de la complexité des algorithmes comme PoW ou PoS, car les nœuds sont connus et autorisés. L'objectif principal est la performance, la cohérence, et la tolérance aux pannes, pas la décentralisation anonyme. RAFT est donc utilisé comme algorithme de consensus rapide, fiable, simple et efficace.

Raft est un protocole de consensus tolérant aux fautes de type crash (Crash Fault Tolerance). Il s'agit d'une variante simplifiée de l'algorithme Paxos, dans lequel les nœuds du réseau peuvent se trouver dans l'un des trois états suivants : leader, follower ou candidat [15].

Dans ce protocole, un seul nœud joue le rôle de leader. Ce dernier reçoit les transactions envoyées par les clients, les valide, les indexe et les organise dans l'ordre chronologique. Les transactions validées sont ensuite regroupées en blocs, et le transmet aux followers. Ces followers doivent accuser réception de la validité des blocs et les répliquer localement, c'est-à-dire reproduire fidèlement les données qu'ils contiennent.

Lorsqu'aucune requête n'est reçue pendant un certain temps cela indique que le leader est probablement tombé en panne. À ce moment-là, une élection est déclenchée pour désigner un nouveau leader. Un nœud entre alors en état candidat, vote pour lui-même, puis envoie une demande de vote aux autres nœuds du réseau. S'il obtient la majorité, il est élu leader et annonce son élection aux autres.

Cependant, il se peut qu'un candidat reçoive un message d'un autre nœud déclarant avoir été élu. Dans ce cas, il compare l'index de terme (représente le temps logique) reçu avec le sien. Si l'index de l'autre est plus élevé, il reconnaît ce nœud comme le nouveau leader. Si son propre terme est plus récent, il rejette la revendication et poursuit le processus de vote. Une valeur de terme plus élevée signifie que le nœud est au courant des derniers événements du système.

Si aucun candidat n'obtient la majorité, une nouvelle élection est relancée après un court délai aléatoire. Le nœud ayant le plus petit délai redémarre l'élection [16].

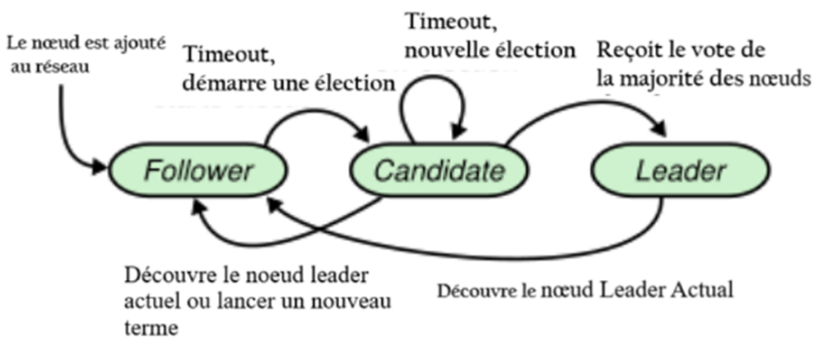


**Figure 1.7** – Transitions d'état des nœuds dans le protocole Raft[17]

### 1.4.4 Practical Byzantine Fault Tolerance

La tolérance aux fautes byzantines (BFT) est une situation dans laquelle un système distribué à continuer de fonctionner correctement même si certains de ses nœuds se comportent de manière malveillante ou défaillante.

Le Practical Byzantine Fault Tolerance (PBFT) est un algorithme qui implémente cette tolérance de façon efficace, capable de résister à des défaillances byzantines, à condition que le nombre de nœuds malveillants ne dépasse pas le tiers du total [4].

Afin de prévenir toute falsification ou confusion, chaque nœud signe les messages avec sa clé secrète. De plus, chaque message est accompagné d'un code d'authentification, et est compressé via une fonction de hachage avant d'être envoyé. Chaque nœud communique avec tous les autres nœuds du système. Les signatures permettent aux nœuds de s'identifier mutuellement et de vérifier l'intégrité des messages reçus (c'est-à-dire détecter s'ils ont été modifiés pendant la transmission) [17].

Dans un réseau PBFT, les nœuds sont organisés hiérarchiquement : un nœud est désigné comme leader, les autres comme des backups. Le rôle de chaque nœud change à chaque nouveau tour selon un mécanisme de rotation (round robin).

Le protocole fonctionne en trois phases principales :

- **Preprepared :** le leader publie une valeur (transaction ou bloc) à enregistrer dans la blockchain en la diffusant aux backups.
- **Prepared :** les nœuds backups reçoivent cette valeur, vérifient sa validité et réenvoient un message de préparation aux autres nœuds.
- **Commit :** si un quorum de 2f + 1 messages de préparation (avec f étant le nombre maximal de nœuds fautifs tolérés) est atteint, alors les nœuds s'engagent à valider la valeur et procèdent à son enregistrement définitif dans la blockchain.

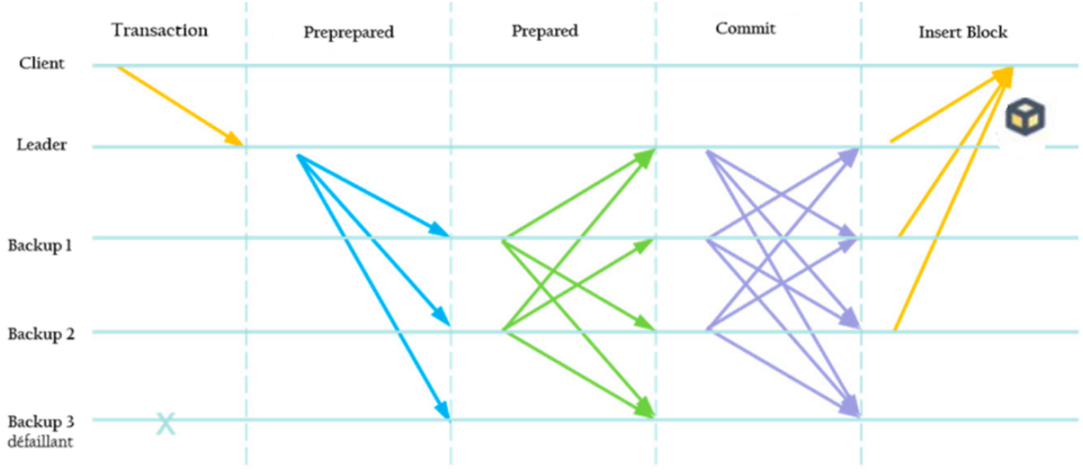


**Figure 1.8** – Processus de Protocole PBFT

Cette mécanique garantit que la valeur validée est correcte et acceptée par la majorité honnête du réseau. Cependant, le protocole PBFT nécessite plusieurs échanges de messages à chaque tour, ce qui implique un coût de communication élevé. Pour cette raison, il n'est pas considéré comme une solution scalable dans les environnements à très grand nombre de participants, comme les blockchains publiques. [18]

## 1.5 Les solutions blockchain

La technologie blockchain a été implémentée par plusieurs groupes afin de répondre à différents besoins. Ces implémentations se différencient de plusieurs manières, notamment par leur méthode de consensus, l'existence d'une cryptomonnaie associée et l'étendue des fonctionnalités de leurs contrats intelligents.

Cette section se concentrera sur trois architectures blockchain les plus connues, soit le réseau Bitcoin (l'origine de la technologie blockchain), l'architecture Ethereum (une plateforme extensible avec plusieurs réseaux publics et la possibilité de créer des réseaux privés configurables),Hyperledger Fabric (une architecture modulaire utilisée dans des applications privées).

### 1.5.1 Bitcoin

La première blockchain publiquement disponible est le Bitcoin. Publiée en 2008, le White paper Bitcoin décrit des méthodes par lesquelles un réseau d'échange décentralisé pourrait être sécurisé par un registre maintenu par des nœuds publics. Dans un tel système, la validité d'une transaction est décidée par le vote de la majorité du réseau par un consensus de PoW [7].

Chaque utilisateur peut commencer une transaction en utilisant une signature numérique [19]. Toute transaction dans la blockchain nécessite un portefeuille numérique, un artefact qui permet aux utilisateurs d'effectuer des transactions électroniques. Les portefeuilles numériques sont construits à partir de matériel cryptographique à clé publique pour authentifier l'utilisateur et ses transactions. L'utilisateur signe ses transactions avec une clé privée et toute autre entité peut vérifier l'authenticité de cette transaction à l'aide de la clé publique de l'utilisateur [20], qui est mathématiquement associée à la clé privée en utilisant ECDSA et les fonctions de hachage (comme le montre la figure 1.9) [21].


**FIGURE 1.9** – Relation entre clé privée et clé publique et adresse bitcoin

Grâce au protocole Gossip, toutes les transactions sont envoyées à tous les nœuds [22]. Afin d'inciter les nœuds mineurs (qui maintiennent le registre et sont essentiels au traitement des transactions), le réseau Bitcoin récompense le mineur qui réussit à valider un bloc avec une quantité de bitcoins (BTC). Cette récompense suit un protocole qui réduit progressivement le nombre de BTC attribués à intervalles réguliers. Ce mécanisme permet de contrôler de manière décentralisée la quantité maximale de BTC en circulation. En divisant par deux la récompense de minage à chaque halving [2], le réseau Bitcoin garantit une inflation maîtrisée de sa cryptomonnaie [7]. Le schéma ci-dessus illustre le processus de validation et d'intégration d'une transaction au sein d'une blockchain Bitcoin reposant sur le mécanisme de consensus PoW.

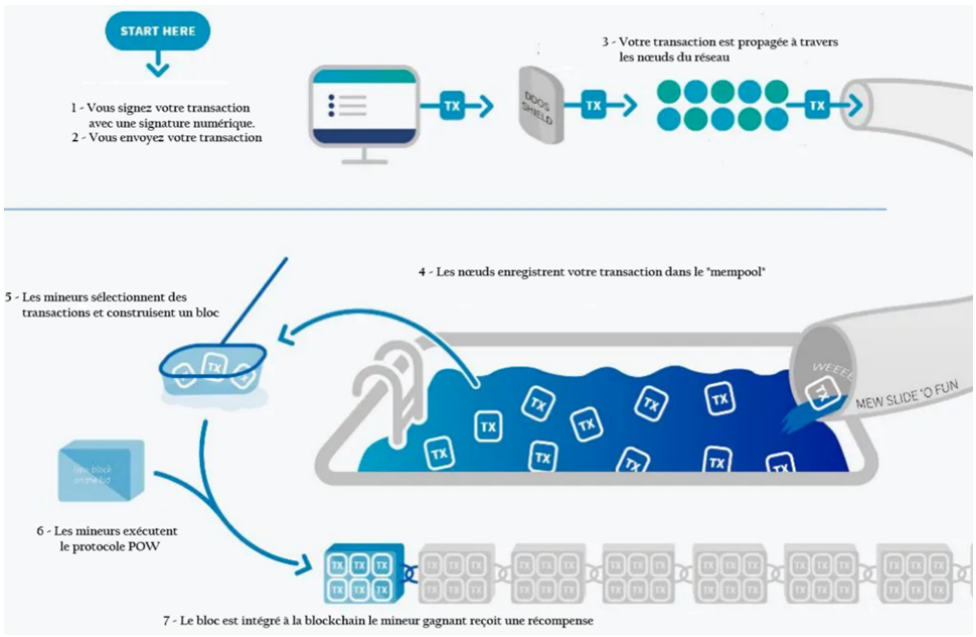


**FIGURE 1.10** – Mécanisme de Validation des Transactions dans un Réseau Bitcoin

### 1.5.2 Ethereum

Ethereum est un réseau blockchain qui a été lancé en 2015 [23]. Il représente l'évolution naturelle de l'écosystème du domaine de la blockchain. Alors que Bitcoin permet d'exécuter certaines validations et conditions sur des transactions via le Bitcoin Script,

2. **halving :** Un événement programmé environ tous les 4 ans pour réduire de moitié la récompense accordée aux mineurs lors de la validation d'un nouveau bloc sur la blockchain dans le réseau Bitcoin [7]

l'Ethereum Virtual Machine (EVM) permet l'automatisation de l'exécution d'une transaction et de la faire de manière autonome grâce aux contrats intelligents. Cette évolution technologique a permis la création des d'Apps, ce qui ouvre la voie à une nouvelle génération de services numériques plus transparents, sécurisés et résistants à la censure.

les contrats intelligents sont des programmes écrits en langage Turing complet, permettant de créer des opérations complexes. Une fois le code écrit, il est déployé sur la blockchain, où il obtient une adresse unique. À partir de ce moment-là, il devient immuable (on ne peut plus le modifier). Bien sûr, avec la possibilité d'insérer des boucles dans le code vient le risque d'exécuter des opérations de longue durée (voire infinies). Puisque la validation d'une transaction requiert l'exécution du contrat intelligent (comme illustre la figure 1.11). Cela peut entraîner un ralentissement du réseau. Une solution implémentée dans Ethereum pour résoudre ce problème est l'utilisation du Gas [24].

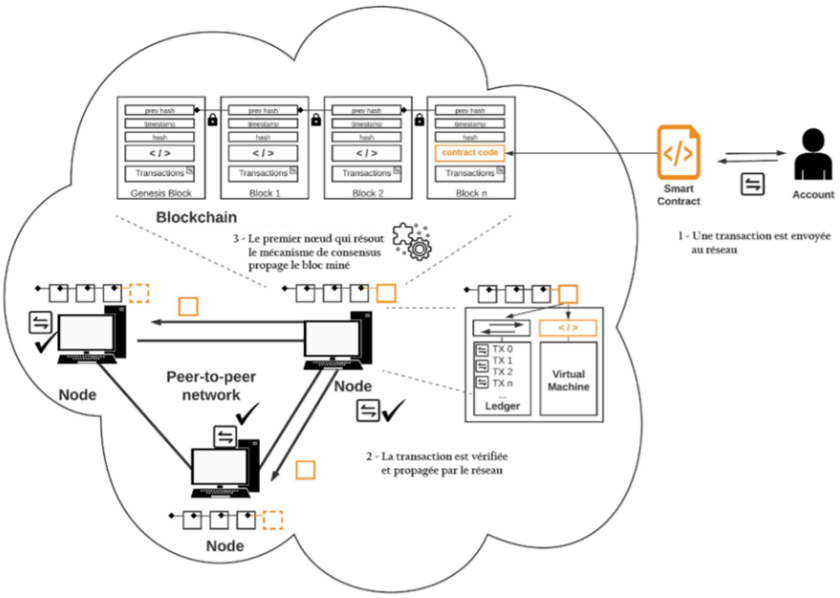


**Figure 1.11** – Interaction entre l'utilisateur et le réseau Ethereum via le contrat intelligent [26]

Le Gas est un coût associé à toute opération effectuée sur Ethereum. Lorsqu'un utilisateur veut exécuter une transaction, il doit fournir une quantité suffisante de Gas qui sera consommée par le réseau et distribuée comme récompense aux nœuds mineurs. Si la transaction manque de Gas (c'est-à-dire si l'opération coûte plus que ce qui a été fourni), elle est marquée comme incomplète et le Gas consommé n'est pas retourné à l'utilisateur. Cela force un degré de déterminisme dans l'exécution des transactions, ce qui permet d'éviter les opérations interminables [25].

Le Gas permet aussi de se protéger contre des opérations malicieuses. Puisqu'il est possible de passer des appels d'un contrat intelligent à l'autre,il existe plusieurs vecteurs d'attaque, tels que la réentrance, qui permettent d'exploiter le système. L'utilisation d'une quantité limitée de Gas force les appels malveillants à échouer, empêchant ainsi l'attaque. [25].

Ethereum 2.0 a migré vers PoS une mise à jour majeure qui a amélioré la sécurité, la scalabilité et la consommation d'énergie du réseau (permet une validation plus rapide des transactions) [26].

### 1.5.3 Hyperledger Fabric

Hyperledger Fabric est une plateforme de blockchain open source dotée d'une architecture modulaire et configurable conçue pour offrir des niveaux élevés de confidentialité, de résilience, de flexibilité et de scalabilité [27]. Elle a été lancée dans sa version 1.0 en 2017 [28]. Contrairement à Bitcoin et Ethereum, Hyperledger Fabric est conçue spécifiquement pour des solutions destinées au secteur privé. Bien qu'il soit possible de créer un réseau Ethereum distinct du réseau public, tous les nœuds sont par défaut égaux. À l'inverse, Hyperledger Fabric permet de créer des nœuds de types différents dont les membres sont connus et validés par une autorité d'affiliation (Membership Service Provider) [28], appartenant à divers organismes. Cette distinction permet d'assigner des identités et des rôles spécifiques à chaque nœud (voir la figure 1.12). De plus, Hyperledger Fabric utilise des nœuds d'ordonnancement (ordering service), responsables de la cohérence et de la synchronisation des données entre les différents organismes. Cette architecture permet également aux architectes de créer des channels, c'est-à-dire des structures de données réservées aux utilisateurs autorisés à y accéder. Sur ces channels réside le chaincode, l'architecture de contrats intelligents qui gère la manipulation et le traitement des données, et permet aux applications d'interagir avec les informations du réseau [25].

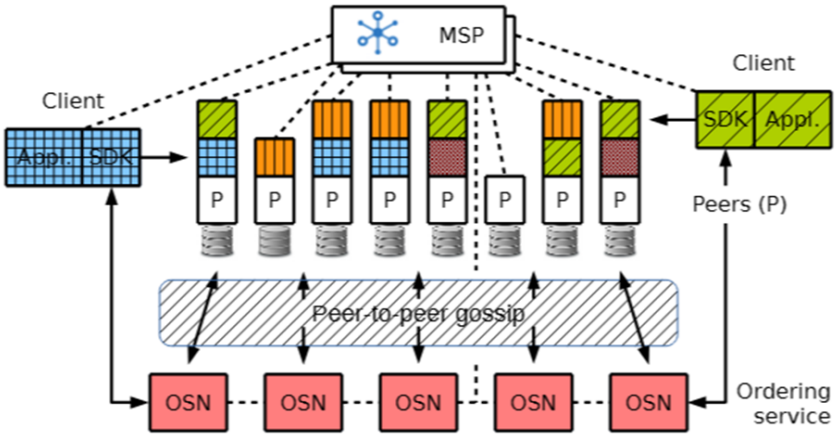


**FIGURE 1.12** – Intégration du MSP dans l'architecture Hyperledger Fabric [31]

#### 1.5.3.1 Les nœuds

Hyperledger Fabric possède plusieurs types de nœuds afin de permettre la création et la gestion d'un consortium. Puisqu'il s'agit d'une solution de blockchain privée, une confiance implicite existe entre les opérateurs de nœuds, tels que chaque nœud dispose d'une identité unique, attestée par un certificat numérique délivré par une autorité de

certification spécifique (CA). Afin d'assurer le bon fonctionnement du réseau, deux principales catégories de nœuds sont distinguées.

**o Peer :**Les peers sont des composants fondamentaux dans le réseau Hyperledger Fabric. Ils hébergent et maintiennent une copie complète ou partielle du ledger (registre distribué) et exécutent les chaincodes afin d'assurer la résilience et la distribution des données. Ils peuvent également héberger des SDK et des API permettant aux utilisateurs d'interagir avec le réseau (la figure 1.13). chaque peer peut jouer plusieurs rôles au sein du réseau tels que :

- **Committer Peer :**Chaque nœud peer dans un canal est un Committer Peer. Il agit comme un nœud comptable, maintenant et stockant les données dans le ledger. Le ledger se compose d'une blockchain et d'un world state, qui est une base de données qui stocke l'état actuel de tous les objets ou actifs du réseau sous forme de paires clé-valeur (key-value pairs) afin d'accéder rapidement aux données. Ces états peuvent être accessibles via l'invocation d'un chaincode (voir la figure 1.13) [29].

- **Leader Peer :**Dans un système où une organisation a plusieurs peers dans un canal, un peer leader est chargé de distribuer les blocs du service d'ordonnancement aux autres peers Committer de l'organisation via un protocole de diffusion P2P Gossip. Il existe deux types de sélection de leadership : statique et dynamique.

  Dans la sélection statique, aucun ou plusieurs peers peuvent être désignés comme leaders, tandis que dans la sélection dynamique, un leader est élu par l'ensemble des peers. En cas de défaillance les pairs restants éliront un nouveau leader. Ce qui renforce la résilience et la scalabilité dans les réseaux.

- **Endorser Peer :**Chaque nœud peer peut être un Endorser Peer s'il a un contrat intelligent installé, et que ce contrat intelligent est utilisé par une application cliente pour générer une réponse de transaction signée numériquement.

- **Anchor Peer :**Dans un système où un peer doit communiquer avec un peer d'une autre organisation, il peut utiliser l'un des Anchor Peers définis dans la configuration du canal pour cette organisation. Cela peut aider dans divers scénarios de communication interorganisationnelle.

**o Ordering Service :** Contrairement aux blockchains sans permission (comme Bitcoin) qui parviennent à un consensus selon un processus probabiliste, le réseau Fabric utilise des nœuds Orderers qui gèrent l'enchaînement des transactions. cet ensemble de nœuds est chargé de trier les transactions et de les assembler en blocs pour les distribuer aux nœuds Leader. Des protocoles de consensus comme Kaft, Raft et PBFT seront implémentés pour garantir la cohérence et la fiabilité de l'ordonnancement (voir la figure 1.13) [30] [31].

#### 1.5.3.2 Les channels

Un channel est une structure propre à Hyperledger Fabric. il est considéré comme un sous-réseau privé regroupant un nombre limité de nœuds, où le partage d'informations est strictement contrôlé. Chaque channel possède son propre registre, ses propres smart contracts et ses propres politiques de gouvernance distinctes. Un nœud peut participer à plusieurs canaux (la figure 1.13) et transmettre des informations et des données de manière privée [32].

La communication entre les différents nœuds d'un channel repose principalement sur des échanges de messages via des protocoles réseau sécurisés comme gRPC, basé sur HTTP/2 et sécurisé par TLS pour garantir la confidentialité et l'intégrité des échanges [33].

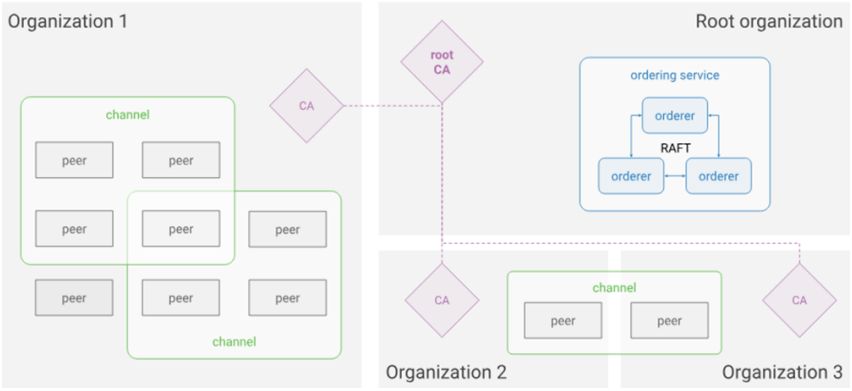


**FIGURE 1.13** – Architecture d'un Réseau Hyperledger Fabric : Répartition des Peers, CA et Service d'Ordonnancement autour des Channels [36]

#### 1.5.3.3 Le chaincode

Dans Hyperledger Fabric, le chaincode est l'équivalent des smart contracts.Il s'agit d'un programme qui définit la logique métier à exécuter sur le réseau blockchain. Son rôle est de valider, traiter et enregistrer les transactions selon des règles prédéfinies. Chaque chaincode fonctionne dans un environnement isolé pour garantir la sécurité et l'indépendance des différents contrats [25].

L'endorsement policy pour un contrat intelligent identifie les organisations dont les peers doivent signer numériquement une transaction générée, avant que celle-ci ne puisse être acceptée et inscrite sur le registre de la blockchain [31].
Exemple : un reseau avec 3 organisations : Org1, Org2, Org3

> Endorsement policy :"AND('Org1.member', 'Org2.member')" cela signifie que les pairs de l'organisation 1 ET ceux de l'organisation 2 doivent signer chaque transaction issue du smart contract.

En plus de la structure en channels, Il existe des données privées qui sont conservées en dehors de la chaîne appelées collections. Ces données ne sont accessibles qu'à un sous-

ensemble d'organisations du channel défini lors de l'installation du chaincode. Seul le hash des données est enregistré dans la blockchain(la figure 1.14) [34].

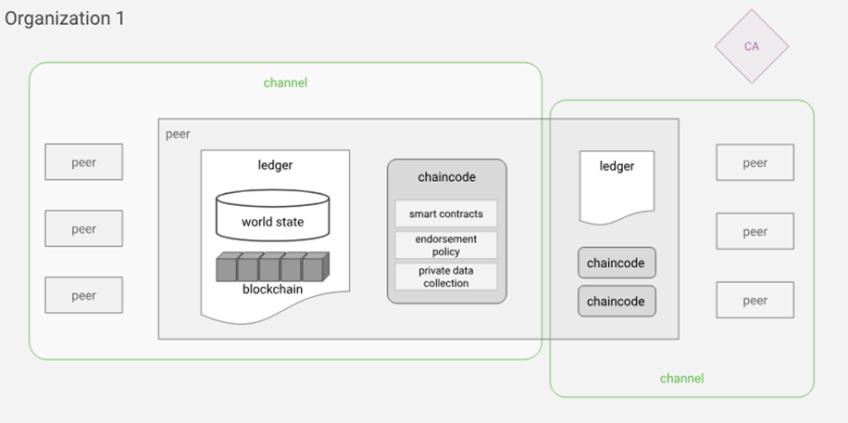


**FIGURE 1.14** – Organisation interne d'un channel Hyperledger Fabric : structuration des peers,du ledger et du chaincode [36]

#### 1.5.3.4 Les applications

Alors qu'il est possible d'exécuter des requêtes directement dans le terminal d'un nœud, Hyperledger Fabric propose des SDK et des API qui permettent d'exécuter des transactions à partir d'applications externes. Pour accéder à un contrat intelligent résidant sur un nœud, l'application doit fournir un profil de connexion contenant les informations du nœud ainsi qu'un Gateway pour accéder à un contrat intelligent résidant sur un nœud. Le Gateway donne accès aux fonctions permettant d'effectuer des opérations de lecture et d'écriture sur le contrat intelligent ciblé [25].

Dans un réseau Hyperledger Fabric, chaque application doit être identifiée de manière sécurisée pour pouvoir interagir avec le réseau blockchain. Cette identification est assurée par le serveur Hyperledger Fabric Certificate Authority (Fabric-CA). Le schéma ci-dessous illustre cette architecture :

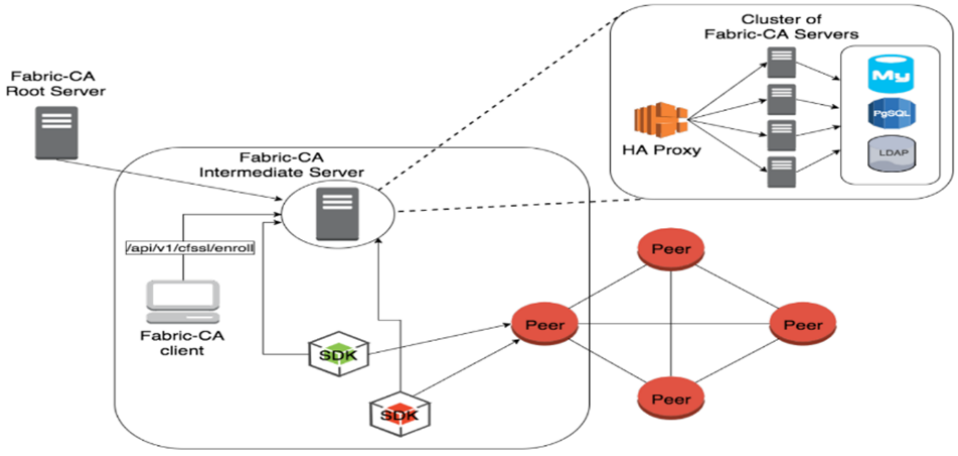


**FIGURE 1.15** – Hyperledger Fabric CA [37]

D'après la figure 1.15, le serveur Hyperledger Fabric CA peut être sollicité de deux manières : soit via le client dédié à Hyperledger Fabric CA, soit à travers l'un des SDK Fabric. Le client envoie le trafic vers un point de terminaison proxy à haute disponibilité, qui répartit la charge entre les membres du cluster fabricca server afin d'obtenir un certificat d'identité (X.509) permettant de s'authentifier sur le réseau blockchain. Tous les serveurs Hyperledger Fabric CA d'un même cluster partagent une base de données commune pour la gestion des identités et des certificats. Si le protocole LDAP est activé, les informations d'identité sont stockées dans LDAP (un système externe de gestion des identités) plutôt que dans la base de données, ce qui facilite l'intégration avec les infrastructures d'entreprise existantes.

#### 1.5.3.5 Flux de transactions

Après avoir présenté les concepts clés liés à Hyperledger Fabric, nous allons maintenant examiner son flux de transactions. Comme l'illustre la figure 1.16, le processus commence par la génération d'une demande de transaction et se termine par son enregistrement dans le registre distribué.

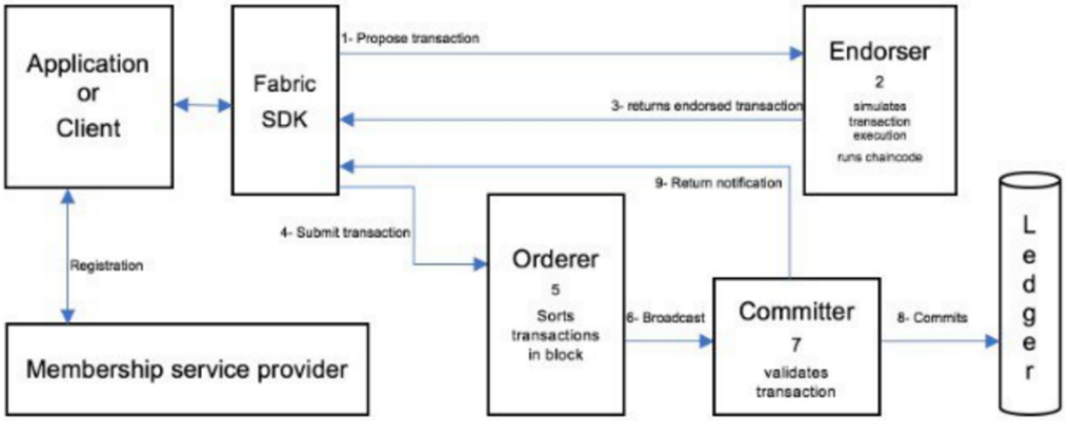


**Figure 1.16** – Flux de transaction [38]

- **Phase d'exécution :** Une application cliente crée et envoie une proposition de transaction signée aux endosseurs, conformément à la politique d'endossement correspondante, afin d'invoquer une fonction de chaincode pour interagir avec le registre de la blockchain. Dès que les endosseurs ont exécuté le smart contract chaincode avec succès, une réponse signée est renvoyée au client. La transaction est ensuite assemblée et signée avec les informations d'identification obtenues auprès d'un fournisseur de services d'adhésion (MSP) [35].
- **Phase d'ordonnancement :** Cette phase intervient après qu'un client a collecté suffisamment d'endossements pour une proposition. Il assemble la transaction et la soumet au service d'ordonnancement. Ce dernier établit un ordre total sur toutes les transactions soumises par canal, regroupe plusieurs transactions en blocs et garantit

ainsi un consensus sur leur ordre, même en présence d'ordonnanceurs défectueux. Enfin, les blocs sont transférés aux pairs concernés [35].

- **Phase de validation :** Les blocs sont livrés aux peer concernés, soit par le service d'ordonnancement, soit via le protocole de gossip. Chaque peer doit ensuite évaluer la politique d'endossement, vérifier les conflits de lecture-écriture, et mettre à jour le registre [35].

## 1.6 Le paiement électronique en Algérie

Aujourd'hui, le paiement électronique est considéré comme un défi stratégique, offrant aux entreprises et aux individus la possibilité d'acheter et de vendre des biens ou des services par voie numérique, sans avoir besoin d'échanger de l'argent liquide ou des chèques papier, ce qui facilite grandement les échanges commerciaux.

### 1.6.1 État de lieux de l'e-paiement en Algérie

L'Algérie, en tant que pays en voie de développement, en est à ses débuts en matière d'économie de marché. La pénétration des Technologies de l'Information et de la Communication (TIC) constitue un levier stratégique pour accélérer cette transition économique [36]. Ces dernières années, le pays a initié plusieurs réformes visant à moderniser le secteur financier et à encourager l'adoption des paiements électroniques. En effet, différents modes de paiement électronique sont désormais disponibles : la carte interbancaire CIB, la carte Edahabia d'Algérie Poste, les cartes bancaires internationales telles que Visa et Mastercard, les services proposés par les banques, le système de paiement mobile RMI ainsi que les services proposés par les banques .

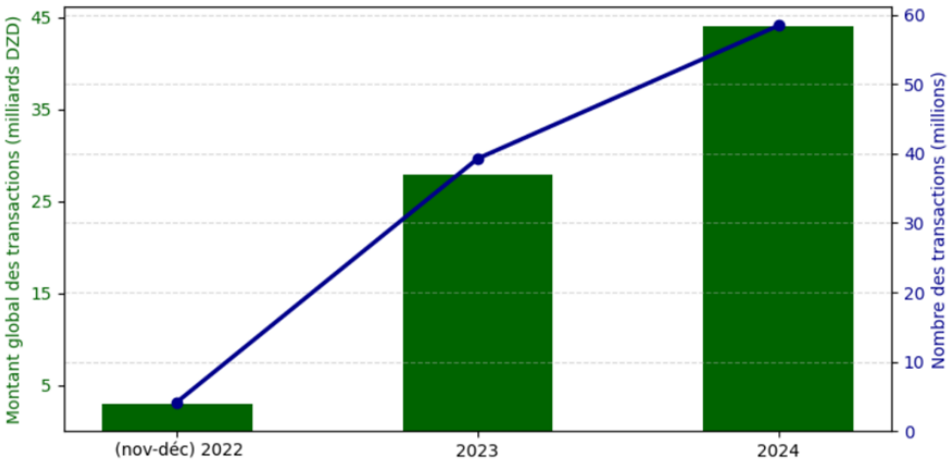


**Figure 1.17** – Évolution de paiement par mobile en Algérie

D'après le GIE Monétique, la figure 1.17 illustre l'évolution des activités de paiement par mobile en Algérie, en volume et en valeur. Ce qui encourage les entreprises et les commer-

çants à intégrer ce mode de paiement dans leurs activités, afin de répondre aux attentes d'une clientèle de plus en plus tournée vers le digital.

L'Algérie poursuit toujours ses efforts pour moderniser ses systèmes de paiement. À titre d'exemple, la société Naftal, spécialisée dans la distribution de produits pétroliers, a lancé la Naftal Card dans ses stations-service en 2023. Cette carte permet de supprimer l'utilisation des bons d'essence et de crédit, tout en offrant aux gestionnaires de flottes de véhicules la possibilité de suivre en temps réel leur consommation de carburant. [37]

### 1.6.2 Impacts opérationnels sur les entreprises

L'adoption des systèmes de paiement électronique offre de multiples avantages potentiels pour les entreprises :

- **Amélioration de l'efficacité et réduction des coûts :** Gain de temps, réduction de la charge administrative, optimisation de la gestion des flux financiers.
- **Expansion de la clientèle et accès à de nouveaux marchés :** Le paiement électronique permet d'accepter les paiements à distance, d'élargir la clientèle et de soutenir la croissance du commerce en ligne. Aujourd'hui, 625 Web marchands sont adhérents au système de paiement sur internet par carte interbancaire [38].
- **Meilleure gestion de la Trésorerie et Traçabilité :** L'utilisation des TPE, de BaridiMob et d'autres solutions de paiement électronique réduit les risques de vol, de perte ou de fraude liés à la manipulation du cash, tout en assurant la traçabilité de chaque opération. Selon Algérie Poste, l'application BaridiMob a traité plus de 13 millions d'opérations durant les cinq premiers mois de l'année 2023 [39].
- **Ouverture à l'international et paiements transfrontaliers :** L'e-paiement facilite les transactions avec des partenaires et des clients étrangers grâce à la compatibilité avec les cartes internationales et les plateformes de paiement en ligne. De nombreuses entreprises algériennes acceptent désormais Visa et Mastercard comme solutions de paiement, et récemment, Djezzy a introduit PayPal parmi les moyens de paiement dans ses activités [40].

## 1.7 L'impact de la technologie de blockchain sur l'entreprise

Au cours de la dernière décennie, la technologie de la blockchain s'impose comme une innovation majeure capable de bouleverser les modes de fonctionnement traditionnels des entreprises grâce à son architecture décentralisée qui offre des solutions mieux adaptées aux enjeux actuels en matière de sécurité . L'intégration progressive de cette technologie présente plusieurs avantages :

- **Sécurité renforcée des transactions :** La blockchain utilise des mécanismes cryptographiques avancés (hashage, signatures numériques) qui protègent contre la falsification, la fraude et les attaques malveillantes.
- **Transparence et traçabilité :** Chaque transaction est enregistrée de façon permanente et immuable sur le registre, rendant toutes les opérations vérifiables en temps réel par toutes les parties prenantes.
- **Automatisation avec les smart contracts :** Les contrats intelligents permettent d'automatiser l'exécution des transactions selon des conditions préalablement définies, ce qui réduit les erreurs humaines et améliore l'efficacité.
- **Résilience et disponibilité accrue :** La nature décentralisée de la blockchain la rend moins vulnérable aux pannes, attaques ou défaillances d'un seul point (single point of failure), assurant une haute disponibilité du service .
- **Confidentialité configurable :** Certains protocoles blockchain (comme Hyperledger Fabric) permettent de configurer différents niveaux de confidentialité et d'accès aux données des transactions selon les besoins des participants.

## 1.8 Conclusion

Les meilleures pratiques pour exploiter la technologie de la blockchain consistent à l'intégrer efficacement aux processus de l'entreprise, dans le but d'améliorer leurs performances tout en garantissant un niveau optimal de sécurité, de transparence des données dans un monde où les risques liés au piratage de données sont omniprésents. Dans ce chapitre, nous présenterons une vue globale de la technologie blockchain, avant d'aborder les différentes architectures des solutions existantes. Le paiement électronique aussi occupe également une place importante dans notre étude, notamment dans le contexte algérien, afin de comprendre comment ces deux technologies impactent les entreprises. La mise en œuvre de ces technologies ouvre la voie à des opportunités significatives d'optimisation, contribuant à la modernisation de l'architecture des systèmes des entreprises, tout en renforçant leur résilience, leur efficacité et leur capacité d'adaptation face aux enjeux numériques actuels ce que nous examinerons dans le chapitre suivant

# Chapitre 2

# Étude de la Situation Existante et Analyse des Besoins

## 2.1 Introduction

Naftal joue un rôle essentiel dans le secteur énergétique algérien, en assurant la distribution et la commercialisation des produits pétroliers à travers le pays. Le paiement électronique est devenu indispensable dans les transactions financières du marché, s'inscrivant dans une démarche de digitalisation tout en garantissant la sécurité et la traçabilité des échanges. Ce chapitre se concentre sur l'étude approfondie de l'architecture du système de gestion des ventes dans les stations-service, en mettant en lumière les travaux connexes visant à moderniser la situation existante. Il pose également les bases du développement d'une solution adaptée aux défis identifiés à travers une analyse des besoins.

## 2.2 Étude de la situation existante

Naftal est une société par actions (SPA) fondée en 1982, filiale à 100 % du groupe Sonatrach, et rattachée à l'activité de commercialisation.

Sa mission principale consiste en la distribution et la commercialisation des produits pétroliers et dérivés sur le marché national. Elle fournit près de 13,3 millions de tonnes de produits pétroliers par an, un chiffre appelé à augmenter en raison d'une demande croissante. Naftal a également adopté une nouvelle vision stratégique à moyen terme, orientée client, accompagnée d'un plan de mise en œuvre. Elle distribue et commercialise les carburants terrestres, aviation et marine, les GPL (gaz de pétrole liquéfié), les lubrifiants, les bitumes, les pneumatiques ainsi que divers produits spéciaux [41].

Le projet présenté dans ce mémoire a été initié par la Direction Centrale des Systèmes d'Information (DCSI), dont sa mission principale est d'organiser la migration des systèmes informatiques existants vers de nouvelles plateformes plus performantes, ainsi que

de mettre en place des progiciels et autres outils modernes (ERP) pour l'analyse des bases de données de gestion et l'aide à la décision.

Naftal dispose d'un réseau de 2 248 stations-service, dont 378 en Gestion Directe (GD), 309 en Gestion Libre (GL), 1 383 Points de Vente Agréés (PVA) et 178 revendeurs ordinaires. Elle compte également 29 stations-service autoroutières situées sur l'axe Est-Ouest [41].

Dans le cadre de la gestion efficace des approvisionnements et des ventes en station-service, il est essentiel de mettre en place une architecture réseau sécurisée et fluide. Nous allons procéder à une analyse de la situation existante. On commence par :

### 2.2.1 Le processus d'achat dans les stations-service

Chaque station surveille en temps réel ou de manière périodique le niveau de ses stocks (carburant, lubrifiants, autres produits). Dès qu'un seuil minimum de stock est atteint, le responsable de la station génère une demande d'approvisionnement à destination du centre régional.

Le centre régional d'approvisionnement vérifie la disponibilité des produits, puis prépare un Bon de Livraison (BL) pour chaque transfert de marchandise. À la réception de la livraison, la station effectue un contrôle quantitatif et qualitatif des produits reçus. La réception est ensuite validée via la plateforme web Mynaftal, ce qui permet la mise à jour automatique des stocks après validation, un Bon de Livraison Facturé (BLF) est généré pour permettre le paiement. Si certains produits livrés s'avèrent non conformes ou endommagés, un Bon de Livraison de Retour (BLR) est émis, ce qui permet de déduire la somme de la prochaine facture.

Les produits sont stockés dans le système SGBD en trois modes : par piste (zone de transit temporaire avant transfert vers le magasin), par baie ou directement en magasin.

### 2.2.2 Gestion des ventes et paiement dans les stations-service

Après le stockage, les produits sont proposés à la clientèle. Chaque transaction peut être réalisée via l'un des quatre modes de paiement : en espèces, par carte CIB, via Naftal Carte ou par TPE avec la carte Edahabia. Il existe également des tickets carburant (DOSN, MDN, NAFT, administratifs classiques) utilisables lors du paiement, chacun offrant un pourcentage de remise spécifique.

Après chaque vente, un bon de vente est automatiquement généré, assurant la traçabilité de chaque opération réalisée au sein de la station-service avant leur saisie dans le système SGBD traditionnel centralisé. Ces bons de vente sont préalablement regroupés par brigade en fonction de la plage horaire concernée :

- brigade 1 : 6h00 – 12h00
- brigade 2 : 14h00 – 18h00
- brigade 3 : 18h00 – 6h00

Ce regroupement par brigade avant l'enregistrement dans le système permet d'optimiser le traitement des ventes, tout en facilitant le contrôle des recettes et la gestion comptable. Ce dispositif contribue également à garantir une organisation efficace et une disponibilité continue du service pour la clientèle, quels que soient l'horaire ou le mode de paiement utilisé.

L'ensemble de ce processus repose sur une architecture centralisée, qui assure une gestion unifiée, une synchronisation en temps réel entre les stations-service et le système central de Naftal(la figure 2.1).

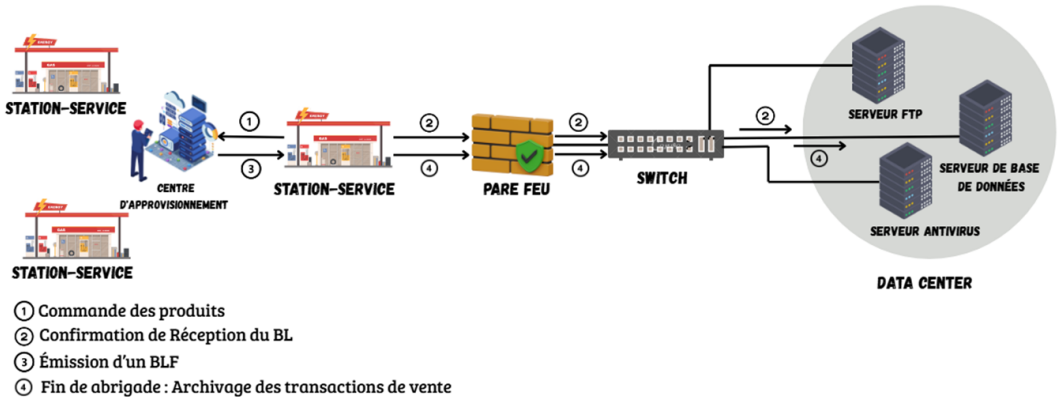


**FIGURE 2.1** – Architecture du Système d'Approvisionnement et de Gestion des Transactions en Station-Service NAFTAL

## 2.3 Travaux connexes

Les travaux suivants s'appuient sur des études récentes pour analyser concrètement l'apport des solutions blockchain, notamment via les plateformes Ethereum et Hyperledger Fabric dans les entreprises, que ce soit pour le partage d'informations , la gestion du crédit dans la chaîne d'approvisionnement, la logistique ou encore la sécurisation des processus comptables.

| Référence | Titre de l'étude | Objectif principal | Technologie utilisée | Résultats / Bénéfices clés |
|---|---|---|---|---|
| [42] | **L'utilisation de la technologie blockchain dans le partage des informations comptables et financières des entreprises** | Évaluer l'efficacité de la plateforme Ethereum pour le partage sécurisé, rapide et fiable d'informations comptables entre entreprises. Étude expérimentale sur 100 entreprises réparties en deux groupes, comparant blockchain vs méthodes classiques. | Ethereum | - Efficacité du partage améliorée de 25,7%<br>- Précision des données augmentée de 19,8%<br>- Réduction des coûts de 13,6%<br>- Moins d'erreurs, meilleure synchronisation entre partenaires |
| [20] | **La technologie blockchain pour le partage des informations de crédit des entreprises dans le financement de la chaîne d'approvisionnement** | Résoudre les limites du partage de données de crédit dans le financement de la chaîne d'approvisionnement, en particulier pour les PME : données falsifiées, manque de traçabilité, sécurité faible. Proposition d'un modèle sécurisé et automatisé. | Ethereum & Hyperledger Fabric | - Traçabilité, intégrité et confidentialité garanties<br>- Échange de données automatisé entre tous les acteurs<br>- Meilleure confiance dans l'écosystème<br>- Accès facilité au crédit pour les PME |
| [43] | **Solutions blockchain pour la gestion logistique** | Évaluer le potentiel de la blockchain (Hyperledger Fabric) dans la logistique : automatisation, sécurité et traçabilité des étapes d'acheminement. Mise en place d'un modèle hiérarchique et d'un cas pratique. | Hyperledger Fabric (modèle hiérarchique d'intégration) | - Traçabilité de chaque étape du processus logistique<br>- Accès en temps réel aux données partagées<br>- Suppression des interventions manuelles<br>- Transparence totale entre les partenaires |
| [44] | **La blockchain pour la prévention de la fraude : une transformation de la comptabilité et de la finance** | Étudier comment les blockchains privées (permissionnées) peuvent automatiser les contrôles internes et améliorer la sécurité des données financières. Focus sur la prévention de la fraude et la conformité. | Hyperledger Fabric | - Automatisation des contrôles comptables<br>- Sécurisation des transactions<br>- Audit en temps réel facilité<br>- Limitation des risques de fraude et accès restreint selon les droits |

**Tableau 2.1** – Études comparatives sur l'utilisation de la technologie blockchain dans le partage d'informations comptables, financières et logistiques en entreprise.

## 2.4 Analyse des besoins

Après avoir fait une analyse approfondie de la situation actuelle de la gestion des ventes dans les stations-service de Naftal, ainsi qu'une exploration de la littérature sur les technologies et méthodes modernes existantes dans le domaine informatique, nous constatons que les stations-service de Naftal ont besoin d'un système de paiement électronique basé sur la technologie blockchain afin de garantir une sécurité et une traçabilité optimales des transactions, ainsi qu'une efficacité accrue dans la gestion des opérations financières.

### 2.4.1 Limitations des méthodes actuelles

Les méthodes actuelles utilisées dans les stations-service de Naftal reposent sur des pratiques traditionnelles qui montrent aujourd'hui leurs limites. L'absence de modernisation impacte la performance globale de l'entreprise. Les principaux inconvénients sont présentés :

- **Saisie manuelle des données :** Le processus de saisie manuelle des données de transaction prend du temps et est sujet aux erreurs humaines, ce qui peut altérer la fiabilité des statistiques financières
- **Manque de mises à jour en temps réel :** Les transactions sont saisies uniquement à la fin d'une brigade, ce qui fausse l'évaluation précise nécessaire pour prendre des décisions juridiques ou opérationnelles sur le fonctionnement de l'entreprise.
- **Centralisation du pouvoir :** Le contrôle de la base de données et la validation des transactions sont entièrement détenues par une seule entité, ce qui expose le système à des risques de manipulation ou à un point de défaillance unique.
- **Manque de traçabilité pour les clients :** L'absence d'un système de suivi fiable qui permettrait au client de reconstituer l'historique complet de ses transactions effectuées.
- **Insuffisance des moyens de paiement :** Le système actuel ne prend pas en compte la diversité des moyens de paiement modernes tels que les cartes bancaires (Visa, Mastercard), ce qui limite la flexibilité pour les clients et le rend moins adapté aux exigences du marché actuel.

### 2.4.2 Intégration du paiement électronique et de la technologie blockchain dans la gestion de paiement dans les stations-service de NAFTAL

Malgré la volonté des entreprises à proposer des formes de paiement électronique adapté à leurs clients, aucun mécanisme de paiement n'est parfait, chacun présente ses

avantages et inconvénients Chaque entreprise doit sélectionner ceux qui semblent les mieux adaptés à sa situation [36]. Nous allons expliquer les propriétés souhaitables d'un système de e-Paiement :

- **L'utilisabilité :** Elle implique que le processus ne devrait nécessiter aucune expertise ou formation par les personnes susceptibles de l'utiliser, afin qu'ils puissent atteindre leurs propres objectifs de manière efficace, efficiente et satisfaisante [36].
- **La sécurité :** Ces dernières années, les cyberattaques contre les infrastructures financières ont causé d'importants dommages. Il devient donc indispensable de préserver les informations confidentielles des usagers et d'assurer la sécurité de leur argent. L'utilisation de technologies innovantes comme la blockchain permet de renforcer la sécurité des transactions grâce à la transparence, la traçabilité et l'immutabilité des données.
- **Anonymat :** Une fois l'achat effectué, la protection de l'identité et des données des clients devient essentielle [36]. Grâce à des mécanismes cryptographiques avancés de la technologie blockchain, il est possible de garantir un haut niveau d'anonymat et de confidentialité lors des transactions.
- **Fiabilité :** Le service doit fonctionner sans interruption et traiter correctement toutes les transactions, même en cas de panne ou de surcharge du système. L'intégration de la technologie blockchain permet d'augmenter la fiabilité du système grâce à sa nature décentralisée, qui réduit les risques de défaillance unique.
- **Rapidité :** Les paiements doivent être traités rapidement, avec une validation quasi-instantanée pour offrir une bonne expérience utilisateur.

### 2.4.3 Défis liés à l'intégration d'une nouvelle solution de E-Paiement

Bien qu'il soit indéniable que l'adoption d'une nouvelle solution de e-Paiement, telle que la technologie blockchain, offre des avantages considérables par rapport aux méthodes traditionnelles, il demeure essentiel de prendre en compte plusieurs défis majeurs liés à son intégration :

- **Compatibilité avec les systèmes existants :** Il est souvent difficile d'intégrer une nouvelle solution de paiement avec l'infrastructure technique déjà en place (applications, systèmes d'exploitation, terminaux, etc.)
- **Coût de migration et de maintenance :** Le déploiement d'une nouvelle technologie implique des coûts directs (achat de matériel, développement logiciel, formation, etc.) et indirects (temps d'adaptation, risques liés à la transition).
- **Profil des employés :** Les compétences et l'expérience des employés doivent être prises en compte, surtout en cas de problème technique, s'ils ne reçoivent pas une formation adéquate.

- **Scalabilité et performance :** Le système doit pouvoir gérer un grand nombre de transactions sans perte de performance, même lors de pics d'activité ou en cas de défaillance soudaine.
- **Connectivité réseau :** Un accès Internet fiable et rapide est essentiel, notamment dans les zones où la couverture réseau est limitée.

## 2.5 Conclusion

Les méthodes actuellement utilisées pour la gestion des ventes dans les stations-service de Naftal présentent d'importants défis non seulement en matière de sécurité, ce qui freine une prise de décision rapide et éclairée. Aujourd'hui, les systèmes de gestion de bases de données (SGBD) présentent plusieurs limites en termes de centralisation du pouvoir et de vulnérabilité face aux fraudes. Ce chapitre aborde ces points en analysant le système existant ainsi que les systèmes modernes utilisés dans les entreprises, afin d'identifier les besoins. L'objectif est de proposer une architecture système adaptée à l'ensemble des besoins fonctionnels et non fonctionnels, moderne et sécurisée, tout en respectant les contraintes techniques et opérationnelles propres à Naftal. Cela sera détaillé dans le prochain chapitre.

# Chapitre 3

# Conception d'une Solution Blockchain Adaptée aux Besoins de l'Entreprise Naftal

## 3.1 Introduction

En architecture logicielle, l'expression « by design » signifie qu'une caractéristique, un comportement ou une qualité particulière d'un système logiciel, telle que la sécurité ou la performance, est intentionnelle et planifiée dès les premières étapes de la conception. Cela garantit que ces aspects essentiels sont pris en compte de manière proactive, et non a posteriori (une réflexion après coup). C'est pourquoi ce chapitre se concentre sur l'architecture et le comportement général du système proposée, ses exigences fonctionnelles et non fonctionnelles, ainsi que sur la conception globale de la solution proposée.

## 3.2 Description de l'architecture et du comportement général du Système

Comme mentionné précédemment, notre objectif est de proposer une solution de système «self-service» au sein des stations-service Naftal. L'objectif principal consiste à modéliser le système existant et à automatiser les transactions financières, tout en offrant une expérience utilisateur optimale, fluide et personnalisée. Cette automatisation vise à améliorer la rapidité du service, à réduire les tâches manuelles et à minimiser les risques d'erreurs, contribuant ainsi à une gestion plus efficace et moderne des opérations.

Dans un premier temps, nous avons besoin d'une application mobile destinée aux clients, leur permettant de consulter l'historique de leurs transactions et d'effectuer de nouveaux paiements en toute autonomie. Le client commence par s'authentifier de manière sécurisée

et confidentielle. Ensuite, il scanne le code barre d'un produit afin de connaître son prix ce qui nécessite une interaction avec le serveur central via des requêtes HTTP.Après la sélection des produits qu'il souhaite acheter, le client se rend dans un espace dédié de la station-service équipé d'une application web. Cette dernière s'intègre avec le système mobile pour finaliser le paiement grâce à une connexion WebSocket, garantissant une interaction sécurisée, fluide et efficace. Par ailleurs, cette application web communique avec un réseau blockchain afin d'enregistrer toutes les transactions de manière sécurisée, transparente et efficace, comme l'illustre la figure suivante :

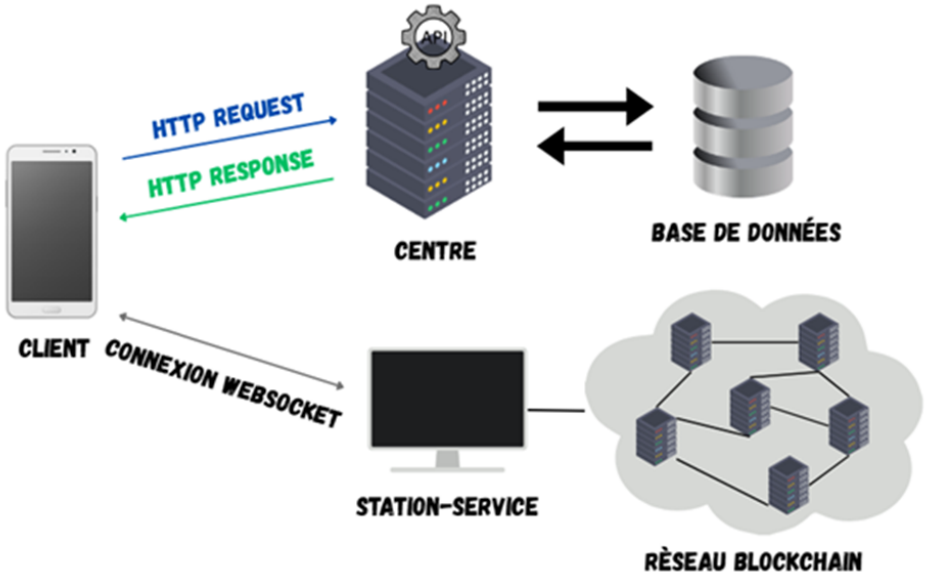


**Figure 3.1** – Présentation de l'architecture globale du système

L'interaction entre l'application mobile client et l'application web de la station-service faite via le protocole réseau WebSocket permettant un échange de données bidirectionnel en temps réel grâce à une unique connexion TCP persistante. Dans un premier temps, l'application web génère un token et l'envoie via une requête HTTP spéciale (upgrade request) au serveur, ce qui déclenche une phase de handshake permettant de passer du protocole HTTP au protocole WebSocket si le serveur accepte la demande. La connexion WebSocket est alors établie et une communication full-duplex est initiée.

Lorsqu'un client scanne un code QR, une session de communication est également initialisée et le canal actif avec le serveur, ce qui garantit un échange continu de messages entre l'application mobile et l'application web du système. Au début du processus, le client envoie la liste des produits sélectionnés pour l'achat. Un agent de la station-service confirme la cohérence entre les produits listés et ceux effectivement présents sur le lieu d'achat avant d'autoriser la procédure de paiement. Concernant les modes de paiement, trois options sont proposées : Carte Naftal, PayPal, ou via une API pour la carte Edahabia. Des tickets carburant (DOSN, MDN, NAFT, administratifs classiques) peuvent également être utilisés lors du paiement, chacun offrant un pourcentage de remise spéci-

fique à prendre en compte dans le calcul du montant total à payer. Une fois le paiement effectué, la connexion est alors fermée entre les applications et le serveur (voir la figure 3.2).

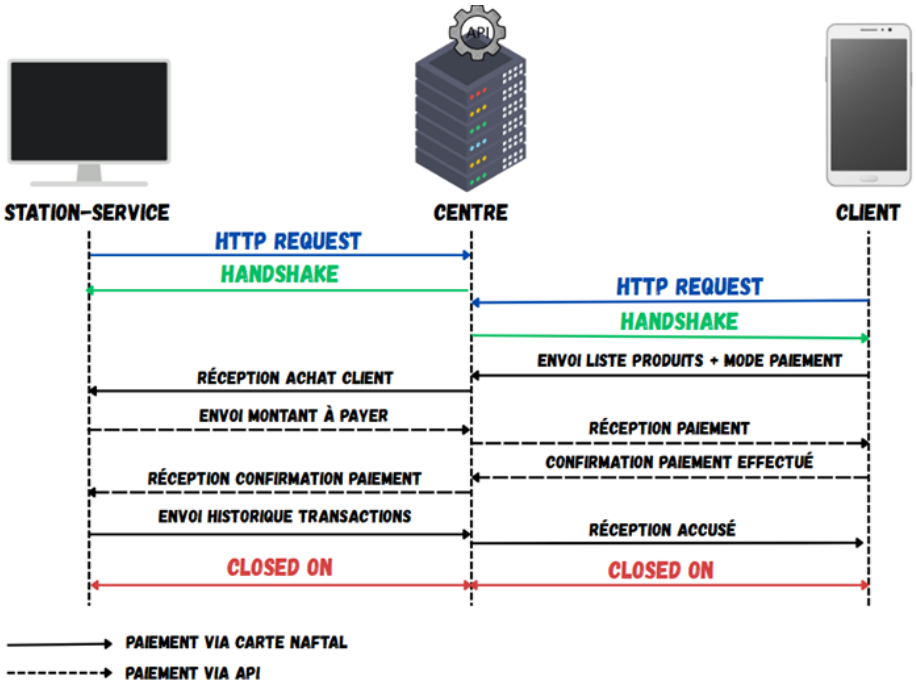


**FIGURE 3.2** – Présentation de L'interaction entre l'application mobile client et l'application web de la station-service

L'intégration de la technologie blockchain dans le processus de traitement et de stockage des transactions permet un suivi en temps réel des produits, de leur provenance, depuis la production jusqu'à la livraison. Elle assure un partage sécurisé de l'information entre le centre et les stations-service de Naftal. De plus, l'enregistrement et le suivi des transactions de paiement effectuées par les clients sont garantis grâce à un historique des données immuable et transparent. L'infrastructure technologique repose sur Hyperledger Fabric, une blockchain privée et permissionnée, offrant un contrôle strict des accès, une traçabilité des opérations, des performances élevées (TPS élevé), ainsi qu'un niveau de sécurité et de résilience accru. La figure 3.3 illustre schématiquement l'architecture technologique de réseau Hyperledger Fabric utilisée. Dans cette architecture, quatre peers sont utilisés ( deux peers par cannel )afin de garantir que chaque transaction soit enregistrée dans au moins deux registres distincts pour la redondance et la sécurité des données. Trois orderers sont mis en place pour assurer la résilience du système, la tolérance aux pannes et l'ordonnancement fiable des transactions. Deux channels (canaux) seront créés : le premier dédié à l'échange d'informations entre le centre et les stations-service, et le second destiné à la gestion des transactions des clients.

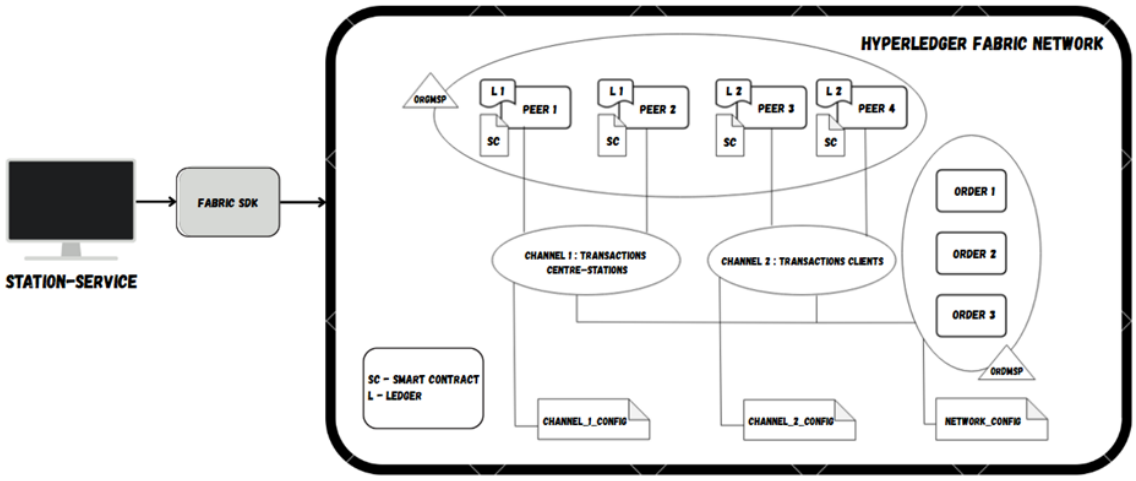


**Figure 3.3** – Architecture des Flux de Transactions entre l'application web de la station-service et le Réseau Hyperledger Fabric

## 3.3 Les exigences fonctionnelles et non fonctionnelles du Système

D'abord, les exigences fonctionnelles spécifient les caractéristiques et fonctionnalités que le système doit fournir afin de répondre aux besoins des utilisateurs, tandis que les exigences non fonctionnelles définissent les attributs de qualité du système.

### 3.3.1 Exigences fonctionnelles

Les exigences fonctionnelles suivantes doivent être respectées par le système proposé :

- **Authentification du Client :** L'application mobile doit permettre à chaque client de s'authentifier de manière sécurisée à l'aide de ses identifiants personnels (par exemple, numéro de téléphone, email et mot de passe) afin de garantir que chaque client peut accéder à son propre espace personnel, consulter ses informations en toute sécurité et effectuer des opérations protégées contre tout accès non autorisé.
- **Consultation de l'historique des transactions :** L'application mobile doit offrir à l'utilisateur la possibilité de consulter l'historique détaillé de toutes ses transactions réalisées au sein des stations-service Naftal. L'historique doit inclure la date, le montant, le type de paiement.
- **Effectuer un paiement :** L'application mobile doit permettre à l'utilisateur d'effectuer des paiements à tout moment de manière autonome, que ce soit en magasin ou directement à la pompe, sans passer par la caisse traditionnelle. Le paiement doit être rapide, fiable et sécurisé.

- **Interaction entre l'application mobile client et l'application web :** Le système doit permettre une communication fluide et sécurisée entre l'application mobile utilisée par le client et l'application web installée dans les stations-service, afin de fluidifier le processus d'achat, de paiement, et d'assurer la traçabilité des opérations.
- **Interface de l'application web avec le réseau blockchain :** L'application web doit être capable d'interagir avec le réseau blockchain (Hyperledger Fabric) pour enregistrer, valider et tracer toutes les transactions. Cette connexion garantit l'intégrité, la traçabilité et la sécurité des opérations effectuées par les clients.

### 3.3.2 Exigences non fonctionnelles :

Le système proposé doit répondre aux exigences non fonctionnelles suivantes :

- **Sécurité des données :** Le système doit garantir la sécurité des données des utilisateurs lors de la transmission, du traitement et du stockage des transactions. L'utilisation de la blockchain Hyperledger Fabric assure le chiffrement, la protection contre les accès non autorisés et la prévention de toute altération des données.
- **Évolutivité :** Le système doit être conçu de manière à pouvoir s'adapter facilement à une augmentation du nombre d'utilisateurs, de transactions ou de nouvelles fonctionnalités, sans perte de performance ni nécessité de refonte majeure de l'infrastructure.
- **Facilité d'utilisation :** L'interface utilisateur des applications (mobile et web) doit être intuitive, conviviale et accessible à tous les profils de clients, afin de garantir une prise en main rapide et de limiter les erreurs de manipulation.
- **Robustesse et résilience :** Le système doit être capable de fonctionner de manière fiable, même en cas de panne partielle, d'erreur technique ou de tentative d'attaque. Il doit assurer une continuité de service et une récupération rapide après incident.

## 3.4 Présentation et description des cas d'utilisation

Cette section présente un aperçu des principaux cas d'utilisation de notre système proposé à travers un Diagramme de cas d'utilisation global. Chaque cas d'utilisation est décrit en détail, incluant le déroulement principal des événements.

### 3.4.1 Acteurs

**1. Acteurs primaires**

- **Client :** l'utilisateur final du système joue un rôle clé dans l'écosystème. Généralement, il s'agit d'un particulier ou d'un représentant d'entreprise venant profiter des services de Naftal. Il utilise l'application mobile pour s'authentifier, consulter l'historique de ses transactions, sélectionner les produits et initier le processus de paiement, afin de bénéficier des services proposés par la station-service.
- **Agent :**L'agent joue un rôle crucial en vérifiant que les produits listés dans l'application correspondent bien à ceux effectivement présents sur le lieu d'achat avant d'autoriser le paiement, afin de garantir la fiabilité de la transaction.

**2. Acteurs secondaires**

- **Service de Paiement :**Le service de paiement est un acteur secondaire chargé de traiter les transactions financières lorsque le mode de paiement utilisé est différent de la Naftal-Carte. Il utilise également des API permettant d'interfacer et d'automatiser les échanges avec d'autres systèmes afin d'assurer le changement de monnaie du compte client vers le compte de la station.

### 3.4.2 Diagramme de cas d'utilisation global

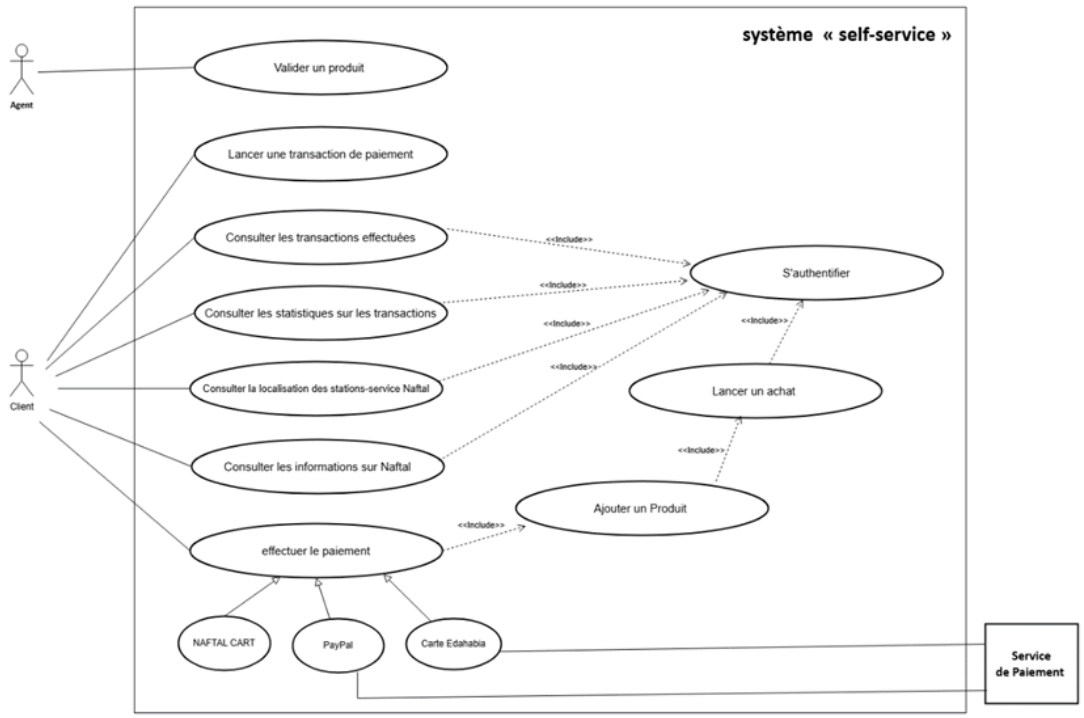


**Figure 3.4** – Diagramme de cas d'utilisation global

## 3.5 Présentation de la structure des données du système

Cette section présente un aperçu de la structure statique de notre système proposé à travers un diagramme de classes, utilisé pour modéliser la logique métier.

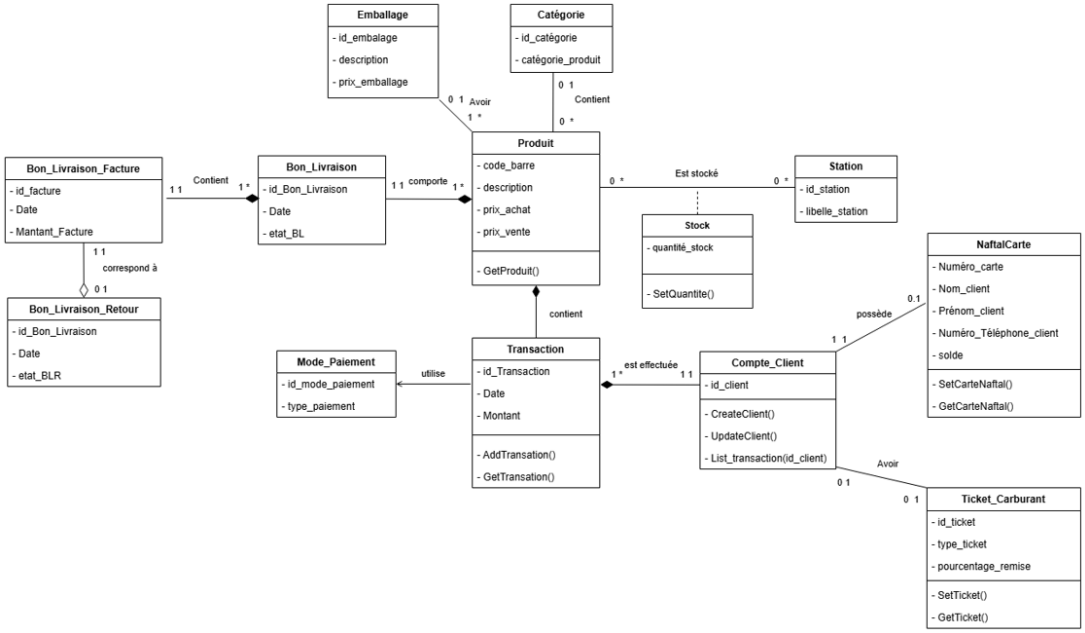


**Figure 3.5** – Diagramme de Classe global

## 3.6 Architecture de l'écosystème applicatif

Dans cette section, nous allons présenter l'architecture adoptée dans notre système, qui repose sur une approche orientée microservices. Cette architecture permet de garantir la modularité, l'évolutivité et la robustesse du système.

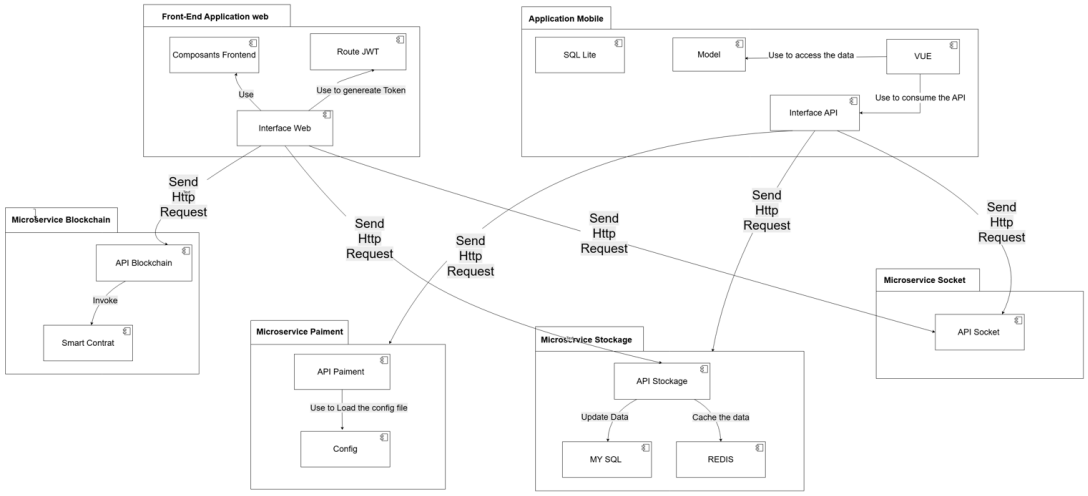


**Figure 3.6** – Architecture de l'écosystème applicatif

Comme illustré dans la Figure 3.6 , le système se compose de plusieurs modules indépendants, chacun étant responsable d'un aspect fonctionnel bien précis. Les principaux composants sont :

- **Front-End Application Web :** Elle contient les différents composants de l'application web, intègre une route dédiée pour générer un token JWT afin de garantir la sécurité des communications, et communique avec le micro-service Blockchain pour enregistrer ou récupérer les transactions.
- **Application Mobile :** Ce module utilise le package Vue pour les interfaces utilisateur et repose sur un modèle permettant l'accès aux données stockées localement dans une base SQL Lite. Toutes les transactions effectuées par l'utilisateur sont enregistrées dans cette base de données locale, ce qui permet de consulter l'historique à tout moment, même hors connexion.L'interaction avec les microservices s'effectue à l'aide de requêtes HTTP envoyées depuis l'interface API.
- **Microservice Blockchain :** Ce microservice est dédié à la gestion des interactions avec la blockchain. Il expose une API permettant d'invoquer des smart contracts pour l'enregistrement et la récupération des transactions, tout en offrant la possibilité d'ajouter d'autres fonctionnalités blockchain selon les besoins futurs du système.
- **Microservice Paiement :**Ce microservice gère l'ensemble des opérations liées aux paiements, en particulier lorsque l'utilisateur effectue une transaction via un service externe. Il permet de charger les fichiers de configuration nécessaires et assure l'intégration fluide des paiements dans le système global.
- **Microservice Stockage :**Ce microservice intervient lors de la sélection de produits par le client. Il accède aux données produits pour récupérer leur prix et leur disponibilité en temps réel. Les informations sont stockées dans une base MySQL et mises en cache via Redis pour optimiser les performances et l'accès rapide aux données.
- **Microservice Socket :** Il permet la communication en temps réel entre l'application mobile et l'application web de la station-service, assurant une synchronisation et une interaction instantanées entre les différents acteurs du système.

L'ensemble des composants communique principalement par des requêtes HTTP,une faible dépendance et une meilleure répartition des responsabilités. Cette organisation facilite également le déploiement, la maintenance et la montée en charge du système.

## 3.7 Exploration des processus interactifs du système

Cette section présente un aperçu de la dynamique de notre système proposé à travers un diagramme de séquence, utilisé pour représenter le processus de traitement et de stockage des transactions des clients C'est la tâche principale dans notre projet.

## Processus de l'authentification

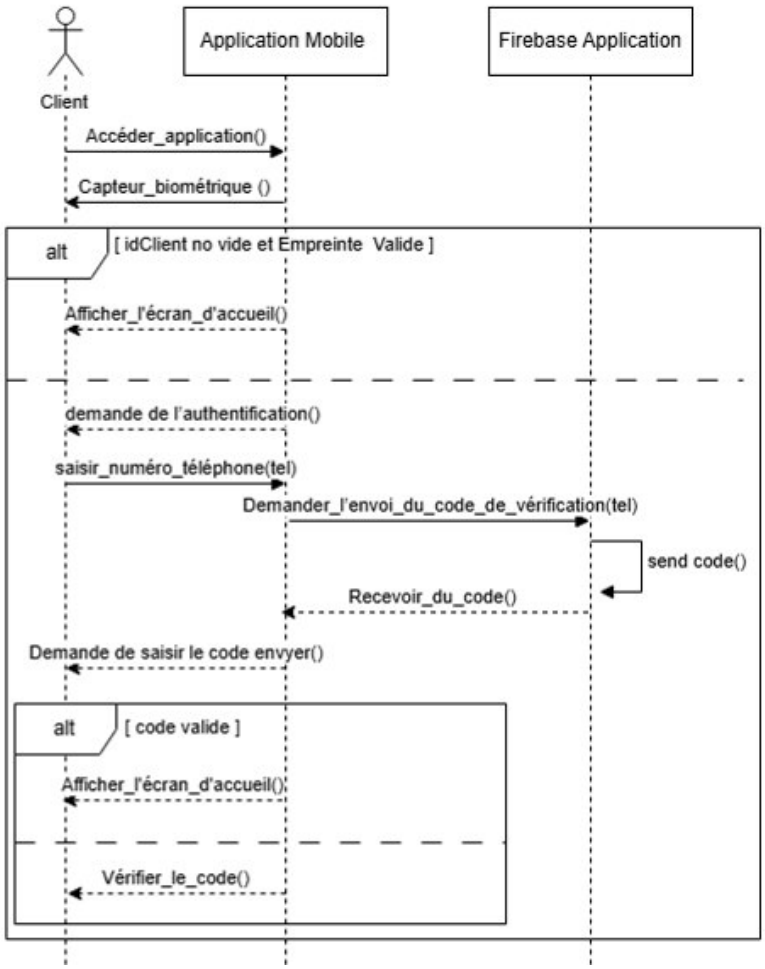


**Figure 3.7** – Diagramme de séquence de processus de l'authentification

# Processus de traitement et de stockage des transactions

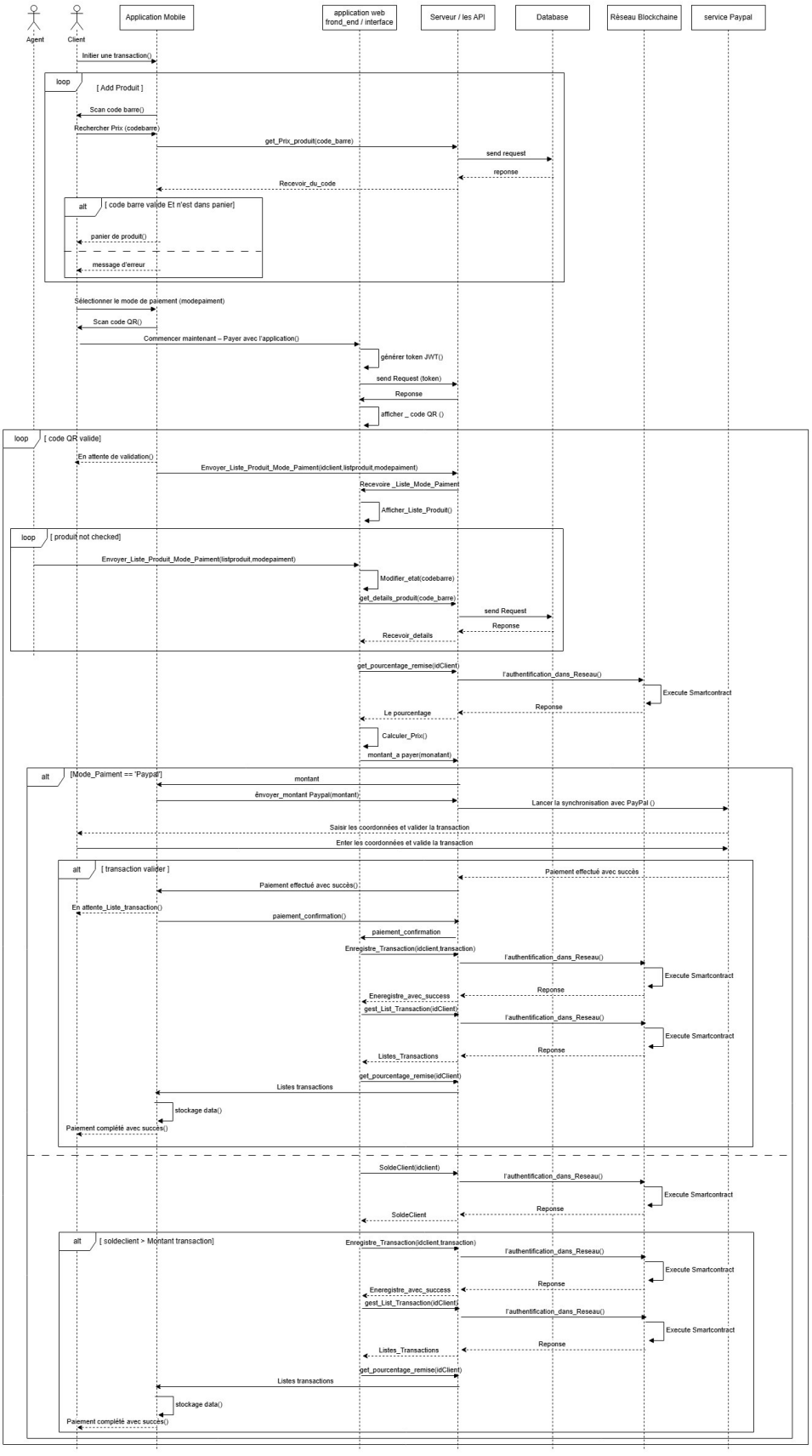


**Figure 3.8** – Diagramme de séquence de traitement et de stockage des transactions

## 3.8 Conclusion

Ce chapitre a permis de poser les fondations essentielles à la compréhension et à la modélisation du système proposé. L'approche par l'architecture du système a d'abord permis de structurer les composants et leurs interactions. La conception s'est appuyée sur les cas d'utilisation pour identifier les besoins fonctionnels et les acteurs principaux. Enfin, les diagrammes de classes et de séquence ont détaillé respectivement la structure statique du système et la dynamique des interactions lors du traitement des transactions des clients.

Cette approche globale de l'architecture applicative répond non seulement aux besoins immédiats d'automatisation du traitement des transactions, mais elle fournit également une base évolutive et maintenable pour les améliorations et extensions futures. En automatisant les tâches répétitives et en garantissant cohérence et précision, l'application contribue à améliorer significativement l'expérience client, dont l'implémentation sera détaillée dans le chapitre suivant..

# Chapitre 4

# La Solution Blockchain : Implémentation et Déploiement

## 4.1 Introduction

La phase de mise en œuvre de notre projet consiste à réaliser une simulation de l'architecture du système proposé, en développant une application mobile qui interagit avec les API et assure l'interfaçage avec une application web. Cette dernière enregistre les transactions via un réseau blockchain Hyperledger Fabric dans un environnement de test.

Les sections suivantes détaillent le processus d'implémentation, en commençant par la mise en place de la partie blockchain, puis le développement du backend, la création de l'application mobile et enfin le développement du frontend.

## 4.2 Environnement de développement

Nous avons utilisé deux machines personnelles et un téléphone pour le test de l'application mobile pendant la durée de notre projet :

| Machine | Spécifications |
|---|---|
| PC | • Processeur : Intel Core i5-1345U 1,60 GHz<br>• GPU : Intel® UHD Graphics<br>• RAM : 16 Go<br>• Système d'exploitation : Linux Mint 22 |
| PC | • Processeur : Intel Core i5-8350U 1,90 GHz<br>• GPU : Intel® UHD Graphics 620<br>• RAM : 8 Go<br>• Système d'exploitation : Windows 11 Pro |
| Téléphone | • Processeur : Samsung Exynos 9611<br>• RAM : 6 Go<br>• Version Android : Android 12 |

**Tableau 4.1** – Caractéristiques techniques des équipements utilisés

## 4.3 Implémentation de la blockchain

Cette section est consacrée à la mise en œuvre détaillée de l'élément clé de notre projet. Nous aborderons et expliquerons pourquoi nous avons choisi une suite logicielle spécifique. Nous passerons par la suite au processus d'implémentation, de la création des peers jusqu'à l'interaction avec les interfaces, en évaluant les performances du réseau déployé et en exposant les défis rencontrés tout au long du développement.

### 4.3.1 Aperçu des choix technologiques


**FIGURE 4.1** – Logos des outils d'implémentation d'un réseau Hyperledger Fabric de test

## Les choix technologiques

**Plateformes logicielles**

- **Docker (version : 24.0) :** Docker est une plateforme open-source qui permet aux développeurs de créer, déployer et exécuter des applications dans des conteneurs. Les conteneurs isolent les applications et leurs dépendances, facilitant ainsi le déploiement, la portabilité et la gestion des environnements applicatifs. [45]
- **Kubernetes (version : 1.27) :** Kubernetes est un système open-source d'orchestration de conteneurs, conçu pour automatiser le déploiement, la mise à l'échelle et la gestion d'applications conteneurisées. Il permet de gérer des clusters d'hôtes exécutant des conteneurs Docker, offrant haute disponibilité, scalabilité et flexibilité. [46]
- **Hyperledger Fabric (version : 3.0) :** Hyperledger Fabric est un framework blockchain open-source destiné aux applications d'entreprise. Il dispose d'une architecture modulaire et flexible, adaptée à la création de blockchains privées et permissionnées, avec un haut niveau de confidentialité, performance et sécurité. [47]
- **Istio (version : 2.5)** est une plateforme open source de service Mesh qui fournit une manière uniforme de connecter, sécuriser, contrôler et observer les microservices. Il facilite la gestion des communications entre services distribués, en offrant des fonctionnalités comme le routage intelligent du trafic, la sécurité renforcée, la télémétrie complète et le contrôle d'accès.
- **Hyperledger Caliper (version : 0.5) :** Hyperledger Caliper est un outil de benchmarking blockchain open-source. Il permet de mesurer les performances des plateformes blockchain (Hyperledger Fabric, Ethereum, ...), en évaluant notamment le débit de transactions, la latence et l'utilisation des ressources. [47]

**Langages de programmation**

- **Golang (version : 1.21) :** Go, ou Golang, est un langage open-source développé par Google. Il est apprécié pour sa simplicité, ses performances, et sa gestion effi-

cace de la concurrence, notamment pour le développement d'applications réseau, de microservices et de systèmes distribués. [48]

## Justification des choix technologiques

Hyperledger Fabric est composé de plusieurs composants (peers, orderers, autorités de certification – CA, chaincodes, etc.) qui doivent tourner simultanément et collaborer dans un réseau distribué. Kubernetes facilite cela en orchestrant ces composants grâce à sa gestion des conteneurs. Chaque composant Fabric fonctionne dans un conteneur Docker, ce qui standardise le déploiement et l'exécution sur n'importe quel environnement, tout en assurant portabilité, légèreté et indépendance. L'intégration d'Istio dans le cluster Kubernetes permet en plus de sécuriser et contrôler le trafic entre les microservices Fabric, grâce au chiffrement automatique (mutual TLS), à la gestion intelligente du trafic et à une observabilité renforcée. Kubernetes facilite cela en offrant :

- une gestion automatique des conteneurs (redémarrage en cas de panne, montée en charge automatique),
- une haute disponibilité et une tolérance aux pannes grâce à la réplication et au load balancing,
- une isolation via les namespaces,
- une sécurité renforcée avec la gestion des accès (RBAC) et des secrets,
- une configuration centralisée et déclarative des ressources à travers les fichiers YAML.

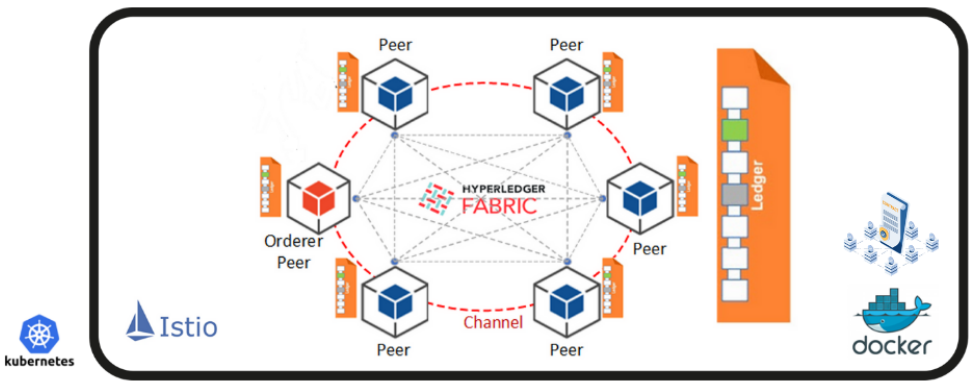


**FIGURE 4.2** – L'interopérabilité entre les différents outils de déploiement dans un réseau Hyperledger Fabric : Kubernetes, Istio et Docker

Pour la visibilité et la mesure des performances du réseau, Hyperledger Caliper peut être intégré afin de générer des charges de test réalistes, mesurer le débit (throughput), la latence des transactions et la consommation des ressources. Cela permet d'évaluer objectivement la performance du réseau Fabric, d'identifier les éventuels goulots d'étranglement et d'optimiser l'architecture avant un déploiement en production.

### 4.3.2 Processus de déploiement du réseau Hyperledger Fabric

Avant d'aborder la mise en œuvre technique sur Kubernetes, il est important de comprendre le flux d'interaction entre les différents composants du réseau Hyperledger Fabric utilisés dans notre architecture, notamment les peers, les orderers, le client SDK et le chaincode. Ces composants échangent entre eux de manière sécurisée grâce à deux types de certificats : les certificats de signature (Sign certificates), utilisés pour l'authentification et la validation des identités, et les certificats TLS (Transport Layer Security), utilisés pour sécuriser les communications réseau entre les nœuds.

La figure ci-dessous illustre ce processus d'interaction sécurisé, depuis l'appel d'une transaction via l'API, son traitement par le client SDK, jusqu'à sa propagation aux peers et à l'orderer dans le réseau Fabric

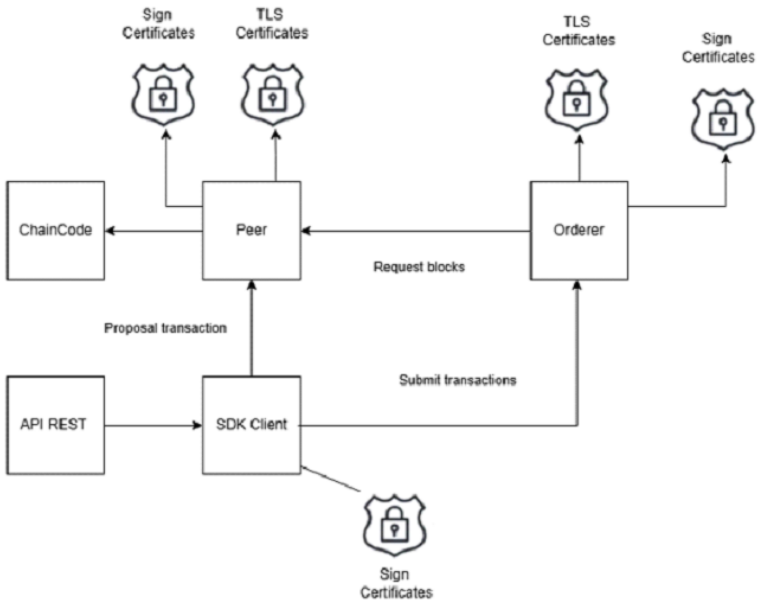


**Figure 4.3** – Le flux d'interaction entre les différents composants du réseau Hyperledger Fabric

# 1 - Création d'un cluster Kubernetes :

Pour commencer le déploiement de notre réseau Hyperledger Fabric, nous devons disposer d'un cluster Kubernetes fonctionnel.

Dans la configuration utilisée, les ports 80 (HTTP) et 443 (HTTPS) de la machine hôte sont mappés respectivement aux ports 30949 et 30950 du cluster. Cela permet de rediriger le trafic externe vers le cluster afin d'accéder aux services déployés.

# 2 - Configuration d'Istio :

Nous allons maintenant configurer Istio, un service Mesh qui facilitera la gestion, la sécurisation et l'observabilité des communications entre les composants de notre réseau Hyperledger Fabric.Pour cela, nous appliquons la configuration qui permet :

- De déployer une passerelle d'entrée (ingressGateway) avec un autoscaling minimal de 2 réplicas, exposant les ports HTTP (80) et HTTPS (443) via des NodePort mappés aux mêmes ports configurés dans le cluster Kubernetes.
- D'activer le composant pilot, qui gère la configuration du maillage Istio, avec un autoscaling minimal d'une réplique.
- De désactiver les composants de monitoring et traçage (Grafana, Kiali, Prometheus, Tracing) afin de simplifier l'installation.
- De configurer la politique réseau pour autoriser tout le trafic sortant ($\mathrm{ALLOW}_{ANY}$).

Pour gérer la résolution DNS interne dans le cluster Kubernetes, nous appliquons un paramétrage spécifique sur CoreDNS. Cela permet de réécrire certains noms DNS afin qu'ils pointent vers le service Istio ingressgateway.

# 3 - Installation de opérateur de tissu hyperledger :

À cette étape, nous installons l'opérateur Kubernetes pour Hyperledger Fabric, ceci va installer :

- CRD (Custom Resource Definitions) pour déployer des pairs, , des orderers et des autorités de certification,
- Le contrôleur de l'opérateur, qui gère automatiquement le cycle de vie des composants Fabric sur Kubernetes.

# 4 - Déploiement d'une organisation de pairs :

**Définition des variables d'environnement**

Avant de commencer le déploiement les composants, nous définissons les variables d'environnement nécessaires :

```
export PEER_IMAGE=hyperledger/fabric-peer
export PEER_VERSION=3.0.0-preview

export ORDERER_IMAGE=hyperledger/fabric-orderer
export ORDERER_VERSION=3.0.0-preview

export CA_IMAGE=hyperledger/fabric-ca
export CA_VERSION=1.5.7
```

**FIGURE 4.4** – les variables d'environnement

### Déploiement de l'autorité de certification (CA)

Nous commençons par déployer l'autorité de certification pour l'organisation Org1. Celle-ci est responsable de délivrer les certificats aux différents composants de l'organisation (pairs, etc).

```
export STORAGE_CLASS=standard

kubectl hlf ca create \
    --image=$CA_IMAGE --version=$CA_VERSION --storage-class=$STORAGE_CLASS \
    --capacity=1Gi --name=org1-ca --enroll-id=enroll \
    --enroll-pw=enrollpw --hosts=org1-ca.localho.st --istio-port=443
```

**FIGURE 4.5** – L'autorité de certification pour l'organisation Org1

Une fois la CA déployée, nous enregistrons un utilisateur de type peer avec ses identifiants. Cet utilisateur sera utilisé pour authentifier les nœuds pairs de l'organisation lors de leur création et de leur intégration au réseau.

```
kubectl hlf ca register \
  --name=org1-ca --user=peer --secret=peerpw \
  --type=peer --enroll-id=enroll --enroll-secret=enrollpw \
  --mspid=Org1MSP
```

**FIGURE 4.6** – enregistrement d'un utilisateur de type peer

### Déploiement des pairs de l'organisation

Conformément à la conception définie précédemment, nous allons déployer quatre pairs. Chaque pair utilise le même utilisateur et mot de passe que nous avons enregistrés.

```
kubectl hlf peer create \
  --statedb=couchdb --image=$PEER_IMAGE --version=$PEER_VERSION \
  --storage-class=$STORAGE_CLASS --enroll-id=peer --mspid=Org1MSP \
  --enroll-pw=peerpw --capacity=5Gi --name=org1-peer0 \
  --ca-name=org1-ca.default --hosts=peer0-org1.localho.st --istio-port=443

kubectl hlf peer create \
  --statedb=couchdb --image=$PEER_IMAGE --version=$PEER_VERSION \
  --storage-class=$STORAGE_CLASS --enroll-id=peer --mspid=Org1MSP \
  --enroll-pw=peerpw --capacity=5Gi --name=org1-peer1 \
  --ca-name=org1-ca.default --hosts=peer1-org1.localho.st --istio-port=443

kubectl hlf peer create \
  --statedb=couchdb --image=$PEER_IMAGE --version=$PEER_VERSION \
  --storage-class=$STORAGE_CLASS --enroll-id=peer --mspid=Org1MSP \
  --enroll-pw=peerpw --capacity=5Gi --name=org1-peer2 \
  --ca-name=org1-ca.default --hosts=peer2-org1.localho.st --istio-port=443

kubectl hlf peer create \
  --statedb=couchdb --image=$PEER_IMAGE --version=$PEER_VERSION \
  --storage-class=$STORAGE_CLASS --enroll-id=peer --mspid=Org1MSP \
  --enroll-pw=peerpw --capacity=5Gi --name=org1-peer3 \
  --ca-name=org1-ca.default --hosts=peer3-org1.localho.st --istio-port=443
```

**Figure 4.7** – La creation des peers

# 5 - Déploiement d'une organisation Orderer :

Le déploiement de l'organisation Orderer suit les mêmes étapes que pour l'organisation de peers décrite précédemment, avec quelques ajustements spécifiques au rôle des orderers dans le réseau. En particulier, les nœuds orderer nécessitent l'option –admin-hosts, qui permet d'autoriser explicitement certaines connexions d'administration via TLS mutualisé. Cette configuration est essentielle pour garantir que seuls des clients ou outils de gestion identifiés puissent interagir avec les orderers de manière sécurisée, notamment lors des opérations sensibles comme la création de canaux ou la mise à jour du consortium.

```
# Création de la CA pour l'organisation Orderer
kubectl hlf ca create \
  --image=$CA_IMAGE --version=$CA_VERSION --storage-class=$STORAGE_CLASS \
  --capacity=1Gi --name=ord-ca --enroll-id=enroll \
  --enroll-pw=enrollpw --hosts=ord-ca.localho.st --istio-port=443

# Enregistrement de l'utilisateur 'orderer' auprès de la CA
kubectl hlf ca register \
  --name=ord-ca --user=orderer --secret=ordererpw \
  --type=orderer --enroll-id=enroll --enroll-secret=enrollpw \
  --mspid=OrdererMSP --ca-url="https://ord-ca.localho.st:443"

# Création du nœud orderer 1
kubectl hlf ordnode create \
  --image=$ORDERER_IMAGE --version=$ORDERER_VERSION --storage-class=$STORAGE_CLASS \
  --enroll-id=orderer --mspid=OrdererMSP --enroll-pw=ordererpw \
  --capacity=2Gi --name=ord-node1 --ca-name=ord-ca.default \
  --hosts=orderer0-ord.localho.st --admin-hosts=admin-orderer0-ord.localho.st \
  --istio-port=443

# Création du nœud orderer 2
kubectl hlf ordnode create \
  --image=$ORDERER_IMAGE --version=$ORDERER_VERSION --storage-class=$STORAGE_CLASS \
  --enroll-id=orderer --mspid=OrdererMSP --enroll-pw=ordererpw \
  --capacity=2Gi --name=ord-node2 --ca-name=ord-ca.default \
  --hosts=orderer1-ord.localho.st --admin-hosts=admin-orderer1-ord.localho.st \
  --istio-port=443

# Création du nœud orderer 3
kubectl hlf ordnode create \
  --image=$ORDERER_IMAGE --version=$ORDERER_VERSION --storage-class=$STORAGE_CLASS \
  --enroll-id=orderer --mspid=OrdererMSP --enroll-pw=ordererpw \
  --capacity=2Gi --name=ord-node3 --ca-name=ord-ca.default \
  --hosts=orderer2-ord.localho.st --admin-hosts=admin-orderer2-ord.localho.st \
  --istio-port=443
```

**Figure 4.8** – Déploiement des Orderer

# 6 - Création du canal :

### Enregistrement et enrôlement des identités administratives

Avant de pouvoir créer un canal, il est nécessaire d'enregistrer et d'enrôler les identités administratives pour les organisations qui vont participer au canal, notamment celles des orderers et des peers.

**o OrdererMSP**

- **Enregistrement de l'administrateur Orderer auprès de la CA :**

```
kubectl hlf ca register \
  --name=ord-ca --user=admin --secret=adminpw \
  --type=admin --enroll-id=enroll --enroll-secret=enrollpw \
  --mspid=OrdererMSP
```

**FIGURE 4.9** – Enregistrement de l'administrateur Orderer auprès de la CA

- **Création de l'identité de signature (sign) :**

```
kubectl hlf identity create \
  --name=orderer-admin-sign --namespace=default --ca-name=ord-ca \
  --ca-namespace=default --ca=ca --mspid=OrdererMSP \
  --enroll-id=admin --enroll-secret=adminpw
```

**FIGURE 4.10** – Création de l'identité de signature

- **Création de l'identité TLS :**

```
kubectl hlf identity create \
  --name=orderer-admin-tls --namespace=default --ca-name=ord-ca \
  --ca-namespace=default --ca=tlsca --mspid=OrdererMSP \
  --enroll-id=admin --enroll-secret=adminpw
```

**FIGURE 4.11** – Création de l'identité TLS

**o Org1MSP**

- **Enregistrement de l'administrateur Org1 :**

```
kubectl hlf ca register \
  --name=org1-ca --namespace=default --user=admin \
  --secret=adminpw --type=admin --enroll-id=enroll \
  --enroll-secret=enrollpw --mspid=Org1MSP
```

**FIGURE 4.12** – Enregistrement de l'administrateur Org1

- **Création de l'identité d'administration Org1 :**

```
kubectl hlf identity create \
  --name=org1-admin --namespace=default --ca-name=org1-ca \
  --ca-namespace=default --ca=ca --mspid=Org1MSP \
  --enroll-id=admin --enroll-secret=adminpw
```

**FIGURE 4.13** – Création de l'identité d'administration Org1

### Préparation des certificats nécessaires

Pour permettre la création du canal, les certificats TLS et de signature (SignCert) doivent être exportés à partir des autorités de certification (CA) correspondantes : Certificats pour l'organisation Orderer

```
# Certificats pour l'organisation Orderer
export ORDERER_TLS_CERT=$(kubectl get fabriccas ord-ca -o=jsonpath='{.status.tlsca_cert}')
export ORDERER_SIGN_CERT=$(kubectl get fabriccas ord-ca -o=jsonpath='{.status.ca_cert}')

# Certificats pour l'organisation Org1 (Peer)
export PEER_ORG_TLS_CERT=$(kubectl get fabriccas org1-ca -o=jsonpath='{.status.tlsca_cert}')
export PEER_ORG_SIGN_CERT=$(kubectl get fabriccas org1-ca -o=jsonpath='{.status.ca_cert}')
```

**FIGURE 4.14** – Création de Certificats pour l'organisation Orderer

### Configuration du consensus

Deux types de mécanismes de consensus ont été utilisés dans le déploiement afin de permettre une comparaison technique et opérationnelle. Cela vise à identifier lequel est le plus adapté aux besoins du réseau.

Le tableau ci-dessous présente une comparaison synthétique des principales configurations appliquées à chacun :

| **Consensus spécifique** | **smartBFT** | **etcdRaft** |
|---|---|---|
| **ordererType** | BFT | etcdraft |
| **batchSize.max Message-Count** | 100 | 10 |
| **batchTimeout** | 2s | 2s |
| **ordererEndpoints** | orderer0-ord.localho.st :443 | ord-node1 :7050 |

**TABLEAU 4.2** – Tableau comparatif des configurations SmartBFT et etcdRaft

Après la création du canal, il est nécessaire de joindre les peers de l'organisation au canal afin qu'ils puissent commencer à recevoir les blocs, valider les transactions et exécuter les contrats intelligents.

### Installation et déploiement du chaincode

l'étape suivante consiste à installer et déployer le chaincode sur les pairs. Cela se fait en plusieurs phases :

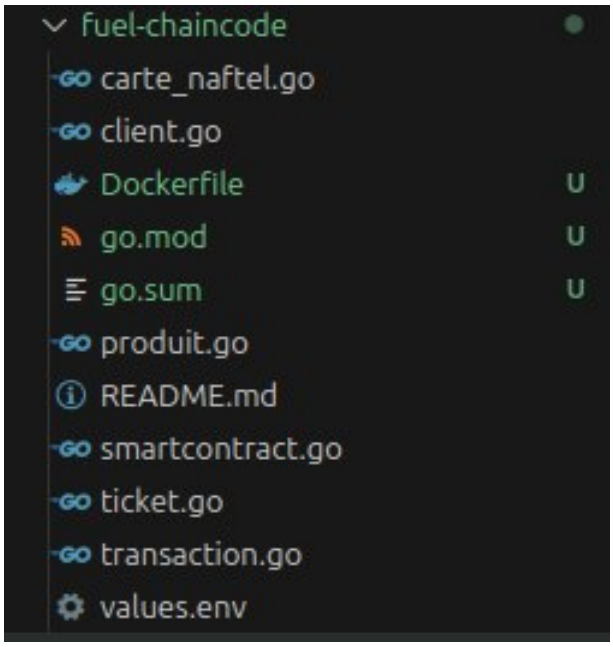


**FIGURE 4.15** – La structure du Chaincode pour les transaction des clients

La première étape consiste à développer et déployer le chaincode sur Docker selon une structure modulaire (Figure 4.15) qui permet une gestion claire des fonctions de gestion d'actifs. Après l'installation, nous procédons à l'approbation (approval) du chaincode pour le canal.Une fois cette approbation effectuée, le commit final du chaincode est réalisé sur le canal, le rendant ainsi actif et prêt à être invoqué par les clients.

```
# 1. Créer le NetworkConfig et exporter la config pour Org1
kubectl hlf networkconfig create \
  --name=org1-cp -o Org1MSP -o OrdererMSP -c etcdraft \
  --identities=org1-admin.default --secret=org1-cp

kubectl get secret org1-cp \
  -o jsonpath="{.data.config\.yaml}" | base64 --decode > org1.yaml

# 2. Calculer le package ID du chaincode
export PACKAGE_ID=$(kubectl hlf chaincode calculatepackageid \
  --path=chaincode.tgz --language=golang --label=$CHAINCODE_LABEL)

# 3. Installer le chaincode sur les peers Org1
kubectl hlf chaincode install \
  --path=./chaincode.tgz --config=org1.yaml \
  --language=golang --label=$CHAINCODE_LABEL \
  --user=org1-admin-default --peer=org1-peer0.default

kubectl hlf chaincode install \
  --path=./chaincode.tgz --config=org1.yaml \
  --language=golang --label=$CHAINCODE_LABEL \
  --user=org1-admin-default --peer=org1-peer1.default
```

```
# 4. Déployer le conteneur chaincode dans le cluster
kubectl hlf externalchaincode sync \
  --image=bersasma/fuel-chaincode:1.0 --name=$CHAINCODE_NAME \
  --namespace=default --package-id=$PACKAGE_ID \
  --tls-required=false --replicas=1

# 5. Approuver la définition du chaincode pour Org1
export SEQUENCE=1
export VERSION="1.0"

kubectl hlf chaincode approveformyorg \
  --config=org1.yaml --user=org1-admin-default \
  --peer=org1-peer0.default --package-id=$PACKAGE_ID \
  --version="$VERSION" --sequence="$SEQUENCE" \
  --name=$CHAINCODE_NAME --policy="OR('Org1MSP.member')" \
  --channel=$CHANNEL_NAME

# 6. Committer la définition du chaincode
kubectl hlf chaincode commit \
  --config=org1.yaml --user=org1-admin-default --mspid=Org1MSP \
  --version="$VERSION" --sequence="$SEQUENCE" --name=$CHAINCODE_NAME \
  --policy="OR('Org1MSP.member')" --channel=$CHANNEL_NAME
```

**FIGURE 4.16** – Déploiement du Chaincode

L'interface graphique présentée ci-dessous montre l'état actuel des pods déployés dans un cluster Kubernetes. Chaque pod représente une unité d'exécution qui héberge un ou plusieurs conteneurs. Cette vue permet de surveiller en temps réel les ressources, l'état de fonctionnement, le redémarrage éventuel des pods, ainsi que les nœuds sur lesquels ils sont exécutés. Elle est essentielle pour assurer la stabilité et la gestion efficace des services dans un environnement distribué.

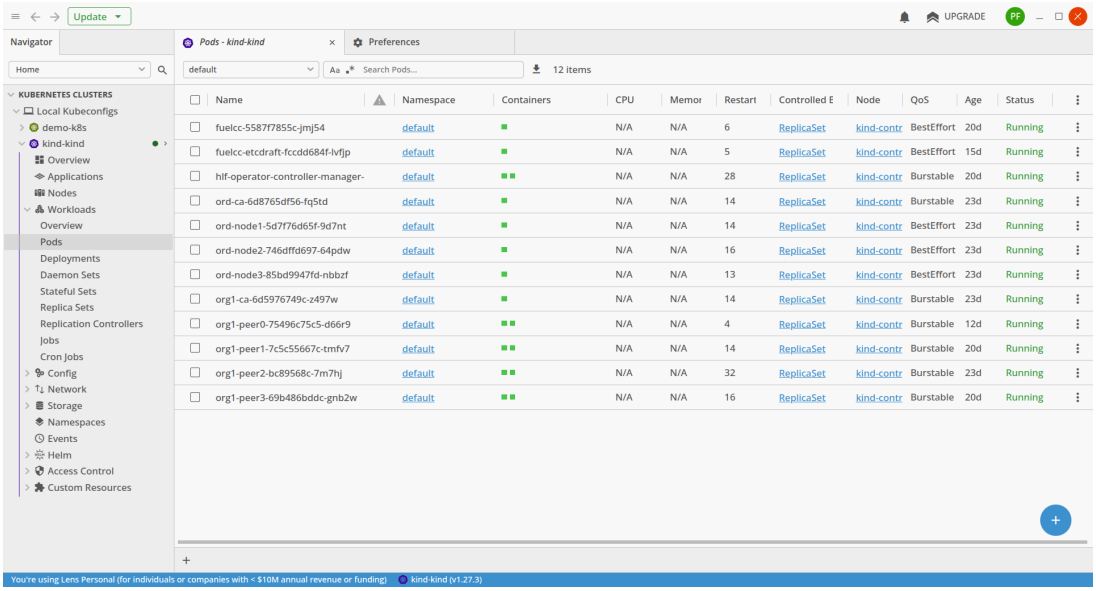


**FIGURE 4.17** – Vue des pods en cours d'exécution dans le cluster Kubernetes via l'interface utilisateur Lens

# 7 - Client SDK :

Une fois le chaincode déployé, les clients peuvent interagir avec la blockchain via un SDK (Software Development Kit). Le SDK permet de :

- Charger les identités utilisateurs (depuis un wallet sécurisé),
- Établir une connexion avec le réseau via une passerelle (Gateway),
- Soumettre des transactions (invoke) ou effectuer des lectures (query),
- Écouter les événements émis par les chaincodes.

En termes de stockage, les données sont enregistrées dans le ledger de la blockchain, assurant l'historique, l'immutabilité et la traçabilité,et de l'autre, une couche base de données (coucheDB) être utilisée pour des besoins applicatifs spécifiques : lecture rapide

# 8 - Api Client :

Pour faciliter l'intégration avec applications web, des API RESTful sont construites au-dessus du SDK. Ces API agissent comme une interface intermédiaire entre les utilisateurs finaux et la blockchain, afin de garantir une abstraction sécurisée, cohérente et accessible de la couche blockchain.
La figure ci-dessus montre que le ledger augmente lorsqu'on ajoute un bloc après 2 secondes d'exécution du smart contract, et devient composé de 49 blocs.

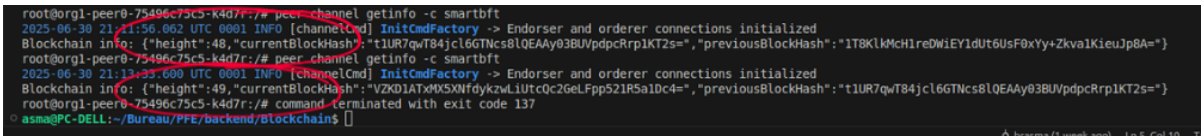


**Figure 4.18** – Déploiement de réseau Hyperledger Fabric avec Kubernetes

## 4.4 Mise en œuvre des microservice

Cette section présente la mise en œuvre des microservices utilisés dans notre système . Chaque microservice est conçu pour remplir une fonction précise, assurant ainsi modularité, performance et évolutivité du système.

### 4.4.1 Technologies utilisées pour l'API


**FIGURE 4.19** – Logos des outils d'implémentation d'une API

**Backend et runtime**

- **Node.js (version : 22.12.0) :** Node.js est un environnement d'exécution JavaScript côté serveur, basé sur le moteur V8 de Chrome. Asynchrone et orienté I/O, il est conçu pour créer des applications réseau scalables et performantes. [49]

**Bases de données**

- **Redis (version : 7.2.5) :** Redis est un système de stockage en mémoire open-source, basé sur des structures de données avancées. Il est principalement utilisé pour le caching, la gestion de sessions et les files d'attente distribuées. [50]
- **MySQL (version : 8.0) :** MySQL est un système de gestion de base de données relationnelle open-source, supportant les transactions, la réplication et garantissant la haute disponibilité. [51]

### 4.4.2 Microservice de Stockage

Ce microservice est dédié à la gestion et à la récupération des informations des produits stockés dans une base de données MySQL. Il expose deux endpoints principaux via une API REST construite avec Express.js. Le premier permet d'obtenir les détails d'un produit à partir de son code-barres, tandis que le second inclut également les informations de stock spécifiques à une station donnée. Pour améliorer les performances, il utilise Redis comme système de cache afin de réduire la charge sur la base de données et accélérer les temps de réponse. Ce microservice joue un rôle central dans la gestion des données produits et contribue à construire une architecture réactive et évolutive.

### 4.4.3 Microservice pour socket

Ce microservice implémente un serveur WebSocket permettant une communication bidirectionnelle en temps réel entre un client web (front-end) et un client mobile (application mobile). Chaque connexion est authentifiée par un token, qui agit comme un canal privé de communication. Le serveur gère différents types de messages métiers comme la demande de liste de produits, le paiement, la confirmation de paiement ou encore la

synchronisation des transactions. Ce système assure une interaction fluide et instantanée entre les deux parties, essentielle pour des cas d'usage comme les paiements rapides ou les mises à jour de panier en temps réel.

### 4.4.4 Microservice pour Paiement

Ce microservice gère actuellement l'intégration avec l'API PayPal, en attendant d'être étendu à d'autres solutions de paiement. Il permet la création et la capture des commandes de paiement et Il expose une API REST qui permet de générer un lien de paiement sécurisé, redirigeant ensuite l'utilisateur vers PayPal pour compléter la transaction. Une fois celle-ci effectuée, le microservice capture le paiement et redirige vers une page de succès ou d'échec. Il utilise Axios pour les appels HTTP vers les serveurs PayPal et repose sur un système d'authentification basé sur OAuth2. Ce composant est essentiel pour offrir un moyen de paiement fiable, sécurisé et reconnu à l'échelle mondiale dans l'application.

## 4.5 Implémentation de l'application mobile

Dans cette section, nous ne mettons pas l'accent sur la structure générale de l'implémentation de l'application mobile, mais plutôt sur les mécanismes de sécurité, de contrôle d'accès et la communication sécurisée avec les microservices.

### 4.5.1 Technologies utilisées pour l'application mobile


**FIGURE 4.20** – Logos des outils d'implémentation d'une application mobile

**Frontend mobile**

- **Flutter (version : 3.19) :** Flutter est un SDK UI open-source de Google permettant de créer des applications mobiles, web et desktop à partir d'un seul codebase. Il repose sur le langage Dart et fournit un riche écosystème de widgets. [52]

**Bases de données**

- **SQLite (version : 2.3.3) :** SQLite est un moteur SQL léger et embarqué, ne nécessitant pas de serveur, idéal pour les applications mobiles et embarquées. [53]

- **Firebase (version : récente) :** Firebase est une plateforme de développement (Google) offrant des services backend comme base de données en temps réel, authentification, hébergement, analytics et notifications. [54]

### 4.5.2 Mécanismes de sécurité et contrôle d'accès

L'authentification des utilisateurs s'effectue via leur numéro de téléphone, en utilisant Firebase Authentication. Lors de la première connexion, l'utilisateur reçoit un code de vérification par SMS, qui permet de valider son identité de manière fiable. Après authentification, le numéro de téléphone est immédiatement haché en SHA-256 et stocké localement dans le stockage sécurisé de l'application. Ce hash est ensuite utilisé pour identifier l'utilisateur lors du traitement des transactions, sans exposer directement ses données personnelles. En complément, une authentification biométrique par empreinte digitale est utilisée comme première couche de sécurité lors de l'accès à l'application.

Concernant les transactions, celles-ci sont enregistrées temporairement dans une base de données SQLite locale intégrée à l'application mobile. Cette base a pour rôle de gérer l'état local des achats en cours, permettant une fluidité dans le parcours utilisateur même en cas de perte temporaire de connexion réseau. Une fois que l'utilisateur se déconnecte de l'application, toutes les données locales relatives aux transactions sont automatiquement vidées, cela renforce la confidentialité des données utilisateur et limite les risques d'exposition en cas de perte ou de vol de l'appareil.

De plus, lors de l'utilisation de méthodes de paiement telles que PayPal, aucune donnée bancaire ou coordonnée personnelle du client n'est stockée sur le téléphone, assurant une conformité stricte avec les bonnes pratiques de sécurité et de respect de la vie privée

### 4.5.3 Communication avec les microservices

L'application communique avec les microservices via des requêtes REST et des WebSockets. Par exemple, pour initier un paiement via PayPal, elle utilise une API distante :

```dart
Future<String?> createPaypalOrder(double amount, String currency) async {
  final response = await http.post(
    Uri.parse('http://$ipHost:3000/api/paypal/create-order'),
    headers: {'Content-Type': 'application/json'},
    body: jsonEncode({
      'amount': amount.toStringAsFixed(2),
      'currency': currency,
    }),
  );
  if (response.statusCode == 200) {
    return jsonDecode(response.body)['approvalUrl'];
  }
  return null;
}
```

**FIGURE 4.21** – Fonction de création d'un order paypal

Pour récupérer les informations d'un produit scanné :

```dart
Future<Map<String, dynamic>?> getProduitInfo(String codeBarre) async {
  final response = await http.get(
    Uri.parse('http://$ipHost:3002/produit/$codeBarre'),
    headers: {'Content-Type': 'application/json'},
  );
  if (response.statusCode == 200) {
    return jsonDecode(response.body);
  }
  return null;
}
```

**FIGURE 4.22** – Fonction pour récupérer les prix des produits

Enfin, pour la communication en temps réel avec application web, l'application mobile utilise un WebSocket sécurisé, s'authentifiant via un token JWT généré en amont par l'application Web :

```dart
token = capture.barcodes.first.rawValue;
if (token != null && token!.isNotEmpty) {
   if (channel == null) {
      channel = WebSocketChannel.connect(
                Uri.parse('ws://$ipHost:8063'),);
      channel!.sink.add(jsonEncode({
         "type": "auth",
         "from": "mobile",
         "token": token,
         }));
   }
}
```

**FIGURE 4.23** – Interconnecté avec une API Socket

## 4.6 Quelques interfaces du produit final

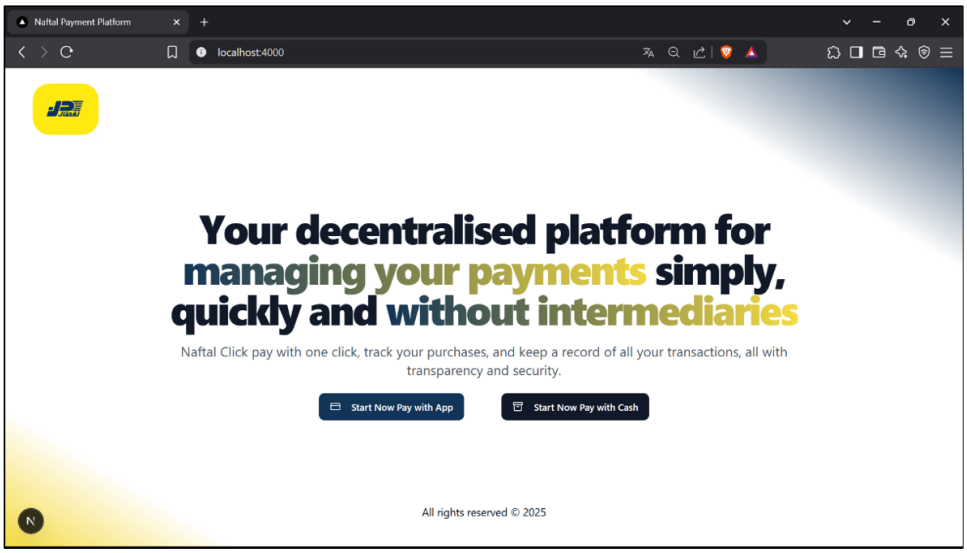


**FIGURE 4.24** – Interface de démarrage du processus de paiement

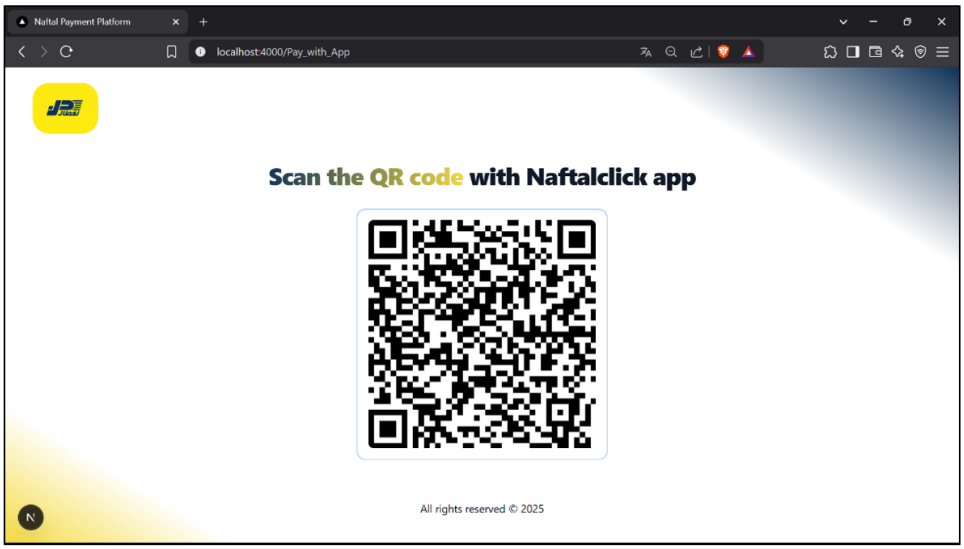


**FIGURE 4.25** – Interface d'interconnexion avec l'application Naftalclick via QR code


**FIGURE 4.26** – Démarrage du processus de paiement – côté application mobile : authentification via empreinte digitale

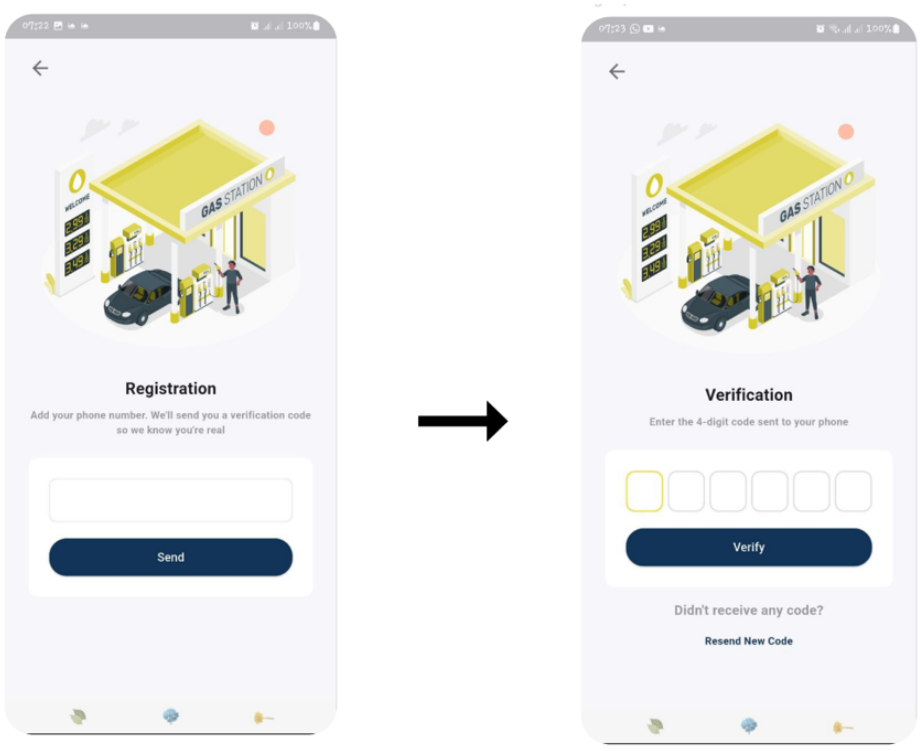


**FIGURE 4.27** – Interface d'authentification avec le numéro de téléphone et le code de vérification

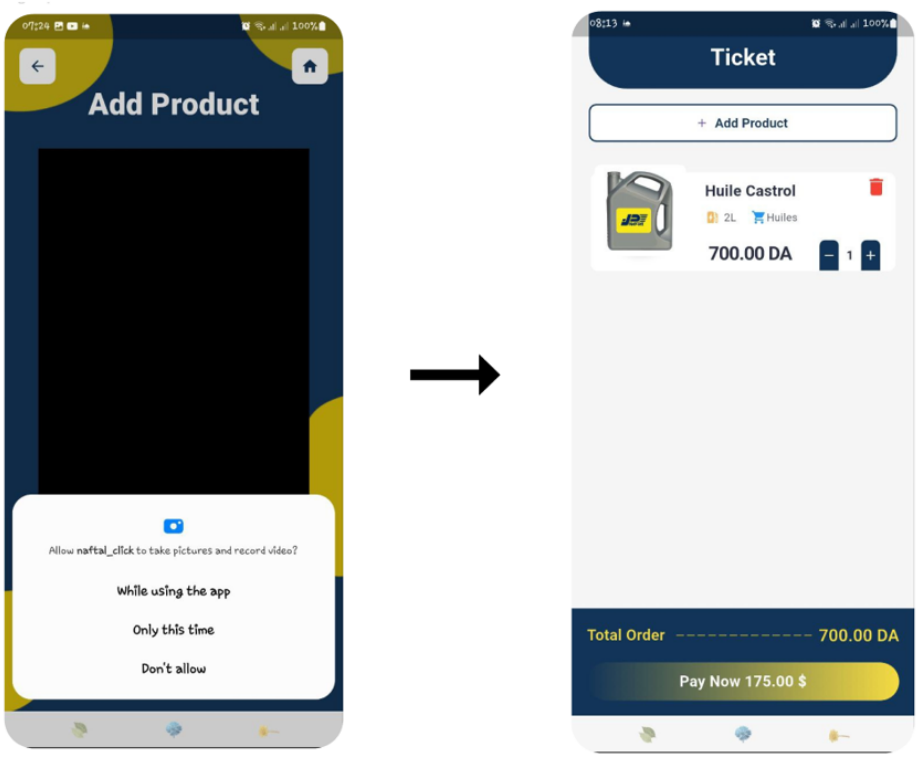


**FIGURE 4.28** – Interface d'ajout de produit et de génération du ticket de commande

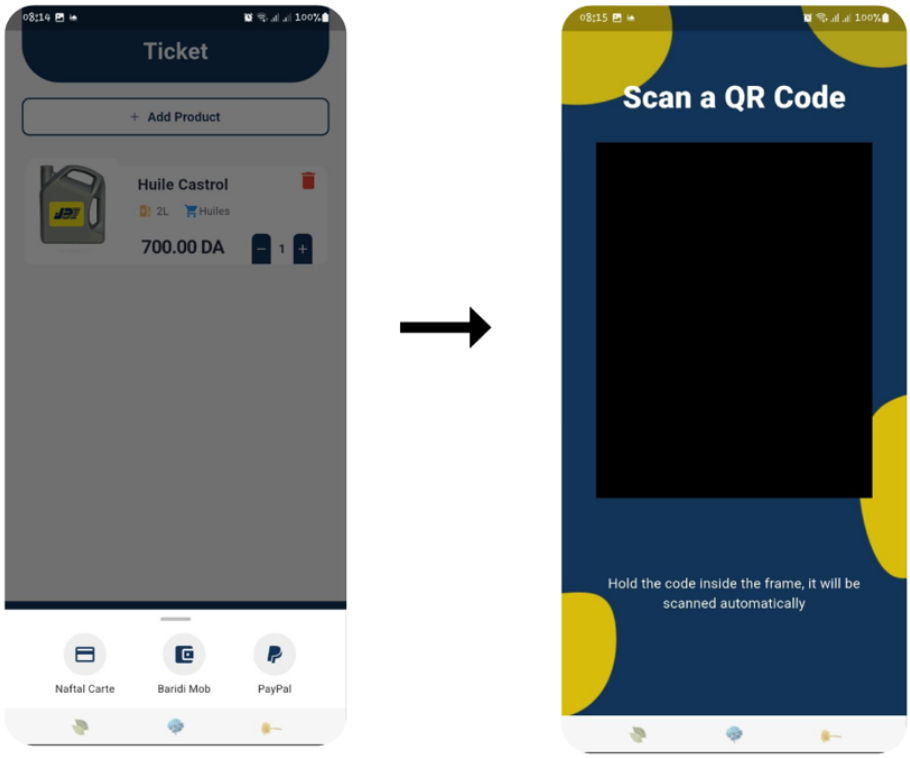


**FIGURE 4.29** – Interface de sélection du moyen de paiement et validation par QR code

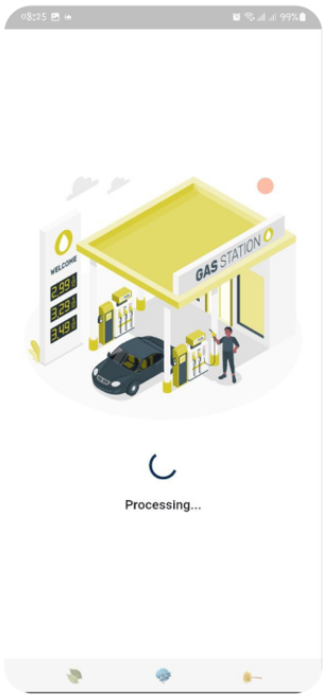


**FIGURE 4.30** – Interface de traitement de la transaction

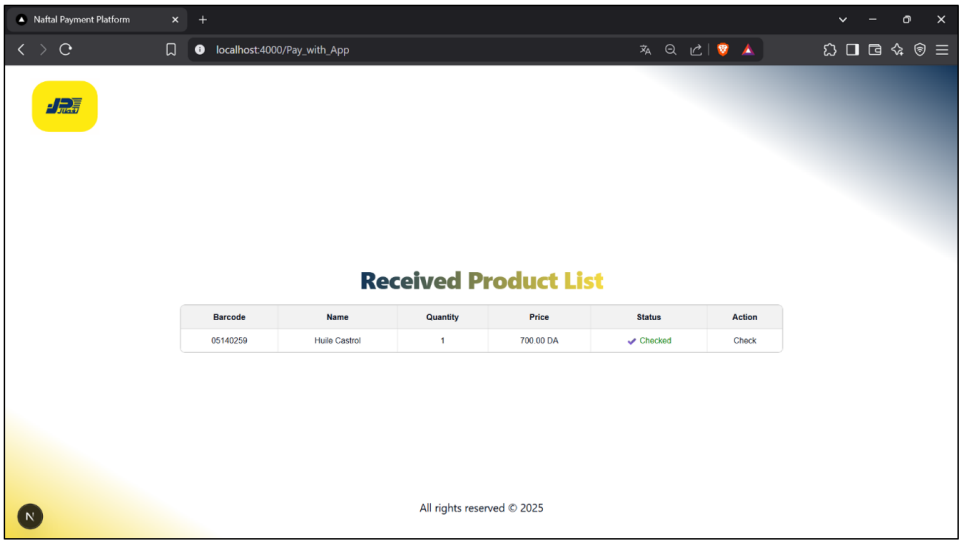


**FIGURE 4.31** – Interface de confirmation des produits sélectionnés

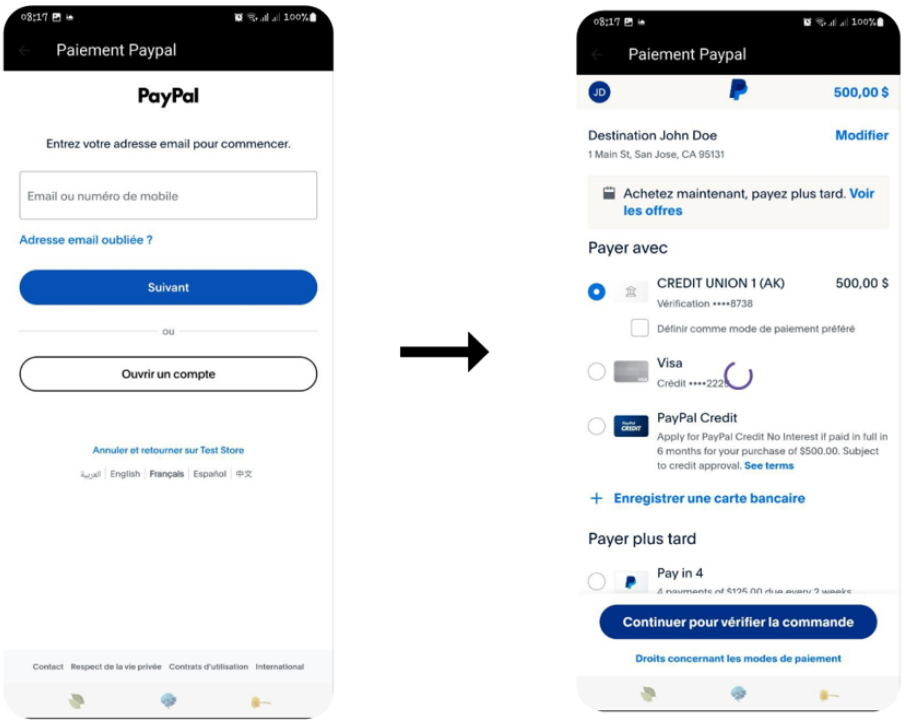


**FIGURE 4.32** – Connexion et validation du paiement via PayPal

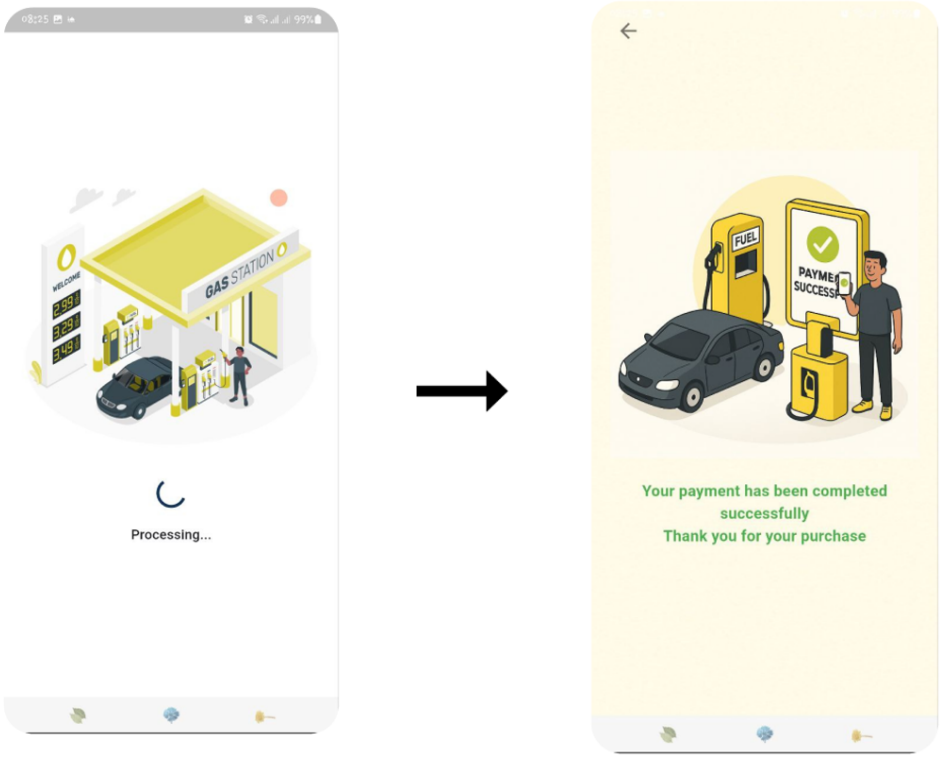


**FIGURE 4.33** – Validation finale de la transaction – paiement effectué avec succès

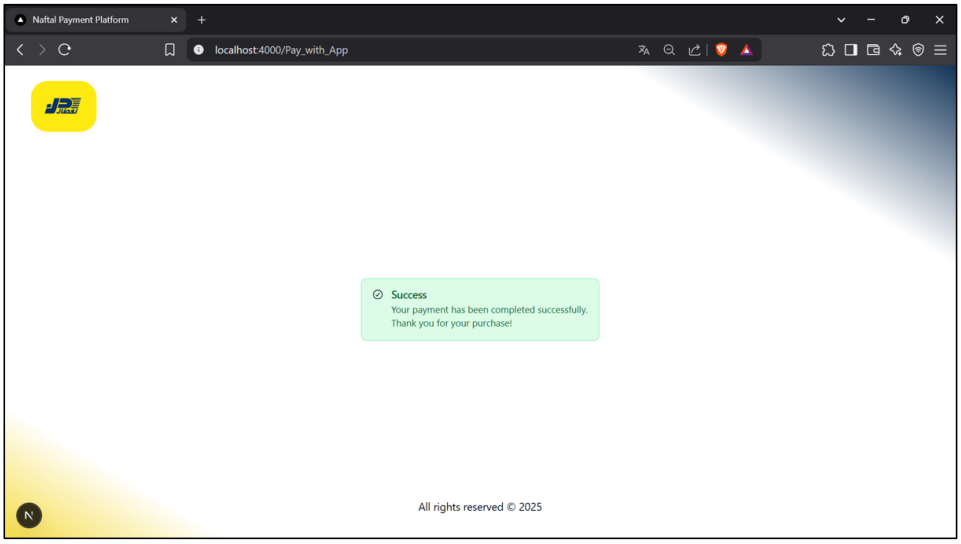


**FIGURE 4.34** – Message de succès du paiement – côté web

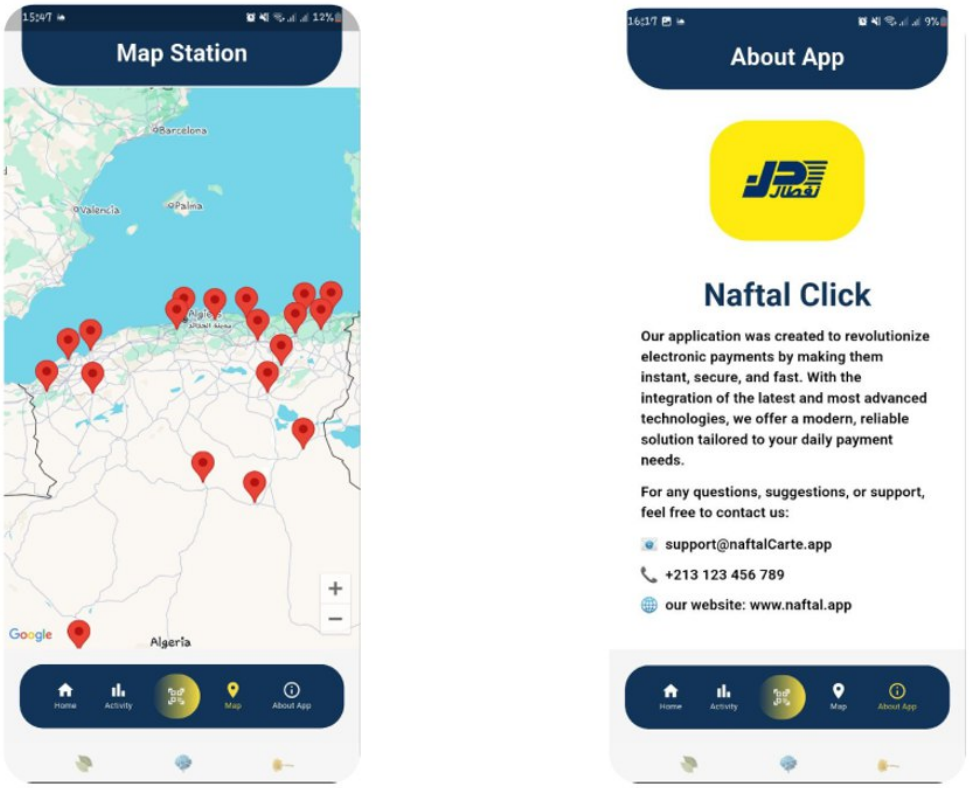


**FIGURE 4.35** – Localisation des stations-service et présentation de l'application

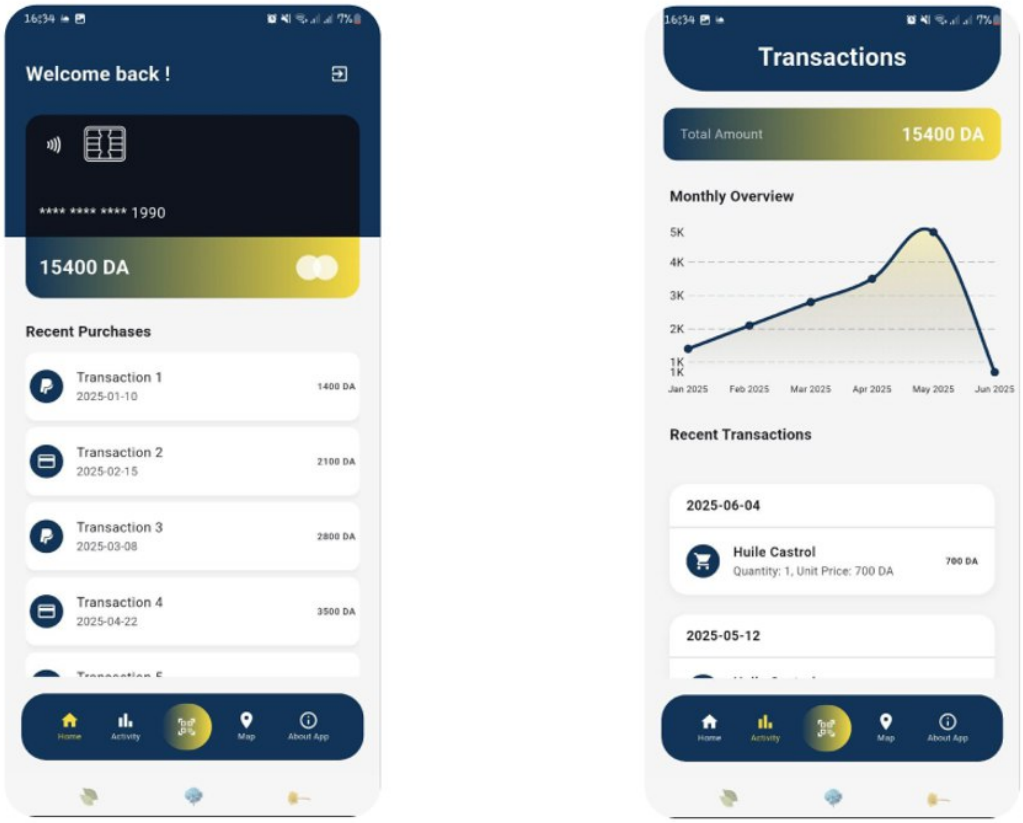


**Figure 4.36** – Vue globale des transactions et des statistiques sur les opérations effectuées par le client

## 4.7 Conclusion

L'intégration conjointe de la technologie blockchain avec l'architecture front-end a représenté une démarche complexe, nécessitant de nombreuses configurations ainsi qu'une coordination rigoureuse entre les différents composants, afin de garantir à la fois la sécurité et la performance du système. L'utilisation des microservices pour le stockage des données, la communication entre l'application web et l'application mobile, ainsi que l'intégration avec le service PayPal, permet d'assurer une architecture modulaire, évolutive et sécurisée pour le système de paiement.

Le choix de ces technologies repose sur des critères de flexibilité, de facilité d'implémentation et de performance. Chaque composant a été sélectionné de manière à garantir une intégration fluide, tout en assurant la gestion des opérations complexes et la sécurité des transactions.

# Conclusion générale

Ce projet peut être perçu comme un tremplin vers une vision plus large, celle de l'exploitation stratégique des nouvelles technologies pour transformer en profondeur la gestion des systèmes industriels et commerciaux. Aujourd'hui, l'intégration de technologies émergentes, devenues essentielles à l'échelle mondiale, telles que la blockchain, l'intelligence artificielle, l'Internet des objets (IoT) ou encore le cloud computing, peut engendrer une véritable révolution dans la gestion des ressources, des processus, et de la relation client. Ces technologies ne sont plus de simples outils de soutien, mais de véritables leviers d'optimisation, de transparence et de sécurité.

Dans le contexte algérien, leur adoption représente une opportunité majeure pour moderniser les systèmes d'information, renforcer la compétitivité des entreprises locales, et créer de nouveaux modèles économiques plus agiles et plus résilients. Par exemple, dans le secteur de l'énergie, leur utilisation permettrait d'automatiser les chaînes d'approvisionnement, de suivre en temps réel les consommations et les transactions, et de réduire considérablement les pertes ou fraudes. Dans les transports, cela ouvrirait la voie à une billetterie intelligente, à des paiements fluides, et à une meilleure coordination logistique. Dans le secteur public, cela pourrait améliorer la gestion des subventions, des aides sociales et des marchés publics, tout en garantissant une traçabilité des opérations et une lutte efficace contre la corruption.

Dans ce projet, nous proposons la mise en place d'un système de paiement en self-service basé sur la technologie blockchain. Ce système permettra aux clients des stations-service Naftal d'effectuer leurs paiements de manière autonome, rapide et sécurisée, tout en garantissant la traçabilité, l'intégrité et la confidentialité des transactions, grâce à l'utilisation du réseau Hyperledger Fabric, une blockchain privée et permissionnée.Cette solution repose sur une application mobile connectée à une plateforme web installée au sein des stations-service, facilitant l'interaction en temps réel entre le client et les équipements de la station. La solution proposée vise non seulement à moderniser le processus de vente, mais aussi à s'intégrer avec le système actuel de gestion de Naftal, grâce à une synchronisation intelligente des données. Elle est également conçue de manière évolutive pour s'adapter aux futurs besoins technologiques, notamment en ce qui concerne l'expansion des fonctionnalités, l'interconnexion avec d'autres services numériques, ou l'intégration de nouveaux

moyens de paiement. Cette approche s'inscrit dans une dynamique de transformation numérique globale, capable d'apporter des réponses concrètes aux défis économiques, sociaux et technologiques du pays.

**Perspectives :**
Ce travail ouvre la voie à plusieurs pistes de recherche et de développement futures :

- **Extension du système à d'autres entités :** La solution pourrait être généralisée à d'autres stations-service en Algérie, voire à d'autres secteurs comme la grande distribution ou le transport public, où la traçabilité et la sécurité des transactions sont également cruciales.
- **Interopérabilité avec d' autres blockchains :** Intégrer des mécanismes de communication entre Hyperledger Fabric et d'autres blockchains (publiques ou hybrides) permettrait d'élargir les usages, notamment pour la traçabilité inter-entreprises.
- **Utilisation de l'IA pour la détection d'anomalies :** En analysant les données de transaction stockées dans la blockchain, des algorithmes de machine learning pourraient détecter automatiquement des comportements suspects ou frauduleux.
- **Amélioration de l'expérience utilisateur :** L'interface client pourrait être enrichie par des fonctionnalités de fidélisation, de recommandations personnalisées ou d'intégration avec d'autres moyens de paiement (portefeuilles numériques locaux, crypto-monnaies...).
- **Évaluation économique et réglementaire :** Une étude approfondie pourrait être menée sur les impacts économiques de l'intégration de la blockchain dans le secteur pétrolier, ainsi que sur les aspects juridiques et réglementaires liés à son adoption en Algérie.

# Bibliographie


[1] J.-P. Delahaye, "La blockchain : un registre distribué infalsifiable," *Pour la Science*, pp. 86–89, october 2014.

[2] M. Pignel and D. Stokkink, "La technologie blockchain : Une opportunité pour l'économie sociale ?," *Notes d'analyse*, 2019.

[3] Chapot, "Les fonctions de contrôle face à la blockchain : risques ou opportunités," *Institut Français de l'Audit et du Contrôle Interne (IFACI)*, 2024.

[4] N. Moosavi, H. Taherdoost, N. Mohamed, M. Madanchian, Y. Farhaoui, and I. U. Khan, "Blockchain technology, structure, and applications : a survey," *Procedia Computer Science*, vol. 237, pp. 645–658, 2024.

[5] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Advances in Cryptology — CRYPTO'90*, pp. 437–455, Springer, 1991.

[6] S. Mssassi and A. Abou El Kalam, "The blockchain trilemma : A formal proof of the inherent trade-offs among decentralization, security, and scalability," *Applied Sciences*, vol. 15, no. 1, p. 19, 2024.

[7] S. Nakamoto, "Bitcoin : A peer-to-peer electronic cash system." https://bitcoin.org/bitcoin.pdf, 2008.

[8] A. Contamin de Filippis, "Comment l'utilisation de la blockchain peut accélérer la décarbonation des entreprises ?," Master's thesis, Université de Montpellier, 2023.

[9] N. Subramanian, A. Chaudhuri, and Y. Kayikci, *Blockchain and Supply Chain Logistics : Evolutionary Case Studies*. Springer Nature, 2020.

[10] N. Subramanian, A. Chaudhuri, and Y. Kayikci, "Basics of blockchain," in *Blockchain and Supply Chain Logistics : Evolutionary Case Studies*, pp. 11–19, Cham : Springer International Publishing, 2020.

[11] Y. e. a. Xiao, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys  Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[12] A. Mittal and S. Aggarwal, "Hyperparameter optimization using sustainable proof of work in blockchain," *Frontiers in Blockchain*, vol. 3, p. 23, 2020.

[13] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st Int. Convention on ICT, Electronics and Microelectronics (MIPRO)*, pp. 1545–1550, IEEE, 2018.

[14] A. Endurthi and A. Khare, "An efficient and robust proof of stake algorithm based on coin-age selection," *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 13, 2024.

[15] J. Hu and K. Liu, "Raft consensus mechanism and the applications," in *Journal of Physics : Conference Series*, vol. 1544, p. 012079, IOP Publishing, 2020.

[16] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm (extended version)," in *USENIX Annual Technical Conference*, 2014.

[17] W. Gao, W. G. Hatcher, and W. Yu, "A survey of blockchain : Techniques, applications, and challenges," in *2018 27th Int. Conf. on Computer Communication and Networks (ICCCN)*, pp. 1–11, IEEE, 2018.

[18] H. Taherdoost, "The role of blockchain in medical data sharing," *Cryptography*, vol. 7, no. 3, p. 36, 2023.

[19] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology : Architecture, consensus, and future trends," in *2017 IEEE Int. Congress on Big Data*, pp. 557–564, IEEE, 2017.

[20] I. Bashir, *Mastering Blockchain : Inner workings of blockchain, from cryptography and decentralized identities, to DeFi, NFTs and Web3*. Packt Publishing Ltd, 2023.

[21] H. Taherdoost, "Blockchain-based internet of medical things," *Applied Sciences*, vol. 13, no. 3, p. 1287, 2023.

[22] G. Wood, "Ethereum : A secure decentralised generalised transaction ledger," tech. rep., Ethereum project yellow paper, 2014.

[23] V. Buterin, "A next-generation smart contract and decentralized application platform." White paper, 3(37), 2-1, 2014.

[24] M. Morales-Sandoval, J. A. Molina, H. M. Marin-Castro, and J. L. Gonzalez-Compean, "Blockchain support for execution, monitoring and discovery of inter-organizational business processes," *PeerJ Computer Science*, vol. 7, p. e731, 2021.

[25] F. Cassez, J. Fuller, and A. Asgaonkar, "Formal verification of the ethereum 2.0 beacon chain," in *Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 167–182, Springer International Publishing, 2022.

[26] S. B. Toumia, C. Berger, and H. P. Reiser, "Evaluating blockchain application requirements and their satisfaction in hyperledger fabric," *arXiv preprint arXiv :2111.15399*, 2021.

[27] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 2018, p. 3976093, 2018.

[28] E. e. a. Androulaki, "Hyperledger fabric : a distributed operating system for permissioned blockchains," in *Proc. of the thirteenth EuroSys conference*, pp. 1–15, 2018.

[29] Y. Sun, Z. Alomari, R. Lian, and J. Lai, "A blockchain-integrated iot system leveraging hyperledger fabric," 2025.

[30] F. e. a. Pelekoudas-Oikonomou, "Prototyping a hyperledger fabric-based security architecture for iomt-based health monitoring systems," *Future Internet*, vol. 15, no. 9, p. 308, 2023.

[31] Hyperledger Fabric Documentation, "Channels." https://hyperledger-fabric.readthedocs.io/en/release-2.4/whatis.html#channels, 2023. Consulté le : 9 mai 2025.

[32] J. Dzikowski, "Hyperledger fabric cheat sheet." https://softwaremill.com/hyperledger-fabric-cheat-sheet/, 2023.

[33] E. e. a. Zhou, "Ledgerdata refiner : a powerful ledger data query platform for hyperledger fabric," in *2019 6th Int. Conf. on IoT : Systems, Management and Security (IOTSMS)*, pp. 433–440, IEEE, 2019.

[34] Hyperledger Fabric CA Documentation, "User's guide." https://hyperledger-fabric-ca.readthedocs.io/en/latest/users-guide.html, 2023. Consulté le : 09 mai 2025.

[35] S. Ibeghouchene and A. Mokrane, "Adoption et utilisation des paiements électroniques en algérie : étude empirique de la période (2016-2022)," *Finance and Business Economies Review*, vol. 6, no. 3, pp. 553–564, 2022.

[36] N. E. Lamine and M. Cherchem, "L'e-paiement en algérie : état des lieux et perspectives de développement," vol. 27, no. 3, pp. 175–191, 2024.

[37] Giemonetique, Service Communication, "Activité paiement sur internet 2025." Site institutionnel de Giemonetique. [En ligne]. Disponible sur : https://giemonetique.dz/qui-sommes-nous/activite-paiement-sur-internet. Consulté le : 4 mai 2025.

[38] APS, Rédaction APS, "Plus de 13 millions d'opérations effectuées via baridimob les 5 premiers mois de 2023." Journal en ligne APS (Algérie Presse Service). [En ligne]. Disponible sur : https://www.aps.dz/sante-science-technologie/157520-plus-de-13-millions-d-operations-effectuees-via-baridimob-les-5-premiers-m Consulté le : 4 mai 2025.

[39] El Watan, Rédaction El Watan, "Djezzy – bdl : Lancement du e-paiement par la carte visa." El Watan, 20 mai 2025. [En ligne].Disponible sur : https://elwatan-dz.com/

djezzy-bdl-lancement-du-e-paiement-par-la-carte-visa. Consulté le : 4 mai 2025.

[40] NAFTAL, "Présentation interne de l'organisme naftal." Document interne non publié, 2025.

[41] N. Subramanian, A. Chaudhuri, and Y. Kayikci, *Blockchain and Supply Chain Logistics : Evolutionary Case Studies*. Springer Nature, 2020.

[42] F. e. a. Yuan, "The evolution and optimization strategies of a pbft consensus algorithm for consortium blockchains," *Information*, vol. 16, no. 4, p. 268, 2025.

[43] Y. Sun, Z. Alomari, R. Lian, and J. Lai, "A blockchain-integrated iot system leveraging hyperledger fabric," 2025.

[44] F. Pelekoudas-Oikonomou, J. C. Ribeiro, G. Mantas, G. Sakellari, and J. Gonzalez, "Prototyping a hyperledger fabric-based security architecture for iomt-based health monitoring systems," *Future Internet*, vol. 15, no. 9, p. 308, 2023.

[45] Docker Documentation Team, "Docker engine 24.0 release notes." https://docs.docker.com/engine/release-notes/24.0/, 2025. Consulter la documentation officielle de Docker version 24.0 consulter le 30 mai 2025.

[46] Kubernetes Release Team, "Kubernetes v1.27 release notes." https://kubernetes.io/blog/2023/04/11/kubernetes-v1-27-release/, 2023. Documentation officielle v1.27 consulter le 30 mai 2025.

[47] Hyperledger Fabric Documentation, "What's new in hyperledger fabric v2.5." https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatsnew.html, 2025. Documentation officielle v3.0 consulter le 30 mai 2025.

[48] The Go Authors, "The go programming language." https://golang.org/, 2025. Site officiel de Go (Golang) consulter le 30 mai 2025.

[49] Node.js Documentation, "Node.js v22.x api documentation." https://nodejs.org/docs/latest-v22.x/api/index.html, 2024. Documentation API officielle v22.x LTS consulter le 30 mai 2025.

[50] Redis Documentation, "Redis docs." https://redis.io/docs/latest/, 2025. Documentation officielle Redis consulter le 30 mai 2025.

[51] MySQL Developer Zone, "Mysql reference manual (v8.0)." https://dev.mysql.com/doc/refman/8.0/en/, 2025. Documentation officielle MySQL 8.0 consulter le 30 mai 2025.

[52] Flutter Documentation, "Flutter : Build apps for any screen." https://flutter.dev/, 2025. Site officiel de Flutter consulter le 30 mai 2025.

[53] SQLite Documentation, "Sqlite documentation." https://www.sqlite.org/docs.html, 2025. Site officiel de SQLite consulter le 30 mai 2025.

[54] Firebase Documentation, “Firebase documentation.” https://firebase.google.com/docs, 2025. Documentation officielle Firebase consulter le 30 mai 2025.