

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'enseignement supérieur et de la recherche scientifique



Centre Universitaire

Abdelhafid Boussouf - Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

**Mémoire préparé en vue de l'obtention du diplôme de Master**

**En : Informatique**

**Spécialité : Sciences et Technologies de l'Information et de la  
Communication (STIC)**

**Thème : Conception et Réalisation d'un  
Système d'Information Sécurisé basé sur la  
Blockchain**

**Soutenu devant le jury :**

**Préparé par :**

Boumeshal Yahya  
Meskine Mourad

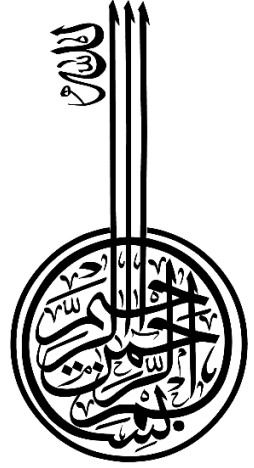
**Président :** Merabet Adil MAA

**Examineur :** Boukhechem Nadir MAA

**Rapporteur :** Meghzili Said MCB

**An**

née Universitaire : 2021/2022



---

# Remerciements



Au début de ce travail, nous remercions dieu de nous avoir accordé le courage, la volonte et la patience pour parvenir à achever ce modeste travail.

Par la suite, nous voudrions exprimer notre profonde gratitude à notre encadreur Meghzili Said, qui nous a fait l'honneur de diriger ce travail, pour ses conseils et ses orientations.

Sans oublier les membres du jury d'avoir accepté d'examiner et dévaluer notre travail. Nous remercions également les enseignants qui ont participé à notre progrès pendant ces 5 ans.

Enfin, nous adressons nos remerciements à toutes les personnes qui, de près ou de loin, nous ont soutenus tout au long de notre parcours.

---

## Dédicaces



Nous dédions cet acte humble :

À toutes nos familles et à tous nos proches, en particulier nos parents, qui nous ont apporté leur soutien et leur amour inconditionnel,

A nos sœurs et nos frères

A toute personnes qui ont attendu l'achèvement de ce mémoire et qui ont prié dieu pour plus de réussite.

## ملخص

تعتبر رقمنة الإدارة، أو التحول الرقمي للدولة، محورًا مركزيًا لبرنامج الحكومة الجزائرية الهادف إلى تحسين أداء الإدارة العامة ومكافحة البيروقراطية.

تساهم مشاريع التنمية المحلية مثل إنشاء شبكات الصرف الصحي وإنشاء البنية التحتية والجسور والطرق والأنفاق وغيرها، في تحسين نوعية حياة المواطنين وكذلك في تنمية بلدنا وتنقسم هذه المشاريع إلى عدة قطاعات، مثل: النقل والزراعة والأشغال العامة والصحة والتعليم ... في هذا السياق، يهدف هذا المشروع إلى تصميم نظام معلومات آمن يتيح مراقبة الوضع المادي والمالي لمشاريع البناء والتنمية المحلية. وقد قمنا بتطوير هذا النظام باستخدام إطار تطوير الويب (الواجهة الأمامية) وسلسلة الكتل (الواجهة الخلفية).

هذا المشروع، من ناحية، سوف يحسن الشفافية طوال دورة حياة المشروع ومن ناحية أخرى لتوفير قاعدة بيانات وإنشاء إحصائيات تسمح للمدراء بالتحكم والإشراف على جميع العمليات وكذلك لمساعدتهم على اتخاذ القرارات المناسبة.

## Résumé

La numérisation de l'administration, ou la transformation numérique de l'État, est un axe central du programme du gouvernement algérien visant à améliorer la performance de l'administration publique et à lutter contre la bureaucratie.

Les projets de développement local tels que la construction de réseaux d'égouts, la construction d'infrastructures, de ponts, de routes, de tunnels, etc., contribuent à l'amélioration de la qualité de vie des citoyens ainsi qu'au développement de notre pays. Ces projets se déclinent en plusieurs secteurs, tels que : transports, agriculture, travaux publics, santé et éducation... Dans ce cadre, ce projet vise à concevoir un système d'information sécurisé permettant de suivre la situation physique et financière des constructions et des projets de développements. Nous avons développé ce système en utilisant un *cadre de développement Web (front-end) et la Blockchain (back-end)*.

Ce projet permettra d'une part d'améliorer la transparence tout au long du cycle de vie du projet et d'autre part de fournir une base de données et de créer des statistiques qui permettront aux gestionnaires de contrôler et de superviser tous les processus et aussi de les aider à prendre les décisions appropriées.

Mots clés : Blockchain, Ethereum

## **Abstract**

The digitization of the administration, or the digital transformation of the state, is a central focus of the Algerian government's program aimed at improving the performance of public administration and combating bureaucracy.

Local development projects such as constructing sewage networks, constructing infrastructure, bridges, roads, tunnels, etc., contribute to improving the quality of life of citizens as well as to the development of our country. These projects are divided into several sectors, such as: transportation, agriculture, public works, health and education... In this context, this project aims to design a secure information system that allows monitoring of the physical and financial situation of construction and local development projects. We have developed this system using web development framework (front-end) and Blockchain (back-end).

This project, on the one hand, will improve transparency throughout the project life cycle and on the other hand to provide a database and create statistics that will allow managers to control and supervise all processes and also to help them take appropriate decisions.

Keywords: Blockchain, Ethereum

---

# Table des matières



Remerciements.....	i
Dédicaces .....	ii
Résumé .....	iii
ملخص.....	iii
Abstract .....	iv
Table des matières.....	v
Table des figures.....	ix
Liste des tableaux .....	xi
Introduction générale.....	1
1. Présentation du domaine d'étude.....	3
1.1. Introduction .....	3
1.2. Présentation de la commune de Sidi Merouane.....	3
1.2.1. Apparition .....	3
1.2.2. Définition et localisation .....	3
1.2.3. Organigramme de la commune de Sidi Merouane.....	6
1.2.4. Détermination du champ d'étude .....	6
1.2.5. moyen humain et matériel .....	7

1.3.	Etude de l'existant .....	10
1.3.1.	Cycle de vie d'un projet au sein de la commune .....	10
1.3.2.	Problématique.....	11
1.3.3.	Objectifs du projet.....	12
1.4.	Conclusion .....	12
2.	Blockchain .....	13
2.1.	Introduction .....	13
2.2.	Définition du terme Blockchain.....	13
2.3.	Historique de la Blockchain.....	14
2.4.	Fonctionnement d'une Blockchain .....	15
2.5.	Concepts de la Blockchain .....	17
2.5.1.	Structure du réseau :.....	17
2.5.2.	Blocs :.....	17
2.5.3.	Transaction :.....	18
2.5.4.	Contrat intelligent : .....	18
2.5.5.	Consensus :.....	18
2.6.	Propriété d'immutabilité de la Blockchain .....	20
2.7.	Types de la Blockchain.....	21
2.7.1.	Publique.....	21
2.7.2.	Privée .....	21
2.7.3.	Consortium (hybride).....	21
2.8.	Les plateformes Blockchain.....	22
2.8.1.	Bitcoin.....	22
2.8.2.	Ethereum.....	23
2.8.3.	Hyperledger Fabric .....	23
2.8.4.	Litecoin .....	24
2.8.5.	Ripple.....	24
2.9.	Les avantages de la technologie.....	25
2.9.1.	La désintermédiation .....	25
2.9.2.	La transparence .....	25
2.9.3.	La sécurité .....	26
2.9.4.	L'autonomie .....	26
2.10.	Les défis de la technologie.....	26
2.10.1.	Consommation d'énergie .....	26



2.10.2. Coût.....	27
2.11. Les applications de la Blockchain .....	27
2.11.1. La certification des droits d’auteur .....	27
2.11.2. Améliore la transparence de chaînes d’approvisionnement.....	27
2.11.3. L’identité numérique .....	27
2.11.4. Les systèmes de vote électronique basé sur la Blockchain.....	28
2.12. Conclusion .....	28
3. Analyse et conception .....	29
3.1. Introduction .....	29
3.2. Le choix de la méthodologie.....	29
3.2.1. Rappel sur le processus unifié .....	29
3.2.2. Rappel sur le langage UML .....	32
3.2. Identification des besoins .....	33
3.2.1. Identification des besoins fonctionnels .....	33
3.2.1.1. Identification des acteurs.....	33
3.2.3. Identification des besoins non fonctionnelles.....	34
3.3. Diagramme de cas d’utilisation .....	35
3.4. Description textuelle .....	35
3.5. Développements du modèle dynamique .....	51
3.6. Conception.....	60
3.6.1. Diagramme de déploiement .....	60
3.6.2. Conception du diagramme de classes .....	61
3.6.4. Passage vers le modèle relationnel .....	63
3.7. Conclusion .....	64
4. Implémentation du système .....	65
4.1. Introduction .....	65
4.2. Outils de développement et langages utilisés .....	65
4.3. Description du système.....	70
4.3.1. Architecture de l’application web .....	70
4.3.2. Contrat intelligent et migration .....	71
4.3.3. Front-end Angular et web3.js : .....	74
4.4. Conclusion .....	86
Conclusion générale .....	87

<b>Bibliographie.....</b>	<b>89</b>
<b>Acronymes.....</b>	<b>92</b>

---

## Table des figures



Figure 1.1 Organigramme de la commune de Sidi Merouane.....	6
Figure 1.2 Organigramme du service des marchés et des programmes .....	7
Figure 1.3 Réseaux informatique.....	9
Figure 1.4 Cycle de vie d'un projet au sein de la commune de Sidi Merouane .....	10
Figure 2.1 Architecture de la Blockchain.....	14
Figure 2.2 Chronologie de la Blockchain.....	15
Figure 2.3 Principe général d'une Blockchain [10] .....	16
Figure 2.4 Structure d'un block.....	18
Figure 2.5 Tentative de modification d'un ancien block [11].....	21
Figure 2.6 Types de la Blockchain.....	22
Figure 2.7 Logo du protocole <i>Bitcoin</i> .....	22
Figure 2.8 Logo du protocole Ethereum [14].....	23
Figure 2.9 Logo du Hyperledger Fabric [15] .....	23
Figure 2.10 Logo du Litecoin [16] .....	24
Figure 2.11 Logo du Ripple [17].....	24
Figure 2.12 Plateformes Blockchain .....	25
Figure 2.13 Les applications de la Blockchain[23].....	28
Figure 3.1 Déroulement du processus unifié.....	30
Figure 3.2 Déroulement du processus unifié.....	31
Figure 3.3 Diagramme de cas d'utilisation .....	35
Figure 3.4 Diagramme d'activité de cas « Authentifier » .....	37
Figure 3.5 Diagramme d'activité de cas « Ajouter Projet » .....	39
Figure 3.6 Diagramme d'activité de cas « Consulter Liste des Projet » .....	40
Figure 3.7 Diagramme d'activité « Supprimer Projet » .....	42
Figure 3.8 Diagramme d'activité de cas « Modifier Projet » .....	44
Figure 3.9 Diagramme d'activité du cas « Changement Situation Projet » .....	46
Figure 3.10 Diagramme d'activité du cas « Consulter Projet » .....	47
Figure 3.11 Diagramme d'activité de cas « Rechercher Projet » .....	49
Figure 3.12 Diagramme d'activité du cas « Consulter Statistiques Projet » .....	50
Figure 3.13 Diagramme d'interaction de cas « authentifier » .....	51

Figure 3.14 Diagramme d'interaction de cas « ajouter projet ».....	52
Figure 3.15 Diagramme d'interaction de cas « Modifier projet ».....	53
Figure 3.16 Diagramme d'interaction de cas « Change Situation projet ».....	54
Figure 3.17 Diagramme d'interaction de cas « Supprimer projet ».....	55
Figure 3.18 Diagramme d'interaction de cas « Consulter projet ».....	56
Figure 3.19 Diagramme d'interaction de cas « Consulter Statistique ».....	57
Figure 3.20 Diagramme d'interaction de cas « Consulter Liste Projet ».....	58
Figure 3.21 Diagramme d'interaction de cas « Recherche Projet ».....	59
Figure 3.22 Diagramme de déploiement.....	60
Figure 3.23 Diagramme de classes.....	63
Figure 4.1 Logo du Protocole Ethereum.....	65
Figure 4.2 Logo de langage Solidity.....	66
Figure 4.3 Logo de l'API Web3.js.....	66
Figure 4.4 Logo du Framework Truffle.....	67
Figure 4.5 Logo de logiciel Ganache.....	67
Figure 4.6 Logo du logiciel Metamask.....	68
Figure 4.7 Logo de Framework Angular.....	68
Figure 4.8 Logo de Framework Node.js.....	69
Figure 4.9 Logo du logiciel Xampp.....	69
Figure 4.10 Logo du logiciel WebStorm.....	69
Figure 4.11 Logo du gestionnaire des packages NPM.....	70
Figure 4.12 Logo de l'ide des EdrawMax.....	70
Figure 4.13 Architecture de l'application.....	71
Figure 4.14 Génération des comptes Ethereum sur la Blockchain personnelle.....	81
Figure 4.15 Déploiement des contrats intelligents sur la Blockchain.....	82
Figure 4.16 Espace d'authentification.....	83
Figure 4.17 Interface Administrateur.....	84
Figure 4.18 Interface permettant d'ajouter des projets.....	85
Figure 4.19 Enregistrement des données dans les blocs après le minage.....	85
Figure 4.20 Interface Consulter statistiques.....	86

---

# Liste des tableaux



Tableau 1 Ressources humaines de service des marches et des programmes .....	7
Tableau 2 les micro-ordinateurs de service des marches et des programmes.....	8
Tableau 3 les imprimantes de service des marches et des programmes.....	8
Tableau 4 les serveurs de service des marchés et des programmes .....	8
Tableau 5 Modalisation du contexte .....	33
Tableau 6 Cas d'utilisation Authentification .....	36
Tableau 7 Cas d'utilisation « Ajouter projet ». .....	38
Tableau 8 Cas d'utilisation « Consulter Liste projet ». .....	40
Tableau 9 Cas d'utilisation Supprimer projet. ....	41
Tableau 10 Cas d'utilisation modifier projet.....	43
Tableau 11 Cas d'utilisation change Situation projet.....	45
Tableau 12 Cas d'utilisation Consulter un projet.....	47
Tableau 13 Cas d'utilisation Recherche projet .....	48
Tableau 14 Cas d'utilisation consulter statistiques d'un projet.....	50
Tableau 15 Description des classes .....	61
Tableau 16 les operations.....	62
Tableau 17 Modèle relationnel.....	64
Tableau 18 Partie1 du contrat intelligent Projet.sol en Solidity.....	72
Tableau 19 Partie2 du contrat intelligent Projet.sol en Solidity.....	73
Tableau 20 Fonction login .....	74
Tableau 21 Code permettant l'initialisation de la librairie Web3.js et l'injection de MetaMask .	76
Tableau 22 code permettant de retourner le contrat projet.sol.....	77
Tableau 23 code permettant d'ajouter un projet à la Blockchain.....	77
Tableau 24 code permettant d'initialiser le Framework express et de créer une connexion avec MySQL.....	79
Tableau 25 Les méthodes HTTP utilisées dans API REST .....	80

---

# Introduction générale



Dans ces dernières années, l'état algérien a commencé un processus de modernisation au niveau des communes notamment la numérisation des documents d'état civile, les plateformes biométriques, les systèmes d'information du suivi de la situation socio-économique... etc. Cependant, l'administration de la commune souffre de manque d'un logiciel sécurisé qui permet la gestion des projets de développement local tels que la réalisation d'éclairage public, la réalisation d'une école primaire, l'aménagement d'un chemin d'une ville... etc.

Plusieurs travaux ont attaqué le problème de la gestion des projets dans différentes entreprises. Dans [1], les auteurs proposent une application client/serveur pour le suivi des projets fibres optiques au sein de la Direction Opérationnelle de Télécommunication de la wilaya de Mila. Dans [2], les auteurs proposent une application client/serveur pour la gestion des programmes propres à l'entreprise SONELGAZ. Dans [3], les auteurs conçoivent une application client/serveur pour le suivi des projets au sein de la Direction des Travaux Publics. Dans [4], les auteurs développent une plateforme pour la gestion et le suivi des projets des entreprises. Cependant, ces travaux n'ont pas pris en compte le problème de sécurité qui garantit l'authentification, la confidentialité et l'intégrité des données. De ce fait, notre contribution principale est la sécurisation de la gestion des projets en utilisant la technologie Blockchain.

La Blockchain est une technologie émergente basé sur une fondation cryptographique solide comme Ethereum et Hyperledger. Plus précisément, la Blockchain est un registre distribué et décentralisé, permettant de stocker et d'échanger des informations de manière sécurisée, sans un tiers de confiance.

Dans ce travail, nous proposons un système d'information sécurisé pour la gestion des projets de développement locale basé sur la technologie Blockchain au sein de la commune de Sidi Merouane, le lieu de notre stage pratique. L'objectif de ce travail est le suivi de la situation physique et financière des projets de développement local. Ce système permettra d'une part d'améliorer la transparence tout au long du cycle de vie d'un projet et d'autre part de fournir une base de données et de générer des statistiques qui offrent aux responsables de la commune la possibilité de contrôler et de superviser toutes les opérations et les aident également à prendre les décisions appropriées.

Ce mémoire est organisé comme suit :

Dans le premier chapitre, nous présentons la commune de sidi merouane de la wilaya de Mila où nous avons effectué notre stage et nous expliquons les notions fondamentales liées au cycle de vie d'un projet.

Le deuxième chapitre présente en détail les concepts fondamentaux et les caractéristiques générales de la technologie Blockchain, en mettant l'accent sur Bitcoin et Ethereum.

Le troisième chapitre est entièrement consacré à l'analyse et la conception détaillée du système. Il contient l'ensemble des diagrammes nécessaires à la bonne compréhension et cohésion du système, du diagramme de cas d'utilisation au diagramme de classe. Le tout est accompagné des scénarios et descriptions nécessaires.

Le quatrième chapitre présente le outils, les langages de programmation et l'environnement de développement que nous avons utilisé expose quelques interfaces de l'application.

---

# 1. Présentation du domaine d'étude

## 1.1. Introduction

Le suivi des projets est une tâche importante dans les communes et nécessite une base de données cohérente, fiable et mise à jour en temps réel pour fournir à tout décideur local les informations nécessaires sur la situation physique et financière des projets de développement local.

Dans ce chapitre, premièrement, nous présentons la commune de Sidi Merouane de la wilaya de Mila et en particulier le service des marchés et des programmes. Deuxièmement, nous présentons le cycle de vie d'un projet au sein de la commune de Sidi Merouane, les problèmes et notre solution.

## 1.2. Présentation de la commune de Sidi Merouane

### 1.2.1. Apparition

La mairie de Sidi Merouane remonte à l'année de 1874. En 1880, elle a été transférée au rang de commune selon le décret 23 de 1880. Après l'indépendance, et en 1963, elle a été annexée à la commune de Grarem-Gouga mais elle était considérée comme un groupe de population secondaire. En 1984, en vertu du décret pris par le Comité Central de Réorganisation Territoriale du Territoire National, Sidi Merouane est devenue une commune avec tous ses pouvoirs administratifs, rattachée à la Daïra de Grarem-Gouga dans la wilaya de Mila.

### 1.2.2. Définition et localisation

La région de Sidi Merouane est située dans la province orientale de l'Algérie, dans la wilaya de Mila. Elle est bordée au nord-ouest par la commune de Chigara, à l'est et au nord-est par la commune de Grarem-Gouga, et au sud-ouest par la commune de Zeghaia. La superficie de



Sidi Merouane est d'environ de 33,27 km<sup>2</sup>. Sa population, selon les dernières statistiques, est de 50088 habitants.

La commune de Sidi Merouane a connu plusieurs chefs de maire depuis sa création. Elle est composée de plus de 27 quartiers.

La commune de Sidi Merouane constituée des dix directions, chaque direction contient plusieurs bureaux et chaque bureau a plusieurs branches :

- **Service du secrétariat général**
  - ✓ Bureau du président du conseil populaire municipal
  - ✓ Bureau du Secrétaire général
  - ✓ Bureau de la documentation et des archives
  - ✓ Bureau de l'information et des statistiques
  
- **Service d'organisation et des affaires publiques**
  - ✓ Bureau des élections et du service national
  - ✓ Bureau de l'organisation et des affaires publiques
  - ✓ Bureau des litiges et des affaires juridiques
  
- **Service d'état civil**
  - ✓ Bureau d'enregistrement des contrats
  - ✓ Bureau des actes d'état civil
  
- **Administration et intérêt public**
  - ✓ Bureau de gestion et de formation des utilisateurs
  - ✓ Bureau du budget et de la comptabilité
  - ✓ Bureau de paye
  - ✓ Bureau de la propriété
  
- **Service des marches et des programmes**
  - ✓ Bureau de préparation des transactions
  - ✓ Bureau de suivi technique des projets
  - ✓ Bureau des programmes divers et du développement

- **Service d'affaires sociales**
  - ✓ Bureau des affaires sportives
  - ✓ Bureau des affaires sociales
  
- **Service de la construction et du bâtiment**
  - ✓ Bureau de construction et de construction
  - ✓ Bureau de la route
  
- **Service de grange et d'entrepôt**
  - ✓ Bureau de gestion de la grange
  - ✓ Bureau de gestion d'entrepôt
  - ✓ Bureau d'entretien
  
- **Service de l'environnement et de l'hygiène publique**
  - ✓ Bureau de la protection de l'environnement
  - ✓ Bureau d'hygiène publique
  - ✓ Bureau des activités de quartier et des campagnes de sensibilisation
  
- **Service de l'inspection, de la prévention et de la santé publique**
  - ✓ Bureau de la santé publique
  - ✓ Bureau d'analyses de laboratoire
  - ✓ Bureau de contrôle de l'eau
  - ✓ Bureau de contrôle et d'inspection de la qualité

### 1.2.3. Organigramme de la commune de Sidi Merouane

La figure ci-dessous présentons l'organigramme de la commune de Sidi Merouane.

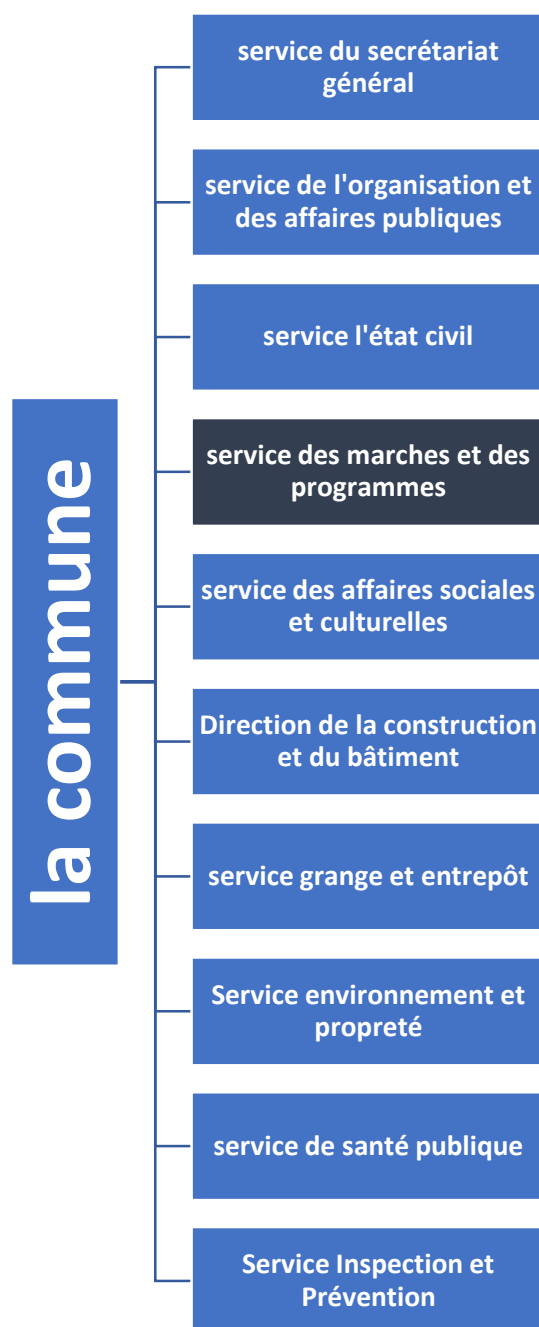


Figure 1.1 Organigramme de la commune de Sidi Merouane

### 1.2.4. Détermination du champ d'étude

➤ **Service des marches et des programmes :**

Le service des marchés et des programmes contient trois bureaux, un bureau de préparation des transactions, un bureau technique de contrôle des projets, un bureau des programmes divers et de développement (figure 1.2).

Ce service supervise les projets de développement local tels que la réalisation des réseaux d'assainissement, la construction d'infrastructures, de ponts, de routes, de tunnels... etc.

Ces projets se répartissent en plusieurs secteurs, tels que : le transport, l'agriculture, les travaux publics, la santé et l'éducation...

➤ **Organigramme du service des marchés et des programmes :**

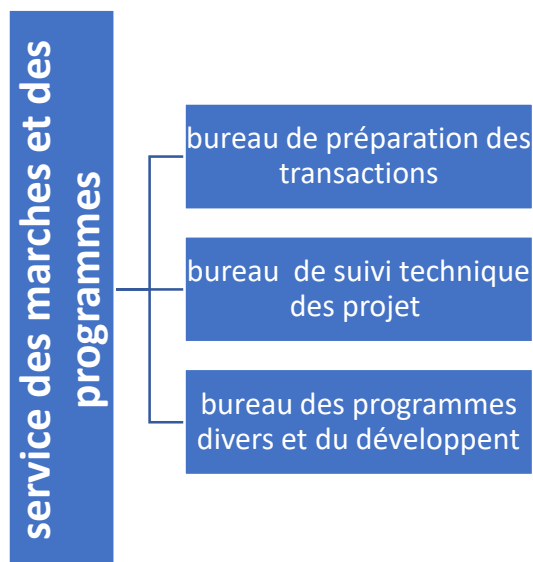


Figure 1.2 Organigramme du service des marchés et des programmes

**1.2.5. moyen humain et matériel**

A travers notre étude au niveau du service des marchés et des programmes de la commune de Sidi Merouane, nous avons recensé les ressources humaines et matérielles disponibles.

**✚ Moyens humains**

➤ **Au niveau de service des marchés et des programmes :**

Ce tableau représenter les ressources humaines de service des marchés et des programmes de la commune de Sidi Merouane (année 2022).

Nombre d'employé	Diplôme	Fonction actuelle et tache assumées
Employé _1	Ingénieur d'état	Chef de service
Employé _2	Technicien supérieure	Informaticien
Employé _3	Technicien supérieure	gestion et économie
Employé _4	Technicien supérieure	Droit et administration
Employé _5	Technicien	Administration
Employé _6	Technicien	Administration
Employé _7	Technicien	Administration

Tableau 1 Ressources humaines de service des marchés et des programmes

## Moyens matériels

### ➤ Au niveau de service des marches et des programmes :

- Espace de travail
  - ✓ 01 salle 20 m<sup>2</sup>
  - ✓ 02 salle 35 m<sup>2</sup>
  - ✓ 03 salle 40 m<sup>2</sup>
- **Micro- ordinateurs**

Modèle	HP	pentium
Pressureur	Cor i3	2.71 GHz
RAM	4 Go	2 Go
Disque dure	1T	500 Go
Système d'exploitation	Windows	Windows
Quantité	3	4

Tableau 2 les micro-ordinateurs de service des marches et des programmes.

- **Les imprimantes**

Marque	Quantité
Canon	2
Hpdeskjet 1050	1

Tableau 3 les imprimantes de service des marches et des programmes.

- **Les logiciels utilisés**

- ✓ Microsoft Word 2010
- ✓ Microsoft Excel 2010

### ➤ Au niveau de la commune de Sidi Merouane.

- Serveur

Marque	HpProliant ML350
Capacité	Intel (R) XEM(R) 2Go et 2RAM
Ressource	3 disques dur 146 Go
Quantité	3

Tableau 4 les serveurs de service des marchés et des programmes

- Réseaux informatiques

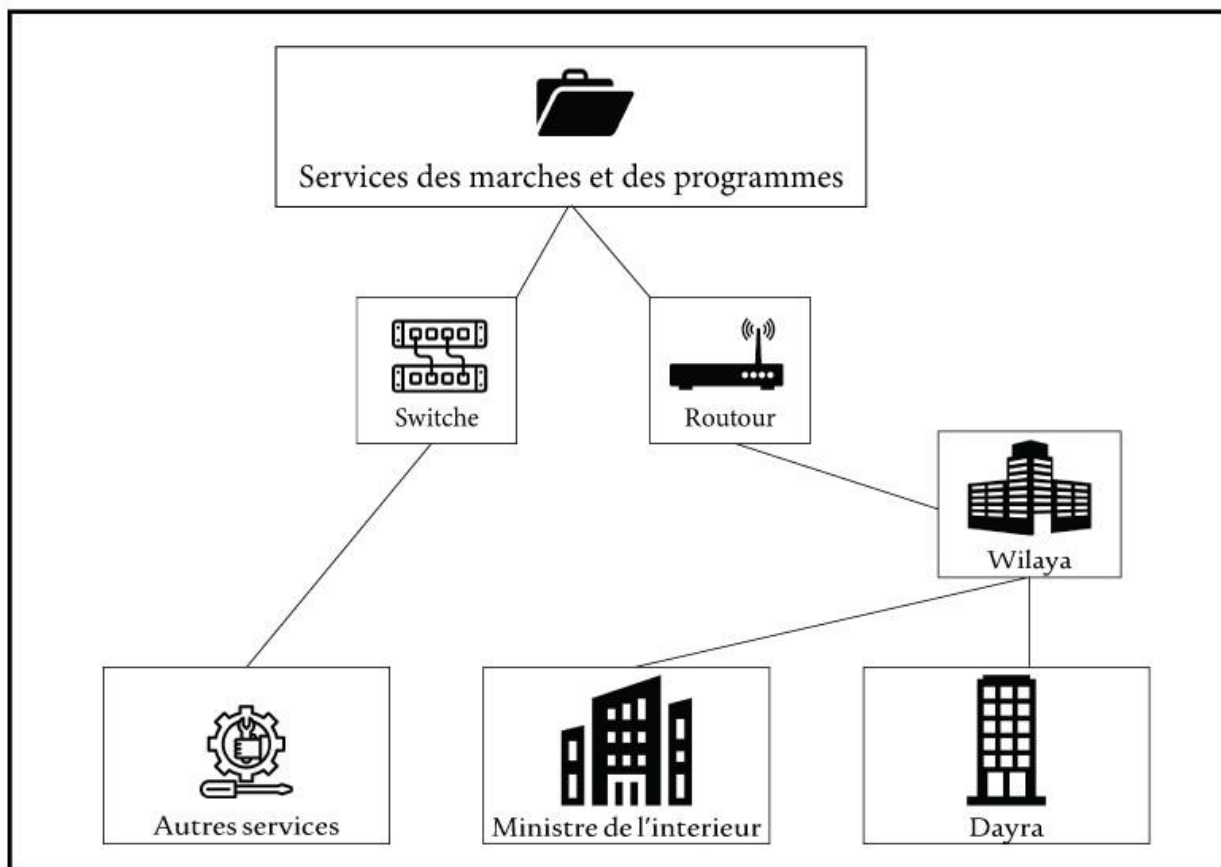


Figure1.3 Réseaux informatique

- ✓ Matériel actif : 3 Switch Cisco CATALIST 24 port 10/100 BTX
- ✓ Routeur Cisco 2620
- ✓ Serveur d'accès distant FTP
- ✓ Serveur d'accès pour le web
- ✓ Serveur de base de données

### 1.3. Etude de l'existant

#### 1.3.1. Cycle de vie d'un projet au sein de la commune

Un projet au sein de la commune doit obligatoirement passer par certaines phases de sa proposition jusqu'à sa réception. Ce processus est suivi par le service des marchés et des programmes comme montré dans la figure 1.4.

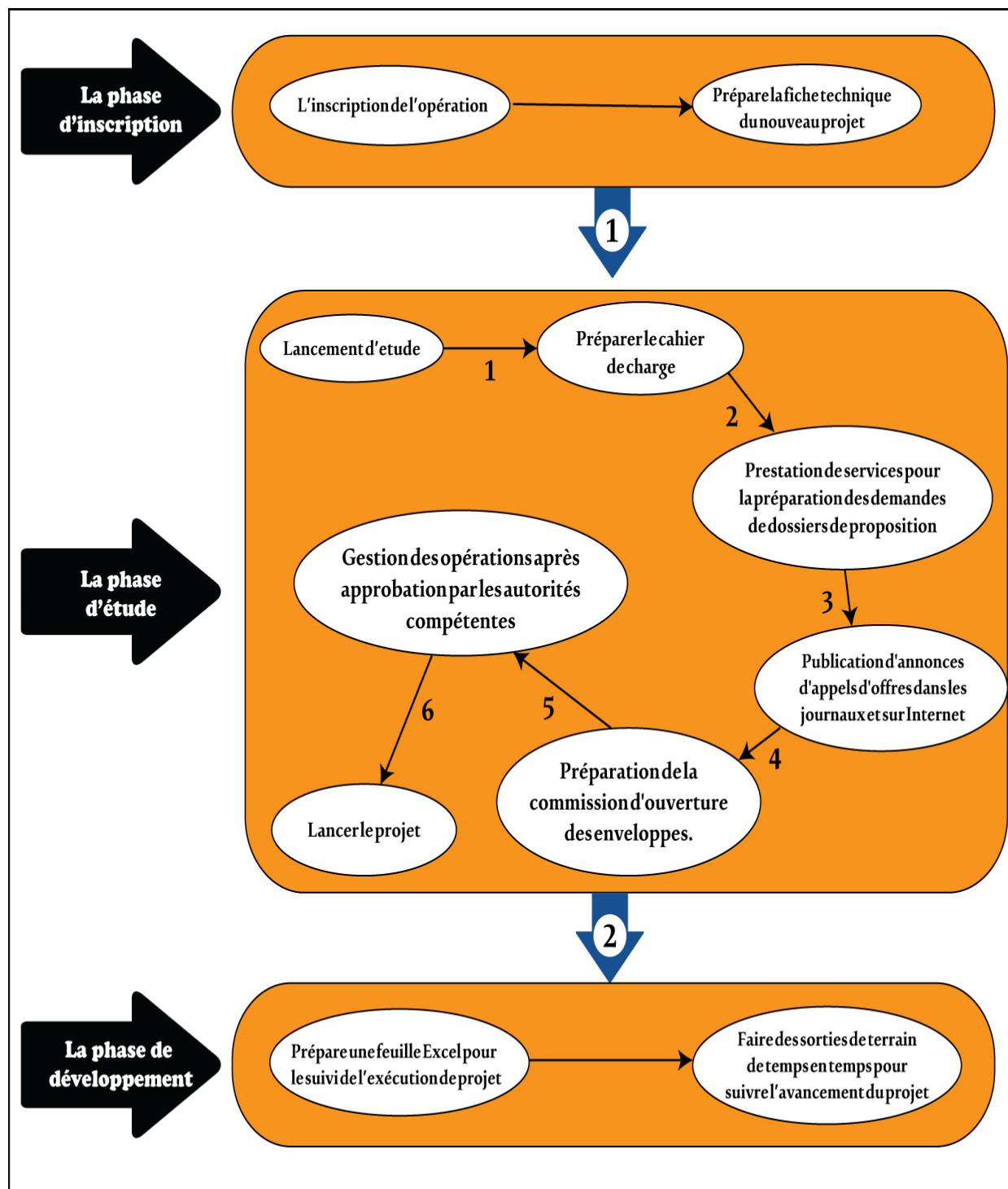


Figure 1.4 Cycle de vie d'un projet au sein de la commune de Sidi Merouane

Le tableau ci-dessous présente les significations des transactions :

Phase	N = <sup>o</sup> opération	Description et services concernés
<b>Inscription</b>	<b>01</b>	<ul style="list-style-type: none"> <li>✓ L'inscription de l'opération</li> <li>✓ prépare la fiche technique du nouveau projet</li> </ul>
<b>étude</b>	<b>02</b>	<ul style="list-style-type: none"> <li>✓ Lancement d'étude</li> <li>✓ Préparer le cahier de charge.</li> <li>✓ Prestation de services pour la préparation des demandes de dossiers de proposition.</li> <li>✓ Publication d'annonces d'appels d'offres dans les journaux et sur Internet.</li> <li>✓ Préparation de la commission d'ouverture des enveloppes.</li> <li>✓ Gestion des opérations après approbation par les autorités compétentes.</li> <li>✓ Lancer le projet.</li> </ul>
<b>Développement</b>	<b>03</b>	<ul style="list-style-type: none"> <li>✓ prépare une feuille Excel pour le suivi de l'exécution de projet .</li> <li>✓ Ce document contient l'intitulé et le prix de l'opération, le délai de réalisation, le taux d'avancement physique, des champs pour le montant facture.</li> <li>✓ Fire des sorties de terrain de temps en temps pour suivre l'avancement du projet</li> </ul>

Table 1 Cycle de vie d'un projet au sein de la commune de Sidi Merouane.

### 1.3.2. Problématique

Après avoir étudié la gestion des marchés publics dans le service des marchés et programmes de la commune de Sidi Merouane, nous avons remarqué les problèmes suivants :

- Absence d'un système informatique pour le suivi du projet malgré la disponibilité de toutes les ressources matérielles.
- La gestion du projet est très complexe, les travailler utilisent toujours des logiciels simples comme Word et Excel.



- Le temps perdu dû à la recherche manuellement des informations.
- Le manque de transparence.
- Le manque de sécurité et fiabilité des données.

### **1.3.3. Objectifs du projet**

Avec les problèmes que nous avons précédemment identifiés au niveau du service des marchés de la commune de Sidi Marouane et qui entravent le suivi des projets, nous proposons d'améliorer ce service et visons ainsi à concevoir un système d'information sécurisé pour le suivi de la situation physique et financière de projets de développement local.

Ce système doit être développé à l'aide d'un framework de développement web (front-end) et d'une Blockchain (back-end). Ce projet permettra d'une part d'améliorer la transparence tout au long du cycle de vie d'un projet et d'autre part de fournir une base de données et de générer des statistiques qui offrent aux responsables la possibilité de contrôler et de superviser toutes les opérations et les aident également à prendre les décisions appropriées.

## **1.4. Conclusion**

Dans ce chapitre, nous avons présenté la commune de Sidi Marouane et notamment le service des marchés et des programmes, lieu de notre stage. Ensuite, nous avons indiqué les principales étapes que le projet doit franchir pour être mis en œuvre. Enfin, nous avons identifié les problèmes liés au suivi des projets et suggéré des solutions pour améliorer ce service.

---

## 2. Blockchain

### 2.1. Introduction

La technologie de la Blockchain est considérée comme l'une des technologies les plus distribuées dans l'histoire du monde. Ses éventuels impacts pourraient révolutionner nos systèmes économiques et nos modes d'échange d'informations, la Blockchain apporte de profondes transformations dans de nombreux secteurs d'application.

Cette technologie porte avant tout la promesse d'une nouvelle gouvernance, au niveau local et global, fondée sur des principes innovants de collaboration, de décentralisation et de transparence.

Dans ce chapitre, nous présentons un aperçu général sur la technologie Blockchain. Nous donnons d'abord la définition et l'histoire de la Blockchain. Ensuite, nous présentons le fonctionnement de la technologie, ses concepts, ses propriétés, ses types et ses plateformes existantes. Enfin, nous présentons les défis de la technologie Blockchain ainsi que ses applications.

### 2.2. Définition du terme Blockchain

La Blockchain est généralement définie comme une technique transparente et sécurisée de stockage et de transmission de l'information qui fonctionne sans dispositif de commande centralisé [5].

Constitue une base de données contenant l'historique de l'ensemble des échanges entre ses utilisateurs depuis sa création, ce dernier peut être comparé à un vaste répertoire public qui inclut tous les échanges effectués par ses utilisateurs. L'une de ses principales caractéristiques est qu'il n'est pas modifiable et protégé par plusieurs processus cryptographiques innovants [5]. Donc, nous pouvons seulement ajouter des opérations, mais pas les modifier ou les supprimer, ils sont inviolables.

Cette technique repose sur un système pair-à-pair décentralisé, les données ne sont pas hébergées par un serveur unique mais réparties entre les utilisateurs et sans intermédiaire. Une partie des utilisateurs conserve des copies de la Blockchain, qui est donc présent partout dans le monde, centaines de copies sont sans cesse mises à jour simultanément.

Contrairement aux bases de données traditionnelles, qui sont administrées par des opérateurs centralisés, la Blockchain est administrée collectivement par tous les nœuds du réseau. Ces nœuds obéissent tous à un même protocole informatique, qui définit les procédures à suivre, ainsi que les conditions à respecter pour mettre à jour la base de données [5].

Concrètement, la technique de la Blockchain prend la forme d'un registre qui collecte des données, généralement des transactions, regroupées dans des blocs reliés entre eux. Un bloc n'est qu'un ensemble d'informations rassemblées, et les blocs sont irréversiblement liés par les chaînes. Chaque bloc est *validé* par les nœuds du réseau, des utilisateurs-validateurs appelés les mineurs. Après la validation, les blocs sont horodatés et intégrés dans la Blockchain. Donc, la transaction devient alors visible par le destinataire et l'ensemble du réseau.

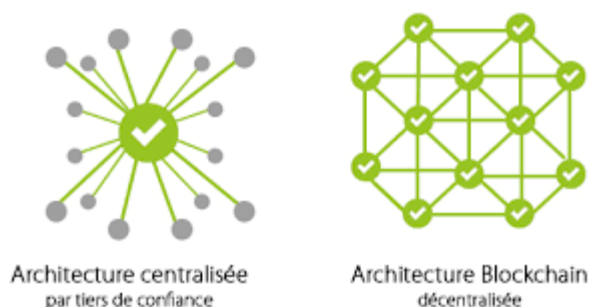


Figure 2.1 Architecture de la Blockchain

### 2.3. Historique de la Blockchain

Dans ce qui suit, nous discutons des phases importantes de cette technologie.

**En 1988** : *D. Chaum* a proposé un nouveau système de paiement permettant aux utilisateurs d'effectuer des transferts d'argent sur internet sans être traçables par le gouvernement [6].

L'année suivante sort l'implémentation, sous le nom de *DigiCash*, cependant le projet est un échec, en partie car le commerce en ligne n'en est encore qu'à ses prémices, et le projet est abandonné faute de moyens suffisants [7].

**En 2008** : *Satoshi Nakamoto* a rédigé l'article de fondation de *Bitcoin*, le premier réel Blockchain. Cet article décrit la structure et le fonctionnement du protocole Bitcoin, de façon

théorique. La première implémentation du Bitcoin arrive l'année suivante, sous le nom de *Bitcoin-QT*. C'est également la première fois que le mot Blockchain est employé [8].

En 2013, V. Buterin décide de reprendre le concept de la Blockchain, en y ajoutant la possibilité d'exécuter des applications décentralisées. Le but de cette nouvelle Blockchain, appelée *Ethereum* [9], est de construire une sorte d'ordinateur géant décentralisé. Ces applications sont possibles grâce aux contrats intelligents.

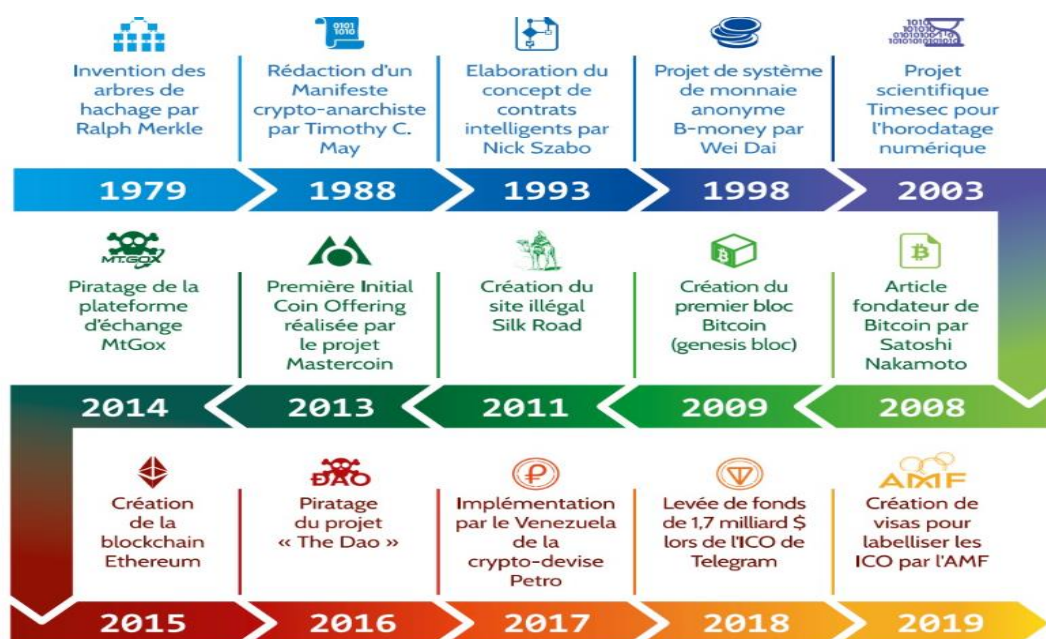


Figure 2.2 Chronologie de la Blockchain

## 2.4. Fonctionnement d'une Blockchain

La particularité de la Blockchain est qu'elle fonctionne sans un organisme de contrôle centralisé. Au lieu d'être regroupées à un seul endroit ou de passer par un seul intermédiaire, les transactions d'achat ou de transfert d'argent, par exemple, dans le cas d'un Blockchain de crypto-monnaie, sont distribuées entre tous les membres d'un réseau par le biais de leur ordinateur.

Pour une première approche du fonctionnement de la Blockchain, le plus simple est de raisonner par un bloc purement monétaire.

On peut prendre l'exemple de *Bitcoin*, ou d'une Blockchain avec des jetons, en commençant par la création d'une transaction qui peut être décrite en quelques étapes (voir la figure 2.3).

- Un compte (ou portefeuille, portemonnaie, wallet) doit être créé pour qu'un utilisateur de la Blockchain puisse envoyer ou recevoir de la crypto-monnaie. A utilise son portefeuille et effectue une transaction vers B. Cette transaction est diffusée sur le réseau.
- A la réception de la transaction, chaque mineur authentifie la transaction à l'aide de la clé publique de A. Cette transaction avec d'autres transactions récentes sont regroupées en bloc, et chaque transaction sera vérifiée et validée par les mineurs. Lors de la vérification de la transaction, l'historique des transactions de A est remonté pour vérifier que l'argent qu'il a reçu précédemment n'a pas été réutilisé. On vérifie en fait tout simplement qu'il n'essaye pas de dépenser deux fois l'argent qu'il reçoit.
- Une fois les vérifications effectuées, le bloc dans lequel se trouve la transaction entre A et B est validé par les mineurs, selon des techniques de consensus qui dépendent du type de la Blockchain, et qui permettent d'atteindre le consensus distribué. Pour cela, les mineurs doivent lancer un procédé crypto graphique, le calcul de la hache du bloc. Donc, chaque bloc possède un identifiant qui prend la forme d'une hache permettant de relier les blocs les uns aux autres. Cette hache est toujours le résultat du hachage du bloc précédent.
- Quand le bloc est validé, il est daté et ajouté à la chaîne de blocs à laquelle tous les utilisateurs ont accès.
- Enfin, B reçoit la transaction de A [10].

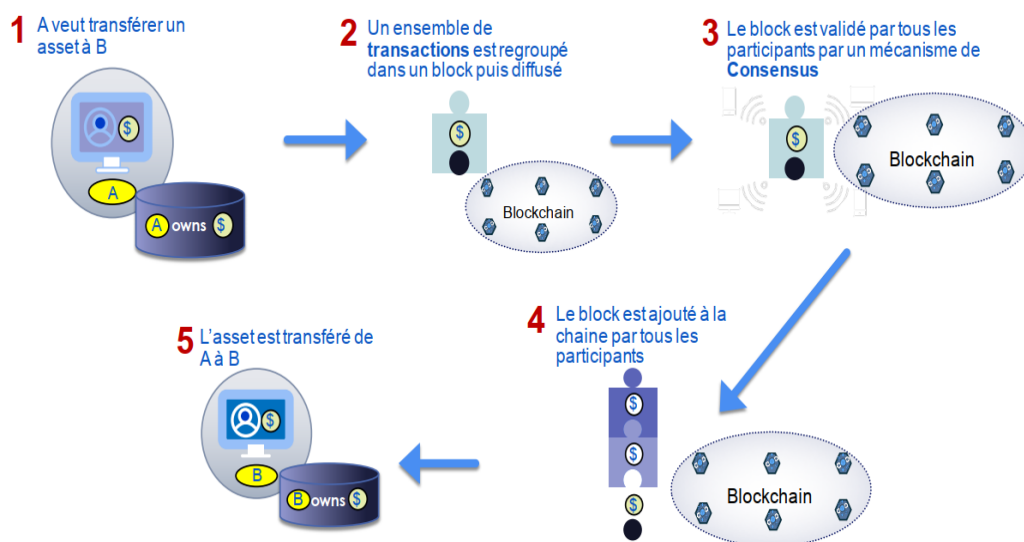


Figure 2.3 Principe général d'une Blockchain [10]

## 2.5. Concepts de la Blockchain

### 2.5.1. Structure du réseau :

Chaque participant du réseau, appelé nœud, est en réalité un ordinateur sur lequel tourne un logiciel spécial appelé client.

Pour une Blockchain, il peut exister différents clients, développés par plusieurs communautés ou entreprises, dans des langages variés.

Le but étant qu'au final tous ces clients s'interfacent sans difficultés, car tous respectent les conventions et la structure de la Blockchain.

Le réseau est organisé en système pair-à-pair, c'est à dire que chaque machine est à la fois client et serveur, et s'organise sans avoir besoin d'une autorité centrale.

Les échanges se font directement d'un ordinateur à un autre.

Lorsqu'un nouveau nœud souhaite rejoindre le réseau, il se connecte à un ou plusieurs nœuds, dont les adresses sont connues via par exemple un site internet ou encore écrites directement dans le code source du client. Puis grâce à ces nœuds, il peut obtenir l'adresse d'autres nœuds, eux-mêmes connectés à d'autres nœuds, etc.

La topologie du réseau est donc complexe et en mouvement, et il est assez compliqué d'obtenir des informations précises en temps réel.

Les réseaux pair-à-pair sont principalement connus du grand public pour le partage de fichiers, illégal dans de nombreux cas, mais leurs usages sont nettement plus larges.

Le pair-à-pair permet, entre autres, de faire du calcul distribué, d'échanger des paquetages de logiciels [11].

### 2.5.2. Blocs :

Chaque bloc est constitué de plusieurs champs.

- Bloc : le numéro du bloc permettant de vérifier l'ordre de la chaîne.
- Hash du bloc précédent : l'empreinte du bloc précédent, qui permet de lier les blocs entre eux. En effet connaissant l'empreinte du bloc précédent, il est possible de s'assurer de son intégrité, et donc de repérer des éventuelles modifications.
- Transaction : la partie qui contient la liste des transactions.
- L'horodatage indiquant la date et l'heure à laquelle le bloc a été créé les blocs se suivent de façon chronologique.

- Hash : identifiant du bloc actuel.

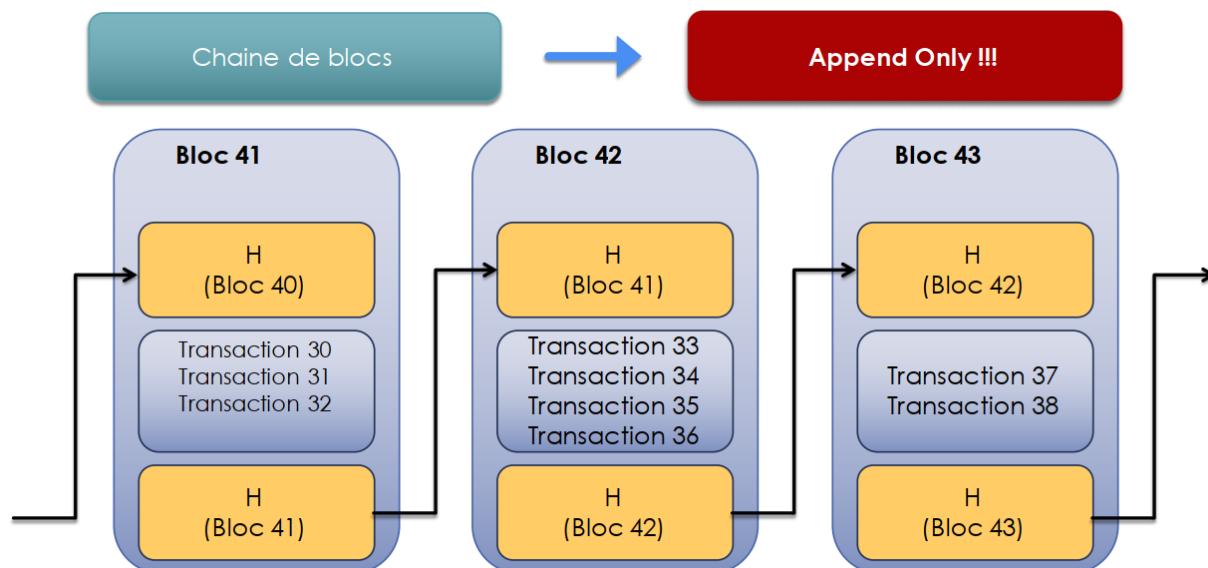


Figure 2.4 Structure d'un block

### 2.5.3. Transaction :

Les transactions sont les choses qui donnent un but à la Blockchain.

Ils sont les plus petits blocs de construction d'un système, les transactions consistent généralement en une adresse de destinataire, une adresse d'expéditeur et une valeur.

Les transactions contiennent une ou plusieurs entrées et une ou plusieurs sorties. Une entrée est une référence à une sortie d'une transaction précédente.

### 2.5.4. Contrat intelligent :

Un contrat intelligent est un accord entre différents protagonistes, écrit sous la forme de code, et s'exécutant de façon automatique selon des critères précis, le contrat peut être fait avec un nombre illimité de participants. Il est stocké directement dans la Blockchain, et bénéficie de ses propriétés d'immutabilité. Il sera également exécuté par tous les nœuds du réseau. Le contrat peut se déclencher sous des conditions internes à la Blockchain (solde d'un compte, vote, durée de temps écoulé, ...).

### 2.5.5. Consensus :

Un consensus est par définition un accord entre plusieurs parties amenant au consentement de la majorité. Dans le cas des Blockchains, il convient de trouver un moyen pour

que l'ensemble du réseau se mette d'accord sur le nouveau bloc et son contenu à ajouter à la chaîne. Un algorithme de consensus valide doit répondre à trois critères [11]:

- Terminaison : chaque participant du réseau doit à la fin avoir choisi un bloc.
- Intégrité : ce bloc doit avoir été proposé par un des nœuds du réseau.
- Accord : tous les participants décident du même bloc.

Il existe de nombreuses méthodes d'obtention du consensus sur une Blockchain les principales sont :

#### **2.5.5.1. Preuve de travail :**

Le principe de preuve du travail (*proof of work*) consiste à résoudre un problème cryptographique complexe. Ainsi en échange d'une quantité de travail effectuée, le nœud compétiteur, appelé mineur, obtient le droit de créer le nouveau bloc de chaîne, et la récompense qui l'accompagne (la récompense du bloc et les éventuels frais de transactions).

La preuve de travail utilisée par Bitcoin est le Hashcash, créé par A. Back en 1997 [12].

L'objectif consiste à trouver une valeur *NONCE*, de sorte que l'empreinte de l'entête du bloc est inférieure à un certain seuil fixé. Ce seuil sert à adapter la difficulté de relever le défi pour créer le bloc à intervalles presque fixes, même en cas de changement de puissance de calcul du réseau, il n'existe aucune véritable stratégie pour relever ce défi, l'objectif est de tester de nombreux *NONCE* pour en trouver un qui fonctionne. Si cette solution coûte du temps et des ressources, sa vérification se fait rapidement, permettre à chaque participant au réseau de faire en sorte que le nouveau bloc créé soit légitime. Cet algorithme facilite le combat contre les attaques cibles, puisque seule la puissance de calcul possédée par un participant lui permet de résoudre le challenge. Cependant, un participant possédant plus de 51% de la puissance de calcul totale du réseau aurait une très forte probabilité de sortir gagnant de chaque challenge, et donc de forger à lui tout seul la chaîne.

#### **2.5.5.2. La preuve d'enjeu**

La preuve d'enjeu (*proof of stake*) est une autre proposition de résolution du consensus, soutenue entre autres par les développeurs d'*Ethereum*.

Pour sélectionner le nœud qui créera le nouveau bloc, le système est basé sur sa participation dans le système. Plusieurs critères peuvent être pris en considération, tels que la quantité de jetons qu'il possède, voire la durée de possession.



## 2.6. Propriété d'immuabilité de la Blockchain

Une des particularités de la Blockchain, est de chaîner les blocs entre eux, grâce à la cryptographie. Pour cela, on ajoute dans chaque nouveau bloc l'empreinte du bloc précédent, qui lui-même contiendra l'empreinte du bloc le précédant. Ainsi, les blocs vont être liés entre eux de façon immuable, permettant de garantir l'intégrité des données. Si une personne malveillante tente d'altérer un bloc au milieu de la chaîne, cette modification aura un impact direct sur son empreinte, et le bloc suivant ne le reconnaîtrait plus.

Pour illustrer cela, dans la figure 2.5, on tente de modifier le contenu du bloc 94. La Blockchain originale est représentée en noir. Pour des raisons de lisibilité, seules les empreintes des blocs précédents sont représentées.

Un attaquant décide de modifier le bloc 94, afin d'y insérer une transaction en sa faveur (par exemple un transfert de fond important vers son compte). Le nouveau bloc, que nous appellerons 94', est représenté en rouge. Comme son contenu est différent du bloc 94, son empreinte sera également différente. Ainsi, le bloc 95 ne reconnaîtra pas le bloc 94' comme son bloc précédent.

Comme il est d'usage de choisir la chaîne la plus longue, les autres nœuds du réseau continueront avec la Blockchain d'origine, en noir. Il faudrait reconstruire toute la suite de la chaîne pour obtenir une chaîne valide.

Il faudrait ainsi changer le bloc suivant, et il sera impossible de faire cette opération qui donne une puissance de la Blockchain [11].

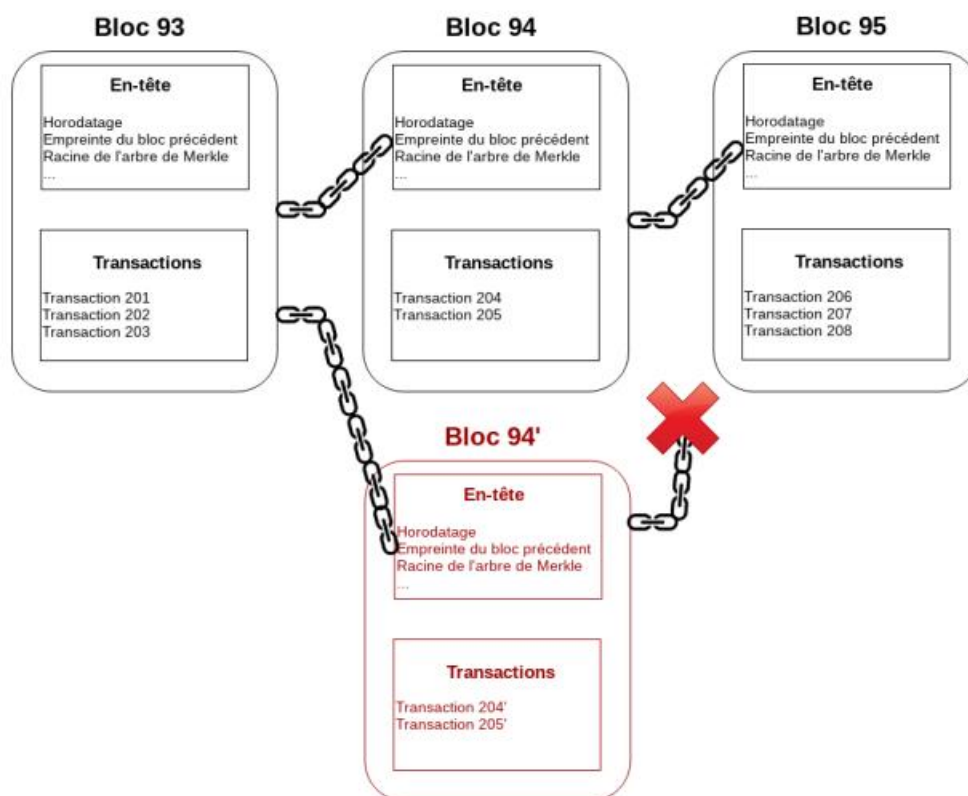


Figure 2.5 Tentative de modification d'un ancien block [11]

## 2.7. Types de la Blockchain

Généralement, il existe trois catégories de la Blockchain : publique, consortium, et privée.

### 2.7.1. Publique

Il s'agit de la Blockchain accessible à n'importe qui dans le monde et aucune permission n'est demandée pour effectuer des transactions ou pour participer au processus de consensus. Tous les acteurs sont en situation égalitaire dans leur participation au réseau.

Les Blockchains principales sont des Blockchains publiques, comme Bitcoin, Ethereum, Monero, Litecoin, Dash.

### 2.7.2. Privée

Il s'agit de la Blockchain tournant sur un réseau privé, dans laquelle tous les participants sont connus et pour laquelle la gouvernance est assurée par une organisation.

### 2.7.3. Consortium (hybride)

Il s'agit de la Blockchain dans laquelle le processus de consensus (validation des transactions/blocs) est contrôlé par un nombre connu et restreint de nœuds. Certains nœuds peuvent être rendus publics (accès autorisé en lecture seule) tandis que d'autres restent privés.

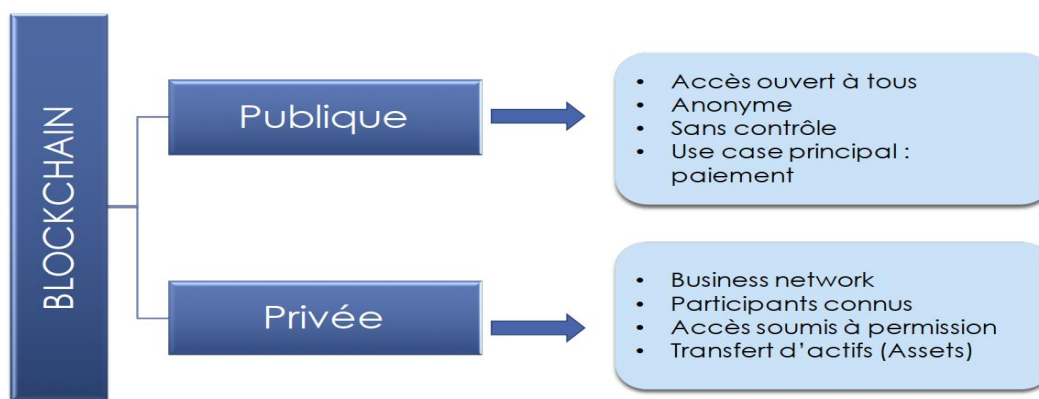


Figure 2.6 Types de la Blockchain

## 2.8. Les plateformes Blockchain

Dans cette partie, nous présentons les plateformes les plus populaires de la Blockchain, dont la liste se résume dans Bitcoin, Ethereum, Hyperledger Fabric, et quelques autres technologies que nous décrivons brièvement dans ce qui suit.

### 2.8.1. Bitcoin

Figure 2.7 Logo du protocole *Bitcoin*

*Bitcoin* est la première crypto-monnaie exploitée sur un réseau de type pair-à-pair. Contrairement aux systèmes bancaires et de paiement traditionnels, Bitcoin est basé sur une confiance décentralisée il n'y a pas d'autorité centrale de confiance dans le système. La confiance est due aux interactions entre les différents participants de l'écosystème.

Dans le système Bitcoin, il existe un grand livre distribué qui stocke toutes les transactions Bitcoin depuis sa naissance en 2008.

Le contenu du *Ledger* est répliqué dans de nombreux nœuds de traitement répartis géographiquement au sein du réseau Bitcoin [13].

### 2.8.2. Ethereum



Figure 2.8 Logo du protocole Ethereum [14].

Ethereum est une Blockchain présentée en 2013 par *V. Buterin*. La principale nouveauté qu'elle apporte est le fait de permettre d'utiliser des applications de façon décentralisée. Ces applications sont appelées contrats intelligents, et sont stockées directement dans la Blockchain, donc répliquées sur tous les nœuds du réseau [14].

### 2.8.3. Hyperledger Fabric



Figure 2.9 Logo du Hyperledger Fabric [15]

Hyperledger est un effort de collaboration multi-projets à code source libre, hébergé par la Fondation Linux, créé pour faire progresser les technologies de la Blockchain intersectorielles [15].

Hyperledger Fabric est un framework Blockchain destiné aux entreprises, pour le développement des solutions de base Blockchain avec une architecture modulaire. Les données peuvent être stockées dans plusieurs formats, et divers algorithmes de consensus peuvent être configurés.

#### 2.8.4. Litecoin



Figure 2.10 Logo du Litecoin [16]

Litecoin est un réseau de paiement mondial open source, qui est entièrement décentralisé sans autorités centrale. Les utilisateurs peuvent maintenant contrôler leurs propres finances, sécurisées par un système basé entièrement sur des mathématiques.

Par rapport à Bitcoin, les transactions Litecoin sont plus rapides à confirmer et ont une efficacité de stockage accrue.

De par son utilisation par l'industrie, le volume d'échange et de liquidité sur les marchés d'échange, Litecoin est un moyen de commerce complémentaire à Bitcoin, qui a fait ses preuves [16].

#### 2.8.5. Ripple



Figure 2.11 Logo du Ripple [17]

Ripple est un système de règlement brut en temps réel, un réseau d'échange de devises et de transferts de fonds entre les institutions financières. Ripple utilise un ledger commun qui est géré par un réseau de serveurs de validation indépendants qui comparent constamment les enregistrements des transactions. Ces serveurs de validation peuvent appartenir à des particuliers ou à des banques [17].

Plateforme	Type	Inventeur	Année	Consensus	Crypto-monnaie
<b>Bitcoin</b>	Public	Satoshi Nakamoto	2008	PoW	BTC
<b>Ethereum</b>	Public	Vitalik Buterin	2014	PoS	Ether (ETH)
<b>Hyperledger Fabric</b>	Privée	La Fondation Linux	2017	PBTF / SIEVE	<b>Aucune</b>
<b>Litecoin</b>	Public	Charles Lee	2011	Scrypt	LTC
<b>TEZOS</b>	Public	Tezos Foundation	2018	PoS	XTZ

Figure 2.12 Plateformes Blockchain

## 2.9. Les avantages de la technologie

Plusieurs caractéristiques sont associées à la Blockchain : désintermédiation, traçabilité, transparence, consensus distribué, ineffaçable, structure distribuée, résilience, sécurité et confiance. Toutes ces caractéristiques constituent le potentiel innovateur de la Blockchain. Dans ce qui suit, nous présentons les caractéristiques principales de la technologie Blockchain.

### 2.9.1. La désintermédiation

La technologie Blockchain permet d'échanger sans le contrôle d'un tiers, la validation et l'ajout d'un bloc résultent d'un consensus entre les utilisateurs-validateurs, qui repose sur la possibilité de vérifier leur travail de validation et qui rend inutile le contrôle par une institution de référence. Tout est effectués sans l'intervention d'une autorité centrale, les utilisateurs opèrent la surveillance, et se contrôlent mutuellement, assurant la certification des sauvegardes et leurs cohérences [18]

### 2.9.2. La transparence

Une fois qu'un document est inscrit sur la Blockchain, cela suffit à prouver que ce dernier existe bien à l'instant et qu'il n'a pas été modifié. La Blockchain est qualifiée de transparente car tout le monde peut la télécharger dans son intégralité et vérifier à tout moment son honnêteté [19].

Tous les utilisateurs de la Blockchain peuvent ainsi voir les transactions présentes et passées. Si la transparence est assurée pour les transactions, l'anonymisation des utilisateurs remet en cause cette caractéristique. En effet, l'anonymat possible sur la Blockchain peut être utilisé pour des activités frauduleuses, difficiles voire impossibles à détecter et à réguler.

### **2.9.3. La sécurité**

L'hébergement décentralisé fait également de la Blockchain une technologie sûre elle rend aussi impossible la suppression de toutes les copies des documents, qui existent sur une multitude de serveurs à travers le monde. La Blockchain a une grande résistance, car toutes les données sont copiées dans les différents serveurs. Cela la rend résistante aux cyber-attaques ou au contrôle de l'État. En effet, s'il est possible de s'attaquer à un ou plusieurs ordinateurs, il est plus compliqué de s'attaquer aux blocs d'informations copiés dans l'ensemble des ordinateurs connectés au réseau. Cela offre à la Blockchain un haut niveau de sécurité. La Blockchain est donc considérée comme inattaquable et inviolable [19].

### **2.9.4. L'autonomie**

La puissance de calcul et l'espace d'hébergement sont fournis par les nœuds du réseau, c'est-à-dire les utilisateurs eux-mêmes. Il n'y a donc pas besoin d'infrastructures centrales. Au sein d'une Blockchain, l'infrastructure n'est plus concentrée dans les mains d'une organisation mais est, au contraire, éclatée dans l'ensemble des points du réseau. Une Blockchain est donc autoportante et indépendante de services tiers [5].

La Blockchain est l'architecture sous-jacente de la crypto monnaie bitcoin, qui reste aujourd'hui le cas d'usage le plus connu. La première fonction de la Blockchain a donc été le transfert d'actifs financiers. Mais cette technologie ne cesse d'évoluer et est à la base de bien d'autres applications qu'un réseau de paiement [5].

Elle est aujourd'hui utilisée aussi par d'autres acteurs et les opérations et données ne sont pas nécessairement financières. Son potentiel peut en effet s'appliquer à tous les domaines qui impliquent un échange, une sauvegarde ou même une preuve et pourrait donner lieu à des usages révolutionnaires pour la santé, l'énergie, les politiques publiques

## **2.10. Les défis de la technologie**

Dans ce qui suit, nous présentons les défis de la technologie Blockchain.

### **2.10.1. Consommation d'énergie**

La consommation d'énergie dans la Blockchain est relativement élevée car, pour une année donnée, la consommation d'énergie des mineurs de bitcoin était à elle seule supérieure à la consommation d'énergie par habitant de 159 pays individuels. La tenue d'un registre en temps

réel est l'une des raisons de cette consommation, car chaque fois qu'un nouveau nœud est créé, il communique avec tous les autres nœuds en même temps.

### **2.10.2. Coût**

Selon les études, le coût moyen de la transaction Bitcoin est de 75 à 160 dollars et la majeure partie de ce coût est couverte par la consommation d'énergie. Il y a très peu de chances que nous puissions résoudre ce problème grâce aux progrès de la technologie. Ainsi un autre facteur relié au problème de stockage pourrait être couvert par les questions d'énergie ne peut pas être résolu.

## **2.11. Les applications de la Blockchain**

Dans cette section, nous présentons certaines applications déjà existantes de la technologie Blockchain.

### **2.11.1. La certification des droits d'auteur**

Comme le propose la start-up *Mediachain23* qui permet aux artistes de déposer leurs créations sur la base de données, et tout en gardant le contrôle sur elles et sur leur authenticité. La Blockchain peut être un outil pour la mise en œuvre de monnaies locales complémentaires (MLC) numériques, à l'image du Léman, une MLC utilisée dans plusieurs communes françaises et suisses. Cela permet de mieux répondre aux besoins spécifiques des entreprises du réseau et de garantir une meilleure traçabilité [20].

### **2.11.2. Améliore la transparence de chaînes d'approvisionnement**

Pour accroître l'efficacité du fret maritime, *Maersk* le plus grand armateur de porte-conteneurs du monde et *IBM* ont lancé une initiative visant à établir un système mondial basé sur la Blockchain pour numériser les flux commerciaux et le suivi des expéditions de bout en bout. Le système permet à chaque partie prenante de la chaîne d'approvisionnement de visualiser la progression des marchandises tout au long de la chaîne, en comprenant où un conteneur est en transit. Les parties prenantes peuvent également voir le statut des documents douaniers et peuvent consulter les connaissements et d'autres données. La technologie de la Blockchain garantit un échange de données sécurisé et un dépôt inviolable de cette documentation [21].

### **2.11.3. L'identité numérique**

La Blockchain pourrait être utilisée autant que moyen sûr de vérification d'authenticité d'une personne, cela implique la réduction ou même l'élimination des possibilités des fraudes. L'Estonie par exemple a mis en œuvre le système *E-Residence* comme système d'identification



électronique basée sur la technologie Blockchain pour ses citoyens [22].

La liste des applications déjà existante n'est pas finie et nous pensons que d'autres applications Blockchain apparaîtront dans un avenir proche dans des domaines aussi divers que l'art, le tourisme, le sport et autres.

#### 2.11.4. Les systèmes de vote électronique basé sur la Blockchain

En février 2015, la Fondation Bitcoin a dévoilé un nouveau projet qui s'articule autour d'un système de vote basé sur la Blockchain, qui offre une transparence encore plus grande dans le processus de vote, chaque vote étant enregistré sur la Blockchain s'appuyant sur l'immutabilité, la transparence et le consensus inhérents à la technologie Blockchain, les systèmes de vote où chaque vote est enregistré sous un code cryptographique sécurisé. À la jonction entre la démocratie électronique et la technologie de la Blockchain, ce type de système de vote a d'abord été mis en œuvre par un parti politique danois pour des élections internet [23].

				
Financial	Public Sector	Retail	Insurance	Manufacturing
<ul style="list-style-type: none"> <li>• Trade Finance</li> <li>• Cross-currency payments</li> <li>• Mortgages</li> <li>• Dispute Management</li> <li>• Energy Trading</li> </ul>	<ul style="list-style-type: none"> <li>• Asset Registration</li> <li>• Citizen Identity</li> <li>• Medical records</li> <li>• Medicine supply chain</li> <li>• Voting</li> </ul>	<ul style="list-style-type: none"> <li>• Supply chain</li> <li>• Loyalty programs</li> <li>• Information sharing (supplier – retailer)</li> <li>• Food Trust</li> </ul>	<ul style="list-style-type: none"> <li>• Claims processing</li> <li>• Risk provenance</li> <li>• Asset usage history</li> <li>• Claims file</li> </ul>	<ul style="list-style-type: none"> <li>• Supply chain</li> <li>• Product parts</li> <li>• Maintenance tracking</li> </ul>

Figure 2.13 Les applications de la Blockchain [23].

## 2.12. Conclusion

Dans ce chapitre, nous avons présenté un aperçu de la technologie Blockchain. Nous avons donné d'abord la définition et l'historique de la Blockchain. Ensuite, nous avons abordé le fonctionnement de la technologie, ses concepts, ses propriétés, ses types et ses plateformes existantes. Enfin, nous avons présenté les avantages et les défis de la technologie Blockchain ainsi que ses applications.

---

## 3. Analyse et conception

### 3.1. Introduction

Dans ce chapitre, nous présentons d'abord le processus unifié (UP) et le langage de modélisation UML (Unified Modeling Language). Ensuite, nous spécifions les cas d'utilisation du système, ses diagrammes d'activité et de séquence, un diagramme de classes et un diagramme de déploiement spécifiant l'infrastructure physique du système. Enfin, nous élaborons le modèle relationnel qui définit l'ensemble des tables de la base de données de notre système.

### 3.2. Le choix de la méthodologie

Un certain nombre de méthodes de développement logiciel sont fondées sur UML comme la méthode: UP, 2 TUP ...

Parmi ses méthodes, notre choix repose sur la méthode UP (procédé unifié).

#### 3.2.1. Rappel sur le processus unifié

##### 3.2.1.1. Définition du processus unifié

Le Processus Unifié (UP, pour *Unified Process*) est un processus de développement logiciel «itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques » [24].

##### 3.2.1.2. Les caractéristiques du processus unifié :

- **UP est itératif et incrémental.**

Le projet est divisé en itérations ou étapes à court terme qui permettent un meilleur suivi de l'avancement global.

À la fin de chaque itération, une partie exécutable du système final est créée de manière incrémentielle voir (Figure 3.1).

- **UP est centré sur l'architecture**

Tout système complexe doit être divisé en composants modulaires afin de faciliter son entretien et son évolution. Cette architecture doit être modélisée en UML, et non uniquement documentée en texte.

- **UP est guidé par les cas d'utilisation d'UML**

La réalisation du projet prend en compte les besoins et les exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, correctement décrits et mis en priorité.

- **UP est piloté par les risques**

Les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement, les actions qui doivent être entreprises dans ce contexte déterminent l'ordre des itérations [24].

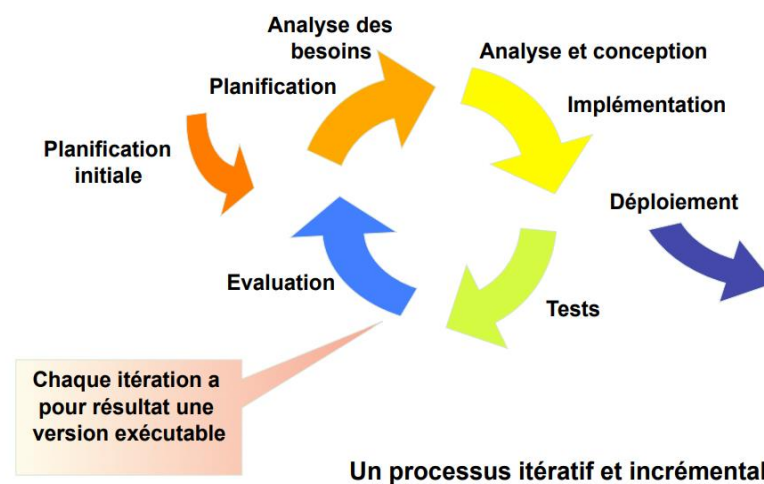


Figure 3.1 Déroulement du processus unifié

### 3.2.1.3. Cycle de vie du processus unifié

Le but d'un processus unifié consiste à contrôler la complexité des projets minimisant les risques, est une série de principes génériques adaptés à la nature particulière de projets et gère le processus de développement à travers deux axes (figure 3.2).

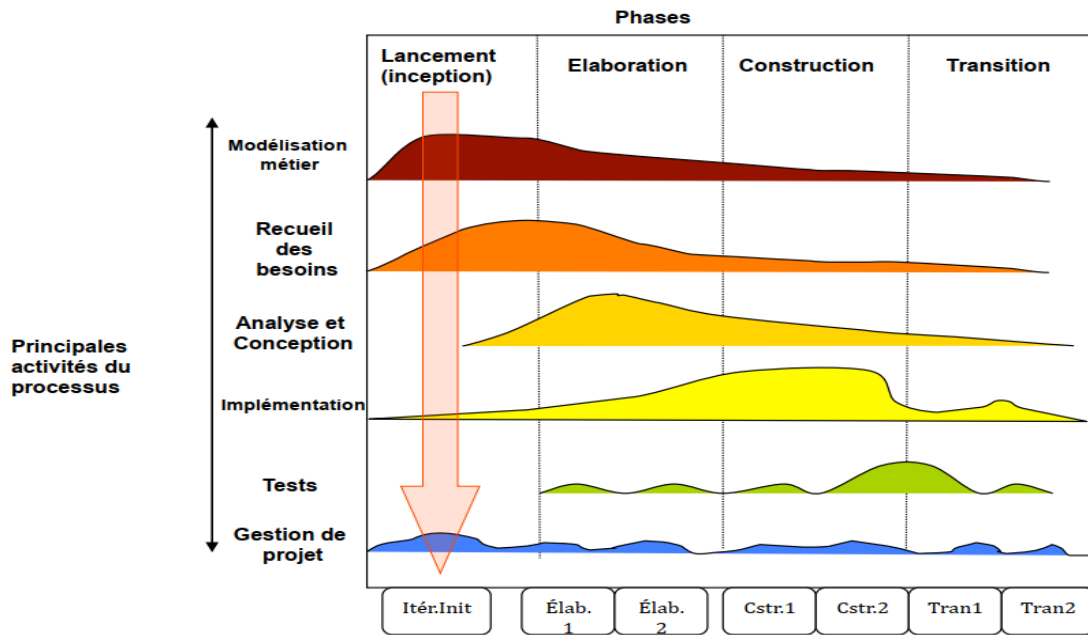


Figure 3.2 Déroulement du processus unifié.

L'axe vertical : représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.

L'axe horizontal : représente le temps et montre le déroulement du cycle de vie du processus, cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en termes de cycles, de phases, d'itérations et de jalons

Pour réaliser un tel cycle de manière efficace, les développeurs ont besoin de toutes représentations du produit logiciel.

Un modèle de cas d'utilisation.

Un modèle d'analyse : détailler les cas d'utilisation.

Un modèle de conception : finissant la structure statique du système sous forme de sous-systèmes, de classes et interfaces.

Un modèle d'implémentation : intégrant les composants.

Un modèle de déploiement : définissant les nœuds physiques des ordinateurs.

Un modèle de test : décrivant les cas de test vérifiant les cas d'utilisation.

Une représentation de l'architecture.

## 3.2.2. Rappel sur le langage UML

### 3.2.2.1. Définition d'UML

C'est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter, spécifier les artefacts de systèmes logiciels, de plus il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquissé des architectures logicielles, concevoir des solutions et communiquer des points de vue, comme il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML résulte de l'unification de techniques ayant fait leurs preuves pour l'analyse et la conception de grands logiciels et de systèmes complexes [24].

### 3.2.2.2. Définition d'un diagramme.

Un diagramme UML est une représentation graphique qui prend en compte un aspect particulier du modèle. Chaque type de diagramme UML dispose de sa propre structure et sémantique spécifique. Les diagrammes que nous avons utilisés pour la modélisation de notre projet, nous avons utilisé différents types de diagrammes (Diagrammes dynamiques et diagrammes statiques).

- **Diagramme de cas d'utilisation :**

Est utilisé dans le cadre de la spécification des exigences. Il présente des interactions fonctionnelles, entre les acteurs et le système mis en œuvre.

- **Diagramme de classe :**

Diagramme de classe Il présente les éléments statiques, classes, associations, interfaces, attributs, opérations, généralisations, etc ...

Le diagramme de classe constitue l'axe d'un développement orienté objet, en analyse, son objectif consiste à décrire la structure des entités manipulées par les utilisateurs, et en conception, le diagramme de classe est la structure du code orienté objet.

- **Diagramme de déploiement :**

Diagramme de déploiement, Il montre le déploiement physique des « artefacts » sur les ressources matérielles.

- **Diagramme de séquence :**

Il montre l'ordre vertical des messages envoyés entre les objets dans une interaction. Ils représentent des échanges de messages entre éléments, dans le cadre d'un fonctionnement

particulier du système. Les diagrammes de séquence servent d'abord à développer en analyse les scénarios d'utilisation du système. Ce dernier permettez de concevoir les méthodes des classes.

## 3.2. Identification des besoins

### 3.2.1. Identification des besoins fonctionnels

Nous avons effectué plusieurs interviews pour identifier au mieux les besoins de l'application, et ceci afin de répondre aux attentes des utilisateurs.

Nous sommes allés chercher les informations au niveau de la commune cette phase correspond à une recherche sur le terrain pour bien définir le cadre de notre système.

#### 3.2.1.1. Identification des acteurs

Le système suppose deux types d'acteurs : administrateur et opérateur.

- **L'opérateur** : est responsable de publier les mises à jour des états d'avancements, tau de consommation et le reste du frais du projet, l'opérateur a des privilèges restreints et peut uniquement ajouter des mises à jour.
- **L'administrateur** : a le droit de créer de nouveaux projets, d'ajouter des utilisateurs, de fermer des projets et de surveiller les mises à jour.

#### 3.2.1.2. Modélisation du contexte

Utilisateurs	Description du besoins fonctionnel
<b>Administrateur</b>	L'application doit permettre à l'administrateur <b>Authentification des administrateurs.</b> Gérer projet et situation. <ol style="list-style-type: none"> <li>1. <b>Ajouter des projets.</b></li> <li>2. <b>Modifier des projets.</b></li> <li>3. <b>Supprimer des projets.</b></li> <li>4. <b>Consulter des projets.</b></li> <li>5. <b>Changements situation des projets.</b></li> <li>6. <b>Consulter Statistique des projets.</b></li> <li>7. <b>Rechercher projet.</b></li> </ol>
<b>Operateur</b>	L'application doit permettre à l'operateur <b>Authentification de l'opérateur.</b> Gérer situation. <ol style="list-style-type: none"> <li>1. <b>Consulter des projets.</b></li> <li>2. <b>Changements situation des projets.</b></li> <li>3. <b>Consulter Statistique des projets.</b></li> <li>4. <b>Rechercher projet.</b></li> </ol>

Tableau 5 Modalisation du contexte

### **3.2.3. Identification des besoins non fonctionnelles.**

Notre système doit répondre aux critères suivants :

- **Traçabilité** : au sein de la chaîne, aucune information ne peut être supprimée. Chaque nœud du réseau dispose d'une copie intégrale du registre de la Blockchain. Il s'agit donc d'un système stable assurant la traçabilité des données.
- **Sécurité** : le système de la Blockchain reste inviolable. La sécurité étant un enjeu majeur, cette caractéristique constitue l'un des principaux avantages de la Blockchain.
- **Scalabilité** : l'utilisateur doit pouvoir augmenter ses capacités de traitement, de stockage, de transmission et de réseaux selon ses besoins.
- **L'utilisabilité** : le système doit offrir à l'utilisateur une interface simple et facile à utiliser.

### 3.3. Diagramme de cas d'utilisation

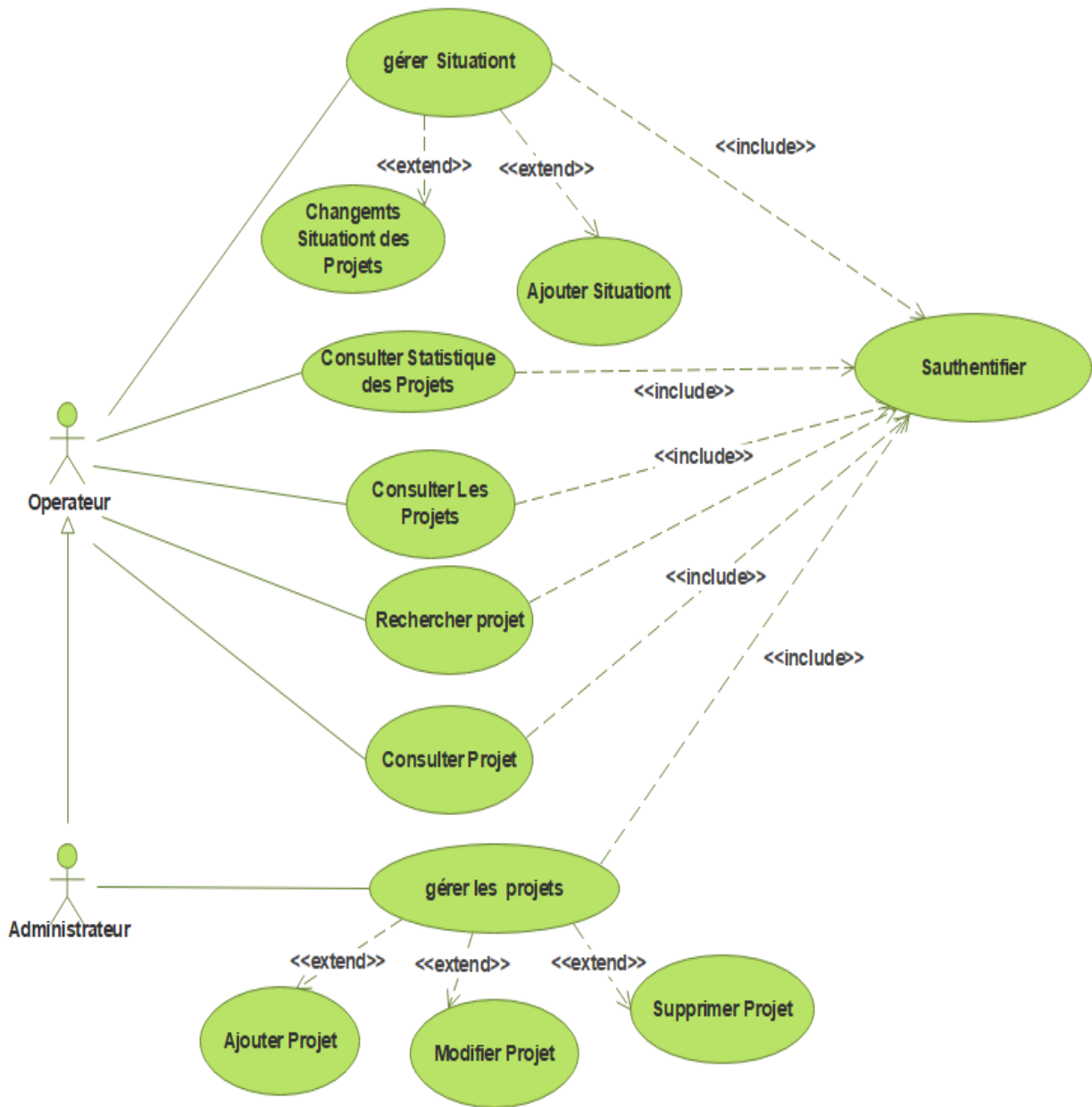


Figure 3.2 Diagramme de cas d'utilisation

### 3.4. Description textuelle

Pour comprendre mieux notre système et les interactions avec les utilisateurs, dans les tableaux ci-dessous nous allons détailler les scénarios des cas d'utilisation avec un diagramme d'activité pour chaque cas d'utilisation.



Cas d'utilisation Authentification
<b>Acteurs:</b> administrateur, operateur.
<b>Résumé :</b> ce cas d'utilisation permet à l'acteur de se connecter au système.
<b>Pré-condition :</b> l'acteur possède un profil et un compte MetaMask.
<b>Post-Condition :</b> l'acteur accède au système.
<b>Description du scenario nominal :</b> <ol style="list-style-type: none"><li>1. L'acteur demande le formulaire d'authentification.</li><li>2. Le système affiche le formulaire d'authentification.</li><li>3. L'acteur saisit ses droits d'accès <i>Nom</i> et <i>mot de passe</i> et confirme la transaction MetaMask.</li><li>4. Le système lit et vérifie les paramètres.</li><li>5. Le système informe l'acteur par notification « <i>login avec Succès</i> ».</li><li>6. Le système redirige l'acteur vers page d'accueil selon son rôle.</li></ol>
<b>Description du scénario alternatif :</b> <p>L'identifiant ou le mot de passe est incorrect :</p> <p>Le système informe l'acteur que les données saisies sont erronées et le scénario reprend aux points 2.</p>

Tableau 6 Cas d'utilisation Authentification

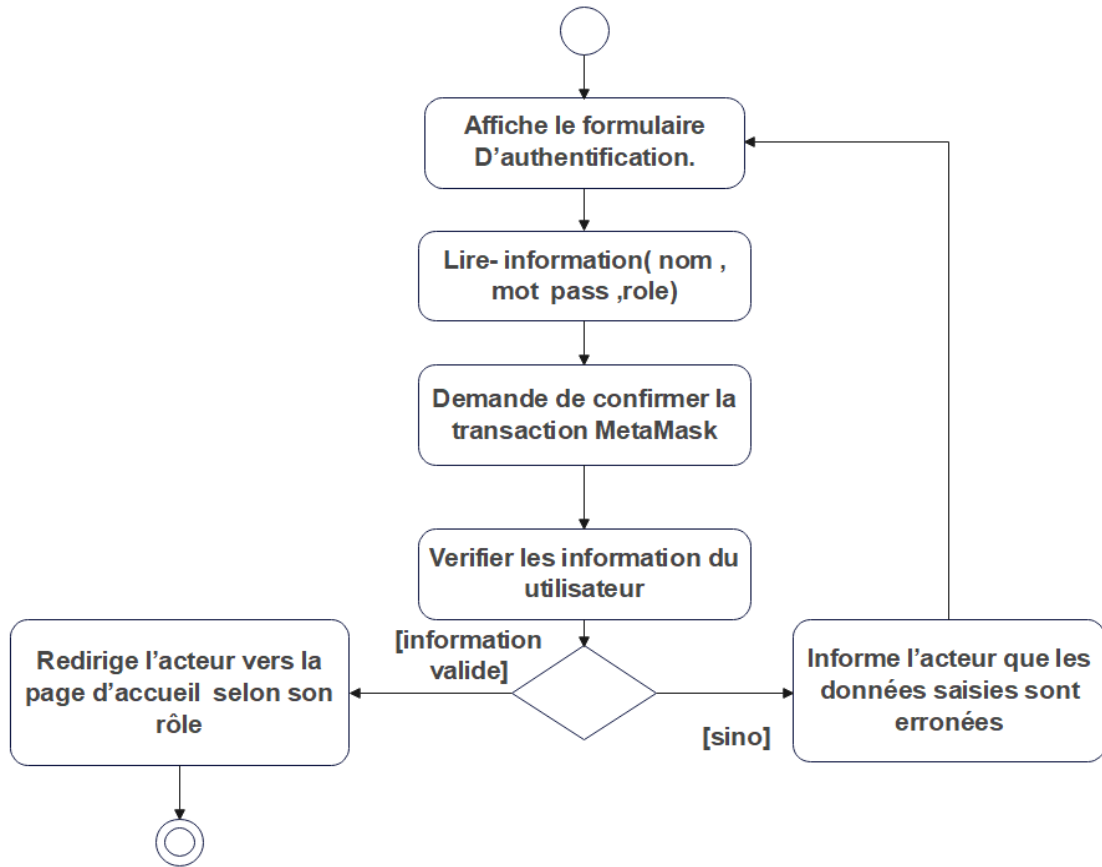


Figure 3.3 Diagramme d'activité de cas « Authentifier ».

<b>Cas d'utilisation ajouter projet</b>
<b>Acteur:</b> administrateur.
<b>Résumé :</b> ce cas d'utilisation permet à l'administrateur d'ajouter un projet dans la Blockchain.
<b>Pré-condition :</b> l'administrateur est authentifié.
<b>Post-Condition:</b> le projet est ajouté avec création de nouveau bloc.
<b>Description du scenario nominal :</b> <ol style="list-style-type: none"><li>1. L'administrateur soumet une demande pour ajouter un nouveau projet.</li><li>2. Le système affiche le formulaire d'ajout.</li><li>3. L'administrateur saisit et entre les paramètres du projet et confirme la transaction MetaMask.</li><li>4. Le système lit les paramètres.</li><li>5. Le système vérifie les paramètres.</li><li>6. Le système ajoute le projet à la Blockchain, crée un block et ouvre une page à l'administrateur pour ajouter un autre projet.</li></ol>
<b>Description du scénario alternatif :</b> <p>Nom ou le numéro du projet existe déjà.</p> <p>Les champs obligatoires sont incomplets.</p> <p>Le système informe l'acteur que les données saisies sont erronées et le scénario reprend aux points 2.</p>

Tableau 7 Cas d'utilisation « Ajouter projet ».

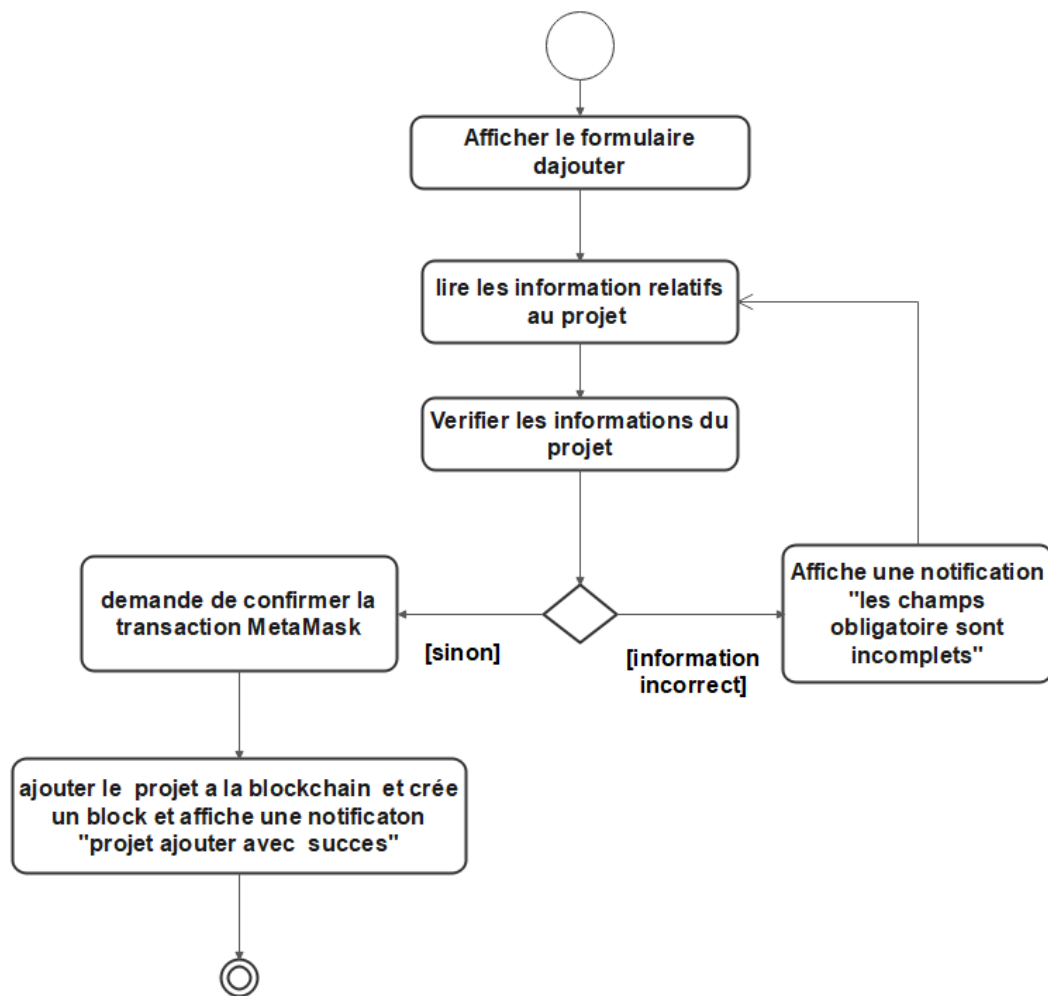


Figure 3.4 Diagramme d'activité de cas « Ajouter Projet ».

Cas d'utilisation consulter Liste des projets.
<b>Acteur</b> : utilisateur
<b>Résumé</b> : ce cas d'utilisation permet à l'acteur de consulter la liste des projets.
<b>Pré-condition</b> : l'acteur est authentifié.
<b>Post-Condition</b> : Le détail des projets est affiché.
<b>Description du scénario nominal</b> : <ol style="list-style-type: none"><li>1. L'utilisateur demande la liste des projets existants.</li><li>2. Le système affiche la liste des projets enregistrés.</li></ol>
<b>Description du scénario alternatif</b> : Si la liste est vide. Le système informe l'acteur par un message d'erreur.

Tableau 8 Cas d'utilisation « Consulter Liste projet ».

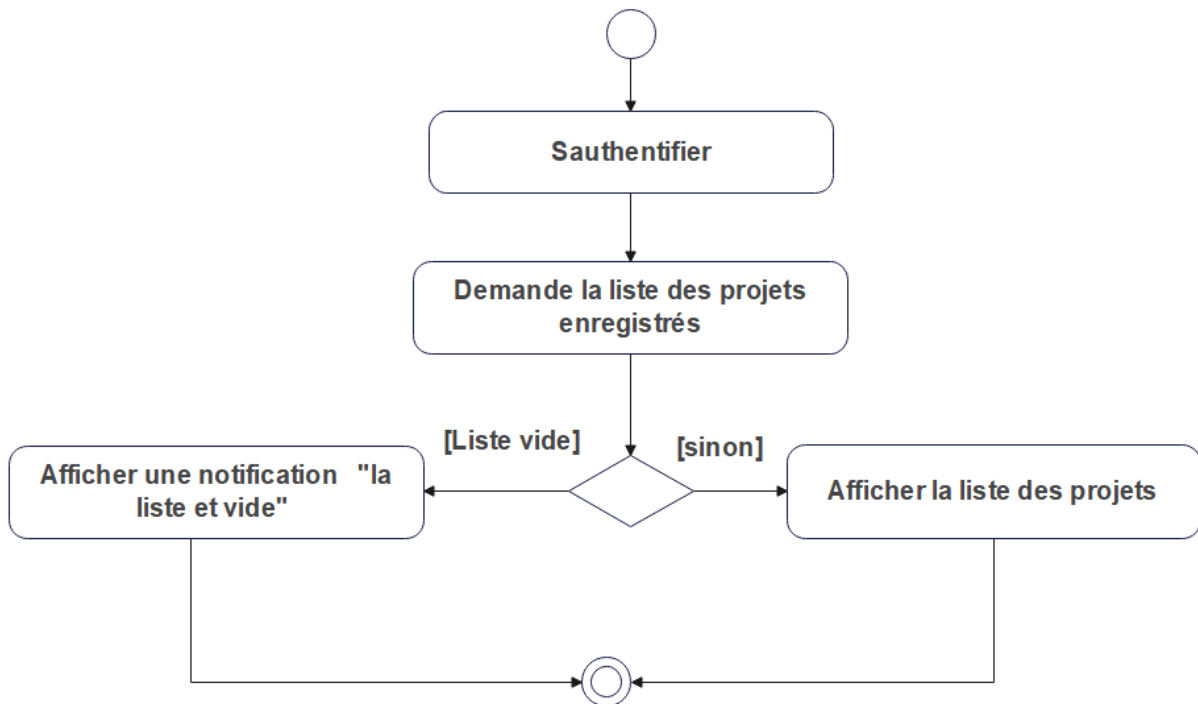


Figure 3.5 Diagramme d'activité de cas « Consulter Liste des Projet ».

<b>Cas d'utilisation supprimer un projet</b>
<b>Acteurs:</b> administrateur
<b>Résumé :</b> ce cas d'utilisation permet à l'acteur de supprimer un projet dans le système
<b>Pré-condition :</b> l'administrateur est authentifié
<b>Post-Condition :</b> Projet est supprimé avec création d'un nouveau bloc.
<b>Description du scénario nominal :</b> <ol style="list-style-type: none"><li>1. L'utilisateur consulte la liste des projets existants</li><li>2. Le système affiche la liste des projets enregistrés.</li><li>3. L'administrateur choisit un projet, demande la suppression et confirme la transaction MetaMask.</li><li>4. Le système supprime le projet, crée un nouveau block et affiche un message de confirmation.</li></ol>
<b>Description du scénario alternatif :</b> <p>En cas d'exception :</p> <p>Le système informe l'acteur par un message d'erreur et le scénario reprend au point 3.</p>

Tableau 9 Cas d'utilisation Supprimer projet.

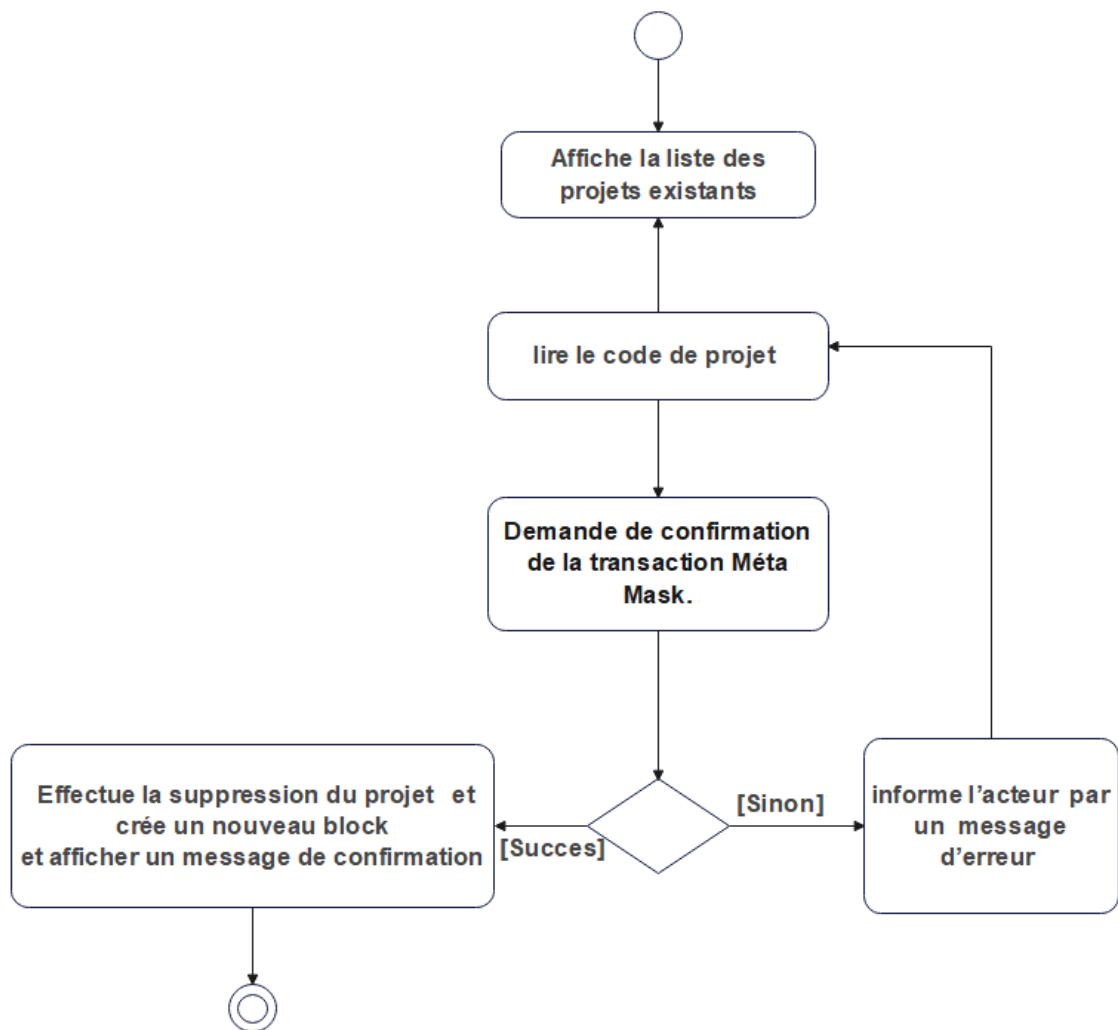


Figure 3.6 Diagramme d'activité « Supprimer Projet »

<b>Cas d'utilisation modifier projet</b>
<b>Acteur:</b> administrateur.
<b>Résumé :</b> ce cas d'utilisation permet à l'acteur de modifier un projet dans le système.
<b>Pré-condition :</b> l'administrateur est authentifié
<b>Post-Condition :</b> Projet est modifié avec création d'un nouveau bloc
<b>Description du scénario nominal :</b> <ol style="list-style-type: none"><li>1. L'utilisateur consulte la liste des projets existants.</li><li>2. Le système affiche la liste des projets enregistrés.</li><li>3. L'administrateur choisit un projet, le modifie et confirme la transaction MetaMask.</li><li>4. Le système effectue la modification du projet puis crée un nouveau bloc et affiche un message de confirmation.</li></ol>
<b>Description du scénario alternatif :</b> <p>Cas exception :</p> <p>Le système informe l'acteur par un message d'erreur et le scénario reprend au points 3.</p>

Tableau 10 Cas d'utilisation modifier projet.



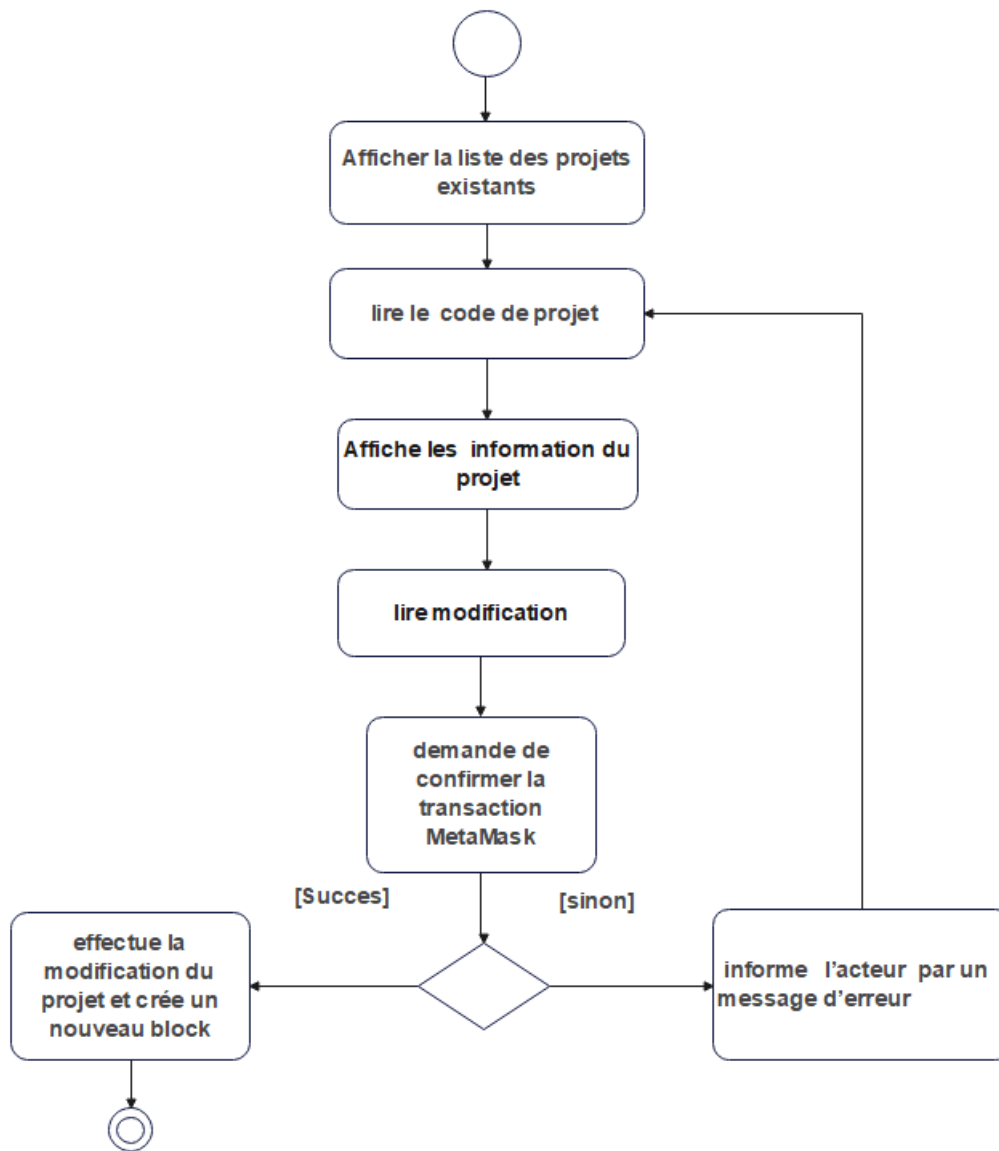


Figure 3.7 Diagramme d'activité de cas « Modifier Projet ».

<b>Cas d'utilisation ChangeSituation projet</b>
<b>Acteur:</b> Operateur.
<b>Résumé :</b> Ce cas d'utilisation permet à l'acteur de change l'état d'un projet dans le système
<b>Pré-condition :</b> l'Opérateur est authentifié.
<b>Post-Condition :</b> état du projet est changé avec création d'un nouveau bloc.
<b>Description du scénario nominal :</b> <ol style="list-style-type: none"><li>1. L'opérateur consulter la liste des projets existants</li><li>2. Le système affiche la liste des projets enregistrés.</li><li>3. L'opérateur choisit un projet et demande le changement d'état et confirme la transaction MetaMask.</li><li>4. Le système effectue le changement du projet, crée un nouveau bloc et affiche un message de confirmation.</li></ol>
<b>Description du scenario alternatif :</b> <p>En cas d'exception :</p> <p>Le système informe l'acteur par un message d'erreur et le scénario reprend au point 3.</p>

Tableau 11 Cas d'utilisation change Situation projet.

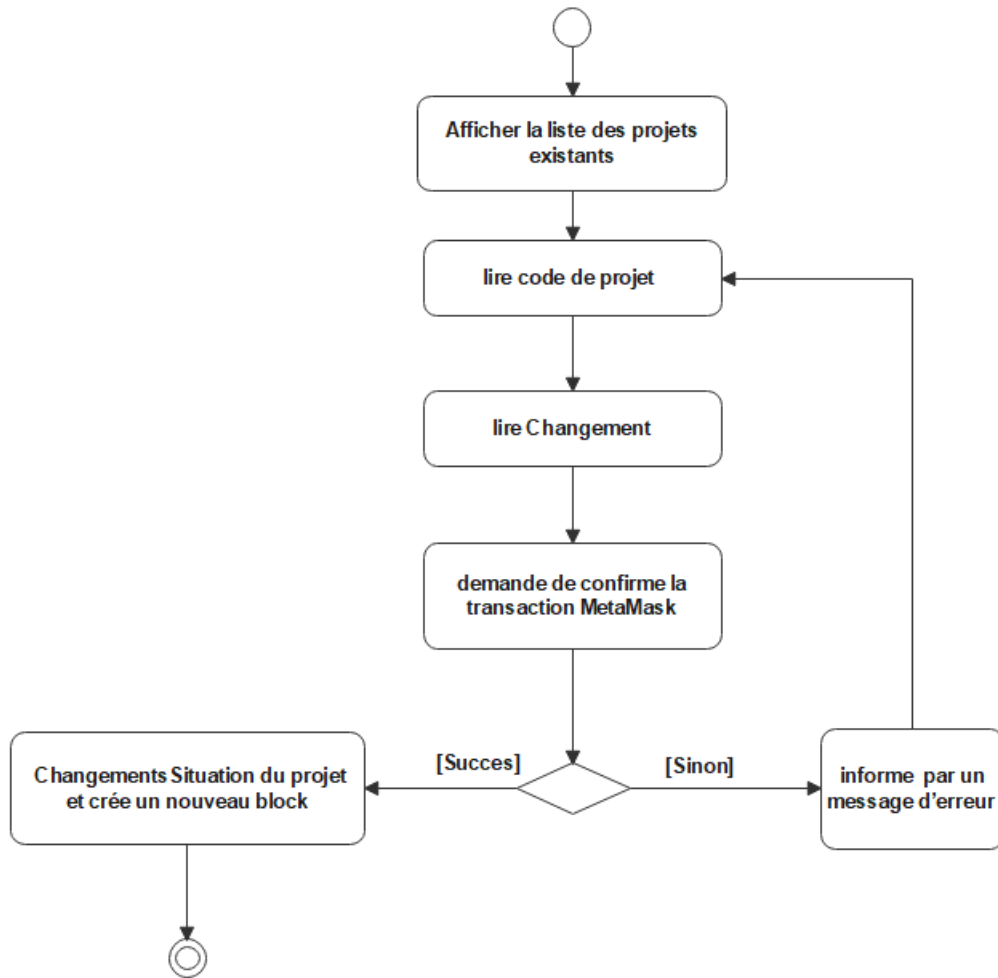


Figure 3.8 Diagramme d'activité du cas « Changement Situation Projet ».

<b>Cas d'utilisation Consulter un projet</b>
<b>Acteur:</b> Administrateur ,Operateur
<b>Résumé :</b> Ce cas d'utilisation permet à l'acteur de Consulter un projet dans le système.
<b>Précondition :</b> l'Actors est authentifié.
<b>Post-Condition :</b> Le détail du projet est affiché.
<b>Description du scenario nominal :</b> <ol style="list-style-type: none"><li>1. L'utilisateur consulte la liste des projets existants</li><li>2. L'utilisateur choisit un projet.</li><li>3. Le système envoie les détails concernant le projet</li></ol>
<b>Description du scenario alternatif :</b> <p>En cas exception : Le système informe l'acteur par un message d'erreur et le scénario reprend au points 2.</p>

Tableau 12 Cas d'utilisation Consulter un projet.

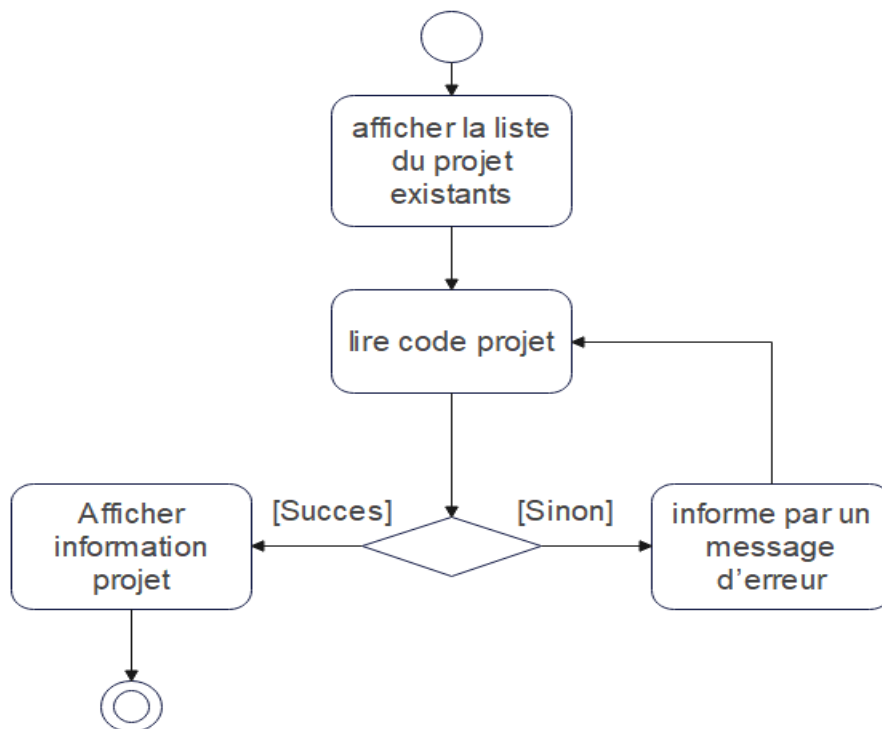


Figure 3.9 Diagramme d'activité du cas « Consulter Projet ».

<b>Cas d'utilisation Rechercher projet</b>
<b>Acteur:</b> administrateur, operateur.
<b>Résumé :</b> Ce cas d'utilisation permet à l'acteur de rechercher un projet dans le système.
<b>Pré-condition :</b> l'acteur est authentifié.
<b>Post-Condition :</b> Projet est trouvé
<b>Description du scénario nominal :</b> <ol style="list-style-type: none"><li>1. L'acteur consulte la liste des projets existants</li><li>2. L'acteur entre le critère de recherche d'un projet et demande la recherche.</li><li>3. Le système effectue la recherche du projet et puis affiche les détails du projet.</li></ol>
<b>Description du scénario alternatif :</b> <p>En cas le projet n'existe pas :</p> <p>Le système informe l'acteur par un message d'erreur et le scénario reprend 2.</p>

Tableau 13 Cas d'utilisation Recherche projet

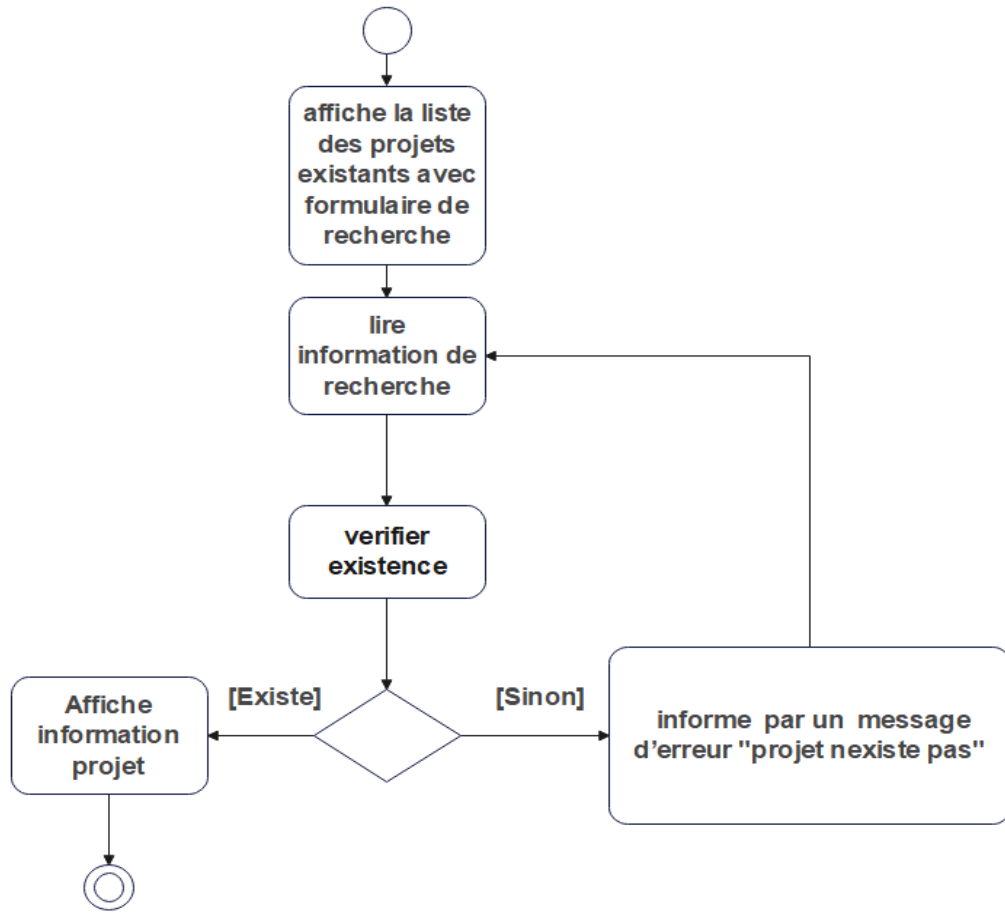


Figure 3.10 Diagramme d'activité de cas « Rechercher Projet ».

Cas d'utilisation consulter statistiques d'un projet
<b>Acteur:</b> Utilisateur
<b>Résumé :</b> Ce cas d'utilisation permet à l'acteur de consulter les statistiques d'un projet.
<b>Pré-condition :</b> l'acteur est authentifié.
<b>Post-Condition :</b> Le détail des statistiques du projet est affiché.
<b>Description du scenario nominal :</b> <ol style="list-style-type: none"> <li>1. L'utilisateur consulte la liste des projets existants.</li> <li>2. Le système affiche la liste des projets enregistrés.</li> <li>3. L'utilisateur choisit un projet afin de visualiser ses chartes graphiques.</li> <li>4. Le système affiche les statistiques détaillées pour le projet concerné.</li> </ol>
<b>Description du scénario alternatif :</b> En cas d'exception : Le système informe l'acteur par un message d'erreur et le scénario reprend au points 3.

Tableau 14 Cas d'utilisation consulter statistiques d'un projet.

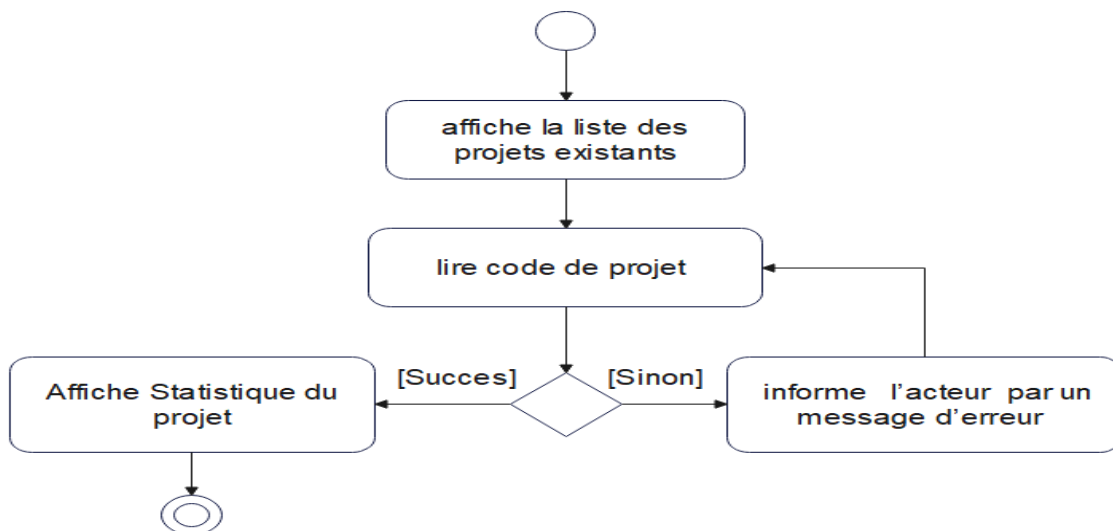


Figure 3.11 Diagramme d'activité du cas « Consulter Statistiques Projet ».

### 3.5. Développements du modèle dynamique

#### 1. Diagramme d'interaction de cas « authentifier »

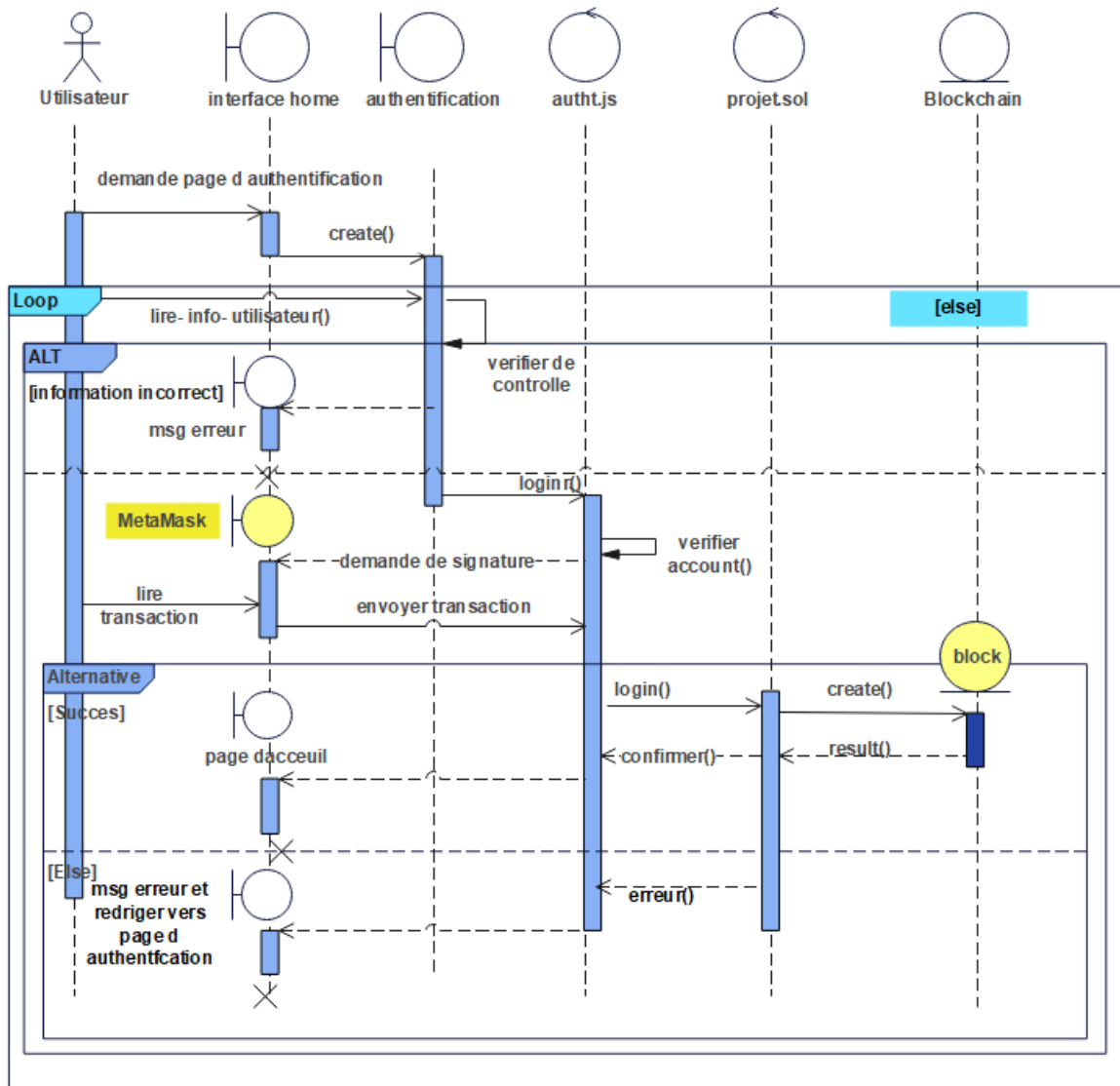


Figure 3.12 Diagramme d'interaction de cas « authentifier ».



2. Diagramme d'interaction de cas « ajouter projet »

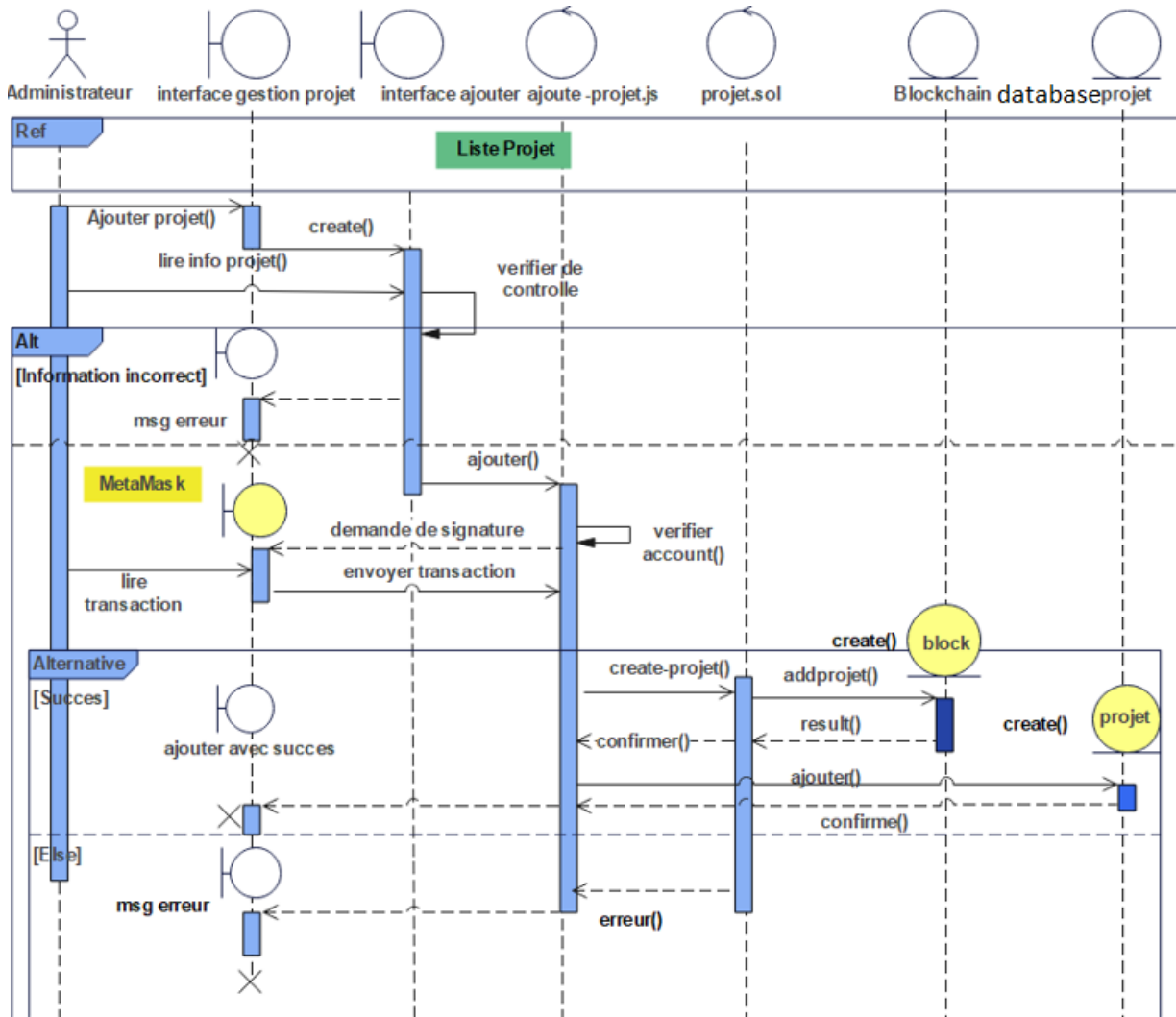


Figure 3.13 Diagramme d'interaction de cas « ajouter projet ».

### 3. Diagramme d'interaction de cas « Modifier projet »

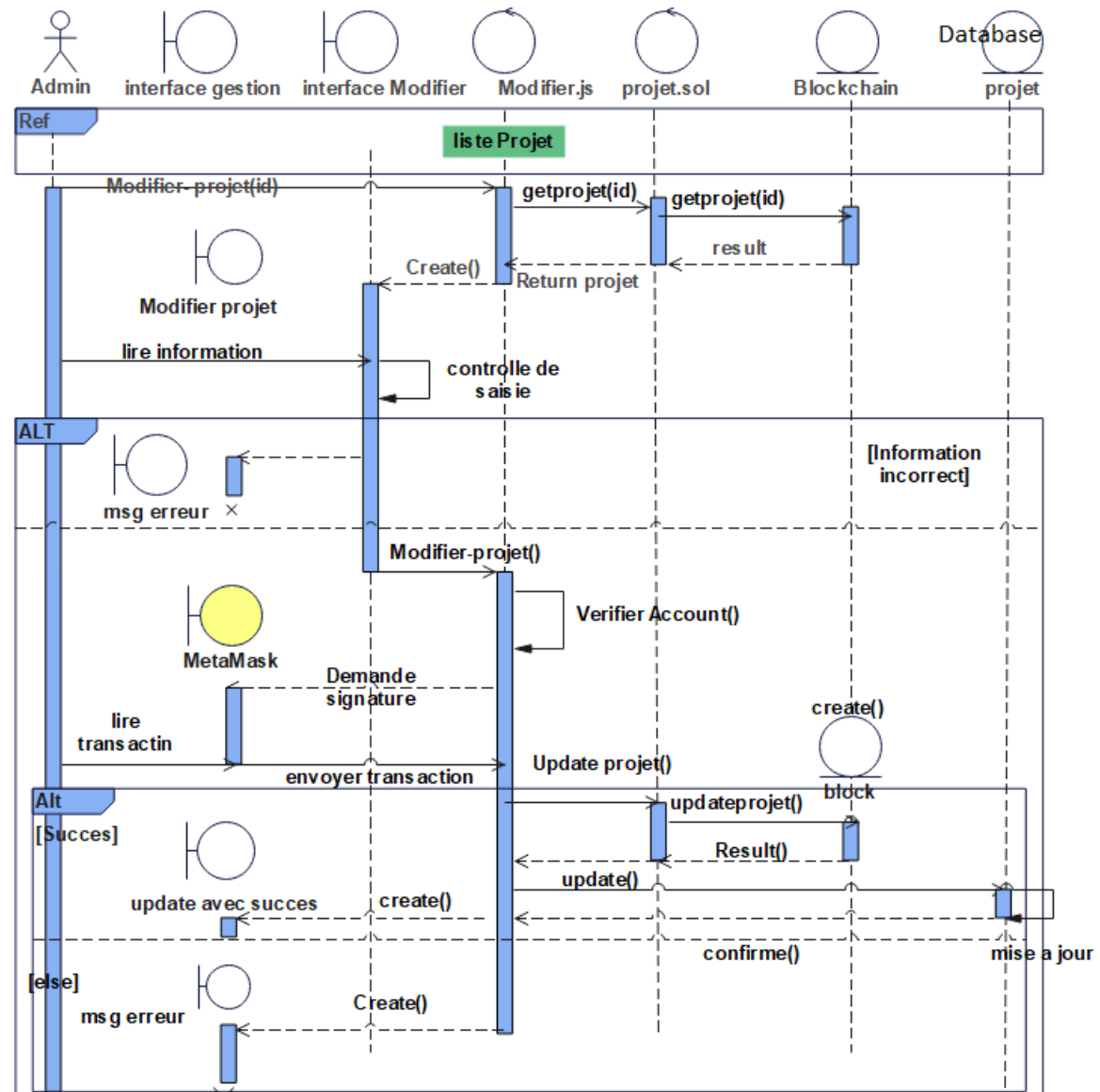


Figure 3.14 Diagramme d'interaction de cas « Modifier projet ».

#### 4. Diagramme d'interaction de cas « Changer Situation projet »

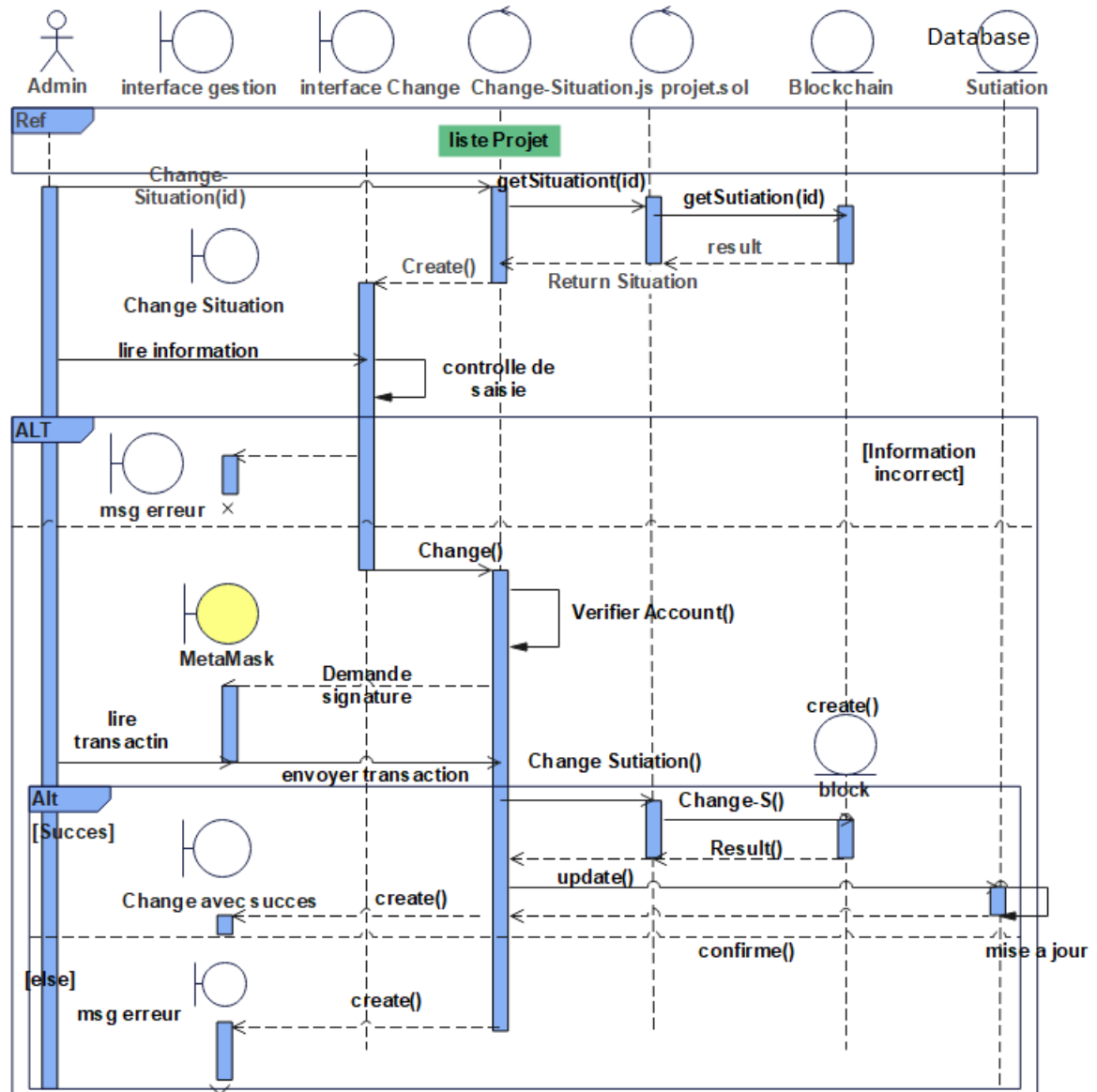


Figure 3.15 Diagramme d'interaction de cas « Change Situation projet ».

### 5. Diagramme d'interaction de cas « Supprimer projet »

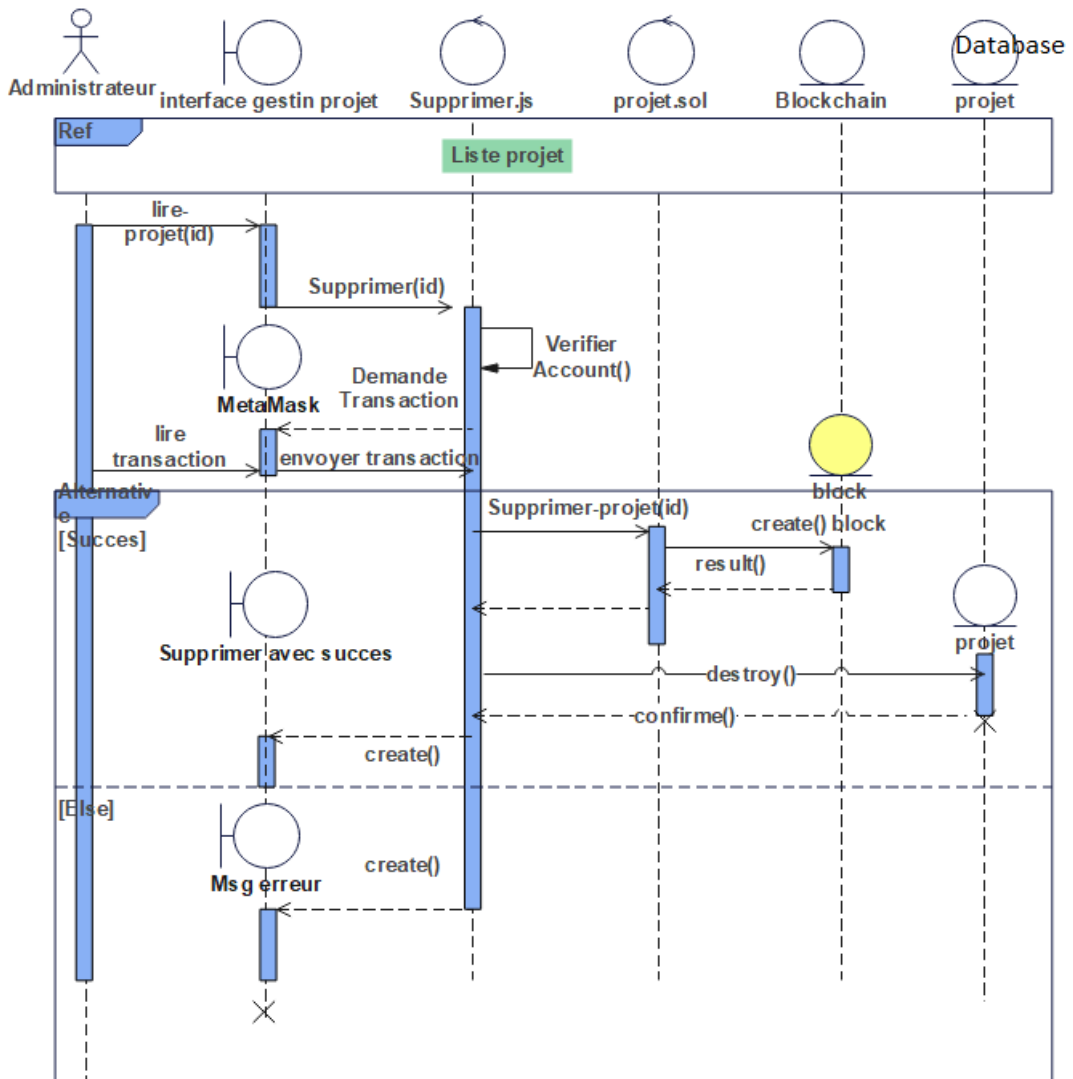


Figure 3.16 Diagramme d'interaction de cas « Supprimer projet ».

### 6. Diagramme d'interaction de cas « Consulter projet »

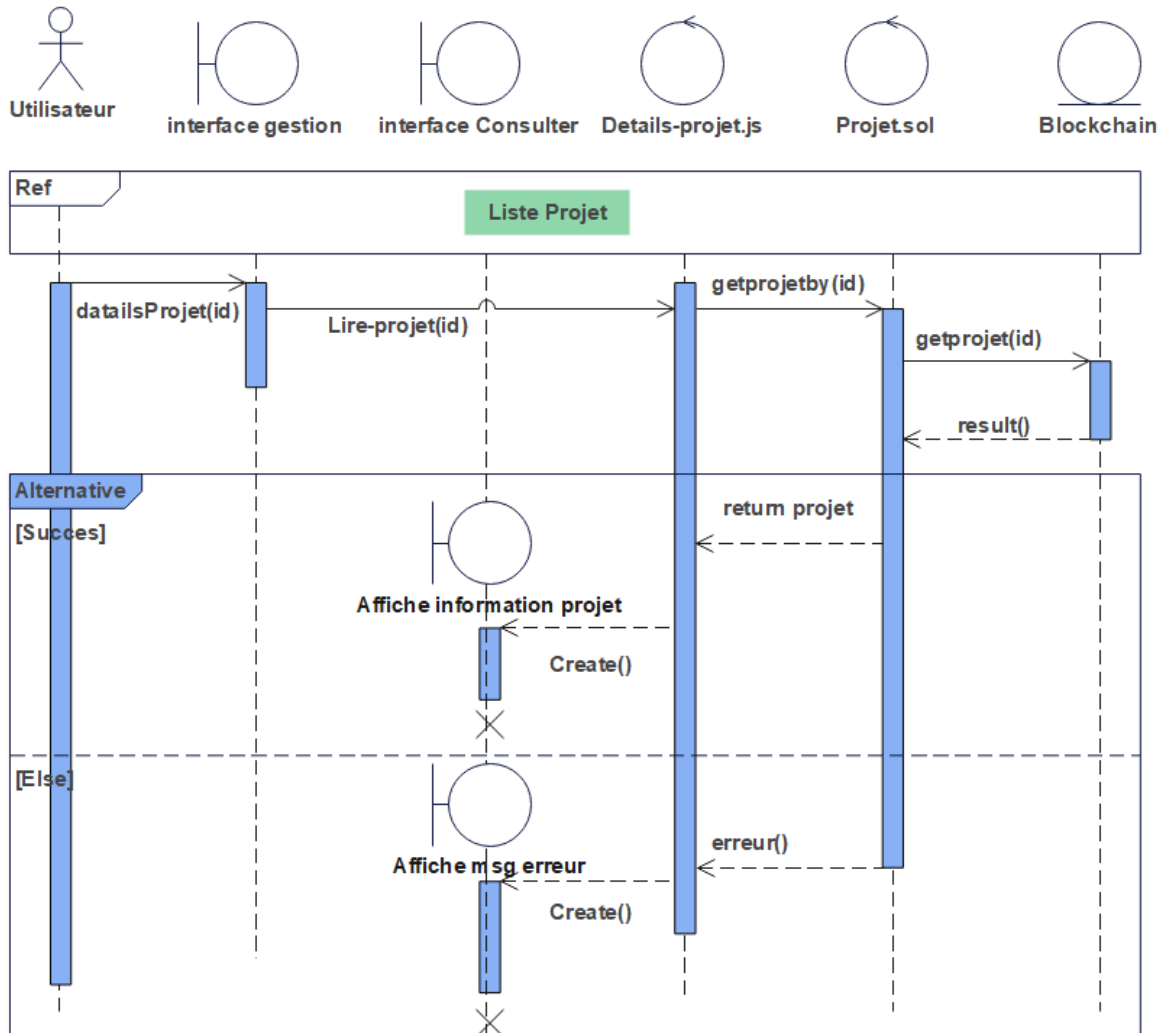


Figure 3.17 Diagramme d'interaction de cas « Consulter projet ».

7. Diagramme d'interaction de cas « Consulter Statistique »

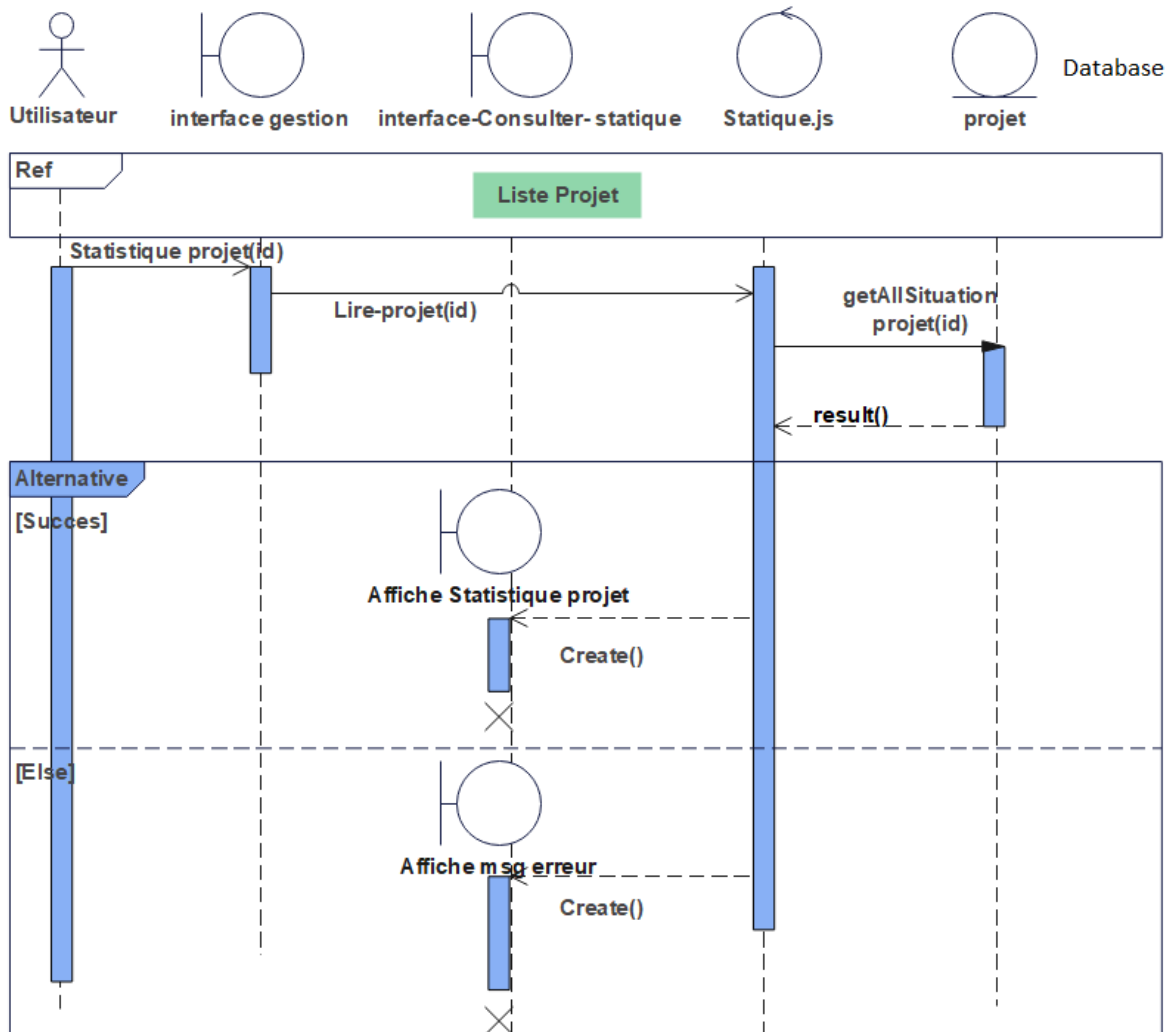


Figure 3.18 Diagramme d'interaction de cas « Consulter Statistique ».

### 8. Diagramme d'interaction de cas « Consulter Liste Projet »

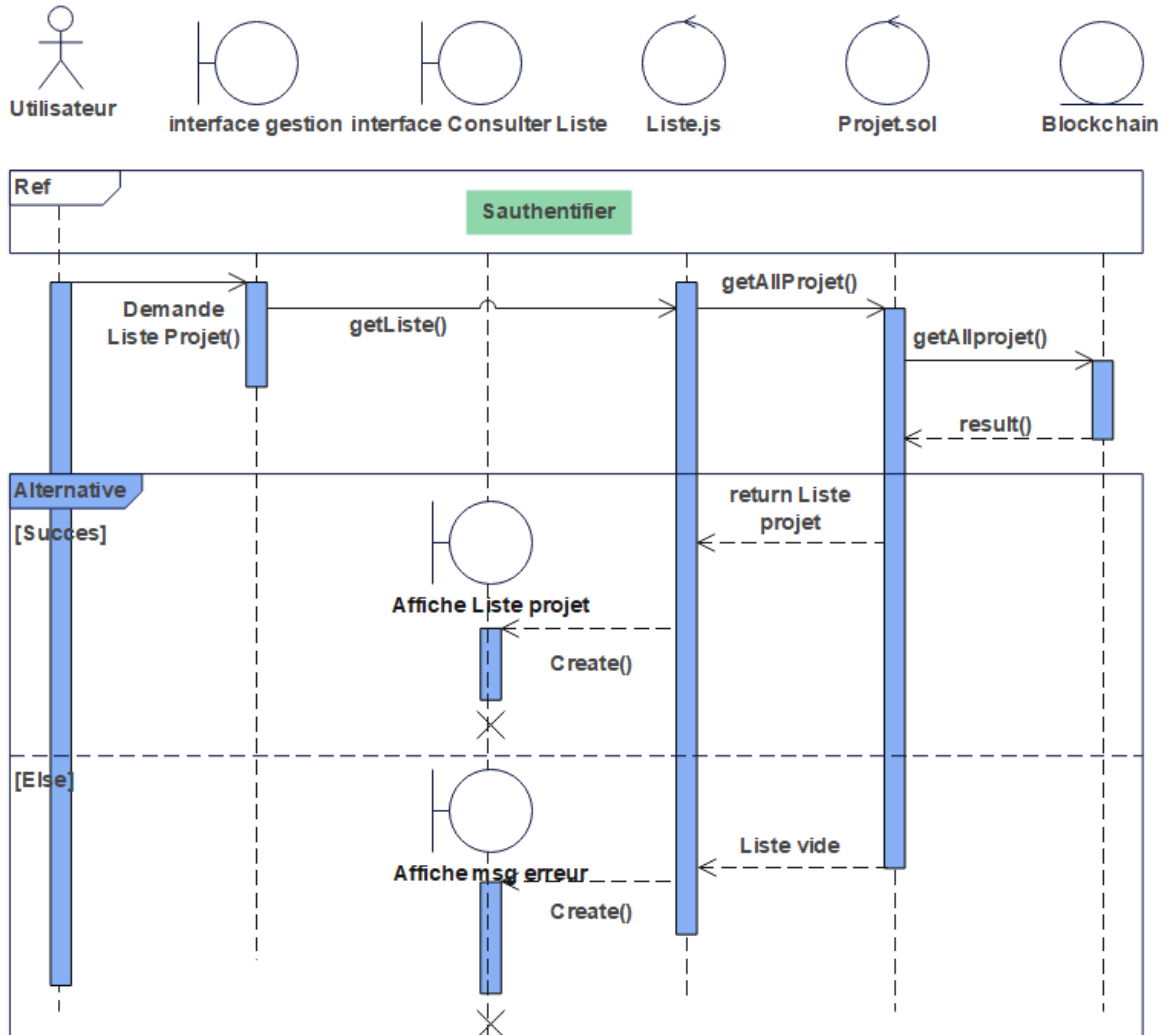


Figure 3.19 Diagramme d'interaction de cas « Consulter Liste Projet ».

### 9. Diagramme d'interaction de cas « Recherche Projet »

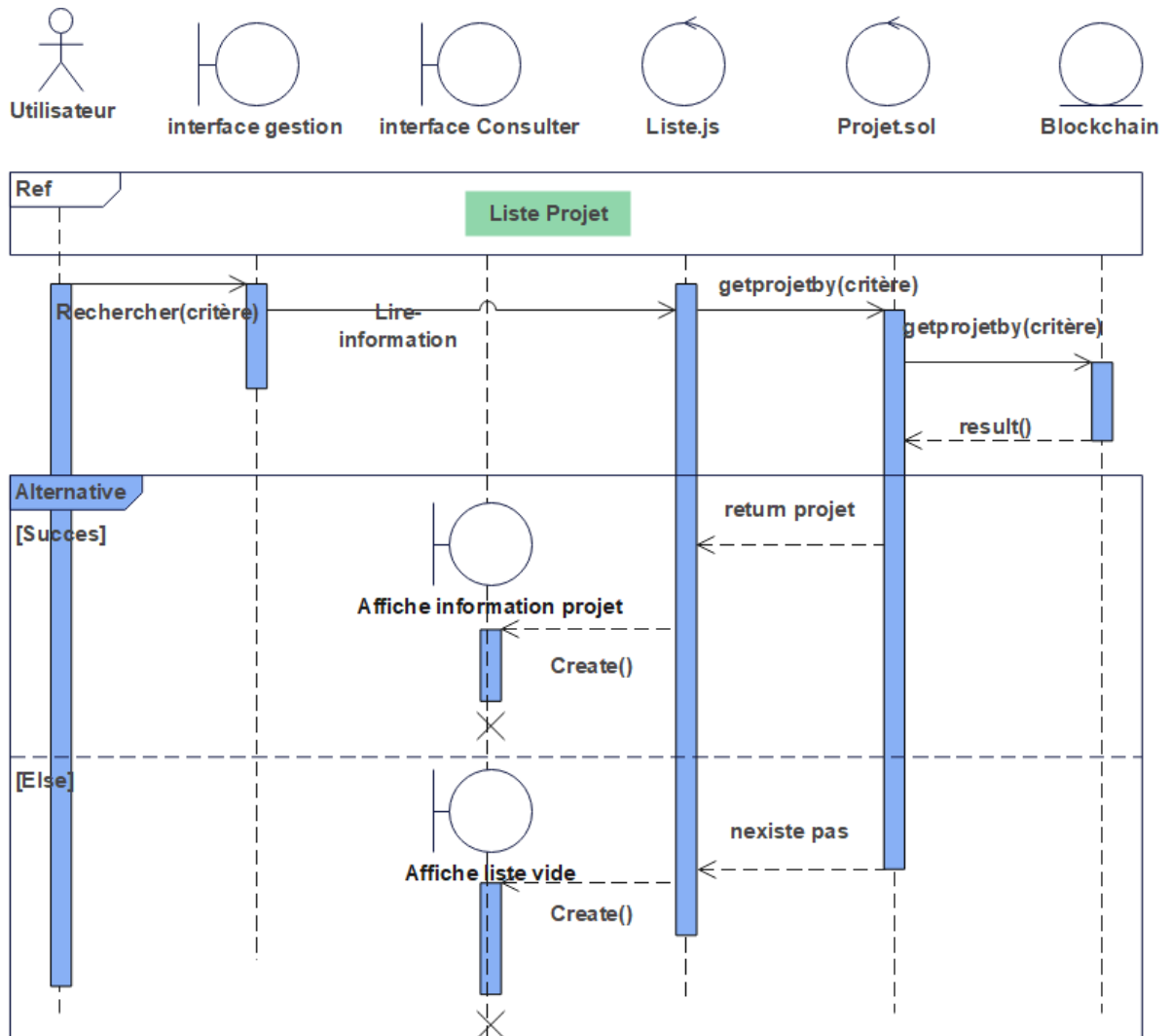


Figure 3.20 Diagramme d'interaction de cas « Recherche Projet »



### 3.6. Conception

#### 3.6.1. Diagramme de déploiement

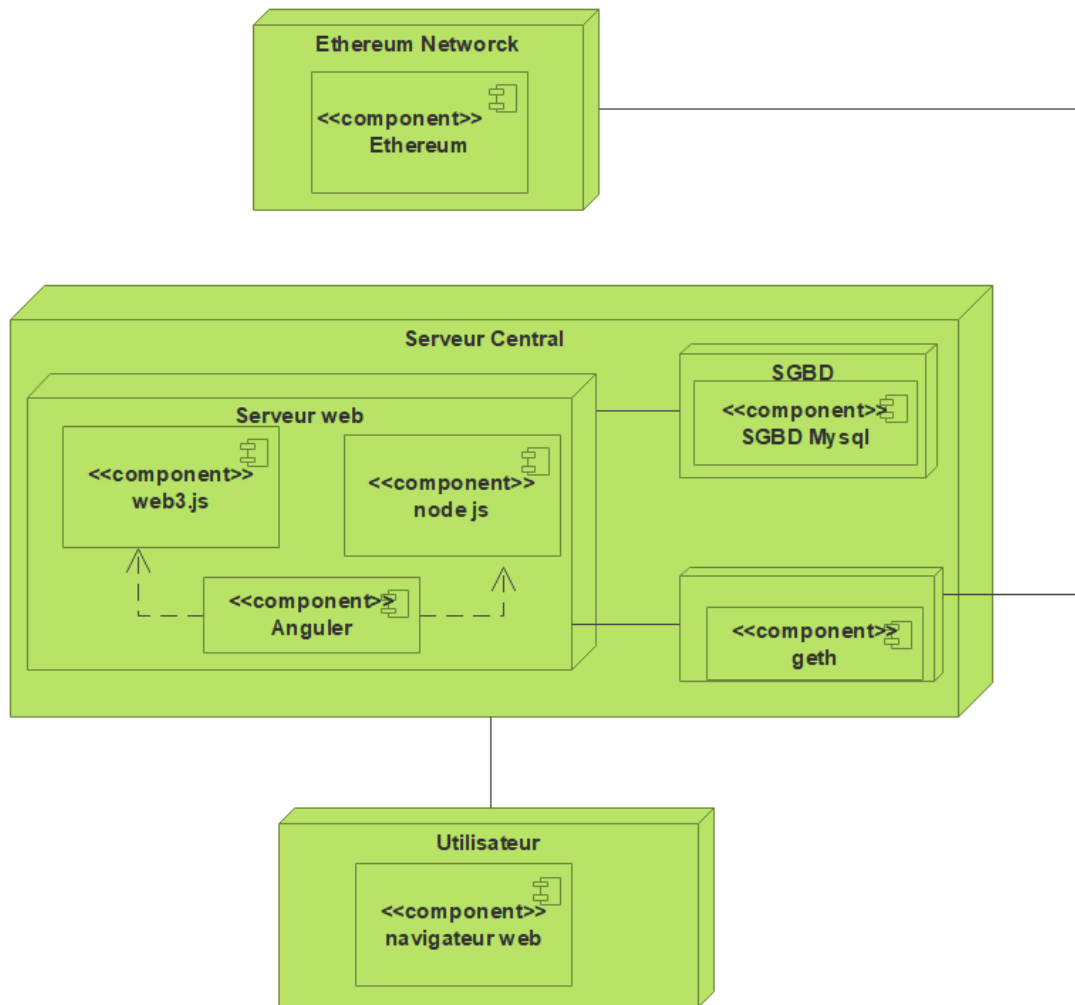


Figure 3.21 Diagramme de déploiement

### 3.6.2. Conception du diagramme de classes

Classe	Description	Codification	Type
Commune	Le numéro de commune Le nom de commune	<b>Commune</b> Libelle	Entier Chaine de caractère
Utilisateur	Le nom Mot de pass Compte MetaMask	Libelle User password	Chaine de caractère Chaine de caractère Chaine de caractère
Rôle	Le numéro Nom du rôle	<b>Rôle</b> Libelle	Entier Chaine de caractère
Projet	Le numéro Nom du projet Prix opération Prix contract	<b>Projet</b> Libelle Prix _opération Prix _contract	Entier Chaine de caractère Double Double
Situation Projet	Le numéro Etat du projet Date de Change Situation projet Taux D'avancements Prix Consommation Prix reste Observation	<b>Situation_Projet</b> Etat Date Avancements Prix Consommation Prix_ reste Observation	Entier Chaine de caractère Date Double Double Double Chaine de caractère
Secteur	Le numéro nom du Secteur	<b>Secteur</b> Libelle	Entier Chaine de caractère
Sous secteur	Le numéro nom du sous-secteur	<b>Sous-secteur</b> Libelle	Entier Chaine de caractère
Chapitre	Le numéro nom chapitre	<b>Chapitre</b> Libelle	Entier Chaine de caractère
Article	Le numéro nom de l'article	<b>Article</b> Libelle	Entier Chaine de caractère

Tableau 15 Description des classes

### Les opérations

Projet	Ajouter Modifier  Supprimer Consulter  Recherche	Ajouter un projet Modifier les informations du projet Supprimer un projet Consulter les informations du projet. Rechercher un projet
Situation projet	Changer Situation Consulter historique	Ajouter Situation projet. Changer Situation projet. Consulter historique projet. Consulter Statistique du projet.
Utilisateur	Ajouter Modifier Supprimer Consulter Recherche	Ajouter un Utilisateur Modifier un Utilisateur Supprimer un Utilisateur Consulter les informations de l'utilisateur. Rechercher un utilisateur

Tableau 16 les operations

### 3.6.3. Diagramme de classes

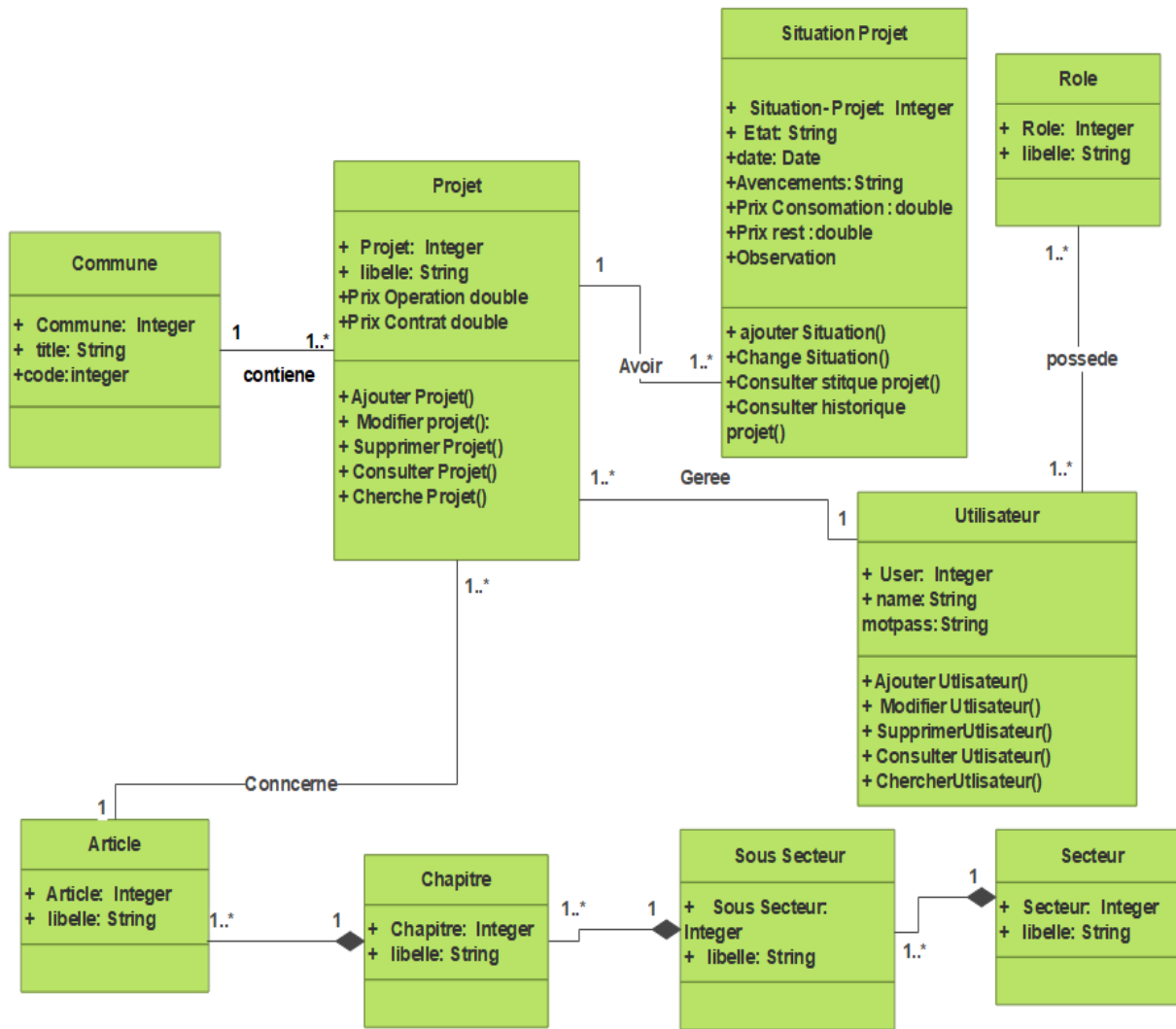


Figure 3.22 Diagramme de classes

### 3.6.4. Passage vers le modèle relationnel

#### Les règles de passages :

La réalisation du modèle relationnel peut être faite à partir de la description conceptuelle. Ce passage de modèle doit respecter un certain nombre de règle.

**Règle 1 :** Toute classe devient une relation, les attributs de la classe deviennent des attributs de la relation, si la classe possède un identifiant il devient la clé primaire de la relation.

**Règle 2 :** pour représenter une association 1 vers 1 (1..1) entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.

**Règle 3 :** pour représenter une association 1 vers plusieurs (1..\*), on procède comme une association 1 vers 1, excepté que c'est forcément la relation du coté plusieurs

qui reçoit comme clé étrangère la clé primaire de la relation du côté 1.

**Règle 4 :** pour représenter une association plusieurs vers plusieurs (\*..\*), il faut introduire une nouvelle relation dont les attributs sont les clés primaires des relations en association, et dont la clé primaire est la concaténation de ces deux attributs.

Si l'association possède des attributs, ils deviennent des attributs de la relation correspondante.

**Règle 5 :** cas de héritage, transformer chaque sous classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.

**Règle 6 :** cas de composition, la clé primaire de la classe composée devient clé étrangère de la classe composant.

**Règle 7 :** cas d'agrégation, le même principe que la Règle 3.

Commune ( <b>commune</b> , libelle), Utilisateur ( <b>User</b> , nom, password), Rôle ( <b>Rôle</b> , libelle), Utilisateur_Rôle ( <b>User</b> , <b>Rôle</b> ), Projet ( <b>Projet</b> , #commune, #User, #Article, libelle, Prix opération, prix contrat), Situation Projet ( <b>Situation Projet</b> , #Projet, Etat, date, avancement, prix consommation, prix reste, observation), Secteur ( <b>Secteur</b> , libelle), Sous-secteur ( <b>Sous-secteur</b> , #Secteur, libelle), Chapitre ( <b>Chapitre</b> , #Sous-secteur, libelle), Article ( <b>Article</b> , # Chapitre, libelle),
--

Tableau 17 Modèle relationnel

### 3.7. Conclusion

Dans ce chapitre, nous avons présenté le principe de fonctionnement et les relations entre les différents acteurs et leurs interactions avec le système, et nous avons pu concevoir une application de la gestion des projets en utilisant les diagrammes UML à savoir le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classes. Dans le chapitre suivant, nous allons présenter les outils de développement ainsi que la réalisation de notre système.

---

## 4. Implémentation du système

### 4.1. Introduction

Dans ce chapitre, nous présentons d'abord l'environnement de travail et les outils de développement utilisés pour construire notre système. Ensuite, nous décrivons l'architecture de notre implémentation ainsi que le contrat intelligent. Enfin, nous donnons quelques captures d'écran des interfaces du système.

### 4.2. Outils de développement et langages utilisés

Premièrement, les principaux outils utilisés lors de la construction de notre système sont les suivants :

#### 4.2.1. Ethereum



Figure 4.1 Logo du Protocole Ethereum

Ethereum est une plateforme logicielle ouverte basée sur la technologie Blockchain présentée en 2013 par V. Buterin, la principale nouveauté qu'elle apporte est le fait de permettre d'utiliser des applications de façon décentralisée, ces applications n'utilisent pas de serveur central. Donc, aucune autorité centrale n'est nécessaire pour les faire fonctionner.

Ces applications sont appelées contrats intelligents, et sont stockées directement dans la Blockchain, donc répliquées sur tous les nœuds du réseau.

Un utilisateur qui souhaite exploiter ces applications doit payer des frais (gaz) puisque les ressources informatiques sont limitées sur Ethereum [25].

Le protocole Ethereum a été choisi pour l'implémentation de notre système cette décision repose sur les raisons suivantes.

- Prise en charge de la création d'applications qui fonctionnent en plus de la technologie de la Blockchain, permettant aux utilisateurs réguliers d'accéder aux données ou d'interagir avec les contrats intelligents grâce à des outils et des cadres qui facilitent le développement.
- De nombreux outils de développement disponibles montrent qu'Ethereum est à la pointe de la technologie d'application décentralisée.

#### 4.2.2. Solidity



Figure 4.2 Logo de langage Solidity

Solidity est un langage de haut niveau orienté objet pour la mise en œuvre de contrats intelligents dans le réseau Ethereum, ce langage a été influencé par C++, Python et Javascript et a été conçu pour viser la machine virtuelle Ethereum [26].

#### 4.2.3. Web3.js



Figure 4.3 Logo de l'API Web3.js

Web3.js est une collection de la bibliothèque API JavaScript qui permet d'interagir avec un nœud Ethereum local ou distant à l'aide des protocoles HTTP, RPC ou Web Socket.

Celle-ci permet d'entrer l'adresse d'un contrat intelligent et d'appeler les fonctions qu'il contient, en passant éventuellement les paramètres nécessaires [27].

#### 4.2.4. Truffle



Figure 4.4 Logo du Framework Truffle

Truffle est le Framework de développement le plus populaire pour Ethereum avec la mission de rendre la vie beaucoup plus facile au développeur Blockchain [28]. Il permet de tester et développer votre contact intelligent. La compilation se fait avec le compilateur solc.

#### 4.2.5. Ganache



Figure 4.5 Logo de logiciel Ganache

Ganache est une Blockchain personnelle pour le développement rapide d'applications décentralisées Ethereum. Il est utilisé pendant tout le cycle de développement et permettant de développer, déployer et tester les applications décentralisées.

Ces outils suffisent pour développer des contrats intelligents en Solidity, les déployer et effectuer quelques tests JavaScript. Ces tests peuvent être utilisés pour valider la bonne exécution des contrats intelligents et mesurer la performance globale du système en termes de coût de transaction, de temps et de stockage des données [29].



#### 4.2.6. Metamask

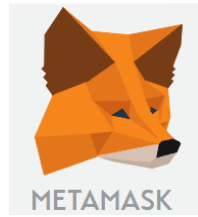


Figure 4.6 Logo du logiciel Metamask

MetaMask est un portefeuille de crypto-monnaie qui peut être facilement utilisé comme extension sur des navigateurs Web tels que Google Chrome et Mozilla Firefox.

Cet outil nous permet d'interagir facilement avec la Blockchain du côté client via le navigateur Web.

Les utilisateurs de notre application peuvent également effectuer des transactions Ethereum en après avoir créé un compte personnel dans Metamask avec une clé privée de compte Ganache [30].

#### 4.2.7. Angular



Figure 4.7 Logo de Framework Angular

Angular est un Framework côté client, open source, basé sur Type Script, et codirigé par l'équipe du projet Angular à Google et par une communauté de particuliers et de sociétés.

Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des *Single Page Applications* des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités [31].

Les outils utilisés pour la réalisation de la deuxième partie de notre application concernant les statistiques sont les suivants.

#### 4.2.8. Node.js



Figure 4.8 Logo de Framework Node.js

Node.js est un environnement d'exécution single-thread, open-source et multiplateformes permettant de créer des applications rapides et évolutives côté serveur et en réseau, nous utilisons cette technologie pour créer une API REST pour être intégrée avec une base de données de type MySQL [32].

#### 4.2.9. XAMPP



Figure 4.9 Logo du logiciel Xampp.

XAMPP est un ensemble de logiciels libres. Le nom est un acronyme venant des initiales de tous les composants de cette suite.

Ce dernier réunit donc le serveur Web Apache, et la base de données relationnelle MySQL [33].

#### 4.2.10. WebStorm



Figure 4.10 Logo du logiciel WebStorm

WebStorm est un environnement de développement intégré pour JavaScript et les technologies connexes. Comme les autres IDE de JetBrains, il rend notre expérience de développement plus conviviale, en automatisant les tâches répétitives et en vous aidant à gérer les tâches complexes avec facilité [34].

### 4.2.11. NPM



Figure 4.11 Logo du gestionnaire des packages NPM

NPM est un gestionnaire de packages pour le langage de programmation JavaScript. Il s'agit du gestionnaire de packages par défaut pour l'environnement d'exécution JavaScript Node.js. Il se compose d'un client de ligne de commande, également appelé *npm*, et d'une base de données en ligne de packages publics et privés payants, appelée registre npm permettant à des milliers de développeurs d'accélérer leurs développements [35].

### 4.2.12. EdrawMax



Figure 4.12 Logo de l'ide des EdrawMax

EdrawMax est un logiciel polyvalent permettant la création de diagrammes qui intègre des fonctionnalités et bibliothèques de tous les autres produits graphiques [36].

## 4.3. Description du système

### 4.3.1. Architecture de l'application web

Pour implémenter notre système, nous avons développé une application Web décentralisée. L'architecture générale de cette application est montrée dans la figure 4.13 et contient trois parties : le front-end, le premier back-end et le deuxième back-end. Dans le front-end (côté client), nous avons utilisé le Framework Angular. Dans le deuxième back-end, nous avons utilisé node.js et XAMPP. Nous avons utilisé la Blockchain Ethereum dans le premier back-end. Le contrat intelligent *Projet. Sol* est chargé de lire et d'écrire des données dans la Blockchain. Le code du contrat intelligent *Projet. Sol* est écrit en langage de programmation Solidity. Ce contrat intelligent sera déployé sur la Blockchain Ethereum.

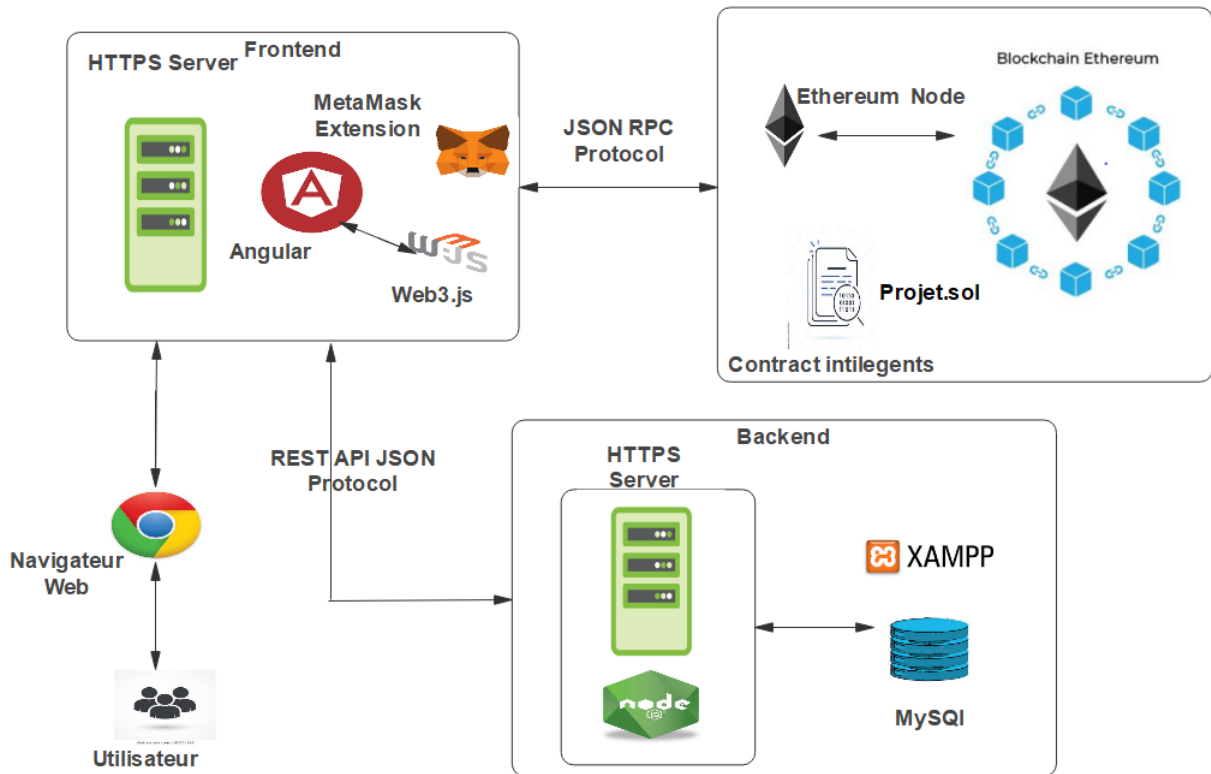


Figure 4.13 Architecture de l'application

### 4.3.2. Contrat intelligent et migration

Dans cette partie, nous présentons le code de notre contrat intelligent *Projet.sol* (Tableaux 18 et 19). Nous déclarons le contrat intelligent avec le mot-clé « **contract** », suivi par son nom « **Projet** ».

Un projet est modélisé avec une structure "Projet" en Solidity, cette structure contient les champs suivants : *id* de type entier non signé, *Title* de type chaîne, *Operation\_Price* de type chaîne et d'autre paramètre. Le variable *counter* représente le nombre de projet existants.

L'événement *ProjectCreated* est utilisé pour déclencher un événement (chaque fois qu'un nouveau projet soit créé, cela mettra à jour notre application côté client).

La fonction *addProjet* est utilisée pour créer un nouveau projet. Elle prend les arguments suivants : *\_title* représente le nom du projet et d'autre paramètre. À l'intérieur de cette fonction, le compteur *counter* est incrémenté pour indiquer qu'un nouveau projet a été ajouté.

### Dapps\backend\truffle\contracts\Projet.sol

```
pragma solidity >0.5.0;
contract Projet {
    uint public counter =0;
    uint public count = 0;
    mapping(uint=> Projet) public projet;
    mapping(uint=> Situation) public situation ;
    mapping(uint => User) public users;
    struct Projet {
        uint id;
        string Title;
        string Operation_Price;
        string Sector;
        string SubSector
        string Chapter;
        string Article;
        string commune;
        string user;
    }
    struct User {
        uint id;
        bytes32 username;
        bytes32 password;
        bytes32 role ;
    }
    struct Situation{
        uint id;
        string date;
        string Consumer_Price;
        string Remaining_Price;
        string state;
        string Advancement;
        string Observation;
        string Pid;
    }
    constructor() {
    }
}
```

Tableau 18 Partie1 du contrat intelligent Projet.sol en Solidity

### Dapps\backend\truffle\contracts\Projet.sol

```
function addProjet( string memory title, string memory operation_price, string memory
operation_contract ,string memory sector, string memory subsector
    string memory chapter, string memory article, string memory commune, string
memory user)
    public returns(bool success) {
        counter++;
        Projet memory _projet = Projet(counter,
            title,
            operation_price,
operation_contract,
            sector,
subsector,
            chapter,
            article,
            commune,
            user);
        projet[counter]= _projet;
        emit ProjectCreated(counter,title);
        return true;
    }
```

Tableau 19 Partie2 du contrat intelligent Projet.sol en Solidity

La fonction login permet aux utilisateurs de s'authentifier. Elle prend les quatre arguments suivants : *\_\_id* représente l'identifiant de l'utilisateur, *\_pass* représente son mot de passe et *\_name* représente son nom. Cette fonction renvoie la valeur « true » si l'identifiant de l'utilisateur, son mot de passe et son nom sont valides. Si non elle retourne la valeur « false ».

#### Dapps\backend\truffle\contracts\Projet.sol\login

```
function login(uint _id ,string memory _username ,string memory _password
,string memory _role ) public view returns(bool){

    require(keccak256(abi.encode(_username))==
users[_id].username,"Incorrect username");

    require(keccak256(abi.encode(_password))==
users[_id].password,"Incorrect password");

        require(keccak256(abi.encode(_role))==
users[_id].password,"Incorrect role");

    return true;

}
```

Tableau 20 Fonction login

Pour déployer le contrat intelligent *projet.sol* sur la Blockchain locale Ganache, nous utilisons la commande : **Truffle migrate**.

### 4.3.3. Front-end Angular et web3.js :

#### Interaction avec le réseau Ethereum.

La librairie web3.js permet de connecter le cote client avec à un nœud Ethereum (Ganache) en utilisant le JSON RPC. Dans ce qui suit, nous montrons comment utiliser la librairie *web3.js* dans notre application cote client

#### Mise en place

La première étape consiste à installer web3.js à l'aide de la commande : **Npm install web3.js** et installer aussi le portefeuille MetaMask dans le navigateur web. Ce portefeuille permet à notre application de demander les comptes (Ganache) des utilisateurs, de lire les données et de suggérer à l'utilisateur de valider des transactions.

#### Interagir avec le contrat

Nous avons créé notre service appelé web3service.ts, dans notre application Angular et importé web3.js dans le fichier de service.

Ensuite, nous avons initialisé web3 et injecter le portefeuille MetaMask voir le (Tableau 21).

Ensuite, nous avons importé le fichier **Projet.json** dans le dossier source de notre application, ce fichier comprend toutes les fonctions disponibles sur le contrat *projet.sol*.

Le fichier doit exporter l'adresse du contrat et l'API afin de les importer dans le *web3Service.ts*.

Nous utilisons la fonction **getContract()** voir le (Tableau 22) , et l'adresse du contrat et l'API importé pour récupérer le variable *contract*.

#### Dapps\frontend\src\app\services\Web3.service

```
import { Injectable } from '@angular/core';
import Web3 from 'web3';
import { Projet } from '../models/projet.model';
declare let window: any;
declare let require: any;
const contractabi = require('../truffle/build/contracts/Projet.json');
@Injectable({
  providedIn: 'root'
})
export class Web3Service {
  public account: string = null;
  public contract: any;
  web3: any;

  constructor() {
    this.loadWeb();
  }
  public async loadWeb(): Promise<any> {
    if (window.ethereum) {
      this.web3 = new Web3(window.ethereum);
      await window.ethereum.enable();
      this.contract = await this.getContract();
    }
  }
  public async verifier
  Account(id): Promise<string> {
    this.web = new Web3(new Web3.providers.HttpProvider('http://localhost:7545'));
    if (this.account == null) {
      this.account = await new Promise(((resolve, reject) => {
        this.web.eth.getAccounts((err, result) => {
          if (err != null){
            return reject(err) ;
          }
        });
      }));
    }
  }
}
```



```
    }
    if ( id === '1'){
    return resolve(result[0]);
    }
    return resolve(result[1]);
  });
  ))) as string;
}
return Promise.resolve(this.account);
}
```

Enfin, nous utilisons ce variable *contract* dans les fonctions qui interagissent avec le contrat intelligent (voir le Tableau 23).

Tableau 21 Code permettant l'initialisation de la librairie Web3.js et l'injection de MetaMask

### Dapps\frontend\src\app\services\Web3.service

```
public getContract(): any {
  return new Promise((resolve) => {
    const web3 = this.web3;
    let networkId;
    web3.eth.net.getId()
      .then((netId: any) => {
        networkId = netId;
        const abi = contractabi.abi;
        const networkAddress= contractabi.networks[networkId].address;
        const contracts = new web3.eth.Contract(abi, networkAddress);
        resolve(contracts);
      });
  });
}
```

Tableau 22 code permettant de retourner le contrat projet.sol

### Dapps\frontend\src\app\services\Web3.service

```
public async CreateProjet( title, operationprice,operationcontract, sector, subsector,
chapter , article, commune, user ): Promise<any> {
  const account = await this.getAccount();
  return new Promise(((resolve, reject) => {
    this.contract.methods.addProjet(title, operationprice, operationcontract, subsector,
sector, chapter, article, commune, user).
    send({ from : account }, (err, result) => {
      if (err != null){
        return reject(err) ;
      }
      return resolve(result);
    });
  }))) as any ;
}
```

Tableau 23 code permettant d'ajouter un projet à la Blockchain

## Interaction avec le node.js

Dans cette partie, nous avons créé la partie relative aux statistiques d'un projet. Nous avons utilisé une *API REST* pour la communication entre le front-end *Angular* et le back-end *Node.JS*.

Nous avons utilisé le Framework Nodejs et Express Javascript pour créer l'API REST pour les opérations de lecture et écriture.

Nous avons utilisé des verbes HTTP standard tels que *get* et *post* pour exprimer les opérations de création et de lecture pour interagir avec la base de données et récupérer toutes les situations d'un projet.

Nous avons créé notre classe *Api Service*, et importé le module *HttpClient* dans notre classe de service qui interagira avec nos points de terminaison Node. JS pour toutes les opérations liées au projet.

Nous avons créé un composant *chart-projet* qui récupérera toutes les situations d'un projet et les affichera sous forme graphiques.

Et enfin, nous avons initialisé le Framework et nous avons lancé notre serveur sur le port 3000 ' `http://localhost:3000/` '.

### Dapps\api-nodejs\index.js

```
const express = require("express");
const bodyparser = require("body-parser");
const cors = require("cors");
const mysql = require("mysql"); //
const app = express();
app.use(cors());
app.use(bodyparser.json());
app.use(express.urlencoded({ extended: true }));
const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "api",
  port: "3306",
});
db.connect((err) => {
  if (err) {
    console.log(err);
  } else {
    console.log("Conected to database succeffly!");
  }
});
app.listen(3000, () => {
  console.log("Server run to port 3000...");
});
```

Tableau 24 code permettant d'initialiser le Framework express et de créer une connexion avec MySQL

Dapps\api-nodejs\index.js

```
app.get("/projets", (req,res) => {  
  
  let qr = "SELECT * FROM projet";  
  
  db. query (qr, (err, result) => {  
  
    if (err) {  
  
      console.log (err, "errs");  
  
    }  
  
    if (result.length> 0) {  
  
res.json(result);  
    }  });  
  
  })  
  
  app.get("/projets/ :id",(req,res) => {  
  
    let id = req.params.id;  
  
    let qr = `SELECT P.id, P.Title , P.OperationPrice,  
P.OperationContract,P.SectorP.SubSector,P.Chapter, P.Article, S.ConsumerPrice,  
S.RemainingPrice, S.Etat, S.Advancement, S.Observation, S.date FROM projet P, situation S  
  
    WHERE P.id = S.projet_id AND P.id = ?`;  
  
    db.query(qr,[id] ,(err,result) => {  
  
      if (err) {  
  
        console.log (err, "errs");  
  
      }  
  
      if (result.length> 0) {  
  
res.json(result);  
      }  
  
    });  
  
  });  
  
});
```

Tableau 25 Les méthodes HTTP utilisées dans API REST

GET - Ceci est utilisé pour fournir un accès en lecture seule à une ressource.

**ACCOUNTS** | BLOCKS | TRANSACTIONS | CONTRACTS | EVENTS | LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK: 1 | GAS PRICE: 2000000000 | GAS LIMIT: 6721975 | HARDFORK: MUIRGLACIER | NETWORK ID: 5777 | RPC SERVER: HTTP://127.0.0.1:7545 | MINING STATUS: AUTOMINING | WORKSPACE: NULL-SENSE

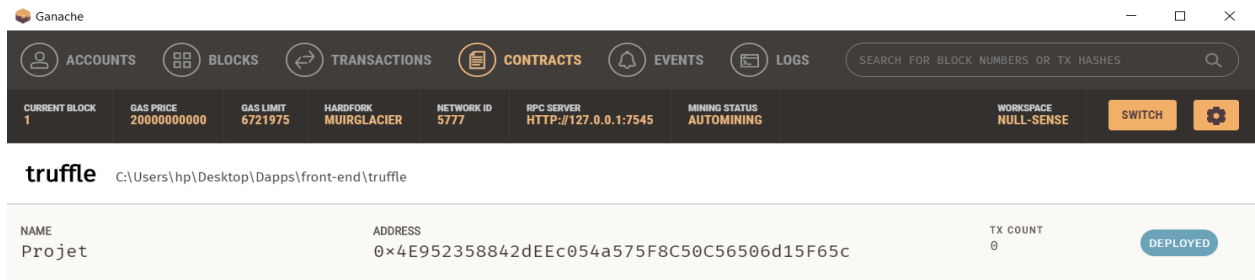
**MNEMONIC** ? rose apology electric coin cash catch rural slight antique kitchen bonus worry

**HD PATH** m/44'/60'/0'/0/account\_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x5f855A598b6363D1e696072aFEbf9A09Ed15cCfB	99.97 ETH	1	0	
0x819C48fcF73386cB78c4B3F8c56C760845282D29	100.00 ETH	0	1	
0xFc3Fe490ebD9853d769c07C296beD47dEC17a1Cb	100.00 ETH	0	2	
0x16035C513419bE7A2a7B7f7A74d1347DfE06cfed	100.00 ETH	0	3	
0x290a65e33e438AA8ae2523b83047B085B285ED23	100.00 ETH	0	4	
0x5382735AD5EB3009118Ecd8017FcADbDC478390F	100.00 ETH	0	5	
0x826b8ae63098BA9ccEA0503Ef7312119021314bE	100.00 ETH	0	6	

Figure 4.14 Génération des comptes Ethereum sur la Blockchain personnelle

Après le lancement de Ganache, il affiche une interface qui contient 10 comptes créés et remplis de 100 ETH pour les opérations initiales et le déploiement des contrats intelligents.



---

Figure 4.15 Déploiement des contrats intelligents sur la Blockchain

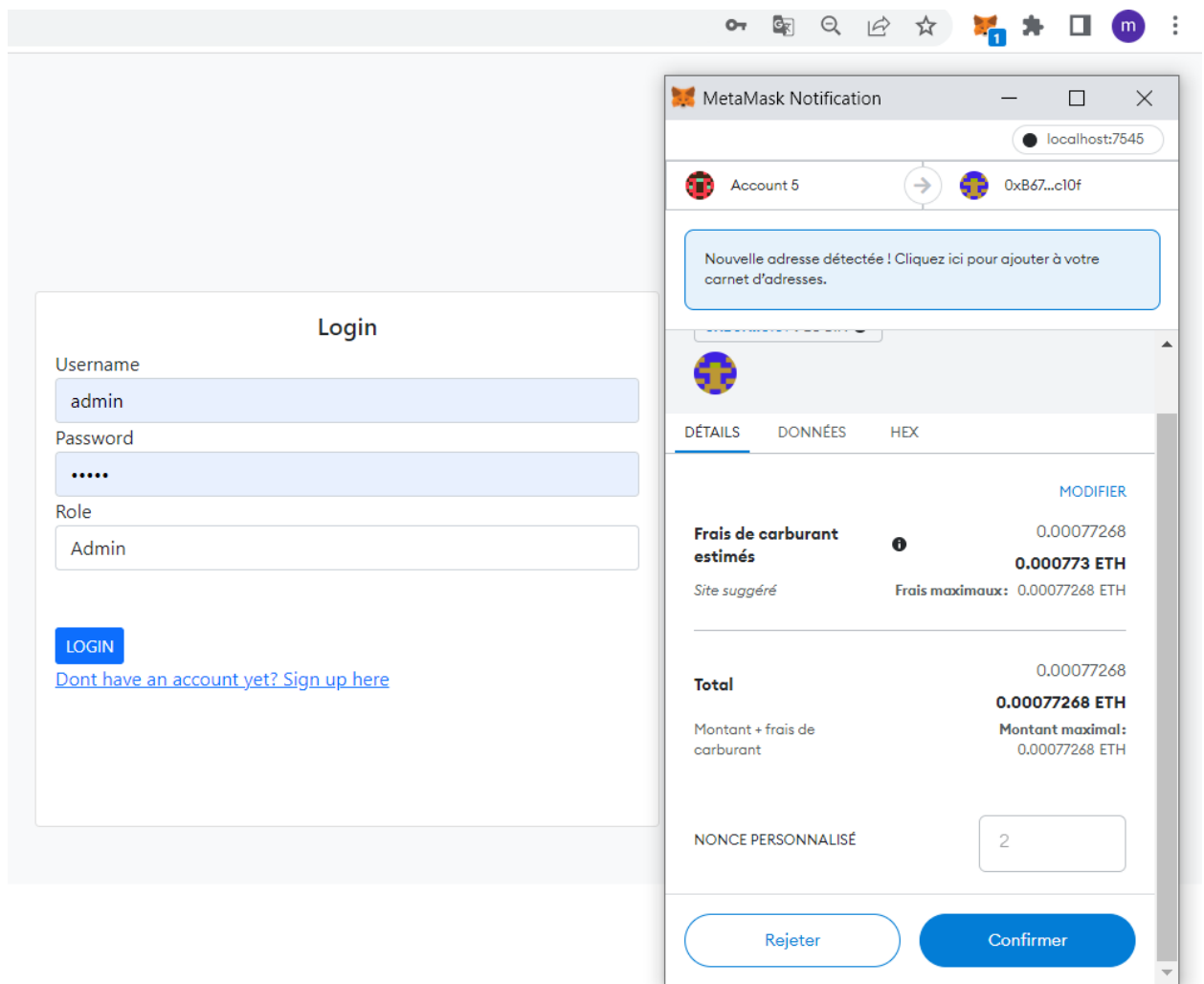


Figure 4.16 Espace d'authentification

Avant toute action, l'administrateur doit passer par l'étape d'authentification qui consiste à saisir ses informations (Figure 4.16), puis confirmer la transaction Metamask pour accéder à la page de l'administrateur.



add Project

Show 4 entries Search:

id	Commune	Title	Operation Price	Operation Contract	Sector	SubSector	Chapter	Article	user	edit	update	delete	new Situation	Statistical
1	sidi Meroune	Achèvement rénovation réseau centre sidi Meroune	2056000 da	2058000 da	Agriculture	Agriculture et hydraulique	Assainissements	Créations neuves	mourad					
2	sidi Meroune	Réalisation Siège Commune de Sidi Merouane	20000000 da	20000000 da	Infrastructures	Infrastructures économiques et administratives	Assainissements	Créations neuves	mourad					
3	sidi Meroune	Réalisation d'une école primaire	40000000 da	40000000 da	Education	Education et Formation	Education et formation	Construction de salles de classes	mourad					
4	sidi Meroune	Réalisation éclairage public sidi Meroune centre	5000000 da	5000000 da	Infrastructures	Infrastructures socio culturelles	Assainissements	Créations neuves	mourad					

Showing 1 to 4 of 4 entries First Previous 1 Next Last

Figure 4.17 Interface Administrateur

Après l'étape d'authentification, l'administrateur consulte la liste des projets sauvegardés dans la Blockchain. En plus, il a la possibilité d'effectuer les opérations suivantes sur un projet : ajouter, modifier, consulter et rechercher. Et chaque opération est une transaction écrite dans la Blockchain de façon permanente et immuable.

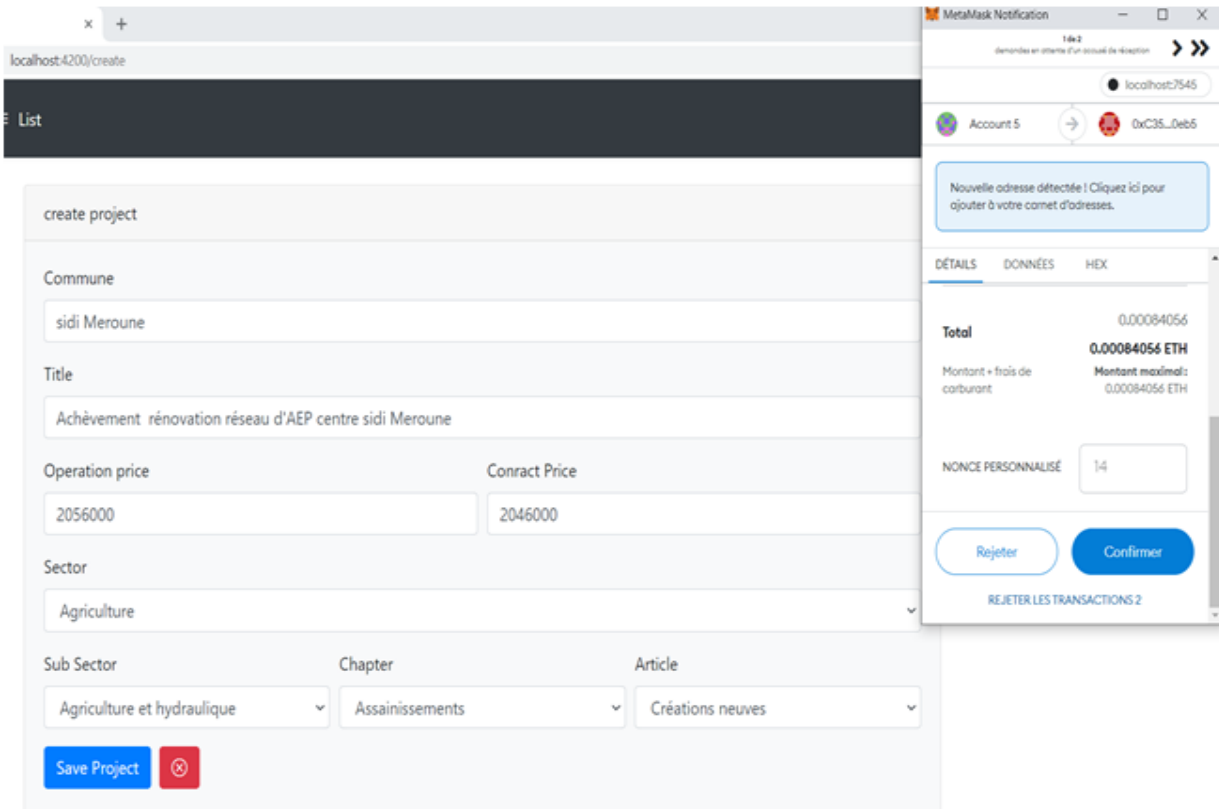


Figure 4.18 Interface permettant d'ajouter des projets

Cette interface permet à l'administrateur d'ajouter un projet à la Blockchain. Cependant, avant l'ajout du projet, il faut confirmer la transaction dans la fenêtre MetaMask.

The image shows the Ganache blockchain explorer interface. The top navigation bar includes 'ACCOUNTS', 'BLOCKS', 'TRANSACTIONS', 'CONTRACTS', 'EVENTS', and 'LOGS'. Below the navigation bar, there is a table of mined blocks. The table has columns for 'BLOCK', 'MINED ON', 'GAS USED', and 'TRANSACTION'. The current block is 10.

CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE DAPPS
10	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	SWITCH
BLOCK 10	MINED ON	2022-06-01 00:30:23			GAS USED	110862	1 TRANSACTION
BLOCK 9	MINED ON	2022-06-01 00:29:44			GAS USED	115062	1 TRANSACTION
BLOCK 8	MINED ON	2022-06-01 00:29:10			GAS USED	115062	1 TRANSACTION
BLOCK 7	MINED ON	2022-06-01 00:28:31			GAS USED	106674	1 TRANSACTION
BLOCK 6	MINED ON	2022-06-01 00:27:19			GAS USED	295223	1 TRANSACTION
BLOCK 5	MINED ON	2022-06-01 00:26:47			GAS USED	295199	1 TRANSACTION
BLOCK 4	MINED ON	2022-06-01 00:25:57			GAS USED	295223	1 TRANSACTION
BLOCK 3	MINED ON	2022-06-01 00:25:14			GAS USED	310127	1 TRANSACTION
BLOCK 2	MINED ON	2022-06-01 00:23:12			GAS USED	25778	1 TRANSACTION
BLOCK 1	MINED ON	2022-06-01 00:19:30			GAS USED	2202306	1 TRANSACTION

Figure 4.19 Enregistrement des données dans les blocs après le minage

Chaque bloc contient des données relatives aux transactions, tous ces blocs sont enchaînés pour créer le grand livre public.

Ce registre public représente toutes les données de la Blockchain qui est protégé par le hachage cryptographique.

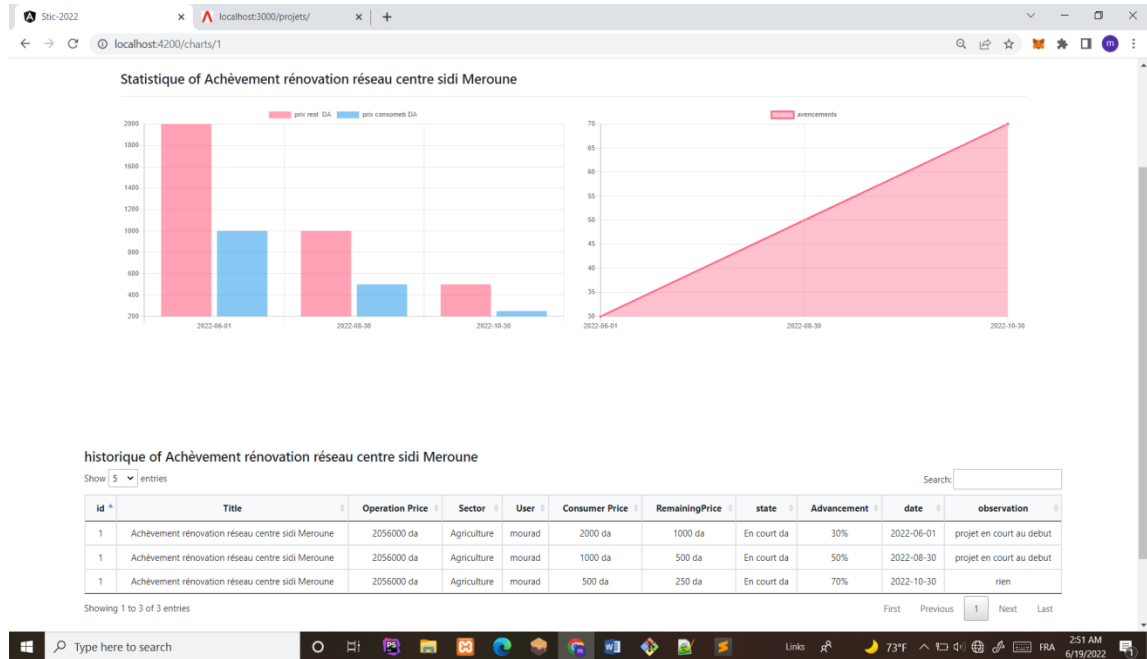


Figure 4.20 Interface Consulter statistiques

Cette interface permet de fournir des statistiques d'un projet sauvegardé dans la Blockchain.

## 4.4. Conclusion

Dans ce chapitre, nous avons présenté le côté implémentation de notre projet en spécifiant les outils, les langages et l'environnement de développement ainsi que les interfaces les plus significatives de notre système.

---

## Conclusion générale



La Blockchain est une technologie de stockage et de transmission des données de manière sécurisée. En plus, cette technologie a amené de nouveaux concepts qui garantissent l'immutabilité et augmentent la sécurité de la donnée, grâce à ses caractéristiques, la technologie Blockchain peut être utilisée dans plusieurs domaines d'application tels que les chaînes d'approvisionnement, le vote électronique sécurisé, l'identité numérique sécurisée et les systèmes financiers.

Dans ce mémoire, nous avons proposé un système d'information sécurisé basé sur la Blockchain pour le suivi des projets au sein de la commune de Sidi Merouane.

Afin de réaliser notre système, nous avons élaboré une conception et une modélisation basée sur l'UP et le langage UML. Nous avons commencé par une étude préliminaire pour identifier les différents acteurs qui interagissent avec le système et procéder à l'analyse des besoins en spécifiant les besoins fonctionnels et non fonctionnels. Ensuite, nous avons décrit le système en utilisant les diagrammes de cas d'utilisation, d'activité, de séquence et de classe. Finalement, nous avons implémenté notre système en utilisant le Framework Angular dans le front-end et la Blockchain Ethereum avec le SGBD MySQL dans le back-end.

Le stage que nous avons effectué au niveau de la commune de Sidi Merouane a été pour nous une expérience très enrichissante et très intéressante, dans la mesure où il nous a permis de mettre en pratique et de développer les connaissances que nous avons acquises dans notre formation universitaire notamment le développement d'applications décentralisées basées sur la Blockchain.

Dans un travail futur, nous planifions de déployer notre système dans toutes les

communes de notre wilaya. Nous planifions également de développer un module d'archivage électronique pour stocker et accéder à tous les documents officiels liés à un projet tels que l'autorisation de programme, l'attestation d'attribution et l'ordre de service.

Enfin, nous prévoyons d'utiliser les techniques de l'apprentissage automatique pour aider les responsables à la prise de décision.

---

## Bibliographie



- [1] A. C. e. M. Bedjaoui, *Conception et réalisation d'une application client/serveur pour le suivi des projets fibres optiques au sein de la Direction Opérationnelle de Télécommunication de la wilaya de Mila*, Centre Universitaire Abdelhafid Boussouf.
- [2] I. AZIZI, *Conception et Réalisation d'une application client/serveur pour la gestion des programmes propres (SONELGAZ)*, Centre Universitaire Abdelhafid Boussouf - Mila, 2015.
- [3] B. S. e. C. Houssam, *Conception et réalisation d'une application client/serveur pour le suivi des projets au sein de la Direction des Travaux Publics (DTP Mila)*, Mila: Centre Universitaire Abdelhafid Boussouf – Mila, 2017.
- [4] J. L. e. S. Moussati, *Réalisation et développement d'une plateforme pour la gestion et le suivi des projets des Entreprises*, Maroc: Université Abdelmalek Essaadi, 2016.
- [5] J.-P. Delahaye, *Les Blockchains, clefs d'un nouveau monde*, Pour la Science, 2015, p. 449.
- [6] A. F. a. M. N. D. Chaum, *Écrivain, Untraceable electronic cash*. [Performance].
- [7] D. Chaum, *Blind signatures for untraceable payments*, Springer, 1983, p. 199–203.
- [8] S. Nakamoto, *Bitcoin : A peer-to-peer electronic cash system*, Manubot, 2019, pp. 15, 25 et 46.
- [9] V. B. e. al, «Ethereum white paper,» 2013. [En ligne]. Available: <https://github.com/>.
- [10] C. Balva, «Qu'est-ce que la blockchain,» [En ligne]. Available: <https://blockchainfrance.net/decouvrir-la-blockchain/c-est-quoi-la-blockchain>. [Accès le 25 03 2022].

- [11] D. PERARD, *Blockchain et stockage efficace*, 16 décembre 2020.
- [12] A. B. e. al, *Hashcash-a denial of service counter-measure*, 2002.
- [13] S. Nakamoto, «Bitcoin: A peer-to-peer electronic cash system,» [En ligne]. Available: <https://bitcoin.org/bitcoin.pdf>.
- [14] M. Swan, *Blockchain Blueprint for a new economy*, 2015.
- [15] [En ligne]. Available: <https://www.hyperledger.org/>.
- [16] [En ligne]. Available: <https://litecoin.com/>.
- [17] [En ligne]. Available: <https://ripple.com>.
- [18] *Comprendre la blockchain*, Livre blanc sous licence Creative Commons, uchange.co, janvier 2016, p. 10.
- [19] C. F. Plisson, *La blockchain, un bouleversement économique, juridique voire sociétal*, vol. 54, I2D, mars 2017, pp. 20-22.
- [20] «Blockchain et droit d'auteur,» [En ligne]. Available: <http://www.master.masterpia.com>. [Accès le 2022 05 21].
- [21] D. T. Research, *BLOCKCHAIN IN LOGISTICS*, 2018.
- [22] R. o. E. E-Residency, «The new digital nation,» [En ligne]. Available: <https://e-resident.gov.ee/>.
- [23] M. Pilkington, *Blockchain Technology: Principles and Applications*, 2016.
- [24] P. Roques, *les Cahiers du Programmeur UML 2 Modéliser une application web*, 4, Éd., 02/10/2008.
- [25] [En ligne]. Available: <https://ethereum.org/en/>.
- [26] [En ligne]. Available: <https://docs.soliditylang.org/en/v0.8.14/>.
- [27] [En ligne]. Available: <https://web3js.readthedocs.io/en/v1.7.3/>.
- [28] [En ligne]. Available: <https://trufflesuite.com/>.
- [29] [En ligne]. Available: <https://trufflesuite.com/ganache/>.
- [30] [En ligne]. Available: <https://sites.google.com/metamaskexts.com/metamask-extension/home>.
- [31] [En ligne]. Available: <https://angular.io/>.

- [32] [En ligne]. Available: <https://nodejs.org/en/>.
- [33] [En ligne]. Available: <https://www.apachefriends.org/fr/index.html>.
- [34] [En ligne]. Available: <https://www.jetbrains.com/fr-fr/webstorm/>.
- [35] [En ligne]. Available: <https://docs.npmjs.com/>.
- [36] [En ligne]. Available: <https://www.edrawsoft.com/fr/download-edrawmax.html>.
- [37] B. A. A. BARDIS, Conception et Réalisation system vote électronique basé Blockchain, Faculté des Nouvelles Technologies de l'Information et de la Communication, 2021.
- [38] K. E. B. Y. CHERIF MEGOURA, La Blockchain et l'IoT pour une gestion sécurisée, Faculté des Nouvelles Technologies de l'Information et de la Communication, 2020.



---

# Acronymes



<b>STIC</b>	Sciences et Technologies de l'information et de la communication
<b>UML</b>	Unified Modeling Language
<b>UP</b>	Unified Process