MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

**Faculté des Sciences Exactes et d'Informatique**

**Département de Mathématiques et informatique**

**Filière : Informatique**

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Réseaux et Systèmes**

Présenté par :

**Maïga Mamadou Idrissa**

THEME :

# La Sélection des candidats à l'aide de la Blockchain

Devant le jury composé de :

| | | |
|---|---|---|
| Mme BENHAMED S. | Université de Mostaganem | Présidente |
| M. MIROUD M | Université de Mostaganem | Examinateur |
| M. BESSAOUD K | Université de Mostaganem | Encadrante |

Année Universitaire 2020-2021

# REMERCIEMENTS

**Dieu merci !** Je vais bien ! Nous ne voyons que ce que nous avons pas et oublions toujours ce que nous avons. Je remercie Le Tout Puissant pour toutes ces choses que j'ai. Louange à Celui qui détient en sa main la royauté sur toute chose.

Je remercie mes parents pour tout ce qu'ils firent et continuent à faire pour moi, des actes que seul Dieu peut récompenser. Je remercie mes frères et sœurs pour leur soutien et leur encouragement.

J'exprime ma profonde gratitude à mon encadrant **Mr. Bessaoud Karim** pour sa disponibilité et ses valeureux conseils et bien plus encore... Ils m'a poussé à toujours faire plus, toujours faire mieux. A ses côtés j'ai appris plusieurs choses essentielles dans les études et dans la vie en général. Puisse Dieu lui accorder les bienfaits de ce monde et ceux de celui d'après.

Mes remerciements à tous les enseignants de la faculté, pour les efforts, les conseils et pour l'accompagnement durant ces cinq dernières années.

# DEDICACES

Je dedie ce modeste travail :

A mes parents,

A mes frères et soeurs,

A mes amis...

Puisse Dieu les combler de tous les bienfaits de ce monde et de ceux de celui d'après.

Maïga Mamadou Idrissa

# Résumé

La « e-démocratie » implique l'utilisation des TIC pour renforcer les principes démocratiques existants. Quand ces TICs sont appliquées dans les élections, on parle de *e-élection*. Les e-élections actuelles ne satisfont ni les électeurs ni les candidats, elles sont trop souvent jugées faibles du point de vue de la sécurité et de la transparence. La Blockchain est l'une des plus grandes innovations du monde informatique. Du fait de son contrôle sans faille de l'intégrité des données et du fait de son caractère décentralisé, elle est de plus en plus utilisée dans les secteurs où la confiance est d'or. Ces deux points font de cette technologie la pièce manquante du puzzle pour garantir la transparence et la sécurité à nos e-élections. Notre travail porte sur l'application de la Blockchain à ces e-élections, plus précisément à des collectes de signatures, que ça soit pour des pétitions de référendum d'initiatives populaires ou pour le parrainage de candidats d'une élection présidentielle.



Mots clés : **Vote Électronique, E-voting, Blockchain, I-voting, Collecte de signature, e-Pétition**.



# Abstract

E-democracy involves the use of ICTs to strengthen existing democratic principles. When these TICs are applied in elections, we speak of e-elections. The current e-elections do not satisfy voters or candidates, they are often seen not secure and not transparent. The Blockchain is one of the greatest innovations in the IT world. Due to its powerful control of data integrity and its decentralized nature, it is increasingly used in industries where trust is needed. These two points make this technology the missing piece of the puzzle to ensure transparency and security in signatures collection. In our work, we propose an innovative model for signatures collection. Our model relies on permissioned Blockchain and smart contracts to ensure verifiability and auditability of signature collection and e-petitions.



Mots clés : **Electronic voting, E-voting, Blockchain, I-voting, e-Pétition**.

# TABLE DES FIGURES

# TABLE DES MATIÈRES

# INTRODUCTION GÉNÉRALE

Les élections jouent un rôle très important dans l'avenir d'un pays, elles permettent de mettre les personnes qu'il faut à la place qu'il faut. Les élections sont le principal pilier de la démocratie. Elles ne doivent cependant, pas être prises à la légère.

L'utilisation de la technologie dans les élections est jusqu'à aujourd'hui toujours mal accueillie, non seulement par les candidats mais aussi les électeurs, leur principale inquiétude étant la sécurité. Un bon point ici est que le vote électronique ou *e-voting* n'est pas un secteur nouveau, il a longtemps fait l'objet de plusieurs études et a été implémenté dans plusieurs pays à travers le monde.

Les méthodes de sécurité utilisées jusqu'à aujourd'hui garantissent-elles vraiment et définitivement l'intégrité des élections ?
Et si l'ordinateur était utilisé comme méthode de surveillance civile ?
Est il possible d'empêcher et d'exclure la manipulation des résultats des élections ?
La principale question à laquelle il faut répondre au final est la suivante : **pouvons-nous, devons-nous et voulons-nous exposer nos élections à de tels risques ?**.

Les candidats à des élections démocratiques doivent remplir un certain nombre de critères, qui sont relatifs à chaque pays et à chaque type d'élection. Ces critères sont appelés « critères d'éligibilité ». Pour la plupart des cas, certains points comme l'age, la nationalité, les qualifications (diplômes, certifications etc.) sont pris en compte. Ces critères permettent de placer les candidats dans un cadre de base. Pour une élection présidentielle, ce cadre correspondrait à des milliers sinon à des millions de personnes. Comment organiser une élection présidentielle avec des millions de

candidats ? Pour éviter les candidatures fantaisistes et ajouter de la crédibilité aux candidatures, certains pays utilisent un système de parrainage ou de signature. Les candidats doivent collecter un nombre de signatures de personnes (citoyens lambdas, élus, anciens membres du gouvernement etc.) pour valider leurs candidatures. D'une part, cette technique permet de donner la possibilité aux citoyens d'exprimer leur accord et d'autre part de réduire le nombre de candidats. Le système des signatures en lui même est très efficace, le principal problème est le facteur humain. Les signatures sont collectées et vérifiées par des humains, des irrégularités peuvent survenir. Est-il possible d'y soustraire l'humain ? Si oui, comment le faire ? Et si le processus électoral était complètement transparent ? Pourquoi ne pas soumettre toutes ses données à la vérification publique ? Ceci pourrait il contribuer à rendre les élections plus sûres ?

« La Blockchain est une nouvelle technologie de base de données s'appuyant et tirant pleinement profit d'Internet, du protocole libre, de la puissance de calcul et de la cryptographie. Cette base de données transactionnelle distribuée est comparable à un grand livre comptable dans lequel chaque nouvelle transaction est écrite à la suite des autres, sans avoir la possibilité de modifier ou d'effacer les précédentes. Ce registre est actif, chronologique, distribué, vérifiable et protégé contre la falsification par un système de confiance repartie (consensus) entre les membres ou participants (nœuds) » [10].

Le but de notre travail est d'étudier l'applicabilité de la technologie Blockchain dans le processus électoral, nous voulons fournir un modèle informatique de collecte de signatures pour un système de parrainage dans les élections basé sur la Blockchain. Ce travail se fera en deux principales étapes : premièrement nous commencerons par faire une étude des concepts clés de notre sujet : notamment le vote électronique et la Blockchain. Puis nous essayerons de fournir notre modèle de collecte de signature en nous basant sur ces notions clés. Ce rapport se divisera donc naturellement en quatre grands chapitres : dans le premier nous parlerons d'élections et de vote électronique, dans le second nous verrons le principe de fonctionnement de la Blockchain, puis le chapitre suivant sera consacré a notre modèle de collecte de signature ; enfin en dernier dans le chapitre nous étudierons la faisabilité de notre modèle en proposant une implémentation de celui-ci en utilisant la Blockchain Ethereum.

CHAPITRE 1

# VOTE ÉLECTRONIQUE

## Introduction

De nos jours, la plupart des élections dans le monde utilisent les TIC d'une certaine manière, au moins pour le calcul des votes. Cette adaptation est le résultat d'une longue période d'évolution au cours de laquelle non seulement les procédures mais aussi les moyens technologiques de vote ont changé considérablement.

Le vote électronique, *e-voting* en anglais, est un vote soutenu par des appareils électroniques. Ces appareils peuvent servir à enregistrer les votes, les compter ou les transmettre à travers Internet [14]. Bien que tous les pays n'ont pas encore expérimenté le concept, il existe de nombreux travaux et recherches sur l'utilisation des nouvelles technologies dans les élections.

Certains considèrent le vote électronique comme un outil pour faire progresser la démocratie, augmenter la confiance dans la gestion électorale et faire tomber les barrières entravant la participation politique de la population. D'autres gardent un point de vue plus pessimiste, le vote électronique n'est qu'un autre moyen de plus pour manipuler les élections et il ne touche que les rares déjà engagés dans la politique.

Bien que les technologies ouvrent de nouvelles possibilités pour le processus électoral, en particulier pour les opérations de vote et l'authentification des électeurs, il peut y avoir des risques imprévus donc une attention particulière doit y être accordée.

Dans ce chapitre nous allons étudier le vote électronique sous plusieurs aspects : démocratique, technique, social, financier, sécurité, faisabilité etc. Nous présenterons au mieux les rouages, les concepts ainsi que les différentes catégories de dispositifs utilisés. Puis nous tenterons de faire une étude comparative entre le vote électronique et le vote classique (”*sur papier*”). Ce n’est qu’alors que nous pourrons juger de la viabilité d’un tel système.

## 1.1 Élections

Avant de plonger dans le vif du sujet, essayons de nous mettre dans le contexte réel des élections et de recenser quelques points clés. « Les élections sont un élément fondamental des valeurs et des principes de la démocratie. Elles permettent aux individus d’identifier et de développer leurs préférences politiques, de prendre part au processus politique et d’exiger que leurs représentants rendent des comptes, sans craindre la répression ou la violence. Elles offrent aux citoyens la possibilité de discuter, de débattre et de se former aux grandes questions de la gouvernance et démontrent que la liberté et la transparence de la compétition et de la campagne politique sont tout aussi importantes que l’acte de vote lui-même. » [1]

L’élection est un acte positif en soi, mais elle perd son sens si cet acte soulève trop de difficultés ou si, une fois exprimé, le vote n’a aucun effet sur la façon dont la nation est gouvernée. **Faciliter** le vote, c’est rédiger des bulletins intelligibles, s’assurer que les bureaux soient accessibles, que les listes des candidats soient à jour, et que le caractère confidentiel du vote soit garanti à l’électeur.

Ce rapport ne jugera pas de l’efficacité des systèmes électoraux existants, le sujet étant trop vaste et trop subjectif, nous en conserverons une vue assez généraliste.

## 1.2 Parrainage ou collecte de signatures

### 1.2.1 Définition

Le parrainage peut être défini comme l’appui moral prêté par une personne d’autorité à une œuvre, ou soutien d’une personne qui demande à être admise dans une société, un ordre. Ainsi, le parrainage est la désignation par un élu et/ou un citoyen d’un candidat à une élection selon la

législation électorale en vigueur. Il s'agit par conséquent d'un critère d'éligibilité [13].

### 1.2.2 Types de parrainage

Il y a principalement trois types de parrainage :

**Parrainage citoyen**

Le parrainage citoyen pour les élections est un mode de sélection des candidats ou une candidature est validée lorsqu'un certain nombre de citoyens donnant leur accord à cette candidature est atteint. Il est mieux accueilli par la population, car ici tout le monde peut s'exprimer, aucune différence n'est faite entre les signataires.

**Parrainage d'élus**

Dans ce cas de figure, ce sont les actuelles membres du gouvernement qui sont mis en avant. Les candidats doivent bénéficier du soutien de députés, de conseillers communaux ou de ministres etc.

**Parrainage mixte**

Il arrive aussi que la loi électorale demande plusieurs types de parrainage, mais que les signatures soient comptées séparément selon les types d'électeurs (citoyen lambda, élus, ex-membres du gouvernement etc.). Ce type de parrainage ajoute un attribut à la signature, nous en viendrons ultérieurement.

**Pétitions**

Les pétitions sont des demandes de faveur ou de réparation d'une injustice, dirigées vers une autorité établie. Au sens courant, une pétition est un document adressé à une autorité publique et signé par de nombreuses personnes. La signature de pétition est une forme plus générale de système de parrainage. Son étude nous à permis de mettre la lumière sur d'autres critères de validité des signatures que nous détaillerons juste après.

### 1.2.3 Critères de validité des signatures

Les conditions pour qu'une signature de parrainage pour une élection soit valide sont généralement déterminées par la loi électorale. Cette loi dépend de la constitution, donc les dites

conditions diffèrent par pays. Cependant il nous est tout de même possible d'en retenir quelques points communs.

- **L'Éligibilité** dans le sens ou une signature doit appartenir à une personne vivante , dont l'identité est vérifiable et jouissant de tous ses droits selon la loi électorale. Dans le cas du parrainage d'élus ou du parrainage mixte, cette propriété peut être plus restrictive en tenant compte du statut social ou de la fonction occupée par le signataire.
- **Vérifiabilité individuelle et universelle** : Le signataire doit pouvoir s'assurer que sa signature et ceux des autres ont bien été prises en compte.
- **Unicité** de la signature, pour une signature il faut trouver un et un seul signataire.
- **Intégrité** de la signature.
- **L'intention** soutient l'idée selon laquelle si le vérificateur peut raisonnablement conclure qu'un électeur avait l'intention de signer, sa signature devrait être comptée et ceci même si cette signature n'est pas conforme.
- **Répartition** implique que les signatures soient **également** ou **équitablement** reparties sur toute l'étendu du territoire. Dans le premier cas, un même nombre de signatures pour toutes les régions et dans le second le nombre de signatures pour chaque région est calculée en fonction de sa population. Cette condition ajoute un nouvel attribut à la signature : **la localisation**.

## 1.3 Exigences relatives au vote électronique

Les exigences auxquelles les procédures de vote électronique doivent répondre diffèrent selon le contexte. Mais dans la plupart des cas, les obstacles ne sont pas suffisamment fondamentaux pour empêcher des solutions communes. Pour visualiser ces exigences, nous allons imaginer le scenario suivant :

Monsieur « *BonCitoyen* » désire se présenter à son bureau de vote un matin de lundi pour voter pour son parti favori que nous appellerons « *BonParti* ». On lui annonce une bonne nouvelle, il lui est désormais possible de voter à avec son smartphone à l'aide de son numéro de sécurité sociale. *BonCitoyen* n'étant pas un adepte des nouvelles technologies, donc un peu **sceptique** sur la chose, décide quand même de se rendre sur la plate-forme Web de vote que nous nommons« e-

Election ». La première question qu'il se pose est la suivante : *Est-ce que e-Election est-elle sûre ?* Monsieur *BonCitoyen* aimerait que son vote ne soit **pas modifier**, qu'il soit **pris en compte** mais sans qu'il ne soit **connu des autres**. Puisqu'il n y a plus de bureaux de vote, il n y a donc plus d'autorité de surveillance. *BonCitoyen* qui n'est pas si bon finalement décide de **voter autant de fois qu'il le peut**. Par crainte de se faire prendre, il arrête et **oblige** toute sa famille même les enfants à voter pour son parti.

Les élections s'annoncent mal pour *BonParti*, ses membres ayant beaucoup dépensé pendant la campagne ne veulent et ne peuvent pas se résoudre à perdre cette fois. L'exécutif du parti décide au final **d'influencer** les élections d'une manière ou d'une autre.

A partir de cet exemple nous pouvons relever les points suivants :

1. **Le Secret du vote** comprend la confidentialité et l'anonymat, il est là pour protéger une personne contre toute pression ou influence contre sa libre expression de préférence politique. C'est un moyen de garantir la liberté de choix. Le système ne doit permettre en aucun cas qu'on puisse relier un vote à un votant et inversement.
2. **L'Éligibilité** regroupe l'authentification et l'autorisation. Le système doit permettre uniquement aux personnes jouissant de leur droit de vote de voter.
3. **L'Incoercibilité** suppose qu'une personne ou même une autorité ne devrait pas être en mesure d'extraire la valeur du vote ou de contraindre un électeur à voter.
4. Le système ne doit en aucun favoriser un candidat à un autre ou un électeur à un autre, on dira dans ce sens qu'il rempli la condition d'**Équité**.
5. Tous les éléments qui participent au processus électoral (y compris le logiciel) doivent (du moins en principe) être facile à comprendre et à vérifier pour tous les citoyens, ces conditions sont vérifiées par un système complètement **Transparent**.
6. La **Précision** dans le calcul des résultats : tous les votes exprimés jugés valides et seulement ceux-ci, doivent être comptés. Pris individuellement, aucun vote ne doit être compté plus d'une fois (unicité).
7. L'**Intégrité** des données : Aucun vote ne doit pouvoir être modifié sans détection et possibilité de réparer la manipulation.
8. Aucun parti politique ou une autorité ne doit pouvoir perturber (déni de services) ou influencer les résultats des élections. Dans ce cas, le système est dit : **robuste**.

A la liste des exigences citées vient s'ajouter une autre notion : ***Software Independence*** en français « Indépendance du Logiciel », un système de vote est indépendant du logiciel si un changement (non détecté) ou une erreur dans son logiciel ne peut pas provoquer un changement indétectable ou une erreur dans le résultat des élections. Un système de vote qui n'est pas indépendant du logiciel et donc *dépendant du logiciel*, est, dans un sens, vulnérable aux erreurs de programmation non détectées et aux codes malveillants [5].

## 1.4 Historique

L'histoire du vote électronique remonte au moins à deux siècles. Officiellement elle débute avec la proposition de loi de l'anglais GEORGE GROTE en 1836 qui présentait une machine à voter. Malgré le fait que sa proposition ne fut jamais acceptée, cette machine marqua le début de l'ère des machines à voter mécaniques. La proposition à ensuite été développée pour donner d'autres machines plus performantes et plus sophistiquées. Un inventeur d'origine polonaise, JAN JOZEF BARANOWSKI, a proposé une nouvelle machine prenant en charge le vote et le dépouillement des votes dans les bureaux de vote en utilisant le concept des machines à calculer [1] à Paris en 1849 . Son objectif était de minimiser les erreurs humaines lors du comptage. D'autres machines sont apparues avec le temps dans les autres parties du monde principalement aux États-Unis.

Depuis l'apparition des premières machines à voter mécaniques, l'électrotechnique s'est développée rapidement. Partout dans le monde, des inventeurs ont contacté leurs parlements respectifs pour suggérer des méthodes d'enregistrement électrique de votes. MARTIN DE BRETTES a approché le Sénat français avec un plan pour une machine qui enregistrerait les votes au parlement en utilisant l'électricité. Il l'a appelé « Appareil pour voter, indiquer, autographier et contrôler les votes ». D'autres propositions sont venues de CLÉRAC et GUICHENOT en 1870, JACQUIN en 1874, et un an plus tard de MORIN ainsi que LALOY etc .
Tous ceux-ci n'eurent aucun impact impactes sur les procédures électorales puisque la plupart de ces machines ne furent jamais utilisées. Ceux-ci marquèrent le début de l'ère des **machines électroniques de votes**.

Ce n'est que pendant les années 70 que les premiers **DRE** commencèrent à apparaître, no-

1. Une machine conçue pour simplifier et fiabiliser des opérations de calculs, et dont le fonctionnement est principalement mécanique.

tamment aux États-Unis, Au Pays-Bas en 1989, en Belgique 1994, au Brésil en 1998 et en Inde en 1999 [9].

Avec la mobilité croissante des électeurs, les administrations électorales ont été obligées d'offrir d'autres possibilités de participation lorsque les électeurs n'étaient pas présents chez eux ou étaient incapables de se présenter aux bureaux de votes. Cette nécessité a donné naissance à une nouvelle catégorie de vote électronique : **le vote en ligne**. Certains pays ont suscité l'intérêt de toute la communauté politique mondiale, notamment le cas de l'Estonie dont nous discutons ci après.

### 1.4.1 L'Estonie et le vote par Internet

Un système de vote électronique a été utilisé en Estonie en 2005 dans des élections parlementaires dans le but de faciliter la participation des populations au processus électorale. Le système est présumé très fiable par les autorités du pays, il est selon eux, aussi sûr que le vote classique. Malheureusement la grande majorité du monde politique et scientifique ne partageait pas le même avis. Étant l'un des premiers du genre, le vote électronique estonien a naturellement suscité l'intérêt du monde entier. Cela fait du *e-voting* estonien un cas d'étude unique et très important dans la sécurité électorale. [17]

#### I-Voting et les exigences relatives au vote sur Internet

L'électeur peut voter à plusieurs reprises par voie électronique ; seul le dernier vote exprimé est compté. Un électeur qui a été contraint de voter peut voter à nouveau après s'être libéré du coercitif, annulant le vote émis sous pression. En cas de problème grave du système, les administrateurs peuvent annuler les votes par Internet. Ceux qui ont voté par Internet peuvent à nouveau voter dans les bureaux de vote, le vote sur papier étant prioritaire et ayant lieu plusieurs jours après. Ce principe protège également les électeurs contre la coercition. Le vote est chiffré avec la clé publique du votant puis déchiffré a la réception avec sa clé privée . L'électeur confirme son choix avec une signature numérique légalement acceptée, cette signature numérique est dérivée des infos sur sa carte ID. Il a la possibilité de vérifier son i-vote (vote par Internet) à l'aide d'un appareil séparé (téléphone portable, tablette etc). De cette manière, il est possible d'augmenter la probabilité de détection d'attaques dirigées contre le système (principalement contre l'ordinateur de l'électeur).

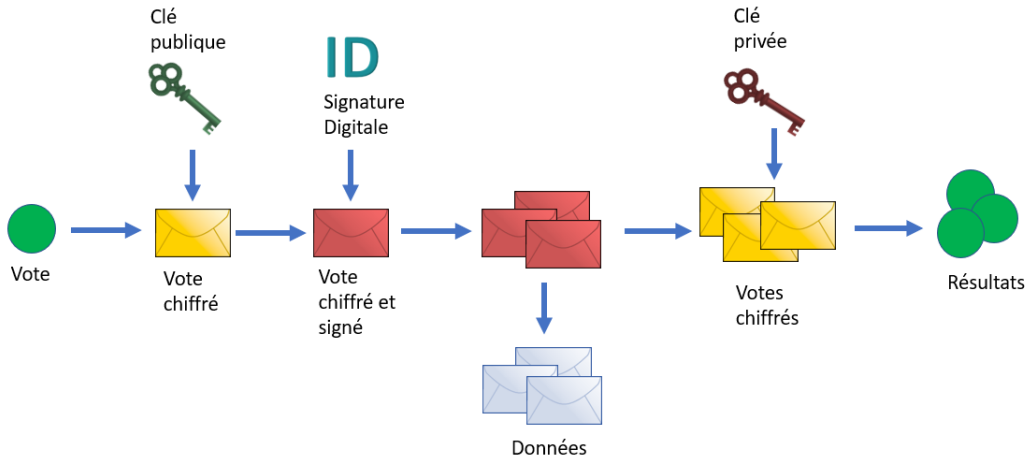


FIGURE 1.1 – Processus de vote : Estonie [2]

Malheureusement le I-Voting estonien présente plusieurs manquements :

— Les responsables électoraux ont rendu tout le processus de vote aussi transparent que possible, ils ont permis une observation publique, ils ont publié des vidéos des tâches des administrateurs et une grande partie du code source du serveur. Toutes ces mesures ont malheureusement été insuffisantes pour établir pleinement l'intégrité des résultats des élections.

— De nombreux bugs et défauts de programmations dans plusieurs versions du code source ont été révélés [16].

— La population doit aveuglement faire confiance à la poignée d'administrateurs gérant tout le processus électoral.

### 1.4.2 Le vote électronique en Inde

Dans les années 90, la **C**ommission **E**lectorale **I**ndienne **(ECI)** a introduit des machines à voter électroniques **(EVM)** pour lutter contre les fraudes électorales et simplifier la procédure électorale. De nombreux problèmes existaient avec l'utilisation bulletins de vote (papier), notamment la difficulté d'organiser les élections dans un pays aussi grand (en terme de population) et le coût que ça impliquerait [4].

- Le premier point était d'empêcher la prise en otage de bureaux de vote, dans lesquels des fraudeurs prenaient de force un isoloir et bourraient l'urne de vote. Les EVMs étaient conçus

pour décourager de telles fraudes en limitant le nombre de vote à la minute (5 en 2017) [4]. Cette fonctionnalité augmentait le temps nécessaire pour lancer de faux bulletins.

- Deuxièmement, avec les machines à voter, les administrateur peuvent arrêter le système pour éviter que des intrus ne prennent le contrôle de l'isoloir.
- Troisièmement, pendant le vote électronique, les signatures ou les empreintes du pouce des électeurs sont conservées dans un base de données qui est ouverte à l'inspection du public ou de toute personne désireuse de faire une demande de contestation des résultats des élections.
- Enfin, la validation des votes sont entre les mains des officiers, ceci diminue la probabilité de fraude.

La commission électorale a affirmé que les votes enregistrés dans les machines sont inviolables et que l'altération physique des appareils est facilement détectable. Cependant, ces affirmations ont plus tard été contestées par des évaluations rigoureuses indépendantes [19]. Des études ont prouvé que ces machines étaient vulnérables a des attaques pouvant modifier les résultats des élections et violer le secret de vote [19].

## 1.5 Catégories de systèmes de vote électronique

Les systèmes de vote électronique remplissent de nombreuses fonctions : l'authentification des électeurs, le calcul des résultats, le cryptage et la transmission des votes etc. Ces différentes fonctionnalités permettent de les repartir en plusieurs groupes [8] :

**Optical Mark Recognition systems (OMR)**

Un électeur utilise une carte de vote avec les noms des candidats imprimés dessus. A cote de chaque nom se trouve une case qu'il peut cocher. A la fin des élections, un dispositif de reconnaissance optique permet de relever le vote et l'enregistrer dans la base de données.

Les **OMR** sont basés sur des scanners capables de reconnaître le choix des électeurs sur des bulletins de vote spéciaux lisibles par machine. Ils peuvent être des systèmes de dépouillement centralisés ou des systèmes de numérisation optique de comptage dans lesquels la numérisation et le comptage des votes ont lieu directement dans le bureau de vote (Cette deuxième variante donne un semblant de transparence aux élections).

**Direct Recording Electronic voting machines (DRE)**

Les systèmes DRE permettent aux électeurs de marquer leurs votes directement dans un appareil électronique, à l'aide d'un écran tactile, de boutons poussoirs ou d'un appareil similaire.

**Electronic Ballot Printers (EBPs)**

Ils sont similaires aux machines DRE + OMR, ils produisent un papier ou un jeton électronique lisible par machine contenant le choix de l'électeur. Ce jeton est introduit dans un scanner de vote séparé qui effectue le comptage. L'utilité principale de ces systèmes est d'éviter les bulletins invalides.

**Internet Voting Systems**

Les votes sont transférés via Internet aux centres de comptage. Les votes peuvent être exprimés soit à partir d'un bureau de vote soit à partir de tout autre appareil qui a accès à Internet. Le vote à distance comporte d'autres risques techniques, des logiciels ou d'autres instruments tels que des cartes à puce sont nécessaires pour l'authentification des votants.

## 1.6 Avantages et Inconvénients

**Avantages**

Le vote électronique présente plusieurs avantages, nous citons quelques uns :

— La **Facilité** dans la mise en place des plate-formes de vote, le processus de vote pour les électeurs et dans le comptage des voix pour les administrations.

— Le **Gain en temps** dans la collecte des votes des électeurs et dans le calcul des résultats.

— La **Précision** en réduisant le taux de votes invalides, les erreurs de comptage, il est donc plus précis.

— Le vote électronique à tendance à assumer une grande partie de la responsabilité du processus électoral et confier cette responsabilité à une administration centrale des élections et les responsables de la mise en œuvre du système. En faisant ainsi, le risque de fraude généralisée et des manipulations depuis le bureau de vote est très réduit. Le e-voting est donc, en fin de compte, plus **fiable**.

— Le vote depuis Internet permet de tenir compte des voix des minorités isolées ou incapables de se déplacer, il rend le vote plus **accessible et augmente le taux de participation**.

— Les systèmes de vote électronique ne nécessitent pas forcement des dépenses exorbitantes à long terme.

— Pour une élection présidentielle par exemple, nous utilisons des millions de feuilles de papier comme bulletins de vote, des centaines de milliers de litres d'essence pour nous rendre aux bureaux de vote et beaucoup d'autres choses. Tout ceci n'est pas sans conséquence sur l'environnement. Le vote électronique est de surcroît un **bon geste pour la nature**.

**Inconvénients**

La plus grande partie des points que nous citons ici sont plus des "lacunes" que des inconvénients.

— Le fonctionnement exacte du système est incompris ou inaccessible au grand public, le vote électronique manque de transparence.

— Du moment que chaque pays développe son propre système de vote électronique ou a recourt à des sociétés privées, il est difficile de définir une norme d'implémentation, la sécurité et l'efficacité du système se retrouve diminuées.

— Les systèmes de vote électronique qui servent également de dispositif d'authentification cachent très difficilement l'identité des électeurs, il y a donc un **risque de violation du secret de vote**.

— Malgré le fait que le risque de manipulation est grandement réduit, il n'est tout de même pas inexistant. Les résultats peuvent toujours être manipulés par les administrateurs des systèmes de vote et les conséquences de cette **manipulation** sont encore plus grandes.

— La mise en place d'un tel dispositif **peut coûter** en fonction du type de technologie et de la zone dans laquelle elle est déployée.

— Il n y a **aucun moyen de vérifier les résultats** s'ils sont directement manipulés dans la base de données.

— Les électeurs ainsi que les candidats **doutent** toujours de l'efficacité du vote électronique. Ils ne voient en celui ci qu'un autre moyen du gouvernement pour manipuler les élections. Pourtant des dispositifs ne peuvent être introduits que si les citoyens décident de faire confiance à leurs systèmes politiques et administratifs.

— Il est fondamentalement impossible d'écrire un programme sans erreur qui ne plantera jamais. Développer des systèmes de vote électronique est une tâche très complexe, faire en sorte qu'elle soit sans failles est un besoin impossible à satisfaire.

# Conclusion

Dans ce chapitre nous avons parlé de démocratie, d'élections et de parrainage de candidats. Nous avons fait une étude sur les systèmes de vote électroniques, nous avons recueillis les points positifs et négatifs et les critères que doivent vérifier ces systèmes. Nous avons analysé plusieurs exemples d'implémentation du vote électronique, leurs points forts et leurs lacunes.

La démocratie dépend des élections, mais les élections sont des processus très complexes et en même temps très fragiles. Les systèmes de e-voting doivent être évalués en termes de sécurité, d'efficacité, de faisabilité, d'accessibilité et de fiabilité. Par conséquent ils doivent reposer sur des principes fondamentaux sûrs.

CHAPITRE 2

# BLOCKCHAIN

## Introduction

La blockchain est une nouvelle technologie de base de données s'appuyant et tirant pleinement profit d'Internet, du protocole libre, de la puissance de calcul et de la cryptographie. Cette base de données transactionnelle distribuée est comparable à un grand livre comptable (registre ou *ledger*) dans lequel chaque nouvelle transaction est écrite à la suite des autres, sans avoir la possibilité de modifier ou d'effacer les précédentes. Ce registre est actif, chronologique, distribué, vérifiable et protégé contre la falsification par un système de confiance repartie (consensus) entre les membres ou participants (nœuds) [10].

Nous allons voir comment Blockchain est née. Nous verrons quelques notions de cryptographie et de réseau puis nous parlerons du **Bitcoin** et des principes fondamentaux de la blockchain.

## 2.1 Généralités

Dans cette section nous avons regroupés quelques notions indispensables pour comprendre le fonctionnement de la blockchain.

### 2.1.1 Systèmes distribués

Les systèmes distribués sont un paradigme informatique par lequel deux nœuds ou plus fonctionnent les uns avec les autres pour exécuter une fonction commune de tel manière qu'ils ne fassent qu'un seul du point de vue d'un utilisateur final [3].

### 2.1.2 Systèmes décentralisés

Il y a trois aspects quand nous parlons de décentralisation [15] :

1. **Selon l'architecture physique** : Combien de composantes physiques ? Combien d'ordinateurs minimum le système entier peut contenir avant que la panne de l'un implique la panne du système entier ? Si la réponse est supérieure à deux (2), le système a une architecture décentralisé.
2. **Selon la politique organisationnelle** : combien d'entités s'occupent de la gestion du système ? Une fois de plus, si la réponse est supérieure à deux (2), le système est décentralisé par sa politique de gestion.
3. **Selon l'architecture logique** : Combien de composants logiques peuvent constituer le système ? S'il est possible de couper le système de sorte que chaque côté soit constitué de fournisseurs de services et de consommateurs, et que les deux côtés arrivent à fonctionner comme des unités indépendantes, le système est décentralisé logiquement ou centralisé sinon. De manière générale, ces composants communiquent directement entre elles sans une autre entité centrale, ce type de communication est dit « **Pair-a-pair** », « **Peer-to-peer** » ou tout simplement **P2P**.

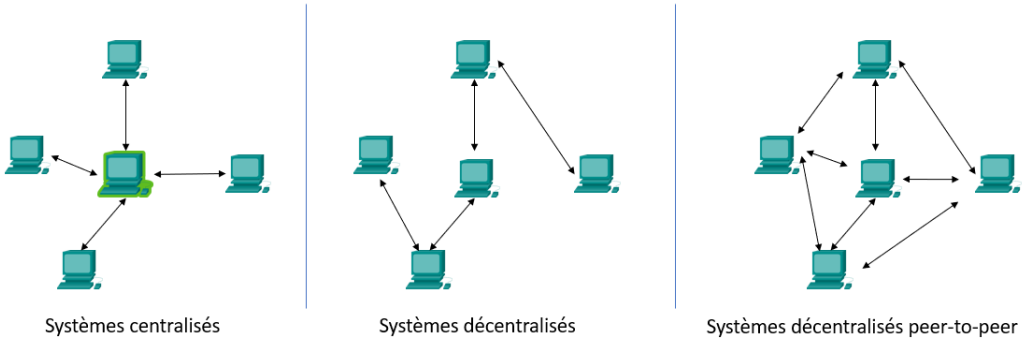


FIGURE 2.1 – Systèmes centralisés / Décentralisés / Peer-to-peer

### 2.1.3 Cryptographie

Impossible de faire ce rapport sans parler de cryptographie, ici nous parlons de quelques principes cryptographiques fondamentaux utilisés (non les seuls) par la Blockchain.

La cryptographie est le processus de conversion de texte brute (message clair) en texte inintelligible et inversement. Elle permet le stockage et la transmission de données sous une forme particulière afin que seuls ceux à qui elles sont destinées puissent les lire et les traiter.

#### Cryptographie asymétrique

Aussi appelée cryptographie à clé publique, dans la cryptographie asymétrique nous avons deux clés : une clé pour chiffrer et une autre pour déchiffrer. La clé de chiffrement est appelée « clé publique », elle est accessible publiquement par tout le monde. La clé de chiffrement est celle secrète et donc la clé privée.

#### Fonction de Hachage

Une fonction de hachage **H** prend en entrée un bloc de données **M** de longueur variable et produit une chaîne de caractères de taille fixe **X** de manière à ce qu'il soit "impossible" de remonter à **M** à partir de **X**. Le hash d'un fichier numérique est comme son empreinte, différents fichiers produisent des hachages différents et ceux identiques produisent des hachages identiques. En théorie, plusieurs fichiers peuvent avoir le même hachage mais c'est "impossible" de le prouver en pratique.

#### Signature numérique

Chaque utilisateur possède une paire de clé (une privée et une publique). La clé privée est confidentielle et est utilisée pour signer les transactions. Les transactions signées numériquement sont diffusées sur tout le réseau.

Le processus de signature numérique comporte deux phases : la phase de **signature** et la phase de **vérification**.
Par exemple, un utilisateur X souhaite envoyer un message à un autre utilisateur Y :

1. Dans la phase de signature, X crypte ses données avec sa clé privée et envoie à Y le résultat chiffré et les données originales.
2. Dans la phase de vérification, Y valide la valeur avec la clé publique d'X. Y peut facilement vérifier si les données ont été falsifiées ou non. L'algorithme de signature numérique généralement utilisé dans les Blockchains est l'algorithme de signature digitale à courbe elliptique **(ECDSA )**.

**Arbre de Merkle**

Un arbre Merkle est un arbre binaire dans lequel les entrées sont d'abord placées au niveau des feuilles (nœud sans enfants), puis les valeurs des paires de nœuds enfants sont hachées ensemble pour produire une valeur pour le nœud parent (nœud interne) jusqu'à ce qu'on obtienne une seule valeur de hachage connue sous le nom de racine Merkle.

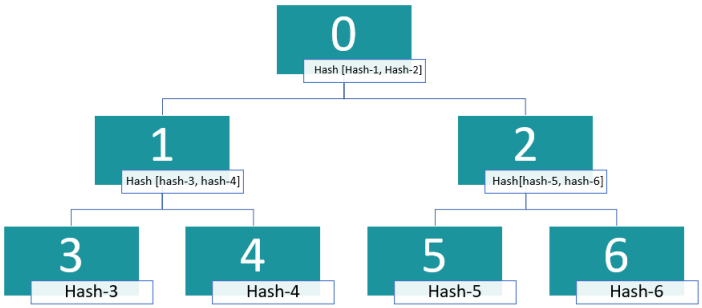


FIGURE 2.2 – Arbre de Merkle

## 2.2 Naissance de la Blockchain : Le Bitcoin

Bitcoin est apparu en 2008 dans la publication « *Bitcoin : A peer-t-peer Electronic Cash System* » d'un individu ou d'un groupe d'individus connu sous le pseudonyme « SATOSHI NAKAMOTO » [1]. Satoshi a utilisé un ensemble de principes cryptographiques pré-existants pour créer un système de monnaie électronique se passant d'autorité centrale pour établir la confiance entre les tiers.

1. « Satoshi Nakamoto » est juste un pseudonyme, l'identité(s) réelle(s) de la ou des personne(s) est ou sont toujours officiellement inconnue(s)

En terme plus exactes : ”Le protocole Bitcoin est en algorithme mathématique posé sur un réseau permettant la gestion de données de transaction et le maintien d’un consensus entre les nœuds du réseau ”. [11].

### 2.2.1 Fonctionnement du Bitcoin

Pour comprendre Bitcoin, nous allons le comparer à un système traditionnel de compte bancaire en ligne.
Dans ce cas de figure, les comptes des clients sont gérés par une autorité centrale généralement une ressource matérielle (un serveur par exemple) ou une ressource humaine (un administrateur). Tout est validé par cette autorité, les retraits, envois et paiements. Celle-ci constitue le point le plus critique du système, il suffit d’une petite erreur (volontaire ou non) de celle-ci pour que tout le réseau soit compromis.

Maintenant L’idée était la suivante : « **Supprimer l’autorité centrale des interactions !** ». Après, plusieurs questions se posent, des problèmes auxquels Bitcoin tente de répondre.

**S’il n y a plus d’autorité centrale, comment vérifier les identités des parties ?**

Il s’agissait de trouver un moyen d’attribuer un compte, un identifiant et un mot de passe aux clients. Bitcoin a trouvé la solution dans la « cryptographie asymétrique » Les identifiants des comptes sont obtenus à partir des clés publiques. Il n’y a plus besoin d’authentification, puisqu’une transaction est signée avec la clef privée de son initiateur.

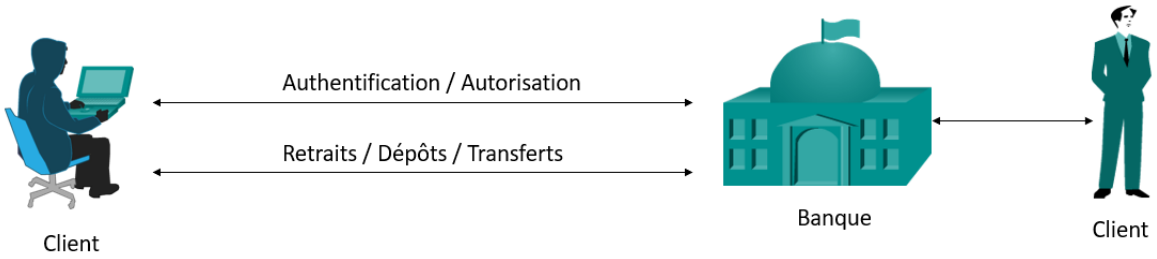


FIGURE 2.3 – Autorité centrale

**Comment garder une trace des transactions effectuées ?**

Bitcoin a proposé d’enregistrer une trace sur tous les nœuds du réseau. Si les mêmes données sont présentes sur chaque nœud, le critère de non-répudiation est assuré.

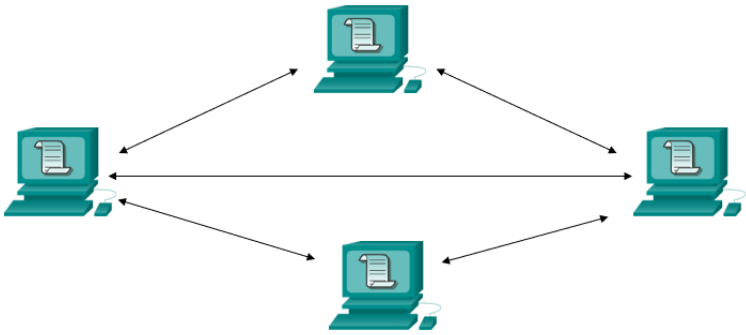

FIGURE 2.4 – Base de données décentralisée

Il est important de noter qu'ici, plus il y a de nœuds, plus le réseau est sûr.

Les nœuds sont des ordinateurs qui lisent et écrivent dans la blockchain. Il existe plusieurs types de nœuds pour différents rôles et pour différentes capacités.

- **Mining Node / Nœuds mineurs** : Les nœuds de minage sont des nœuds qui produisent des blocs. Ils disposent généralement d'énormes ressources informatiques. Les mineurs ne valident pas les blocs, car ils ne stockent pas la blockchain. Pour créer de nouveaux blocs, il leur suffit d'un ensemble de transactions récentes et d'un hachage du dernier bloc.
- **Full Node / Nœuds complets** : Les nœuds complets sont les nœuds qui stockent la copie entière de la blockchain. Ils valident les blocs qu'ils reçoivent et propagent les blocs validés sur le réseau. Ils peuvent également envoyer leurs propres transactions au réseau.
- **Light Node / Nœuds legers** : Les nœuds légers ne stockent pas toute la blockchain. Ils stockent généralement des entêtes de bloc et un très petit ensemble de transactions. Les nœuds légers ne peuvent pas valider les blocs et doivent faire confiance aux nœuds complets. Cependant, les nœuds légers peuvent se connecter à d'autres nœuds, recevoir de nouveaux blocs et envoyer les transactions.
- **Web Nodes / Nœuds Web** : Les nœuds Web ne sont pas vraiment des nœuds, ce sont des portefeuilles qui se connectent aux serveurs centralisés. Ils ne stockent aucune partie de la blockchain. Les portefeuilles Web ne peuvent pas se connecter à d'autres nœuds, recevoir des blocs ou envoyer les transactions. Ils ne peuvent que demander à un serveur

de le faire et s'attendre à ce que le serveur fournisse des informations correctes. Ils font entièrement confiance au serveur auquel ils sont connectés.

**Comment maintenir la cohérence des données sur chaque nœud ?**

Au lieu d'enregistrer directement une transaction à la fois, l'idée était d'en enregistrer plusieurs d'un coup. Les transactions sont d'abord inscrites sur une liste d'attente puis elles sont sélectionnées par les nœuds pour être validées. D'où les **blocs de transactions**.

A ce stade, nous dirons qu'un bloc contient donc : une liste de transactions, un timestamp et un numéro. Pour créer un bloc, un nœud doit présenter une « preuve de travail » en anglais ***Proof of Work (PoW)***, ceci fait partie de l'algorithme de consensus.

« Une transaction est juste une structure de donnée enregistrée sur un bloc. En termes plus simples, une transaction consiste à montrer au réseau qu'un utilisateur propriétaire d'une certaine quantité de Bitcoin a autorisé le transfert de cette somme à un autre utilisateur. Celui-ci pourra ainsi transférer une autre quantité à un autre et ainsi de suite [3]. »

Le mot « consensus » est défini dans le dictionnaire *Larousse* comme étant un *accord* et *consentement* du plus grand nombre, de l'opinion publique. Ou encore comme étant une procédure qui consiste à dégager un accord sans procéder à un **vote formel**, ce qui évite de faire apparaître les objections et les abstentions.

Comme la majorité des systèmes informatiques distribués, les nœuds du réseau doivent se mettre d'accord sur l'état de la blockchain. Comment parvenir à un consensus entre des nœuds non forcement dignes de confiance ? Ce problème est connu sous le nom « *Byzantine Generals Problem* » ou « problème des généraux byzantins » en français.

En quelques mots, le problème des généraux byzantins est un dilemme logique qui décrit comment un groupe de généraux byzantins peut avoir des problèmes de communication lorsqu'il essaie de s'accorder sur son prochain coup stratégique.

Le dilemme suppose que chaque général a sa propre armée et que chaque groupe armé est situé à différents endroits autour d'une ville qu'ils désirent assiéger. Peu importe qu'ils attaquent ou qu'ils battent en retraite, il faut que les généraux parviennent à un accord commun. Chaque général doit impérativement prendre une décision, une fois la décision prise, elle ne peut être changée. D'autres variables viennent s'ajouter, un général ne peut communiquer avec un autre que par l'intermédiaire de messages qui sont transmis par un émissaire donc les messages peuvent être retardes, détruits, perdus, modifiées ou même ignorés.

Un système capable de continuer à fonctionner même quand plusieurs noeuds agissent de manière non conforme au consensus est dit : **Byzantine Fault Tolerant** ou **BFT**.

« Un PoW est en fait une donnée difficile à produire en termes de calcul et temps, mais facile à vérifier. Chaque pair est incité à utiliser sa « puissance de calcul » en résolvant des énigmes pour valider les blocs qu'il veut créer. Bitcoin utilise un PoW basé sur le hachage qui implique de trouver une valeur (le nonce), de sorte que lorsqu'il est haché avec des paramètres de bloc supplémentaires (par exemple, le racine de Merkle, le hash du bloc précédent), la valeur du hachage doit être inférieure à la valeur cible actuelle (Cette valeur constitue la difficulté de l'algorithme) . Lorsqu'un tel nonce est trouvé, le mineur crée le bloc et le transmet aux autres nœuds du réseau . » [6]

La PoW dans Bitcoin tire parti de la nature apparemment aléatoire des hachages cryptographiques. Un bon algorithme de hachage cryptographique convertit des données arbitraires en un nombre apparemment aléatoire. Si les données sont modifiées de quelque manière que ce soit et que le hachage est ré-exécuté, un nouveau nombre "aléatoire" est produit, il n'y a donc aucun moyen de modifier les données pour rendre le nombre de hachage prévisible.

Les blocs sont créés à environ dix minutes d'intervalle dans Bitcoin. Le processus de création de blocs est appelé : **Minage**.

Chaque transaction validée correspond à une petite commission et même quand il n'y a pas de transaction en attente, les mineurs peuvent continuer à s'envoyer des commissions (une petite quantité de bitcoin par blocs) , d'où l'intérêt de continuer à miner. Le bénéfice d'un mineur varie avec sa puissance de calcul. C'est là l'un des principaux inconvénients de la preuve de travail. La preuve de travail ne permet pas une validation équitable des blocs pour les mineurs. Les fermes de minage disposant d'une plus grande puissance de calcul sont privilégiées. Ceci rend la tâche plus compliquée et moins rémunératrice pour les petits mineurs (qui ne disposent pas de grandes ressources).

Il faut aussi noter que le réseau s'auto-équilibre, c'est à dire : s'il y a plus de puissance de calcul et que les blocs sont créés trop rapidement, la difficulté de trouver la preuve de travail augmente. Dans le cas contraire elle diminue.

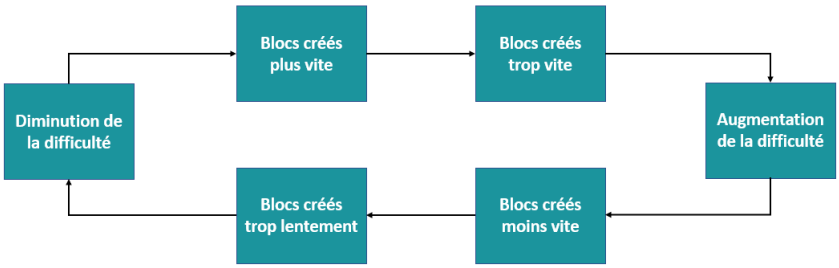


FIGURE 2.5 – Gestion de la difficulté PoW [11]

.

**Comment empêcher un mineur de tricher ?**

Il est dit qu'un mineur peut valider un bloc même s'il ne contient aucune transaction. Dans ce cas, il peut faire le calcul de la preuve de travail de plusieurs blocs à venir et attendre que les numéros correspondants arrive puis les valider, ce mineur est donc sûr de gagner. Heureusement, SATOSHI a pensé à tout. Il trouve la solution à ce problème dans une autre notion de cryptographie : Les arbres de Merkel. On construit un arbres de Merkle avec les transactions puis la racine de cet arbre est enregistrée dans l'entête du bloc. Après le hash de ce bloc est ajouté au bloc suivant. Cette opération permet de lier les blocs entre eux.

Le chaînage des blocs rend impossible la modification des transactions incluses dans un bloc sans modifier tous les blocs suivants. En conséquence, le coût de modification d'un bloc particulier augmente avec chaque nouveau bloc ajouté à la chaîne de blocs, amplifiant l'effet de la preuve de travail.

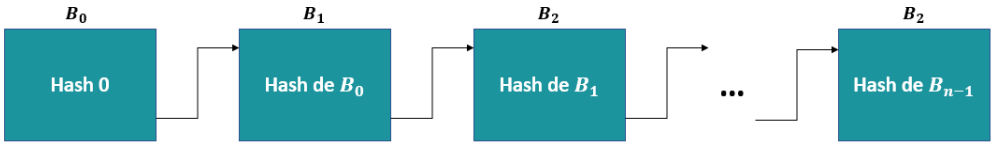


FIGURE 2.6 – Chaînage de blocs

La preuve de travail ne suffit pas pour empêcher les conflits de blocs, deux mineurs peuvent théoriquement valider deux blocs ayant le même numéro au même moment, dans ce genre de conflit, le réseau choisit toujours celui qui a la plus longue chaine. C'est l'ensemble de toutes ces règles qui définissent le **consensus**.

La figure suivante résume le fonctionnement de Bitcoin.

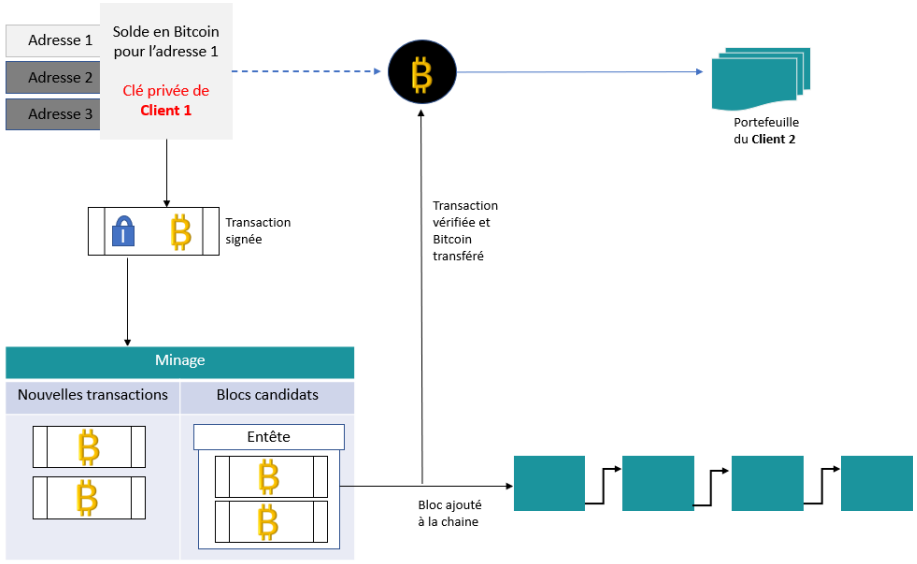


FIGURE 2.7 – Bitcoin Process [18]

.

## 2.3 Principe de la Blockchain

La Blockchain n'est pas une generalisation du Bitcoin mais plutot la technologie sous-jacente de celui ci. Nous pouvons la definir comme etant : une technologie pour une nouvelle génération d'applications transactionnelles qui, grâce à un mécanisme de consensus collectif couplé avec l'utilisation d'un grand livre de compte public, décentralisé et partagé, établit la confiance, la responsabilité et la transparence [10]. Elle regroupe quatres notions essentielles qui sont :

**Base de données décentralisée et distribuée** : Comprendre les systèmes distribués est essentiel pour comprendre la Blockchain, car elle est un système distribué à sa base. C'est un registre distribué qui peut être centralisé ou décentralisé.

**Cryptographie** : On dit que la base de donnée est cryptographiquement sûre, c'est à dire qu'elle repose sur des principes cryptographiques réputées incassables jusqu'à maintenant.

**Système de confiance ou consensus** : Ce sont des nœuds qui font la validation des transactions, cette validation se fait suivant certaines règles, ces règles font partie du **consensus**.

**Règles d'interaction ou contrats intelligents** : Un contrat intelligent est un programme informatique sécurisé et imparable représentant un accord qui est automatiquement exécutee quand

toutes les conditions sont reunies [3]. Il est constitué des accords entre les parties. Le contrat est stricte, ce qui signifie que toutes les conditions contractuelles sont exécutées comme définies et attendues, même en présence de conflit.

## 2.4 Types de Blockchains

Il y a plusieurs manières de classer les Blockchains :

### 2.4.1 Classification selon l'accès aux données

En fonction de la nature de l'accessibilité des données, les blockchains peuvent des catégories comme ci-dessous :

- **Blockchain publique** : N'importe qui peut lire et soumettre des transactions.
- **Blockchain privée** : Dans ce type de blockchain, une seule organisation ou toutes les sous-organisations du même groupe sont autorisées à lire et à soumettre des transactions. Ce sont généralement des blockchains d'entreprises.
- **Blockchain de Communauté / Consortium** : Plusieurs groupes d'organisations forment une communauté ou un consortium et sont autorisés à soumettre des transactions et à lire des données transactionnelles.
- **Blockchain hybride** : Une combinaison de plusieurs types de blockchain (privée et publique). Une plate-forme Blockchain peut être configurée en multi-mode à l'aide de la Blockchain hybride.

### 2.4.2 Classification selon l'autorisation de participation

- **Permissionless / Sans autorisation** : Aucune autorisation préalable n'est nécessaire pour participer à ce type de blockchain. Tout le monde est autorisé à participer au processus de validation et peut rejoindre le réseau avec sa propre puissance de calcul.
- **Permissioned / Avec autorisation** : Une autorisation préalable est nécessaire pour rejoindre ce type de blockchain. Seules les parties autorisées peuvent valider les transactions sur le réseau.

— **Hybrid / Hybride ou Mixte** : Une plate-forme Blockchain peut également être configurée pour prendre en charge le modèle avec autorisation (pour une communication inter-blockchain par exemple) et sans autorisation (pour le reste des communications).

### 2.4.3 Classification selon les fonctionnalités

En ce qui concerne les fonctionnalités de base et la prise en charge des contrats intelligents.

- **Stateless** : Le système Blockchain sans état se concentre uniquement sur l'optimisation des transactions et la vérification de la transaction. Il est indépendant de la couche logique de contrat intelligent et n'est donc pas affecté par les bogues et les vulnérabilités du code de contrat intelligent.
- **Stateful** : Ce type de Blockchain fournit des capacités de traitement des contrats et des transactions.

## 2.5 Exemples de blockchains

### 2.5.1 Ethereum

Le bloc de genèse d'Ethereum [2] a été lancé en juillet 2015 et la plate-forme est actuellement l'une des plates-formes les plus utilisées pour les contrats intelligents‘’. Du point de vue informatique, Ethereum est une machine à états déterministe, composée d'un état accessible à l'échelle mondiale et d'une machine virtuelle qui applique des changements à cet état.

D'un point de vue plus pratique, Ethereum est une infrastructure informatique open source et décentralisée à l'échelle mondiale qui exécute des contrat intelligents. Il utilise une blockchain pour synchroniser et stocker les changements d'état du système, ainsi qu'une crypto-monnaie appelée « Ether » pour mesurer et limiter les coûts des ressources d'exécution.

La plate-forme Ethereum permet aux développeurs de créer de puissantes applications décentralisées avec des fonctions économiques intégrées. Tout en offrant une haute disponibilité, un contrôle, une transparence et une neutralité sans faille. Il comprend les éléments suivants, essentiels à la compréhension du réseau et de ses nombreuses applications.

2. Le premier bloc d'une blockchain est appelé **bloc de genèse**

**Ether (ETH)** : C'est la monnaie numérique native d'Ethereum. Il sert a stocker la valeur en ETH, régler les transactions en permettant aux utilisateurs d'envoyer ou de recevoir des paiements en ETH et faciliter les opérations du réseau via les frais de transaction payés en ETH.

**Smart contract** : Ce sont des lignes de code qui facilitent l'échange de tout ce qui est représentatif de la valeur, comme de l'argent, des informations, des biens ou des droits de vote. À l'aide de contrats intelligents, les utilisateurs peuvent envoyer ou recevoir des ETH, créer des marchés, stocker des registres de dettes ou de promesses, représenter la propriété d'un bien ou d'une entreprise, transférer des fonds à partir d'un ensemble d'instructions logiques et former de nouveaux actifs numériques dans des offres ou des émissions conformes.

**Ethash** : Un algorithme de recherche de preuve de travail implémenté par Ethereum et les crypto-monnaies basées sur Ethereum.

**Machine virtuelle Ethereum (EVM)** : Une machine virtuelle (VM) est un logiciel qui émule le comportement d'un ordinateur, agissant essentiellement comme un environnement d'exécution pour toute activité que l'on peut effectuer sur un ordinateur ordinaire.

**Solidity** : C'est le principal langage de programmation d'Ethereum. Il est utilisé pour écrire des contrats intelligents, développer des **DApps**, structurer des **DAO** et exploiter des appareils **IoT** (Internet des objets) basés sur la technologie Ethereum.

**DApps (Applications décentralisées)** : Ce sont des applications, des programmes ou des outils qui utilisent des contrats intelligents intégrés au réseau Ethereum.

**Organisations autonomes décentralisées (DAO)** : Les organisations autonomes décentralisées sont des organisations qui fonctionnent indépendamment d'un organe central. Contrairement aux entreprises traditionnelles où la propriété est divisée entre les actionnaires, un DAO appartient à ceux qui contribuent aux jetons, qui ont également des droits de vote. De plus, les règles d'un DAO sont déterminées par sa collection de contrats intelligents.

### 2.5.2 Hyperledger

Hyperledger a débuté en 2015 lorsque de nombreuses entreprises intéressées par la technologie blockchain ont décidé de mettre en commun leurs ressources et de créer une technologie open source que tout le monde pourrait utiliser. Hyperledger a été placé sous la tutelle de la *Linux Foundation* et a connu une croissance très rapide [7].

Hyperledger fournit plusieurs Framework pour différents cas d'usage [7].

**Hyperledger Burrow** : Un client blockchain modulaire avec un interprète de contrat intelligent autorisé développé en partie selon les spécifications de la machine virtuelle Ethereum (EVM).

**Hyperledger Fabric** : Une plate-forme pour la création de solutions de registres distribués avec une architecture modulaire qui offre un haut degré de confidentialité, de flexibilité, de résilience et d'évolutivité. Cela permet aux solutions développées avec Fabric d'être adaptées à toutes les industries.

**Hyperledger Indy** : Un registre distribué qui fournit des outils, des bibliothèques et des composants réutilisables spécialement conçus pour une identité décentralisée.

**Hyperledger Iroha** : Un framework blockchain conçu pour être simple et facile à intégrer dans les projets d'infrastructure d'entreprise.

**Hyperledger Sawtooth** : Une plate-forme modulaire pour créer, déployer et exécuter des registres distribués. Sawtooth propose un nouveau type de consensus, preuve du temps écoulé (**PoET**) qui consomme beaucoup moins de ressources que la preuve de travail **(PoW)** [7].

# Conclusion

Dans ce chapitre, nous avons fait une petite étude de la blockchain, nous avons vu les principales notions qui la composent : ses principes cryptographiques, les systèmes distribuées, la décentralisation, le mécanisme de consensus etc. Nous avons parlé des types de blockchain en nous basant sur plusieurs critères de classification. Nous avons discuté de quelques blockchains connues, notamment Bitcoin et Ethereum, et comment elles ont contribué à leurs manières dans leurs secteurs respectifs.

Les blockchains sont réputées très sécurisées, cependant ce cote sécurisé peut disparaître d'un jour à l'autre vu qu'elles reposent majoritairement sur des principes mathématiques qui comme toujours finissent par être cassés.

CHAPITRE 3

# COLLECTE DE SIGNATURES BASÉE SUR LA BLOCKCHAIN

## Introduction

Nous avions parlé des exigences que doivent vérifier les systèmes de vote électronique. Ces exigences sont très difficilement satisfaites compte tenu du caractère contradictoire de certaines d'entre elles, comme assurer l'auditabilité tout en garantissant le secret de vote. Un système de vote électronique doit également faire face à des problèmes non techniques tels que le fait de s'assurer qu'un électeur ne vote pas sous pression (seul l'isoloir peut prétendre le garantir).

Promouvoir le vote électronique lors d'une élection est difficile à mettre en œuvre si nous voulons garantir toutes les conditions d'un vote démocratique. Nous préférons prendre un cas de figure qui, finalement, est très proche du vote électronique mais dont les critères sont un peu plus souples. Comme évoqué précédemment nous nous consacrerons à la sélection des candidats à une élection présidentielle ou au déclenchement d'un référendum d'initiative populaire pour but d'apporter plus de simplicité et de crédibilité à l'approche classique. Par la suite, nous utiliserons le terme « collecte de signature » pour désigner les deux cas de figures.

Il est évident que toute solution à un problème donné se doit de résoudre ce problème ; la nôtre étant une approche parmi tant d'autres, pourquoi la Blockchain ?
Voici quelques points qui nous ont poussé à faire ce choix :

- Les données étant difficilement modifiables sur une Blockchain, nous pouvons assurer l'in-

tégrité des signatures.

- Tout se qui passe sur un réseau Blockchain est répertorié sous forme de transactions, nous pouvons garantir une totale transparence des interactions pour toutes les parties concernées par les pétitions.
- Du fait du caractère décentralisé du réseau de Blockchain et des smart-contrats, nous pouvons faire en sorte que la gestion d'une pétition ne dépende ni du pétitionnaire ni d'une quelconque institution gouvernementale.

Dans ce chapitre nous tenterons d'apporter toutes les clarifications possibles à notre modèle (de son architecture physique à son architecture logique) ainsi que le processus de collecte (de son enclenchement à sa fermeture).

## 3.1 Rôles et Architecture

Un citoyen voulant initier une collecte de signature qu'il s'agisse d'une pétition ou d'un parrainage pré-électoral (Sponsorship) doit discuter des modalités avec une institution gouvernementale ou une organisation concernée par l'initiative. Le pétitionnaire présente une description en bonne et due forme à d'autres personnes qui décideront de signer ou non la dite pétition. En supposant qu'ils décident de la signer, ils doivent fournir des informations d'identification et une preuve d'éligibilité (typiquement une carte d'électeur). Dans la majorité des cas, la signature en elle-même est un document papier sur lequel se trouvent les identifiants du signataire et sa signature manuscrite. Les signatures ainsi collectées sont présentées à une autorité qui se charge de la vérification et du comptage.

En nous basant sur le scenario précédent, nous avons relevé les rôles suivants :

- **Public**

  Les utilisateurs simples, peuvent suivre l'évolution d'une ou de plusieurs autres pétitions en cours (éventuellement la description, les statistiques, etc). Ces observateurs n'ont pas besoin d'authentification, du moment que la collecte concernée est ouverte au grand public et que les interactions avec ce public sont passives (c'est à dire qu'il n y a pas de modification de l'état des données).

- **Signataire**

Quand un utilisateur jugé éligible, identifié par une clé publique, disposant d'un client Web sécurisé pour accéder à la Blockchain, et dont chaque transaction est signée par sa clé privée, décide de signer une pétition, il devient SIGNATAIRE de cette pétition.

- **Pétitionnaire**

  Quand un utilisateur commence une nouvelle collecte de signature, on dit qu'il a le rôle d'initiateur du point de vue de cette collecte. Pour une pétition ou pour un parrainage pré-électoral, l'initiateur est respectivement un signataire simple ou une autorité désignée.

- **Candidat**

  Les signatures, lors d'une collecte de parrainage pré-électoral sont faites pour le compte des candidats. Bien que ces entités n'aient pas d'interactions actives avec la collecte, ils peuvent notamment accéder à plus de contenu que le public ordinaire.

- **Autorités décentralisées** $DA$

  Le critère d'éligibilité suggère la présence d'une autorité de vérification d'éligibilité qui fournirait à chaque utilisateur, jugé éligible, une preuve d'éligibilité. De plus, les collectes de type Sponsorship seront enclenchées par des autorités désignées.

  Pour mettre le point sur l'aspect décentralisé du système, il peut y avoir plusieurs autorités ayant le même niveau d'habilitation. Nous pourrions imaginer alors un ensemble coordonné d'autorités qui exécutent des fonctions selon une politique prédéfinie (suivant l'avis de la majorité, suivant un ordre ou peut être même de manière aléatoire).

- **Mineurs**

  Les mineurs aident à maintenir l'aspect décentralisé du réseau et, comme leur nom l'indique : pour **le minage de blocs**, ils sont les seuls à pouvoir miner. Ils peuvent être des organisations indépendantes, des institutions gouvernementales ou des personnes soucieuses du déroulement des pétitions.

  Puisqu'il n'est pas question d'utilisation de valeurs (crypto-monnaies) et que les mineurs n'ont pas forcément de grandes ressources de calcul, notre suggérons l'utilisation d'un algorithme simple de consensus (par exemple : ROUND ROBIN) pour choisir lequel des nœuds mineurs doit valider le prochain bloc de transactions.

Les nœuds forment ainsi entre eux un réseau pair-a-pair, toutes les données sont stockées sur la Blockchain et propagées sur tout le réseau.

## 3.2 Protocole de collecte de signatures basé sur une Blockchain

Le processus de collecte peut être divisé en plusieurs phases séquentielles. Chaque phase sera exprimée sous forme de transaction, ces transactions recevront des données en entrée puis produiront en sortie d'autres données qui seront ensuite utilisées par les phases suivantes et ainsi de suite.

Les notations suivantes seront utilisées dans la suite du document :

- $x$ ou $y$ pour l'utilisateur $x$ ou l'utilisateur $y$.
- $PuK_x$ est la clé publique de $x$, sa clé privée sera utilisée pour signer ses transactions.

### 3.2.1 Phase d'initiation de la collecte

Un citoyen initialise une collecte de signatures en spécifiant son cadre, ses objectifs ou impacts (vis-a-vis des citoyens et des institutions concernées). Cette instance est ensuite jugée (ou non) selon des règles prédéfinies (selon la constitution par exemple).

Dans le cas d'un parrainage pré-électoral, le processus est enclenché par une autorité connue des candidats. Elle fixe les modalités de la collecte (la date de début, la durée, le nombre de signatures visé, la répartition géographique des signatures etc.).

La résultante de ce processus est la preuve qu'une collecte $i$, dirigée par des règles d'une instance de contrat, a été initiée par un utilisateur $x$ ou une autorité $DA$.

$$P(X\,;\,i)\ :\ initiate(i)_{signedByX} \tag{3.1}$$

Le processus de collecte de signatures pour un parrainage pré-électoral est lancé par les Autorités Décentralisées $DA$ en utilisant la transaction suivante :

$$P(DA\,;\,i)\ :\ initiate(i)_{signedByDA} \tag{3.2}$$

Ces transactions déclencheront le contrat intelligent où les conditions et les objectifs de la consultation sont fixés et permettront aux signataires de signer. Les conditions peuvent être :

- timestamps : Date d'enregistrement de la consultation.

- durée : Durée de la consultation.
- objectifs : Nombre de signatures requis pour une proposition ou un candidat.
- zones : Zones concernées par la collecte ;

Les transactions qui permettent aux signataires d'être autorisés à signer sont signés par le DA comme suit :

$$E(x;\ p(y\,;\ i))\ :\ Eligibility(PuK_x\,;\ P(y\,;\ i))_{signedByDA} \tag{3.3}$$

Où $E(x;\ p(y;\ i))$ représente l'éligibilité d'un utilisateur $x$ à signer une pétition $i$.

$$E(x;\ p(y\,;\ i))\ :\ Eligibility(PuK_x\,;\ P(y\,;\ i))_{signedByDA} \tag{3.4}$$

Où $E(x;\ p(y;\ i))$ représente l'éligibilité d'un signataire $x$ à être candidat ou signer pour un candidat. Nous supposons que le $DA$ peut ajouter/mettre à jour la liste des citoyens éligibles à signer. Le $DA$ certifie automatiquement (le smartcontract lance cette opération) l'éligibilité des citoyens selon les critères souhaités. Ce processus se fait en donnant un certificat à l'utilisateur $x$, techniquement une signature numérique $PuK_x$ et la collecte $P\,()$ signée par $DA$.

Une autre question est de savoir quelle autorité peut fournir des preuves d'éligibilité. Cela peut être restreint à une ou à plusieurs $DA$, où une preuve peut être fournie par une autorité individuelle, par un nombre prédéfini d'autorités ou par la majorité etc.

### 3.2.2 Phase de candidature

Dans le cas d'une présélection de candidats, après l'ouverture de la procédure par $DA$ (voir transaction (2), un signataire $x$ peut devenir un candidat $c()$ en prouvant d'abord qu'il est éligible $E()$ durant tout le processus de consultation grâce à l'éligibilité accordée par $DA$. Cette opération est évidemment signé par le candidat $x$.

$$C(x;\ P\,(DA;\ i))\ :\ Candidat(E(x;\ P\,(DA;\ i)))_{signedByX} \tag{3.5}$$

### 3.2.3 Phase de signature

Dans le cas d'une pétition, un signataire x peut signer une pétition en prouvant son éligibilité à signer la dite pétition. La transaction correspondante est de la forme :

$$S(x;\ P\,(y;\,i))\ :\ sign(E(x;\ P\,(y;\,i)))_{signedByX} \tag{3.6}$$

Et maintenant si la signature est destinée à un candidat qui participe à une pré-sélection, le signataire doit prouver son éligibilité à signer $E()$ la collecte $P()$ et il doit aussi spécifier le candidat $C()$ pour lequel il signe. La transaction devient donc :

$$S(x;\ P(y;\,i))\ :\ sign(E(x;\ P\,(DA;\,i));\ C(y;\ P\,(DA;\,i)))_{signedByX} \tag{3.7}$$

### 3.2.4 Fermeture de la collecte

La collecte continue jusqu'à ce que la durée soit épuisée ou éventuellement si le nombre de signatures visé est atteint, toutes les règles de fermeture sont précisées dans le smart-contrat qui veille au respect de celles-ci.

Dans le cas de la pré-sélection des candidats, l'opération est fermé à la fin de la période des collectes. D'autre part, les candidats vérifiant les conditions en termes d'objectifs sont automatiquement extraits et ils quittent la collecte.

Les transactions faites sur le réseau sont signées, pas chiffrées, elles sont visibles par tous les parties de la Blockchain. Toutes les signatures sont dénombrables, un simple parcours des transactions permet de donner les statistiques relatives à chaque collecte.

## Conclusion

La Blockchain est une approche innovante pour stocker des informations, exécuter des transactions, exécuter des fonctions et établir la confiance dans un environnement ouvert.

Le système de collecte des signatures joue un rôle clé dans les processus de consultation. Dans ce chapitre, nous avons analysé et montré comment la Blockchain peut considérablement

améliorer le processus de collecte de signatures pour atteindre les objectifs d'efficacité et de transparence. Bien vrai que le modèle que nous proposons sert principalement pour des pétitions ou pour des présélections de candidats avant une élection, il reste tout de même très générale et peut être modifié légèrement et s'adapter à des cas bien plus spéciaux.

CHAPITRE 4

# CONCEPTION ET IMPLÉMENTATION D'UNE DAPP DE COLLECTE DE SIGNATURES

## Introduction

Les applications décentralisées ou **DApps** sont devenues une tendance dans le développement logiciel. Elles fonctionnent généralement sur une Blockchain ou tout simplement sur des réseaux de nœuds peer-to-peer. Malheureusement, la production de logiciels Blockchain manque toujours d'un cadre de développement discipliné, organisé et ou mûr.

Pour la conception de notre DApp de collecte de signatures, nous avons décidé de prendre une toute nouvelle approche : la méthode ABCDE pour **A**gile **B**lock**c**hain **D**App **E**ngineering parue dans [12]. ABCDE considère l'intégration logicielle entre les composants de la Blockchain , contrats intelligents, bibliothèques, structures de données et les composants extérieurs à la chaîne de blocs, tels que les applications Web ou mobiles, qui constituent tous, ensemble un système DApp complet. Elle préconise l'utilisation des pratiques agiles, car celles-ci sont jugées adaptées pour développer des systèmes dont les exigences ne sont pas complètement comprises depuis le début, ou tendent à changer, comme c'est le cas de la plupart des applications basées sur la Blockchain. ABCDE, basée sur SCRUM [1], est itératif et incrémentale.

ABCDE propose de séparer les activités de développement en deux flux : l'un pour les

1. SCRUM est un autre framework de developpement agile (*www.scrum.org*)

contrats intelligents et l'autre pour les logiciels hors chaîne qui interagissent avec la Blockchain. Chacun, effectué de manière itérative, avec des activités d'intégration toutes les 2 à 3 itérations. Un diagramme dérivé d'un diagramme de classe UML aide à modéliser efficacement la structure de données des contrats intelligents, tandis que l'échange de messages entre les entités du système est modélisé à l'aide d'un diagramme de séquence UML modifié.

Un autre point intéressant est que la méthode se concentre sur la Blockchain Ethereum et son langage Solidity. Cependant il reste toujours assez général et, avec des modifications appropriées, pourrait être appliqué à tout projet de développement de logiciel Blockchain.

Dans ce chapitre nous étudierons plus en détails les rouages qu'il faut connaître pour la conception de notre modèle de collecte de signatures.

## 4.1 Processus de développement et modélisation selon ABCDE

L'approche ABCDE, prend en compte la différence substantielle entre le développement de logiciels traditionnels (Desktop ou Web) et le développement de contrats intelligents, et sépare les deux activités.

Notre DApp de collecte de signatures comprend deux parties, une application Web qui gère les interactions avec les utilisateurs (en fournissant des GUIs dédiées et en effectuant des contrôles en amont) et une partie Blockchain qui gérera les contrats intelligents et les interactions avec la Blockchain Ethereum.

Bien vrai que les cas d'utilisations de notre système soient très dépendants et successifs, nous avons tout de même essayer de les décomposer pour pouvoir les étudier indépendamment et pour que les diagrammes soient les plus explicites possible. Nous considérons les processus suivants pour les deux cas de figures (Pétition et Sponsorship) :

- Attribution de preuve d'éligibilité.
- Consultation, recherche, suivi de collecte.
- L'initiation et la validation des collecte.
- La signature de collecte.

### 4.1.1 Définition des objectifs et identification des acteurs

Cette partie décrit les trois premières étapes de conception, à la fin de celles-ci nous devons avoir une vue de la manière dont les différents types d'utilisateurs interagissent avec le système pour des tâches déterminées. Nous décrirons les objectifs des utilisateurs, les interactions entre les utilisateurs et le système et le comportement requis du système pour atteindre ses objectifs.

Nous pouvons regrouper toutes les actions utilisateurs sous forme de rôles, des rôles qui deviendront ensuite des acteurs (secondaires ou primaires). D'après les besoins que nous avons pu mettre en évidence (ces acteurs ne sont pas forcement les mêmes que ceux dans le modèle **Chapitre 3**) :

1. **Citoyen** : Il s'agit du rôle le plus basique, ce sont les utilisateurs qui peuvent regarder les collectes ouvertes, lire les descriptions (impacts, nombre de signatures, institutions visées etc.). Quand un utilisateur du grand public décide de commencer une pétition et qu'il y arrive, il est considéré comme initiateur de celle-ci, cela lui permet certains privilèges (toujours par rapport à la collecte initiée). Lors de l'initiation d'une collecte de signatures, l'utilisateur ou les utilisateurs pour lesquels les signatures sont faites sont les bénéficiaires. Le terme étant un peu ambigu, nous parlerons de *pétitionnaire* ou de *candidats* en fonction des cas. Après l'initiation d'une collecte, des utilisateurs éligibles peuvent la signer. Il est évident que tous les utilisateurs simples ne peuvent pas faire toutes ses actions, ils seront contrôlés par des règles d'accès (nous en reviendrons).
2. **Autorités** : Il s'agit des autorités compétentes pouvant prouver l'éligibilité des signataires et qui peuvent initier ou valider les collectes.
3. **Administrateur** : Pour tout système informatique, il faut un administrateur pour la gestion et la correction de certaines erreurs d'utilisation. Cet administrateur n'aura pas forcement tous les droits sur le système, nous verrons comment cela est géré dans la suite.

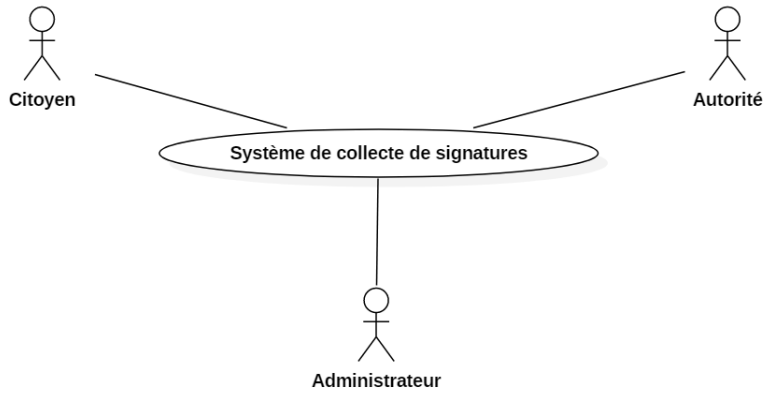


FIGURE 4.1 – Diagramme de contexte statique du système

### Diagrammes des cas d'utilisation

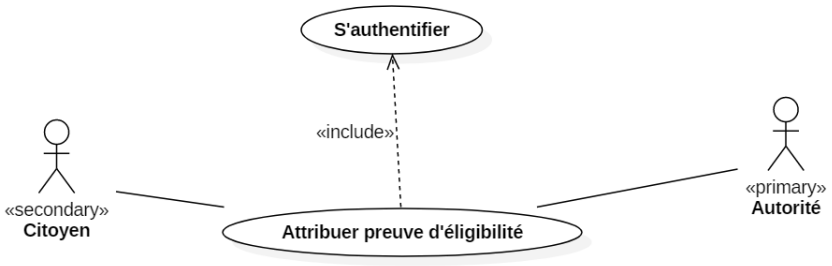


FIGURE 4.2 – Diagramme de cas d'utilisation : Attribution de preuve d'éligibilité

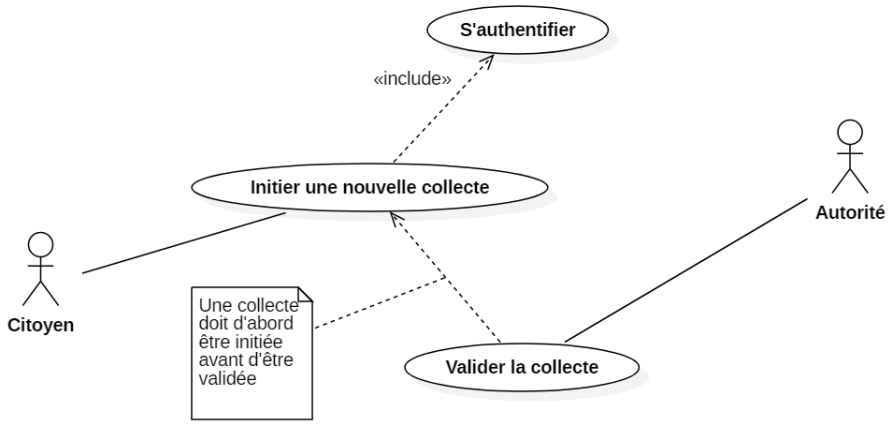


FIGURE 4.3 – Diagramme de cas d'utilisation : Initiation et validation de collecte

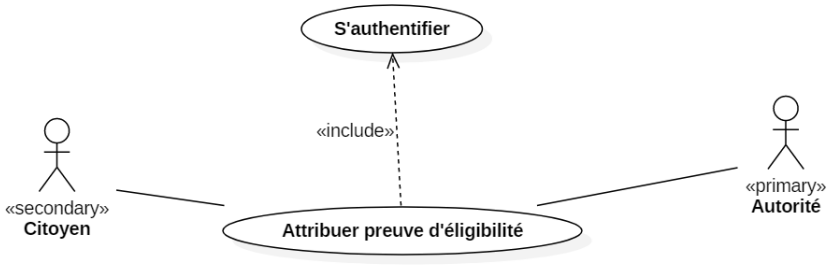


FIGURE 4.4 – Diagramme de cas d'utilisation : Signature de collecte

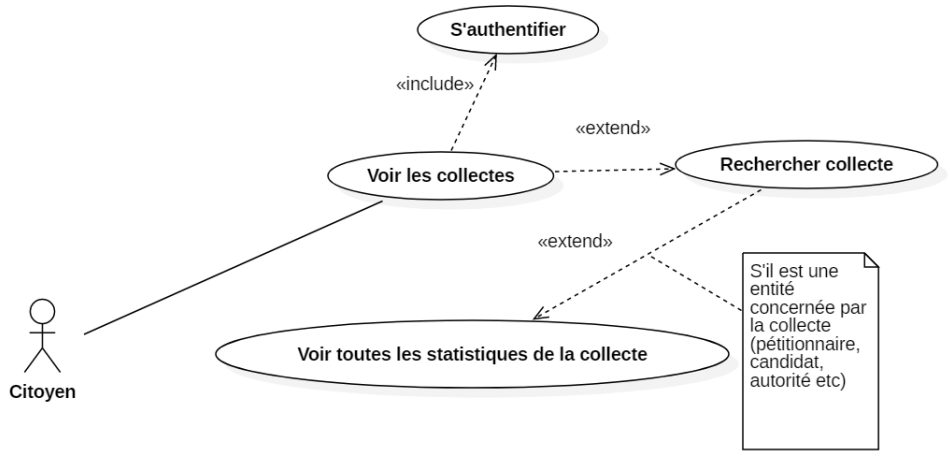


FIGURE 4.5 – Diagramme de cas d'utilisation : Consultation, Recherche et Suivi

Tout d'abord, un utilisateur quelconque accède au système, il s'authentifie. Après une authentification réussie il peut parcourir la liste des collectes ouvertes. S'il trouve une collecte qui l'intéresse et qu'il est éligible, il peut la signer (dans ce cas il devient automatiquement un signataire pour la collecte signée).

Après une authentification réussie et après avoir prouvé son éligibilité, un utilisateur ou une autorité peut commencer une nouvelle collecte en spécifiant le ou les bénéficiaire(s) (lui-même si c'est une pétition simple et des candidats si Sponsorship). Une autorité compétente peut juger ou non si la collecte initiée est valide.

### 4.1.2 Division du système

A ce stade, tous les objectifs de base sont déjà posés, nous avons identifié les acteurs et les cas d'utilisation. La conception se divise maintenant en deux parties :

- L'infrastructure Blockchain : Elle enregistre les contrats et les transactions et garantit la sécurité et la confidentialité.
- L'application Web : Pour la surveillance des événements (réponses, erreurs etc.), c'est une plate-forme web simplifiée avec une interface permettant d'afficher les collectes , d'insérer les données de nouvelles collectes etc.

La directive est que les smart-contrats gèrent les données et traitements qui doivent être transparents et non altérables pour garantir la confiance vis à vis des utilisateurs. Ces données sont : les informations d'authentification des utilisateurs, les données critiques relatives aux collectes de signatures etc.

Les autres données, traitements et interfaces utilisateur sont gérées hors de la chaîne de blocs. Comme pour les données qui se doivent d'être fiables, mais qui ne peuvent pas être stockées dans la Blockchain en raison de la transparence sur la chaîne. Ces données seront des mots de passe temporaires, des pseudos etc.

### 4.1.3 Les acteurs et les cas d'utilisation « Point de vue du Smart-contrat »

Cette étape concerne la conception des SC, l'activité est réalisée par des itérations qui doivent inclure le codage et la livraison de versions de smart-contrats, qui sont les cas d'utilisations choisis pour chaque itération. Nous nous concentrons uniquement sur les acteurs qui interagissent directement avec les contrats. Nous définirons les fonctions internes, privées et les modificateurs, les fonctions spéciales qui testent généralement les conditions préalables (les règles du contrat). Nous Définirons les tests et réaliserons les pratiques d'évaluation de sécurité.

Dans cette implémentation nous définissons essentiellement trois smart-contrats :

- Le contrat de contrôle et d'attribution de jeton d'éligibilité.
- Le contrat de traitement des collectes de type *Pétition* avec les fonctions d'initiation, de validation, de suivi et de signature de collecte.

- Le contrat de traitement des collectes de type *Sponsorship* avec les fonctions d'initiation, de validation, de suivi et de signature de collecte.

**Contrat de contrôle et d'attribution de preuve d'éligibilité**

Chaque citoyen (signataire, initiateur ou bénéficiaire d'une collecte) doit avoir une preuve d'éligibilité. Et cette preuve d'éligibilité est attribuée par une autorité. Il y a donc deux acteurs : le Citoyen et l'Autorité.

La manière de demander la preuve d'éligibilité à l'autorité ne sera pas prise en compte, nous allons supposer que l'autorité en question a pu s'assurer de la vrai identité du citoyen (en présentiel ou à travers un identifiant unique sur une carte d'électeur, un permis de conduire ou un numéro de sécurité social etc).

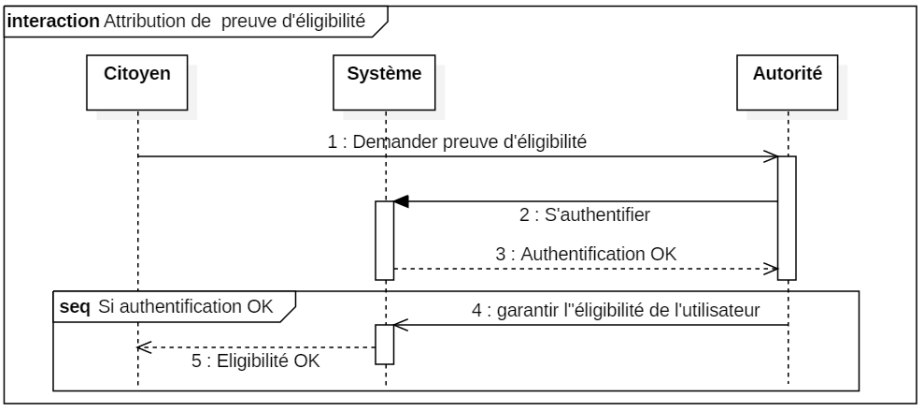


FIGURE 4.6 – Diagramme de séquence : Attribution de preuve d'éligibilité

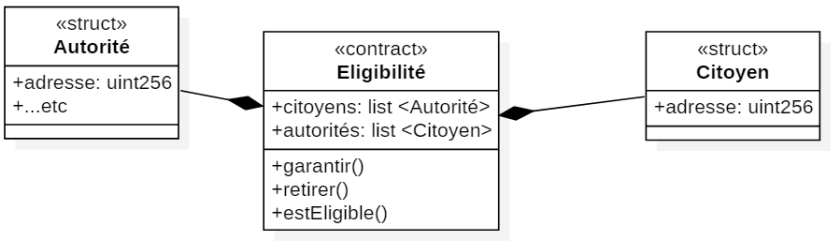


FIGURE 4.7 – Diagramme de classe Stéréotypé : Contrat Éligibilité

Le contrat pourrait donc avoir la forme suivante :

contrat **Eligibility** {

**Structure** Citoyen {

adresse : **Adresse ou Clé publique**;

eligible : **Vrai ou Faux**;

...

}

**Structure** Autorité {

adresse : **Adresse ou Clé publique**;

...

}

// Listes des citoyens, autorités

**Liste Autorité** autorités : **Liste de toutes les autorités**;

**Liste Citoyen** citoyens : **Liste de tous les citoyens**;

Fonction **garantir(** $user$ : **Adresse**, $DA$ : **Adresse)** {

L'autorité $DA$ est-elle autorisée à exécuter cette fonction ?

**OUI** ? alors faire :

- Changer l'attribut $eligible$ de $user$ à VRAI.

**NON** ? ne rien faire

}

...

Fonction **retirer(** $user$ : **Adresse**, $DA$ : **adresse)** {

L'autorité $DA$ est-elle autorisée à exécuter cette fonction ?

**OUI** ? alors faire :

- Changer l'attribut $eligible$ du citoyen $user$ à FAUX.

**NON** ? ne rien faire

}

...

Fonction **estEligible(** $adresse$ : **Adresse)** {

Le citoyen ayant l'adresse $adresse$ est-il éligible ?

**OUI** ? alors faire :

- Retourner Vrai.

**NON** ? - Retourner Faux.

}

}

FIGURE 4.8 – Contrat pseudo-code « Preuve d'éligibilité »

**Contrat d'initiation, validation et suivi de collecte de type « Pétition »**

Après la vérification de l'éligibilité d'un citoyen, il peut décider d'initier une nouvelle pétition qui doit ensuite être validée par une autorité compétente. Après la validation, la possibilité est donnée d'autres citoyens peuvent la signer ; le pétitionnaire peut continuer à suivre l'évolution et voir les signatures qui seront ajoutées jusqu'à la fermeture (Il n y a plus aucun intérêt à garder les statistiques d'une pétition secrètes une fois celle-ci terminée).

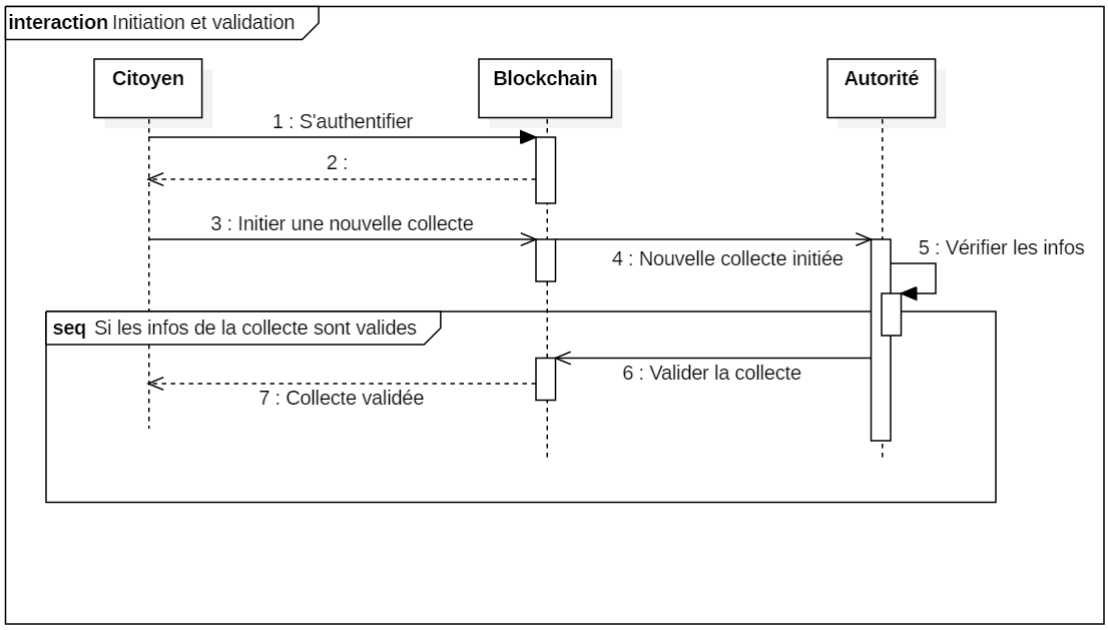


FIGURE 4.9 – Diagramme de séquence : Initiation et validation de pétition

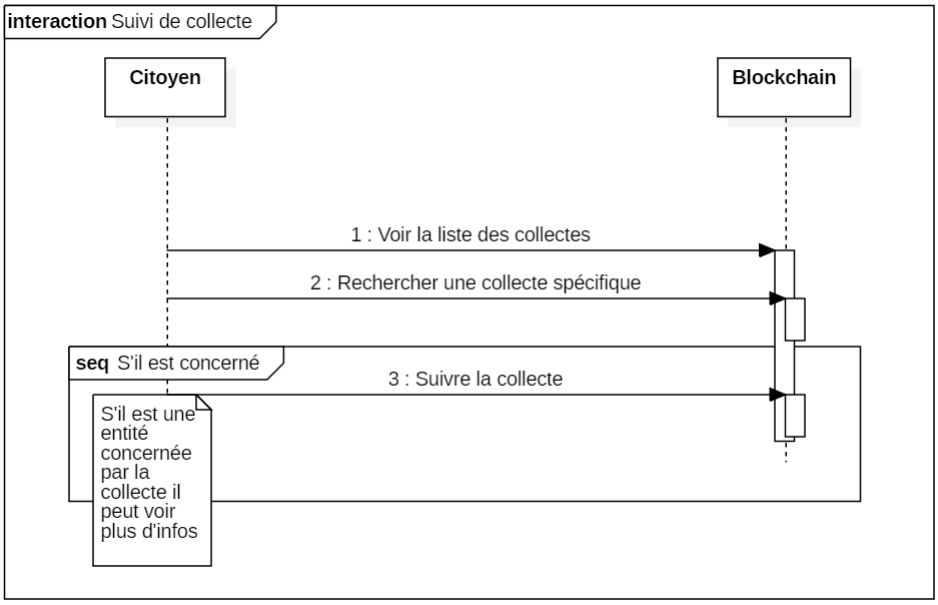


FIGURE 4.10 – Diagramme de séquence : Suivi de pétition

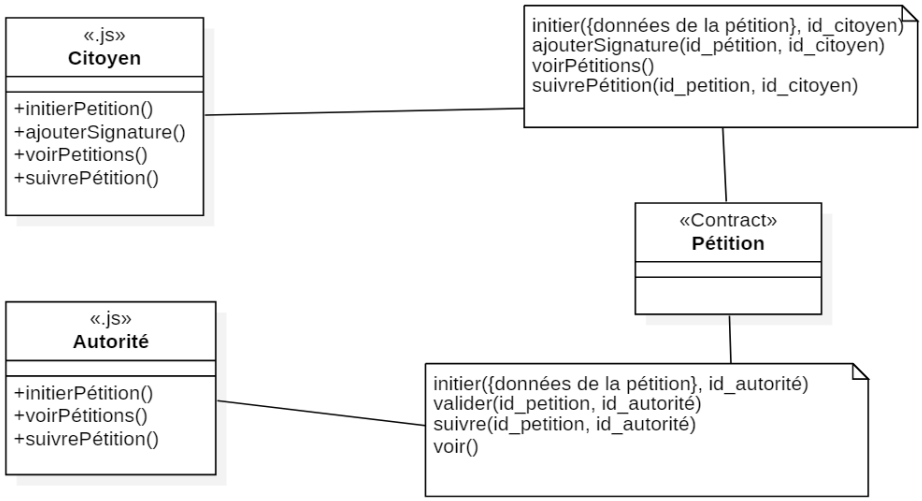


FIGURE 4.11 – Diagramme de classe Stéréotypé : Contrat Initiation, Validation, suivi de pétition

Le contrat correspondant peut se présenter comme suit :

contrat Petition {

**Structure** Citoyen {

adresse : **Adresse ou Clé publique** ;

eligible : **Vrai / Faux** ;

}

**Structure** Autorité {

adresse : **Adresse ou clé publique** ;

... (autres infos)

}

**Structure** Collecte {

id : **Identifiant de la collecte** ;

debut : **Date** ;

fin : **Date** ;

zone : **Identifiant de la zone** ;

objectif : **Nombre de signatures visé** ;

description : **Texte** ;

etat : (En attente de validation / Ouverte / Fermée) ;

initiateur : **Citoyen** ;

}

...

// Listes des collectes, citoyens, autorités

**Liste Collecte** pétitions : **Liste de toutes les pétitions** ;

**Liste Autorité** autorités : **Liste de toutes les autorités** ;

**Liste Citoyen** citoyens : **Liste de tous les citoyens** ;

...

Fonction **initier(** {$debut$, $fin$, $zone$, $objectif$, $description$ }, $initiateur$ : **Adresse){**

Le citoyen $initiateur$ est-il éligible ?

**OUI** alors Faire :

Les données ($debut$, $fin$, $zone$, $objectifs$, $description$) sont-elles valides ?

**OUI** ? Alors faire :

- Créer une nouvelle instance de Collecte avec les attributs données.

- Soumettre cette collecte à l'étape de validation.

**NON** ? Ne rien faire

**NON** ? Ne rien faire

}

...

Fonction **valider(**$DA$ : **Adresse**, $pet$ : **IdentifiantCollecte){**

L'autorité $DA$ a t-il le droit de valider une pétition ?

**OUI** ? alors faire :

La pétition $pet$ existe t-elle ?

Si **OUI** ? alors Faire :

- Changer l'état de la pétition à « Ouverte ».

- Créer le jeton d'autorisation de signature.

Si **NON** ne rien faire

- Refuser la pétition et donner les raisons de ce refus.

**NON** ? Ne rien faire

}

...

Fonction **signer(**$pet$ : **IdentifiantCollecte**, $signataire$ : **Adresse){**

Le citoyen $signataire$ est-il éligible ?

Si *OUI* ? alors Faire :

La pétition $pet$ existe t-elle ?

**OUI** ? alors Faire :

La pétition $pet$ est-elle ouverte ?

**OUI** alors Faire :

Le citoyen $signataire$ est-il autorisé à signer la pétition $pet$ ?

**OUI** ? alors Faire :

Le citoyen $signataire$ a t-il déjà signé la collecte $pet$ ?

**OUI** alors Faire :

- Ajouter la signature de *signataire* à la collecte.

- Exécuter toutes les actions post - ajout de signature.

**NON** ? alors ne rien faire

**NON** ? ne rien faire

**NON** ? ne rien faire

**NON** ? ne rien faire

**NON** ? ne rien faire

}

...

Suivi d'une pétition par son initiateur.

Fonction **suivre(***pet* : **IdentifiantCollecte**, *citoyen* : **Adresse ){**

La collecte *pet* existe t-elle ?

**OUI** ? alors Faire :

Le citoyen est-il l'initiateur de la petition *pet* ?

**OUI** alors Faire :

- Retourner toutes les infos concernant la pétition (description, nombre de signatures, statistiques)

**NON** ? ne rien faire

**NON** ? ne rien faire

}

...

// Suivi d'une pétition par une autorité.

Fonction **suivre(***pet* : **IdentifiantCollecte**, *DA* : **Adresse ){**

La collecte *pet* existe t-elle ?

**OUI** ? alors Faire :

L'autorité *DA* a t-elle le droit de voir toutes les informations sur la petition *pet* ?

**OUI** alors Faire :

- Retourner toutes les informations concernant la pétition ;

**NON** ? ne rien faire

**NON** ? ne rien faire

}

...

// Suivi des pétitions par le public.

Fonction **voirPétitions(){**

- Retourner les descriptions de toutes les collectes $petitions$ ;

}

}

FIGURE 4.12 – Contrat pseudo-code « Initiation, Validation, Signature de Pétition »

**Contrat d'initiation, validation et suivi de collecte de type « Sponsorship »**

Une autorité peut initier une nouvelle collecte de signatures pour le parrainage de candidats d'une élection, cette collecte doit ensuite être validée par une autre autorité selon des règles du contrat. Après la validation, les citoyens peuvent signer pour leurs candidats ; l'autorité (initiatrice) et les candidats concernés peuvent suivre l'évolution de la collecte et voir les signatures qui seront ajoutées en leur noms jusqu'à la fermeture (Une fois leurs objectifs atteints, les candidats quittent la collecte).

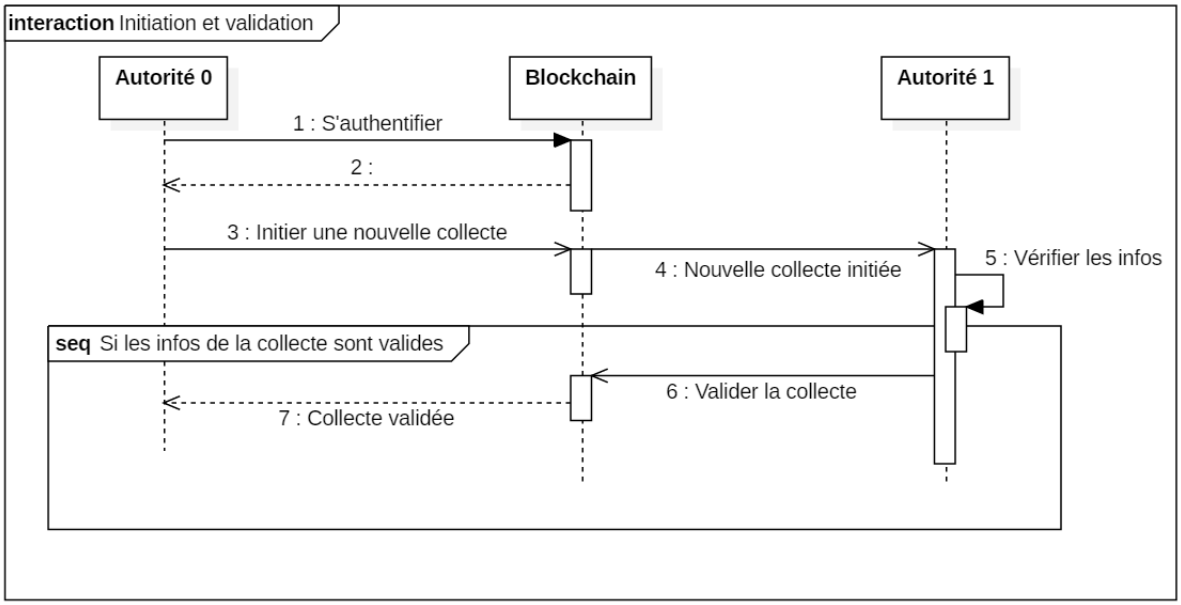


FIGURE 4.13 – Diagramme de séquence : Initiation et validation de collecte type « Sponsorship »

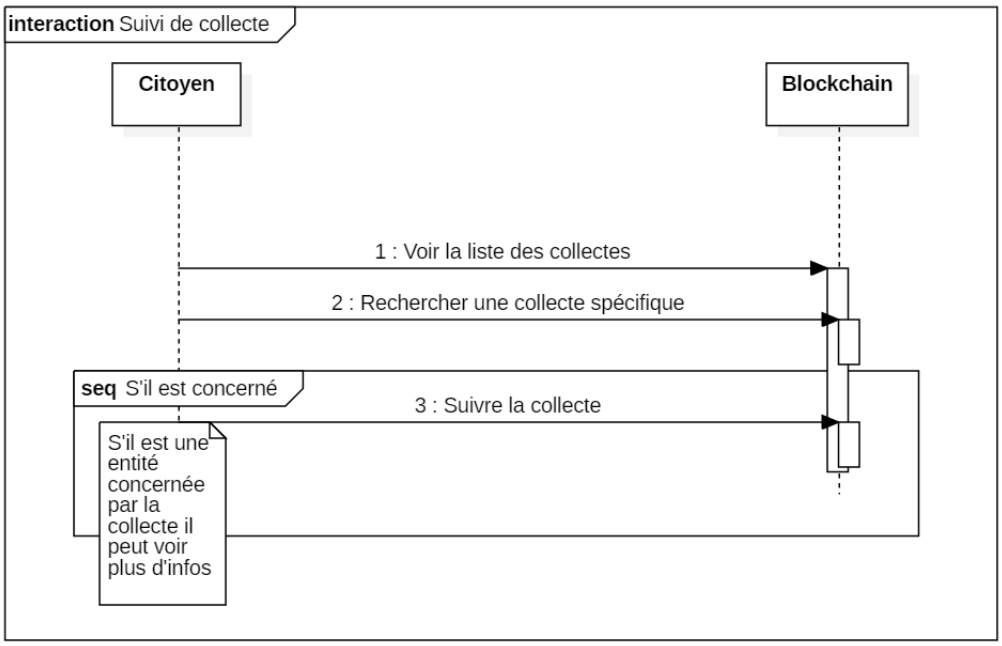


FIGURE 4.14 – Diagramme de séquence : Suivi de collecte type « Sponsorship »

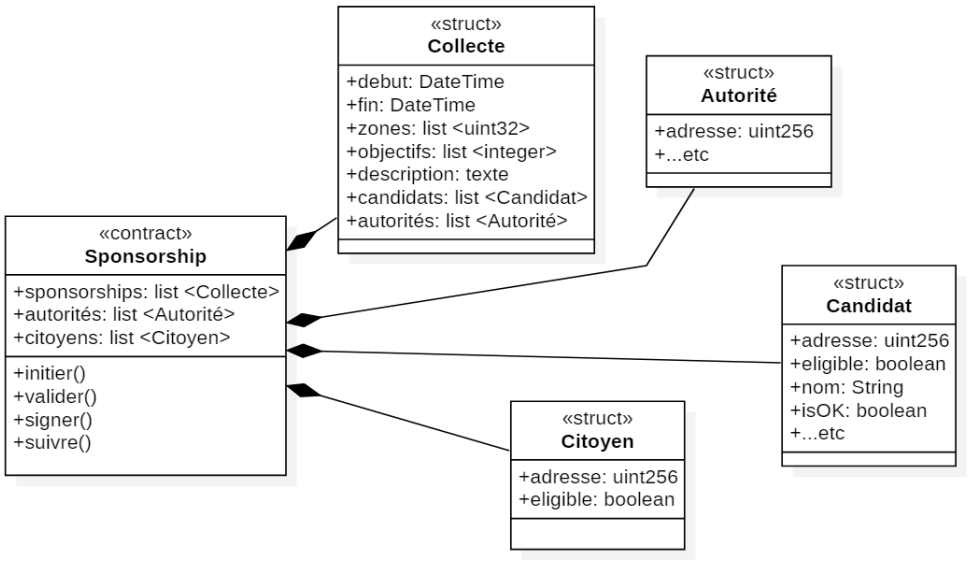


FIGURE 4.15 – Diagramme de classe Stéréotypé : Contrat Initiation, Validation, suivi de collecte Sponsorship

Le contrat correspondant se présente comme suit :

contrat Sponsorship {

**Structure** Citoyen {

adresse : **Adresse ou Clé publique**;

eligible : **Vrai / Faux**;

}

**Structure** Autorité {

adresse : **Adresse ou Clé publique**;

... (autres infos)

}

**Structure** Candidat {

id : **Identifiant du candidat**;

adresse : **Adresse ou Clé publique**;

eligible : **Vrai / Faux**;

nom : **Texte (Nom du candidat ou de son parti politique)**

isOK : **Vrai ou Faux, pour savoir si le candidat a fini sa collecte**;

... (Autres informations);

}

**Structure** Collecte {

id : **Identifiant de la collecte**

debut : **Date**;

fin : **Date**;

zones : **Tableau des zones**;

objectifs : **Tableau des objectifs correspondants aux zones**;

description : **Texte**;

etat : (En attente de validation / Ouverte / Fermée);

candidats : **Liste des candidats**;

initiateur : **Autorité ayant initié la collecte**;

}

// Listes des collectes, citoyens

**Liste Collecte** sponsorships : **Liste de toutes les collectes**;

**Liste Autorité** autorités : **Liste de toutes les autorités**;

**Liste Citoyen** citoyens : **Liste de tous les citoyens**;

Fonction **initier(**{$debut$, $fin$, $zones$, $objectifs$, $description$, $candidats$ }, $initiateur$ : **Adresse de l'Autorité){**

L'autorité $initiateur$ figure t-elle parmi les autorités indiquées ?

**OUI** alors Faire :

Les données ($debut$, $fin$, $zones$, $objectifs$, $description$) sont-elles valides ?

**OUI** alors faire :

Les $candidats$ sont-ils des citoyens éligibles ?

**OUI** ? alors faire :

- Créer une nouvelle instance de collecte Sponsorship avec les attributs données.

- Soumettre cette collecte à l'étape de validation.

**NON** ? Ne rien faire

**NON** ? Ne rien faire

**NON** ? Ne rien faire

}

...

Fonction **valider(**$DA$ : **Adresse de l'Autorité**, $sponsorship$ : **Identifiant de la Collecte){**

L'autorité $DA$ a t-elle le droit de valider une collecte de type Sponsorship ?

**OUI** ? alors faire :

La collecte $sponsorship$ existe t-elle ?

**OUI** ? alors faire :

L'autorité $DA$ est elle différente de celle qui a initié la collecte $sponsorship$ ?

Si **OUI** ? alors Faire :

- Changer l'état de la collecte à « Ouverte ».

- Créer le jeton d'autorisation de signature.

Si **NON** ne rien faire

- Refuser la collecte et donner la les raisons du réfus.

**NON** ? Ne rien faire

**NON** ? Ne rien faire

**}**

Fonction **signer(**$sponsorship$ : **Collecte**, $signataire$ : **Adresse du signataire**, $candidat$ : **Identifiant du Candidat)}**

Le citoyen $signataire$ est-il éligible ?

Si *OUI* ? alors Faire :

La collecte $sponsorship$ existe t-elle ?

**OUI** ? alors Faire :

La collecte $sponsorship$ est-elle ouverte ?

**OUI** alors Faire :

Le citoyen $signataire$ est-il autorisé à signer la collecte $sponsorship$ ?

**OUI** ? alors Faire :

Le citoyen $signataire$ a t-il déjà signé la collecte $sponsorship$ ?

**OUI** alors Faire :

Le candidat $candidat$ est il dans la collecte $sponsorship$ ?

**OUI** alors Faire :

- Ajouter la signature de $signataire$ à la collecte $sponsorship$ au nom du candidat $candidat$ et dans l'index de la zone correspondant au signataire.

- Exécuter toutes les actions post - ajout de signature.

**NON** ? alors ne rien faire

**NON** ? alors ne rien faire

**NON** ? ne rien faire

**NON** ? ne rien faire

**NON** ? ne rien faire

**NON** ? ne rien faire

}

// Suivi d'une collecte de signatures par un candidat n'ayant pas encore terminé sa collecte.

Fonction **suivre(***sponsorship* : **Identifiant de la Collecte**, *candidat* : **Identifiant du Candidat )}**

La collecte *sponsorship* existe t-elle ?

**OUI** ? alors Faire :

Le candidat *candidat* est-il concerné par la collecte *sponsorship* ?

**OUI** alors Faire :

- Retourner toutes les statistiques liées au candidat *candidat* (description, nombre de signatures etc) ;

**NON** ? ne rien faire

**NON** ? ne rien faire

}

// Suivi d'une collecte de signatures par une autorité.

Fonction **suivre(***sponsorship* : **Identifiant de la collecte**, *DA* : **Adresse de l'Autorité )}**

La collecte *sponsorship* existe t-elle ?

**OUI** ? alors Faire :

L'autorité DA est-elle le droit de voir les statistiques de la collecte *sponsorship* ?

**OUI** alors Faire :

- Retourner toutes les informations concernant la collecte *sponsorship* ;

**NON** ? ne rien faire

**NON** ? ne rien faire

}

// Description des collectes de signatures vues par le public.

Fonction **voirCollectes(){**

- Retourner les descriptions de toutes les collectes $sponsorships$ ;

}

}

### 4.1.4 Conception de l'application Web

Il nous faut à présent concevoir l'application Web qui servira d'interface entre les utilisateurs et la Blockchain. Dans cette section nous décortiquerons toutes les étapes de conception de la plate-forme Web.

#### Interactions avec l'application Web

L'objectif principale de l'interface Web est de fournir des formulaires pour la saisie des données de l'utilisateur. Les diagrammes 4.16, 4.17, 4.18 décrivent ces interactions.

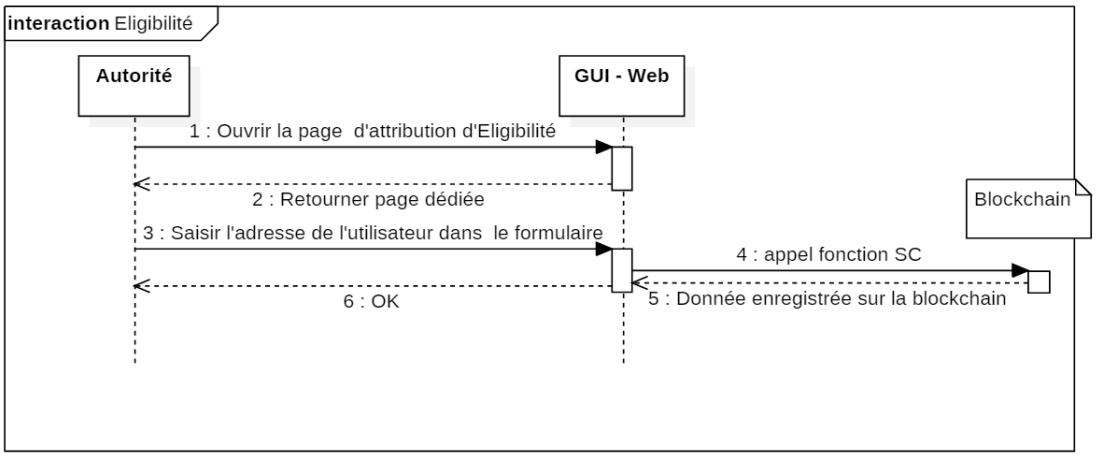


FIGURE 4.16 – Diagramme de séquence - Web : Éligibilité

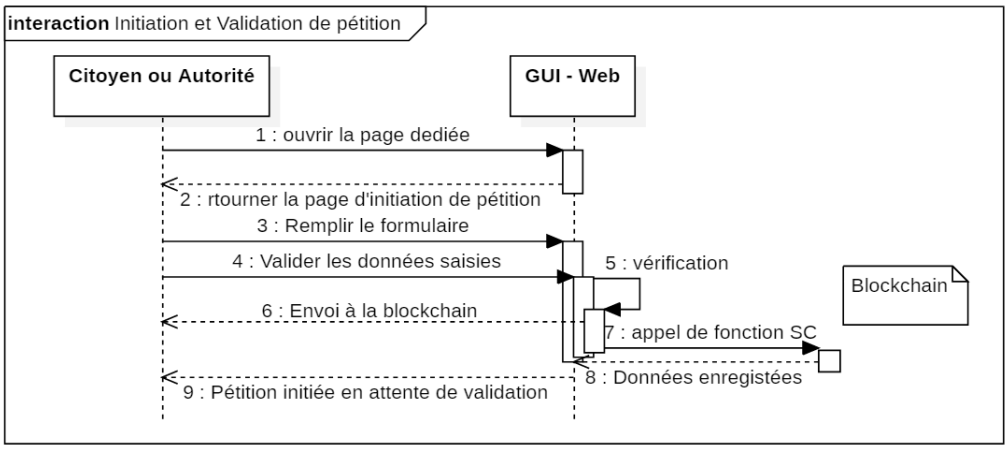


FIGURE 4.17 – Diagramme de séquence - Web : Initiation et validation

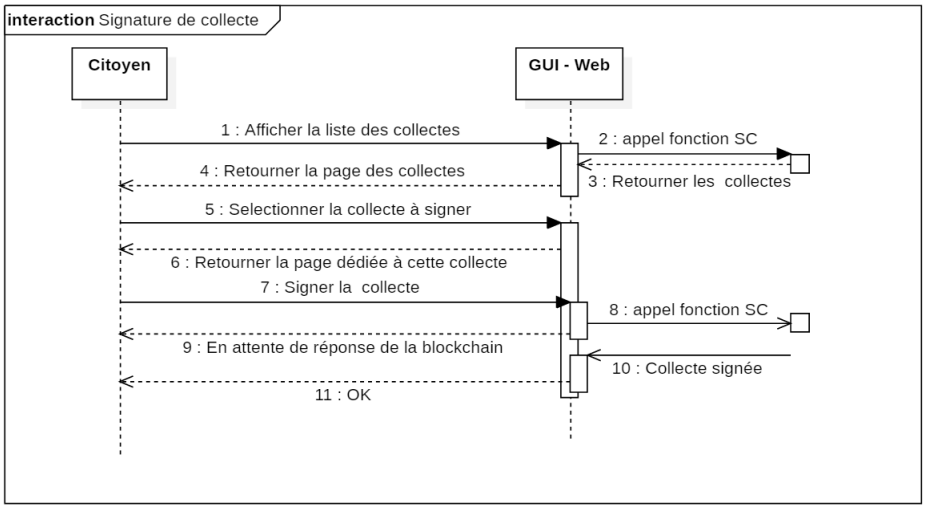


FIGURE 4.18 – Diagramme de séquence - Web : Signature

### Interfaces utilisateurs

Nous ne pouvons pas présenter toutes les interfaces du système dans ce rapport, cependant en voici quelques unes jugées importantes.

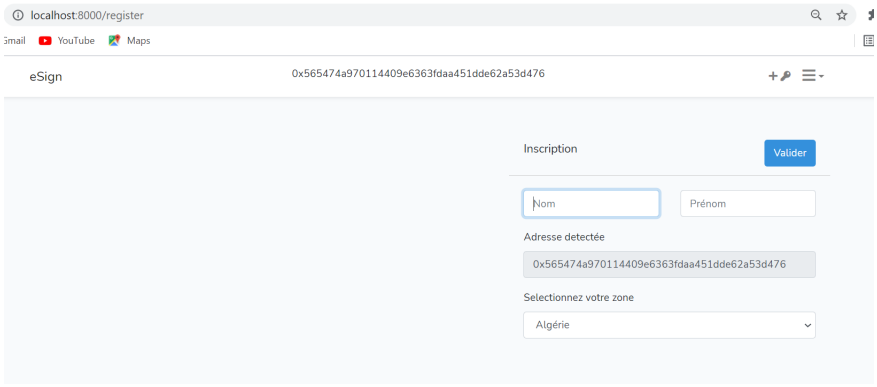


FIGURE 4.19 – Page d'enregistrement de clé


FIGURE 4.20 – Création de pétition


FIGURE 4.21 – Page des pétitions


FIGURE 4.22 – Page individuelle de Pétition

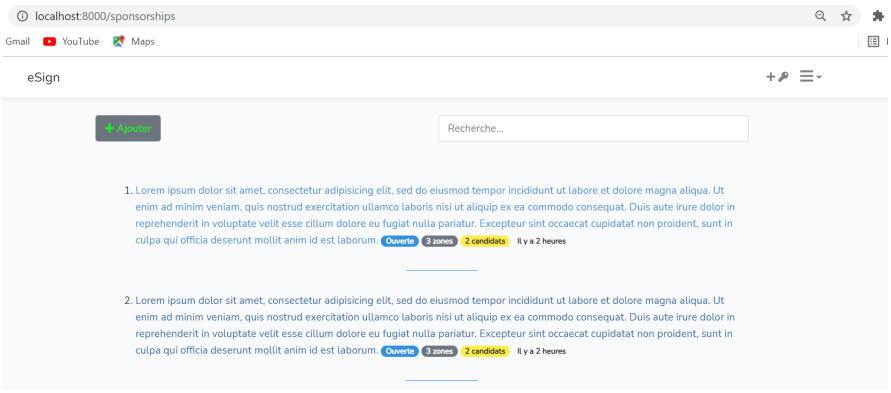


FIGURE 4.23 – Page des Sponsorship

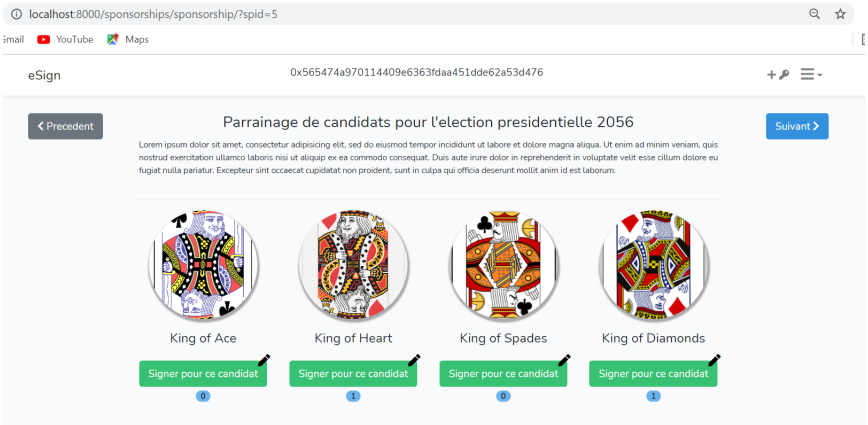


FIGURE 4.24 – Page individuelle de Sponsorship

### 4.1.5 Codage de l'application Web

#### Architecture MVC

L'architecture logicielle MVC permet de distinguer la logique du reste de l'interface utilisateur. Pour ce faire, il sépare l'application en trois parties : le modèle, la vue et le contrôleur.

Le modèle gère les comportements fondamentaux et les données de l'application. Il peut répondre aux demandes d'informations, aux instructions qui modifient l'état de ses informations et même notifier les observateurs dans les systèmes événementiels lorsque les informations changent. En bref, ce sont les données et la gestion des données de l'application.

La vue fournit l'interface utilisateur de l'application, elle restituera les données du modèle sous une forme beaucoup plus adaptée et compréhensible pour l'utilisateur. Le contrôleur quand à lui, il reçoit les entrées de l'utilisateur et effectue des appels aux objets de modèle et à la vue pour effectuer les actions appropriées.

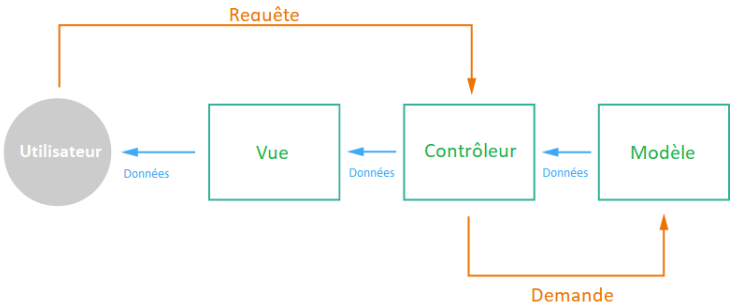


FIGURE 4.25 – Architecture MVC

L'avantage dans le développement de notre DApp est que la majeure partie de la logique de notre système reste sur la Blockchain. Cela a allégé considérablement le travail à faire dans cette partie, l'objectif était de recueillir les actions et les données des utilisateurs, les acheminer vers la Blockchain puis refaire la même chose dans le sens inverse.

Nous avons utilisé les outils de création d'application Web suivants :

**Laravel 8**

Laravel est un framework open-source PHP très connue et très utilisé par les développeurs Web, du fait de la simplicité et de la clarté du code. Il nous a permis de gérer le routage des requêtes du DApp et le rendering de certaines pages Web (avec **blade**).

**Vue.js**

Vue.js est un framework open-source et très puissant pour le développement d'applications Web côté client, qui s'inspire des principes de design du développement côté serveur et qui applique ces principes aux éléments HTML.

**Web3**

Web3.js est un ensemble de librairies qui permettent d'interagir avec une Blockchain Ethereum à travers des protocoles prédéfinis.

### 4.1.6 Intégration

Cette phase consiste à combiner les deux parties du système, en réalité les composantes ne seront jamais intégrées totalement, elles seront liées par le code JavaScript, plus précisément par **Web3**, l'architecture finale ressemble au suivant 4.26

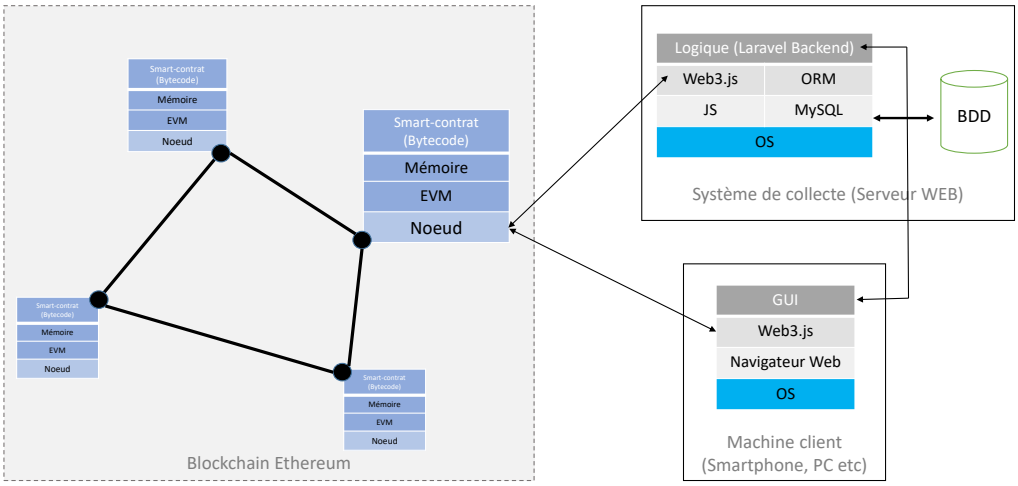


FIGURE 4.26 – Architecture finale du DApp

Pour des raisons divers, le DApp de collecte de signature n'a pas été déployée sur des serveurs publiques disponibles à partir d'Internet. Par contre nous avons mis en place un environnement assez performant de test sur un LAN domestique. La Blockchain en elle même ainsi que les comptes et adresses associées sont fournis et gérés par Ganache et Truffle qui font tous les deux parties de la suite Truffle [2].

# Conclusion

Dans ce chapitre nous avons essayé de présenter le processus de conception de notre DApp de collecte de signature. Nous avons utilisé une approche de modélisation et de conception peu connue mais pourtant assez performante :**ABCDE**. Ce chapitre est en réalité une preuve de concept du modèle présenté dans le chapitre 3. Nous avons vu qu'il était très simple de faire une implémentation type de la proposition et d'en évaluer les performances.

2. www.trufflesuite.com

# CONCLUSION GÉNÉRALE

Le travail présenté dans ce rapport porte essentiellement sur la Blockchain et son application dans le processus de sélection de candidats plus précisément d'un système de collecte de signature de parrainage de candidats pour des élections, aussi pour des pétitions de referendum d'initiatives populaires.

Dans le premier chapitre nous avons discuté du vote électronique, de quelques approches existantes, leurs points forts et leurs lacunes. Nous avons vu qu'il y avait plusieurs catégories de système de e-voting notamment les DRE et les OMRs. Nous avons étudié le fonctionnement d'une implémentation de vote par Internet (le cas de l'Estonie), puis nous avons essayé d'en retenir les inconvénients.

Le deuxième chapitre du rapport est consacré à la Technologie Blockchain. Nous avons étudié les principes fondamentaux de la Blockchain, nous avons vu ce qu'est une base de données décentralisée, puis nous avons compris la notion de consensus et comment la Blockchain permettait de mettre de la confiance dans un réseau sans tiers de confiance. Nous avons constater que le caractère décentralisé et l'immutabilité font de la Blockchain l'une des meilleures solutions pour que le vote électronique devienne plus ouvert, plus transparent, et plus vérifiable.

Après l'étude des différentes technologies et concepts clés de la Blockchain, le chapitre 3 est consacré à la présentation et à la description de notre modèle de collecte. Nous avons commencé par définir les différents rôles clés dans une collecte de signatures, d'un côté pour une pétition sous sa forme la plus connue, notamment celle destinée aux referendums d'initiatives populaires et celle destinée aux parrainages de pré-sélection des candidats d'élections présidentielles. Nous

avons noter les différences entres ces rôles par des fonctions ou méthodes, toutes présentes sur des contrats intelligents. A travers ces contrats, nous avons posé des règles de conduite que doivent suivre tous les acteurs clés des collectes. Toutes les actions se feront sous forme de transactions et ces transactions sont toutes enregistrées sur la Blockchain. A travers tout ceci, il est possible de savoir à n'importe quel *moment*, *qui* fait *quoi* et *comment* il le fait de manière très transparente et très contrôlée.

Une fois, le modèle proposé, nous avons jugé nécessaire de fournir une preuve de concepts, ce que rapporte la dernière partie de ce document. Nous avons étudié les concepts clés des applications décentralisées de manière générale puis de manière approfondie leur applicabilité à notre modèle. Durant l'étape d'analyse et de conception, nous avons utilisé l'approche **ABCDE** pour **A**gile **B**lockchain **D**App **E**ngineering pour la modélisation et l'implémentation. Les applications décentralisées sont généralement divisés en deux composantes :

- La première composante regroupe tout ce qui interagit directement avec la Blockchain et toutes les sous-composantes que nous avons jugées importantes. Nous avons utilisé la pile Ethereum (*Truffle*, *Ganache-GUI*, *Ganache-CLI* et **NodeJS** pour maintenir le tout ensemble) dans la mise en place de cette composante.
- La deuxième composante regroupe les interfaces utilisateurs et les sous-composantes de gestion de ces interfaces (**Laravel 8** pour le routage et les fonctions de Middleware, **VueJS** pour le rendering des pages, **Web3**, **Metamask** pour la gestion des clés)

La Blockchain est une approche innovante pour stocker des informations, exécuter des transactions, des fonctions et établir la confiance dans un environnement ouvert. Le système de collecte de signatures joue un rôle clé dans les processus de lois et les élections, donc dans la démocratie. Dans ce mémoire, nous avons analysé et montré une manière d'améliorer ces collectes de à l'aide de la Blockchain pour atteindre les objectifs optimaux d'efficacité et de transparence.

Des améliorations sont envisageables, notamment du point de vue des smart-contrats, il est possible d'ajouter plus de règles de validations des signatures, des pétitions, des collectes de parrainages ; Elles pourraient avoir des propriétés individuelles ou des propriétés de groupes. Les smart-contrats sont très extensibles, ils peuvent subir toute sorte d'adaptation dans la mesure du possible.

# BIBLIOGRAPHIE


[1] *Renforcer la démocratie : une stratégie destinée à améliorer lintégrité des élections dans le monde : rapport de la commission mondiale sur les élections, la démocratie et la sécurité Septembre 2012*. International IDEA, 2012.

[2] *General Framework of Electronic Voting and Implementation thereof at National Elections in Estonia*. Tallin, 2017.

[3] Imran Bashir. *Mastering blockchain : distributed ledger technology, decentralization and smart contracts explained*. Packt Publishing Ltd., 2018.

[4] Sisir Debnath, Mudit Kapoor, and Shamika Ravi. The impact of electronic voting machines on electoral frauds, democracy, and development. *SSRN Electronic Journal*, 01 2017. doi:10.2139/ssrn.3041197.

[5] Peter Y. A. Ryan Feng Hao. *REAL-WORLD ELECTRONIC VOTING DESIGN, ANALYSIS AND DEPLOYMENT*. Taylor & Francis Group, 2017.

[6] Arthur Gervais, Ghassan Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. pages 3–16, 10 2016. doi:10.1145/2976749.2978341.

[7] Blockchain Technology for Business Hyperledger. *An Introduction to Hyperledger*.

[8] IDEA. *Introducing Electronic Voting : Essential Considerations*. Policy Paper. International Institute of Democracy an Electoral Assistance, 2011.

[9] Robert Krimmer. *The Evolution of E-voting : Why Voting Technology is Used and How it Affects Democracy*. PhD thesis, 11 2012.

[10] Laurent Leloup. *Blockchain, la révolution de la confiance*. Eyrolles, 2017.

[11] Antony Lewis. *The basics of Bitcoins and Blockchains : An Introduction to cryptocurrencies and the Thechnoloy that Powers them*. Mango Publishing, 2018.

[12] Lodovica Marchesi, Michele Marchesi, and Roberto Tonelli. Abcde –agile block chain dapp engineering. *Blockchain : Research and Applications*, 1 :100002, 12 2020. doi:10.1016/j.bcra.2020.100002.

[13] Nadia Nata. *Pourquoi recourir au parrainage électoral*. WATHI, 2019.

[14] Harald Baldersheim Norbert Kersting. *Electronic Voting and Democracy : A comparative Analysis*. Palgrave macmillan, 2004.

[15] Bikramaditya Singhal, Gautam Dhameja, and Priyansu Sekhar Panda. *Beginning blockchain : a beginners guide to building blockchain solutions*. Apress, 2018.

[16] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Halderman. Security analysis of the estonian internet voting system. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 703–715, 11 2014. doi:10.1145/2660267.2660315.

[17] Anna-Greta Tsahkna. E-voting : Lessons from estonia. *Centre for European Studies*, 2013.

[18] Max Hooper Vikram Dhillon, David Metcalf. *BLOCKCHAIN ENABLED APPLICATIONS : Understanting the Blockchain Ecosystem and How to Make it Work for You*. APRESS, 2017.

[19] Scott Wolchok, Eric Wustrow, J. Halderman, Hari Prasad, Arun Kankipati, Sai Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. Security analysis of india's electronic voting machines. pages 1–14, 09 2010. doi:10.1145/1866307.1866309.