الجمـــهوريـــة الجـــزائريـــة الديـمقـراطيـــــــة الشـعبيـــــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعـلـيـم الـعـالـــي و الـبـحـث الـعـلـمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جـامـعـة عمّـــار ثـلـيـــجـــي بالأغـــواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كـلـيـــــة الـعـلـــــوم
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire de MASTER

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Réseaux, Systèmes et Applications Réparties

Par:

KATIA INES CHERIFI **ISRA BENDJEMA**

THEME

Système de vote distribué préservant la confidentialité

Soutenu publiquement le 14/06/2022 devant le jury composé de:

Laradj CHELLAMA	*M.A.A*	*Président*
Noureddine CHAIB	*M.C.A*	*Examinateur*
Leila BENAROUS	*M.C.B*	*Encadreur*

Année Universitaire : 2021/2022.

DÉDICACE

Nous dédions ce modeste travail, comme preuve de respect, de gratitude, et de reconnaissance

A nos chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études,

A mon binôme et sœur pour son soutien moral, son aide, sa patience et sa compréhension tout au long ce projet.

A à tous ceux qui ont contribué de près ou de loin à la réalisation de cette étude.

REMERCIEMENT

Nous adressons nos remerciements tout d’abord, à Dieu le tout puissant, de nous avoir donné la force, la patience et durant 17 ans d’étude et de nous aider de réaliser ce modeste travail.

Notre gratitude s’adresse à notre directrice de mémoire Dr. Leila Benarous pour son encadrement, son orientation, ses conseils et la disponibilité qu’elle nous a témoignée pour nous permettre de mener à bien ce travail.

De plus, nous tenons à remercier nos parents de nous avoir supportés et aides durant toutes ces années.

Enfin nous tenons également à remercier l'ensemble des membres de jury, nous les remercions tous pour l'intérêt qu'ils ont porté à notre travail en participant à ce jury.

TABLE DES MATIÉRS

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES EQUATIONS

LISTE DES ABREVIATIONS

AE: Autorité d'Eneregistrement. 20, 21 , 44

API: Application Programming Interface. 36

ABI: Abstract Binary Interface. 38

BES: Blockchain-based Electronic Voting scheme. 15, 16

DATE: Decentralized, Anonymous, and Transparent E-voting system. 14, 16

DLT: Distributed Ledger Technologie. 15

DOS: Deni of Service. 15, 16

Dapp: Decentrelized Application. 36, 41

EVM: Ethereum Virtual Machine. 34

JSON: JavaScript Object Notation. 38

NPM: Node Packet Manager. 36

OVN : Open Vote Network. 14, 16

POW : Proof Of Work. 10, 11, 12 , 15 , 16 , 24 , 45 , 60

POS : Proof Of Stake. 10 , 11 ,12

POA: Proof Of Authority. 10 , 21 , 24 , 25 , 44 , 59 ,60

RPC: Remote Procedure Call. 35, 40

SHA: Secure Hash Algorithm. 15,50

SDK: Software Development Kit. 37

UML: Unified Modeling Language. 32, 59

UI: User Interface. 35

ملخص

لقد تجاوزت مجالات تطبيق البلوكشين تطبيق العملة المشفرة الموزع الذي لا يعتمد على سلطة والتي كانت السبب في اشتهار استعماله. جلبت هذه التقنية أفكارًا ومفاهيم جديدة في مجال الأمن تضمن ثبات البيانات وتقوي ثقة المستخدم. Blockchain عبارة عن مجموعة من الكتل المتسلسلة يتم نشرها في شبكة واحد الى واحد.

حاليًا، تجذب تكنولوجيا الــ Blockchain الانتباه على مستويات مختلفة وهي مناسبة لعدة مجالات، مثل: تسجيل تبادل الممتلكات وأنظمة التصويت والرعاية الصحية.

في عملنا، تعلمنا أولاً عن تقنية Blockchain واستخداماتها وكيفية تنفيذها. ثم درسنا أنظمة التصويت الحالية. أخيرًا، قمنا بتصميم تطبيق تصويت إلكتروني شفاف وآمن ويحافظ على الخصوصية قائم على Blockchain. قمنا ببرمجة تطبيقنا في Ethereum من خلال وضع قواعد التصويت كعقود ذكية. كانت الواجهة الأمامية في شكل تطبيق ويب يمكن للناخبين استخدامه بسهولة.

الكلمات الرئيسية: Blockchain، التصويت الإلكتروني، الأمان، الخصوصية، تطبيق الويب، Ethereum، شفاف

RESUME

Les domaines d'application de la blockchain sont allés bien au-delà de l'application distribuée de monnaie cryptographique sans autorité qui lui a valu sa réputation. Cette technologie a apporté de nouvelles idées et concepts dans le domaine de la sécurité qui assure l'immuabilité des données et renforce la confiance des utilisateurs. La blockchain est un ensemble de blocs chaînés et publiés dans un réseau peer-to-peer.

Actuellement, la blockchain attire l'attention à différents niveaux et convient à plusieurs domaines, tels que : l'enregistrement d'échange de propriété, les systèmes de vote et les soins de santé.

Dans notre travail, nous avons d'abord étudié la technologie blockchain, ses usages et son implémentation. Ensuite, nous avons étudié les systèmes de vote existants. Enfin, nous avons proposé un design pour un système de vote électronique basé sur la blockchain qui est transparent, sécurisé et préservant la confidentialité. Nous avons implémenté le backend de notre application sur Ethereum en définissant les règles de vote sous forme de contracts intelligents et le front-end sous la forme d'une application Web pouvant être utilisée facilement par les électeurs.

Mots-clés : Blockchain, vote électronique, sécurité, confidentialité, application web, Ethereum, transparent.

ABSTRACT

The blockchain application domains have gone far beyond the distributed authority-free crypto money application that got it its fame. This technology has brought new ideas and concepts in the field of security that ensures the immutability of data and strengthens user confidence. The blockchain is a set of blocks chained and published in a peer-to-peer network.

Currently, blockchain is attracting attention at different levels and it is appropriate for several areas, such as property exchange registration, voting systems and healthcare.

In our work, we first learned about blockchain technology, its uses and its implementation. Then, we studied the existing voting systems. Lastly, we designed a transparent, secure and privacy-preserving blockchain-based e-voting application. We implemented our application over Ethereum by setting the voting rules as smart contracts. The front-end was in the form of a friendly web application that can be used with ease by the voters

Key-words: Blockchain, e-voting, security, privacy, web application, Ethereum, transparent.

INTRODUCTION GENERALE

La technologie de blockchain est apparue en 1991 et devenue populaire avec l'apparition de la monnaie électronique bitcoin. Cette technologie est un registre décentralisé sur un réseau décentralisé où tous les nœuds communiquent entre eux sans intermédiaire. Cette technologie permet de stocker et de transmettre les données de manière transparente et sécurisée, ainsi, elle aide à éviter la fraude et les activités non autorisées. Toutes les informations sont sauvegardées sur un réseau des ordinateurs et pas sur un seul serveur, ce qui rend difficile l'altération ou l'écrasement des données par les attaquants. La technologie de blockchain est utilisée dans plusieurs domaines d'application autre que le domaine de la crypto-monnaie.

Notre système démocratique permet au peuple d'exercer le pouvoir de voter. Pour exprimer sa volonté, à l'occasion d'un scrutin. C'est une façon d'exercer la citoyenneté. Le vote se déroule avec plusieurs manières et la plus célèbre méthode utilisée est le vote traditionnel à l'urne où le votant de se déplacer au centre de vote pour voter.

Le vote traditionnel à l'urne nécessite des moyens matériels, humains et financiers importants. En plus, ce vote ne garantit ni la traçabilité, ni l'intégrité des résultats. Les élections électroniques centralisés, minimisent les taches de vote, les ressources et le temps,. En plus, ils automatisent le dépouillement des bulletins de vote ce qui offre une rapidité accrue et une marge d'erreur beaucoup plus réduite par rapport au vote traditionnel. Cependant, ils ont des inconvénients comme l'absence de l'intégrité des données et la transparence car tout le processus de vote est géré par un serveur centralisé privé au niveau de l'autorité de vote.

Dans notre projet de fin d'étude, nous avons proposé une application web de vote électronique à base blockchain qui assure l'anonymat, la sécurité et la transparence. Ainsi, elle satisfait les conditions de vote comme l'unicité de vote en interdisant la tentative de multi-votes sans la violation de la confidentialité du votant. Le développement de l'application était précédé par une phase de recherche et d'analyse des articles scientifiques, des documentations relatives aux blockchains, des systèmes de votes électroniques en plus des solutions de sécurité et de cryptographie.

Notre mémoire est composé de trois chapitres qui sont structurés comme suit :

- Le premier chapitre présente les différents types de vote avec l'analyse critique de chacun. Ainsi, il présente les concepts fondamentaux de la technologie blockchain.
- Le deuxième chapitre consiste à la présentation de la solution que nous avons proposée en détaillons les différentes phases de système.
- Le troisième chapitre montre les phases de l'implémentation de notre solution proposée, en plus de sa démonstration et son analyse de sécurité.

A la fin, le mémoire est clôturé par une conclusion générale et perspective d'avenir. Ainsi qu'une grille d'auto-évaluation.

CHAPITRE I :

Les systèmes de vote- état de l'art

I.1 Introduction :

Les organisations formelles ou informelles ont recours au vote avec plusieurs techniques pour répondre aux besoins des individus. Le vote suit la règle de la majorité, chaque votant doit voter une seule fois et doit respecter le résultat de vote. Le vote peut être publique ou privé et anonyme. Dans ce chapitre, nous présentons l'analyse préalable qui doit être élaboré avant d'entamer la mise en place de notre système de vote électronique distribué sécurisé qui est à la fois transparent et confidentiel. Pour cela nous présenterons les différentes méthodes de vote existantes et/ou utilisées ainsi que leurs caractéristiques, avantages et inconvénients. Cela comprend le vote traditionnel, le vote électronique classique (système centralisé), le vote électronique basé sur la technologie de blockchain. Nous expliquons aussi la technologie de la blockchain, ses types et son principe de fonctionnement.

I.2 Le vote traditionnel à l'urne :

Le droit de vote permet aux citoyens d'un État de voter pour exprimer leur volonté, à l'occasion d'un scrutin. La première apparition des élections remonte à l'Antiquité, depuis environ 508 av. J.-C., la Grèce antique semble avoir mis en place la première forme de démocratie. Les Grecs avaient une élection (négative) c'est-à-dire que chaque année, les électeurs, qui étaient les hommes propriétaires terriens, étaient invités à voter pour le chef politique ou les "candidats" qu'ils souhaitaient le plus voir exilés pendant les dix prochaines années [1].

La première loi électorale a été promulguée le 6 août 1965 par le président Lyndon Johnson. Il a interdit les pratiques électorales discriminatoires adoptées dans de nombreux États du sud après la guerre civile [2].

Le processus d'un bureau de vote est soumis à la loi électorale. Commençant par l'organisation matérielle du vote à l'affichage de résultat, en passant par les acteurs concernés (président de bureau, assesseurs…), cela garantit la liberté du vote.

Chaque commune de wilaya est divisée, en autant de centres de vote et ce dernier contient plusieurs bureaux selon les besoins et le nombre d'électeurs. Afin d'assurer le bon fonctionnement des opérations électorales, le votant est obligé d'avoir une carte électorale.

Avant l'ouverture des bureaux de vote, les cartes des candidats sont déposées sur une table appelée 'table de décharge'.

Dès l'ouverture du scrutin qui dure de 8hr de matin jusqu'à 19hr généralement, l'électeur peut se déplacer au centre de vote qui est défini sur sa carte électorale. Les électeurs, à leur arrivés, présentent leur carte électorale et une pièce d'identité pour vérifier s'ils ont été inscrits sur le registre électoral. Ensuite l'électeur prend une enveloppe, les cartes des candidats et entre un isoloir pour être libre à choisir. L'isoloir et l'enveloppe permettent d'assurer la confidentialité des votes. Puis il rend son choix dans l'urne. L'urne transparente permet à chacun de vérifier le nombre des voix ce qui offre un très bon niveau de sécurité. L'urne scellée doit être surveillée, de début du scrutin jusqu'au dépouillement par des personnes responsables.

Après cela le votant signe à côté de son nom sur la liste prévue à cet effet. La carte électorale est tamponnée par un assistant avec un timbre portant date du scrutin.

A la fin de scrutin, les centres sont fermés et le processus de dépouillement commence. Une fois tous les bulletins comptés, le secrétaire rédige le procès-verbal. Le président du bureau de vote annonce les résultats et les affiche dans la salle de vote.

Cependant, ce type de vote présente l'inconvénient de l'exigence de la présence et l'identification de tous les électeurs dans le vote après la confirmation de ses existences sur la liste électorale, et exclut ainsi toutes les personnes qui ont eu des empêchements. Lors d'un vote par présence, on se décide très souvent pour un jour et non pas pour une période de temps, car il est plus facile ainsi de garantir le processus entier. En plus, l'inconvénient des coûts d'organisation, il faudrait fournir les besoins nécessaires (des bureaux, des urnes, isoloirs, enveloppes...), comme il est nécessaire aussi intégrer les agents électoraux qui s'occupent de la réalisation, et la surveillance de l'urne qui n'est pas toujours facile à réaliser pour des élections menacées par les fraudes. Ainsi, il est difficile de trouver des personnes confiantes pour les affecter à la surveillance de l'urne et assister au dépouillement de vote. De plus, le vote classique consomme beaucoup de papier utilisé soit par les candidats dans leurs campagnes électorales, soit le jour du scrutin. Ce papier est un gaspillage qui n'est ni bénéfique pour l'environnement ni pour l'économie des nations. Enfin, compter les votes soumis est fastidieux, prend du temps et est propice aux erreurs et aux falsifications.

I.3 Le vote électronique (e-vote) :

Aujourd'hui et dans de nombreux grands pays, la technologie est présente dans les processus électoraux et elle est essentielle à la conduite des élections. La technologie est utilisée pour compiler

Les listes électorales, tracer les limites des circonscriptions, gérer et former le personnel, imprimer les bulletins de vote, mener des campagnes électorales, enregistrer les votes exprimés, compter et consolider les résultats des votes et publier les résultats des élections.

Le vote électronique consiste à utiliser les technologies de l'informatique dans le processus de vote. En utilisant un dispositif (ordinateur, téléphone) qui permet au l'utilisateur d'exprimer son vote et de compter automatiquement le résultat de scrutin.

Un système de vote électronique doit satisfait toutes les propriétés usuelles de sécurité. Pour cela il faut d'abord, garantir l'identité des votants et la confidentialité du vote. Il faut aussi assurer le bon enregistrement du vote. Enfin, il faut valider le compte global des voix, sans oublier qu'il faut interdire la coercition, l'achat de votes ou tout autre type de falsifications.

Ainsi, pour assurer la sécurité et la vérifiabilité des votes, les systèmes de vote modernes reposent sur la cryptographie pour rendre leur sécurité comparable ou meilleure que le vote traditionnel.

I.3.1 Les formes de vote électronique :

Il existe multiples formes du vote électronique :

- **Vote par boîtier** : il s'agit d'une salle de délibération qui contient de boîtiers qui sont utilisés par le votant pour voter. Les boitiers sont reliés avec un ordinateur qui maintient les résultats [3].
- **Vote par internet** : Les électeurs procèdent depuis n'importe quel ordinateur connecté à internet. Il faut se connecter sur le site officiel de vote hébergé sur l'ordinateur du bureau centralisateur. Après les phases d'authentification, l'électeur peut exprimer son vote. Après confirmation, il reçoit un accusé de réception de son choix. Les échanges d'informations nécessitent le réseau internet. L'ordinateur qui fait office de serveur fait les traitements nécessaires, la collecte des votes et leur réception, et du dépouillement à la fin de la durée du scrutin. [3]
- **Machine à voter ou urnes électroniques** : C'est un ordinateur comme il est montré dans la **Figure I.1**, qui est conçu pour recueillir le vote de chaque votant. L'électeur va voter à l'aide d'un appui sur bouton, d'un clic de souris, ou de la saisie d'un numéro. Les votes sont enregistrés en mémoire de cet ordinateur. À la fin de la période de vote, ce système compte et énonce le nombre de voix obtenues par chaque candidat. [3]

Figure I.1: Machine à vote iVotronic [4]

- **Stylo numérique** : Les bulletins de vote sont imprimés sur un papier spécial permettant à la caméra installée dans le stylo numérique de les repérer. Pour que l'électeur peut exprimer son choix, il utilise le stylo numérique, ce dernier réalise des enregistrements constituent une mémorisation de choix de vote. Le bureau centralisateur analyse ces enregistrements en utilisant un logiciel approprié. [3]
- **Vote par téléphone (SMS)** : l'électeur appelle une ligne réservée à la gestion du vote. Après l'identification et l'authentification, il peut exprimer son choix au moyen des touches de son téléphone ou par l'envoi d'un SMS à cette ligne. [3]

Le vote électronique présente de nombreux avantages qui rendent le vote plus facile .

D'abord, Le vote électronique garantie une meilleure participation, efficacité écologique, facilitation des traitements, modernisation de la démocratie… de nombreuses raisons existent pour évoluer du triptyque isoloir – bulletin - urne. L'importance du vote électronique a été encore plus soulignée pendant la pandémie de Covid-19 a pu relancer l'idée du e-vote pour éviter le brassage des populations dans les bureaux de vote. Le e-vote permet votant de voter de n'importe où.

De plus, il est possible d'allonger la durée du vote comme en Estonie, où la plateforme de vote en ligne est utilisable pendant plusieurs jours avant l'ouverture des bureaux de vote physiques.

Le vote à distance permet ensuite d'augmenter et faciliter la participation des citoyens qui ont des empêchements comme la résidence à l'étranger ou sont dans des endroits éloignée de leur résidence de vote (comme les étudiants par exemple).

Ainsi, certains candidats ont parfois du mal à financer l'impression de leurs bulletins, donc ils ne peuvent pas être fournis dans tous les bureaux. Avec le vote électronique, tous les bulletins sont disponibles pour les électeurs. L'autre avantage réside dans la rapidité de traitement des résultats qui peut se réaliser à la fin du scrutin en quelques minutes, sans long dépouillement.

Toutefois, pour des enjeux importants, le vote électronique pose des problèmes de vérification des votes individuels. C'est pour ça que les Pays-Bas ont envisagé de revenir à la méthode de vote traditionnel avant d'abandonner totalement le vote électronique en mai 2008 [5]

Malgré ces avantages, il n'est pas encore largement utilisé. Premièrement, pour garantir l'unicité des votes, il est nécessaire de pouvoir identifier les électeurs. Pour cela un système d'authentification doit être utilisé car les risques de doubler le vote et l'usurpation d'identité sont présents avec l'absence d'authentification Afin qu'il soit réellement efficace, il faut être géré de façon indépendante.

De plus, le vote électronique peut se faire à distance. Dans ce cas, une personne peut facilement usurper plusieurs identités numériques pour effectue un vote plusieurs fois (attaque sybil). C'est difficile de remplacer le contrôle physique des électeurs en présentiel par des méthodes de vérifications d'identités. Aussi le système de vote peut être saturé par des multiples requêtes, et donc ne peut être plus en mesure de répondre en un temps raisonnable et finit pas se mettre en indisponibilité (Déni de service issu de point de défaillance unique). Lorsqu'une élection est organisée en ligne, le système de vote doit assurer que le droit à voter n'est utilisé qu'une seule fois. À cette fin, il se voit confier à la fois l'identité d'un électeur et son vote tout en respectant la confidentialité du vote, l'électeur a le droit d'exprimer un vote blanc.

Avant d'expliquer le vote électronique basé sur la blockchain , on va prendre une vue générale sur le sens de la technologie de blockchain et ses principales notions.

I.4 Technologie de blockchain :

L'architecture derrière la technologie de la Blockchain est apparue dès 1991 quand les chercheurs Stuart Haber et W. Scott Stornetta ont présenté une solution informatique qui permet l'horodatage des documents numériques qui ne soient jamais altérés. [6]

La blockchain est une technologie de stockage et de transmission d'informations. Elle possède en particulier quatre caractéristiques majeures : transparence, sécurité, immuabilité et la décentralisation c'est-à-dire qu'elle fonctionne sans organe central de contrôle.

- La blockchain est ***transparente***, car chaque utilisateur peut observer l'ensemble des échanges inscrits sur une blockchain depuis sa création.
- Elle est ***sécurisée*** car elle est basée sur l'utilisation de la cryptographie asymétrique de clé publique et privée pour les signatures numériques. Aussi, la validation des blocs nécessite la vérification de la validité d'un ensemble de transactions sans avoir appel à un intermédiaire de confiance ou une autorité centrale (Sans organe de contrôle, car la blockchain est basée sur des relations de Pair-à-Pair). Elle est sécurisée grâce à sa redondance sur l'ensemble des nœuds du réseau (pas de point de défaillance unique).
- Elle est ***immuable*** car toute tentative de fraude sera détectée par la majorité des utilisateurs rapidement grâce à son incohérence avec l'historique des transactions enregistrées dans la blockchain par conséquence la fraude serait rejetée.

En pratique, une blockchain est une base de données numérique infalsifiable sur laquelle sont inscrits tous les échanges relatifs à une propriété physique ou logique effectués entre ses utilisateurs depuis sa création. C'est parce que les échanges successifs y sont enregistrés sous forme de blocs de transactions chainés que l'on appelle une "blockchain". Notant que une fois les transactions sont insérées à la blockchain, elles ne sont plus ni modifiables ni supprimables. La **Figure I.2** montre le principe général d'une blockchain et les différentes étapes de déploiement d'une transaction.

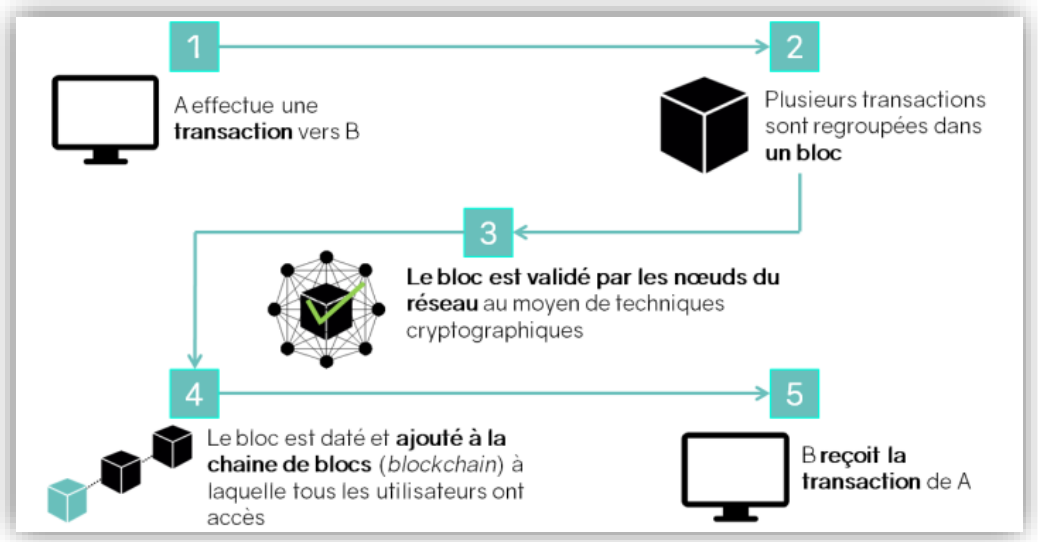


Figure I.2: Le fonctionnement de blockchain [7]

I.4.1 Structure de blockchain :

Chaque bloc consiste en un ensemble de données structurées, disposant d'une position dans la chaîne. Le block est composé d'une partie de données qui contient une liste des transactions, et d'un entête composé des métas données. Tous les éléments contenant dans un block se différencier d'une blockchain à une autre. La **Figure I.3** illustre le chainage des blocks et comment un block est structuré.

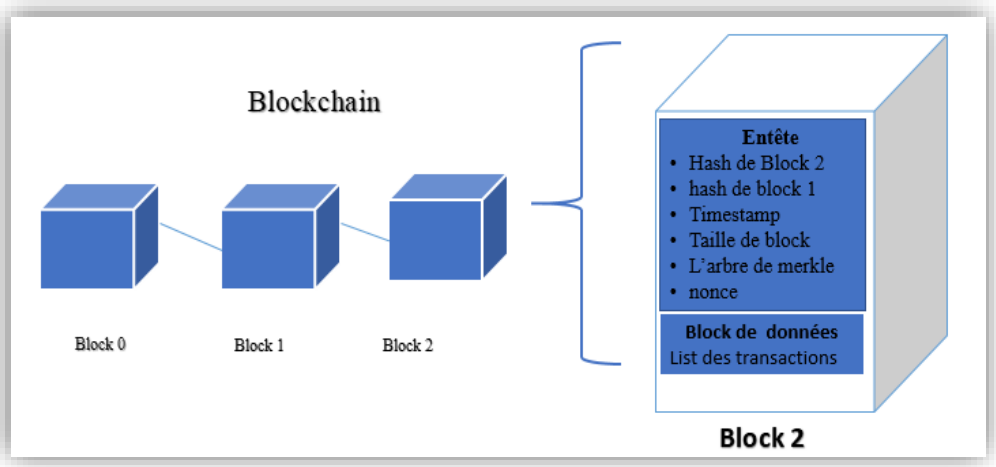


Figure I.3 blockchain et la structure d'un block

- **Hash de block :** tous les éléments d'un block sont transmis via un algorithme de hachage. Cela donne une sortie contenant une valeur de 256 bits et 64 caractères, qui est appelée le hachage du bloc.
- **Hash du bloc précédent :** cette adresse de hachage localise le bloc précédent, donc la modification de bloc ou l'injection d'un nouveau block est facilement détectée et elle est rejetée.
- **Timestamp:** le temps de création approximatif de ce bloc.
- **Taille d block** : La taille du bloc, en octets.
- **L'arbre de merkle** : est un arbre de hachage dans lequel chaque nœud feuille est identifié avec le hachage cryptographique d'un bloc de données, et chaque nœud non feuille est étiqueté avec le hachage cryptographique des étiquettes de ses nœuds enfants. La majorité des implémentations d'arbre de hachage sont binaires (chaque nœud a deux nœuds enfants), mais elles peuvent également avoir beaucoup plus de nœuds enfants. [8]
- **Liste des transactions :** toutes les transactions qui doivent avoir lieu. Le contenu de la transaction dépend de but et de l'implémentation de la blockchain, mais en général il contient

l'adresse de l'expéditeur, la clé publique de l'expéditeur, signature digital et block d'entrée et sortie. La **Figure I.4** montre la structure générale d'une transaction.

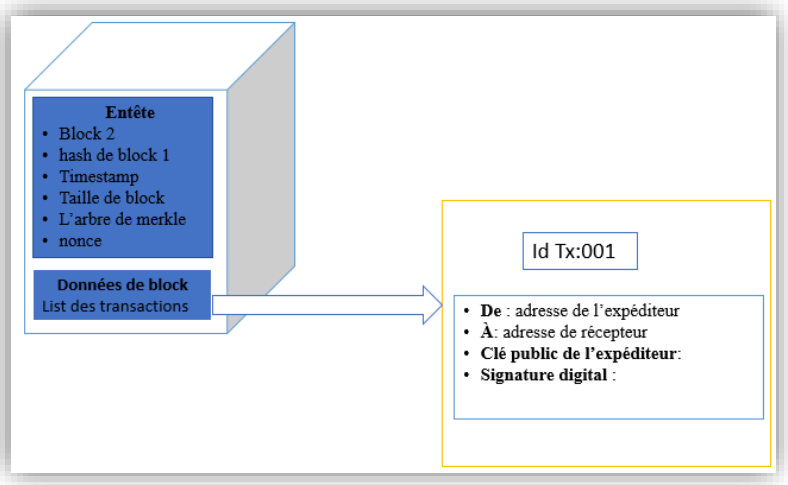


Figure I.4 structure générale d'une transaction.

Les trois principales méthodes actuelles de validation au sein d'une chaîne de blocs sont :

- **La preuve de travail PoW (proof-of-work) :** l'ajout d'un nouveau block se fait par le premier utilisateur qui effectue des calculs mathématiques pour trouver le hach d'un block qui est inférieur à la valeur de seuil de nonce (résolution du défi). [9]
- **La preuve d'enjeu (proof-of-stake) :** le PoS ne repose pas sur des calculs intensifs, au lieu de ça, l'utilisateur avec le solde le plus élevé a la chance d'ajouter le nouveau block et l'utilisateur qu'a le moins solde est le plus susceptible d'abuser la blockchain. [9]
- **La preuve d'autorité (proof-of-authority) :** le PoA c'est un autre modèle utilisé dans les blockchains. Seuls les nœuds autorisés (validateurs) de la blockchain vont créer et valider les blocs. [10]

Chacune des méthodes citées ci-dessus requièrent un consensus, plus ou moins élaboré en fonction des blockchains pour sélectionner qui va valider un bloc.

I.4.2 Types de blockchain :

Il existe trois types basiques différents de blockchain, ils sont les suivants :

a. Blockchain Privée :

Les blockchains privées fonctionnent sur des réseaux privés et sont beaucoup plus utilisées dans les entreprises et organisations privées. L'opérateur réseau peut configurer les autorisations et les

rôles des utilisateurs et des nœuds : qui participe au processus de consensus, qui est capable de lire et d'écrire, et comment les nœuds de la blockchain sont alloués sur le réseau. [11]

b. Blockchain publique :

Si l'on souhaite créer une blockchain complètement ouverte, similaire à Bitcoin, qui permet à tout le monde de rejoindre et de contribuer au réseau, on peut choisir une blockchain publique. Dans une blockchain publique, tous les membres peuvent lire, écrire et auditer les activités en cours sur le réseau de blockchain, ce qui aide à conserver sa nature autonome. Un algorithme de consensus est utilisé pour vérifier l'authenticité des informations, la preuve d'enjeu (PoS) et la preuve de travail (PoW) sont deux méthodes de consensus fréquemment utilisées. [11]

c. Blockchain du consortium :

Une blockchain de consortium est une forme hybride de blockchains publiques et privées. Au lieu d'un système ouvert dans lequel n'importe qui peut valider des blocs ou d'un système fermé dans lequel une seule partie sélectionne les validateurs de blocs, une chaîne de consortium emploie un petit nombre de parties tout aussi puissantes comme validateurs. Par rapport à un réseau blockchain public, une blockchain de consortium est plus sécurisée, évolutive et efficace. Elle a également des contrôles d'accès, tout comme la blockchain privée. Cependant, une blockchain de consortium est moins transparente. [11]

Les types précédemment expliqués sont les principaux existants, mais comme les blockchains sont en évolutions continue, aujourd'hui on trouve d'autres concepts avancés dans la littérature comme la blockchain des blockchains et l'internet de blockchains … etc.

I.4.3 Contrats intelligents :

Certaines blockchains utilisent des contrats intelligents, la blockchain avec contrat intelligent est l'évolution des blockchains originales apparues avec la version initiale de Bitcoin. Un contrat intelligent ou un smart contract est un code qui s'exécute automatiquement sur la blockchain dès que les conditions spécifiées (triggers) sont remplies. Ainsi, les Smart contracts promettent un faible frais de transaction par rapport aux systèmes traditionnels qui nécessitent un tiers de confiance pour appliquer et exécuter les termes d'un accord. Le Smart contract peut être considéré comme un système qui transmet des actifs numériques à toutes ou à certaines parties impliquées une fois que des règles prédéfinies ont été respectées. [12]

I.4.4 Les Blockchains principales :

Aujourd'hui il existe plusieurs blockchains, chacune a plusieurs caractéristiques. Nous présentons ci-dessous les blockchains le plus populaires :

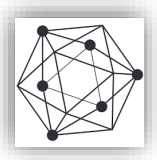
Blockchains		Descriptions
	Bitcoin	▪ Créé par Satoshi Nakamoto en 2008 Bitcoin. [13] ▪ Elle repose sur la preuve de travail PoW des mineurs. [14] (récemment le PoS est utilisé) ▪ Bitcoin c'est un type de crypto-monnaie, il utilise la cryptographie pour la sécurité.
	Ethereum	▪ Créé par Vitalik Buterin en 2015, c'est une blockchain similaire à Bitcoin. [15] ▪ Système basé sur la preuve de travail PoW [16] ▪ Il peut être utilisé pour créer des contracts automatisés ou faire circuler une monnaie numérique appelée Ether.
	HyperLedger	▪ Hyperledger est une plateforme open source de développement de Blockchain qui offre un environnement, des normes, des directives et les outils nécessaires pour créer des blockchains open source. [17] ▪ Développé en décembre 2015 par Linux. [18] ▪ Utilise le consensus PBFT [19]

Tableau I.1 blockchains principales

I.5 Principe de Vote basé sur la Blockchain :

La technologie Blockchain utilise des mécanismes performants. Ce qui permet à un vote d'être clair et infalsifiable, avec un haut niveau de protection des données et exactitude des résultats. Se baser sur la technologie de la blockchain dans les systèmes de vote électroniques est très important pour surmonter les risques et le vote électronique. Dans la **Figure I.5** on peut voir la différence principale entre les deux systèmes : les systèmes de vote traditionnels utilisent un seul serveur central pour le vote. La modification et la falsification de vote sont faciles à faire, puisque personne ne sait comment vérifier le vote. Dans le system de vote basé sur les blockchains, on note l'absence de l'autorité centrale, les données sont stockées d'une façon décentralisée dans plusieurs nœuds, il est impossible de pirater tous les nœuds et modifier les données. Ainsi, de cette façon, on ne peut pas détruire les votes et vérifier efficacement les votes par pointage avec d'autres nœuds.

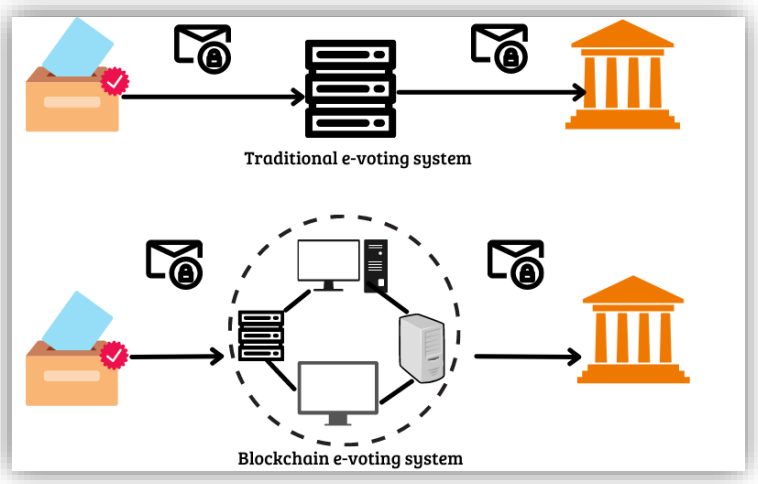


Figure I.5: vote traditionnel vs vote basé sur blockchain

L'association du vote électronique avec la blockchain est une idée souvent avancée. Il s'agit de concevoir des systèmes de votes en ligne sécurisés pour assurer un résultat transparent et auditable par tous les membres ou encore de donner la possibilité à chaque votant de vérifier son vote. Le système vote basé sur la blockchain se repose sur les Smart Contracts, qui permet de spécifier les règles utilisées pour assurer la transparence dans le processus de vote. Il offre également la possibilité de vérification à tout moment l'état et le résultat de la transaction de vote. La blockchain assure l'immuabilité du vote confidentiel pour chaque électeur et l'unicité de vote par électeur ainsi elle garantit la traçabilité : toutes les transactions sont traçables à chaque étape.

Dans un système de vote blockchain, l'administrateur initialise le vote, il écrit les informations sur le vote dans un contrat intelligent et les publie sur la blockchain avant le déclenchement de vote.

Ensuite, les électeurs peuvent se renseigner sur les informations de vote dans le contrat intelligent. La

Figure I.*6* montre le principe général de system de vote basé sur les blokchains.

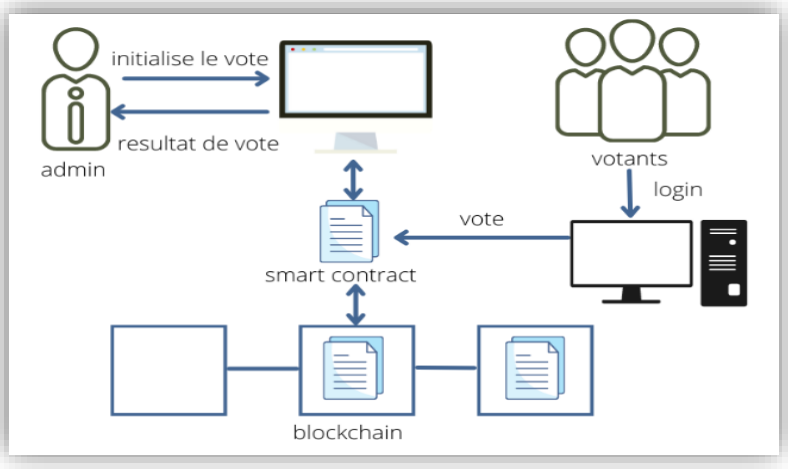


Figure I.6 : système de vote basé sur les blockchains

La transparence directe assure la confiance dans les résultats de vote, les électeurs sont capables à constater les états des élections et à témoigner l'avancement de ce processus. Sachant que le droit de vérifiabilité contredit le principe de l'anonymat, c'est-à-dire la rupture de tout lien entre une expression de vote et un électeur. Donc une solution d'e-vote basée sur la blockchain doit assurer à la fois la vérifiabilité, la transparence, l'immutabilité et l'anonymat pour être acceptés.

I.6 Travaux connexes :

Le réseau de vote ouvert (OVN) a été présenté par F. Hao ,S. Shahandashti et P. Mccory [20] qui est le premier déploiement d'un protocole de vote par Internet transparent et auto-comptable avec une confidentialité totale de l'utilisateur en utilisant Ethereum. Open Vote Network garantit la confidentialité des votes puisque les votes sont cryptés avant d'être exprimés. Il support seulement les élections avec deux options (oui ou non) et au maximum 50 votants en raison de les outils mathématiques qu'ils utilisaient. L'OVN est incapable d'empêcher les mineurs frauduleux de corrompre le système. Un électeur frauduleux peut également contourner le processus de vote en envoyant un vote invalide. Le protocole ne fait rien pour garantir la résistance à la violence, et l'administrateur électoral ne veut confiance. [21]

J.L. Wu al. [22] Ont suggéré un système de vote électronique décentralisé anonyme et transparent qui nécessitant un degré minimal de confiance entre les participants. Ils pensent que pour les élections électroniques à grande échelle, la méthode de vote actuelle DATE est appropriée. La vie privée de chaque électeur est protégée par un mécanisme efficace de ring signature, ainsi dans ce system ce qui garantit que toute personne pouvant accéder au réseau blockchain est en mesure de

comptabiliser le résultat par elle-même, c'est-à-dire qu'aucun tiers n'est requis après la phase de vote. Ce système ne convient qu'aux petites échelles en raison de la limitation de la plateforme. Aussi, il utilise le consensus PoW, qui présente des inconvénients importants tels que la consommation d'énergie. Malheureusement, leur système proposé n'est pas assez puissant pour se protéger des attaques DoS parce qu'il n'y avait pas d'autorité tierce sur le régime responsable de l'audit du voter après le processus électoral.

B.Shahzad et J.Crowcroft [23] ont proposé la preuve d'exhaustivité du BSJC comme méthode de vote électronique fiable. Ils ont utilisé un modèle de processus pour décrire la structure de l'ensemble du système. À plus petite échelle, il a également tenté de résoudre les problèmes d'anonymat, de confidentialité et de sécurité lors des élections. Pour cela ils utilisent l'algorithme de SHA256 qui hache toutes les données de block , et un autre numéro aléatoire généré par le système est ajouté dans le hachage pour le rendre plus sécurisé. Cependant, il existe un problème qui est la participation d'un tiers car il existe un risque important de falsification des données, de fuite et de résultats tabulés injustes, tous qui peuvent avoir un impact sur la vérification de bout en bout

H. Yi [24] a présenté le système de vote électronique basé sur la blockchain (BES) qui offrait des méthodes pour améliorer la sécurité du vote électronique leur BES utilise le registre distribué (DLT) pour éviter la falsification des votes. Dans cette technique, les agressions de contre-mesure constituent un enjeu important. Cette méthode nécessite l'intervention de tiers responsables et n'est pas bien adaptée à une utilisation centralisée dans un système comportant de nombreux agents dans cette situation, les dépenses informatiques sont plus importantes et peut-être prohibitives si le calcul la fonction est complexe et il y a trop de participants

Tableau 2 comparaison entre les systèmes de e-vote basé sur la blockchain

	Type de blockchain	Principe de fonctionnement	Consensus	Vulnérabilité détectée	Avantages	Inconvénients	Confidentialité de vote	Anonymat du votant	Scalabilité
OVN [20] [21]	Publique autorisée	- un administrateur ajoute les votants éligibles à la liste blanche -vote à deux phases, 1ère phase : tous les électeurs enregistrent leur intention de voter par diffusion de clé de vote, 2éme phase : les électeurs votent. - protocole de vote avec auto-comptage (self-tallying) -Preuve de connaissance (preuve de défi zéro) -Clés éphémères	POW	DOS	Collision complète de tous les électeurs Le décompte échoue si les électeurs refusent de voter, ce qui peut entraîner des retards Le processus d'expulsion de l'attaquant n'est pas expliqué	- Le vote est crypté avant qu'il soit envoyé - transparent - Aucune autorité de confiance -Indifférence Efficace	**Confidentialité de vote est assurée**	Oui	**Scalabilité n' est pas assuré**
DATE [22]	Publique	Ring signature, adresse fultive, clé image et self-tallying	POW	-La collision des clés du gestionnaire provoque le décompte des votes anticipés	-Transparent -Aucune autorité de confiance -Consommation minimale -Consommation de gaz minimale	- Pour les systèmes avec une taille limitée - cout - consommation d'énergie -Nécessite une grande puissance de calcul -Difficulté à gérer les entités de plusieurs signataires Pas d'audit		Oui	
BSJC [23]	Privée	- une solution basée sur les machines à voter électroniques et l'authentification biométrique des électeurs avant de pouvoir voter - crée d'abord les blocs puis les scelle	Proof of complete-ness	-falsification des données de fuite et de résultats tabulés injustes. -Vulnérable à l'attaquant interne	-Aucun accès non autorisé ne peut être effectué de l'extérieur -Audit Vérifiable	Pour les systèmes avec une taille limitée -consommation d'énergie -fondé sur l'autorité -Pas de cryptage -Non conscient de la confidentialité -Exiger une présence physique dans les bureaux de vote -Le processus de scellement cause des retards		Non	
BES [24]	Publique	-Identifiant utilisateur basé sur ECC (Cryptographie à courbe elliptique) - Calcul de hash base sur SHA-256 - Utilise l'algorithme de signature numérique à courbe elliptique (ECDSA)	PoW	-Les agressions de contre-mesure quantique -Tolérer l'injection de données Attaque Sybil et problème de double dépense Retrait de vote non expliqué	-Non-répudiation -permet aux électeurs de modifier leur vote avant une date limite prédéfinie	-nécessite les dépenses informatiques.		Oui	

I.7 Conclusion :

Dans ce chapitre, nous avons présenté et analysé les différentes techniques de vote. En premier lieu, nous avons expliqué le processus de vote traditionnel. Ensuite, nous avons défini les techniques de vote électronique en déterminant ses différentes formes. Finalement, nous avons détaillé la technologie de blockchain avant d'expliquer les systèmes de vote électronique basé sur les blockchains et ses travaux connexes avec leurs comparaisons. Ce qui nous a offert une vue sur l'efficacité et l'imparfait de chacune. Dans le prochain chapitre, nous allons présenter et expliquer notre solution qui est un système de vote distribué, sécurisé, transparent et qui assure la confidentialité.

CHAPITRE II :

Système de vote distribué préservant la confidentialité

Chapitre II : Système de vote distribué sensible à la confidentialité

II.1 Introduction :

La technologie Blockchain offre la possibilité de concevoir de nouveaux types d'applications décentralisées qui permettent à leurs utilisateurs de stocker des données de manière sécurisée et transparente. Comme vue dans le chapitre précédent, un ensemble de systèmes de vote électronique basés sur la technologie Blockchain ont été proposés, mais qui n'ont pas été utilisés réellement, car ils ne supportent pas certaines propriétés de sécurité. Notant qu'il est important de répondre aux besoins des citoyens que ce soit des parties politiques ou des électeurs qui veulent avoir la possibilité de vérifier que tous les votes éligibles sont collectés et comptés sans avoir été altérés ou supprimés, en assurant le secret des votes et l'anonymat des électeurs.

Dans ce chapitre, nous allons présenter notre solution qui est une application décentralisée de vote qui s'appuie sur la technologie de la blockchain en combinant des techniques qui nous ont permis de garantir plusieurs mesures de sécurité. D'abord, nous présentons le design de notre système suivi par une explication globale de son fonctionnement et les différentes techniques utilisées pour assurer sa sécurité. Puis, en détaillerons les différentes phases traitées par les acteurs de cette application avec quelques diagrammes. Enfin, on terminera avec une conclusion.

II.2 Design de system :

Certains systèmes existants de vote électronique basés sur les blockchains souffrent de beaucoup de problèmes de sécurité, notamment à cause de l'absence de l'anonymat, l'électeur subit des problèmes de coercition où il est pressé de la part de quelqu'un pour voter pour lui. En plus les problèmes de fraude sont présents dans des systèmes de vote électroniques qui ne les empêchent pas et ne détectent pas l'invalidité de votes. L'objectif de notre solution est d'assurer la transparence, l'unicité de vote, la confidentialité, l'anonymat et la sécurité.

Le Framework que nous proposons (voir **Figure II.7**) est une variante publique et autorisée du blockchain Ethereum gérée par une autorité. Cette blockchain enregistre les votants anonymes avec leurs votes.

Le backend de blockchain contient toutes les informations personnelles des votants autorisés (nom, prénom, wilaya, commune, numéro national) qui sont enregistrés par les pairs (peers) représentant l'autorité d'état, ces informations ne doivent pas être apparues dans la blockchain. Notant que les termes validateurs, administrateur ou pairs autorisés seront utilisés de manière interchangeable dans le reste de ce document et signifient la même chose. L'autorité d'état est le point de confiance

des citoyens qui joue un rôle très important dans le system de vote. D'abord l'autorité garantie la démocratie de system de vote car un système démocratique est défini seulement si les électeurs éligibles peuvent voter, donc elle détermine qui est éligible pour voter avant que le vote ne soit a débuté, ainsi elle assure la confidentialité des informations de chaque votant, une autre fonction est assurée par l'autorité : personne ne devrait pouvoir dupliquer le scrutin.

Ce sont eux qui fixent les règles de scrutin par exemple : ils font à jour la liste des votants révoqués dans le cas où un votant meurt sa clé publique sera supprimé, personne ne peut l'utiliser après.

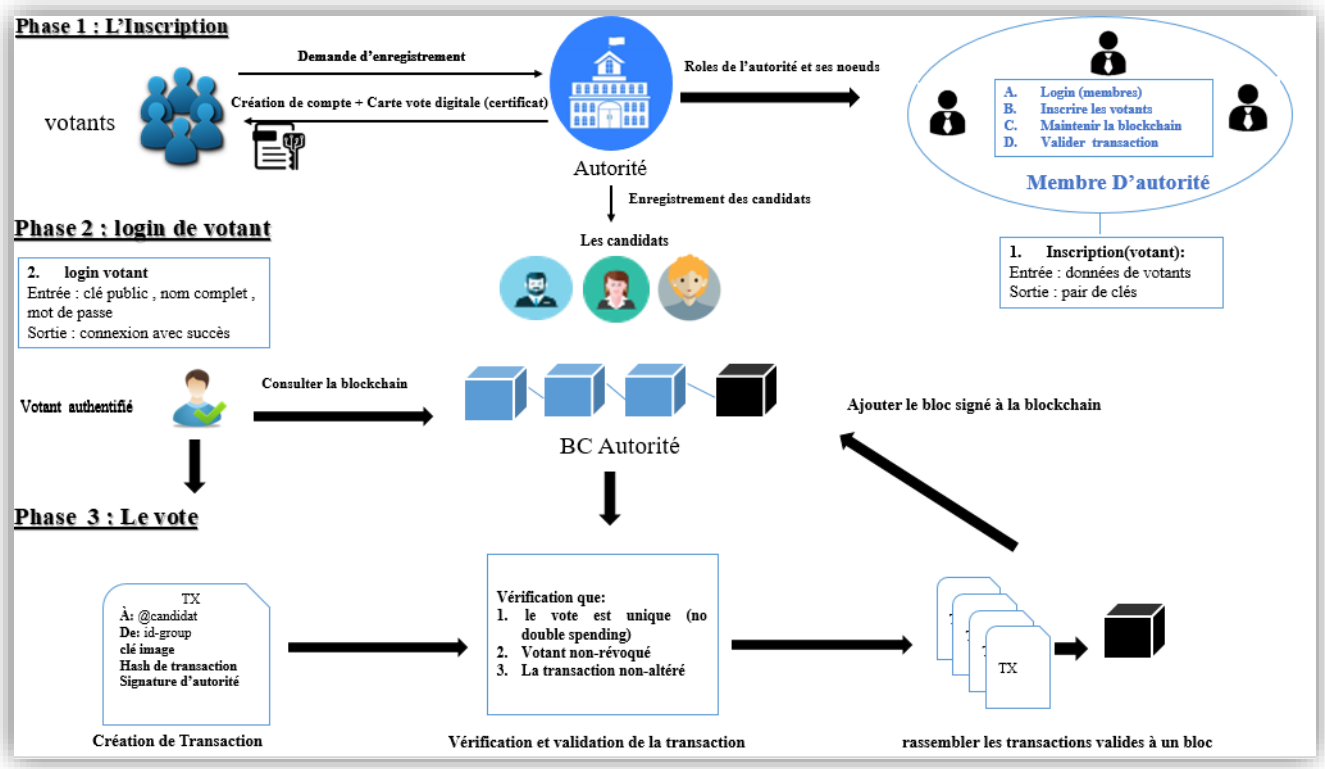


Figure II.7 design de système

Pour bien assurer et faciliter les fonctionnalités d'autorité, nous mettons en œuvre un site web qui assure les différentes taches : inscription des votants, l'enregistrement des candidats, maintenance de blockchain, validation des transactions, et pour qu'un membre d'autorité peut faire les différentes taches, il doit passer par un login.

Afin de s'inscrire et obtenir des paramètres d'authentification, chaque personne ayant le droit de voter et souhaite participer à l'élection, se déplace vers un bureau de vote. Pour vérifier son éligibilité, chaque électeur doit fournir ses preuves d'identité à l'autorité d'enregistrement (AE), qui vérifie si

l'électeur est éligible pour participer à l'élection. Ainsi que le candidat doit aussi fournir au AE ses papiers spéciales et nécessaires pour poser sa candidature.

Chaque votant inscrit préalablement par l'état a le droit de connecter au notre site web en ligne I-VOTE, consulter la blockchain et voter. Contrairement à l'existant, notre système assure l'anonymat, pour cela nous avons utilisé un mécanisme qui s'appelle « cacher dans la foule» , il consiste d'affecter à chaque groupe de votants un seul identifiant "identifiant de group ", quand un électeur vote, une transaction est créée. Elle contient l'identifiant de groupe et son choix, cet identifiant assure qu'il n'y a aucune liaison entre le votant et son vote : donc l'anonymat est assuré.

Au contraire de quelques systèmes cités dans le chapitre précèdent. Le système que nous avons proposé utilise le modèle de consensus et de validation appelé preuve d'autorité (PoA) où chaque block est validé par l'autorité avant qu'il inséré à la blockchain. Ce qui garantit une sécurité très efficace contre des actions malveillantes et empêche les électeurs d'envoyés des transactions invalides. Les blockchains PoA sont sécurisées par les nœuds de validation qui sont sélectionnés en tant qu'entités de confiance.

Sachant que s'assurer que l'anonymat est protégé dans les systèmes de vote basé sur blockchain souvent provoque un autre problème inévitable qui est la possibilité qu'un électeur envoi multiples votes pour un seul candidat. Notre système résout ce problème en garantissant l'unicité de vote, i.e. chaque votant à le droit pour voter une seule fois. Quand un vote est effectué une clé unique va être effectuée à chaque électeur, en utilisant la technique « L'image de la clé privée ».

Les images clés sont inventées par Monero qui est une cryptomonnaie open source, ils sont utilisés comme clé sécurisée dans les transactions pour empêcher les utilisateurs de dépenser deux fois les mêmes pièces Monero. [25]

Dans notre cas cette technique, permet d'empêcher les électeurs de voter plusieurs fois, pour cela il y a une image clé unique pour chaque transaction effectuée par l'électeur et enregistrée dans la blockchain. Si un votant tente de voter encore une fois, les nœuds validateurs peuvent rapidement déterminer si les images clés associées ont déjà été utilisées auparavant. Et cela peut empêcher la deuxième tentative de vote de se dérouler avec succès.

Le vote doit être envoyé par l'adresse source qui est l'adresse de groupe vers l'adresse destination qui signifie à l'adresse de candidat choisit ou une adresse spéciale en cas où le votant n'a choisi aucun des candidats (vote blanc). L'autorité avec son rôle va vérifier l'unicité de la clé générée

par « image de clé » et assure que l'adresse source et l'adresse destination sont correcte, cela fait selon un contrat intelligent qui définit des règles pour assurer la sécurité totale de ce vote et finalement publier la transaction validée.

La transaction validée contient l'adresse de groupe attribué au votant, l'adresse de candidat choisi ou l'adresse définie pour le vote blanc, le hash de transaction, l'image de clé et la signature de l'autorité.

II.3 Les phases de système :

Initialement, chaque session d'élection se commence avec l'enregistrement des candidats, ce processus est fait par l'autorité après la vérification de certaines conditions démocratiques, elle est suivie par l'enregistrement des électeurs et la définition des périodes d'enregistrement, votes, etc. chacune de ces phases est expliquée dans la suite :

II.3.1 Login d'administrateur :

Chaque membre d'autorité doit d'abord s'authentifier, pour bien protéger le compte d'administrateur contre des vulnérabilités notre système garanti une authentification sécurisé à trois niveaux (login avec clé publique, avec mot de passe et l'empreinte), puis il se connecte avec un compte MetaMask qui est un portefeuille utilisé comme extension d'un navigateur web pour simplifie l'interaction avec les applications décentralisées.

II.3.2 L'inscription des candidats :

Initialement l'élection se commence avec l'attribution des candidats, ce processus est fait par l'autorité après la vérification de certaines conditions démocratiques. L'inscription des candidats se fait dans une période définie, une attestation d'acceptation sera délivrée pour chaque candidat lors la validation des conditions de candidature, chaque candidat qui dispose d'une attestation d'inscription à l'élection agréée par l'état, sera inscrit par l'autorité avec ses informations dans l'interface de vote de notre application et les votants peuvent le voir dans la liste des choix. (voir la **Figure II.8**).

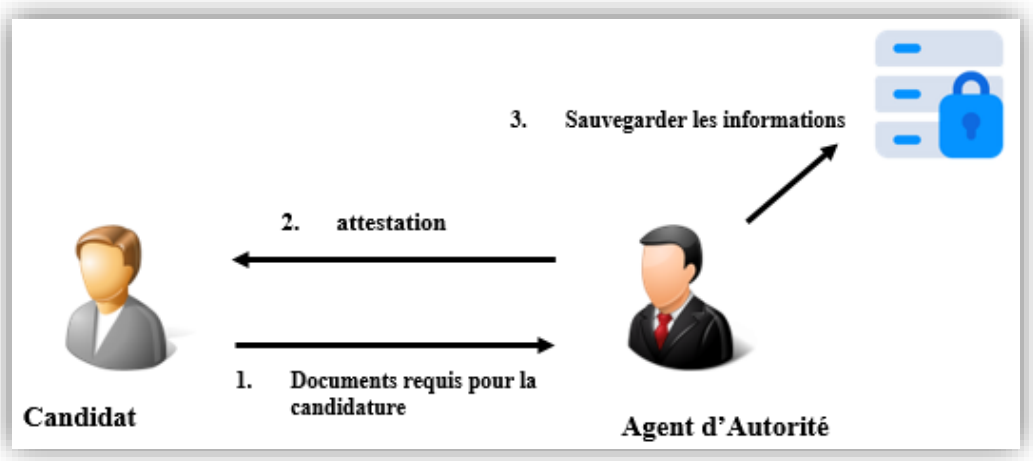


Figure II.8 phase d'inscription de candidat

II.3.3 L'inscription de votant :

C'est la première phase que l'électeur doit passer par, elle est faite et garantie par l'autorité avant le processus de vote avec une date précise et limitée. Afin d'éviter qu'un citoyen obtienne plusieurs identifiants, et pour s'assurer que la demande provient d'un utilisateur légitime, le system nécessite une autorité de confiance. L'inscription initiale du citoyen se fait avec la carte d'identité.

La présence physique de votant avec sa carte d'identité confirme sa légitimité qui lui permet au plus tard d'authentifier pour exprimer son choix.

Le votant doit y aller à l'autorité qui vérifie sa l'éligibilité en lui demandant une carte d'identité. Puis elle l'enregistre sur sa base de données décentralisée des votants qui ont le droit de voter, le système va donner au votant un identifiant de groupe, c'est-à-dire chaque groupe de votants a un identifiant unique permettant d'appliquer la technique de « se cacher dans la foule », ainsi en le répondant avec une carte électronique qui contient la paire de clés (clé privée et clé publique) qui sont signées par la clé privée de l'autorité. Il est nécessaire pour le votant de sauvegarder ses clés d'une maniéré sécurisée. Après la période définie pour l'inscription, le votant inscrit peut connecter à la plateforme de vote (voir la **Figure II.9**).

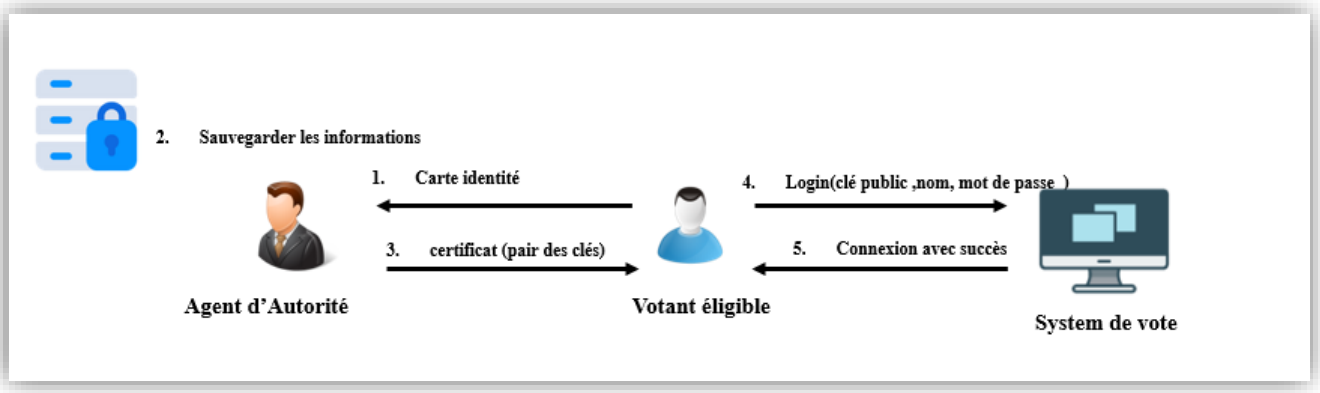


Figure II.9 phase d'inscription, login de votant

II.3.4 Login de votant :

La deuxième phase (voir la **Figure II.9**) c'est la connexion de votant, après son inscription réussite, le votant peut consulter notre site pendant la période précise pour le vote depuis son smartphone ou son pc, une interface va s'afficher pour la connexion. L'accès nécessite des paramètres d'authentification pour chaque électeur.

En premier lieu, il doit entrer sa clé publique, si la condition est satisfaite, une interface apparait dont il doit rentrer son nom complet et son mot de passe, si les paramètres entrés sont valides le votant peut accéder à l'interface de vote. Seuls les électeurs enregistrés peuvent se connecter.

II.3.5 Voter :

C'est la dernière phase qui permet de fournir au votant la possibilité d'exprimer son choix de vote, il peut choisir un des candidats ou personne (Vote blanc). Une fois le vote est affecté, il ne peut ni n'être changé, ni n'être supprimé. Une transaction de vote est créée et vérifiée par les membres d'autorité, qui jouent le rôle de modérateurs du système. A cet égard, le consensus PoA offre des performances élevées et une tolérance aux pannes. Dans PoA, les droits de génération de nouveaux blocs sont attribués aux nœuds qui ont prouvé leur autorité pour le faire. Dans notre schéma, une ou plusieurs machines de validation représentant les nœuds d'autorité d'état sont chargées de générer chaque nouveau bloc de transactions qui sera inclus dans la Blockchain.

Par rapport au consensus PoW, le consensus PoA n'exige pas que les nœuds dépensent des ressources de calcul pour résoudre des tâches mathématiques complexes ni que les nœuds possèdent des droits équitables pour valider les blocs. Nous avons choisi ce modèle de consensus parce que la nature de notre application interdit l'enregistrement des faux votant, et de faux candidats. Aussi, elle doit assurer la transparence, la licité et l'éthique de vote. Enfin, comme le PoW utilise beaucoup de

ressource, il consomme beaucoup d'énergie ce qu'est mauvais pour l'environnement, pour cela nous avons choisi le PoA pour économiser la consommation d'énergie et les couts d'élaboration.

À la création de transaction l'autorité va vérifier l'existence et l'exactitude d'identifiant de group associer au votant. Cet identifiant est une combinaison de numéro de commune et wilaya d'électeur, par exemple : pour Laghouat ce serait 0303, de point de vue de sécurité c'est risqué car s'il assure la confidentialité des utilisateurs individuels (cacher dans une foule), il met en péril la sécurité de group en raison de sa signification sémantique qui le lie aux citoyens de la ville. Ainsi, le candidat gagnant pourrait être influencer à l'avenir, il va favoriser les citoyens de la ville qui lui ont le plus voté. Donc pour résoudre ce problème nous devons rompre le lien sémantique, pour cela nous avons utilisé une fonction de hachage qui va hacher la valeur de combinaison comme l' **Équation II.1** : ce qui implique que les identifiants de group générés ne sont pas enregistrés tels quels dans la blockchain et l'anonymat est assuré.

$$IDgroup = Hash\ (\ code\ wilaya + code\ commune\ )$$

Équation II.2 : calcul d'identifiant de groupe

En plus et comme nous avons déjà dit, en utilisant la technique de « clé image » un identifiant unique est généré pour chaque votant utilisant l'**Équation II.3** les membres d'autorité vont vérifier que cet identifiant n'a était jamais utilisé auparavant. S'ils trouvent qu'elle a été utilisée, le vote multiple est détecté et la transaction ne sera pas validée. Sinon, le vote va s'envoyer vers l'adresse de candidat choisi par le votant après que les membres d'autorité vérifient que l'adresse de candidat et l'adresse publique de group sont correctes et la clé image est unique, un d'eux signe toute la transaction pour la valider.

$$clé\ image = clé\ privé * hach(clé\ publique)\ [26]$$

Équation II.4 calcul clé image

Ainsi, chaque votant, souhaitant vérifier que son dernier vote valide a été inclus dans le résultat final, peut accéder à la Blockchain de l'élection et vérifier l'existence de valeur de vote, qu'il a sauvegardée pendant la phase de vote, dans les blocs de la blockchain.

II.4 Conception :

Dans cette partie, nous allons définir l'analyse des besoins. Nous allons aussi présenter la conception et la modélisation du système de vote en ligne en se basant sur la technologie de

blockchains à travers le diagramme de cas d'utilisation, diagramme de classe et nous allons décrire le comportement de notre application à travers les diagrammes de séquence.

II.4.1 Spécification des besoins :

Cette phase est très importante pour mieux comprendre le système et définir les besoins fonctionnels et non fonctionnels. Ensuite, on établit le diagramme des cas d'utilisation qui donne une vision globale du comportement fonctionnel de la solution.

a. Les besoins fonctionnels :

Pour les besoins fonctionnels. Le système doit offrir les fonctionnalités de vote que nous avons mentionné précédemment dans les différentes phases.

b. Les besoins non fonctionnels :

A part les besoins fondamentaux, notre système doit répondre aux critères suivants :

- ***Traçabilité et intégrité*** : Les données sont enregistrées chez l'état dans le temps et sceller dans un registre décentralise et infalsifiable. Les données sont ainsi certifiées et non répudiables.
- ***Sécurité*** : La blockchain assure un stockage des informations d'une manière non modifiable et toutes ces informations pourraient se retrouver de façons chronologiques dans un registre sécurisé et aisément consultable.
- ***Performances*** : Un logiciel doit être avant tout performant c'est -à-dire à travers ces fonctionnalités, répond aux exigences des utilisateurs d'une manière optimale.
- ***Utilisabilité :*** Le système doit offrir à l'utilisateur une interface simple et facile à utiliser.
- ***Vote privé :*** après le vote est affecté, personne ne peut lier l'identité de votant avec son vote c'est-à-dire assurer l'anonymat.
- ***Démocratie :*** un system démocratique est seulement pour les votants légitimes et un seul vote doit être affecté par chaque votant enregistré, et aucun votant ne peut dupliquer le vote.

II.4.2 Diagramme de cas d'utilisation :

D'abord comme la **Figure II.10** montre notre système a 2 acteurs qui sont le votant et l'administrateur qui appartient à l'autorité. Le système demande aux acteurs de s'authentifier pour se connecter. Si les informations d'identification envoyées par l'acteur sont valides, il peut connecter et continuer l'utilisation de notre système. Sinon, le système lui informe que les données saisies sont erronées et le scénario d'authentification se répète. L'administrateur authentifié peut ajouter les électeurs et les candidats dans une date défini avant la période des élections. En remplissant les champs de l'interface affichés par les informations requises pour chacun. Il peut aussi fixer les dates de début et de fin de vote et même lancer l'élection.

L'ajout de l'électeur par l'administrateur lui permet de récupérer son compte de vote et de MetaMask. Le système invite l'acteur d'entrer la clé public, le nom complet et le mot de passe et se connecter à son compte MetaMask. Une fois authentifié, le votant choisit un candidat et valide son choix en émettant une transaction. La transaction sera rejetée par le système (invalide) si l'électeur a déjà voté, le vote n'est pas encore lancé, l'identifiant de group est invalide ou le key image existe déjà dans la blockchain (cas de multiple vote). Dans le cas contraire, la transaction sera validée et signée par le validateur. Elle sera insérée à la blockchain et le vote sera comptabilisé. Le système actualise et affiche les résultats en temps réel dès la validation des transactions. Les nœuds des réseaux blockchains (administrateur, validateurs, candidat et électeur) peuvent consulter les résultats qui sont publiques et suivre le vote d'après la blockchain de vote. (notant que la consultation de résultats et des candidats se fait sans login)

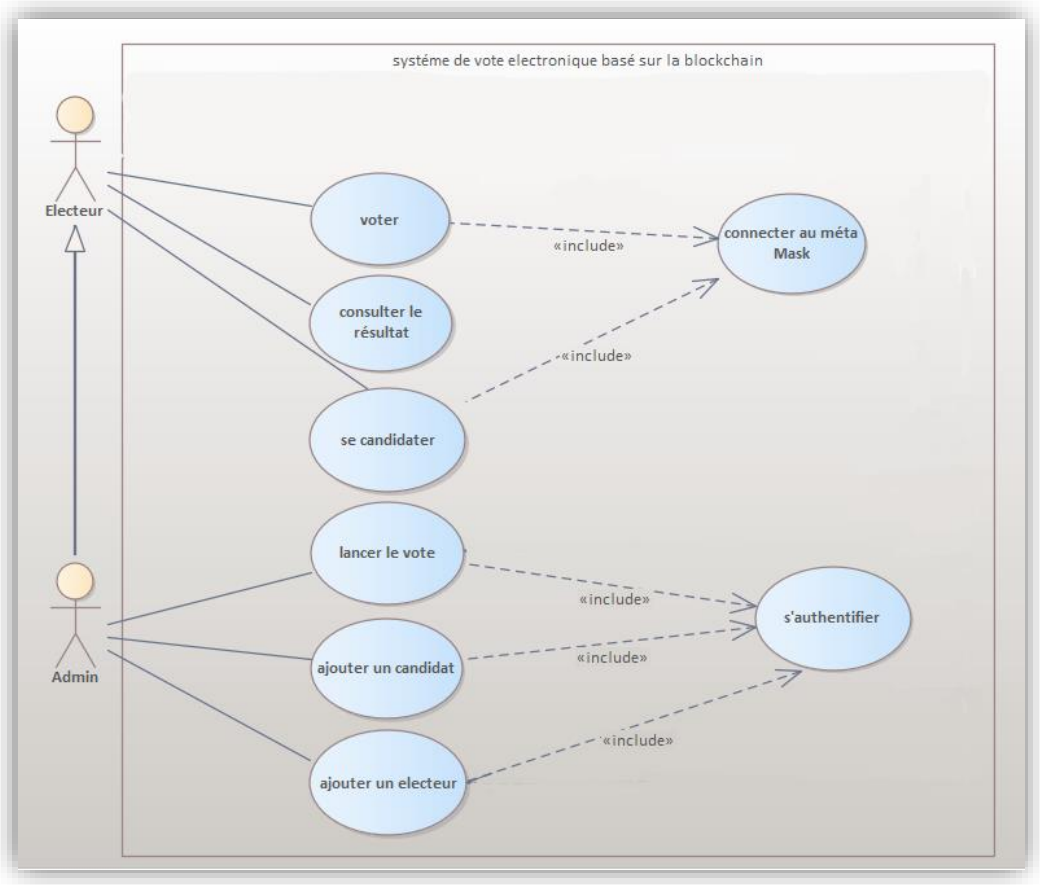


Figure II.10 diagramme de cas d'utilisation de système de vote

II.4.3 Diagramme de class :

Le système constitue de plusieurs classes (voir la **Figure II.11**): personne qui regroupe les attributs et les fonctions communs entre les classes administrateur et candidat et électeur. C'est-à-dire l'administrateur et l'électeur et le candidat sont tous personne qui peut voter et consulter le résultat.

L'administrateur a des fonctionnalités spécifiées qui permet d'ajouter les électeurs et les candidats et lancer le vote. La cardinalité entre l'électeur et le candidat montre qu'il peut voter pour un seul candidat.

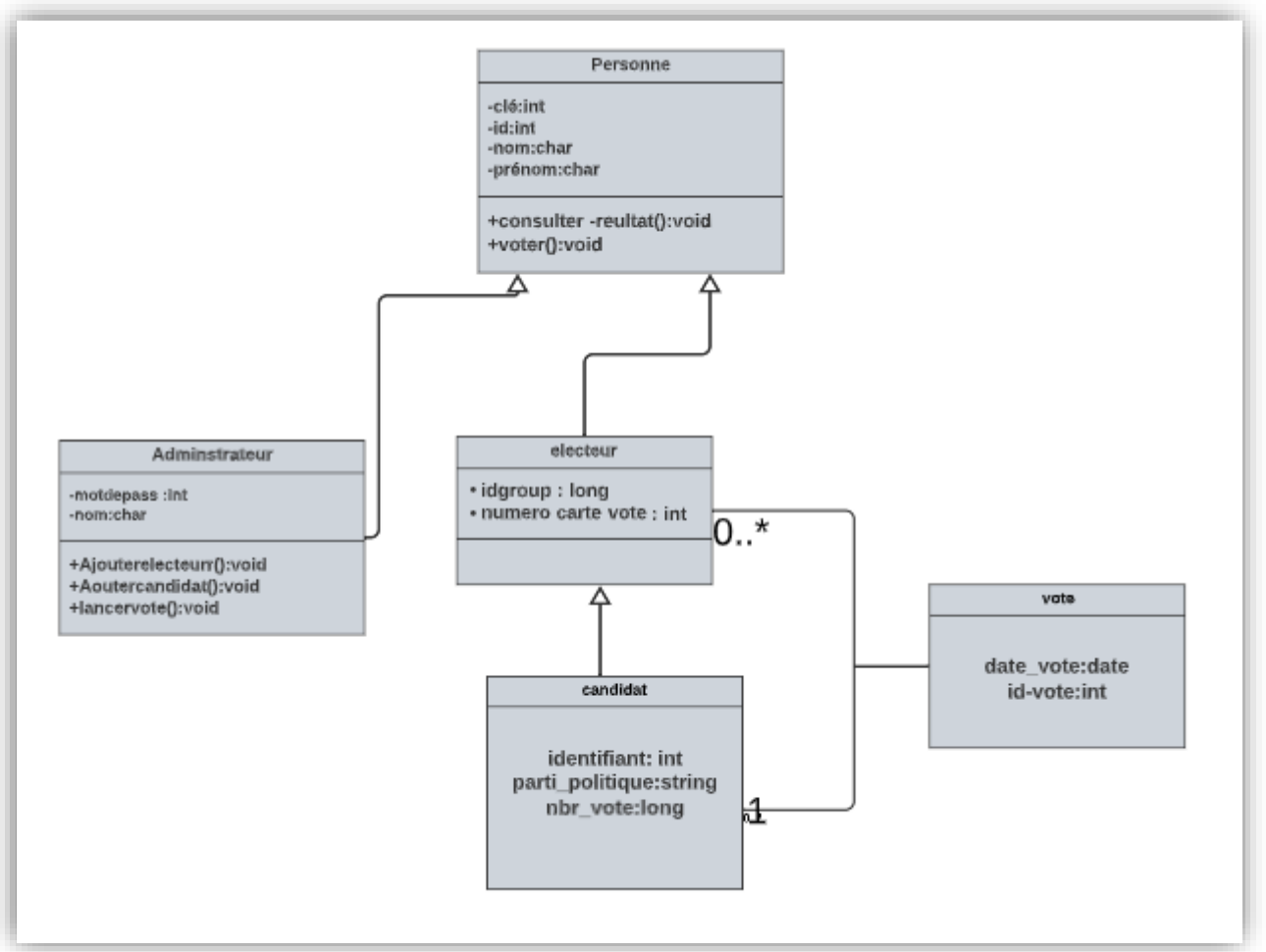


Figure II.11 diagramme de class de system de vote

II.4.4 Diagrammes de séquences :

a. Diagramme de séquence « Authentification » :

Le diagramme de séquence « Authentification » détaille le séquencement des différentes interactions entre l'administrateur et le système (voir la **Figure II.12**). Pour lui permettre à faire ses taches, l'authentification est obligatoire. Dans ce diagramme L'opérateur « **alt »** signifie une condition qui permet d'afficher la page de l'administrateur voulue et d'afficher un message de succès si les informations sont valides après la vérification, sinon, le système va afficher un message d'erreur.

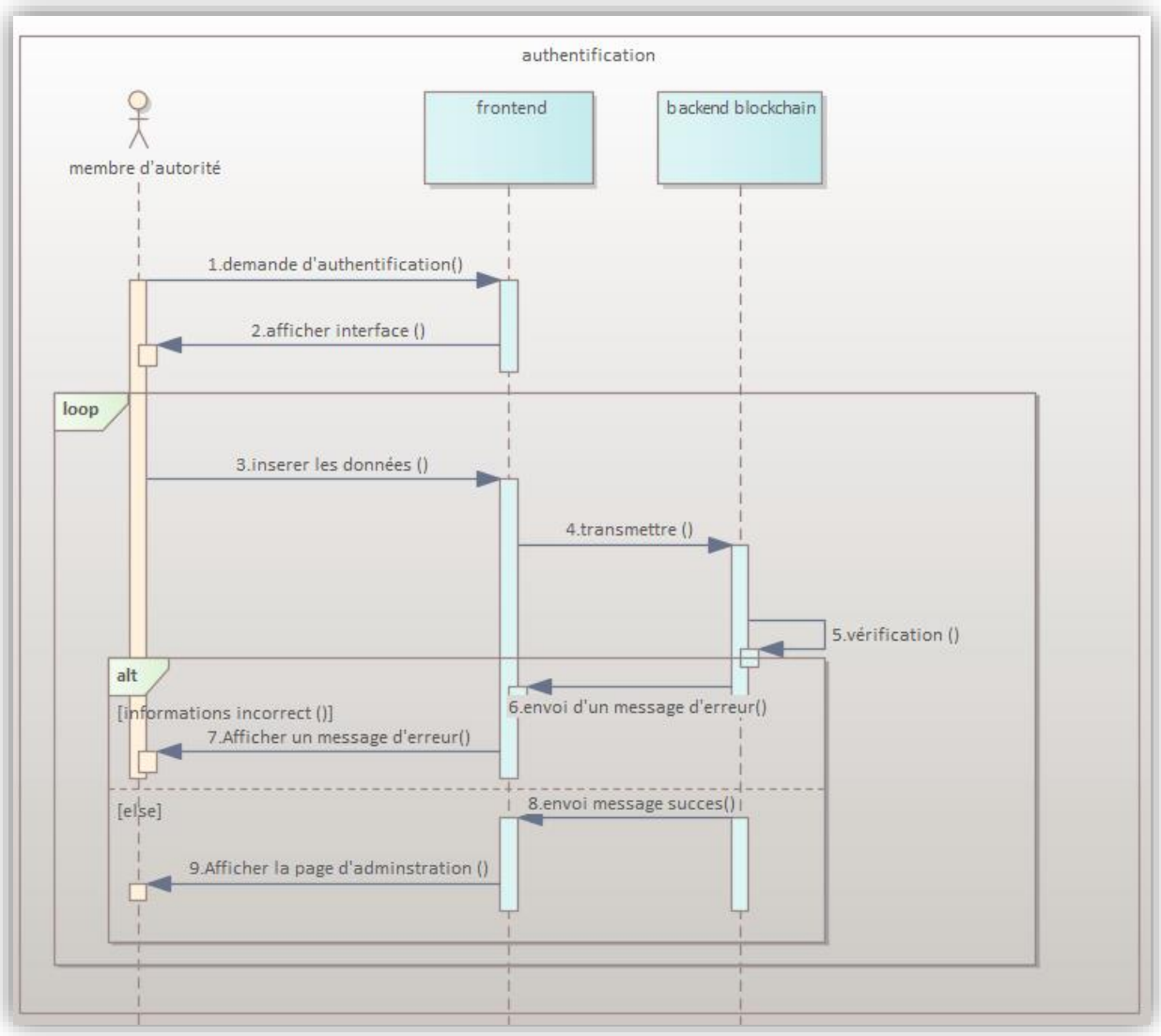


Figure II.12 diagramme de séquence "authentification membre autorité"

b. Diagramme de séquence « ajouter » :

Ce diagramme présente l'interaction de l'administrateur avec la blockchain. Comme la **Figure II.13** indique Après l'authentification de l'administrateur, il peut ajouter un électeur ou un candidat via le remplissage du formulaire spécifique affiché avec une interface. Les données saisies vont être validées après la vérification de format des champs. L'opérateur « **alt »** indique la condition qui va permettre à l'administrateur le droit d'ajouter et d'afficher un message de succès en plus de créer un compte dans le portefeuille Ethereum MetaMask seulement si les données entrées sont valides, sinon un message d'erreur va être affiché.

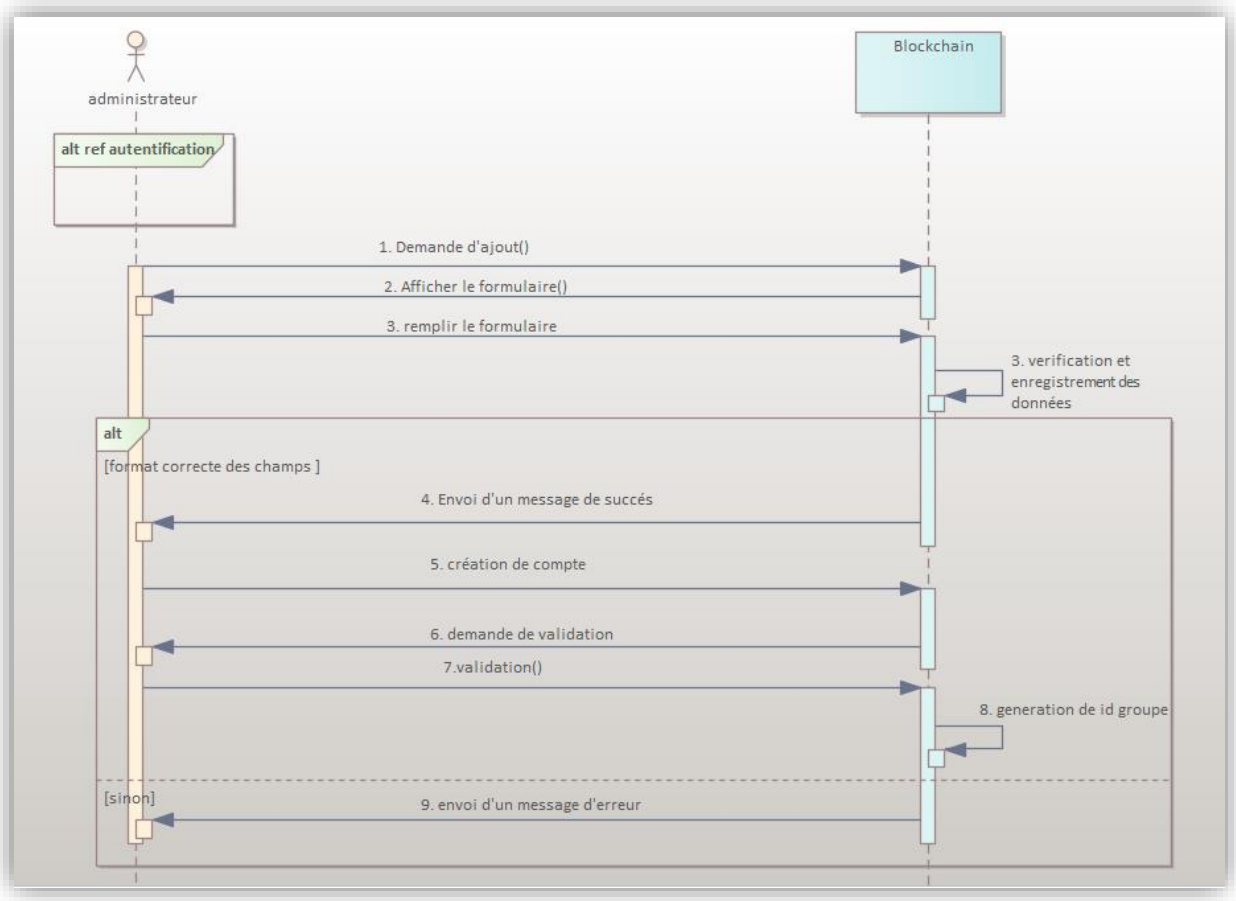


Figure II.13 diagramme de séquence "ajouter un électeur"

c. Diagramme de séquence « login » :

Le diagramme de séquence « login » indique le séquencement des interactions entre l'électeur et le système afin de se connecter (voir la **Figure II.14**). L'électeur dispose d'un navigateur pour accéder à l'interface de connexion et un portefeuille Ethereum qui est MetaMask pour lui permettre d'accéder à la blockchain et émettre des transactions. Pour cela l'électeur va remplir les champs affichés et se connecter à son compte MetaMask par le système.

Les deux opérateurs « alt » dans ce diagramme, indiquent deux conditions. La première condition vérifie le format des champs, s'il est correct les données vont être transmises pour vérifier la deuxième condition qui est la validation des données entrées, cela permet d'afficher la page contient les informations de compte d'électeur seulement, sinon le système affiche un message d'erreur.

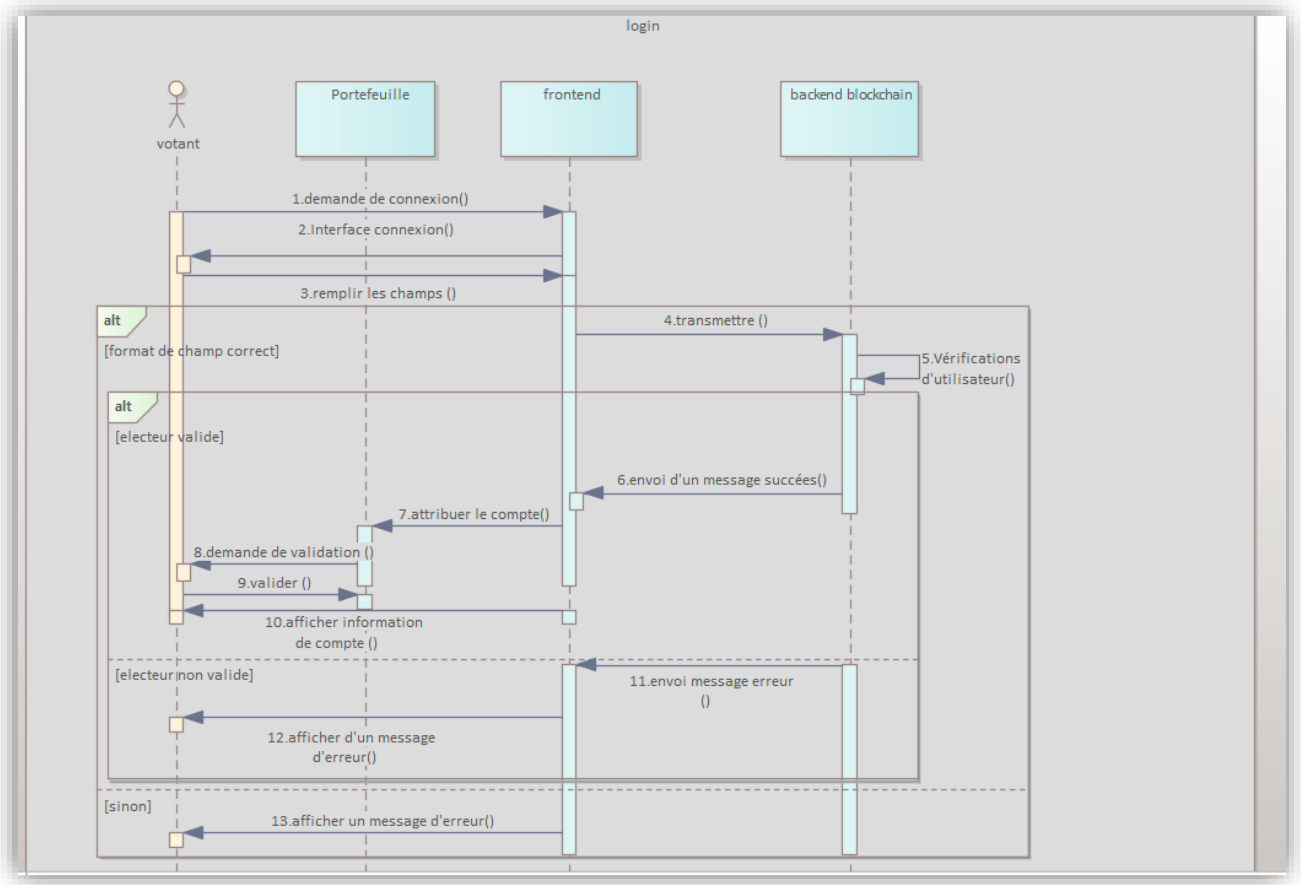


Figure II.14 diagramme de séquence "login votant"

d. Diagramme de séquence « vote » :

Le diagramme de séquence « voter » comme la **Figure II.15** diagramme de séquence "vote et validation de vote" montre il indique le séquencement des interactions entre l'électeur et le système afin de voter. L'électeur dispose d'un navigateur pour accéder à l'interface de vote et un portefeuille Ethereum qui est MetaMask pour lui permettre d'accéder à la blockchain et émettre des transactions. Pour cela le système va vérifier la date de vote pour que l'électeur peut voter.

Une fois l'électeur entre son choix le système va lui générer une clé image et sa transaction va s'envoyer pour exécuter le smart contract, ce dernier vérifie la validité de la transaction avec plusieurs règles. Un message de succès ou d'erreur va être envoyé à l'électeur selon son cas de condition.

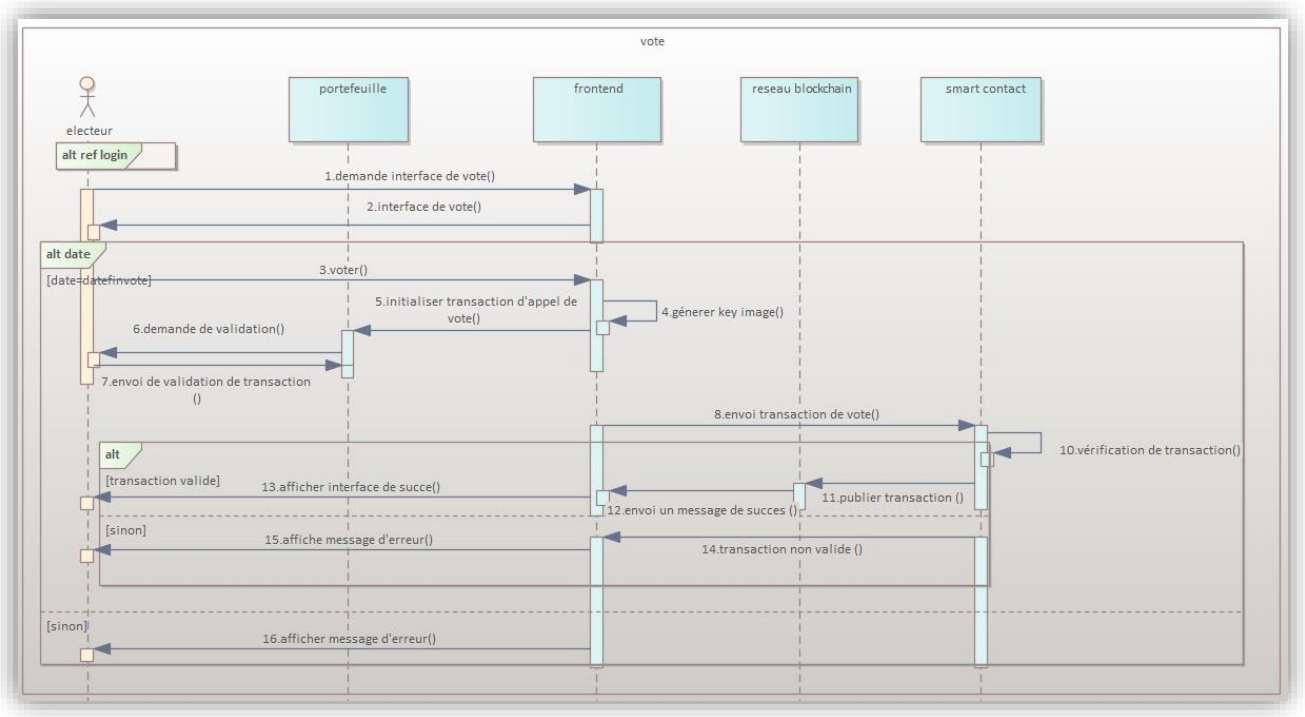


Figure II.15 diagramme de séquence "vote et validation de vote"

II.5 Conclusion :

Au cours de ce chapitre, nous avons montré le design de notre solution et son principe de fonctionnement et les différentes phases établies. En plus, les relations qui existent entre les différents acteurs et leurs interactions avec le système. Nous avons conçu une application pour le vote en s'adaptant sur les diagrammes d'UML à savoir le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classe, à cause de la limite des outils de développements que nous avons utilisés et le la limite de temps nous avons relâchés quelques conditions, ces derniers sont expliqués dans le chapitre suivant, où nous allons présenter les outils de développement ainsi les interfaces réalisées et leurs implémentations.

CHAPITRE III :

Implémentation et analyse de sécurité

Chapitre III : Implémentation et analyse de sécurité

III.1 Introduction :

Après avoir défini notre solution et élaboré la conception de notre application, dans ce chapitre nous allons concevoir et développer un prototype d'application décentralisé de vote électronique. D'abord en commence par spécifier et définir les outils et langages utilisés dans la réalisation de notre application décentralisée. Ensuite, nous allons montrer toutes les étapes de configurations des environnements. Après, nous allons indiquer comment les phases de système sont implémentées en citant les différentes interfaces implémentées avec ses descriptions, puis nous avons élaboré une évaluation au niveau de sécurité de solution proposé. A la fin, nous avons conclus par une petite conclusion.

III.2 Outils de développement :

Les outils utilisés pour la réalisation de notre site web sont :

III.2.1 Truffle :

Framework de développement sur Ethereum créé par Consensus, qui fournit une suite d'outils pour développer des contacts intelligents Ethereum avec le langage de programmation Solidity. [27]

Nous avons choisi truffle pour le développement de notre site web car c'est un Framework puissant qui nous facilite l'interaction avec notre smart contract et il nous permet de la déployer et la tester.

Voici un aperçu de toutes les fonctionnalités que nous obtiendrons avec le framework Truffle:

- **Gestion des contracts intelligents :** il permet d'écrire des smarts contracts avec le langage de programmation Solidity et les compiler en bytecode qui sera exécuté sur la machine virtuelle Ethereum (EVM)
- **Déploiement et migrations :** rédiger des scripts pour migrer et déployer des contracts intelligents sur n'importe quel réseau public de blockchain Ethereum.
- **Gestion du réseau :** connecter à n'importe quel réseau blockchain Ethereum public, ainsi qu'à tout réseau blockchain personnel que vous pourriez utiliser à des fins de développement.
- **Console de développement** : interagir avec les contracts intelligents dans un environnement d'exécution JavaScript avec la console Truffle. Vous pouvez vous connecter à n'importe quel réseau blockchain que vous avez spécifié dans votre configuration réseau pour ce faire.
- **Développement côté client :** configurer le projet truffle pour héberger des applications côté client qui communiquent avec vos contracts intelligents déployés sur la blockchain.

Pour installer le Framework truffle, Voici la commande d'installation :

$ npm install truffle –g

III.2.2 Ganache :

Ganache est une application qui fait partie de la suite truffle qui permet de créer une blockchain locale à base ethereum qui rend le développement d'applications distribuées plus rapide, plus facile et plus sûr, il permet d'exécuter et tester des contracts intelligentes.

Ganache se décline en deux versions : une interface utilisateur et une interface de ligne de commande. Ganache UI est un logiciel qui utilise les technologies Ethereum. L'outil de ligne de commande, ganache-cli (anciennement connu sous le nom de Test RPC), est disponible pour le développement Ethereum. Il est disponible sur Windows, Mac et Linux [28] . **Figure III.16** montre l'interface graphique de ganache.

Ganache nous permet d'avoir des comptes ethereum avec une balance des faux ethers pour chaque compte. Il permet aussi de voir l'état actuel de tous les comptes, y compris leurs adresses, clés privées, transactions et soldes (balances). Aussi d'examiner tous les blocs et transactions pour avoir un aperçu de ce qui se passe.

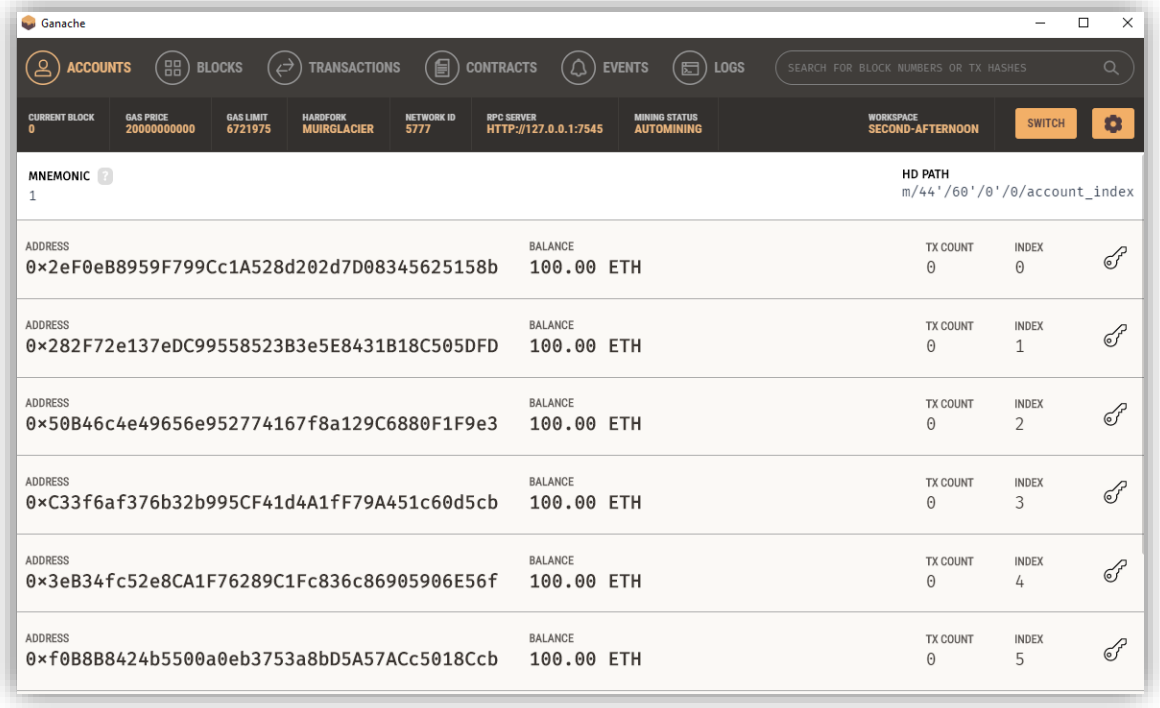


Figure III.16 interface Ganache

Nous l'avons choisi car il nous permet de tester et déployer le contrat intelligent et d'examiner tout ce qui se passe dans cette blockchain pendant le développement (examiner les transactions, les évènements ... etc.).

III.2.3 Node.js :

Un environnement d'exécution côté serveur open source basé sur le moteur JavaScript V8 de Chrome. Il peut être utilisé pour créer différents types d'applications telles que les applications Web, les applications de chat en temps réel, il fonctionne sur différentes plates-formes (Windows, Linux, Unix, Mac OS X, etc.).

Pour développer des smart contracts, nous devons configurer notre environnement par l'installation de Node Package Manager(NPM), fourni avec Node.js. [29]

III.2.4 Web3.js :

Web3.js est une bibliothèque populaire qui permet aux programmeurs d'interagir avec la blockchain Ethereum. Il représente une liaison de langage JavaScript pour l'interface JSON RPC d'Ethereum, ce qui le rend directement utilisable dans la technologie Web, car JavaScript est pris en charge nativement dans presque tous les navigateurs Web3.js est également couramment utilisé du côté serveur dans les applications Node.js et dans les applications bureau. Web3.js peut être utilisé pour se connecter au réseau Ethereum via n'importe quel nœud qui permet l'accès via http. [30]

III.2.5 Le Portefeuille Meta mask :

Il s'agit d'une extension de navigateur populaire pour les navigateurs Web, il apparaît comme une solution pour faciliter aux utilisateurs d'interagir facilement avec les applications décentralisées de la blockchain Ethereum, il fonctionne comme un portefeuille de cryptomonnaies et joue le rôle d'un pont entre les navigateurs normaux et la blockchain Ethereum, il permet de rendre le développement d'applications décentralisées plus simple.

L'extension injecte l'API web3 Ethereum dans le contexte javascript de chaque site Web, afin que les applications décentralisées puissent lire à partir de la blockchain.

Nous avons choisi de travailler par meta mask parce qu' il nous permet la gestion des comptes Blockchain, ainsi lorsqu'un Dapp souhaite effectuer une transaction et écrire dans la blockchain, l'utilisateur dispose d'une interface sécurisée pour examiner la transaction, avant de l'approuver ou de la rejeter. Nous avons installé cette extension depuis la boutique Google [31].

III.2.6 Visual Studio Code :

C'est un éditeur de code extensible développé par Microsoft pour des différents system d'exploitation (Windows, Linux et MacOS). Les fonctionnalités offertes par cet éditeur permettent le débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, il supporte plusieurs langages de programmation (java , JavaScript, Go, Node.js et C ++) . [32]

III.2.7 Solidity :

C'est un langage de programmation orienté objet de haut niveau qui présente des similitudes avec JavaScript ou C++, il sera compilé en langage de bas niveau (bytecode) et interprété par l'environnement d'exécution d'Ethereum. [33]

Solidity permet de coder des smart contracts pour mettre à jour l'état de la blockchain en fonction des conditions et des variables définies par leur créateur. [33]

Solidity est de type statique, prend en charge l'héritage, les bibliothèques et les types complexes définis par l'utilisateur, entre autres fonctionnalités [34]

III.2.8 Moralis :

Moralis est une plateforme qui facilite la réalisation des applications décentralisées, il permet de visualiser les processus faites dans le backend de la blockchain tels que : les transactions, la synchronisation des adresses et les actions effectués par les utilisateurs de la blockchain ainsi que les événements de contrat intelligents, en plus il fournit toutes les fonctionnalités en inter-chaines qui sont accessibles via un SDK facile à utiliser. [35]

III.3 Configuration d'environnements :

Dans cette section, nous allons citer les configurations nécessaires pour chaque outil avant de commencer l'implémentation :

III.3.1 Configurations ganache et smart contract :

Nous allons d'abord créer un répertoire qui va contenir les fichiers de notre projet comme ceci :

$ mkdir dappvote

$ cd dappvote

Ensuite, nous initialisons un nouveau projet truffle pour développer notre projet comme suit :

$ truffle init

Après il faut installer les dépendances par cette commande :

$ npm install

La sortie de terminal indique que le projet a été créé avec succès. Maintenant, nous pouvons ouvrir l'éditeur de texte et voir que de nouveaux fichiers et répertoires ont été créés. La **Figure III.17** montre la structure de répertoire dappvote.

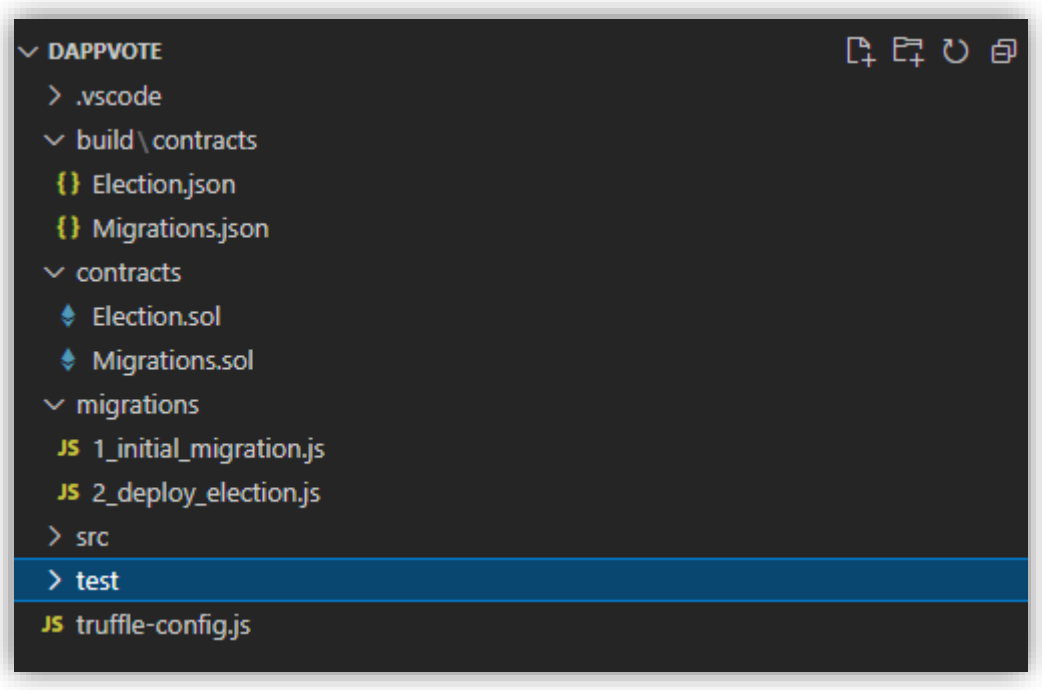


Figure III.17 Structure de répertoire

- **Répertoire build/contracts :** il contient des fichiers JSON qui sont générés après chaque compilation des smarts contracts. Le fichier JSON est l'ABI (Abstract binary interface) de smart contract qui sera exécuté sur la machine virtuelle de blockchain**.**

Remarque : l'ABI est un code binaire qui décrit l'interface du contract et les méthodes qui peuvent être appelées par d'autres smart contracts ou par des applications.

- **Répertoire contracts :** c'est là que se trouve tous les Smart contracts, Nous avons déjà un contract de migration qui gère nos migrations vers la blockchaine.
- **Répertoire migrations :** il contient tous les fichiers de migration. Chaque fois que nous déployons des smarts contracts sur ganache ,il est obligé de mettre à jour la blockchain à l'aide d'une migration.
- **Fichier truffle-config.js :** c'est un fichier de configuration principale de projet truffle, il nous permet de gérer la configuration de réseau.

Maintenant nous pouvant créer notre smart contract par la création d'un fichier Election.sol dans le répertoire 'contracts'. Nous commençons par la spécification de la version de notre compilateur solidity. Après ça, nous compilons le smart contract et vérifions qu'il n y'ait pas d'erreur en utilisant la commande :

$ truffle compile

Maintenant pour accéder au contrat intelligent sur ganache dans la console Truffle. Nous devons effectuer quelques taches :

- D'abord en Mettre à jour le fichier de configuration **trufle-config.js** pour spécifier le réseau blockchain personnel auquel nous voulons nous connecter (Ganache).

```
networks: {
    development: {
    host: "127.0.0.1",     // Localhost (default: none)
    port: 7545,            // Standard Ethereum port (default: none)
    network_id: "*",       // Any network (default: none)
   },
   compilers: {
    solc: {
      version: "0.8.10",    // Fetch exact version from solc-bin (default: truffle's version)
      // docker: true,        // Use "0.5.1" you've installed locally with docker (default: false)
       settings: {          // See the solidity docs for advice about optimization and evmVersion
       optimizer: {
         enabled: true,
        runs: 200
        },
      //  evmVersion: "byzantium"
       }
    }
  },
```

Figure III.18 Configuration réseau

- Ensuite nous allons créez un fichier sous le nom 2_deploy_election.js dans le répertoire migrations. Ce fichier contient un script de migration qui indique à Truffle comment déployer le contrat intelligent sur le réseau blockchain personnel.

```
1  const Election = artifacts.require("./Election.sol");
2  module.exports = function (deployer) {
3    deployer.deploy(Election);
4  };
5
```

Figure III.19 Fichier de déploiement

- Après nous allons éxecuter le script de migration nouvellement créé, en déployant le contrat intelligent sur le réseau blockchain personnel (Cela consommera du gas) et en éxecutant une commande dans la ligne de commande comme suit :

$ truffle migrate

III.3.2 Configuration MetaMask :

Pour que on peut intéragir avec le smart contract deployé dans la blockchain et éffectuer des transactions, nous allonns d'abord ajouter le réseau blockchain local dans MetaMask en saisissant l'URL RPC et l'ID de chaîne (voir la **Figure III.20**):

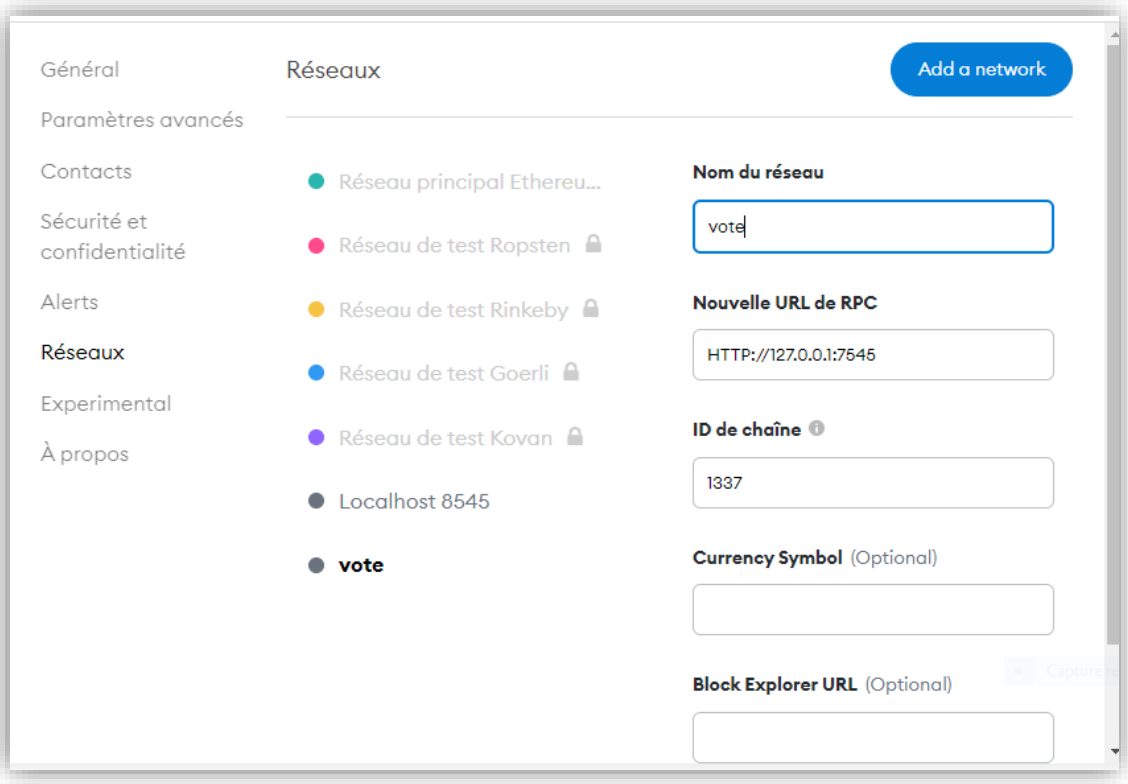


Figure III.20 Création de réseau

Ensuite pour utiliser les comptes offerts par ganache, nous devons créer un nouveau compte en saisissant la clé privée copiée d'un compte existant sur ganache après cela on obtient un compte MetaMask avec 100ETH (configuration par defaut).

III.3.3 Configuration initiale de Moralis:

En suivant les étapes ci-dessous [36], on aura des choses prêtes à créer des applications distribuées (dApps):

1. **Création d'un compte Moralis gratuitement** : se fait par un email et un mot de passe.
2. **Création d'un serveur Moralis :** choisir un serveur selon les besoins, utiliser le testnet Mumbai de Polygon. Cependant, lorsqu'on est prêt à présenter notre dApp aux utilisateurs, nous devons choisir le « serveur Mainnet » pour atteindre le public.
3. **Entrer dans les Détails du serveur créé** : pour savoir toutes les informations comme URL du serveur et ID de l'application... Etc.
4. **Initialiser Moralis :** Pour obtenir un accès complet à la puissance de Moralis, qui couvrira nos besoins backend, nous remplissons le fichier de codage avec l'ID de l'application et l'URL du serveur comme indiqué dans **Figure III.21**

Figure III.21 initialisation de moralis

5. **Liaison de Moralis avec ganache** en suivant ces étapes (voir **Figure III.22**):

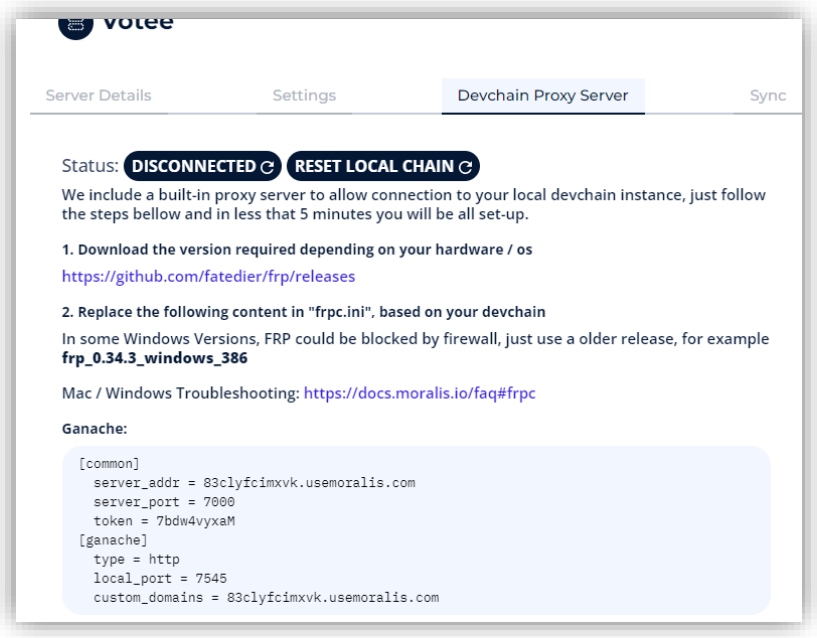


Figure III.22 Configuration de serveur Moralis

- Télécharger le fichier rpc [37]

- Remplacez le contenu suivant dans "frpc.ini" (**Figure III.23**) :

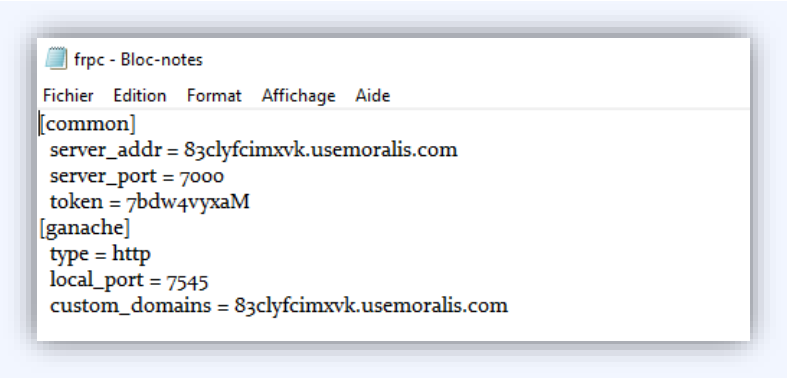


Figure III.23 Fichier "frpc.ini"

- Exécuter la commande dans le cmd (Windows):

$ frpc.exe -c frpc.ini

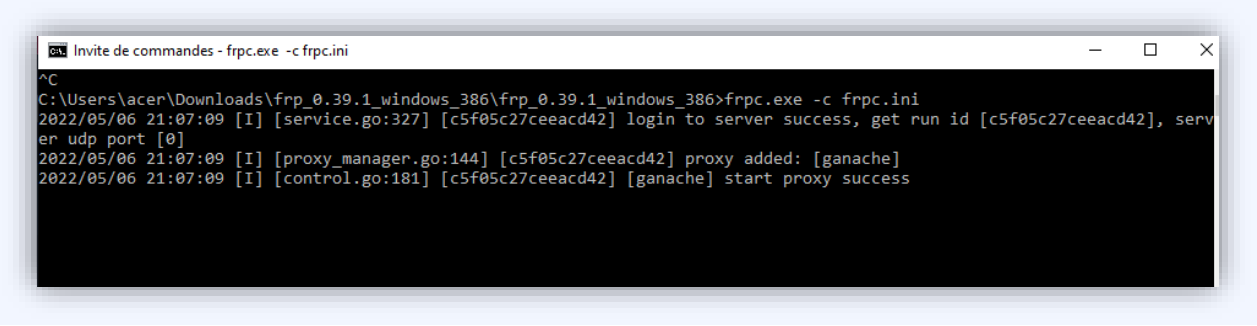


Figure III.24 Connexion de Ganache avec Moralis

- Le serveur maintenant est connecté comme la ***Figure III.25*** montre :

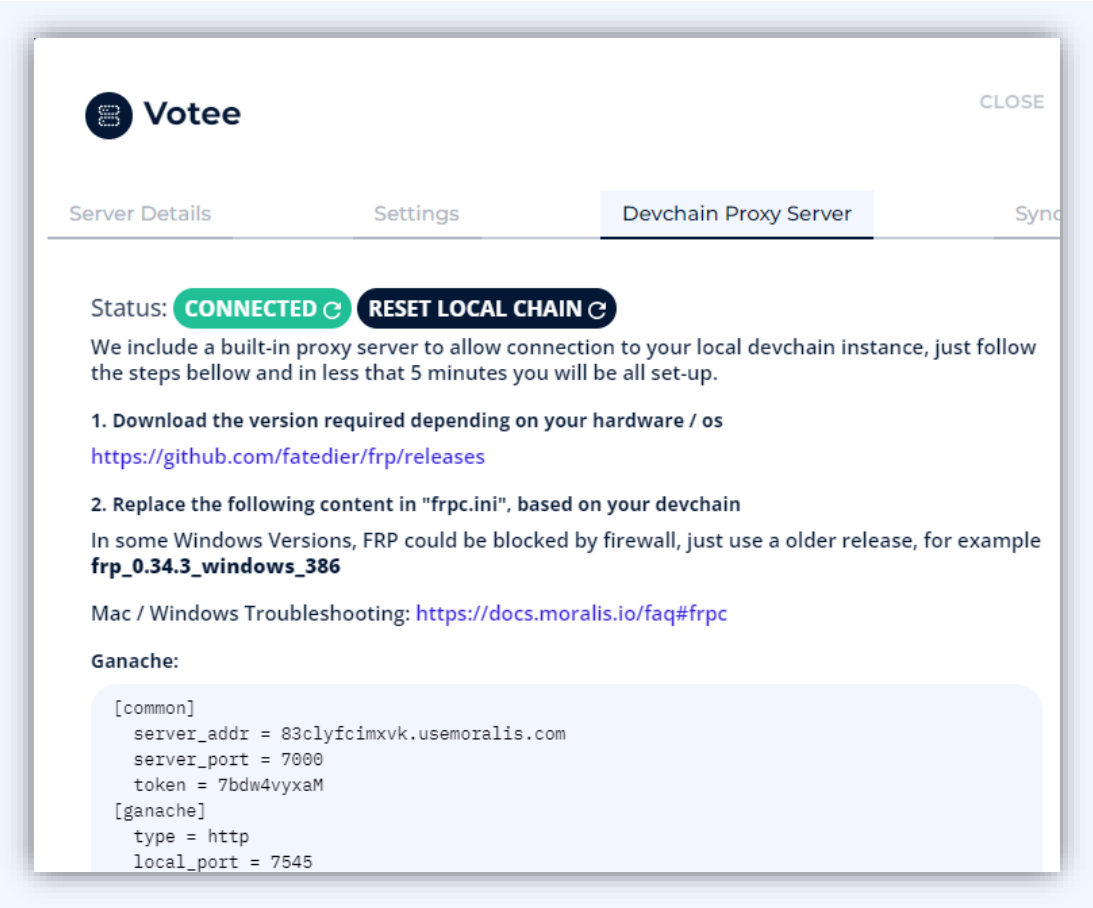


Figure III.25 Serveur connecté

III.4 L'implémentation de l'application décentralisée :

Dans cette partie, nous allons définie notre travail qui est une application décentralisée de vote électronique qui s'appuie sur le système de blockchain en montrant comment les différentes phases de systèmes cité dans chapitre 2 sont réalisés, et en démontrant comment la sécurité est assuré en justifiant certaines conditions qui sont soulagées, ainsi en expliquant le déroulement de smart contract.

III.4.1 L'implémentation des phases :

Nous avons divisé les explications de l'implémentation de notre application web en 2 parties. La première partie, comprend les différentes fonctions d'administrateur (membre d'autorité) et leurs implémentations. La deuxième partie concerne l'électeur. La **Figure III.26** illustre les deux types des utilisateurs.

Figure III.26 Interface d'accueil

A. Coté Administrateur:

a. Login d'admin :

La première phase, comme elle est déjà expliquée dans le chapitre 2 elle est exécutée par l'agent d'AE. En utilisant les outils précédemment listés nous avons pu réaliser des interfaces qui facilitent les tâches de l'administration. Comme nous n'avons pas une autorité de vote réel, nous avons créé un seul utilisateur admin pour représenter l'autorité.

D'abord, une authentification d'administrateur doit être effectué , il doit saisir son clé publique après il doit s'authentifier par un mot de passe et nom, puis une authentification avec le portefeuille MetaMask est nécessaire : nous allons attribuer un compte MetaMask au administrateur , à cause de la limite de temps et de matérielle nous n'avons pas pu réaliser l'authentification proposé dans notre solution.

Comme nous avons mentionné auparavant dans la description de solution proposé, le modèle de consensus proposé était le PoA qui autorise seulement les membres d'autorité de valider les transactions et créer les blocs chainés. Cependant, dans notre implémentation courante nous utilisons

le PoW offert par ganache. Notant que ce dernier n'offre pas clairement l'option de changer le modèle de consensus ni de définir un nouveau mécanisme de consensus. Aussi, à cause de la limite de temps, l'absence de la documentation sur ce point et la difficulté technique relative à la localisation et la modification des différentes procédures relatives au consensus, nous avons décidé de ne pas modifier le consensus dans ce prototype et de reporter cette tâche à la future lors de la finalisation de l'application.

Une fois que la connexion est établie avec succès il peut accéder aux autres interfaces. La **Figure III.27** montre l'interface de login :

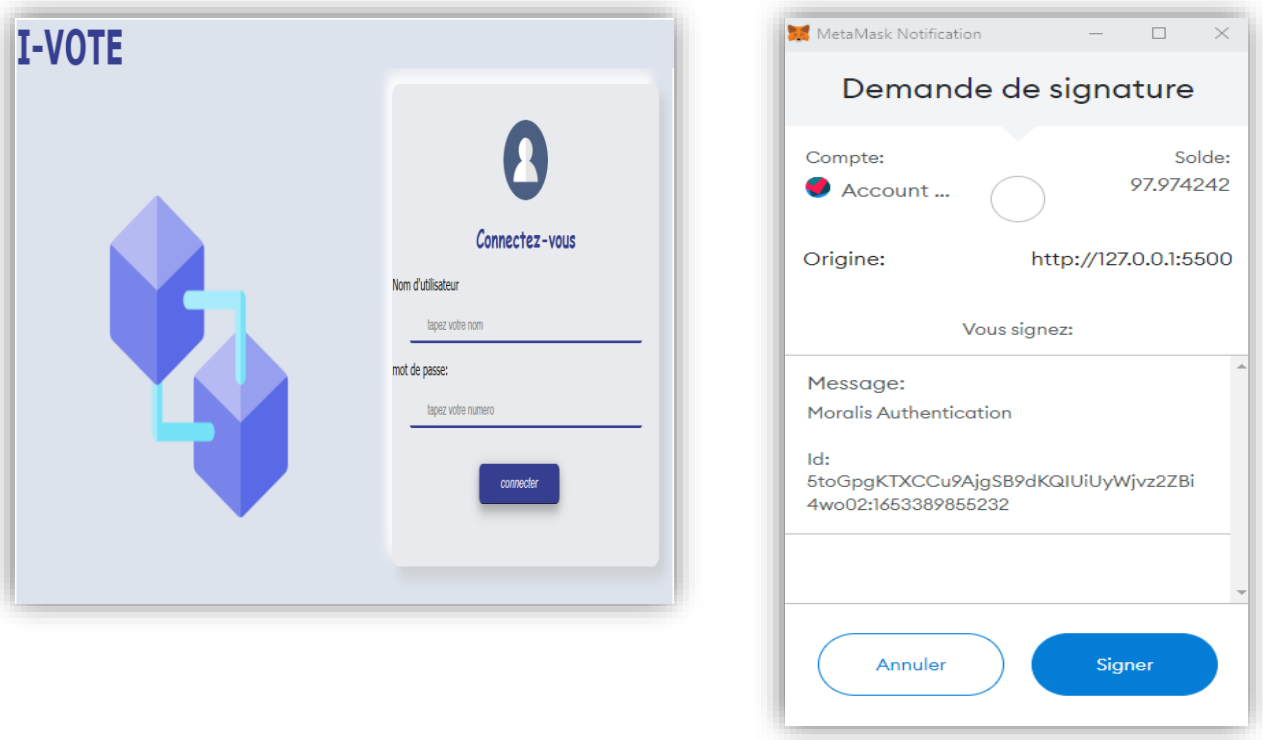


Figure III.27 : interface login administrateur

b. Phase d'inscription :

Cette phase représente dans notre prototype l'inscription des candidats et des votants à travers une base de données sur Moralis.

- **Candidats :**

D'abord, une interface (voir **Figure III.28**) contient des champs de saisie pour entrer les informations des candidats éligibles (nom complet, numéro national, parti politique et l'identifiant et les informations de candidats). Dans ce prototype, nous avons attribué trois comptes MetaMask pour les candidats et un compte pour le vote blanc.

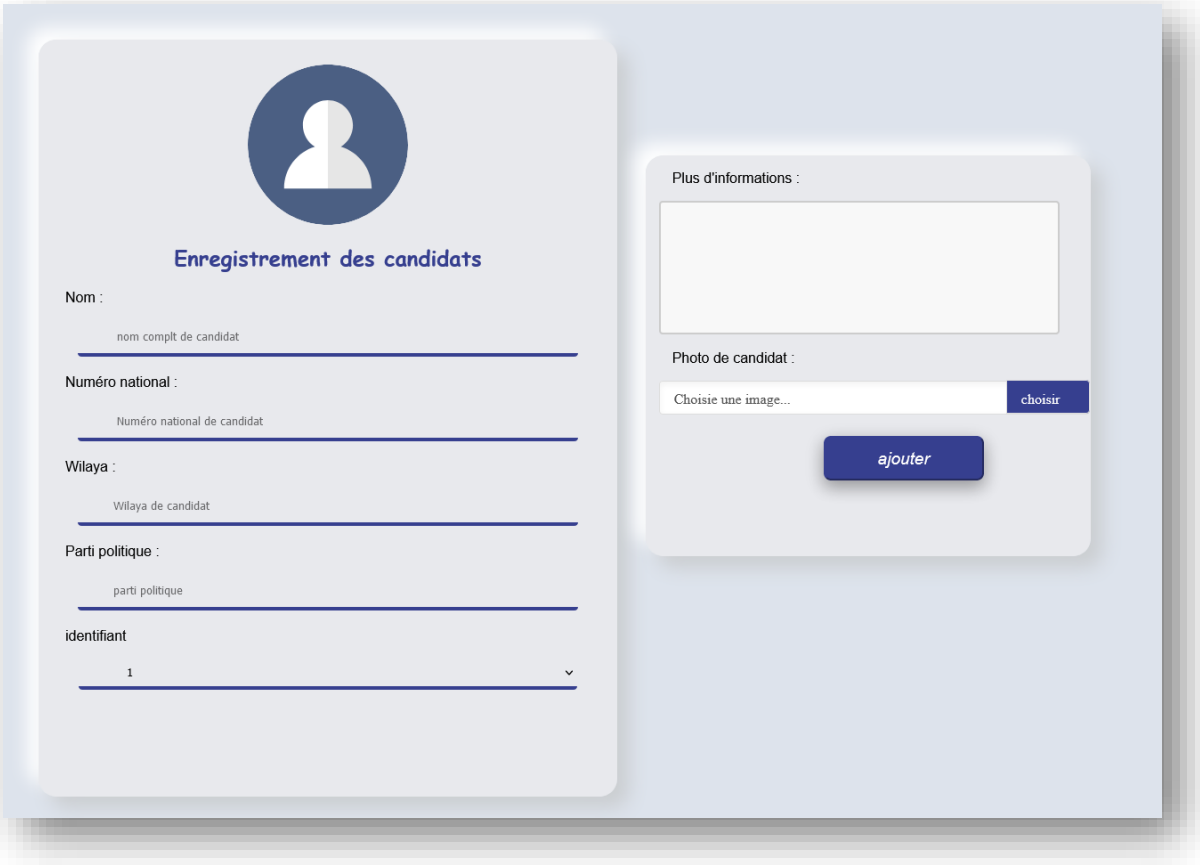


Figure III.28 interface d'ajout des candidats

- **Votants :**

L'inscription des votants se fait par le remplissage des champs avec leurs informations présentent dans leurs cartes d'identités, qui sont : le nom complet numéro national, numéro de wilaya, numéro de commune. Ses données sont enregistrées dans la base de données Moralis.

L'autorité (dans notre cas notre admin à travers le système) attribue un identifiant de groupe aux votants à leur inscription en plus des comptes MetaMask qui leur permettent de voter interagir avec la blockchain**.** La **Figure III.29** indique l'interface d'inscription de votant.

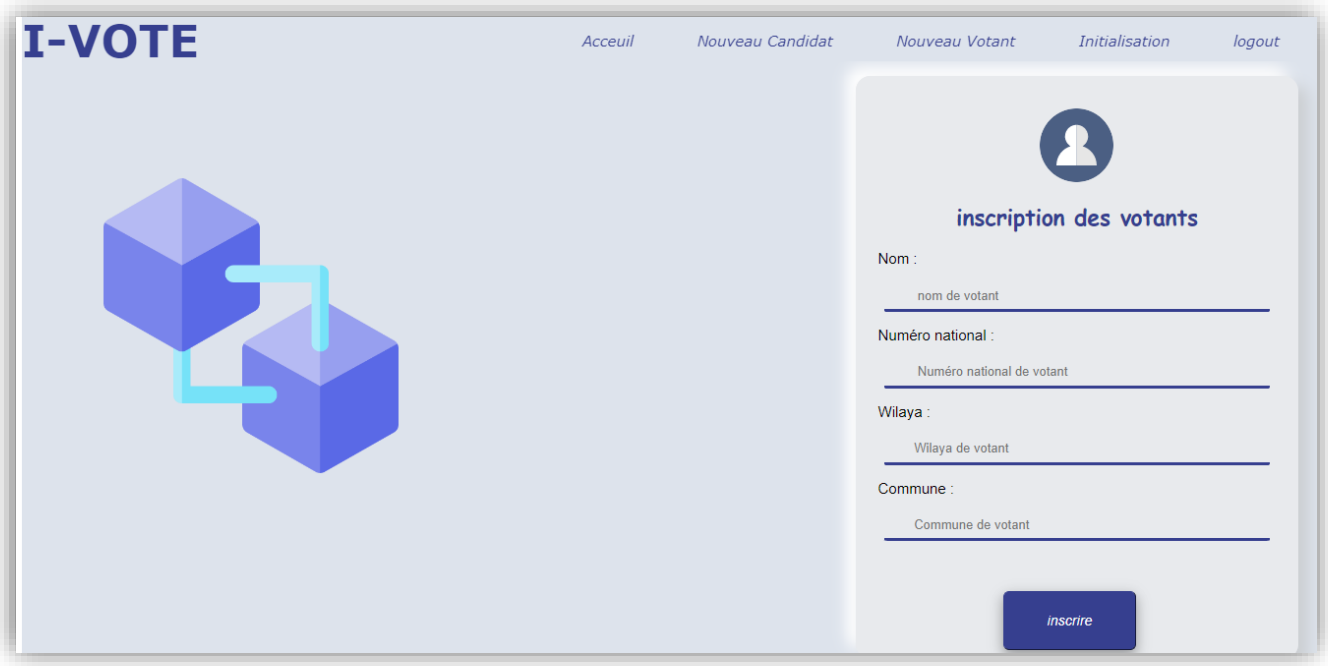


Figure III.29 interface d'inscription de votant

c. Initialisation de vote :

L'administrateur doit initialiser le vote avec une date précise, le vote se fait dans un seul jour à travers cette interface :

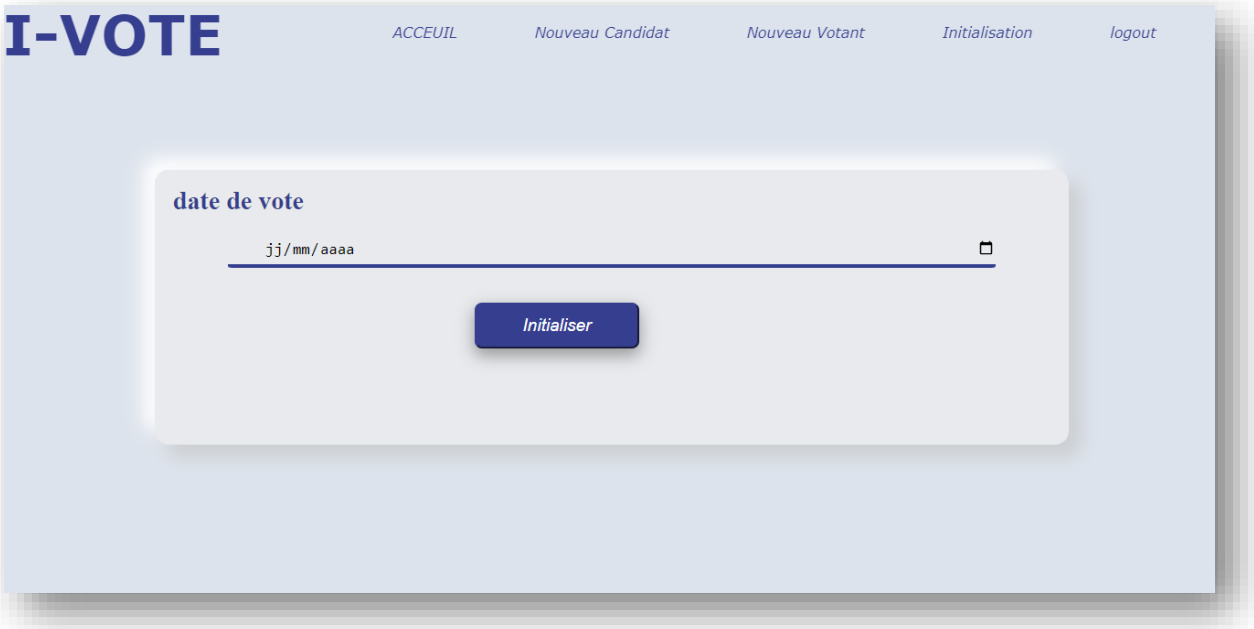


Figure III.30 interface d'initialisation de vote

B. Coté votant :

a. Login :

Le votant doit utiliser sa clé publique **Figure III.31** , son nom d'utilisateur et son mot de passe pour connecter à notre application. Ensuite, il doit aussi s'authentifier à MetaMask (voir **Figure III.32**). Notant qu'il sera demandé de changer son mot de passe lors de la première connexion à notre système comme illustré dans la **Figure III.33**.

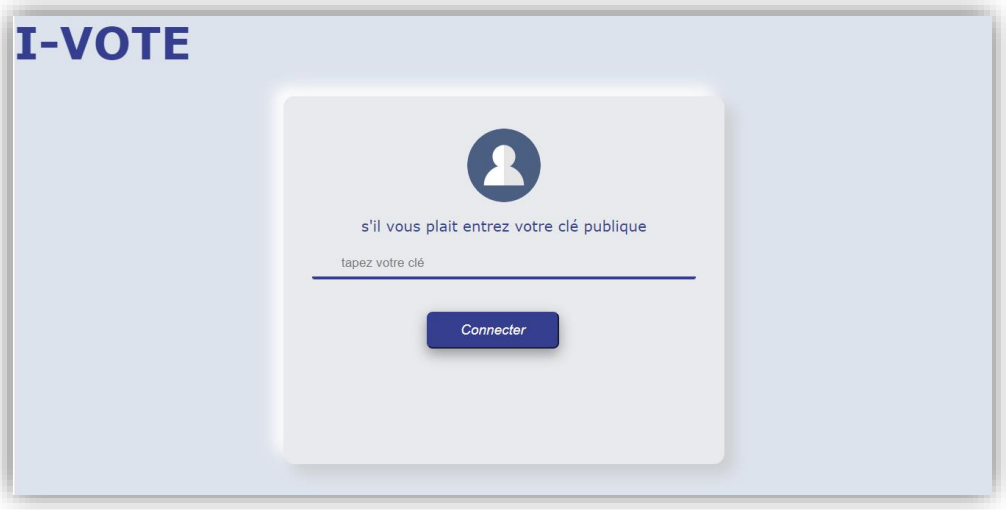


Figure III.31 clé publique

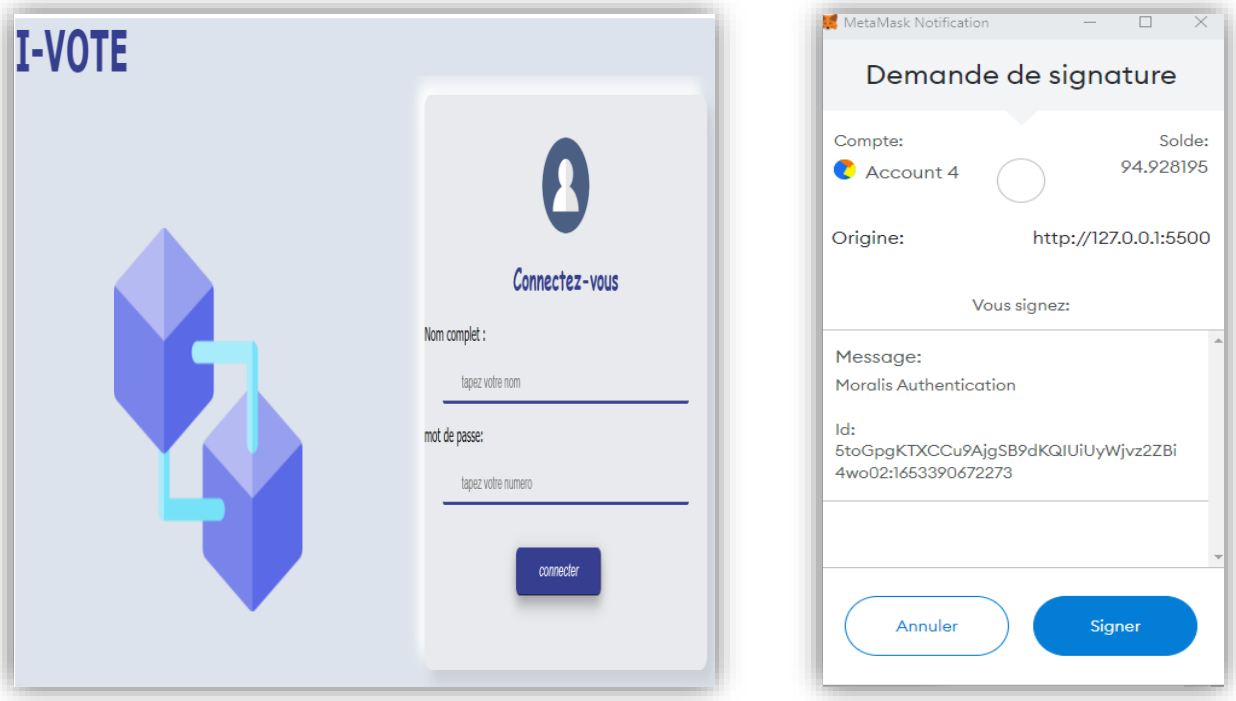


Figure III.32 interface login électeur

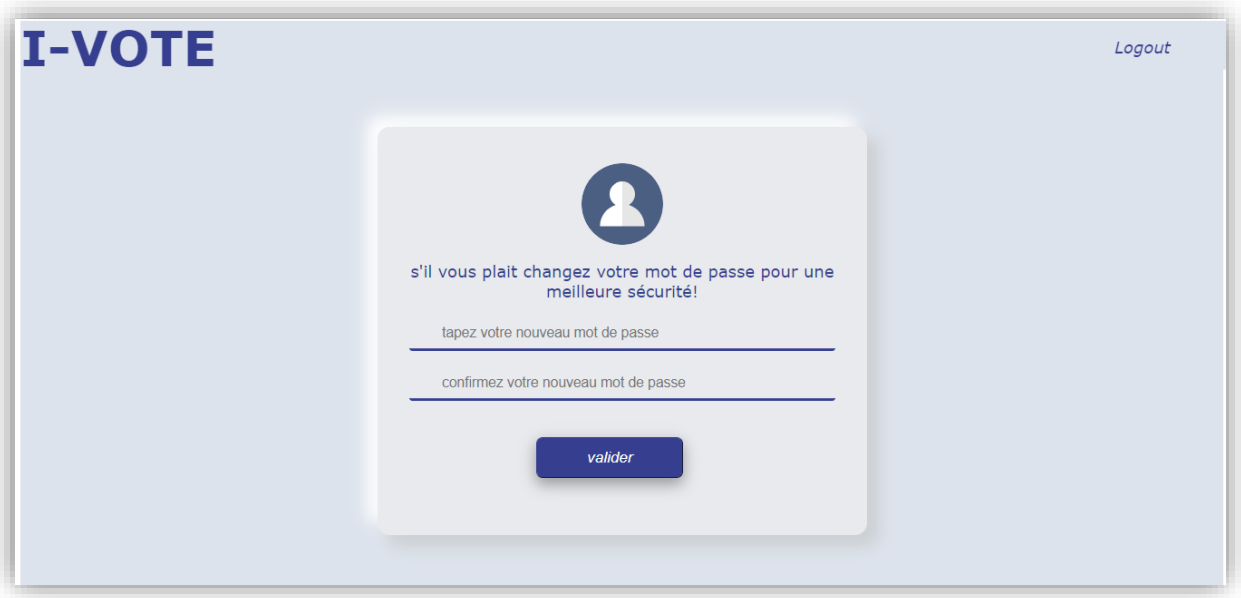


Figure III.33 interface change mot de passe

b. Consulter les candidats :

L'électeur peut voir les candidats et leurs informations, ainsi que leurs perspectives et visions en visitant l'onglet « consultation des candidats » de notre application comme illustré dans la **Figure III.34**

Figure III.34 interface de consultation candidats

c. Voter :

Notre application permet au votant de voter seulement pendant la période du vote précisée par l'administrateur qui y'a initié la session de vote. Pour assurer ça, la fonctionnalité de vote sera disponible seulement si la date courante est la date correspondante au jour de vote, après cette date les votants ne pourront pas voter (bouton voter et la zone de saisie de numéro de candidats seront cachés) mais pourront voir le nombre de vote obtenu pour chaque candidat. Notant que cette phase de vérification est programmée sur le programme JavaScript de vote puisque Solidity n'a pas de type date.

Pour voter, l'électeur doit choisir un candidat en tapant son identifiant correspondant à 1-3 et 4 pour le vote blanc (**Figure III.35**) Après qu'il entre l'identifiant, une image clé va être générée pour se faire, il doit signer avec son compte sur le hache SHA-256 de clé publique par l'utilisation d'une fonction offert par Web3.js .

La signature qui est la clé image sera envoyé au smart contract avec l'identifiant choisie, le nom correspondant et l'identifiant de group généré auparavant utilisant le web3.js

Dans la fonction de vérification exécuté par le smart contract:

- Premièrement, l'identifiant de groupe est vérifié s'il est correct. Dans la phase de conception nous avons proposé que l'identifiant du groupe sera l'adresse publique du votant dans la blockchain pour lui permettre de garder sa confidentialité en se cachant dans la foule. Cependant, dans la phase de l'implémentation, nous avons utilisé MetaMask pour la gestion de compte et aussi pour la confirmation des transactions, et comme il ne permet pas de modifier l'adresse publique et il utilise seulement celle du ganache, nous avons inséré l'identifiant du groupe comme un champ dans la transaction au lieu de l'utiliser comme adresse.
- Deuxièmement, l'exactitude d'identifiant de candidat saisie par le votant est vérifiée, si l'identifiant entré n'est pas correct une alerte sera affichée.
- Troisièmement, pour interdire la double tentative de vote et assurer que chaque électeur vote une seule fois, l'existence unique de la clé image dans la blockchain est vérifiée. Si, la clé image déjà existe, le votant ne peut pas voter et le bouton **voter** sera cacher avec la zone de saisie et le système va afficher à la fois une alerte indique qu'il a déjà voté. Sinon, si la clé n'existe pas et toutes les autres données sont validées le vote est réalisé résultant à la publication de deux transactions dans la blockchain chacune dans un block. La première transaction est pour la validation des informations qui est adressée vers le smart contract comme montré dans **Figure III.36**. La deuxième est pour envoyer un Ether à l'adresse candidat choisie **Figure III.37**

Figure III.35 interface de vote

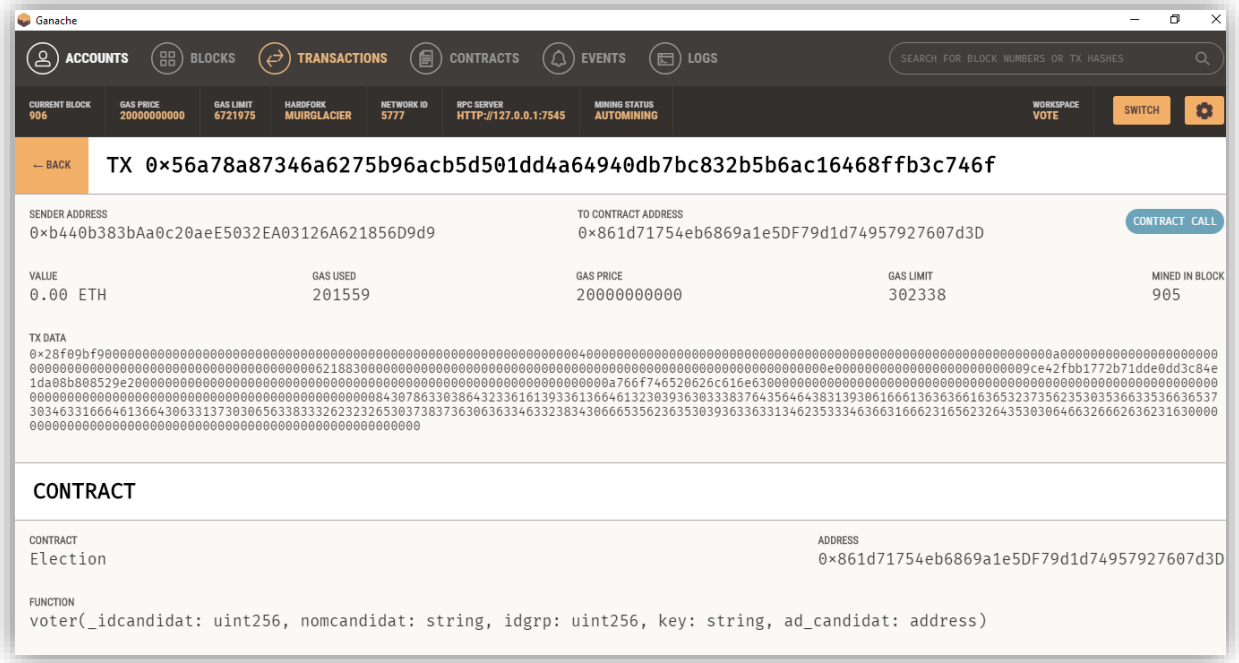


Figure III.36 transaction vers le smart contract.

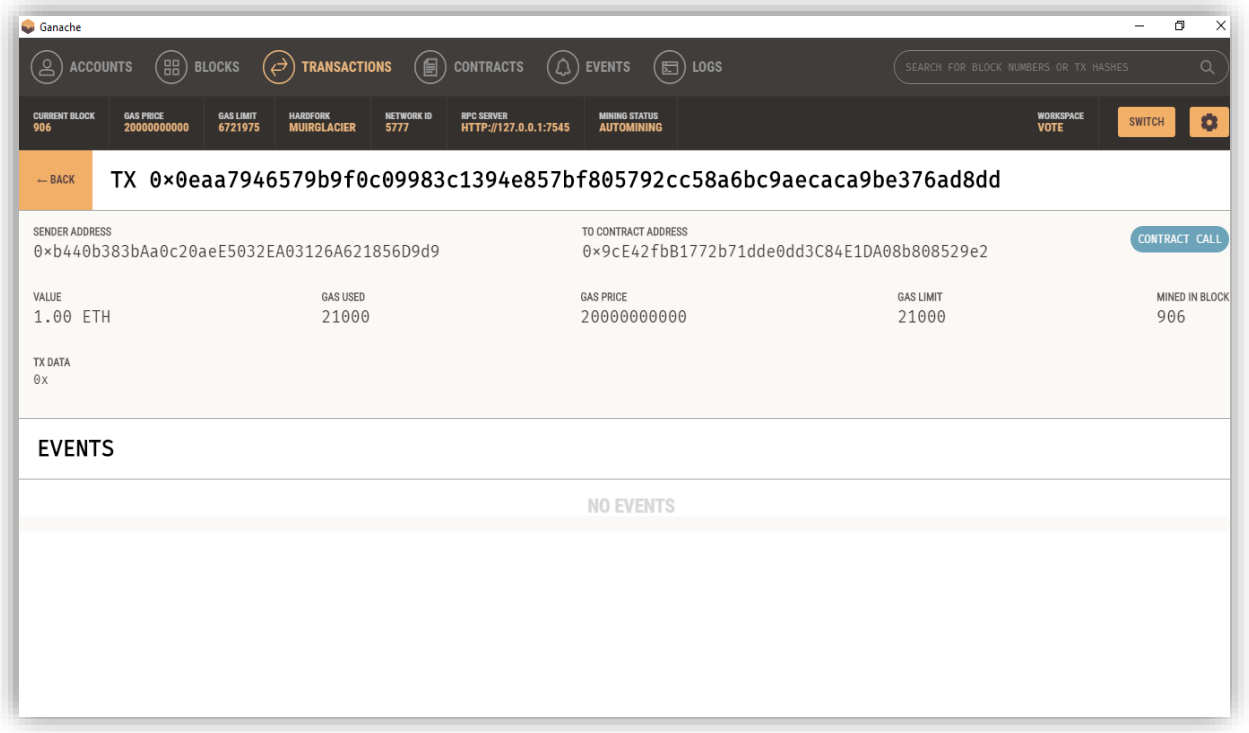


Figure III.37 transaction vers l'adresse de candidat

- **Remarque** : afin d'appelé la méthode de vérification définie sur le smart contract dans JavaScript nous avons utilisé des fonctionnalités Web3.js qui sont proposées par Moralis. [38]

d. Résultat :

Cette phase permet le votant de visualiser en temps réel le résultat actuel de vote par un graphe circulaire (**Figure III.38**), nous avons le réalisé en utilisant la fonctionnalité offerte par le service gratuit Google graphique [39]

Figure III.38 interface de résultat

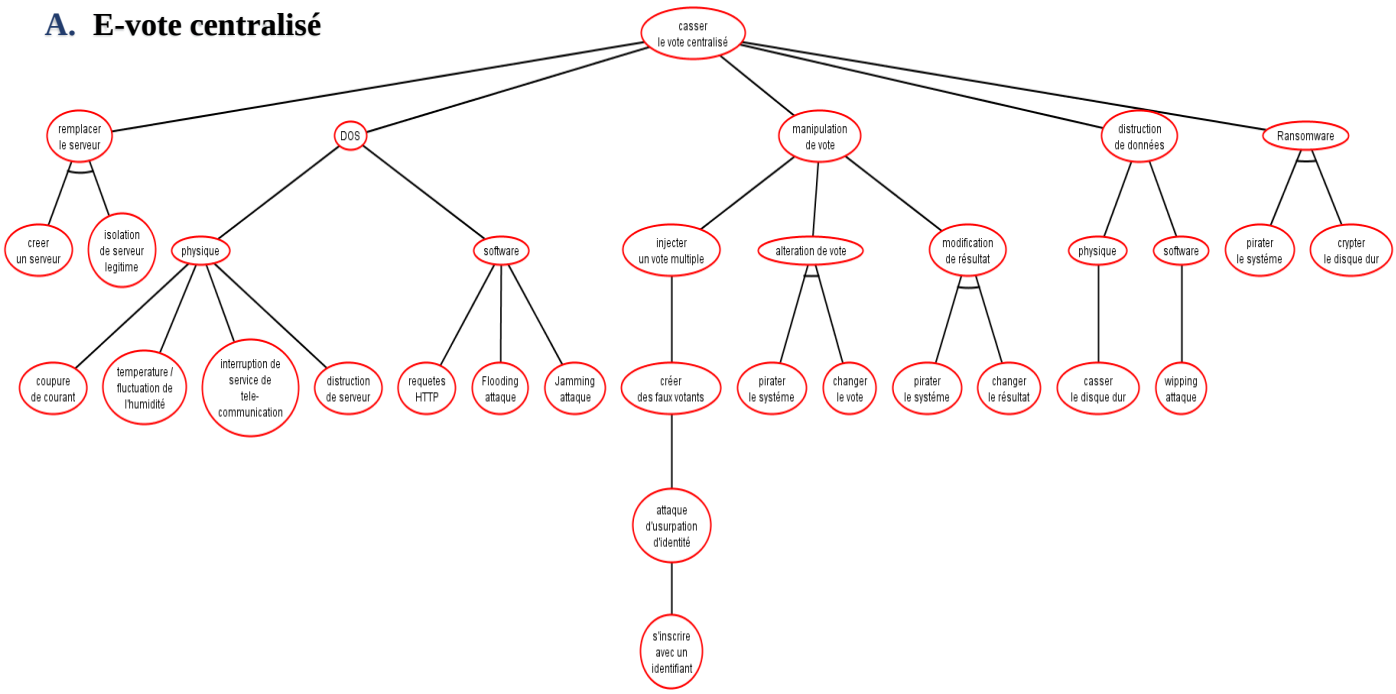
III.5 Évaluation de sécurité :

Chaque système informatique nécessite une sécurité contre des vulnérabilités et des attaques, la technologie de blockchain offre une structure de données avec des qualités de sécurité inhérentes, elle est basée sur les principes de cryptographie, de décentralisation et de consensus qui garantissent la confiance. Le niveau de sécurité dans les blockchains nous permet de sécuriser notre système de vote.

Pour bien évaluer la sécurité de notre solution proposée nous avons réalisé des arbres d'attaque (voir la **Figure III.39**) comparatifs pour un système de vote centralisé (**Figure III.39** A) et décentralisé (**Figure III.39** B) utilisant le logiciel ADTool [40], qui représentent les attaques possibles qui peuvent être exécutés afin de casser la sécurité de systèmes. La racine d'arbre s'agit de but final de l'attaquant et les descentes sont les vulnérabilités exploitées ou les moyens d'atteindre l'objectif principal.

Pour réaliser une attaque il faut prendre en compte plusieurs aspects dont la possibilité de réussir, nous avons calculé la probabilité totale d'atteindre l'objectif de l'attaque (racine d'arbre) en affectant aux feuilles les trois mesures suivantes : le cout d'attaque, la difficulté de détection et la difficulté d'exécution, le ***Tableau III.3*** indique les normes standard de chaque mesure.

A. E-vote centralisé



B. E-vote décentralisé

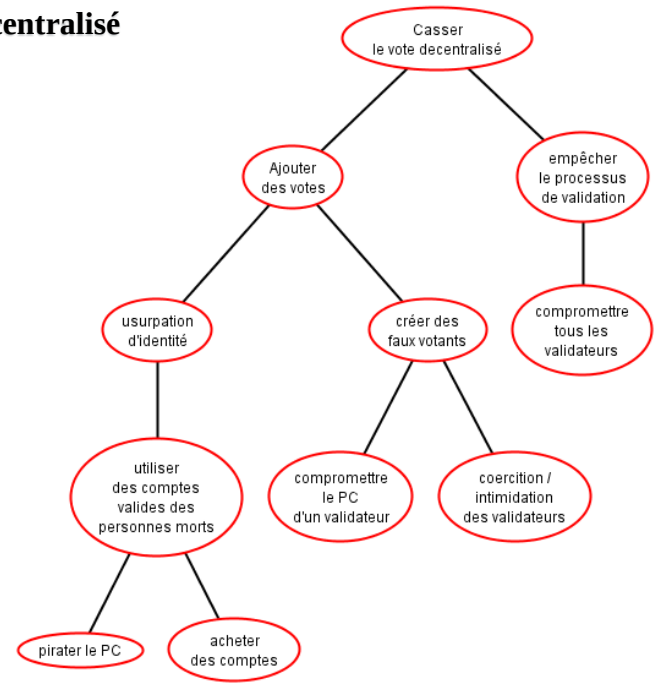


Figure III.39 arbre d'attaque de vote centralisé et décentralisé

Tableau III.3 les normes standard. *[41]*

Évaluation de cout d'attaque		*Difficulté de détection*		*Difficulté d'exécution*	
Grade	**Coût**	**Grade**	**Difficulté**	**Grade**	**Difficulté**
5	Assez cher	**5**	Assez Difficile	**5**	Assez Difficile
4	Cher	**4**	Difficile	**4**	Difficile
3	Coût Modéré	**3**	Médiate	**3**	Médiate
2	Pas cher	**2**	Simple	**2**	Simple
1	Pas cher du tout	**1**	Assez simple	**1**	Assez simple

Afin de calculer la probabilité de succès d'attaque, il faut trouver la probabilité d'occurrence de chaque attaque (feuille) (**Tableau III.4**) pour cela on a utilisé l'**Équation III.5**.

$$p = w(U(cout) + U(détection) + U(éxecution))$$

Équation III.6 probabilité d'occurrence.

Où w est le poids de chaque paramètre on l'a considéré 1/3 et U c'est l'utilité, elle est calculée avec la formule **Équation III.7**)suivante comme dans [42] :

$$U(f) = Cf/f$$

Équation III.8 fonction d'utilité

Où Cf = 0.2

Tableau III.4 Probabilité de succès d'attaque

Type de vote	*Attaque*	*Evaluation de cout d'attaque*	*Difficulté De Détection*	*Difficulté d'exécution*	*Probabilité d'occurrence*	*Probabilité de succès d'attaque*
Arbre d' attaque pour l' e-vote centralisé	Isolation de serveur légitime	5	4	5	0,043	***0,816***
	Créer un serveur	4	5	5	0,043	
	Coupure de courant	5	1	5	0,093	
	Température /fluctuation de l'humidité	5	1	5	0,093	
	Interruption de service de télécommunication	5	1	4	0,096	
	Destruction de serveur	5	1	5	0,093	
	DDOS	5	1	5	0,093	
	S'inscrire avec un identifiant	1	5	1	0,146	
	Pirater le système	5	5	4	0,043	
	Changer le vote	2	5	2	0 ,08	
	Changer le résultat	2	5	2	0,08	
	Casser le Disque Dur (DD)	5	1	5	0,093	
	Effacement de données (wiping)	5	1	5	0,093	
	Crypter le DD	5	1	5	0,093	
Arbre d' attaque de notre solution	Pirater le pc	2	5	3	0,068	***0,25***
	Acheter des comptes	5	5	3	0,048	
	Compromettre tous les validateurs	5	5	5	0,04	
	Coercition/Intimidation Des validateurs	5	5	3	0,048	
	Compromettre le pc d'un validateur	5	5	4	0,043	

Le calcul de probabilité totale de succès de l'attaque se fait par remplacer les opérateurs algébriques OR avec la somme des feuilles opérandes et les opérateurs AND sont remplacés par la multiplication entre les feuilles opérandes jusqu'à la racine.

Exemple :

$$p(casser\ le\ vote\ décentralisé) = p(ajouter\ des\ votes) + p(empecher\ le\ processus\ de\ validation)$$

Équation III.9 Probabilité de succés d'attaque

Après le calcul comparatif des probabilités de succès d'attaque de chaque système on a trouvé que la sécurité est plus élevée dans le système de vote décentralisé.

III.6 Etude comparative :

Auparavant nous avons mentionné les différentes phases de notre système et l'analyse de sécurité le tableau suivant **Tableau III.5** présente une étude comparative entre l'application de vote centralisée dans notre projet de licence et la solution décentralisée proposée :

Tableau III.5 étude comparative

Critère de comparaison	*Système de vote centralisé Réalisé en licence*	*Système de vote décentralisé Proposé*
Inscription des votants	- Par le votant - Date d'inscription illimitée - Obtenir un code vote	- Faite par l'autorité - Date d'inscription limitée - Obtenir une paire de clés certifiés
Login votant	- Code vote, numéro national	- Clé publique, nom et mot de passe (2 niveaux)
Login admin	- Nom et mot de passe	- Login de 3 niveaux (clé publique, nom et mot de passe, empreinte biométrique)
Vote	- Insertion de code vote. - Interdiction de double vote utilisant un booléen.	- Interdiction de double vote en utilisant image clé.
Interface de résultat	- Résultat Statique, à la fin - Nécessité de login	- Résultat dynamique, actuel (en temps réel)
Confidentialité de vote	- Administrateur peut voir les choix des votants	- Personne ne peut voir les choix de votants
Type d'applications	- Application mobile	- Application site web
Cryptographie	- Non	- Oui

III.7 Conclusion :

Dans ce chapitre, nous avons proposé l'implémentation de notre solution proposée qui est un smart contract et un site web coté client, où nous avons montré les détails d'implémentation pour la réalisation de notre prototype. Nous avons aussi fait l'analyse de sécurité de notre solution proposée avec une étude comparative entre notre système de vote centrale réalisé dans notre projet de fin d'étude de la licence et la proposition courante. Nous avons trouvé que le système de vote décentralisé est plus sécurisé que le système de vote centralisé puisqu'on a trouvé la probabilité de succès des attaques d'après un arbre d'attaque : pour le système centralisé 0,816 et pour le système décentralisé 0,25.

Enfin, nous résumons quelques limites que nous avons rencontrées pendant le développement de ce prototype :

- La transaction ne peut pas être envoyé directement à l'adresse candidat, car d'abord il faut être valider. La validation se fait par la vérification que le votant est éligible pour voter ce qui est réalisé par le smart contract. Donc, une transaction vers le smart contract doit être créée. Comme web3 ne permet pas d'envoyer une transaction vers 2 récepteurs à la fois, nous devions générer la deuxième transaction vers le candidat .
- La deuxième transaction sera publiée dans la blockchain si la première transaction est vérifiée et envoyé, notant que dans le cas où l'utilisateur a déjà voté aucune transaction sera effectuée .
- Dans l'implémentation, nous étions limités par MetaMask et nous ne pouvons pas changer l'adresse du votant par l'adresse du groupe. Aussi, nous n'avons pas pu arrêter ou modifier le processus de signature. Bien que l'identité de votant ne soit pas explicitement publiée dans la blockchain, mais, la liaison entre son vote et son adresse publique existe. Cette traçabilité peut permettre les candidats de savoir qu'a voté pour eux, s'ils vont utiliser d'autres techniques de social engineering, suivi à long terme, coercition ou menace des membres de l'autorité de certification qu'a généré ces adresses lors de l'enregistrement. Ces scénarios semblent difficiles à réaliser, cependant, ils ne sont pas impossibles. Malgré que notre conception protège contre eux, le prototype actuel ne fait pas et des améliorations futures sont nécessaires.

CONCLUSION GÉNÉRALE

La technologie Blockchain gagne en popularité de jour en jour. Elle a prouvé son efficacité dans le domaine de sécurité.

L'utilisation de la blockchain dans le système de vote aidera à réaliser des élections sûres et rentables tout en garantissant la confidentialité des électeurs. En adoptant la technologie de blockchain dans les systèmes de vote électronique , on peut réduire l'une des sources de tricherie de la manipulation des données.

Notre projet consiste à étudier la technologie de la blockchain et concevoir une solution de vote électronique basée sur cette technologie afin de garantir la confidentialité, la transparence, l'intégrité et l'immutabilité.

Pour réaliser notre application de vote en ligne basée sur la blockchain. Nous avons élaboré une conception et une modélisation basée sur L'UML, et spécifié les différents acteurs qui interagissent avec le système et nous avons analysé les besoins fonctionnels et non fonctionnels. Après, nous avons d´écrit le système à travers les diagrammes de cas d'utilisation, de séquence et de classe. L'implémentation a nécessité la manipulation des outils spécifiques à savoir la blockchain Ethereum et le développement d'un smart contract pour assurer les règles de sécurité et de vote. Notre solution permet de garantir les besoins essentiels d'un vote démocratique. Mais la technologie blockchain est encore très complexe pour réaliser la solution parfaitement.

Dans nos travaux futurs, nous poursuivrons la mise en œuvre et les améliorations de notre application décentralisée. Cependant, il existe encore des fonctionnalités que nous souhaitons les appliquer à titre des perspectives :

- Remplacer l'utilisation de l'outil Métamask en programmant un outil alternatif personnalisé pour répondre à nos besoins.
- Implémenter et utiliser PoA comme modèle de consensus
- Résoudre le problème de scalabilité.
- Réalisation de la politique de révocation des votants morts ou qu'ont perdu leur droits civils (droit de vote)

Auto-évaluation

Tâche/objectif	Statu	Details et remarques
☒: achevé. ☐: non-achevé		
Comprendre les Blockchains (principe de fonctionnement et types)	☒	
Comprendre les différents cas d'utilisation des blockchains hors la crypto-monnaie	☒	
Étudier et comprendre des notions de sécurité (cryptographie et attaques)	☒	
Étude des systèmes de votes existants	☒	
Conception d'un système de vote sécurisé, distribué, transparent, immuable et assurant la confidentialité de vote et des votants.	☒	
Installation et utilisation de la blockchain localement	☒	
Implémentation des smart contract de vote	☒	
L'apprentissage de Solidity pour la programmation des smart contract	☒	
L'apprentissage de l'utilisation de Web3js	☒	
Création des comptes pour les validateurs, les électeurs à partir de l'application web	☐	Ce prototype est initié avec 10 comptes entre validateurs, candidats et électeurs.
Utilisation de PoA	☐	Due à la limite de temps et le manque de documentation, nous avons remplacé l'utilisation de PoA par PoW dans ce prototype.
Comptage automatique de vote	☒	
Visualisation graphique des résultats de vote	☒	
Contrôle et validation des champs saisie	☒	
Réalisation de frontend (application web) de notre système à base blockchain	☒	
La réalisation de la tache de la gestion des comptes et de clés	☐	Nous avons utilisé Métamask
L'immuabilité et l'unicité des votes	☒	
La confidentialité de votant	☐	Partiellement assuré car cette version utilise le PoW et pas le PoA
L'étude de sécurité de notre solution proposée	☒	
L'apprentissage de l'utilisation des « attack trees » pour l'évaluation des systèmes de sécurité	☒	
Comparaison de notre système de vote à base blockchain avec notre système de vote centralisé proposé dans notre projet de fin d'étude (L3).	☒	

BIBLIOGRAPHIE

[1] «duval county,» [En ligne]. Available: https://www.duvalelections.com/General-Information/Learn-About-Elections/History-Of-Elections.

[2] «national archive,» national archive, [En ligne]. Available: https://www.archives.gov/milestone-documents/voting-rights-act#:~:text=This%20act%20was%20signed%20into,as%20a%20prerequisite%20to%20voting..

[3] . C. E, «Introduction à l'analyse de chimères technologiques, le cas du vote électronique,» *Cahiers Droit, Sciences & Technologies*, n° %13, pp. 261-280, 2010.

[4] L.David, «Verified Voting,» [En ligne]. Available: https://verifiedvoting.org/election-system/ess-ivotronic/. [Accès le 12 1 2022].

[5] «Techni-science.com,» Techni-science.com, 21 1 2021. [En ligne]. Available: https://www.techno-science.net/glossaire-definition/Vote-electronique.html.

[6] «academy binance,» [En ligne]. Available: https://academy.binance.com/fr/articles/history-of-blockchain.

[7] D. Mattias, «ippon positive technologie,» [En ligne]. Available: https://blog.ippon.fr/2018/01/08/fonctionnement-dune-blockchain/. [Accès le 23 1 2022].

[8] simplilearn, «simplilearn,» 8 11 2021. [En ligne]. Available: https://www.simplilearn.com/tutorials/blockchain-tutorial/merkle-tree-in-blockchain.

[9] «journalducoin,» journalducoin, [En ligne]. Available: https://journalducoin.com/lexique/blockchain-consensus-pow-pos-dpos/#:~:text=Le%20terme%20de%20%C2%AB%20consensus%20%C2%BB%20signifie,et%20automatique%20de%20son%20r%C3%A9seau..

[10] «pensezblockchain,» pensezblockchain, [En ligne]. Available: https://www.pensezblockchain.ca/la-preuve-dautorit#:~:text=Ce%20m%C3%A9canisme%20de%20consensus%20fonctionne,identit%C3%A9%20%C3%A9tant%20%C2%AB%20en%20jeu%20%C2%BB..

[11] . D. F, «tout-savoir-sur-la-technologie-block-chain,» 19 1 2022. [En ligne]. Available: https://financededemain.com/tout-savoir-sur-la-technologie-block-chain/.

[12] «journalducoin,» journalducoin, [En ligne]. Available: https://journalducoin.com/lexique/smart-contract/.

[13] «capital,» capitall, [En ligne]. Available: https://photo.capital.fr/bitcoin-a-10-ans-les-20-grandes-dates-de-son-histoire-31189#bitcoin-a-10-ans-les-20-grandes-dates-de-son-histoire-538309.

[14] «investopidia,» [En ligne]. Available: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp.

[15] «futura Tech,» [En ligne]. Available: https://www.futura-sciences.com/tech/definitions/cryptomonnaies-ethereum-18351/.

[16] «ethereum,» ethereum, [En ligne]. Available: https://ethereum.org/fr/developers/docs/consensus-mechanisms/pow/#:~:text=Ethereum%2C%20comme%20Bitcoin%2C%20utilise%20actuellement,certains%20types%20d'attaques%20%C3%A9conomiques..

[17] «hyperledger,» hyperledger, [En ligne]. Available: https://fr.hyperledger.org/.

[18] «stringfixer,» [En ligne]. Available: https://stringfixer.com/fr/Hyperledger.

[19] «hyperledger foundation,» [En ligne]. Available: https://www.hyperledger.org/blog/2019/02/13/introduction-to-sawtooth-pbft.

[20] F. Hao, S. Shahandashti et P. Mccory, «A smart contract for boardroom voting with maximum voter privacy.,» chez *In Proceedings of the International Conference on Financial Cryptography and Data Security*, Selima ,Malta, 2017.

[21] M. Chaib, M. Koscina, S. Yousfi, P. Lafourcade et R. Robbana, «DABSTERS: Distributed Authorities using Blind Signature To Effect Robust Security in e-voting,» *International Conference on Security and Cryptography (SECRYPT).*, 07 2019.

[22] u. Jafar, M. Aziz et z. Shukur, «Blockchain for Electronic Voting System-review and Open Research Challenges,» *Sensors*, vol. 21, n° %117, p. 5847.

[23] B. Shahrazad et J. Crowcroft, «Trustworthy Electronic Voting Using Adjusted,» *IEEE ACCESS*, vol. 7, pp. 24477-24488, 2019.

[24] H. Yi, «Securing e-voting based on blockchain in P2P network,» *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, n° %11, pp. 1-9, 2019.

[25] «monero blog,» [En ligne]. Available: https://blog.mymonero.com/what-is-a-key-image-f904614b3c23.

[26] «MONERO,» MONERO, [En ligne]. Available: https://monerodocs.org/cryptography/asymmetric/key-image/. [Accès le 12 3 2022].

[27] «truffle,» truffle, [En ligne]. Available: https://trufflesuite.com.

[28] «trufflesuite,» [En ligne]. Available: https://trufflesuite.com/ganache/.

[29] «nodejs,» [En ligne]. Available: https://nodejs.org/en/.

[30] «mycryptopedia,» mycryptopedia, [En ligne]. Available: https://www.mycryptopedia.com/what-is-web3-js-a-detailed-guide/.

[31] «google store,» [En ligne]. Available: https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=fr.

[32] «devloppez.com,» [En ligne]. Available: https://visualstudio.developpez.com/actu/308869/L-extension-Cplusplus-pour-Visual-Studio-Code-passe-en-version-1-0-et-apporte-un-riche-ensemble-de-fonctionnalites-de-productivite-adaptables-a-diverses-plateformes-et-architectures/#:~:text=Visual%20Studio%.

[33] «journal du coin,» [En ligne]. Available: https://journalducoin.com/ethereum/solidity-langage-ethereum/#:~:text=Fonctionnement%20de%20Solidity,d'ex%C3%A9cution%20d'Ethereum..

[34] «solidity,» [En ligne]. Available: https ://solidity.readthedocs.io/en/v0.6.11/ .

[35] «moralis,» moralis, [En ligne]. Available: https://docs.moralis.io/introduction/readme.

[36] «morioh,» morioh, [En ligne]. Available: https://morioh.com/p/2a2db4e8fdef.

[37] «github,» [En ligne]. Available: https://github.com/fatedier/frp/releases .

[38] «Moralis,» moralis, [En ligne]. Available: https://docs.moralis.io/moralis-dapp/web3/web3.

[39] «google charts,» [En ligne]. Available: https://developers.google.com/chart.

[40] «université du luxembourg,» [En ligne]. Available: https://satoss.uni.lu/members/piotr/adtool/.

[41] d. Ren, S. Du et H. Zhu, «A Novel Attack Tree Based Risk Assessment,» *IEEE International Conference on Communications (ICC)*, pp. 1-5, 2011.

[42] L. Benarous, B. Kadri et A. Bouridane, «Blockchain-based privacy-aware pseudonym management framework for vehicular networks.,» *Arabian Journal for Science and Engineering*, vol. 45, n° %18, pp. 6033-6049, 2020.