



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS – MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : Réseaux et Systèmes

THEME:

**Utilisation des Blockchain pour la gestion des votes au sein d'un
GDSS**

Présenté par :

FERRAOUN Asmaa

BEZZEKHAMI Aya

Encadrant:

Mr. Laredj Mohammed Adnane

Mr. Bessaaoud Karim

Année Universitaire 2024-2025

Résumé

Les Group Decision Support Systems (GDSS) jouent un rôle crucial dans la prise de décision collective en offrant des environnements collaboratifs qui structurent et facilitent les processus décisionnels au sein d'organisations, de communautés ou d'institutions publiques. Ces systèmes utilisent des mécanismes comme le vote pour agréger les choix individuels en décisions collectives, tout en intégrant des fonctionnalités telles que l'anonymisation, la communication facilitée et la réduction des biais. Cependant, la sécurité des processus de vote dans les GDSS reste un défi majeur, notamment en termes d'intégrité, de transparence et de protection contre les manipulations. Pour répondre à ces enjeux, cette recherche propose d'exploiter la technologie blockchain, dont les propriétés d'immutabilité, de traçabilité et de sécurité permettent de garantir un processus de vote fiable, transparent et anonyme, renforçant ainsi la qualité et la confiance dans les décisions collectives prises via les GDSS.

Abstract

Group Decision Support Systems (GDSS) play a crucial role in collective decision-making by providing collaborative environments that structure and facilitate decision processes within organizations, communities, or public institutions. These systems use mechanisms such as voting to aggregate individual preferences into collective decisions, while incorporating features such as anonymization, facilitated communication, and bias reduction. However, ensuring the security of voting processes in GDSS remains a major challenge, particularly in terms of integrity, transparency, and protection against manipulation. To address these challenges, this research proposes the use of blockchain technology, whose properties of immutability, traceability, and security can ensure a reliable, transparent, and secure voting process, thereby enhancing the quality and trust in collective decisions made through GDSS.

Remerciements

Avant tout, nous adressons notre gratitude la plus sincère à Allah le Tout-Puissant, pour nous avoir accordé la santé, la force, la patience et la persévérance nécessaires tout au long de ce travail. C'est par Sa miséricorde infinie et Sa guidance constante que nous avons pu surmonter les obstacles, relever les défis et mener ce projet à son terme. Sans Sa lumière pour éclairer notre chemin, rien de tout cela n'aurait été possible.

Nous exprimons également notre profonde reconnaissance à nos parents et à nos familles, piliers inébranlables de notre parcours. Leur amour inconditionnel, leur soutien moral indéfectible, leurs encouragements constants et leurs prières silencieuses mais puissantes nous ont donné la force d'aller de l'avant, même dans les moments les plus difficiles. Leur confiance et leur bienveillance nous ont toujours motivés à donner le meilleur de nous-mêmes.

Nos plus sincères remerciements vont aussi à nos encadrants, pour leur accompagnement précieux, leur disponibilité, leurs conseils éclairés et leur encadrement bienveillant. Leur expertise et leur patience ont grandement contribué à l'orientation et à la qualité de notre travail.

Nous ne pouvons pas oublier de mentionner nos véritables amis, qui ont été à nos côtés tout au long de ce parcours. Leur soutien moral, leurs encouragements et les moments de partage ont rendu cette expérience encore plus enrichissante.

Enfin, nous souhaitons remercier l'établissement et tous les enseignants que nous avons eu la chance de rencontrer durant notre parcours académique. Votre dévouement à l'enseignement et votre passion pour la transmission du savoir ont profondément marqué notre formation.

Table des figures

1.1	Structure d'un SIAD selon	14
1.2	Organigramme du processus de prise de décision collective	18
3.1	Chiffrement asymétrique	34
3.2	signature numérique	35
3.3	Structure d'un bloc	36
3.4	L'algorithme Proof of Work	37
3.5	Fonctionnement du blockchain	38
4.1	Diagramme de cas d'utilisation pour la gestion de vote dans un GDSS	44
4.2	Diagramme de séquence pour le processus de participation au vote	47
4.3	Diagramme de séquence pour le processus de participation au vote	48
4.4	Diagramme de classes du système de vote basé sur la blockchain	49
5.1	Smart Contract VotingRoles.sol	56
5.2	Smart contract Voting.sol 1	57
5.3	Smart contract Voting.sol 2	58
5.4	Smart contract Voting.sol3	58
5.5	Smart contract Voting.sol 4	59
5.6	Smart contract Voting.sol 5	59
5.7	Smart contract Voting.sol 6	60
5.8	Node.js et npm version	60
5.9	Truffle version	61
5.10	Script de migration du contrat votingRoles.sol	61
5.11	Script de migration du contrat voting.sol	62
5.12	Commande de migration	62
5.13	Interface de connexion	63
5.14	Interface de création de vote	63
5.15	Interface de création de vote	64
5.16	Interface de consultation des résultats	65

Liste des abréviations

GDSS	Group Decision Support System (Système d'Aide à la Décision de Groupe)
DSS	Decision Support System (Système d'Aide à la Décision)
SGBD	Système de Gestion de Bases de Données
SGBM	Système de Gestion de Bases de Modèles
SIAD	Système d'Aide à la Décision
IA	Intelligence Artificielle
IDSS	Intelligent Decision Support System (Système Intelligent d'Aide à la Décision)
NGT	Nominal Group Technique (Technique du Groupe Nominal)
IIA	Independence of Irrelevant Alternatives (Indépendance des Alternatives Non Pertinentes)
BS	Borda Score (Score de Borda)
PoW	Proof of Work (Preuve de Travail)
PoS	Proof of Stake (Preuve d'Enjeu)
DApp	Decentralized Application (Application Décentralisée)
EVM	Ethereum Virtual Machine (Machine Virtuelle Ethereum)
ABI	Application Binary Interface (Interface Binaire d'Application)
CLI	Command Line Interface (Interface en Ligne de Commande)
GUI	Graphical User Interface (Interface Graphique Utilisateur)
ETH	Ether (Cryptomonnaie d'Ethereum)

Table des matières

1	Les Systèmes d'Aide à la Décision de Groupe	11
1.1	Introduction	11
1.2	La décision et l'aide à la décision	11
1.2.1	La décision	11
1.2.2	L'aide à la décision	12
1.2.3	Processus de Décision	12
1.3	Les systèmes d'aide à la décision	13
1.3.1	Définition	13
1.3.2	Caractéristiques fondamentales d'un DSS	13
1.3.3	Structure d'un système d'aide à la décision	14
1.3.4	Applications des systèmes d'aide à la décision	15
1.3.5	Intelligence Artificielle et les systèmes d'aide à la décision	15
1.4	L'aide à la décision de groupe	15
1.4.1	Définition	16
1.4.2	Types de groupes	16
1.4.3	Composition de groupe	16
1.4.4	Le processus de prise de décision de groupe	17
1.5	Les systèmes d'aide à la décision de groupe (GDSS)	18
1.5.1	La typologie des GDSS	19
1.5.2	Les avantages du GDSS	19
1.6	Techniques utilisées dans la prise de décision de groupe	20
1.6.1	Le brainstorming	20
1.6.2	La technique du groupe nominal (NGT)	20
1.6.3	La méthode Delphi	20
1.6.4	La négociation	20
1.7	Le consensus	21
1.8	Conclusion	21
2	Les mécanismes de vote	22
2.1	Introduction	22
2.1.1	Les méthodes de vote	22
2.1.2	Classification des méthodes de vote	22
2.1.3	Evaluation des méthode de vote	25
2.2	Choix de méthode de vote	29
2.3	Conclusion	30

3 La blockchain	31
3.1 Introduction	31
3.2 Histoire	31
3.3 Définition	32
3.4 Caractéristiques de la Blockchain	32
3.5 Notion cryptographique	33
3.5.1 Les fonctions de hachage	33
3.5.2 La cryptographie asymétrique	34
3.5.3 La signature numérique	34
3.6 Structure de la blockchain	35
3.6.1 Bloc	35
3.6.2 Transaction	36
3.6.3 Consensus	36
3.7 Fonctionnement	38
3.8 Classification de la blockchain	38
3.8.1 Blockchain publique	38
3.8.2 Blockchain privée	39
3.8.3 Blockchain de consortium	39
3.9 Application de la blockchain	39
3.10 Conclusion	40
4 Conception et modélisation	41
4.1 Introduction	41
4.2 Systèmes de vote centralisés	41
4.3 Système proposé	42
4.4 Description fonctionnelle	43
4.5 Modélisation du système	44
4.5.1 Diagramme des cas d'utilisation	44
4.5.2 Diagramme de séquence	46
4.5.3 Diagramme de classe	48
4.6 Conclusion	50
5 Implémentation du système	51
5.1 Introduction	51
5.2 Concepts fondamentaux	51
5.2.1 Smart contract	51
5.2.2 Le web 3.0	51
5.2.3 Les applications décentralisées (DApps)	52
5.3 Environnement technique et outils utilisés	52
5.3.1 Choix de la plateforme blockchain	52
5.3.2 Langage de programmation	52
5.3.3 Frameworks et outils de développement	53
5.4 Conception et développement des smart contracts	55
5.4.1 Conception du contrat VotingRoles.sol	55
5.4.2 Conception du contrat Voting.sol	56
5.5 Déploiement de la DApp	60
5.5.1 Outils nécessaires pour le fonctionnement de la DApp	60
5.5.2 Déploiement des contrats	61

5.6 Développement de l'interface utilisateur (UI)	62
5.7 Conclusion	65

Introduction générale

Dans un monde où les décisions influencent profondément l'évolution des organisations, la gestion efficace des processus décisionnels est devenue un enjeu majeur pour les entreprises, les institutions publiques et les gouvernements. Face à des contextes de plus en plus complexes, marqués par une explosion des données et une diversification des parties prenantes, les décisions ne peuvent plus être prises de manière isolée ou sur la base d'intuitions subjectives. C'est dans ce cadre que se sont développés les Systèmes d'Aide à la Décision (DSS - Decision Support Systems), visant à fournir aux décideurs des outils capables d'analyser des données et de formuler des recommandations pertinentes.

Avec l'évolution des entreprises et la complexité croissante des enjeux, la prise de décision est devenue un processus collectif. Cette évolution vers une prise de décision collective a directement conduit à l'émergence des Systèmes d'Aide à la Décision de Groupe (GDSS), qui visent à soutenir et à optimiser ces dynamiques collaboratives en facilitant les échanges, la collaboration et l'analyse d'informations issues de multiples sources.

Un des aspects fondamentaux de ces systèmes repose sur la gestion des votes, un mécanisme essentiel pour structurer la participation et favoriser un consensus équilibré. Le vote, dans le cadre des GDSS, permet de formaliser les préférences individuelles tout en intégrant des méthodes d'agrégation pour refléter une décision collective cohérente. Cependant, malgré leurs nombreux avantages, les GDSS sont confrontés à plusieurs défis majeurs liés à ce processus, notamment en matière de sécurisation des votes, d'authentification des participants, de transparence des décisions et de gestion des conflits d'intérêts, nécessitant des solutions robustes pour garantir l'efficacité et la fiabilité du processus décisionnel collectif.

L'intégrité et la confidentialité des votes dans un processus de prise de décision collective sont des éléments clés qui conditionnent la fiabilité et l'acceptabilité des résultats obtenus. Or, dans un environnement centralisé classique, ces critères ne sont pas toujours garantis. Les bases de données conventionnelles peuvent être exposées à des manipulations frauduleuses, et les biais humains ou institutionnels peuvent altérer le bon déroulement des processus décisionnels. De plus, l'absence de traçabilité des actions entreprises par les différents acteurs peut poser un problème en cas de contestation des décisions prises.

C'est dans ce contexte que la blockchain apparaît comme une solution innovante et prometteuse. Technologie décentralisée et distribuée, elle repose sur un registre infalsifiable permettant d'enregistrer et de sécuriser des transactions de manière transparente et auditable. Ses propriétés de traçabilité, d'immuabilité et de résistance aux fraudes en font un candidat idéal pour améliorer la gestion des votes au sein des GDSS. Grâce à l'intégration des contrats intelligents

(smart contracts), il devient possible d'automatiser les processus de vote tout en garantissant un haut niveau de sécurité, de transparence et de fiabilité.

Ce mémoire est structuré en cinq chapitres :

Le premier chapitre est consacré à une analyse détaillée des Systèmes d'Aide à la Décision (DSS), des GDSS et de leurs principaux mécanismes de fonctionnement et de leur rôle dans la prise de décision collective.

Le deuxième chapitre analyse les différents types de systèmes de vote utilisés dans la prise de décision collective, leurs caractéristiques, ainsi que leurs limites.

Le troisième chapitre sera dédié à l'étude de la blockchain, en mettant en évidence son fonctionnement technique, ses différents types (publics, privés, hybrides), ses avantages et ses limites. Nous nous focaliserons sur ses applications potentielles dans les GDSS, en explorant comment elle peut renforcer la confiance et la sécurité des processus décisionnels.

Le quatrième chapitre proposera une approche combinée, où nous étudierons l'impact concret de l'intégration de la blockchain dans un GDSS dédié à la gestion des votes.

Le cinquième chapitre couvrira la partie implémentation technique. À travers une mise en œuvre concrète, nous illustrerons les concepts étudiés dans les chapitres précédents. Cette implémentation permettra de démontrer les bénéfices et les défis liés à l'utilisation de la blockchain dans un système de vote.

Chapitre 1

Les Systèmes d'Aide à la Décision de Groupe

1.1 Introduction

Dans un monde où la prise de décision collective est devenue cruciale face à la complexité croissante des problématiques organisationnelles, les Systèmes d'Aide à la Décision de Groupe (GDSS) émergent comme des outils essentiels pour structurer et faciliter les processus décisionnels impliquant plusieurs parties prenantes. Ces systèmes permettent non seulement de centraliser l'information, mais aussi d'encadrer efficacement les interactions et les choix des membres d'un groupe.

Ce chapitre s'inscrit dans le cadre de notre recherche sur l'utilisation de la blockchain pour la gestion des votes au sein d'un GDSS. Il a pour objectif de poser les bases conceptuelles nécessaires à la compréhension des systèmes d'aide à la décision, en particulier dans un contexte collaboratif. Nous présenterons les principes fondamentaux de l'aide à la décision, les caractéristiques des systèmes qui la soutiennent ainsi que les approches collectives qui permettent d'organiser les échanges entre participants et de soutenir les décisions de groupe.

1.2 La décision et l'aide à la décision

1.2.1 La décision

Dans [1], la décision est définie comme un choix entre plusieurs alternatives. Pour d'autres, la décision concerne aussi le processus de sélection de buts et d'alternatives. Les approches cognitives finalement traitent la décision comme le résultat d'un processus global de résolution de problèmes.

D'après [2], la décision ne se limite pas à un simple choix individuel entre plusieurs alternatives, mais résulte d'un processus collectif impliquant plusieurs acteurs aux intérêts parfois divergents. Elle est influencée par des interactions, des compromis et des arbitrages, et se construit à travers des temps forts où différentes options sont successivement arrêtées. Ainsi, la décision finale n'est que la synthèse des choix successifs faits au cours du processus.

Dans le même ordre d'idée, il est indiqué que la décision est une action prise pour faire face à une difficulté ou répondre à un changement d'environnement, c'est-à-dire résoudre un problème individuel ou organisationnel [3].

1.2.1.1 Types de décisions selon leur structure

Les processus de décision peuvent être classés en trois grandes catégories [4] :

Décision structurée :

- Problème clairement défini en termes techniques.
- Données fiables et quantitatives.
- Modèles précis permettant une résolution algorithmique.
- Exemples : facturation, achat, gestion comptable.

Décision peu structurée :

- Problème mal défini, données souvent qualitatives et instables.
- Accès limité aux informations.
- Résolution difficilement exprimable sous forme algorithmique.
- Majorité des décisions dans le cadre du pilotage des systèmes.

Décision non structurée :

- Problème non clairement posé.
- Rationalité limitée appliquée à toutes les étapes.
- Justification rationnelle difficile.
- Exemple : réaction imprévue d'un opérateur face à un aléa inédit.

1.2.2 L'aide à la décision

L'aide à la décision est l'activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponse aux questions que se pose un intervenant dans un processus de décision, éléments concourant à favoriser un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service desquels cet intervenant se trouve placé d'autre part [5].

L'aide à la décision est également définie comme l'activité visant à fournir des éléments de réponse à un intervenant dans un processus de décision, en s'appuyant sur des modèles clairement explicités, bien que non nécessairement entièrement formalisés. Son objectif est d'éclairer la décision, en facilitant la compréhension des différentes options possibles et en proposant des recommandations ou des orientations visant à assurer une cohérence entre l'évolution du processus décisionnel, les objectifs poursuivis et le système de valeurs de l'intervenant [2].

1.2.3 Processus de Décision

Herbert Alexander Simon [6], pionnier de la théorie de la décision, propose un modèle canonique en quatre phases non séquentielles :

- **Recherche d'information (Intelligence)** Cette phase consiste à identifier et collecter les **informations pertinentes** nécessaires à la prise de décision. Elle vise à cerner les **objectifs et priorités** du décideur et à analyser l'environnement pour mieux définir le problème à résoudre. La qualité des informations obtenues influence directement les étapes suivantes du processus.
- **Conception (Design)** À cette étape, le décideur élabore et analyse différentes **alternatives de décision**. Il sélectionne des **modèles adaptés** à la complexité du problème, identifie les critères d'évaluation, les variables de décision et établit les relations entre ces éléments afin de construire des solutions exploitables.

- **Choix (Choice)** Cette phase consiste à **évaluer les alternatives** définies précédemment et à sélectionner la solution la plus appropriée en fonction des critères retenus. Le choix peut être basé sur :
 - une **approche analytique**, où toutes les alternatives sont évaluées pour trouver la solution optimale ;
 - une **approche heuristique**, où une solution satisfaisante est choisie sans évaluation exhaustive.
- **Évaluation (Review)** Cette dernière phase permet de **valider ou d'ajuster la décision** en fonction des nouvelles informations disponibles. Une **rétroaction intelligente** améliore la qualité des décisions en apportant des corrections si nécessaire. Elle garantit une meilleure adaptation aux évolutions du contexte décisionnel et renforce l'efficacité du processus.

1.3 Les systèmes d'aide à la décision

1.3.1 Définition

Un Système d'Aide à la Décision (DSS), tel qu'initialement conceptualisé par Scott Morton dans les années 1970, est défini comme un système informatique interactif conçu pour aider les décideurs à exploiter des données et des modèles afin de résoudre des problèmes non structurés ou semi-structurés [7].

Le DSS combine les ressources intellectuelles des individus avec les capacités des ordinateurs pour améliorer la qualité des décisions. Il s'agit d'un outil informatisé destiné aux décideurs de gestion confrontés à des problèmes complexes qui ne peuvent être entièrement résolus par des méthodes traditionnelles [8].

Le DSS se distingue par sa capacité à combiner des données, des modèles analytiques sophistiqués et des interfaces utilisateur intuitives pour fournir un support décisionnel personnalisé [9]. Little souligne que pour être efficace, un DSS doit être **simple, robuste, facile à contrôler, adaptatif, complet sur les questions importantes, et facile à communiquer avec**. Ces caractéristiques permettent au DSS de s'intégrer harmonieusement dans le processus décisionnel du manager, en amplifiant ses capacités d'analyse sans remplacer son jugement.

1.3.2 Caractéristiques fondamentales d'un DSS

Les caractéristiques clés d'un DSS, telles que définies dans [9], sont les suivantes :

- **Simplicité** : Le modèle doit être facile à comprendre pour l'utilisateur, en incluant les phénomènes importants tout en évitant les détails superflus.
- **Robustesse** : Le modèle doit être conçu de manière à éviter des réponses absurdes, en contraignant les résultats à une plage de valeurs significative.
- **Facilité de contrôle** : Le manager doit pouvoir ajuster les entrées pour obtenir les sorties souhaitées, tout en restant maître du processus décisionnel.
- **Adaptabilité** : Le modèle doit pouvoir être mis à jour avec de nouvelles informations, que ce soit au niveau des paramètres ou de la structure.
- **Complétude sur les questions importantes** : Le modèle doit inclure les phénomènes critiques pour la prise de décision, même si cela nécessite des estimations subjectives.
- **Facilité de communication** : Le manager doit pouvoir interagir facilement avec le modèle, modifier les entrées et obtenir les résultats rapidement.

1.3.3 Structure d'un système d'aide à la décision

Il n'existe pas d'architecture standard pour un Système d'Aide à la Décision (SIAD). Le concept de SIAD englobe, selon les auteurs, un ensemble variable de composants, reflétant ainsi une certaine diversité dans sa définition et son application. En effet, de nombreuses architectures ont été proposées.

La figure 1.1 représente les composants fondamentaux d'un SIAD identifiés par Sprague et Carlson [10].

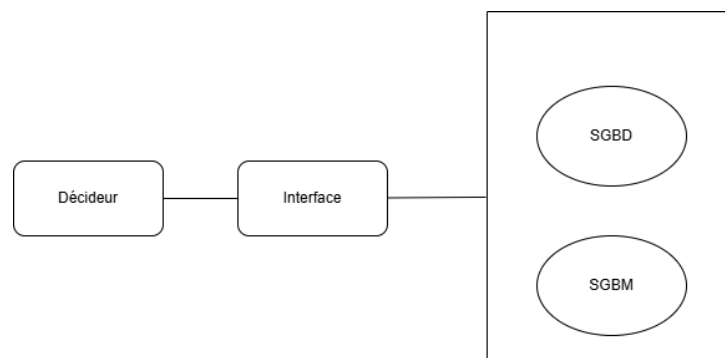


FIGURE 1.1 – Structure d'un SIAD selon

- **Le système de gestion et de génération de dialogue (interface utilisateur)**

Un système de gestion et de génération de dialogue (interface utilisateur) qui est un élément clé des fonctionnalités du système global. Il constitue l'interface entre l'utilisateur et le reste du système. Par l'intermédiaire des interfaces gérées par ce module, le décideur accède aux données et aux fonctions de calcul, et le système utilise le même vecteur pour lui communiquer le résultat des manipulations qu'il a effectuées [11].

- **Le Système de Gestion de Bases de Données (SGBD)**

Un Système de Gestion de Base de Données (SGBD) et sa base de données associée jouent un rôle central dans un Système d'Aide à la Décision (SIAD). Le module **Base de données** agit comme une mémoire fonctionnelle, assurant le stockage des données permanentes (statistiques, historiques) et temporaires (résultats intermédiaires issus des traitements). Il permet également la gestion dynamique des données volatiles et leur suppression selon les besoins de l'utilisateur.

Dans un SIAD, le système doit intégrer des données provenant de sources variées, faciliter leur recherche par des requêtes et des rapports, et supporter des opérations complexes de manipulation et de corrélation pour répondre aux exigences décisionnelles.

- **Système de Gestion de Bases de Modèles (SGBM)**

Un Système Gestionnaire de Base de Modèles (SGBM), ainsi que la base de modèles associée, joue un rôle similaire à celui d'un Système de Gestion de Base de Données (SGBD), mais il se concentre sur l'organisation, le stockage et la gestion des modèles quantitatifs utilisés par l'organisation.

Le module **Modèle de données** regroupe l'ensemble des modèles et procédures de calcul mis à disposition des utilisateurs pour les traitements standards des données. Ces outils incluent des modèles mathématiques, des techniques de prédiction et de simulation, des modèles financiers ainsi que des méthodes de calcul d'indices et d'indicateurs.

1.3.4 Applications des systèmes d'aide à la décision

Les Systèmes d'Aide à la Décision (DSS) sont principalement appliqués à des problèmes non-structurés et semi-structurés. Ils couvrent un large éventail de domaines et d'applications [12], notamment :

— **Planification financière d'entreprise :**

- Amortissement de prêts
- Dépréciation
- Choix entre location et achat.
- Analyse des flux de trésorerie actualisés et de la valeur actuelle nette
- Analyse du seuil de rentabilité

— **Analyse marketing :**

- Prévisions
- Analyse des ventes
- Analyse des promotions
- Audits de ventes aux consommateurs

— **Investissements immobiliers :**

- Alternatives de financement
- Flux de trésorerie
- Impact sur les taxes
- Retour sur investissement

— **Exploration minéralogique**

— **Routage des transports**

— **Analyse de portefeuille**

1.3.5 Intelligence Artificielle et les systèmes d'aide à la décision

La prise de décision constitue une activité fondamentalement humaine dont les répercussions peuvent être considérables. Ainsi, les chercheurs ont cherché à améliorer la qualité des décisions en développant des technologies informatiques visant à accroître et étendre les capacités humaines. Les progrès récents en intelligence artificielle (IA) ont permis d'atteindre cet objectif dans de nombreuses applications. L'intégration de l'IA dans les systèmes de soutien à la décision a donné naissance aux systèmes intelligents d'aide à la décision (Intelligent Decision Support Systems, IDSS) [13]. Les systèmes d'aide à la décision intelligents (IDSS) sont des systèmes interactifs basés sur ordinateur qui utilisent des données, des connaissances d'experts et des modèles pour soutenir les décideurs au sein des organisations dans la résolution de problèmes complexes, imprécis et mal structurés, en intégrant des techniques d'intelligence artificielle. L'inclusion des technologies d'intelligence artificielle (IA) dans les DSS représente un effort pour développer des systèmes informatiques capables de reproduire des qualités humaines telles que l'approximation, le raisonnement, l'intuition et le simple bon sens. L'utilisation des IDSS vise à améliorer la capacité des opérateurs et des décideurs à mieux accomplir leurs tâches et à collaborer efficacement [14].

1.4 L'aide à la décision de groupe

Les Systèmes d'Aide à la Décision (SIAD) ont initialement été développés pour des utilisateurs travaillant de manière autonome, en interaction avec des outils informatiques souvent so-

phistiqués, mais conçus comme des systèmes fermés, limités dans leurs capacités de communication externe. Cette conception encourage une approche individualiste de la prise de décision, où l'utilisateur est implicitement amené à endosser le rôle de décideur unique. Cependant, cette vision ne reflète pas la réalité organisationnelle, car les décisions sont presque toujours le fruit d'une réflexion collective impliquant un ou plusieurs groupes de décideurs. Cette dynamique collaborative devient encore plus prégnante lorsqu'il s'agit de décisions stratégiques, dont l'importance nécessite une analyse approfondie et une concertation entre plusieurs parties prenantes [15]. Face à ces limites des SIAD traditionnels, les Systèmes d'Aide à la Décision de Groupe (GDSS) ont émergé comme une réponse adaptée aux besoins des organisations modernes. Conçus pour favoriser la collaboration et la participation active de multiples acteurs, les GDSS permettent de structurer et d'enrichir les processus décisionnels collectifs, en intégrant des fonctionnalités spécifiques telles que la gestion des échanges, la capitalisation des idées et la synthèse des points de vue divergents

1.4.1 Définition

La prise de décision collaborative est définie comme une activité conduite par une entité collective composée de deux ou plusieurs individus et caractérisée à la fois en termes de propriétés de l'entité collective et de celles de ses membres individuels [16].

L'activité de la prise de décision collaborative est "une convergence d'interactions cognitives et visuelles, planifiées ou opportunistes, où des personnes acceptent de se rassembler pour un objectif commun, dans une période de temps définie, soit au même endroit, soit dans des endroits différents, dans le but de prendre des décisions" [17].

1.4.2 Types de groupes

Le travail de groupe peut être organisé selon trois modalités distinctes, telles que décrites dans [18] :

- **Groupe interactif** : Les membres collaborent directement, comme lors des réunions en face-à-face, où une seule personne peut s'exprimer à la fois. Ce mode favorise les synergies sociales, mais peut souffrir d'inefficacités liées aux conflits ou à la gestion des interactions.
- **Groupe nominal** : chaque membre travaille individuellement sur la même tâche, et un résultat est sélectionné pour représenter le groupe. Bien que ce modèle évite les effets indésirables du travail interactif (comme l'inhibition de production ou la peur du jugement), il ne profite pas des dynamiques sociales.
- **Équipe** : Le groupe est divisé en sous-équipes de petite taille (2 à 5 personnes) qui combinent travail autonome et interactions limitées. Cette approche réduit les inefficacités des grands groupes tout en tirant parti des synergies sociales, bien qu'elle reste exposée aux tensions interpersonnelles.

1.4.3 Composition de groupe

Dans un processus d'aide à la décision, différents acteurs participent à divers niveaux et rôles. L'analyse de ces acteurs (leurs objectifs, valeurs, interactions, etc.) est essentielle pour modéliser et concevoir efficacement le processus.

Parmi les acteurs, deux rôles se distinguent particulièrement : le facilitateur et les participants .

- **Le Facilitateur** : Le facilitateur est une figure centrale dans le processus de prise de décision de groupe. Accepté par tous les participants, il orchestre les différentes phases de la réunion, définit le problème à résoudre et structure le groupe de décideurs. À la fin de la session, il est responsable de la diffusion des résultats et veille à ce que le processus aboutisse à une décision finale.
- **Les Participants** : Les participants sont des experts qui contribuent activement à la session de prise de décision. Ils produisent et partagent diverses informations, telles que des idées, des commentaires, ou des analyses.

1.4.4 Le processus de prise de décision de groupe

Le processus de prise de décision de groupe fait référence aux étapes successives par lesquelles un groupe passe pour aboutir à une décision [17]. Ce processus inclut plusieurs phases clés :

1. **Cerner la problématique** : identifier et définir clairement le problème ou l'objectif à atteindre.
2. **Collecter des données** : rassembler les informations nécessaires pour éclairer la décision.
3. **Élaborer des règles de fonctionnement** : Établir un cadre contractuel entre les participants, garantissant leur engagement à accepter le résultat final.
4. **Évaluer les solutions proposées** : Analyser les options en fonction des points de vue retenus.
5. **Appliquer les règles convenues** : Utiliser les critères décidés pour sélectionner la meilleure solution.
6. **Vérifier l'adéquation des recommandations** : S'assurer que les décisions prises répondent aux besoins initiaux.

Les observations empiriques, confrontées à des études existantes, montrent que ces processus peuvent être modélisés de manière générique (voir la figure 1.2). Cette modélisation permet de mieux comprendre et structurer les interactions complexes inhérentes à la prise de décision collective.

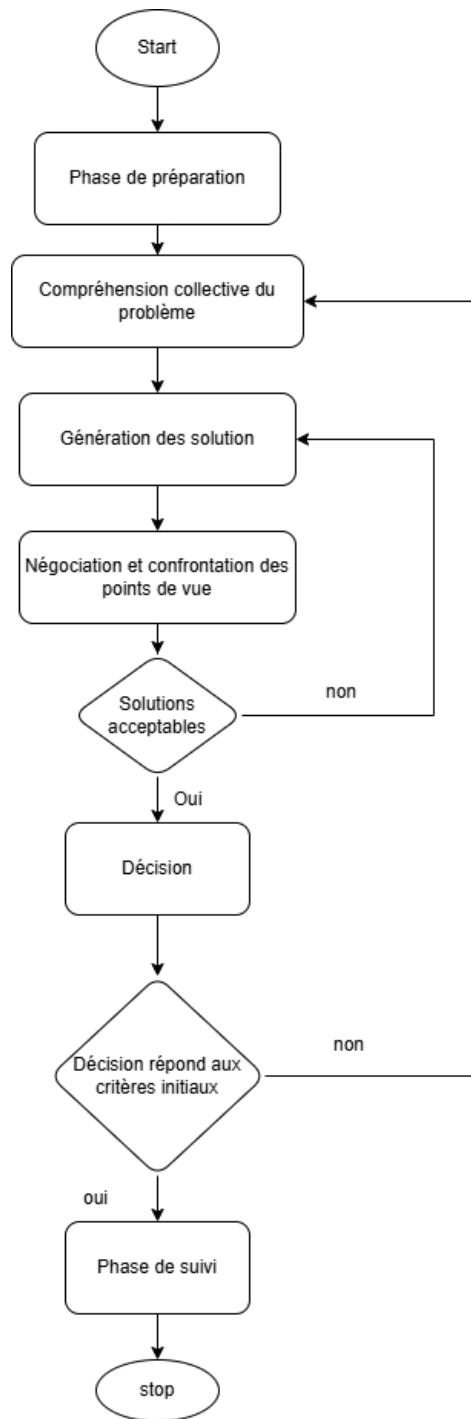


FIGURE 1.2 – Organigramme du processus de prise de décision collective

1.5 Les systèmes d'aide à la décision de groupe (GDSS)

Un GDSS est un système informatique interactif qui combine des technologies de communication, de calcul et de soutien à la décision. Il facilite la résolution de problèmes non structurés ou semi-structurés par un groupe de décideurs [19], en présentant des informations et des interprétations relatives aux différentes alternatives, tout en favorisant la communication entre les membres de l'équipe, indépendamment des limitations géographiques et des obstacles à la prise de décision collective. Ainsi, ce type de système peut aider le groupe à surmonter ces

défis.

Les Systèmes d'aide à la décision de Groupe (GDSS), qui sont étroitement liés aux DSS, facilitent la résolution de problèmes non structurés et semi-structurés par un groupe de décideurs travaillant ensemble telle une équipe [20].

Dans [21] Les GDSS sont définis comme "des environnements informatiques interactifs qui soutiennent les efforts concertés et coordonnés d'une équipe pour accomplir des tâches communes".

D'autre part, ils sont définis comme "une collectivité de technologies informatiques spécifiquement conçues pour soutenir les activités et processus liés à la prise de décision collective impliquant plusieurs participants" [22].

1.5.1 La typologie des GDSS

Une classification des Systèmes d'Aide à la Décision de Groupe (GDSS) en trois catégories est proposée, correspondant à différents niveaux d'évolution fonctionnelle [19] :

- **GDSS de premier niveau** : Ces systèmes ont pour objectif principal d'améliorer la communication entre les décideurs. Ils fournissent des outils permettant de faciliter les interactions et les échanges d'informations au sein du groupe, sans nécessairement intégrer des mécanismes avancés de modélisation ou d'agrégation des préférences.
- **GDSS de deuxième niveau** : Outre la facilitation des communications, ces systèmes incorporent des procédures destinées à modéliser et agréger les préférences individuelles afin de parvenir à un consensus. Dans ce cadre, un facilitateur est souvent impliqué pour guider et structurer le processus décisionnel, assurant ainsi une meilleure coordination des contributions des participants.
- **GDSS de troisième niveau** : Ces systèmes automatisent la structuration des échanges d'information et des communications grâce à l'intégration de recommandations générées par des systèmes experts. Ils visent à réduire la charge cognitive des participants en proposant des solutions intelligentes et en optimisant les dynamiques de groupe.

Cette classification met en lumière l'évolution progressive des GDSS, depuis des outils simples centrés sur la communication jusqu'à des systèmes plus sophistiqués capables de modéliser des préférences complexes et d'automatiser des aspects clés du processus décisionnel.

1.5.2 Les avantages du GDSS

Les GDSS présentent de nombreux avantages qui contribuent à améliorer les processus décisionnels collectifs. Parmi ses avantages :

- **Réduction des inefficacités** : Les GDSS minimisent les pertes de temps et les dérives organisationnelles des réunions traditionnelles.
- **Accélération de la prise de décision** : Grâce à des outils automatisés, ils facilitent la collecte et l'analyse des données pour une prise de décision plus rapide.
- **Amélioration de la coordination** : Ces systèmes permettent aux participants de collaborer efficacement, y compris à distance, tout en assurant une traçabilité complète des décisions prises.
- **Optimisation cognitive** : Les GDSS aident à mieux gérer l'information, réduisent les biais cognitifs et sociaux, et facilitent une agrégation rationnelle des préférences individuelles.
- **Bénéfices sociaux** : Ils favorisent une participation équitable, limitent les conflits interpersonnels et stimulent les synergies de groupe, renforçant ainsi la créativité et l'enga-

gement des membres.

1.6 Techniques utilisées dans la prise de décision de groupe

Les techniques de prise de décision en groupe constituent un domaine essentiel de l'amélioration des études commerciales. Ces méthodes de résolution collective des problèmes sont des outils essentiels utilisés dans les entreprises de toutes tailles pour trouver des solutions efficaces. La bonne technique peut rationaliser le processus de prise de décision, en particulier dans les grands groupes où la diversité des points de vue peut compliquer le processus.

1.6.1 Le brainstorming

Le brainstorming (ou remue-méninges) est une technique de créativité collective mise au point par Alex Osborn, visant à générer un grand nombre d'idées en un temps limité pour résoudre un problème ou explorer des solutions innovantes. Il se caractérise par une liberté d'expression sans jugement pendant la phase de proposition, permettant aux participants de partager spontanément leurs suggestions.

1.6.2 La technique du groupe nominal (NGT)

La technique du groupe nominal (Nominal Group Technique, NGT) est une méthode structurée de prise de décision collective conçue pour maximiser la participation individuelle tout en minimisant les biais sociaux et les dynamiques de pouvoir propres aux interactions en groupe. Développée dans les années 1960 [23], cette approche combine des phases d'introspection individuelle et d'échange structuré pour générer des idées, les prioriser et parvenir à un consensus. L'objectif principal de la NGT est de réduire les effets de la dominance verbale, de la conformité sociale ou de la pensée de groupe, tout en garantissant une représentation équitable des contributions de chaque membre. Cette méthode est particulièrement adaptée aux contextes où la diversité des perspectives et la neutralité du processus sont critiques.

1.6.3 La méthode Delphi

La méthode Delphi est une technique itérative et structurée de consultation d'experts visant à obtenir un consensus qualifié sur des questions complexes, incertaines ou controversées. Elle repose sur des cycles anonymes d'interrogation (questionnaires ou grilles d'évaluation) et une rétroaction contrôlée des réponses, permettant aux participants d'ajuster leurs avis à chaque itération. Le but final est de rassembler plusieurs avis d'experts sur un sujet précis, de mettre en évidence des convergences d'opinions et de dégager un éventuel consensus.

1.6.4 La négociation

La négociation peut être définie comme une discussion structurée visant à établir un accord entre des participants interdépendants dont les positions initiales peuvent être conflictuelles, en raison de perceptions divergentes de la problématique. Son objectif est de trouver un terrain d'entente, ou une "plate-bande", où les positions respectives des parties convergent. Ce processus repose sur trois étapes clés : (a) un échange bilatéral d'informations, (b) l'évaluation de ces informations par chaque partie selon son propre point de vue, et (c) l'obtention d'un

accord final par consentement mutuel. Dans ce cadre, la négociation implique deux éléments essentiels : la communication et la prise de décision [11].

1.7 Le consensus

Le terme consensus est souvent utilisé dans différents domaines tels que l'économie, la gestion et même les sciences politiques. Sa compréhension est essentielle pour naviguer dans des environnements où la coopération et la prise de décision collective sont cruciales.

Dans [24], le consensus est défini comme un « procédure qui consiste à dégager un accord sans procéder à un vote formel, ce qui évite de faire apparaître les objections et les abstentions. »

Dans un environnement collectif, le consensus est un accord général parmi les membres d'un groupe sur une décision ou une stratégie, même si ce n'est pas l'option préférée de chacun. Il permet de favoriser la coopération et réduire les conflits en gestion.

Il est important de distinguer entre un consensus et une décision unanime. Dans unanimité, tous les participants doivent être totalement d'accord avec une décision. En revanche, le consensus permet une certaine marge de désaccord, tant que l'on accepte la décision finale.

Dans le cadre d'un système d'aide à la décision en groupe (GDSS), le consensus représente donc un objectif central. Il s'agit de parvenir à une décision collective qui prend en compte les points de vue de tous les participants, même si certains ne sont pas entièrement satisfaits. Le but est de construire un accord suffisamment solide pour être accepté par l'ensemble du groupe, ce qui renforce l'efficacité du processus décisionnel et favorise l'engagement de chacun.

Cependant, dans certains cas, les GDSS n'exigent pas un consensus. Lorsque la rapidité est essentielle ou que les priorités divergent, le groupe peut opter directement pour un vote afin de prendre une décision efficacement. De même, lorsque parvenir à un consensus s'avère impossible en raison de désaccords persistants, le vote devient un mécanisme alternatif pour trancher et avancer. Ces approches, qui privilégient le vote, permettent de répondre aux contraintes de temps ou aux défis liés à la diversité des opinions, tout en maintenant un processus décisionnel structuré.

1.8 Conclusion

La prise de décision en groupe nécessite des outils fiables et structurés. Les GDSS jouent un rôle clé en facilitant la collaboration et en intégrant des méthodes de vote adaptées.

Dans ce chapitre, nous avons exploré les fondements théoriques des systèmes d'aide à la décision, en mettant l'accent sur leur évolution vers des dispositifs collaboratifs adaptés aux contextes complexes de prise de décision collective. Nous avons étudié les composantes des GDSS, leurs apports, ainsi que les approches favorisant les interactions entre les membres d'un groupe.

Cependant, malgré leur efficacité dans de nombreux contextes, les GDSS révèlent certaines limites lorsqu'il s'agit de gérer des situations où les préférences individuelles doivent être traduites en une décision collective. Cette problématique soulève des enjeux spécifiques que nous explorerons à travers l'étude des mécanismes qui permettent de formaliser et d'agréger les choix au sein d'un groupe.

Chapitre 2

Les mécanismes de vote

2.1 Introduction

La prise de décision collective s'appuie souvent sur des processus organisés pour transformer des avis individuels en un choix partagé. Parmi ces processus, les mécanismes de vote jouent un rôle clé, car ils permettent de regrouper les préférences et de parvenir à une décision équilibrée. Pour fonctionner correctement, ces mécanismes doivent être impartiaux, représenter fidèlement les choix des participants et garantir transparence et fiabilité, surtout dans des contextes complexes. Ces éléments montrent l'importance d'étudier en détail les principes et les défis liés aux systèmes de vote dans les environnements collaboratifs.

Ce chapitre vise à présenter de manière claire les différents mécanismes de vote. Nous commencerons par en présenter les principes fondamentaux, ainsi que les avantages et les limites qu'ils présentent. Ensuite, nous nous pencherons sur leur application dans des contextes collaboratifs. Cette analyse nous permettra de mieux comprendre comment ces mécanismes peuvent être adaptés aux besoins spécifiques des environnements de décision collective.

2.1.1 Les méthodes de vote

Les méthodes de vote peuvent être définies comme des procédures systématiques permettant de sélectionner un ou plusieurs gagnants parmi un ensemble d'alternatives, en prenant en compte les opinions de tous les membres d'un groupe. Elles jouent un rôle crucial dans la prise de décision collective, car elles fournissent un mécanisme structuré pour résoudre les désaccords et parvenir à une décision équitable, notamment dans des contextes où les opinions sont divergentes ou nombreuses [25].

Dans un environnement collectif, les mécanismes de vote constituent un ensemble de procédures formelles permettant d'agréger les préférences individuelles des membres d'un groupe en une décision collective. Ces mécanismes jouent un rôle central dans la structuration du processus décisionnel, ils visent à garantir l'équité, la transparence et la légitimité de la décision finale.

2.1.2 Classification des méthodes de vote

2.1.2.1 Les méthodes de vote ordonnées

Une méthode de vote ordonnée est un type de système de vote dans lequel les votants expriment leurs préférences sur un ensemble d'alternatives en les classant selon un ordre de

préférence. Ce classement permet d'exprimer non seulement leur choix préféré, mais aussi leurs préférences secondaires et ultérieures.

— **A. Le vote majoritaire**

Le vote majoritaire est une méthode de vote largement utilisée en raison de sa simplicité et de son intuitivité. Dans ce système, chaque individu dispose d'une voix et attribue cette voix à l'alternative qu'il préfère dans l'ensemble des choix. L'alternative qui reçoit le plus de votes est déclarée gagnante. En cas d'égalité entre plusieurs alternatives en première position, une procédure supplémentaire est mise en place pour déterminer le vainqueur final.

Bien que ce système soit simple, il ne garantit pas toujours la sélection de l'option la plus préférée comme vainqueur, s'il en existe un. Dans certains cas extrêmes, il peut même désigner un candidat moins préféré par la majorité comme vainqueur. Un problème bien connu associé à cette méthode est celui du « fractionnement de la majorité » : deux candidats populaires peuvent diviser les voix des électeurs, permettant ainsi à un troisième candidat, bénéficiant d'un soutien minoritaire, de remporter l'élection.

Pour pallier ces limites, plusieurs variantes de la majorité simple ont été proposées. Certaines exigent que l'alternative gagnante atteigne un certain seuil pour être déclarée vainqueur, ce seuil pouvant varier de la moitié des votes à un soutien total (unanimité). Les électeurs doivent voter de manière répétitive jusqu'à ce qu'une alternative dépasse ce seuil.

Une autre variante du vote majoritaire est le *vote majoritaire avec second tour*, dans lequel les électeurs choisissent un candidat au premier tour, et si aucun candidat n'obtient une majorité absolue (plus de 50 % des suffrages exprimés), un second tour est organisé. Au second tour, seuls les candidats remplissant certaines conditions, comme l'obtention d'un minimum de voix au premier tour, peuvent se maintenir. En général, ce sont les deux candidats ayant reçu le plus grand nombre de suffrages qui s'affrontent. Le candidat qui obtient alors la majorité des voix au second tour est déclaré élu [26].

— **B. La méthode borda**

La méthode de Borda, formalisée en 1770 par Jean-Charles de Borda [27], est un système électoral préférentiel qui prend en compte l'ensemble du classement des préférences individuelles pour déterminer le vainqueur. Dans cette méthode, chaque candidat se voit attribuer un score basé sur son rang dans les classements des électeurs. Pour n candidats, le score attribué est généralement défini comme suit :

- Le candidat classé premier reçoit $n - 1$ points,
- Le candidat classé deuxième reçoit $n - 2$ points,
- Et ainsi de suite, jusqu'à 0 point pour le dernier rang.

Le score total d'un candidat, appelé *Borda Score (BS)*, est calculé en sommant les points qu'il obtient de tous les électeurs. Le candidat avec le score total le plus élevé est déclaré vainqueur.

Mathématiquement, le score de Borda pour un candidat A peut être exprimé comme suit :

$$\begin{aligned} BS(A) = & (n - 1) \times \#\{i \mid i \text{ ranks } A \text{ first}\} \\ & + (n - 2) \times \#\{i \mid i \text{ ranks } A \text{ second}\} + \dots \\ & + 1 \times \#\{i \mid i \text{ ranks } A \text{ second to last}\} + 0 \times \#\{i \mid i \text{ ranks } A \text{ last}\} \end{aligned}$$

La méthode Borda est une forme de règle de notation (scoring rule), où des poids spécifiques sont attribués aux candidats en fonction de leur position dans les classements des électeurs. Par exemple, le vote majoritaire peut être vu comme une règle simplifiée attribuant 1 point au candidat classé premier et 0 point aux autres.

— C. Le vote Condorcet

Le vote de Condorcet trouve son origine dans les travaux du mathématicien Condorcet [28]. Son objectif principal est de concevoir une méthode permettant de dégager un choix collectif reflétant fidèlement les préférences individuelles des électeurs.

Cette méthode repose sur l'idée qu'un candidat doit être jugé en fonction de sa capacité à battre tous les autres candidats dans des confrontations deux à deux.

Dans le vote de Condorcet, chaque électeur classe les candidats par ordre de préférence. Ensuite, des "duels" sont simulés entre chaque paire de candidats pour déterminer quel candidat est préféré dans chaque confrontation. Le candidat qui remporte une majorité dans tous ses duels est désigné comme le vainqueur de Condorcet.

Cependant, cette méthode peut parfois conduire à des situations où aucun candidat n'est clairement préféré à tous les autres, ce qui soulève ce qu'on appelle le paradoxe de Condorcet.

• *Le paradoxe de Condorcet :*

Bien que la méthode de Condorcet soit conçue pour identifier un gagnant représentatif, elle peut être confrontée au **paradoxe de Condorcet**. Ce phénomène survient lorsque les préférences collectives forment un cycle, rendant impossible l'identification d'un gagnant clair. Par exemple, supposons que :

- Une majorité préfère A à B,
- Une majorité préfère B à C,
- Une majorité préfère également C à A.

Cela crée un cycle de préférences collectives :

$$A > B > C > A$$

Dans ce cas, aucun candidat ne peut être désigné comme gagnant de Condorcet, soulignant une limitation fondamentale des systèmes électoraux basés sur des comparaisons deux à deux.

2.1.2.2 Les méthodes de vote non ordonnées

Les méthodes de vote non ordonnées peuvent être définies comme des systèmes électoraux dans lesquels les électeurs n'expriment pas leurs préférences en classant les candidats ou les options par ordre de priorité. Au lieu de fournir un classement, les électeurs sélectionnent une option unique, attribuent des notes ou approuvent un ou plusieurs candidats sans hiérarchisation explicite

— A. Le vote par approbation

Le **vote par approbation** est un système électoral dans lequel chaque électeur peut approuver autant de candidats qu'il le souhaite. Concrètement, cela signifie qu'il sélectionne un sous-ensemble de candidats qu'il considère comme acceptables. Le ou les candidats ayant reçu le plus grand nombre d'approbations sont alors déclarés élus [25].

Si un électeur inclut un candidat X dans son bulletin, cela signifie qu'il l'approuve. À l'inverse, l'absence d'un candidat dans le bulletin indique une absence d'approbation.

Ce système offre une grande liberté aux électeurs, car ils ne sont pas contraints de choisir un seul candidat ou de classer leurs préférences. Par conséquent, la notion

de stratégie "sincère" devient indéterminée : voter uniquement pour le candidat le plus préféré est considéré comme sincère, mais voter également pour le deuxième choix peut également être perçu comme une expression sincère des préférences [29].

— B. Le vote par jugement majoritaire

Le jugement majoritaire repose sur une approche où les électeurs ne se contentent pas de classer ou de choisir un candidat, mais évaluent explicitement les mérites de chacun à l'aide d'une échelle commune d'évaluation. Cette échelle, appelée "langage commun", permet aux électeurs d'attribuer des mentions qualitatives telles que "Excellent", "Bien", "Passable", etc., à chaque candidat.

L'idée centrale est de donner plus de liberté aux électeurs en leur permettant non seulement de comparer ou de classer les candidats, mais aussi de les juger individuellement selon leurs performances perçues. Ce système corrige certaines lacunes des méthodes traditionnelles [30].

2.1.3 Evaluation des méthode de vote

L'évaluation des méthodes de vote repose sur leur capacité à transformer les préférences individuelles en une décision collective juste, stable et représentative. Pour cela, plusieurs critères ont été proposés par Kenneth J. Arrow [31] dans son célèbre théorème d'impossibilité, qui constitue une base théorique incontournable dans l'analyse des systèmes de vote.

Le théorème d'Arrow démontre qu'il est impossible de concevoir un système de vote par classement qui respecte tous ces critères à la fois, dès lors qu'il y a au moins trois alternatives. Ce résultat met en lumière les limites fondamentales de nombreux systèmes traditionnels, notamment ceux basés sur des classements ordonnés.

1. **Domaine non restreint** : Cette condition impose que toutes les configurations possibles de préférences individuelles soient prises en compte. Toutefois, des contraintes structurelles raisonnables, telles que l'unimodalité des préférences, peuvent permettre de contourner certaines limitations théoriques et assurer une décision collective cohérente.
2. **Non-dictature** : Cette condition est essentielle et difficilement relaxable. Toutefois, dans certains contextes, chaque individu peut exercer une forme de "dictature partielle" sur des décisions personnelles ou privées, comme l'illustre le paradoxe du libéral parétien développé par Amartya Sen. Cela montre que la tension entre liberté individuelle et cohérence collective demeure une problématique centrale.
3. **Unanimité (ou optimalité de Pareto)** : Cette condition stipule que si tous les électeurs préfèrent un candidat A à un candidat B, alors le système de vote doit obligatoirement classer A devant B dans le résultat final. Ce principe garantit qu'une préférence collective unanime soit toujours respectée par la méthode de scrutin.
4. **Indépendance des alternatives non pertinentes (IIA)** : Cette condition, souvent jugée peu réaliste dans les choix réels, est contournée dans de nombreux contextes pratiques. En effet, les préférences individuelles sont souvent influencées par la présentation des options (effet de cadrage), un phénomène bien connu dans le domaine du marketing. Certains chercheurs ont proposé des alternatives normatives plus souples dans des contextes économiques spécifiques.

2.1.3.1 Limites des méthodes de vote dans le contexte du théorème d'Arrow

En effet, les méthodes de vote présentent des limites significatives qui en restreignent l'efficacité dans l'agrégation des préférences individuelles.

1. Le vote majoritaire :

Ce système est vulnérable aux manipulations stratégiques, comme l'exclusion volontaire de candidats mineurs pour favoriser un candidat principal. Par ailleurs, il peut écarter un candidat qui serait préféré face à tous les autres en duel, illustrant son incapacité à respecter le principe d'indépendance des alternatives non pertinentes (IIA). Le résultat dépend souvent de la présence ou de l'absence de candidats supplémentaires, ce qui introduit des anomalies dans le processus décisionnel.

Exemple concret :

Considérons une élection avec 100 électeurs. Les préférences sont les suivantes :

Scénario 1 : Candidats A et B

- 60 électeurs : préfèrent A à B ($A > B$)
- 40 électeurs : préfèrent B à A ($B > A$)

Lors d'un vote majoritaire :

- A obtient 60 votes.
- B obtient 40 votes.

Le candidat A gagne.

Scénario 2 : Le candidat C entre dans l'élection.

Le candidat C est idéologiquement proche de A et va principalement attirer les votes des partisans de A.

Les préférences fondamentales des électeurs entre A et B n'ont pas changé. Cependant, certains partisans de A préfèrent maintenant C à A.

Les 60 électeurs qui soutenaient A à l'origine sont maintenant répartis :

- 35 électeurs : votent toujours pour A ($A > C > B$)
- 25 électeurs : votent maintenant pour C ($C > A > B$)

Les 40 électeurs qui soutenaient B à l'origine :

- 40 électeurs : votent toujours pour B ($B > A > C$ ou $B > C > A$)

Lors d'un vote majoritaire :

- A obtient 35 votes.
- B obtient 40 votes.
- C obtient 25 votes.

Le candidat B gagne.

Analyse et interprétation :

Dans le premier scénario, où seuls les candidats A et B sont en lice, le candidat A est clairement désigné vainqueur, bénéficiant d'une majorité absolue (60 voix contre 40).

Dans le deuxième scénario, l'introduction d'un troisième candidat, C, idéologiquement proche de A mais incapable de battre B en duel direct, modifie le résultat de l'élection. Bien que les préférences des électeurs entre A et B soient restées inchangées (60 préfèrent A à B, 40 préfèrent B à A), l'apparition de C, considéré comme une alternative « non pertinente », fragmente l'électorat favorable à A. Ce phénomène entraîne une répartition des voix en défaveur de A, conduisant à la victoire du candidat B.

Cette situation illustre une violation manifeste du **critère d'indépendance des alternatives non pertinentes (IIA)**, qui stipule que le choix collectif entre deux options ne devrait pas être influencé par la présence ou l'absence d'une troisième alternative. Il s'agit ici d'un exemple typique du **spoiler effect**, dans lequel une candidature additionnelle modifie l'issue de l'élection sans que les préférences relatives entre les candidats principaux ne soient altérées. Ces limites montrent qu'il est insuffisant pour des décisions complexes, justifiant l'intérêt pour des méthodes alternatives plus robustes.

2. La méthode Borda :

La méthode de Borda, bien qu'elle prend en compte l'ensemble des préférences individuelles, contrairement au vote majoritaire simple qui ne considère que le premier choix, elle est sensible à l'ajout ou à la suppression de candidats "paille" (candidats peu pertinents ou mineurs). Cela peut modifier significativement les résultats, violant ainsi le critère d'indépendance des alternatives non pertinentes.

Exemple :

Considérons une élection avec trois candidats : A, B et C, et 100 électeurs. Les points sont attribués selon la méthode de Borda :

- 2 points pour la première place
- 1 point pour la deuxième place
- 0 point pour la troisième place

Scénario 1 : Candidats A, B et C

- 40 électeurs : $A > B > C$ → A : 2 pts, B : 1 pt, C : 0 pt
- 35 électeurs : $B > C > A$ → B : 2 pts, C : 1 pt, A : 0 pt
- 25 électeurs : $C > A > B$ → C : 2 pts, A : 1 pt, B : 0 pt

Calcul des scores de Borda :

- Score de A : $(40 \times 2) + (35 \times 0) + (25 \times 1) = 80 + 0 + 25 = 105$
- Score de B : $(40 \times 1) + (35 \times 2) + (25 \times 0) = 40 + 70 + 0 = 110$
- Score de C : $(40 \times 0) + (35 \times 1) + (25 \times 2) = 0 + 35 + 50 = 85$

Le classement social est donc : **B (110) > A (105) > C (85)**.

Le gagnant est le candidat B, et la préférence sociale entre A et B est : **B > A**.

Scénario 2 : Le candidat C se retire

Il ne reste plus que les candidats A et B. Les préférences entre A et B restent identiques.

Nouveaux points attribués (à deux candidats) :

- 1 point pour la première place
- 0 point pour la deuxième place

Préférences restantes :

- 40 électeurs : $A > B$ → A : 1 pt, B : 0 pt
- 35 électeurs : $B > A$ → B : 1 pt, A : 0 pt
- 25 électeurs : $C > A > B$ → $A > B$ ⇒ A : 1 pt, B : 0 pt

Calcul des scores de Borda (avec seulement A et B) :

- Score de A : $40 + 0 + 25 = 65$
- Score de B : $0 + 35 + 0 = 35$

Le classement social devient : **A (65) > B (35)**.

Le gagnant est désormais le candidat A, et la préférence sociale devient : **A > B**.

Analyse et interprétation

Dans le **scénario 1**, avec la présence du candidat C, la méthode de Borda a déterminé la préférence sociale **B > A**.

Dans le **scénario 2**, lorsque le candidat C (qui n'était ni B ni A) a été retiré, la préférence sociale entre A et B s'est inversée en **A > B**.

Les préférences individuelles des électeurs concernant A par rapport à B n'ont pas changé. Le changement dans l'issue sociale entre A et B est uniquement dû à la suppression de l'alternative « non pertinente » C. Cela constitue une violation évidente du critère d'indépendance des alternatives non pertinentes (IIA).

Cette vulnérabilité rend la méthode de Borda susceptible à des manœuvres stratégiques de nomination ou de retrait de candidats, car la présence de certains candidats peut modifier l'issue de l'élection entre d'autres candidats.

En plus de violer le critère d'indépendance des alternatives non pertinentes, la méthode de Borda ne respecte pas non plus le critère de l'unanimité, tel que défini par le théorème d'Arrow. Ce critère stipule qu'un candidat préféré par une majorité absolue des électeurs devrait être élu. Cependant, la méthode de Borda peut ignorer cette majorité stricte en favorisant un candidat qui obtient de meilleurs scores globaux grâce à des classements intermédiaires.

Exemple :

Supposons une élection avec trois candidats : A, B et C. Les préférences des 100 électeurs sont réparties comme suit :

- 51 électeurs préfèrent : $A > B > C$
- 49 électeurs préfèrent : $B > C > A$

Le candidat A est préféré par une majorité absolue des électeurs (51 sur 100), ce qui devrait normalement lui assurer la victoire selon le critère de majorité stricte.

Cependant, lorsque la méthode Borda est appliquée, chaque candidat reçoit des points en fonction de son rang dans les préférences des électeurs :

- 3 points pour une première place,
- 2 points pour une deuxième place,
- 1 point pour une troisième place.

En calculant les scores, le candidat A obtient **202 points** ($51 \times 3 + 49 \times 1$), le candidat B obtient **249 points** ($49 \times 3 + 51 \times 2$) et le candidat C obtient **149 points** ($49 \times 2 + 51 \times 1$).

Bien que le candidat A soit préféré par une majorité absolue (51%), la méthode de Borda désigne le candidat B comme gagnant. Cela s'explique par le fait que la méthode Borda prend en compte tous les classements et permet à B de compenser son manque de première place par de bons scores en seconde position.

3. Le vote de Condorcet :

Le vote condorcet, bien qu'axé sur la comparaison deux à deux, souffre du paradoxe de Condorcet, où aucun vainqueur clair ne peut émerger en cas de cycles. Ces limitations illustrent l'impossibilité, soulignée par le théorème d'Arrow, de concevoir un système parfait capable de satisfaire simultanément toutes les conditions d'équité, de cohérence et de rationalité dans une décision collective.

2.2 Choix de méthode de vote

Le choix d'un mécanisme de vote approprié dans un contexte collaboratif dépend de plusieurs facteurs qui influencent la pertinence et l'efficacité du système. Ces facteurs incluent notamment la taille du groupe, la nature de la décision à prendre, le niveau de consensus souhaité, ainsi que les contraintes pratiques et organisationnelles.

La taille du groupe influence la complexité de mise en œuvre : des méthodes simples comme le vote majoritaire peuvent suffire pour de petits groupes, tandis que des méthodes plus raffinées comme Borda ou Condorcet sont mieux adaptées aux groupes plus larges. La nature de la décision joue également un rôle : des décisions sensibles ou à fort impact nécessitent souvent un système garantissant un haut niveau de légitimité. De plus, la complexité du système de vote doit être adaptée au public concerné. Des méthodes comme le jugement majoritaire ou la méthode de Borda nécessitent une compréhension préalable des règles, ce qui peut être un obstacle dans des contextes où les participants n'ont pas reçu de formation spécifique. Dans ces cas, des systèmes simples et intuitifs, comme le vote par approbation ou le vote majoritaire, sont souvent plus appropriés.

Dans le cadre de notre travail, nous avons choisi d'utiliser le **vote par approbation** comme mécanisme de vote privilégié. Ce choix repose sur une analyse des facteurs contextuels et des exigences spécifiques liées à notre environnement collaboratif. Le vote par approbation présente en effet plusieurs avantages qui répondent aux critères essentiels de *simplicité*, de *transparence* et d'*adaptabilité* à la nature des décisions que nous envisageons.

Tout d'abord, le vote par approbation est particulièrement bien adapté à des groupes de taille variable, car il permet à chaque participant d'approuver autant d'options ou de candidats qu'il juge acceptables, sans imposer de classement complexe. Cette flexibilité est cruciale dans des contextes où les préférences individuelles peuvent être variées ou nuancées, tout en restant simples à exprimer. Contrairement à des méthodes comme la méthode de Borda ou le jugement majoritaire, qui nécessitent une évaluation plus détaillée, le vote par approbation réduit la charge cognitive des participants, ce qui est un atout majeur pour garantir une adoption fluide du processus.

Ensuite, cette méthode est particulièrement robuste face aux *votes stratégiques*, un aspect important dans les environnements collaboratifs où la confiance et l'équité sont primordiales. En permettant aux électeurs d'approuver plusieurs options, elle limite les incitations à manipuler les résultats en favorisant une approche plus honnête et inclusive. De plus, le vote par approbation est moins sensible aux paradoxes classiques observés dans d'autres systèmes, comme le **paradoxe de Condorcet** ou les effets induits par l'ajout ou le retrait de candidats (*paradoxe des alternatives non pertinentes*).

Enfin, la transparence et la facilité de mise en œuvre du vote par approbation en font un choix judicieux pour des décisions collectives où la clarté du processus est essentielle. Les participants peuvent immédiatement comprendre comment leurs approbations influencent le résultat final, ce qui renforce leur confiance dans la légitimité de la décision. Cette simplicité est particulièrement avantageuse dans des contextes où les participants n'ont pas nécessairement une expertise en théorie du choix social ou en systèmes de vote complexes.

Pour ces raisons, nous considérons que le vote par approbation représente un **compromis optimal** entre simplicité, équité et efficacité pour notre travail. Il permet de recueillir des préférences individuelles de manière directe et inclusive, tout en minimisant les risques de distorsion ou de manipulation des résultats.

2.3 Conclusion

Les méthodes de vote jouent un rôle fondamental dans les processus de prise de décision collective. Qu'elles soient ordonnées, comme le vote majoritaire, la méthode Borda ou le vote de Condorcet, ou non ordonnées, ces méthodes visent à refléter de manière équitable les préférences individuelles au sein d'un groupe.

Toutefois, malgré leur diversité, les méthodes de vote rencontrent souvent des problèmes liés à la transparence, à la sécurité et à la fiabilité. Ces défis sont encore plus importants lorsque la confiance entre les participants est faible. Dans ce contexte, les technologies émergentes, notamment la blockchain, offrent de nouvelles perspectives. Par ses propriétés de décentralisation, d'immutabilité et de transparence, la blockchain constitue une piste prometteuse pour renforcer l'intégrité et la fiabilité des systèmes de vote.

Chapitre 3

La blockchain

3.1 Introduction

Dans les Systèmes d'Aide à la Décision de Groupe (GDSS), la sécurité du vote est une question importante. Lorsqu'un groupe de personnes doit prendre une décision ensemble, il est essentiel que chaque voix soit entendue de manière équitable et claire. Cependant, dans des contextes où les participants peuvent être nombreux ou géographiquement dispersés, garantir l'intégrité des votes devient un défi. C'est là que la blockchain peut jouer un rôle clé. Grâce à ses caractéristiques de décentralisation, d'immuabilité et de transparence, elle permet d'assurer que les votes sont sécurisés, traçables et inviolables, renforçant ainsi la confiance dans le processus décisionnel.

Ce chapitre propose une exploration détaillée de la blockchain, en retraçant son histoire, en présentant ses caractéristiques fondamentales, ses concepts cryptographiques essentiels ainsi que sa structure interne. Il aborde également les différents types de blockchains existants, en mettant en lumière leurs spécificités et leurs domaines d'application.

3.2 Histoire

La technologie de la blockchain représente une avancée majeure dans le domaine de l'enregistrement des transactions, offrant un paradigme sécurisé, transparent et immuable. Initialement popularisée par la cryptomonnaie Bitcoin, la blockchain a été véritablement révolutionnée en 2008, lorsque Satoshi Nakamoto a publié le célèbre whitepaper [32]. Ce document exposait un système de monnaie numérique basé sur la blockchain, soulignant comment les transactions pouvaient être effectuées directement entre utilisateurs, sans nécessiter d'intermédiaires.

L'idée fondamentale présentée par Nakamoto était de créer un réseau décentralisé où la sécurité des transactions est assurée par un protocole de consensus, rendant ainsi toute altération des données presque impossible. Cette innovation a non seulement permis le développement de Bitcoin, mais a aussi ouvert la voie à l'exploration de la blockchain dans de nombreux autres secteurs. La blockchain, initialement perçue comme une solution pour les cryptomonnaies, est désormais reconnue pour son potentiel à transformer de nombreux domaines, de la gestion des chaînes d'approvisionnement à la gouvernance décentralisée.

3.3 Définition

Littéralement, la blockchain désigne une chaîne de blocs, c'est-à-dire une succession de conteneurs numériques dans lesquels sont enregistrées des informations de nature variée, telles que des transactions, des contrats, des titres de propriété ou encore des œuvres numériques. Cette chaîne constitue une base de données distribuée, assimilable à un grand registre comptable partagé. Contrairement aux bases de données traditionnelles, la blockchain est décentralisée : elle n'est pas hébergée sur un serveur unique, mais est répliquée et maintenue par un ensemble de nœuds répartis dans le réseau. Les données inscrites dans les blocs sont protégées par des techniques cryptographiques avancées, garantissant leur intégrité et rendant toute modification a posteriori pratiquement impossible [33].

Selon le document technique du *National Institute of Standards and Technology (NIST)*, Les blockchains sont des registres numériques distribués contenant des transactions signées cryptographiquement, regroupées en blocs. Chaque bloc est lié cryptographiquement au précédent (ce qui rend toute altération visible) après validation et après avoir fait l'objet d'une décision de consensus. À mesure que de nouveaux blocs sont ajoutés, les anciens deviennent plus difficiles à modifier (créant une résistance à la falsification). Les nouveaux blocs sont répliqués à travers les copies du registre au sein du réseau, et tout conflit est automatiquement résolu en utilisant des règles établies [34].

La blockchain est une technologie de registre distribué prenant la forme d'une base de données transactionnelle répartie, sécurisée par des mécanismes cryptographiques et gouvernée par un protocole de consensus [35].

3.4 Caractéristiques de la Blockchain

La blockchain présente plusieurs caractéristiques fondamentales :

1. **Désintermédiation** : La désintermédiation dans la blockchain fait référence à la capacité de cette technologie à supprimer les intermédiaires dans divers processus transactionnels, organisationnels ou contractuels.

La blockchain permet des transactions peer-to-peer, sans nécessiter d'intermédiaires traditionnels tels que les banques, les courtiers ou les dépositaires. Cela réduit les délais et les coûts associés aux processus de gestion d'actifs, en éliminant les étapes supplémentaires nécessaires pour passer par des tiers [36].

2. **transparence** : La transparence est un avantage incontestable de la blockchain. Chaque transaction est enregistrée de manière permanente et vérifiable, ce qui permet à tout participant de suivre et de vérifier l'authenticité des transactions [37].

La transparence dans les blockchains fait référence à la visibilité et à l'accessibilité publique de toutes les transactions enregistrées sur une blockchain. Chaque nœud du réseau a accès aux mêmes informations, y compris les détails des transactions et leur historique complet. Cette transparence est un élément clé de la confiance numérique que la technologie blockchain vise à établir [38].

3. **Sécurité** : La sécurité de la blockchain repose sur sa décentralisation : les données sont copiées sur de nombreux serveurs, ce qui les rend très difficiles à supprimer ou à falsifier. Cette architecture la rend résistante aux cyberattaques et au contrôle centralisé, assurant ainsi un haut niveau de protection. Toutefois, cette même sécurité rend la technologie difficile à réguler [39].

La sécurité dans la blockchain est un concept multidimensionnel qui garantit l'intégrité, la confidentialité et la disponibilité des données grâce à une combinaison de techniques cryptographiques, d'algorithmes de consensus distribués et d'une structure en chaîne immuable [40].

3.5 Notion cryptographique

La cryptographie est la colonne vertébrale de la technologie blockchain. Elle regroupe un ensemble de techniques mathématiques destinées à sécuriser les données échangées, tout en garantissant leur confidentialité, leur intégrité et leur authenticité dans un environnement décentralisé. Au sein d'un réseau blockchain, la cryptographie permet de chiffrer les informations, transformant les données lisibles en un format codé qui ne peut être déchiffré que par ceux qui possèdent la clé correspondante. Elle repose principalement sur deux types de chiffrement : le chiffrement symétrique, où une seule clé est utilisée pour chiffrer et déchiffrer les données, et le chiffrement asymétrique, qui repose sur une paire de clés (publique et privée) offrant une sécurité renforcée. Cette dernière méthode est particulièrement utilisée pour la signature numérique des transactions, permettant ainsi d'authentifier les expéditeurs tout en empêchant toute falsification ou usurpation d'identité.

La cryptographie assure également l'intégrité des blocs de données via des fonctions de hachage cryptographique. Chaque bloc contient l'empreinte du bloc précédent, créant une chaîne inviolable : toute modification dans un bloc entraînerait la rupture de cette chaîne, immédiatement détectable par l'ensemble du réseau. En plus de la sécurité des transactions, la cryptographie joue un rôle central dans les mécanismes de consensus, tels que la Preuve de Travail (Proof of Work) et la Preuve d'Enjeu (Proof of Stake). Ces systèmes, basés sur des puzzles et des algorithmes cryptographiques, permettent de valider les transactions, d'ajouter de nouveaux blocs à la chaîne, et de maintenir la sécurité du réseau sans faire appel à une autorité centrale.

3.5.1 Les fonctions de hachage

Un composant essentiel de la technologie blockchain est l'utilisation des fonctions de hachage cryptographiques dans de nombreuses opérations. Elles sont au cœur de la structure des blocs, assurant la liaison sécurisée entre eux, ce qui garantit l'intégrité et l'immuabilité des données enregistrées. Chaque bloc contient le haché du bloc précédent, créant ainsi une chaîne inviolable. Toute tentative de modification d'un bloc changerait son haché, rendant immédiatement cette altération détectable par le réseau.

Définition : Un algorithme de hachage est un algorithme déterministe et efficace, qui prend comme entrée une donnée (un message) binaire M de taille quelconque et produit à la sortie un haché h de taille fixe (en général entre 128 et 512 bits). Cet haché, appelé aussi "condensé" ou "empreinte", doit dépendre de tous les bits du message, et il est utilisé comme représentant comprimé de celui-ci [41]. L'algorithme de hachage H peut être formulé comme suit :

$$H: \{0, 1\}^* \rightarrow \{0, 1\}^n$$
$$m \mapsto H(m)$$

où $\{0, 1\}^*$ représente l'ensemble des messages binaires de taille quelconque et $\{0, 1\}^n$ l'ensemble des hachés de taille fixe.

Caractéristiques d’une fonction de hachage : Une fonction de hachage cryptographique est conçue pour satisfaire plusieurs propriétés fondamentales qui assurent sa sécurité et son efficacité [35] :

- **Facilité de calcul :** Pour toute entrée donnée, le calcul de son empreinte doit être rapide et efficace, même pour des données volumineuses.
- **Résistance à la préimage (preimage resistance) :** Il doit être computationnellement infaisable, à partir d’une empreinte h donnée, de retrouver le message m tel que $H(m) = h$. Cela garantit le caractère unidirectionnel de la fonction.
- **Résistance aux collisions (collision resistance) :** Il doit être presque impossible de trouver deux messages différents m_1 et m_2 tels que $H(m_1) = H(m_2)$. Cette propriété est essentielle pour garantir l’unicité des empreintes et l’intégrité des données.

3.5.2 La cryptographie asymétrique

La technologie blockchain utilise la cryptographie asymétrique (également appelée cryptographie à clé publique). Les algorithmes à clé asymétrique, également appelés algorithmes à clé publique, reposent sur l’utilisation d’une paire de clés mathématiquement liées : une clé publique et une clé privée. La clé publique peut être librement diffusée sans compromettre la sécurité du système, tandis que la clé privée doit rester strictement confidentielle afin de garantir la protection cryptographique des données. (voir la figure 3.1)

Bien que les deux clés soient reliées par une relation mathématique, il est computationnellement infaisable de déduire la clé privée à partir de la clé publique. Cette dissymétrie constitue le fondement de la sécurité de ces algorithmes, largement utilisés dans la blockchain pour l’authentification, la confidentialité et la signature numérique des transactions.

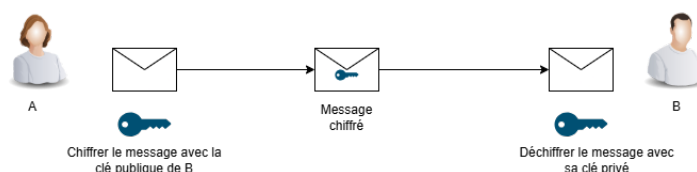


FIGURE 3.1 – Chiffrement asymétrique

3.5.3 La signature numérique

Une des applications les plus importantes des fonctions de hachage est leur utilisation dans les signatures électroniques. La signature électronique est un mécanisme analogue de la signature manuscrite permettant de garantir l’intégrité d’un document électronique et prouvant au lecteur du document l’identité de son auteur. De tels mécanismes utilisent les procédés de la cryptographie asymétrique [41].

Un système de signature numérique repose généralement sur deux composantes : un algorithme de signature et un algorithme de vérification. L’algorithme de signature est utilisé pour générer une signature numérique à partir d’un message donné. Cette signature est créée à l’aide d’une clé de signature, qui est gardée secrète et exclusivement contrôlée par le signataire.

L’algorithme de vérification, quant à lui, permet de vérifier l’authenticité de la signature attachée à un message. Il utilise une clé de vérification, rendue publique, ce qui permet à tout tiers d’authentifier facilement la signature sans compromettre la sécurité du système [42].

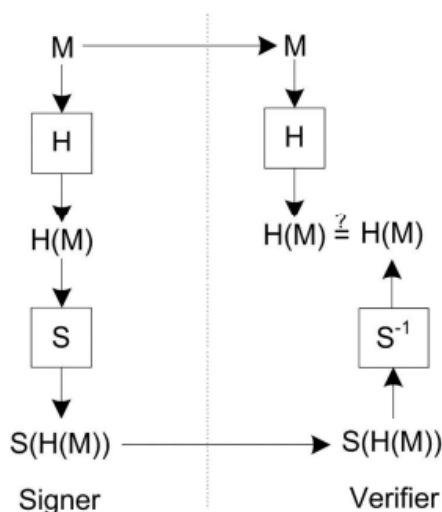


FIGURE 3.2 – signature numérique

3.6 Structure de la blockchain

La blockchain est un registre distribué (distributed ledger) qui permet de stocker de manière sécurisée et transparente l'ensemble des transactions effectuées au sein du réseau. Chaque transaction validée est regroupée avec d'autres au sein d'un bloc. Ces blocs sont ensuite liés entre eux de façon chronologique et immuable, formant ainsi une chaîne continue depuis le bloc initial. Les principaux composants de la blockchain sont :

3.6.1 Bloc

Une blockchain est un registre décentralisé et distribué qui enregistre des données transactionnelles à travers un réseau d'ordinateurs. Elle est composée d'une chaîne continue de blocs, chacun contenant des enregistrements de transactions.

Chaque bloc de la blockchain se compose de deux parties principales tel qu'indiqué dans la Figure 3.3 :

- **Données du Bloc** : Cette section contient les enregistrements de transactions.
- **En-tête du bloc** : L'en-tête du bloc contient des métadonnées essentielles qui assurent l'intégrité du bloc et sa place dans la chaîne. Il comprend généralement :
 - **Numéro de bloc** : identifie de manière unique chaque bloc au sein de la chaîne. Il sert à organiser les blocs dans un ordre séquentiel et permet aux nœuds du réseau de référencer facilement un bloc spécifique.
 - **Hash du bloc précédent** : Le hash de l'en-tête du bloc précédent, créant ainsi un lien entre les blocs.
 - **Horodatage** : l'heure à laquelle le bloc a été créé.
- **Hash du bloc** : est une empreinte cryptographique unique générée à partir de toutes les données contenues dans le bloc actuel.

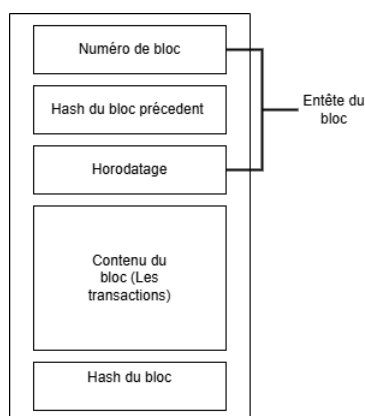


FIGURE 3.3 – Structure d'un bloc

3.6.2 Transaction

Une transaction dans la blockchain représente un échange d'actifs ou d'informations entre deux parties. Elle contient des données telles que l'expéditeur, le destinataire, le montant échangé et une signature numérique permettant de vérifier son authenticité. Elle permet ainsi d'enregistrer les détails d'une activité ayant eu lieu sur la blockchain.

3.6.3 Consensus

Un consensus est un accord général positif et unanime au sein d'un groupe de personnes permettant de décider ou d'agir ensemble sans vote préalable ou délibération particulière [43].

Dans les blockchains, le consensus est un élément fondamental pour leur fonctionnement. Il s'agit d'un ensemble de règles et de mécanismes permettant aux nœuds d'un réseau de s'accorder sur un état unique et valide de la chaîne, sans avoir besoin d'une autorité centrale de confiance. Chaque fois qu'un nouveau bloc est proposé, le mécanisme de consensus permet de déterminer si ce bloc doit être ajouté à la blockchain, garantissant ainsi l'intégrité, la cohérence et la continuité du registre distribué [34].

Dans les premiers systèmes blockchain, des algorithmes de consensus comme Proof of Work (PoW) et Proof of Stake (PoS) ont été utilisés. Cependant, un grand nombre de nouveaux algorithmes de consensus ont émergé récemment. Ces algorithmes visent à assurer que seul un bloc valide soit ajouté, et que l'ensemble du réseau partage la même version de l'historique des transactions [44].

1. **Preuve de travail (Proof of Work) :** est le premier algorithme de consensus introduit dans les réseaux blockchain. L'idée de la Preuve de Travail a été initialement proposée par Cynthia Dwork et Moni Naor [45], puis Nakamoto [32] l'a appliquée dans le livre blanc du Bitcoin en 2008.

Le PoW fonctionne en obligeant les nœuds du réseau à résoudre un problème mathématique complexe pour pouvoir créer le prochain bloc et vérifier la légitimité des transactions. Cette résolution repose sur l'utilisation de fonctions de hachage, qui génèrent des résultats aléatoires à partir de formules mathématiques complexes. Comme illustré dans la Figure 3.4, tous les nœuds du réseau entrent en compétition pour trouver la solution par force brute, nécessitant un très grand nombre de tentatives. Le premier nœud à trouver la solution obtient le droit de créer un nouveau bloc, qui sera ensuite vérifié et ajouté à la blockchain. Les nœuds qui participent à ce processus sont appelés mineurs, et l'ensemble du processus est connu sous le nom de minage [44].

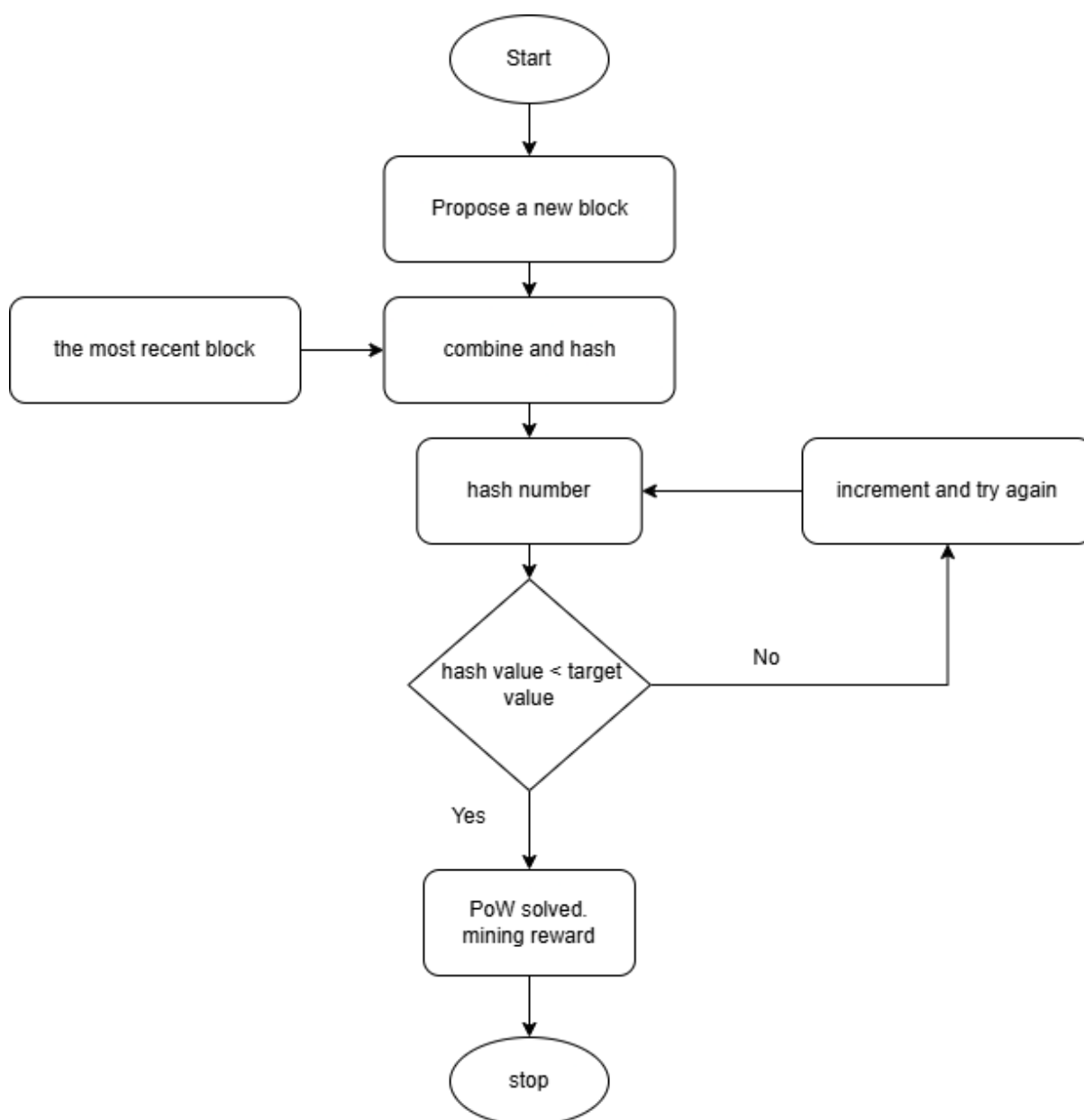


FIGURE 3.4 – L’algorithme Proof of Work

Cependant, malgré son efficacité à garantir la sécurité du réseau, le mécanisme de Proof of Work (PoW) présente plusieurs inconvénients majeurs. Le problème principal réside dans sa consommation énergétique extrêmement élevée. En effet, pour soutenir les opérations de minage, les nœuds doivent utiliser des équipements de plus en plus puissants, ce qui entraîne une augmentation exponentielle de la demande énergétique. De plus, cette course à la puissance de calcul favorise la centralisation du minage : les petits mineurs, incapables de rivaliser avec les grandes fermes de minage mieux équipées, sont progressivement exclus du réseau. Cela va à l’encontre du principe de décentralisation initialement défendu par les blockchains publiques [46].

2. **Proof of Stake (PoS) :** Pour résoudre le problème de la forte consommation de puissance de calcul du Proof of Work (PoW), les chercheurs ont proposé l’algorithme de consensus Proof of Stake (PoS). Le processus du PoS est différent de celui du PoW, car les utilisateurs du protocole PoS ne sont pas obligés de résoudre un problème

mathématique pour atteindre un consensus [44].

Contrairement au PoW, où les nœuds sont en compétition pour résoudre un problème mathématique, dans le PoS, un nœud est sélectionné par tirage au sort pour créer le prochain bloc, en fonction de son stake dans le réseau, c'est-à-dire la quantité de tokens qu'il possède. Le nœud choisi utilise une signature numérique pour prouver la possession de ses tokens, plutôt que de résoudre un problème de hachage complexe, ce qui réduit considérablement la consommation de ressources. Dans ce modèle, tous les tokens sont disponibles dès le début, et les validateurs sont récompensés uniquement par des frais de transaction. Cependant, bien que le PoS élimine les lourdes exigences computationnelles du PoW, il crée de nouveaux problèmes, tels que la centralisation autour des détenteurs des plus grands stakes [47].

3.7 Fonctionnement

La blockchain est une structure de données distribuée où les transactions sont enregistrées de manière permanente au sein de blocs. Chaque bloc est lié cryptographiquement au précédent par un identifiant unique appelé hash, formant ainsi une chaîne continue de blocs.

Avant d'être ajoutées à la blockchain, les transactions doivent être validées selon un mécanisme de consensus impliquant les nœuds du réseau. Ce processus assure l'intégrité des données, empêchant toute modification ou suppression ultérieure sans détection.

Grâce à ses propriétés d'immuabilité, de transparence et de sécurité, la blockchain permet à plusieurs acteurs de collaborer en toute confiance, sans nécessiter d'intermédiaire de confiance centralisé.

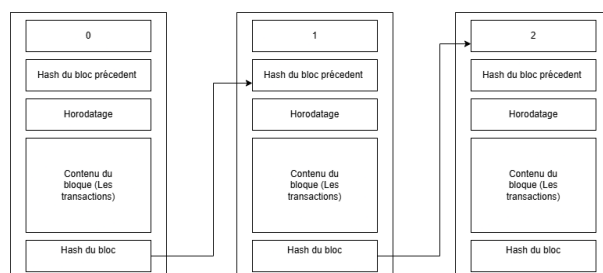


FIGURE 3.5 – Fonctionnement du blockchain

3.8 Classification de la blockchain

Une classification générale divise la blockchain en trois catégories : la blockchain publique, la blockchain privée et la blockchain de consortium.

3.8.1 Blockchain publique

Les blockchains publiques sont un type de blockchains décentralisées et sans permission, dans lesquelles tous les membres du réseau ont accès aux informations et peuvent participer à leur validation. Bitcoin et Ethereum sont des exemples de blockchains publiques. Ce type de blockchain est sécurisé grâce à son mécanisme de consensus, qui garantit un accord entre tous les pairs.

3.8.2 Blockchain privée

Les blockchains privées sont des blockchains avec permissions, dans lesquelles l'information est exclusivement accessible à un groupe restreint, et l'acceptation des modifications est limitée à ce groupe, comme c'est le cas, par exemple, d'un système de paie basé sur blockchain. Il s'agit d'une blockchain centralisée, ce qui signifie qu'une autorité centrale décide qui peut lire, écrire ou participer à la blockchain. Par conséquent, une seule entité définit le mécanisme de consensus dans les blockchains privées.

3.8.3 Blockchain de consortium

Les blockchains de consortium, également appelées blockchains fédérées, sont des plateformes administrées par un ensemble d'organisations plutôt que par une seule entité centrale. Elles se positionnent entre les blockchains publiques et privées, en combinant des caractéristiques propres à ces deux modèles. Contrairement aux blockchains privées, où l'autorité décisionnelle est concentrée entre les mains d'une seule organisation, les blockchains de consortium permettent à plusieurs entités de participer conjointement au processus de validation et de gouvernance.

3.9 Application de la blockchain

La blockchain, initialement développée pour les cryptomonnaies comme Bitcoin, s'est révélée être une technologie polyvalente avec des applications dans de nombreux secteurs grâce à ses propriétés de décentralisation, de transparence et de sécurité. Voici un aperçu des principales applications de la blockchain :

1. **Finance et Cryptomonnaies** : La blockchain reste au cœur des cryptomonnaies comme Bitcoin et Ethereum, permettant des transactions peer-to-peer sans intermédiaires. Au-delà des monnaies numériques, elle soutient la finance décentralisée (DeFi), offrant des services comme les prêts, les échanges et les contrats intelligents, accessibles sans institutions financières traditionnelles.
2. **Gestion des Chaînes d'Approvisionnement** : La blockchain améliore la traçabilité et la transparence dans les chaînes d'approvisionnement. En enregistrant chaque étape du parcours d'un produit, elle permet de vérifier l'authenticité et l'origine des biens, réduisant ainsi les fraudes et les contrefaçons. Des entreprises comme IBM utilisent la blockchain pour optimiser la logistique et garantir la qualité des produits.
3. **Santé** : Dans le secteur de la santé, la blockchain sécurise les dossiers médicaux, permettant un partage de données entre professionnels tout en préservant la confidentialité des patients. Elle facilite également la traçabilité des médicaments, réduisant les risques liés aux contrefaçons pharmaceutiques.
4. **Gouvernance et Identité Numérique** : La blockchain offre des solutions pour une gestion décentralisée de l'identité numérique, permettant aux individus de contrôler leurs données personnelles sans dépendre d'autorités centrales. Des initiatives comme uPort utilisent la blockchain pour créer des identifiants numériques sécurisés et vérifiables.

Ces applications démontrent le potentiel de la blockchain à transformer les interactions numériques et physiques en offrant des solutions sécurisées, transparentes et décentralisées.

3.10 Conclusion

La blockchain joue un rôle crucial dans les systèmes d'aide à la décision de groupe (GDSS), en particulier pour sécuriser le vote. Sa capacité à garantir la transparence, l'intégrité et la traçabilité des votes renforce la confiance dans les processus décisionnels, même dans des contextes impliquant de nombreux participants ou des localisations géographiquement dispersées. En éliminant les intermédiaires et en assurant l'inviolabilité des données, elle répond aux enjeux critiques de l'équité et de la fiabilité dans les GDSS.

Ce chapitre a établi les fondements nécessaires pour comprendre la technologie blockchain et ses éléments clés. Nous avons retracé son histoire et son développement, depuis sa création innovante par Satoshi Nakamoto jusqu'à son statut actuel d'innovation majeure aux usages variés. En définissant clairement la blockchain et en examinant ses propriétés essentielles — décentralisation, transparence et sécurité —, nous avons souligné sa capacité à transformer de nombreux domaines, notamment les systèmes de décision collective.

Chapitre 4

Conception et modélisation

4.1 Introduction

Dans le cadre de la prise de décision de groupe, les systèmes d'aide à la décision de groupe (GDSS) jouent un rôle crucial en facilitant la collaboration et en structurant les processus de vote. Cependant, les architectures centralisées traditionnellement utilisées pour les systèmes de vote présentent des limites significatives, notamment en termes de transparence, de sécurité et de confiance. Ces défis, combinés à la nécessité d'assurer des décisions justes et représentatives, motivent l'exploration de solutions décentralisées basées sur des technologies innovantes telles que la blockchain.

Ce chapitre expose la conception d'une application décentralisée (DApp) dédiée à la gestion des votes au sein d'un système de prise de décision de groupe (GDSS). Cette solution, s'appuyant sur la technologie blockchain, a été élaborée pour satisfaire les impératifs fondamentaux de transparence, d'intégrité et de confidentialité inhérents au vote électronique. En outre, ce chapitre détaille le fonctionnement conceptuel de cette DApp, en explicitant ses composants essentiels, son architecture logicielle et les interactions orchestrées entre les différents acteurs impliqués dans le processus de vote.

4.2 Systèmes de vote centralisés

Les systèmes de vote centralisés, bien qu'étant largement utilisés dans divers contextes, présentent plusieurs limites qui remettent en question leur fiabilité et leur efficacité dans des processus décisionnels critiques.

Premièrement, ces systèmes souffrent d'un manque de transparence, car les résultats sont souvent sous le contrôle d'une autorité centrale qui peut être tentée ou contrainte de manipuler les données à son avantage.

Deuxièmement, ils sont vulnérables aux attaques en raison de leurs points uniques de défaillance, tels que les serveurs ou les bases de données centralisées, qui peuvent être ciblés par des cyberattaques ou des pannes techniques.

Troisièmement, ces architectures reposent sur une confiance aveugle envers une entité centrale, ce qui est problématique dans des environnements où la neutralité et l'impartialité doivent être garanties.

Enfin, elles peinent à assurer un véritable anonymat, car la centralisation des données rend difficile la protection de la confidentialité des votes contre des accès non autorisés ou des fuites d'informations.

Ces limites ne sont pas théoriques et ont été observées dans des cas réels ou hypothétiques. Par exemple, lors de certaines élections électroniques nationales, des systèmes centralisés ont été critiqués pour leur manque de transparence, alimentant des soupçons de fraude électorale. Un cas notable est celui des élections électroniques en Estonie, où, malgré un système robuste, des préoccupations ont été soulevées quant à la dépendance envers une infrastructure centralisée pour la vérification des votes. De manière hypothétique, dans une organisation utilisant un logiciel de vote centralisé pour choisir un projet, un administrateur malveillant pourrait modifier les résultats pour favoriser une option spécifique, sans que les participants ne puissent le détecter. Ces exemples soulignent l’incapacité des architectures centralisées à garantir simultanément transparence, sécurité et confiance.

4.3 Système proposé

Pour surmonter les limitations des systèmes de vote centralisés, nous proposons une architecture décentralisée basée sur la technologie blockchain. Cette approche offre des solutions robustes aux défis critiques identifiés dans les systèmes traditionnels, notamment le manque de transparence, la vulnérabilité aux attaques, les problèmes de confiance et l’absence d’anonymat. Les caractéristiques fondamentales de la blockchain – immuabilité, transparence, décentralisation, sécurité et confidentialité – en font une base idéale pour concevoir un système de vote sécurisé et fiable, particulièrement adapté à un système de prise de décision de groupe (GDSS). Dans cette section, nous analysons comment la blockchain répond aux limites des systèmes centralisés, soulignons ses avantages supplémentaires pour un GDSS.

1. Transparence et vérification indépendante

Les systèmes de vote centralisés souffrent d’un manque de transparence, car une autorité unique contrôle les résultats, ce qui peut entraîner des manipulations indétectables. En revanche, la blockchain résout ce problème grâce à son registre distribué, qui est soit public (dans les blockchains publiques), soit accessible aux participants autorisés (dans les blockchains privées ou de consortium). Chaque vote est enregistré comme une transaction sur ce registre, et l’immuabilité de la blockchain garantit que ces données ne peuvent être modifiées une fois enregistrées. Cette transparence permet à tous les participants de vérifier indépendamment l’intégrité des votes et des résultats, éliminant ainsi le besoin de faire confiance à une entité centrale.

Une étude récente [48] met en avant que la blockchain offre une ”vérification de bout en bout” (end-to-end verification), permettant aux utilisateurs de suivre l’ensemble du processus de vote, de la soumission à la comptabilisation. Cela renforce la confiance des participants envers le système, car ils peuvent s’assurer que leurs votes ont été correctement enregistrés et pris en compte.

2. Résilience contre les attaques et élimination des points uniques de défaillance

Les systèmes centralisés sont intrinsèquement vulnérables en raison de leurs points uniques de défaillance, tels que les serveurs ou les bases de données centrales, qui peuvent être ciblés par des cyberattaques ou subir des pannes techniques. La blockchain, en revanche, repose sur un réseau distribué de nœuds, éliminant ces points uniques de défaillance. Même si certains nœuds sont compromis, le réseau reste opérationnel tant que la majorité des nœuds sont honnêtes.

Les mécanismes de consensus, tels que la Proof of Work (PoW) ou la Proof of Stake (PoS), jouent un rôle crucial dans la validation des transactions. Ces mécanismes garantissent que seules les transactions valides (votes) sont ajoutées à la blockchain,

renforçant ainsi la résilience contre les attaques malveillantes et assurant l'intégrité globale du système.

3. Répartition de la confiance et neutralité

Dans les systèmes centralisés, les participants doivent faire confiance aveuglément à une autorité centrale pour garantir la neutralité et l'impartialité du processus de vote. Cette centralisation de la confiance est problématique, en particulier dans des contextes où la neutralité est essentielle. La blockchain répartit cette confiance entre les participants du réseau grâce à sa nature décentralisée. Le registre distribué et immuable garantit que les votes, une fois enregistrés, ne peuvent être modifiés sans l'accord du réseau, éliminant ainsi le risque de manipulation par une entité unique.

De plus, la transparence inhérente à la blockchain permet de dissiper toute suspicion de partialité. Les participants peuvent vérifier que le processus de vote respecte les règles prédéfinies et que les résultats reflètent fidèlement les intentions des votants.

4. Protection de l'anonymat et confidentialité des votes

Un autre défi majeur des systèmes centralisés est leur incapacité à garantir l'anonymat des votants. Dans ces systèmes, les votes sont souvent liés à des identités stockées dans une base de données centrale, exposant les votants à des risques de surveillance ou de fuites de données. La blockchain utilise des techniques cryptographiques avancées pour protéger la confidentialité des votants tout en assurant la transparence du processus global.

4.4 Description fonctionnelle

Le système de vote proposé repose sur un modèle fonctionnel structuré autour de trois phases principales : l'authentification des utilisateurs, la création et la gestion des sessions de vote, ainsi que la participation active au vote et la publication des résultats. Chaque utilisateur du système dispose d'une identité numérique unique lui permettant de s'authentifier de manière sécurisée et de participer aux opérations selon son rôle attribué. Deux types d'acteurs principaux interviennent dans le processus : **le facilitateur**, chargé de la création et de la participation aux votes, et **les participants**, qui expriment leurs choix dans le cadre des sessions autorisées.

Le facilitateur joue un rôle central dans l'initiation du processus électoral. Il définit les paramètres essentiels du vote, notamment le nombre des votants et les options soumises au suffrage. Une fois ces paramètres configurés, il lance la session de vote, rendant celle-ci accessible aux participants autorisés. Ces derniers peuvent alors intervenir pendant la durée définie pour émettre leur vote selon une méthode spécifique : **le vote par approbation**. Ce mode de vote permet à chaque votant d'approuver une ou plusieurs options parmi celles proposées, ou n'approuve rien, c'est-à-dire de voter blanc, sans hiérarchisation ni pondération. L'agrégation finale des résultats est effectuée automatiquement, en additionnant le nombre total d'approbations reçues par chaque option.

Le système garantit à la fois la validité du processus et la fiabilité des résultats grâce à un mécanisme de contrôle des accès, une traçabilité des actions effectuées, ainsi qu'un suivi précis du nombre de votants attendus et réels. Le vote est clos automatiquement dès que tous les votants autorisés ont exprimé leur choix, ce qui met fin à la phase de vote de manière irréversible, sans possibilité d'intervention manuelle ou anticipée de la part du facilitateur. Une fois cette étape atteinte, les résultats sont calculés conformément à la méthode du vote par approbation et publiés de manière transparente, restant accessibles à tous les utilisateurs du système. En cas d'égalité entre les options reçues, le facilitateur a la possibilité d'initier un

second vote afin de trancher, sous les mêmes conditions de participation et de sécurité. Cette fonctionnalité offre au système une capacité d'auto-régulation et de poursuite du processus décisionnel en cas d'absence de décision claire lors du premier tour.

4.5 Modélisation du système

Afin de formaliser et structurer la conception du système de vote proposé, nous allons utiliser les diagrammes UML. Ces diagrammes ont permis de représenter visuellement les interactions entre les acteurs et le système, les flux d'opérations ainsi que la structure interne des composants logiciels. Trois types de diagrammes principaux ont été utilisés : le diagramme des cas d'utilisation, le diagramme de séquence et le diagramme de classes.

4.5.1 Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation (figure 4.1) illustre les interactions entre les acteurs principaux du système (le Participant et le Facilitateur) et les fonctionnalités du système. Ce diagramme met en évidence les différentes actions que chaque acteur peut effectuer, ainsi que les relations entre ces actions.

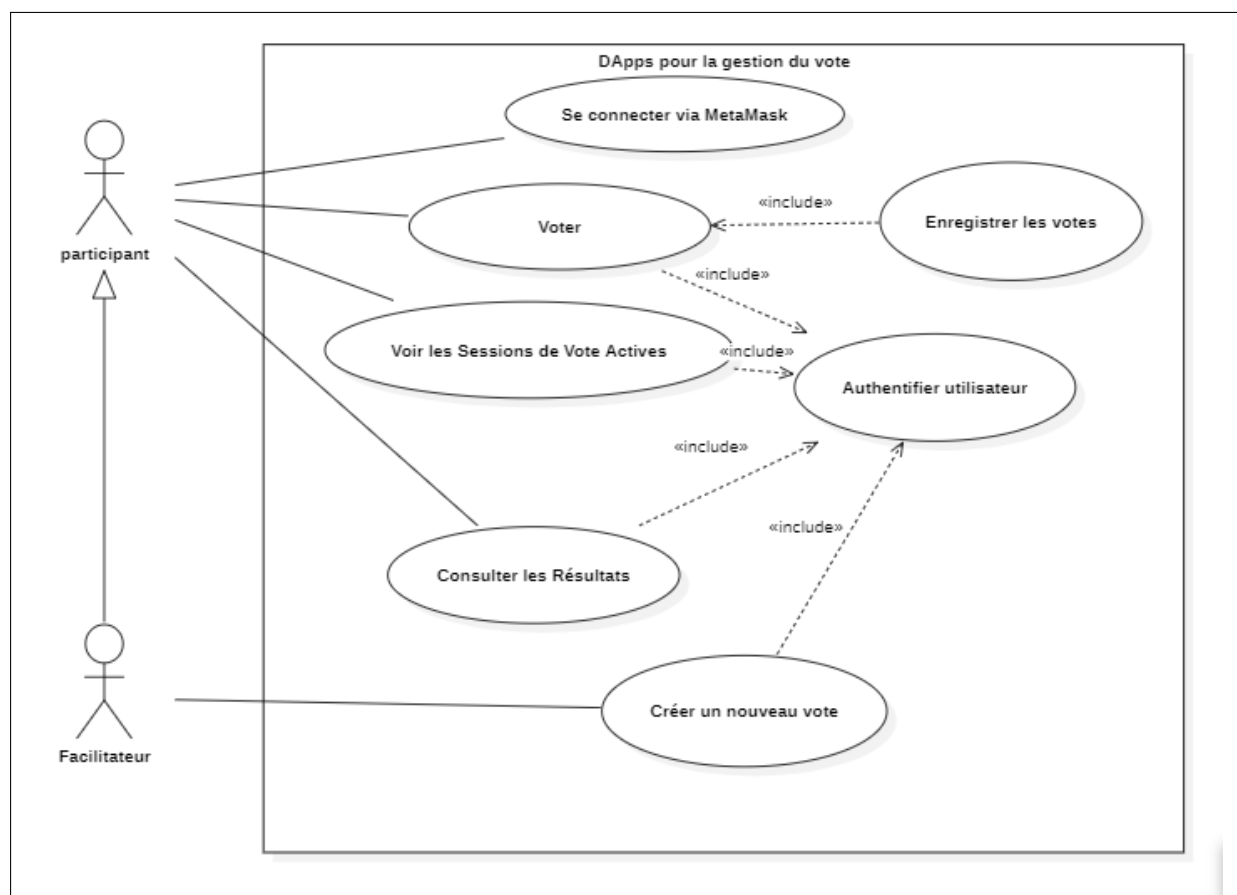


FIGURE 4.1 – Diagramme de cas d'utilisation pour la gestion de vote dans un GDSS

Description du diagramme :

a. Cas d'utilisation du Participant

Le **participant** interagit principalement avec le système pour consulter les votes en cours, soumettre son vote et accéder aux résultats des sessions clôturées. Trois cas d'utilisation majeurs sont associés à ce rôle :

- **Voter** : ce cas d'utilisation permet au participant de sélectionner et soumettre ses choix lors d'une session de vote active. Il dépend obligatoirement du cas. *Se connecter via MetaMask*, par une relation de type *include*, garantissant une authentification préalable et sécurisée.
- **Voir les sessions de Vote Actives** : ce cas offre au participant une vue actualisée des votes en cours auxquels il peut potentiellement participer.
- **Consulter les Résultats** : accessible uniquement une fois la session de vote clôturée, ce cas permet d'afficher les résultats de manière transparente, tout en préservant l'anonymat des participants.

b. Cas d'utilisation du Facilitateur

Le **Facilitateur** détient des privilèges spécifiques liés à la gestion des votes. Ses interactions principales comprennent :

- **Créer un Nouveau Vote** : ce cas permet de définir et d'initier une session de vote en précisant les paramètres essentiels (titre, description, options, nombre attendu de participants).

De plus, le facilitateur bénéficie implicitement de tous les cas d'utilisation associés au rôle de **participant**, dans la mesure où il en hérite les fonctionnalités.

c. Cas d'utilisation du Smart Contract

Le **smart contract** agit comme un acteur technique passif, déclenchant automatiquement certaines opérations critiques à la réception d'instructions légitimes :

- **Enregistrer les Votes** : activé à partir de l'action *voter* du participant, ce cas assure l'intégrité et la traçabilité des votes soumis, en les stockant de manière immuable.
- **Calculer les Résultats** : également appelé par le facilitateur ou automatiquement par le système, ce cas formalise le décompte des approbations et détermine les résultats finaux du vote.

De plus, les relations « include » dans le diagramme indiquent que certaines fonctionnalités dépendent d'autres. Par exemple : le cas d'utilisation « Voter » inclut nécessairement « Enregistrer les votes ». Le cas d'utilisation « Fermer le vote » inclut « Calculer les résultats », afin de produire les résultats finaux dès la fin du vote.

4.5.2 Diagramme de séquence

Deux diagrammes de séquence ont été élaborés afin de représenter les interactions temporelles entre les acteurs du système et ses différents composants logiciels. Ces diagrammes illustrent respectivement :

- Le processus de création d'un vote , piloté par le facilitateur.
- Le processus de participation au vote , mené par les participant.

Ces deux diagrammes permettent de formaliser les flux d'informations, les appels de fonctions, ainsi que les vérifications nécessaires à l'exécution correcte de chaque opération critique. Ils offrent une vue chronologique claire des étapes impliquées, des responsabilités attribuées à chaque entité, et des conditions préalables à remplir pour garantir la sécurité et l'intégrité des actions effectuées.

1. Le processus de participation au vote

Le diagramme de séquence (figure 4.2) illustre le déroulement d'une session de vote depuis l'authentification de l'utilisateur jusqu'à la publication des résultats. Il met en évidence les interactions principales entre les différents acteurs : le participant, MetaMask, le système de vote et le smart contract.

Le processus commence lorsque le participant initie une connexion via son portefeuille MetaMask. Une fois l'authentification réussie, il peut consulter les votes actifs et soumettre son vote. Lors du vote, le système envoie les données au Smart Contract, qui enregistre les choix de manière immuable sur la blockchain. Lorsque tous les votes sont collectés, le système déclenche le calcul des résultats. Ce calcul est effectué automatiquement par le Smart Contract . Finalement, les résultats sont publiés de façon transparente, accessibles aux utilisateurs autorisés.

Ce diagramme permet de visualiser la coordination entre les composants techniques et les rôles humains, tout en assurant la traçabilité et la sécurité de chaque étape du processus électoral.

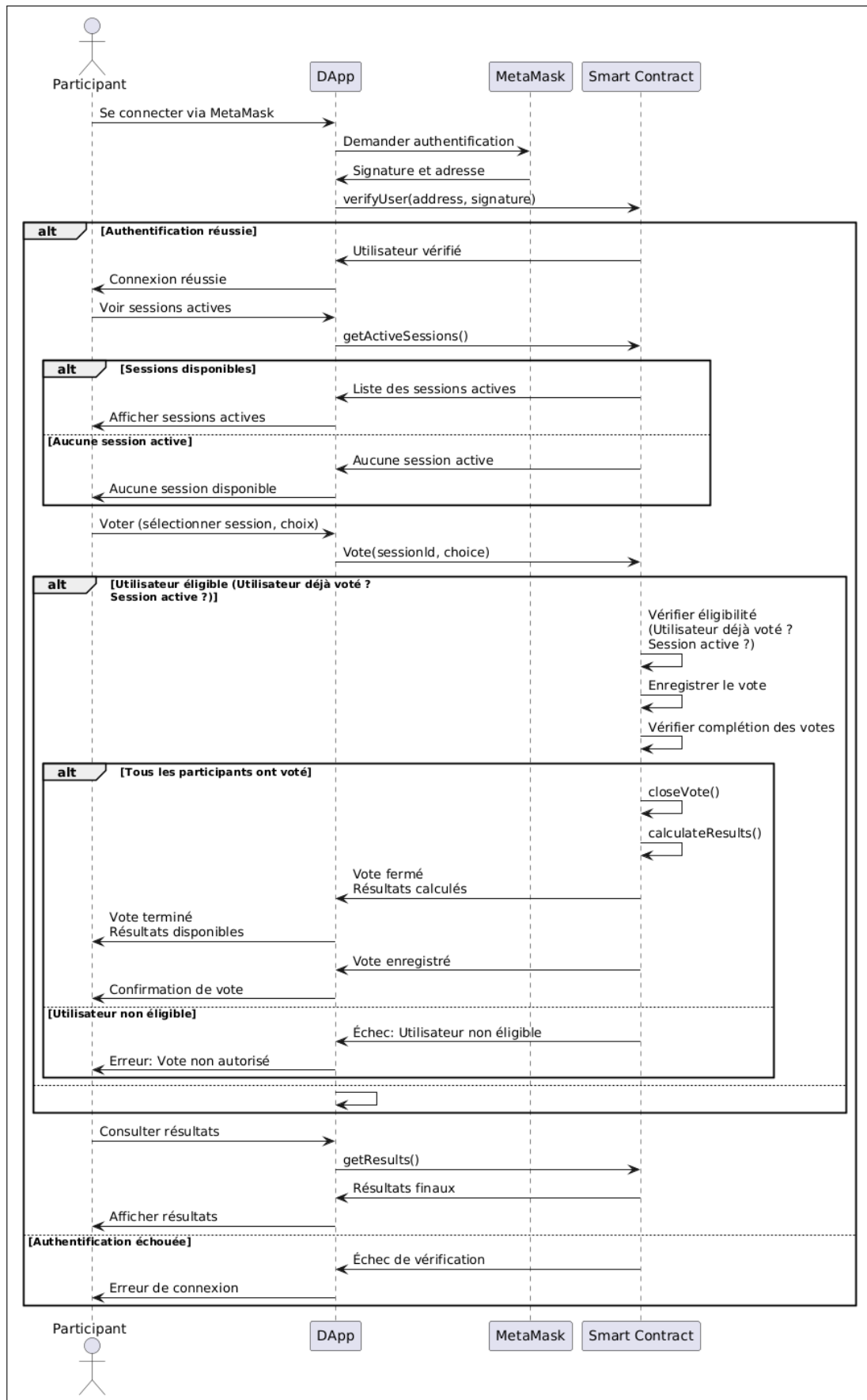


FIGURE 4.2 – Diagramme de séquence pour le processus de participation au vote

2. Le processus de création d'un vote

Le diagramme de séquence 4.3 illustre le processus de création d'une session de vote par un facilitateur. Ce diagramme met en lumière les étapes critiques du processus, y compris l'authentification, la création de la session de vote et la vérification des droits administratifs.

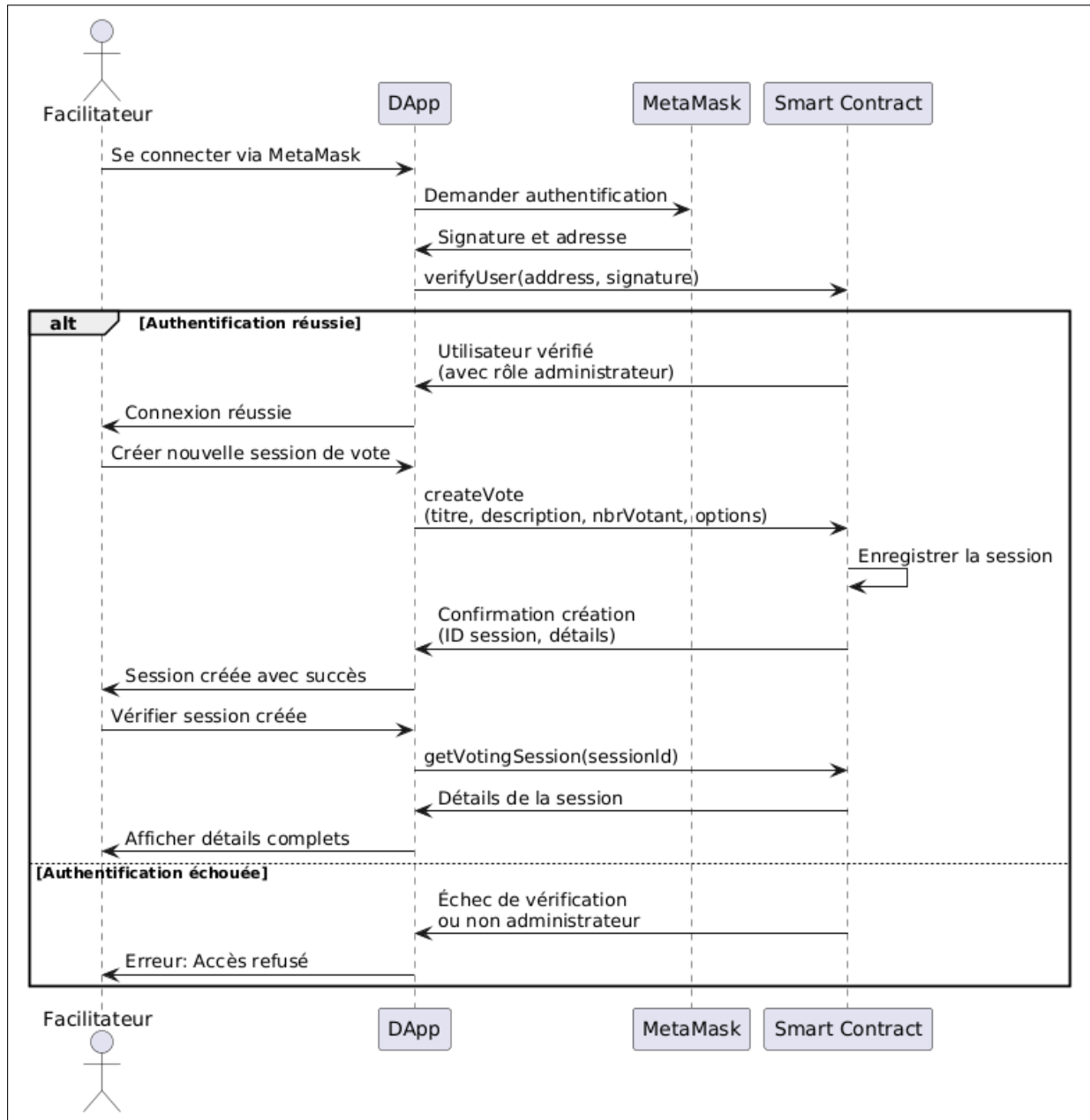


FIGURE 4.3 – Diagramme de séquence pour le processus de participation au vote

4.5.3 Diagramme de classe

Le diagramme de classes présenté ci-dessus (figure 4.4) constitue une représentation de la structure du système. Il met en lumière les principales classes impliquées dans le processus électoral, ainsi que leurs relations et leurs responsabilités respectives. Bien que ce diagramme ne couvre pas l'ensemble des composants du système, il fournit une vue claire des interactions

fondamentales entre les acteurs principaux (Facilitateur, Participant), les sessions de vote et les votes individuels.

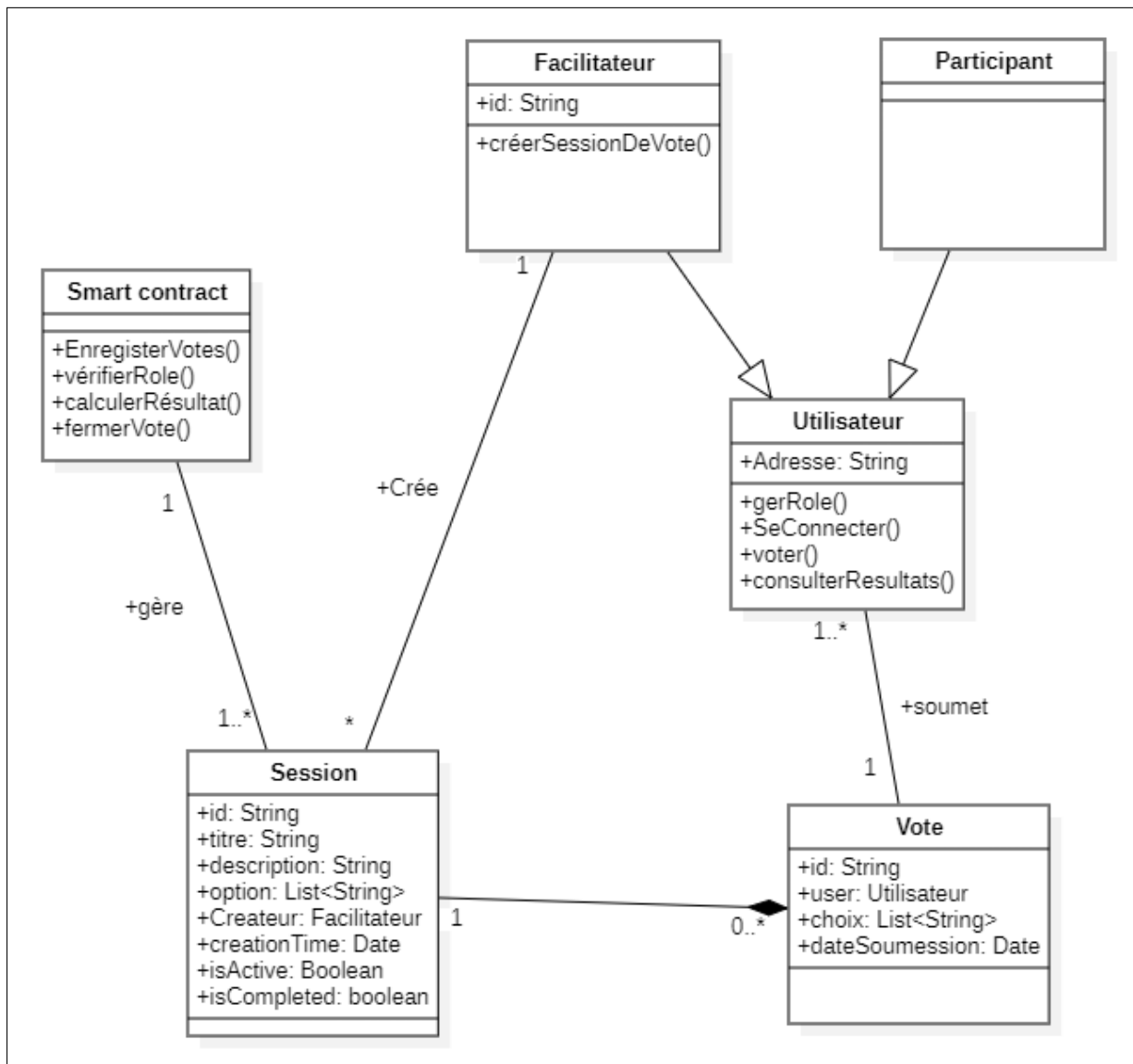


FIGURE 4.4 – Diagramme de classes du système de vote basé sur la blockchain

Classes principales :

- **Utilisateur** : La classe Utilisateur constitue la classe de base pour tous les utilisateurs du système. Elle encapsule les informations communes aux Facilitateurs et aux Participants.
- **Facilitateur** : La classe Facilitateur hérite de la classe Utilisateur et ajoute des responsabilités spécifiques liées à la gestion des sessions de vote.
- **Participant** : La classe Participant hérite également de la classe Utilisateur. Elle représente un votant autorisé à participer aux sessions de vote.
- **Session** : La classe Session représente une session de vote spécifique, avec ses paramètres (titre, options, état, etc.).
- **Vote** : modélise le choix exprimé par un participant dans une session donnée.

- **SmartContract** : gère les opérations critiques telles que l'enregistrement des votes, la vérification des rôles, le calcul des résultats et la clôture automatique du vote.

Ce diagramme fournit une vue abstraite mais structurée du fonctionnement du système, en mettant en évidence les interactions clés nécessaires à une implémentation fiable et sécurisée sur la blockchain.

4.6 Conclusion

Ce chapitre a permis de présenter de manière structurée la conception et la modélisation du système de vote dans un GDSS basé sur la blockchain.

La conception de cette DApp représente une avancée significative vers des processus de vote plus transparents, sécurisés et équitables. En exploitant les avantages de la technologie blockchain, cette solution répond aux défis posés par les architectures centralisées traditionnelles tout en garantissant l'intégrité et la confidentialité des données. Cette base conceptuelle ouvre la voie à une exploration approfondie des aspects techniques et pratiques nécessaires à sa mise en œuvre.

Chapitre 5

Implémentation du système

5.1 Introduction

Ce chapitre se concentre sur l'implémentation de l'application décentralisée dédiée à la gestion des votes au sein d'un GDSS. Après avoir établi les fondements conceptuels de la DApp, il est essentiel de traduire ces idées en une solution technique fonctionnelle qui répond aux exigences de transparence, de sécurité et d'intégrité.

L'objectif de ce chapitre est de détailler les étapes clés de développement, depuis les technologies utilisées jusqu'à la réalisation technique des smart contracts et de l'interface utilisateur. Nous aborderons successivement l'environnement technique retenu, la structure et les fonctionnalités des contrats intelligents écrits en Solidity, le déploiement sur le réseau blockchain ainsi que le développement de l'application front-end (DApp).

5.2 Concepts fondamentaux

5.2.1 Smart contract

Un smart contract (contrat intelligent) est un programme informatique auto-exécutable dont les termes sont codifiés directement dans le code et déployés sur une blockchain. Conçu initialement dans les années 1990, ce concept repose sur l'idée d'un contrat dont l'exécution est automatique lorsque des conditions prédéfinies sont remplies, sans nécessiter d'intermédiaire centralisé.

Les smart contracts sont stockés, répliqués et mis à jour sur des blockchains distribuées, garantissant ainsi leur transparence, leur inviolabilité et leur disponibilité permanente. Ils permettent de remplacer les mécanismes traditionnels de confiance reposant sur des tiers de confiance (comme les notaires, les banques ou autres institutions) par une confiance algorithmique, assurée par la logique du code et la sécurité du réseau blockchain [49].

5.2.2 Le web 3.0

Le Web 3.0, également appelé *Web décentralisé*, représente la troisième génération de l'internet. Il se distingue par une architecture fondée sur des protocoles décentralisés, visant à réduire, voire éliminer, la dépendance aux intermédiaires centralisés et à redonner le contrôle aux utilisateurs. Contrairement au Web 2.0 — dominé par des plateformes numériques centralisées comme les réseaux sociaux ou les services de cloud —, le Web 3.0 repose sur des tech-

nologies telles que la *blockchain* et les *smart contracts* (contrats intelligents). Ces innovations permettent une gestion transparente, sécurisée et participative des données et des interactions numériques.

5.2.3 Les applications décentralisées (DApps)

Une **DApp** (*Decentralized Application*) est une application logicielle construite sur une architecture décentralisée, le plus souvent basée sur la technologie *blockchain*. Contrairement aux applications traditionnelles du Web 2.0, qui s'appuient sur des serveurs centralisés gérés par une autorité unique, les DApps fonctionnent de manière autonome grâce à l'exécution de *smart contracts* (contrats intelligents) sur un réseau *peer-to-peer*. Leur code source est généralement *open source*, et les règles de fonctionnement ainsi que les données sont publiquement accessibles, ce qui garantit à la fois la **transparence**, l'**intégrité** et la **résistance à la censure**.

5.3 Environnement technique et outils utilisés

5.3.1 Choix de la plateforme blockchain

Le développement d'un système de vote basé sur la blockchain nécessite une sélection rigoureuse de la plateforme sous-jacente, en fonction de critères tels que la sécurité, la maturité technologique, l'écosystème de développement et la facilité d'intégration avec les outils existants. Après analyse des solutions disponibles, la plateforme Ethereum a été retenue comme fondement de notre implémentation.

Plusieurs caractéristiques techniques ont motivé le choix de la plateforme **Ethereum** pour le développement de notre application décentralisée de gestion des votes :

- **Support avancé des smart contracts** : Ethereum utilise le langage *Solidity*, qui offre une expressivité suffisante pour implémenter les règles de gestion électorale : définition des scrutins, vérification des rôles, enregistrement sécurisé des votes, et fermeture automatique des sessions.
- **Décentralisation et résistance à la censure** : Grâce à son architecture *pair-à-pair* et à son *modèle de consensus sécurisé (Proof of Stake)*, Ethereum garantit une forte résistance aux manipulations de données, un critère essentiel pour la fiabilité d'un processus électoral.
- **Interopérabilité avec MetaMask** : Ethereum est compatible avec des portefeuilles numériques comme *MetaMask*, facilitant ainsi l'*authentification sécurisée*, la *signature des transactions*, et la *gestion des identités numériques* sans intermédiaire central.
- **Facilité d'intégration avec les DApps** : Ethereum propose des interfaces standardisées comme l'ABI et des protocoles robustes comme *JSON-RPC*, assurant une communication fluide entre la DApp et les smart contracts.

5.3.2 Langage de programmation

Le développement des smart contracts constitue un aspect central de la mise en œuvre de notre système. Pour ce projet, le langage utilisé est **Solidity**, un langage de haut niveau orienté objet spécialement conçu pour la programmation de contrats intelligents sur la blockchain Ethereum. Ce choix s'inscrit dans la continuité du précédent, Solidity étant le langage le plus répandu et le mieux supporté par l'écosystème Ethereum.

Présentation de Solidity :

Solidity est un langage de programmation orienté objet, de haut niveau, conçu pour l'implémentation de contrats intelligents s'exécutant sur la Machine Virtuelle Ethereum (EVM). Proposé en 2014 sous l'impulsion de Gavin Wood, CTO d'Ethereum à l'époque, il a été développé par Christian Reitwiessner afin de répondre au besoin d'un langage Turing-complet permettant de coder des smart contracts complexes et auto-exécutables.

Le développement de Solidity s'est fortement inspiré de langages existants tels que C++, JavaScript et Python. En tant que langage compilé, Solidity est transformé en bytecode compris par la EVM, puis exécuté de manière immuable sur la blockchain. Cette propriété garantit une exécution transparente, sans possibilité de modification post-déploiement, ce qui est essentiel pour la sécurité des contrats critiques.

Grâce à son écosystème riche (Truffle, Hardhat, Remix, etc.) et à sa compatibilité avec les DApps front-end via Web3.js ou Ethers.js, Solidity occupe aujourd'hui une place centrale dans le développement Web3, constituant un outil incontournable pour la conception de systèmes fiables, traçables et autonomes comme celui proposé dans cette recherche.

5.3.3 Frameworks et outils de développement

L'implémentation d'un système de vote décentralisé repose non seulement sur un langage de programmation adapté tel que Solidity, mais aussi sur l'utilisation de frameworks et bibliothèques facilitant le développement, le test, le déploiement et l'intégration des contrats intelligents (smart contracts) avec une interface utilisateur. Dans ce contexte, plusieurs outils ont été utilisés, parmi lesquels Truffle et Web3.js, qui constituent des éléments essentiels de l'écosystème Ethereum.

1. Truffle :

Truffle est un framework de développement reconnu pour les applications décentralisées (dApps) qui interagissent avec la blockchain Ethereum. Sa conception est spécialement pensée pour fournir aux développeurs les outils nécessaires à la compilation, au test et au déploiement efficace de smart contracts écrits en Solidity, le langage de programmation des contrats intelligents sur Ethereum.

Truffle offre un ensemble de fonctionnalités structurées permettant de simplifier et d'optimiser le développement des smart contracts sur la blockchain Ethereum :

- **Compilation** : Truffle intègre le compilateur *Solc* (Solidity Compiler), qui transforme les contrats écrits en *Solidity* en bytecode optimisé, prêt à être exécuté par la EVM. Cette étape est essentielle pour assurer la compatibilité des contrats avec les différents réseaux Ethereum.
- **Migrations** : Truffle propose un système de *migrations automatisées*, permettant de déployer les contrats intelligents sur divers environnements, qu'il s'agisse d'un *réseau local (Ganache)*, d'un *testnet* ou du *mainnet*. Ce mécanisme garantit une gestion efficace et versionnée des contrats déployés.
- **Tests** : L'environnement de test intégré de Truffle prend en charge les langages *JavaScript* et *Solidity*, permettant la création de *tests unitaires et fonctionnels automatisés*. Cela renforce la fiabilité du code en détectant précocement les anomalies avant le déploiement sur la blockchain.

2. Web3.js :

Une fois les contrats déployés, il est nécessaire de permettre à l'application front-end (DApp) d'interagir avec eux. C'est ici qu'intervient Web3.js, une bibliothèque JavaScript riche et modulaire permettant aux développeurs d'interagir avec des nœuds Ethereum locaux ou distants via des connexions HTTP, WebSocket ou IPC. Cette bibliothèque constitue une implémentation TypeScript complète de l'API JSON-RPC d'Ethereum, offrant ainsi un accès direct aux fonctionnalités du réseau Ethereum et des blockchains compatibles EVM.

Grâce à Web3.js, les développeurs peuvent réaliser une large gamme d'opérations essentielles telles que l'accès aux comptes utilisateurs, l'envoi de transactions, la consultation des données stockées sur la blockchain ou encore la gestion des contrats intelligents. Ces capacités en font un outil incontournable dans le développement des DApps, particulièrement dans des contextes comme notre système de vote, où la transparence et la sécurité sont primordiales.

3. Ganache :

Ganache est un outil de développement essentiel dans l'écosystème Ethereum, conçu pour simuler un environnement blockchain local. Il permet aux développeurs de tester leurs smart contracts et DApps de manière rapide et économique, sans passer par des réseaux publics comme le *mainnet* ou les *testnets*, évitant ainsi les frais de *gas* et les délais liés à la validation réelle des transactions.

Développé par la suite logicielle **Truffle Suite**, Ganache se décline en deux versions :

- **Ganache CLI** : une version en ligne de commande, adaptée aux utilisateurs expérimentés souhaitant automatiser leur flux de travail ;
- **Ganache GUI** : une interface graphique intuitive, offrant une visualisation claire des comptes, des transactions, des blocs minés et des interactions avec les smart contracts.

Lorsqu'un projet DApp est lancé dans Ganache, le système génère automatiquement :

- Une liste de comptes préconfigurés, chacun crédité d'un solde fictif en ETH.
- Un port RPC local (par défaut 7545 pour ganache UI et 8545 pour ganache CLI) pour permettre la communication avec l'application.
- Un ensemble d'informations détaillées sur les blocs, les transactions, les événements et les logs des contrats, facilitant ainsi le débogage.

4. Metamask :

MetaMask est un portefeuille numérique logiciel conçu pour permettre aux utilisateurs d'interagir facilement avec la blockchain Ethereum et les DApps. Disponible sous forme d'extension pour navigateurs (Chrome, Firefox, Brave) et en version mobile, MetaMask constitue aujourd'hui l'un des outils les plus populaires et accessibles pour accéder à l'univers du Web3.

Dans le cadre de notre système de vote décentralisé, MetaMask joue un rôle fondamental en tant que :

- outil d'authentification des utilisateurs (votants et facilitateurs) ;
- mécanisme de signature des transactions ;
- interface sécurisée d'interaction avec les smart contracts.

MetaMask agit à la fois comme gestionnaire de clés privées et comme passerelle entre l'utilisateur final et la blockchain. Il permet de stocker, envoyer et recevoir de l'Ether (ETH), ainsi que des tokens conformes aux standards *ERC-20* et *ERC-721*.

Contrairement aux portefeuilles complets nécessitant de synchroniser toute la blockchain, MetaMask fonctionne en tant que client léger, ce qui le rend simple, rapide et accessible aux utilisateurs non techniques.

L'un de ses avantages clés réside dans sa capacité à connecter les utilisateurs aux DApps de manière sécurisée et intuitive, sans recourir à un identifiant ou mot de passe traditionnel. L'authentification s'effectue via la clé privée de l'utilisateur, associée à son adresse Ethereum, garantissant ainsi un haut niveau de sécurité et de confidentialité.

5.4 Conception et développement des smart contracts

Les smart contracts constituent la pierre angulaire du système de vote proposé. Ils encapsulent l'ensemble de la logique métier, assurent la gestion des interactions critiques entre les acteurs (facilitateur et participant), garantissent la sécurité des données et offrent une transparence totale du processus électoral.

Deux contrats intelligents principaux ont été développés :

- Le contrat **VotingRoles.sol** permet d'assigner ou de révoquer des rôles spécifiques (facilitateur ou participant), et de vérifier les autorisations associées à chaque action.
- Le contrat **Voting.sol** implémente la logique centrale du processus électoral, comprenant :
 - la création du scrutin par le facilitateur ;
 - la soumission des votes par les participants ;
 - la comptabilisation des voix selon le principe du *vote par approbation* ;
 - la fermeture automatique du scrutin dès que tous les votants ont exprimé leur choix.

5.4.1 Conception du contrat VotingRoles.sol

Le contrat VotingRoles joue un rôle central dans la définition de l'identité des utilisateurs au sein du système. Il gère les rôles des utilisateurs impliqués dans le processus de vote. Il distingue deux types de rôles : le *facilitator* et les *participants*.

Les principales fonctionnalités du contrat VotingRoles incluent :

- Attribution initiale des rôles lors du déploiement.
- Enregistrement dynamique d'un utilisateur comme participant via la fonction `registerAsParticipant`.
- Vérification du rôle d'un utilisateur via la fonction `getRole(address)`.

La figure [5.1](#) présente une capture du contrat intelligent VotingRoles.sol, détaillant les principales structures de données et méthodes implémentées.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract VotingRoles {
    mapping(address => string) public roles;
    address public facilitator;

    event RoleAssigned(address indexed user, string role);

    constructor(address _facilitator, address[] memory _participants) {
        roles[_facilitator] = "facilitator";
        facilitator = _facilitator;
        emit RoleAssigned(_facilitator, "facilitator");

        for (uint i = 0; i < _participants.length; i++) {
            roles[_participants[i]] = "participant";
            emit RoleAssigned(_participants[i], "participant");
        }
    }

    function registerAsParticipant() public {
        require(bytes(roles[msg.sender]).length == 0, "Utilisateur deja enregistre");
        roles[msg.sender] = "participant";
        emit RoleAssigned(msg.sender, "participant");
    }

    function getRole(address user) public view returns (string memory) {
        return roles[user];
    }
}
```

FIGURE 5.1 – Smart Contract VotingRoles.sol

5.4.2 Conception du contrat Voting.sol

Le contrat Voting.sol est au cœur du système de vote. Il implémente les fonctionnalités principales du processus électoral, y compris la création des scrutins, la soumission des votes et la compilation des résultats.

1. **Création d'un vote** : la fonction `createVote()` permet au facilitateur de définir :
 - un titre et une description du vote,
 - les différentes options soumises au vote,
 - le nombre total de votants attendus.
2. **Soumission des votes** : la fonction `vote(uint256 _voteId, uint256[] memory _optionIndices)` permet à un votant autorisé de sélectionner plusieurs options (selon la méthode du vote par approbation) :
 - Vérification que l'utilisateur n'a pas déjà voté.
 - Incrémentation des compteurs de votes pour chaque option choisie.
 - Suivi du nombre de votants ayant exprimé leur choix.
3. **Fermeture automatique du vote** : le scrutin est clos automatiquement dès que tous les votants attendus ont participé, grâce à une vérification effectuée à la fin de la fonc-

tion `vote()`. Une fois le scrutin clos, aucun vote supplémentaire n'est possible, et les résultats deviennent accessibles.

4. **Consultation des résultats** : la fonction `getVoteResults(uint256 _voteId)` retourne un tableau indiquant le nombre de votes reçus par option, ce qui permet une interprétation claire du résultat final selon la méthode du vote par approbation.

Les figures 5.2, 5.3, 5.5, 5.6 et 5.7 représentent des captures du contrat intelligent `Voting.sol`, illustrant l'implémentation de la logique centrale du système de vote. Cette capture met en évidence la structure des données, les fonctions essentielles (comme la création de scrutin et la soumission du vote), ainsi que les vérifications de sécurité effectuées avant chaque action critique.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
pragma experimental ABIEncoderV2;

contract Voting {
    struct Vote {
        string title;
        string description;
        string[] options;
        uint256 creationTime; // Ajout du temps de création
        address creator;
        bool isActive;
        bool isCompleted; // Nouveau champ pour suivre l'état
    }

    Vote[] public votes;

    mapping(uint256 => mapping(address => bool)) public hasVoted;
    mapping(uint256 => mapping(uint256 => uint256)) public voteCounts;
    mapping(uint256 => address[]) public voters;
    mapping(uint256 => uint256) public expectedVotersCount;
    mapping(uint256 => uint256) public actualVotersCount;

    event VoteCreated(uint256 voteId, string title, address creator);
    event Voted(uint256 voteId, uint256[] optionIndices, address voter);
    event VoteCompleted(uint256 voteId); // Nouvel événement
```

FIGURE 5.2 – Smart contract `Voting.sol` 1

```

function createVote(
    string memory _title,
    string memory _description,
    string[] memory _options,
    uint256 _expectedVoters
) public {
    require(_options.length >= 2, "At least 2 options required");
    require(_expectedVoters > 0, "Expected voters must be positive");

    votes.push(Vote({
        title: _title,
        description: _description,
        options: _options,
        creationTime: block.timestamp,
        creator: msg.sender,
        isActive: true,
        isCompleted: false
    }));

    uint256 newVoteId = votes.length - 1;
    expectedVotersCount[newVoteId] = _expectedVoters;

    emit VoteCreated(newVoteId, _title, msg.sender);
}

```

FIGURE 5.3 – Smart contract Voting.sol 2

```

function vote(uint256 _voteId, uint256[] memory _optionIndices) public {
    require(_voteId < votes.length, "Invalid vote ID");
    require(votes[_voteId].isActive, "Vote is not active");
    require(!votes[_voteId].isCompleted, "Vote is already completed");
    require(!hasVoted[_voteId][msg.sender], "Already voted");

    // Check for duplicate options
    for (uint i = 0; i < _optionIndices.length; i++) {
        for (uint j = i + 1; j < _optionIndices.length; j++) {
            require(_optionIndices[i] != _optionIndices[j], "Duplicate options");
        }
    }

    for (uint i = 0; i < _optionIndices.length; i++) {
        require(_optionIndices[i] < votes[_voteId].options.length, "Invalid option index");
        voteCounts[_voteId][_optionIndices[i]]++;
    }

    hasVoted[_voteId][msg.sender] = true;
    voters[_voteId].push(msg.sender);
    actualVotersCount[_voteId]++;

    // Vérifier si tous les participants ont voté
    if (actualVotersCount[_voteId] >= expectedVotersCount[_voteId]) {
        votes[_voteId].isCompleted = true;
        votes[_voteId].isActive = false;
        emit VoteCompleted(_voteId);
    }
    emit Voted(_voteId, _optionIndices, msg.sender);
}

```

FIGURE 5.4 – Smart contract Voting.sol3

```
function getVotersCount(uint256 _voteId) public view returns (uint256) {
    require(_voteId < votes.length, "Invalid vote ID");
    return voters[_voteId].length;
}

function hasEveryoneVoted(uint256 _voteId) public view returns (bool) {
    require(_voteId < votes.length, "Invalid vote ID");
    return actualVotersCount[_voteId] >= expectedVotersCount[_voteId];
}

function isVoteCompleted(uint256 _voteId) public view returns (bool) {
    require(_voteId < votes.length, "Invalid vote ID");
    return votes[_voteId].isCompleted;
}

function setVoteActive(uint256 _voteId, bool _isActive) public {
    require(_voteId < votes.length, "Invalid vote ID");
    require(msg.sender == votes[_voteId].creator, "Only creator can modify");
    require(!votes[_voteId].isCompleted, "Completed vote cannot be modified");
    votes[_voteId].isActive = _isActive;
}

function getVotesCount() public view returns (uint256) {
    return votes.length;
}
```

FIGURE 5.5 – Smart contract Voting.sol 4

```
function getVote(uint256 _voteId) public view returns (
    string memory title,
    string memory description,
    string[] memory options,
    uint256 creationTime,
    address creator,
    bool isActive,
    bool isCompleted
) {
    require(_voteId < votes.length, "Invalid vote ID");
    Vote storage v = votes[_voteId];
    return (
        v.title,
        v.description,
        v.options,
        v.creationTime,
        v.creator,
        v.isActive,
        v.isCompleted
    );
}
```

FIGURE 5.6 – Smart contract Voting.sol 5

```

function getVoteResults(uint256 _voteId) public view returns (uint256[] memory) {
    require(_voteId < votes.length, "Invalid vote ID");
    require(votes[_voteId].isCompleted, "Vote not completed yet");

    Vote storage v = votes[_voteId];
    uint256[] memory results = new uint256[](v.options.length);

    for (uint256 i = 0; i < v.options.length; i++) {
        results[i] = voteCounts[_voteId][i];
    }

    return results;
}

function getVoters(uint256 _voteId) public view returns (address[] memory) {
    require(_voteId < votes.length, "Invalid vote ID");
    return voters[_voteId];
}

function getExpectedVoters(uint256 _voteId) public view returns (uint256) {
    require(_voteId < votes.length, "Invalid vote ID");
    return expectedVotersCount[_voteId];
}

```

FIGURE 5.7 – Smart contract Voting.sol 6

5.5 Déploiement de la DApp

Une fois les smart contracts développés, l'étape suivante consiste à déployer le smart contract.

5.5.1 Outils nécessaires pour le fonctionnement de la DApp

Pour exécuter la DApp en local, certaines dépendances doivent être installées :

- **Node.js** : environnement d'exécution JavaScript côté serveur, nécessaire pour lancer un serveur local et compiler le projet.
- **npm (Node Package Manager)** : outil de gestion des dépendances permettant d'installer les bibliothèques utilisées dans le projet (comme Web3.js).
- **Navigateur web moderne** (ex. Google Chrome ou Firefox) pour l'affichage de l'interface utilisateur et le bon fonctionnement de l'extension MetaMask.
- **MetaMask** : Extension de navigateur permettant de se connecter à la blockchain, de signer les transactions, et d'interagir avec les contrats intelligents via l'interface.

1. Installation de Node.js et npm

- Télécharger et installer la version LTS (Long Term Support) de Node.js depuis le site officiel : <https://nodejs.org>
- Une fois l'installation terminée, ouvrir un terminal (ou une invite de commande) et vérifier que Node.js et npm sont correctement installés en exécutant les commandes suivantes :

```

PS C:\Users\ferre\OneDrive\Bureau\M2_ReSys\PFE_Documentation\Code> node -v
v20.17.0
PS C:\Users\ferre\OneDrive\Bureau\M2_ReSys\PFE_Documentation\Code> npm -v
11.2.0

```

FIGURE 5.8 – Node.js et npm version

2. truffle et ganache :

Pour installer Truffle, on exécute la commande suivante dans le terminal :

```
1 npm install -g truffle
```

On vérifie la version installée pour s'assurer du bon fonctionnement :

```
PS C:\Users\ferre\OneDrive\Bureau\W2_ReSys\PFE_Documentation\Code> truffle version
Truffle v5.11.5 (core: 5.11.5)
Ganache v7.9.1
Solidity - 0.8.21 (solc-js)
Node v20.17.0
Web3.js v1.10.0
```

FIGURE 5.9 – Truffle version

Maintenant, on installe ganache via le paquet téléchargé depuis le site officiel : <https://trufflesuite.com/ganache/>. Après l'installation, on lance l'application et on crée un nouveau projet de type *new workspace*. On note les comptes et adresses générés pour les tests.

3. Extension MetaMask :

- Installer l'extension MetaMask depuis le site officiel <https://metamask.io/> dans votre navigateur (Chrome, Firefox, Brave, etc.).
- Créer un nouveau portefeuille ou importer un portefeuille existant à l'aide d'une phrase de récupération (seed phrase).

5.5.2 Déploiement des contrats

Les étapes suivantes décrivent le processus de déploiement :

- Créer un script de migration dans le dossier migrations/, par exemple `2_deploy.js`, qui contient les instructions pour déployer les contrats intelligents. Les figures 5.10 et 5.11 représentent les scripts de migration du contrat `votingRoles.sol` et du contrat `voting.sol`.

```
const VotingRoles = artifacts.require("VotingRoles");

module.exports = function (deployer) {
  // Déployer avec un facilitateur et une liste vide de participants
  const facilitatorAddress = "0xd32d51992ED703CcE9D3a1a50253A609c631dD2C";
  const participants = [
    "0xb332D4cA7f38B2f79d48A4FbC821FD87053a0F98",
    "0xad5Ce40152bef87f8F19C373C3a430c24Ed134D5",
    "0x23aC51B60E3eb78c9345777C32E2823eb5716eF9",
    "0x7915abb91cE16a86a69cc08fb2f79Bbd6b77C77b",
    "0x22B524F3b9a4a62b9e7990CDea66cA27849b1C9b",
    "0x2C224877fEc7c841943f33D6d7A8b358B28B6829",
    "0x81Bd5A3d5dD4Af66946cab7496aAf9C0779d15C4",
    "0xcAb47777c60a7d0D64446D01562e03f73Fd1f41c"
  ]; // Liste vide pour commencer

  deployer.deploy(VotingRoles, facilitatorAddress, participants);
};
```

FIGURE 5.10 – Script de migration du contrat `votingRoles.sol`

```
const Voting = artifacts.require("voting");

module.exports = function (deployer) {
  ...
  deployer.deploy(Voting);
};
```

FIGURE 5.11 – Script de migration du contrat voting.sol

- Lancer la migration vers la blockchain locale (Ganache) avec la commande suivante :

```
truffle migrate --network development
```

FIGURE 5.12 – Commande de migration

- On vérifie dans Ganache que les contrats apparaissent bien dans l'onglet "Contracts", preuve de leur déploiement réussi.

5.6 Développement de l'interface utilisateur (UI)

L'interface utilisateur (UI) constitue l'élément clé d'un système de vote décentralisé, car elle permet aux votants et au facilitateur d'interagir avec les smart contracts de manière intuitive, accessible et sécurisée. Dans le cadre de ce projet, l'interface a été développée à l'aide des technologies web fondamentales : HTML pour la structure, CSS pour la mise en forme, et JavaScript pour la logique dynamique et l'intégration avec la blockchain. Ce choix s'est appuyé sur la volonté de proposer une interface légère, facile à déployer et compatible avec les navigateurs modernes, tout en restant parfaitement fonctionnelle dans le contexte Web3.

Fonctionnalités clés de l'interface

1. Connexion via MetaMask :

La première étape consiste à connecter l'utilisateur via MetaMask, ce qui permet de récupérer son adresse Ethereum et de vérifier son rôle (facilitateur ou participant). Cette approche élimine toute dépendance à un système centralisé de gestion des identités.

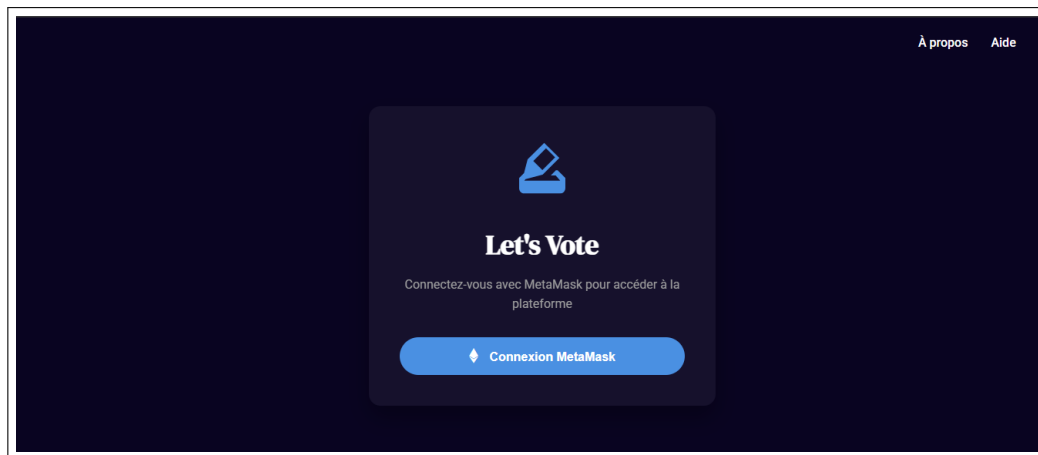


FIGURE 5.13 – Interface de connexion

2. Création d'un vote :

Le facilitateur peut créer un nouveau vote grâce à un formulaire HTML contenant :

- Le titre et la description du scrutin.
- Les options proposées.
- Le nombre total de votants attendus.

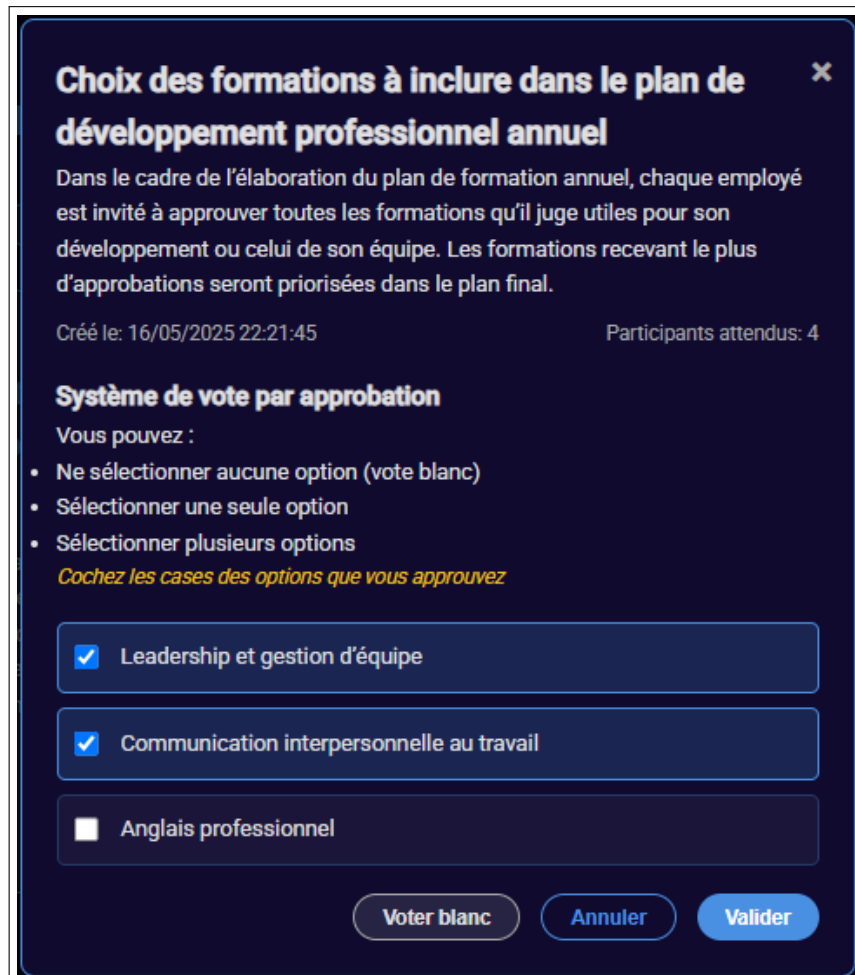
Les données sont ensuite envoyées au smart contract via Web3.js pour initialiser le vote.

The image shows a dark-themed form titled 'Créer un nouveau vote' with a close button (X) in the top right corner. The form contains several input fields: 'Titre du vote' (a single-line text box), 'Description' (a multi-line text area), 'Nombre de votants attendus' (a single-line text box), and 'Options' (two single-line text boxes labeled 'Option 1' and 'Option 2'). Below the options is a blue button labeled 'Ajouter une option'. At the bottom right of the form are two buttons: 'Annuler' and 'Créer'.

FIGURE 5.14 – Interface de création de vote

3. Participation au vote :

Les votants peuvent consulter la liste des sessions actives et exprimer leur choix en sélectionnant une ou plusieurs options (vote par approbation). L'interface vérifie si l'utilisateur a déjà voté et empêche toute tentative multiple.



The screenshot shows a voting interface titled "Choix des formations à inclure dans le plan de développement professionnel annuel". It includes a description of the voting process, the creation date (16/05/2025 22:21:45), and the number of participants (4). The voting system is "Système de vote par approbation", allowing users to select one or more options. The options are "Leadership et gestion d'équipe", "Communication interpersonnelle au travail", and "Anglais professionnel". The first two are selected with checkboxes, and the third is not. At the bottom, there are buttons for "Voter blanc", "Annuler", and "Valider".

Choix des formations à inclure dans le plan de développement professionnel annuel ✕

Dans le cadre de l'élaboration du plan de formation annuel, chaque employé est invité à approuver toutes les formations qu'il juge utiles pour son développement ou celui de son équipe. Les formations recevant le plus d'approbations seront prioritaires dans le plan final.

Créé le: 16/05/2025 22:21:45 Participants attendus: 4

Système de vote par approbation

Vous pouvez :

- Ne sélectionner aucune option (vote blanc)
- Sélectionner une seule option
- Sélectionner plusieurs options

Cochez les cases des options que vous approuvez

☒ Leadership et gestion d'équipe

☒ Communication interpersonnelle au travail

☐ Anglais professionnel

Voter blanc Annuler Valider

FIGURE 5.15 – Interface de création de vote

4. Consultation des résultats :

Après la fermeture du scrutin, l'interface affiche les résultats finaux sous forme de graphiques, calculés selon les données retournées par le smart contract.



FIGURE 5.16 – Interface de consultation des résultats

5.7 Conclusion

Ce chapitre a présenté l'implémentation technique de notre application décentralisée (DApp) pour la gestion des votes au sein d'un Group Decision Support System (GDSS). À travers une approche structurée, nous avons détaillé les choix technologiques, l'architecture des smart contracts, leur déploiement sur une blockchain locale, ainsi que le développement de l'interface utilisateur.

Conclusion générale

Dans un contexte où les systèmes de prise de décision collective évoluent vers plus de transparence, de sécurité et de décentralisation, cette recherche s'est inscrite dans une démarche innovante visant à concevoir et implémenter un système de vote basé sur la blockchain, utilisant des smart contracts pour garantir l'intégrité du processus électoral. En s'appuyant sur les technologies clés du Web3 (MetaMask, Web3.js, Ethereum), ce projet a permis de développer une solution fonctionnelle, sécurisée et transparente, applicable à divers contextes organisationnels.

Le système proposé s'intègre dans une approche GDSS (Group Decision Support System), en facilitant la collecte d'opinions individuelles et en assurant une traçabilité fiable des résultats. La méthode de vote par approbation a été choisie pour sa simplicité et sa pertinence dans des situations de choix collectif, comme le vote pour des projets, des décisions internes ou des préférences de groupe.

Les résultats obtenus montrent que la blockchain, bien qu'encore jeune en termes d'adoption institutionnelle, offre des bases solides pour repenser les mécanismes traditionnels de prise de décision collective. Notre système illustre parfaitement comment ces technologies peuvent être appliquées à des scénarios concrets, tels que la sélection de projets communautaires, l'organisation d'événements internes ou même des consultations citoyennes.

Dans la continuité de ce travail, plusieurs perspectives d'amélioration peuvent être envisagées afin de renforcer la pertinence et la robustesse du système proposé. Tout d'abord, l'ajout d'un seuil d'acceptation au sein du vote par approbation constituerait une avancée importante. Ce seuil, défini par un pourcentage minimal de votes positifs (par exemple 50 %), permettrait de valider une proposition uniquement si elle recueille un soutien suffisant. Dans le cas contraire, le système relancerait automatiquement un nouveau vote.

Par ailleurs, pour accompagner des processus de décision plus complexes, il serait pertinent d'intégrer des méthodes de vote préférentiel, telles que la méthode de Condorcet ou le vote de Borda, permettant aux participants de classer les options par ordre de préférence. Cette extension offrirait une représentation plus nuancée des choix collectifs et élargirait le champ d'application du système à des scénarios décisionnels plus élaborés.

En somme, ce travail démontre que les technologies blockchain, combinées à une interface utilisateur accessible et à une modélisation rigoureuse, sont capables de transformer profondément la manière dont les décisions collectives sont prises.

Bibliographie

- [1] D. K. Schneider, *Modelisation de la demarche du decideur politique dans la perspective de l'intelligence artificielle*. Thèse de doctorat en sciences économiques et sociales, Université de Genève, Faculté des sciences économiques et sociales, 2000. Mention science politique.
- [2] B. Roy and D. Bouyssou, *Aide multicritère à la décision : méthodes et cas*. Paris, France : Economica, 1993.
- [3] P. Lévine and J.-C. Pomerol, *Systèmes interactifs d'aide à la décision et systèmes experts*. Hermès, 1990.
- [4] D. Trentesaux, *Conception d'un système de pilotage distribué, supervisé et multicritère pour les systèmes automatisés de production*. PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 1996.
- [5] B. Roy, *Méthodologie multicritère d'aide à la décision*. Economica, 1985.
- [6] H. A. Simon, *Science of Management Decision*. New Jersey : Prentice-Hall, 1977.
- [7] E. Turban, J. E. Aronson, and T.-P. Liang, *Decision Support Systems and Intelligent Systems*. Upper Saddle River, NJ : Pearson Education, seventh ed., 2005. With contributions by Richard V. McCarthy.
- [8] P. G. W. Keen and M. S. Scott-Morton, "Decision support systems : An organizational perspective," 1978.
- [9] J. D. C. Little, "Models and managers : The concept of a decision calculus," *Management Science*, vol. 16, no. 8, pp. B-466-B-485, 1970.
- [10] R. H. Sprague and E. D. Carlson, *Building Effective Decision Support Systems*. Englewood Cliffs, NJ : Prentice-Hall, Inc., 1982.
- [11] A. Adla, *Aide à la Facilitation pour une prise de Décision Collective : Proposition d'un Modèle et d'un Outil*. PhD thesis, Université Toulouse III - Paul Sabatier, Toulouse, France, 2010.
- [12] M. ER, "Decision support systems : A summary, problems, and future trends," *Journal of Decision Systems*, vol. 9, no. 2, pp. 123-145, 2000.
- [13] G. Phillips-Wren, "Ai tools in decision making support systems : A review," *International Journal of Artificial Intelligence Tools*, vol. 21, p. 1240005, April 2012.
- [14] A. Tariq and K. Rafi, "Intelligent decision support systems - a framework," *Information and Knowledge Management*, vol. 2, no. 6, 2012.
- [15] T. Bui and A. Schenker-Wicki, "445 co-op : Un système d'aide à la décision de groupe pour les décisions stratégiques de l'entreprise."
- [16] H. A. Simon, *Administrative Behaviour : A Study of Decision-Making Process in Administrative Organizations*. New York, USA : Free Press, 1997.

- [17] F. Laborie, *Le concept de salle de décision collective et son application aux processus complexes EADS*. PhD thesis, Université Paul Sabatier, Toulouse (France), 2006.
- [18] R. B. Gallupe, W. H. Cooper, J. S. Valacich, L. M. Bastianutti, and J. F. Nunamaker, “Electronic brainstorming and group size,” *Academy of Management Journal*, vol. 35, no. 2, pp. 350–369, 1992.
- [19] G. DeSanctis and R. B. Gallupe, “A foundation for the study of group decision support systems,” *Management Science*, vol. 33, no. 5, pp. 589–609, 1987.
- [20] J. Nunamaker, A. Dennis, J. Valacich, and D. Vogel, “Information technology for negotiating groups : Generating options for mutual gain,” *Management Science*, vol. 37, no. 10, pp. 1325–1346, 1991.
- [21] J. F. N. Jr., R. O. Briggs, D. D. Mittleman, D. R. Vogel, and P. A. Balthazard, “Lessons from a dozen years of group support systems research : A discussion of lab and field findings,” *Journal of Management Information Systems*, vol. 13, no. 3, pp. 163–207, 1997.
- [22] G. M. Marakas, *Decision Support Systems in the 21st Century*. Upper Saddle River, NJ : Prentice Hall, 2003.
- [23] A. L. Delbecq, A. H. Van de Ven, and D. H. Gustafson, *Group Techniques for Program-ming Planning*. Wisconsin, USA : Green Briar Press, 1986.
- [24] Larousse, *Dictionnaire de la langue française*.
- [25] E. Pacuit, “Voting Methods,” in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta and U. Nodelman, eds.), Metaphysics Research Lab, Stanford University, Summer 2024 ed., 2024.
- [26] K.-E. Cheng and F. P. Deek, “Voting methods and information exchange in group support systems,” in *Proceedings of an academic conference (unspecified)*, 2006. Source : DBLP, Conference Paper.
- [27] B. J. C., “M’emoire sur les’elections au scrutin,” *Histoire de l’Acad’emie Royale des Sciences*.
- [28] M. J. A. N. C. Condorcet, *Essai sur l’application de l’analyse à la probabilité des décisions rendues à la pluralité des voix*. 1785.
- [29] A. Bassi, “Voting systems and strategic manipulation : An experimental study,” Sept. 2008.
- [30] M. Balinski and R. Laraki, “Jugement majoritaire vs. vote majoritaire (via les présidentielles 2011-2012),” tech. rep., CNRS, École Polytechnique, Département d’Économie, Route de Saclay, 91128 Palaiseau Cedex, France, 2011.
- [31] K. J. Arrow, *Social Choice and Individual Values*. Yale University Press, 2012.
- [32] S. Nakamoto, “Bitcoin : A peer-to-peer electronic cash system,” 2008. Email : satoshi@gmx.com.
- [33] Uchange.co, “Comprendre la blockchain : Anticiper le potentiel de disruption de la blockchain sur les organisations,” 2016.
- [34] National Institute of Standards and Technology (NIST), “Blockchain technology overview,” Tech. Rep. NISTIR 8202, U.S. Department of Commerce, 2018.
- [35] R. Beck, M. Avital, M. Rossi, and J. B. Thatcher, “Blockchain technology in business and information systems research,” *Business & Information Systems Engineering*, 2017. Published online : 15 November 2017.

- [36] A. Richard, “Blockchain et désintermédiation : un retour des intermédiaires ?” <https://sommets-patrimoine-performance.com>, 2021. Table ronde au Sommet du Patrimoine et de la Performance, Paris, 12 juillet.
- [37] EXED, “La blockchain : explorer les nouvelles frontières de la confiance et de la transparence,” 2024. Consulté le 13 avril 2025.
- [38] J. Sedlmeir, J. Lautenschlager, G. Fridgen, and N. Urbach, “The transparency challenge of blockchain in organizations,” *Business Information Systems Engineering*, 2022.
- [39] M. Pignel, “La technologie blockchain : Une opportunité pour l’Économie sociale ?,” 2019. Notes d’Analyse Juin 19, Économie sociale.
- [40] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, “Blockchain security : A survey of techniques and research directions,” *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2490–2510, 2022.
- [41] C. Boura, *Analyse de fonctions de hachage cryptographiques*. Theses, Université Pierre et Marie Curie - Paris VI, Dec. 2012.
- [42] S. Zhai, Y. Yang, J. Li, C. Qiu, and J. Zhao, “Research on the application of cryptography on the blockchain,” *Journal of Physics : Conference Series*, vol. 1168, p. 032077, feb 2019.
- [43] L. Leloup, *Blockchain : La révolution de la confiance*. Eyrolles, 1st ed., 2017. Original work published 2017.
- [44] J. Us-off, Z. Moha-mad, and M. Anuar, “A review : Consensus algorithms on blockchain,” *Journal of Computer and Communications*, vol. 10, pp. 37–50, 2022. Accessed : April 26, 2025.
- [45] P. Hooda, “Proof of work (pow) consensus,” 2022. Accessed : April 25, 2025.
- [46] J. Jung, M. Islam, and H. In, “Proof of work with random selection (powr) : An energy saving consensus algorithm with proof of work and the random selection function,” *Sustainability*, vol. 16, no. 21, p. 9342, 2024. Academic Editor : Francesco Nocera.
- [47] M. Salimitari and M. Chatterjee, “A survey on consensus protocols in blockchain for iot networks,” *Department of Computer Science, University of Central Florida*, 2024.
- [48] U. Jafar, M. J. A. Aziz, and Z. Shukur, “Blockchain for electronic voting system—review and open research challenges,” *Sensors*, vol. 21, no. 17, 2021.
- [49] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, “An overview on smart contracts : Challenges, advances and platforms,” *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الاجتماعية و الإنسانية

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،
الطالب(ة): فرعون أسماء رقم التسجيل الجامعي:
.....202037033619.....
الحامل لبطاقة التعريف الوطنية رقم: 110020977000550006 والصادرة
بتاريخ:28/02/2022.....
عن...حجاج/مستغانم.....
المسجل بكلية العلوم الدقيقة والإعلام الآلي / قسم ...رياضيات واعلام
آلي.....
شعبةاعلام آلي..... / التخصص.. شبكات
وأنظمة.....

والمكلف بإنجاز مذكرة ماستر بعنوان:

Utilisation de la blockchain pour la gestion de vote dans un GDSS

.....
أصرح بشرفي أنني ألتزم
بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث
المذكور أعلاه .

التاريخ:25/06/2025.....

إمضاء المعني



* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس - مستغانم -
كلية العلوم الاجتماعية و الإنسانية

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): بزخامي اية رقم التسجيل الجامعي:

.....202037033606

الحامل لبطاقة التعريف الوطنية رقم:110020981000810006 والصادرة بتاريخ:

.....2022/06/08.....

عن حجاج / مستغانم

المسجل بكلية العلوم الدقيقة والإعلام الآلي / قسم : رياضيات و اعلام الي

شعبة .. اعلام الي / التخصص : شبكات و أنظمة

والمكلف بإنجاز مذكرة ماستر بعنوان:

Utilisation des Blockchain pour la gestion des votes au sein d'un GDSS

أصرح بشرفي أنني ألتمزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ 2025/06/24

إمضاء المعني



* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.