UNIVERSITÉ DU QUÉBEC

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

ÉTUDE DE L'IMPACT DE LA TECHNOLOGIE BLOCKCHAIN SUR L'EFFICACITÉ DE LA RÉPONSE À LA DEMANDE ÉLECTRIQUE DANS LA PROVINCE DU QUÉBEC

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA

MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR
SAMIR FRANCK AMADOU COULIBALY

NOVEMBRE 2025

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MAÎTRISE EN GÉNIE ÉLECTRIQUE (M. Sc. A.)

Direction de recherche :

Prof. Adam W. Skorek PhD, M. Sc. Eng., Fellow ICI., IEEE Life Fellow	Directeur de recherche

Jury d'évaluation

Prof. Hamidou Tembine, PhD	Évaluateur interne
Prof. Jonathan Roy, PhD, B.Eng.	Évaluateur externe
Prof. Adam W. Skorek, PhD, M. Sc. Eng., Fellow ICI., IEEE Life Fellow	Évaluateur interne

Remerciements

Je tiens à exprimer à travers ses mots, ma profonde gratitude et reconnaissance:

À mes parents, pour leur amour inconditionnel et leur soutien lors de ma formation à l'Université du Québec à Trois-Rivières.

À mon directeur de recherche, le Prof. Skorek, Adam Waldemar, PhD, M. Sc. Eng., Fellow ICI., IEEE Life Fellow, et aux évaluateurs de ce document pour leurs apports significatifs à l'amélioration de ce travail.

À l'ensemble des professeurs du département de génie électrique et de génie informatique de l'Université du Québec à Trois-Rivières pour l'attention et les efforts portés sur notre formation.

À mon frère Mickaël pour son amour fraternel et ses conseils.

À mon oncle Drissa pour ses conseils et son aide durant mon séjour d'étude au Canada

À mes amis Vivaldi, Rasmané, Akim, Brice, Simon, Coretta et tant d'autres pour leur soutien, conseils et moments passés ensemble.

À ma très précieuse amie Guillène pour son soutien et ses conseils.

À toutes personnes ayant contribué de près ou de loin à l'accomplissement de ce travail.

Pour terminer, je dédie ce travail à la mémoire de ma grande mère maternelle.

Résumé

L'effacement de la consommation électrique du point de vue du consommateur est une pratique qui consiste pour ledit consommateur à diminuer ou effacer sa consommation électrique à des périodes définies. Cette pratique a donné naissance à des programmes, très importants pour la réponse à la demande électrique dans la province du Québec. Étant donné cette importance, il est nécessaire de continuellement explorer des pistes d'amélioration. La technologie de la chaîne de blocs permettant d'améliorer la protection des données semble être un excellent choix pour l'amélioration de ces programmes notamment face au nombre croissant de cyberattaques encouru par Hydro-Québec ces dernières années. Cependant ce choix pose donc la problématique de son impact réel sur la réponse à la demande électrique. Ce travail a permis d'évaluer l'impact de cette technologie sur un programme d'effacement de la consommation électrique au Québec : « le demand response option ». Plus spécifiquement, il a permis la mise en place et la simulation du programme qui à travers un contrat intelligent gère l'ensemble des étapes essentielles. De plus, une comparaison entre le modèle actuellement utilisé et celui proposé a été réalisée. Il ressort de cette comparaison que le modèle proposé présente des avantages tels qu'une automatisation efficace des tâches et une transparence accrue dans le traitement des données. À titre d'exemple, l'automatisation des tâches grâce à la blockchain pourrait permettre un traitement de données plus rapide : un minimum de 90 fois moins de temps environ. Néanmoins, des limites peuvent être trouvées telles que la dépendance de ce modèle à un serveur ou oracle fonctionnel et non corrompu.

Mots clés : Blockchain, Province du Québec, Effacement de la consommation électrique, demande électrique

Table des Matières

Liste des Figures

Liste des Tableaux

Liste des Abréviations

CAN : Canada

dApps : decentralized Applications

DSM : Demand Side Management ou Maîtrise de la demande du côté du consommateur

EVM : Etherum Virtual Machine

FD : Flux de Données

Geth : Go Etherum

IA : Intelligence Artificielle

IDE : Integrated Development Environment ou Environnement de Développement Intégré

IoT : Internet of Things

kW : kiloWatt

N° ou # : Numéro

N/A : Aucune réponse

PoA; Proof of Authority ou preuve d'autorité

PoS : Proof of Stake ou preuve de travail

PoW : Proof of Work ou preuve de travail

SHA : Secure Hash Algorithm

STRIDE : Spoofing (Usurpation), Tampering (Altération de données), Repudiation (Négation d'action), Denial of service (Indisponibilité de service) and Elevation of Privilege (Élévation des privilèges)

SWOT : Strengths, Weaknesses, Opportunities and Threarts ou Forces, Faiblesses, Opportunités et Menaces

Chapitre 1 - Introduction

1.1 Mise en contexte

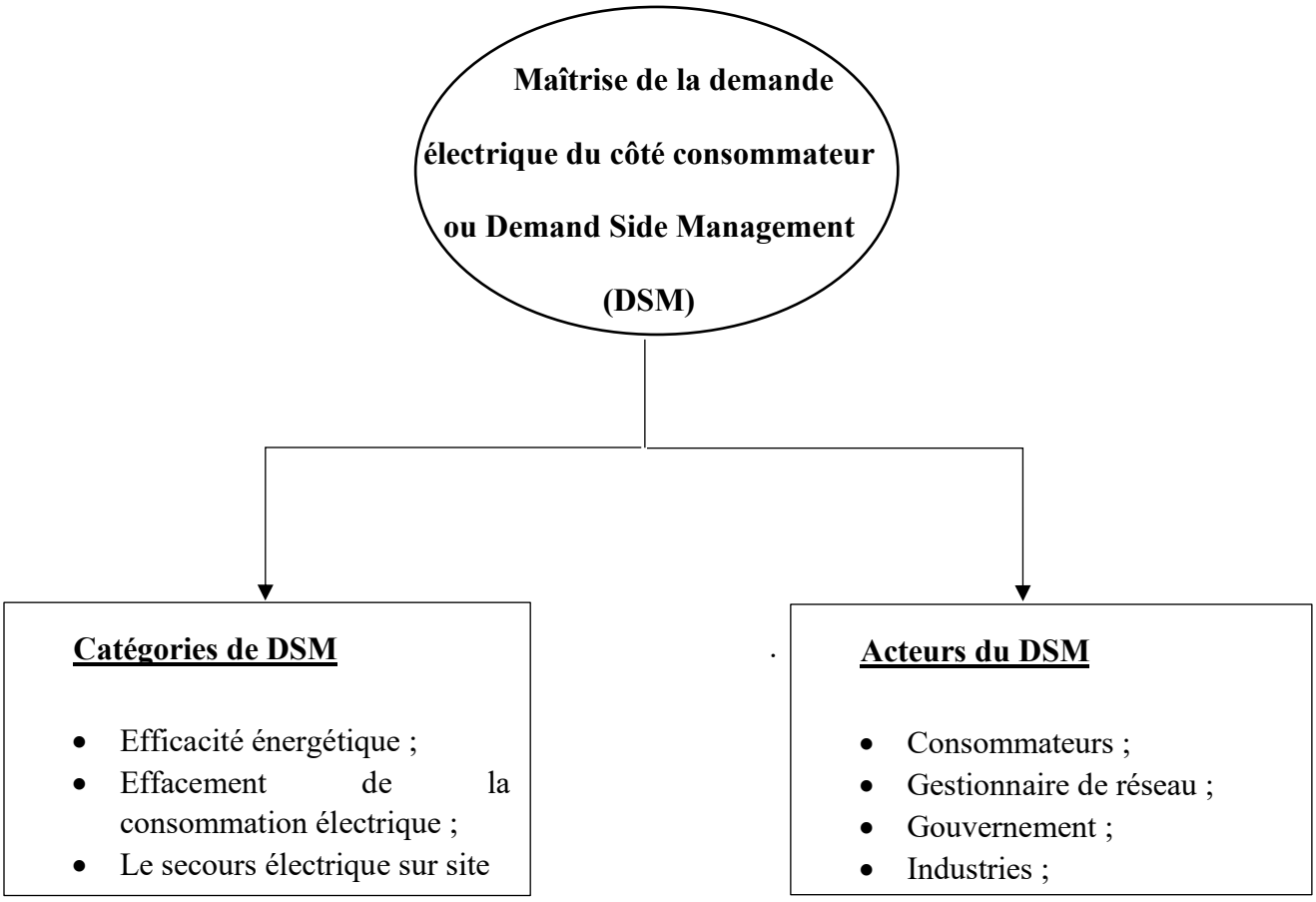
Dans le contexte actuel de réchauffement planétaire global, l'intégration des énergies renouvelables (photovoltaïques et éoliennes notamment) dans les réseaux ou systèmes électriques a notamment pris de l'ampleur ces dernières années. Ces technologies, caractérisées par leur intermittence, imposent aux gestionnaires de réseaux électriques le développement de stratégies afin de satisfaire la demande [1]. De plus, il faut ajouter à cela l'effet rebond des équipements électriques [2, 3] et l'utilisation croissante des véhicules électriques, ce qui soulève la problématique de gestion de la charge engendrée par lesdits véhicules. Conséquence de ce paysage, la demande électrique est de plus en plus croissante et difficile à équilibrer vis-à-vis de l'offre. L'une des solutions à ces problématiques est l'emploi de stratégies de gestion de la demande électrique telle que la maîtrise de la demande électrique du côté du consommateur.

La maîtrise de la demande électrique du côté du consommateur désigne les technologies, actions et programmes pris du côté consommation du compteur d'énergie, pour une meilleure gestion ou une diminution de la consommation énergétique dans le but de réduire les dépenses liées au système d'énergie ou de contribuer à la mise en œuvre de politiques énergétiques telles que la réduction des émissions ou l'équilibre entre l'offre et la demande [4]. Elle est usuellement subdivisée en trois catégories : l'efficacité énergétique, le secours électrique sur site et l'effacement de la consommation électrique [5, 6]

comme présenté dans la Figure 1-1. Parmi ces catégories récemment énumérées, un accent particulier peut être mis sur l'effacement de la consommation électrique. L'effacement de la consommation électrique est une pratique visant à réduire, volontairement ou involontairement, la consommation électrique à des périodes définies ou non. La justification de son importance peut être trouvée dans la littérature. En effet, selon les auteurs [7], l'effacement de la consommation électrique est un moyen efficace pour éliminer les fluctuations de puissance électrique et satisfaire les contraintes liées aux besoins de puissance dans les systèmes électriques. Plus encore, il est largement reconnu comme étant un élément critique pour l'avenir des réseaux électriques intelligents et optimisés comprenant une large variété de sources d'énergies renouvelables et traditionnelles, tant centralisé que distribué pour les auteurs [5].

C'est donc à la vue de son importance que notre cas d'étude, la province du Québec au Canada, à travers sa société d'électricité, applique l'effacement de la consommation électrique pendant les périodes de forte demande.

Parallèlement à ce qui précède, il est à noter qu'avec l'avènement récent des nouvelles technologies telles que l'Internet des objets (IoT), la technologie blockchain et l'intelligence artificielle (IA), il est désormais possible d'améliorer l'efficacité des programmes d'effacement de la consommation électrique avec la technologie blockchain notamment, comme le suggère plusieurs études telles que celles réalisées par les auteurs [1], [8] et [7].



1.1 ***Définition de la maîtrise de demande électrique du côté consommateur. [6, 9]***

1.2 Problématique

L'effacement de la consommation électrique est particulièrement avantageux pour la province du Québec. Ces avantages peuvent être présentés sous deux principaux aspects complémentaires. En effet, dans un premier temps, l'effacement de la consommation électrique permet de maîtriser la demande électrique croissante enregistrée dans la province, notamment en période hivernale et ainsi la satisfaire, tout en évitant les délestages (type d'effacement de la consommation électrique dont le recours n'est pas souhaitable). Enfin, il peut permettre d'envisager la vente d'un surplus d'électricité, conséquence de cette maîtrise, aux provinces et territoires voisins (la Nouvelle-Écosse, île du Prince Edward, le Nouveau-Brunswick, l'Ontario) et aux États-Unis (l'État de New York et les États dits de la Nouvelle Angleterre notamment) qui ne bénéficient pas des mêmes avantages que le Québec en matière de disponibilité des ressources énergétiques flexibles (hydroélectricité et éolien) [10]. Afin d'améliorer la sécurité des données, des programmes d'effacement de la consommation électrique et dans le même temps accroître le niveau de confiance entre les parties prenantes, la technologie blockchain est perçue comme une piste de solutions dans la littérature. Par sa particularité et en fonction de son domaine d'application, la blockchain permet d'automatiser (contrats intelligents) les tâches, de décentraliser l'accès, la gestion des données, d'améliorer la confidentialité et la sécurité de ces dernières également. Ces atouts de la technologie blockchain permettent de répondre aux défis posés par la gestion centralisée des programmes d'effacement de la consommation électrique [7] et de réduire les coûts. Néanmoins, il existe très peu d'applications de la technologie blockchain pour ce type de programme dans le contexte québécois, ce qui soulève la problématique de l'évaluation de ladite approche.

1.3 Objectif

Ce projet de recherche a pour objectif principal de proposer une évaluation de l'efficacité de l'application de la technologie blockchain sur un programme d'effacement de la consommation électrique avec collaboration des consommateurs dans la province du Québec.

Plus spécifiquement, il s'agira d'abord de mettre en place un système de gestion du programme d'effacement de la consommation électrique utilisant la technologie blockchain, ensuite procéder à son évaluation avec le programme actuellement proposé par Hydro-Québec sur la base de critères pertinents.

1.4 Méthodologie

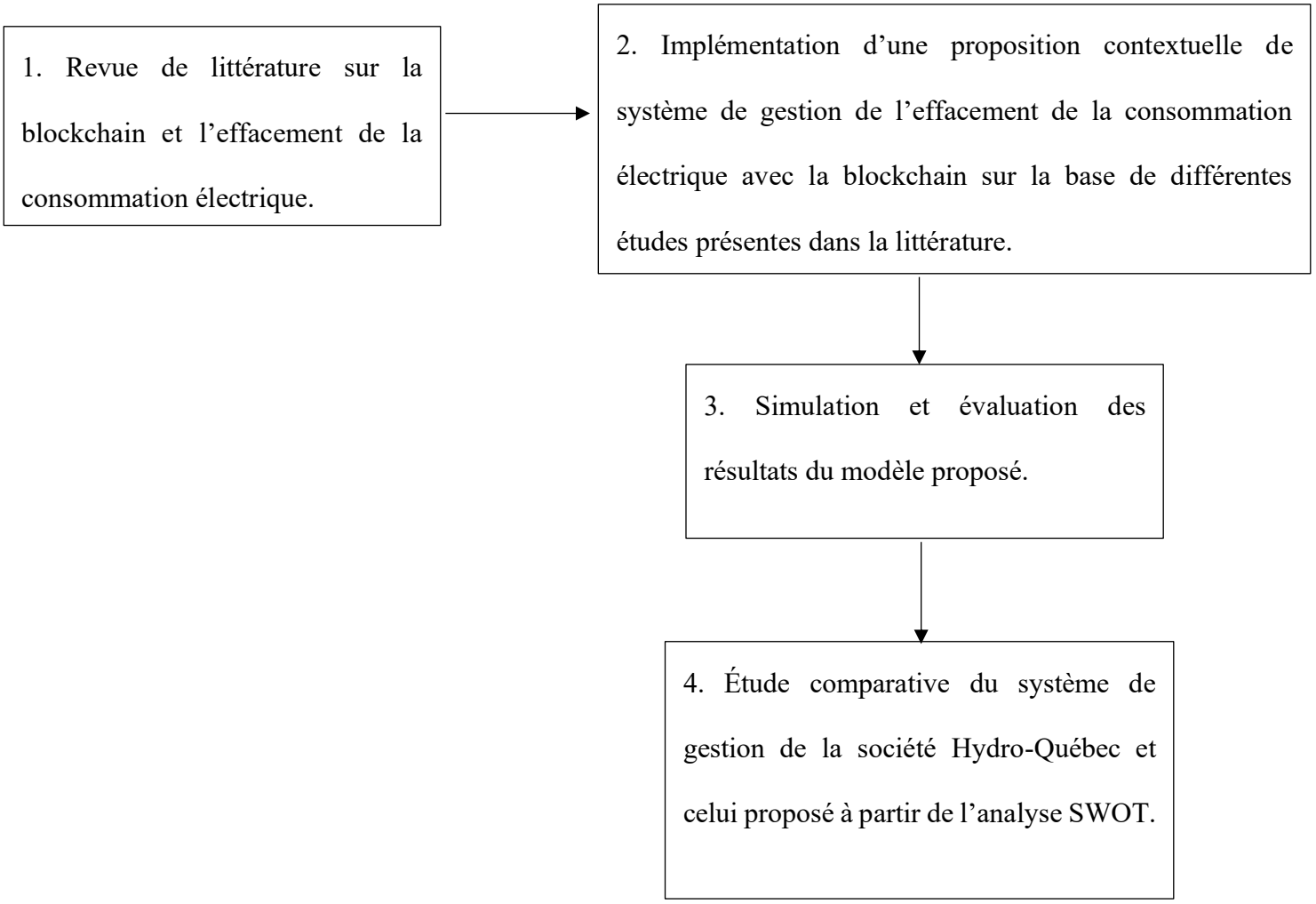
Ces travaux ont pour intérêt de constituer une phase d'évaluation afin d'apprécier les possibilités d'implémentation à plus grande échelle de la technologie Blockchain sur le programme d'effacement de la consommation électrique. D'un point de vue méthodologique, ce travail sera structuré comme suit :

Dans un premier temps, une revue de littérature des articles sur l'effacement de la consommation électrique et la technologie Blockchain sera réalisée afin de faire ressortir l'état de l'art sur ses différents sujets. L'objectif de cette partie est de présenter les définitions des termes majeurs de ce projet de recherche et poser les bases qui permettront d'effectuer des choix pertinents pour l'implémentation du système de gestion de l'effacement. Pour se faire, une analyse comparative des différentes études retrouvées dans la littérature sera réalisée.

Dans un second temps, il sera mis en place, un système de gestion du programme d'effacement de la consommation électrique utilisant la technologie blockchain. Pour cela,

une sélection des caractéristiques essentielles et pertinentes pour l'étude d'une chaîne de blocs sera présentée à travers la littérature.

Enfin, il sera réalisé une étude comparative du modèle de gestion qui sera proposé et celui actuellement en utilisation par la société Hydro-Québec. L'objectif de cette comparaison étant de faire ressortir les avantages et les inconvénients de chaque système, l'analyse STRIDE et l'outil d'analyse stratégique SWOT seront employés à cette fin.



1.2 ***Méthodologie du projet de recherche. (Source : S. Franck A. Coulibaly)***

1.5 Structure du mémoire

Ce document est constitué de cinq chapitres. Le premier présente le contexte, la problématique, les objectifs et la méthodologie de travail. Dans le second chapitre, des définitions des mots clés du projet seront données et une bibliographie des études similaires présentent dans la littérature sera réalisée. Dans le troisième chapitre, la méthodologie, la structure du système de gestion et les résultats de simulation seront présentés et commentés. Le quatrième chapitre portera sur une évaluation des systèmes de gestion des programmes

d’effacement de la consommation électrique de la société Hydro-Québec et celui implémenté. Pour finir, le cinquième chapitre intitulé conclusion générale, résumera et conclura l’ensemble des travaux réalisés lors de ce projet de recherche.

Chapitre 2 - Revue de littérature sur l'effacement de la consommation électrique et la technologie blockchain

Dans ce chapitre, il est présenté les définitions des différents mots-clés de ce projet de recherche, en plus de l'état de l'art sur la création et l'implémentation de programmes d'effacement de la consommation électrique avec la technologie blockchain.

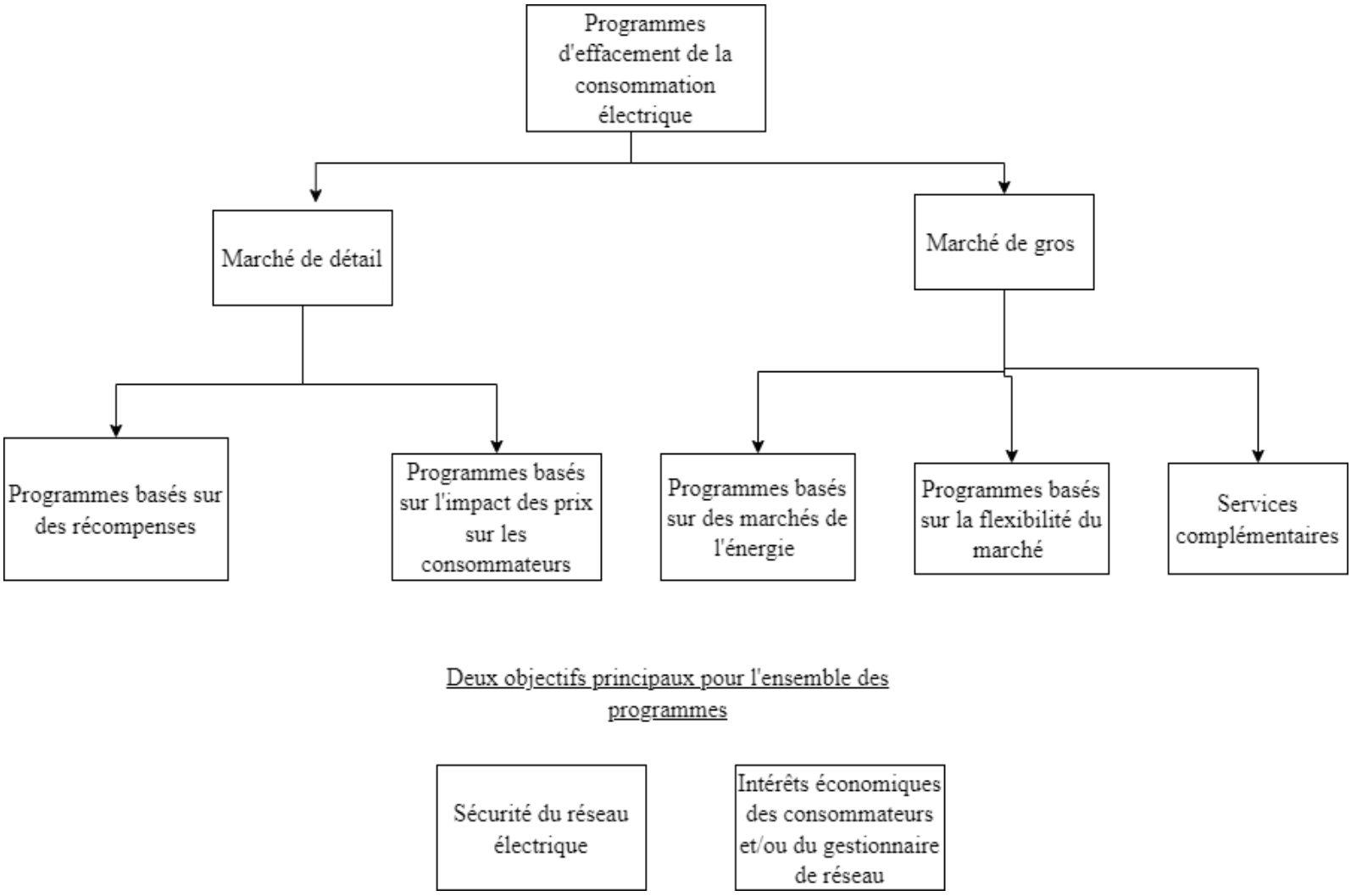
2.1 Effacement de la consommation électrique

2.1.1 Définition

L'effacement de la consommation électrique, ou **lissage de la courbe de charge électrique par pilotage de la demande**, peut se définir comme l'ensemble des moyens (programmes, produits et prix de l'électricité) mis en œuvre afin d'avoir un impact sur la forme du profil de consommation électrique d'une zone donnée [10]. Pour les gestionnaires de réseau électrique, les programmes d'effacement de la consommation électrique vont de la collaboration avec les consommateurs au délestage programmé ou non (blackout). L'effacement de la consommation électrique présente des avantages à la fois pour les consommateurs et le gestionnaire de réseau. À court terme, le gestionnaire du réseau réalise des économies en réduisant, de manière favorable à la consommation électrique des clients, la consommation électrique des clients, et, à moyen terme, les clients peuvent obtenir une diminution des coûts de l'énergie électrique ou une rémunération pécuniaire lorsque les programmes appliqués impliquent leur collaboration. Dans la littérature, plusieurs catégorisations de programmes d'effacement de la consommation électrique peuvent être trouvées, dépendamment de leur finalité pour les clients y prenant part (récompenses ou

incidence des prix) [10, 11], de leur finalité pour le gestionnaire de réseau (économies ou fiabilité du réseau), du type de marché auquel il s'adresse (détail ou de gros) [12], des mécanismes de contrôle [11] et d'activation [13], etc. L'une des classifications est celle des auteurs [12] est résumée dans la figure 2.1.1 ci-dessous. Dans cette étude, les programmes d'effacement de la consommation électrique sont regroupés selon les types de marché auxquels ils s'adressent. L'objectif des auteurs était de fournir une classification qui faciliterait le choix des consommateurs et/ou des opérateurs de réseaux en fonction de leurs objectifs. Par conséquent, elle se veut pratique.

Ce projet de recherche ne développera guère toutes les définitions des différents programmes d'effacement de la consommation électrique existants. Cependant, à travers la section suivante, un accent particulier sera porté sur les programmes retrouvés dans notre cas d'étude : la province du Québec.



2.1 ***Classification résumée des programmes d'effacement de la consommation électrique. [12]***

2.1.2 État des lieux sur le programme d'effacement de la consommation électrique au Québec

L'effacement de la consommation électrique dans la province du Québec peut impliquer à la fois les secteurs résidentiel, commercial, agricole et industriel. Il existe plusieurs programmes d'effacement de la consommation électrique au Québec (Hilo, EcoPerformance, Rénoclimat, etc.). Cependant, cette étude se concentrera sur le « demand response option ». Le tableau 2-1 ci-dessous résume les différentes catégories de consommateurs éligibles à participer au programme considéré, selon le site officiel d'Hydro-Québec [14]. À titre de définition, le « demand response option » est un programme à participation volontaire, orienté vers le marché du détail en accord avec la classification réalisée par les auteurs [12] (avec une récompense à la clé pour les participants). Ce programme est mis en place en hiver et les clients éligibles à y prendre part, sont notifiés par courriel (le premier ou troisième mardi du mois de novembre de chaque année). Une fois, les inscriptions terminées, Hydro-Québec détermine les périodes de pics de puissance. Ces périodes possèdent les caractéristiques suivantes selon le site officiel de la compagnie. [14] :

- elles peuvent aller du lundi au vendredi entre 6 h et 9 h, le matin et entre 16 h et 19 h dans la soirée ;
- le programme s'étend du 1er décembre au 31 mars inclus, tout en excluant les jours tels que le 24 ; 25 ; 26 ; 31 décembre, le 1er ; 2 janvier, le vendredi saint et le lundi de Pâques lorsqu'ils coïncident avec la période hivernale ;
- les épisodes d'effacement peuvent être d'un maximum de 2 par jour et doivent être espacés d'au moins 7 heures d'intervalle.

L'objectif de ce programme est de satisfaire de façon efficiente les besoins en énergie électrique pendant les périodes de forte demande. Cela implique l'utilisation du minimum de centres de production et la préservation du système électrique. De plus, le programme a également pour but d'impliquer les consommateurs dans la transition énergétique [14]. De nos jours, le programme Hilo par Hydro-Québec est de plus en plus prisé, car il récompense en argent la réduction de la consommation électrique à travers des défis proposés aux participants.

Les participants ayant rempli les conditions, à l'issue du programme, c'est-à-dire le respect du seuil de réduction minimale de l'appel de puissance se verront obtenir des réductions pécuniaires sur leur facturation post-hivernale. Dans ce cas de figure, la valeur de référence est la valeur moyenne de réduction de la puissance lors des différents évènements (périodes de pics de puissance). Les réductions sur la facture sont effectuées lorsqu'elles sont supérieures ou égales à 10 kW en moyenne sur la base du tableau 2-2 ci-dessous. En plus de cette dernière, d'autres conditions, telles que le profil de consommation comme présenté dans le tableau 2-1, doivent être également considérées afin de bénéficier des réductions sur la facturation :

Comme mentionné plus haut, les programmes d'effacement de la consommation électrique sont variés et présentent des avantages pour le marché de l'électricité. Cette dernière constitue la raison justifiant la perpétuelle évolution de ces programmes avec l'implication des nouvelles technologies. Dans la section suivante, l'une de ces technologies, la technologie Blockchain sera abordée afin de faire ressortir son apport dans l'amélioration des programmes d'effacement de la consommation électrique et les perspectives pour la province du Québec.

2.1 ***Types de contrats éligibles à la participation au programme d'effacement de la consommation électrique au Québec. [15]***

Types de consommateurs ou de contrats	**Caractéristiques**
DP	Consommateurs résidentiels ou fermes ayant un appel de puissance électrique supérieur ou égal à 50 kW, au moins une fois dans l'année.
DM	Logements sociaux ou bâtiments ayant plusieurs appartements avec un compteur pour tous
G	Consommateurs dont l'appel de puissance électrique n'excède pas 65 kW au moins une fois dans l'année ou à ceux n'ayant aucune demande électrique.
G9	Consommateurs utilisant des charges électriques importantes pendant une période limitée (terrain de golf…)
M	Consommateurs considérés comme ayant une demande moyenne et devant s'acquitter mensuellement du coût de leur consommation électrique. Très semblable au contrat DP
LG	Consommateurs dont l'appel de puissance électrique est supérieur à 5000 kW ou plus

2.2 ***Montant unitaire des récompenses par intervalle de réduction de puissance à la date du 10 mars 2025 [16]***

Intervalles de réduction (en kW)	**Crédit applicable (en dollars CAN par kW)**
De 10 à 100	78.825
De plus de 100 à 400	68.315
De plus de 400 à 1200	63.060
De plus de 1200	57.805

2.1 La technologie Blockchain

2.1.1 Définition d'une chaîne de blocs

La **chaîne de blocs** ou **blockchain** peut être définie comme étant une structure de données numériques qui peut contenir un nombre en continuelle expansion de transactions, ordonnées chronologiquement. En d'autres termes, c'est une structure de données, équivalent, à un registre d'historiques qui peut contenir des transactions digitales, des données enregistrées, etc. Les transactions sont agrégées de sorte à former de larges structures nommées blocs, qui sont horodatées et cryptographiquement liées aux précédents blocs de sorte à former une chaîne de données enregistrées qui déterminent l'ordre séquentiel des évènements [17]. Le concept de chaîne de blocs a été introduit par [18], mais a été largement vulgarisée par la cryptomonnaie 'Bitcoin' suite aux travaux de [19]. De façon plus détaillée, il existe au sein de chaque bloc à l'intérieur d'une chaîne de blocs, des informations telles que le haché dudit bloc et le haché du bloc qui le précède. Un haché est une donnée obtenue après utilisation d'un algorithme produisant des hachés sécurisés ou Secure Hash Algorithm (SHA). En d'autres termes, le haché correspond à l'image ou au reflet d'un message donné. À ce jour, l'algorithme SHA 256 est l'algorithme le plus communément rencontré dans les blockchains.

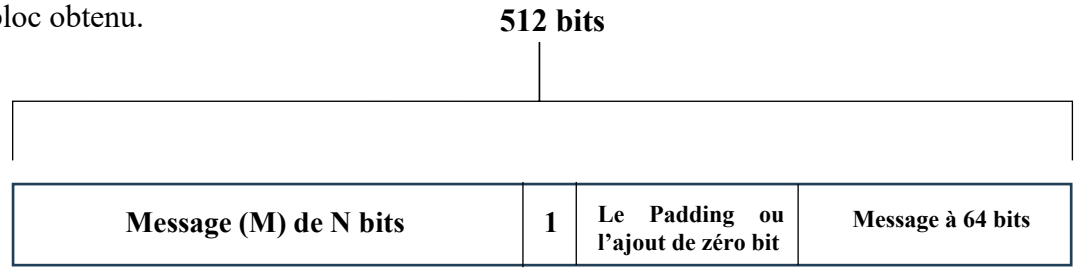
2.1.2 SHA 256

L'algorithme SHA 256 est une construction de Merkle-Damgård [20, 21] de la famille de SHA 2, qui prend en entrée un message de taille arbitraire qu'il transforme en hachés de 256 bits grâce à une fonction dite de compression. Cet algorithme suit les étapes suivantes :

Soit M un message de taille quelconque. La première étape, appelée prétraitement, consistera dans un premier temps, à convertir ce message suivant la table ASCII [22] afin d'obtenir un message de N bits de long avec $N \leq 446\ bits$. Puis deuxièmement, le message obtenu est étendu, de manière à satisfaire l'équation (1) ci-dessous [23] :

$$M + 1 + k = 448\ mod\ 512 \qquad (2\text{-}1)$$

Avec k qui indique le nombre de bits (zéro) à ajouter. M le message, est aussi converti en une représentation de 64 bits (soit en étant tronqué ou étendu par l'ajout de zéro bit) et ajouter au bloc. Le résultat obtenu à l'issue de cette opération est un bloc de 512 bits, mettant ainsi fin à la phase de prétraitement. La figure 2-2 ci-dessous montre les différentes parties du bloc obtenu.



2.2 *Structure du bloc obtenu après le prétraitement [23, 24]*

À la suite de la phase de prétraitement, le bloc obtenu sera divisé en 16 sous blocs de 32 bits qui passeront dans une fonction de compression.

Une fonction de compression est une fonction dont l'objectif premier est de permettre l'encryption d'un message de sorte à exclure le plus possible, la possibilité d'attaques informatiques telles que les attaques par collusion [25] ou par préimages [26]. Dans le cas de SHA 256, la fonction de compression inclut huit variables d'état (A, B, C, D, E, F, G, H) dont les valeurs varient tout au long du processus, car la génération du résultat final se fait à

l'issue de 64 itérations. Ces 64 itérations correspondent aux 64 « mots » obtenus après une expansion du bloc de prétraitement à partir de formule ci-dessous [24]:

$$\begin{cases} W_m \\ W_n = W_{n-16} + \sigma_0 + W_{n-7} + \sigma_1 \end{cases} \quad (2\text{-}2)$$

Avec $0 \leq m \leq 15$ et $16 \leq n \leq 63$. Les expressions de σ_0 et σ_1 sont les suivantes [24]:

$$\sigma_0 = RR^{r1}(W_{n-15}) \oplus RR^{r2}(W_{n-15}) \oplus SR^{r3}(W_{n-15}) \quad (2\text{-}3)$$

$$\sigma_1 = RR^{q1}(W_{n-2}) \oplus RR^{q2}(W_{n-2}) \oplus SR^{q3}(W_{n-2}) \quad (2\text{-}4)$$

Avec $RR^n(X)$, le décalage logique du « mot » X vers la droite de n bits et $SR^n(X)$, le décalage arithmétique du « mot » X de n bits. De plus, on a $r_1 = 7$, $r_2 = 18$ et $r_3 = 3$ et les exposants q_1, q_2 et q_3 ont respectivement les valeurs 7 ; 19 et 10.

La fonction de compression inclut également six fonctions logiques dont les expressions se présentent comme suit :

$$Ch\,(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G) \quad (2\text{-}5)$$

$$Maj\,(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C) \quad (2\text{-}6)$$

$$\Sigma_0(V) = RR^{r1}(V) \oplus RR^{r2}(V) \oplus RR^{r3}(V) \quad (2\text{-}7)$$

$$\Sigma_1(V) = RR^{t1}(V) \oplus RR^{t2}(V) \oplus RR^{t3}(V) \quad (2\text{-}8)$$

Avec $\oplus$ qui correspond au ET logique et les valeurs $r1 = 28$; $r2 = 13$; $r3 = 22$; $t1 = 6$; $t2 = 11$ et ; $t3 = 25$.

À l'issue des 64 itérations impliquant l'ensemble des équations présentées plus haut, l'obtention du haché final se fait par concaténation de hachés intermédiaires de 32 bits comme présentées dans [23] et [24].

Grâce à la procédure décrite plus haut, SHA 256, produit des hachés, qui sont utilisés dans chacune des parties composant les blocs d'une chaîne de blocs. En effet, un bloc peut être subdivisé en deux parties : le haut ou tête et le corps du bloc. Au sein du haut de bloc, des informations telles que l'horodatage, le nonce, etc. peuvent être trouvées en plus des hachées. Le tableau 2-2 ci-dessous présente les informations contenues dans les hauts de blocs rencontrés sur la plateforme de la cryptomonnaie Bitcoin. Le corps du bloc quant à lui contient les informations sur la transaction effectuée. Les deux figures ci-dessous sont des illustrations d'un bloc et d'une chaîne de blocs.

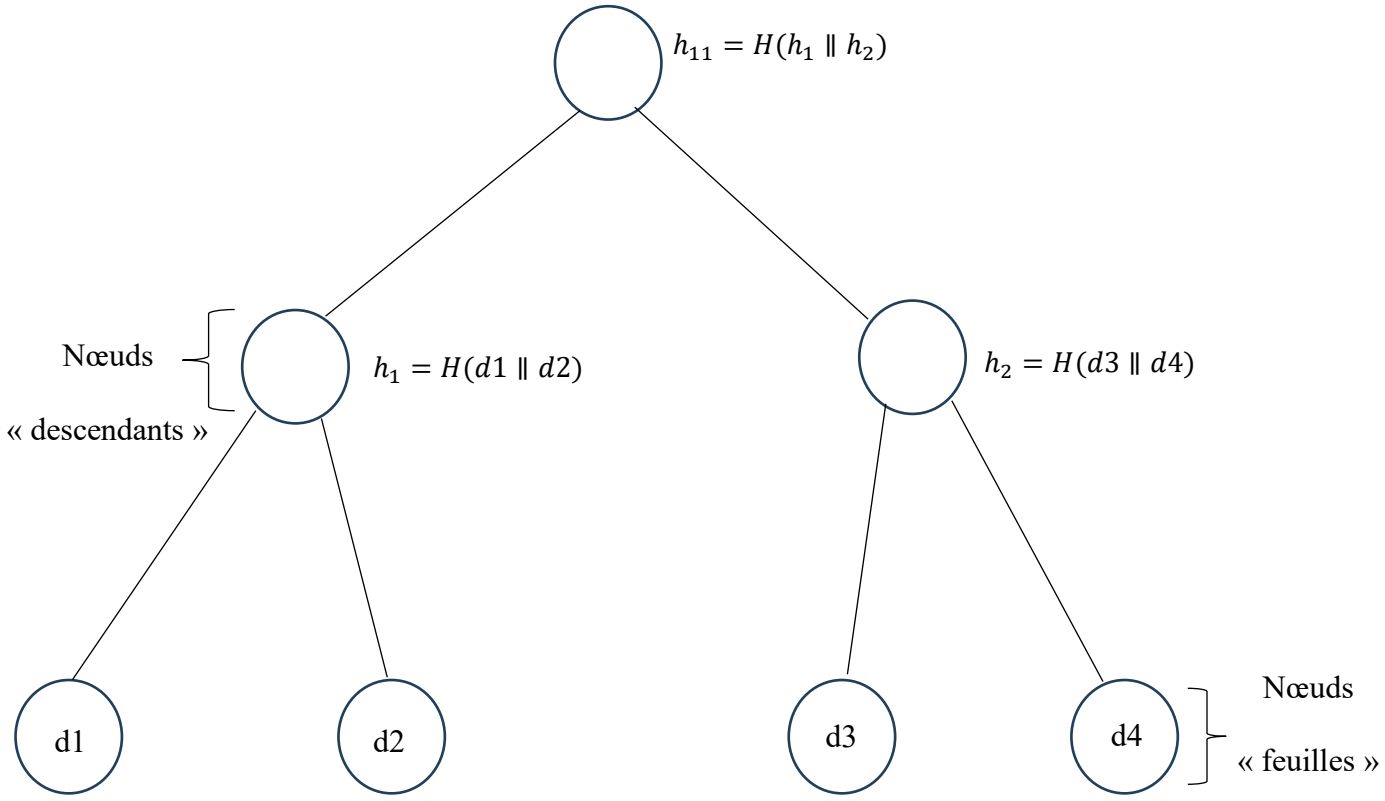
2.3 ***Illustration d'un bloc. [27]***

2.4 ***Illustration d'une chaîne de blocs. [27]***

2.3 ***Éléments contenus dans un haut de bloc sur la plateforme Bitcoin avec une brève description. [28]***

Champ	Taille	Description
Version	32 bits	Version du bloc basé sur la version informatique de la plateforme Bitcoin
hashPrevBlock	256 bits	Le haché du précédent bloc accepté par le réseau Bitcoin
hashMerkleRoot	256 bits	Les transactions sur la Bitcoin sont indirectement hachées à travers l'arbre de Merkle
Horodatage (Timestamp)	32 bits	L'actuel horodatage en seconde depuis le 1er janvier 1970 T00 :00 UTC
Cible (Target)	32 bits	La cible actuelle est une un format compact de 32 bits
Nonce	32 bits	Va de 0×00000000 à 0×FFFFFFFF et est incrémenté après l'essai d'un nouveau haché
Padding + **Longueur**	384 bits	Padding ou complétion standard de SHA 256 lié aux données

L'algorithme SHA 256 est également présent dans l'organisation des données au sein du corps du bloc. Dans le cas de la plateforme Etherum notamment, les informations associées à une transaction sont présentées et sécurisées à l'aide d'une structuration de données nommée arbre de Merkle [29, 30]. L'arbre de Merkel peut se définir comme un arbre binaire dans lequel tous les nœuds « feuilles » contiennent le haché d'informations du bloc et tous les nœuds « non-feuilles » contiennent le haché cryptographique de la concaténation des nœuds descendants ou « enfants » [29, 30]. La figure 2-5 ci-dessous, montre la structure de l'arbre de Merkle,



2.5 ***Structure de l'arbre de Merkle [31]***

2.1.3 Les types de chaînes de blocs et les mécanismes de consensus

Du point de vue de la catégorisation des chaînes des blocs, trois ou quatre types de blockchain peuvent être présentés : Les blockchains publiques, les blockchains privées, les blockchains de consortium [17, 32, 33] et celles hybrides [34, 35] . Les blockchains publiques sont accessibles par tout utilisateur d'internet [17], qui peut par la même occasion participer aux évènements sur la chaîne de blocs notamment les processus de consensus qui seront développés plus loin dans ce document. L'autre caractéristique de la blockchain publique est la difficulté de modifier les informations déjà admises sur celle-ci [34]. La blockchain publique représente le type le plus décentralisé de blockchain. Des exemples de blockchains publiques sont : LiteCoin [36], Bitcoin [19], Etherum [37], etc.

À l'opposé de la blockchain publique, une blockchain privée est administrée par une entité ou une organisation qui fournit des autorisations aux autres utilisateurs. Celle-ci vérifie la validité des évènements et des transactions effectuées sur la chaîne de blocs tout en recherchant le consensus entre les participants. Ce type de blockchain est souvent nommée blockchain à permission, car contrairement aux blockchains publiques sans autorisation, les participants se connaissent mutuellement [38]. Du fait de la présence d'une entité administrative, la blockchain privée est moins décentralisée que celle publique. Des exemples de blockchain privées sont : Hyperledger Fabric [39] et Corda [40].

Les blockchains de consortium, sont des blockchains administrées par plusieurs organisations, ce qui les rendent plus décentralisées que les blockchains privées, mais moins que celles publiques. Des auteurs [33] situent les blockchains de consortium entre celles publiques et celles privées. Des exemples de blockchain de consortium sont : Tendermint [41] et Quorum [42].

Les blockchains hybrides dans certaines classifications plus récentes [34] sont des blockchains très semblables aux blockchains de consortium à la différence qu'elles permettent l'intégration des concepts public et privé vue précédemment. En effet, les blockchains hybrides intègrent la possibilité de réaliser une blockchain privée; avec plusieurs autres secondaires qui sont publiques. Ce type de blockchain correspond aux applications nécessitant des niveaux d'accessibilité de l'information avec simultanément, des données publiques accessibles à tous les utilisateurs et d'autres, privées réservées à un nombre restreint. Des exemples de blockchains hybrides sont : le réseau Ripple [43] et Dragonchain [44].

En plus de tout ce qui précède, un réseau constitué de parties prenantes participe au fonctionnement d'une blockchain. Ce réseau de type réseau pair-à-pair (un réseau d'ordinateurs connecté entre eux) subdivise les parties prenantes de la blockchain en nœuds validateurs et utilisateurs. Les nœuds utilisateurs ont la capacité de pouvoir initier des transactions et/ou des actions, en recevoir et détiennent une copie de la blockchain. Les nœuds validateurs quant à eux sont responsables de toutes modifications sur la blockchain. Afin de pouvoir réaliser des modifications, valider des transactions et créer de nouveaux blocs, il est nécessaire aux nœuds validateurs de parvenir à un consensus entre l'ensemble des nœuds [17]. Cela se fait via un processus appelé mécanisme de consensus. Il en existe plusieurs dépendamment de l'objectif. La preuve du travail ou Proof of Work (PoW) notamment utilisées dans le cadre de la cryptomonnaie Bitcoin et la preuve d'enjeu ou Proof of Stake (PoS) en sont quelques exemples. Les travaux des auteurs [17] et [45] développent avec plus de détails, les mécanismes de consensus existants à ce jour.

L'usage de la blockchain ainsi vulgarisée, l'auteur [46] a, par la suite, publié en 2014, des travaux sur des usages de la blockchain qui impliqueraient des accords entre des parties : Ce sont les contrats intelligents. **Les contrats intelligents ou smarts contracts** peuvent être définies comme étant des lignes de codes écrites et basées sur des accords d'échanges établis entre des parties avec pour objectif de créer un contrat décentralisé, qui s'auto exécute [47]. Les contrats intelligents sont obligatoirement soumis à approbation par l'ensemble des nœuds d'un réseau et apportent plus de transparence dans les échanges entre parties.

À titre de résumé, la combinaison de la technologie blockchain et des contrats intelligents a permis d'améliorer la décentralisation [1] et la transparence [48] des transactions financières. En effet, il est désormais possible de se passer d'intermédiaire pour la gestion de données et l'exécution des termes d'un contrat, tout en conservant un haut niveau de protection de données et de la confidentialité. Par conséquent, l'usage de la technologie blockchain avec contrats intelligents sur des programmes d'effacement de la consommation électrique a été employé afin d'en retirer les avantages. Dans la section suivante seront présentés les travaux combinant à des programmes d'effacement de la consommation électrique, la technologie blockchain.

2.1.4 Cas d'utilisation de la technologie blockchain sur des programmes d'effacement de la consommation électrique dans la littérature

Plusieurs études utilisant la technologie blockchain dans les programmes d'effacement de la consommation électrique sont présents dans la littérature. Dans le cadre de ce travail, les moteurs de recherche Google Scholar et IEEE Explore ont été utilisés avec les mots clés : **Blockchain Demand Response Program**. Les résultats obtenus à la date du 08 septembre

2024 donnent 116 articles sur IEEE Explore et plusieurs autres sur Google Scholar. La lecture des résumés des articles a servi de méthode de sélection des articles les plus pertinents.

Le tableau 2-4 ci-dessous présente une liste de travaux de quelques auteurs, ainsi que leurs objectifs et les choix effectués.

2.4 ***Liste de quelques travaux utilisant la technologie blockchain dans un programme d'effacement de la consommation électrique (Source : S. Franck A. Coulibaly)***

Articles	**Objectifs principaux**	**Taxinomies de la blockchain (Plateformes)**	**Mécanismes de consensus**
C. Pop et *al* [1]	Apporter une solution décentralisée permettant la gestion des programmes d'effacement de la consommation électrique dans un contexte de réseaux électriques intelligents.	N/A (Etherum)	Proof of Work (PoW)
Y. Yan et *al*. [7]	Création d'un mécanisme centralisé de gestion des offres de réduction de la demande des consommateurs dans un programme d'effacement de la consommation électrique en dan une région chinoise.	Blockchain de Consortium	N/A

A. Aoun et *al* [49]	Régulation de la consommation électrique des consommateurs à l'aide d'un système de plafonnement de la demande, assorti de récompenses et de pénalités.	N/A	N/A
B. Li et *al* [47]	Mise en place d'un programme d'effacement de la consommation électrique avec récompense, basé sur l'autoreportage par les participants de leur consommation	N/A (Etherum)	Proof of Work (PoW)
M. Di Sylvestre et *al* [38]	Implémentation d'un programme d'effacement de la consommation électrique avec rémunération et agrégation de charges électriques faisant suite de l'étude [50]	Blockchain de Consortium (Hyperledger Fabric)	N/A

C. Pop et *al* [51]	Améliorer le niveau de confidentialité des participants d'un programme d'effacement de la consommation électrique	Blockchain Publique (Etherum)	Proof of Authority (PoA)
V. Deshpande et *al* [52]	Proposition d'un programme d'effacement de la consommation électrique efficient et optimal fonctionnant avec la technologie blockchain.	Blockchain de Consortium	N/A
O. Kawnhen et *al* [53]	Étude dont la contribution est l'optimisation d'un programme d'effacement par la combinaison de réseaux de neurones récurrents pour la prédiction des prix de l'énergie à court terme avec une blockchain à permission utilisant le sharding pour la protection et la confidentialité des données.	Blockchain de Consortuim (Etherum)	Tolérance aux pannes byzantines pratiques (pBFT)

Dans le tableau 2.2, un accent particulier a été accordé aux catégories de blockchain, aux mécanismes de consensus et aux plateformes, car ils occupent une part importante dans le fonctionnement d'une blockchain. L'objectif de cette sélection d'articles est d'identifier en fonction des objectifs pour cette étude, les catégories, les plateformes et mécanismes les plus pertinents et les plus utilisés dans ce type d'étude. Dans le chapitre suivant, les résultats présentés seront exploités afin de réaliser un choix éclairé des caractéristiques de la blockchain ainsi que son implémentation.

2.2 Conclusion

La revue de littérature a permis d'appréhender les définitions des mots clés de ce projet de recherche que sont la technologie blockchain et l'effacement de la consommation électrique, tout en passant en revue, quelques travaux similaires. Dans la suite du travail, la proposition d'une structure de blockchain adaptée à notre contexte d'étude sera présentée, en plus de son implémentation.

Chapitre 3 - Implémentation du programme d'effacement de la consommation électrique dans la province du Québec avec la technologie blockchain

Dans ce chapitre, sera présenté et implémenté le programme d'effacement de la consommation électrique présenté avec la technologie blockchain.

3.1.1 Choix des caractéristiques du modèle à implémenter

Comme préalable à l'implémentation du programme d'effacement de la consommation électrique avec la technologie Blockchain, il est nécessaire de préciser les caractéristiques du modèle à utiliser. Par caractéristique, il s'agit de la catégorie ou type de blockchain, du mécanisme de consensus et de la plateforme.

3.1.2 Le type de blockchain

Le choix du type de blockchain peut être réalisé en évaluant la nécessité de partager des informations à des personnes anonymes [52], le cas échéant, une chaîne de blocs publique et sans permission serait appropriée. Dans le cas contraire, une chaîne de blocs avec permission (privée ou de consortium) serait plus indiquée. Le programme à implémenter doit pouvoir permettre l'intégration dans le réseau pair-à-pair de nouveaux participants lors de la phase d'adhésion. Cependant, leur anonymat n'est pas requis. De plus, afin d'offrir davantage de confidentialité et de sécurité, la blockchain publique a été écartée. Par conséquent, une chaîne

de blocs avec permission en occurrence, la chaîne de blocs privée a été retenue dans le cadre de ce travail.

3.1.3 Mécanisme de consensus

Le choix du mécanisme de consensus ou système de validation se révèle important, car il permet de préserver l'intégrité du système. Dans le cadre de cette étude, la méthodologie retenue pour la sélection du système de validation est la réduction du champ des possibilités, aux mécanismes les plus retrouvés dans la littérature. Par conséquent, les éléments listés dans le tableau 2.2 du chapitre précédent ont été considérés. Il s'agit :de la preuve de travail (PoW), la preuve d'enjeu (PoS) et la preuve d'autorité (PoA). Les avantages et les limites de ces mécanismes sont listés dans le tableau 3.1 ci-dessous. La tolérance aux pannes byzantines n'a pas été considérée, car très peu présente.

3.1 ***Avantages et limites des mécanismes de consensus les plus rencontrés dans la littérature pour le type d'application désirée (Source : S. Franck A. Coulibaly)***

Mécanismes de consensus	**Brève description**	**Avantages**	**Limites**
Preuve de travail (PoW) [54]	Elle consiste à mettre les participants en compétition afin de valider les transactions. Ces participants appelés mineurs utilisent une grande puissance de calcul pour résoudre des puzzles cryptographiques	• Rémunération des mineurs qui réussissent à valider des blocs. • Décentralisation effective la facilité de devenir mineur	• Très énergivore pour les nœuds validateurs • La grande consommation électrique engendrée par les nœuds validateurs a pour conséquence une grande pression sur le réseau électrique [55]
Preuve d'enjeu (PoS) [56]	Elle consiste pour les participants à mettre en compétition leurs intérêts (généralement des cryptomonnaies ou tokens) sur la blockchain afin d'être sélectionné comme validateurs. L'idée étant de prévenir une gestion malicieuse en responsabilisant les participants qui ont le plus à perdre.	• Moins énergivore que la preuve de travail • Validation plus rapide des transactions par rapport au PoW [56].	• Consensus favorisant les plus nantis [57]. • Moins sécurisé que le PoW avec l'introduction de nouveaux risques
Preuve d'autorité (PoA) [58]	Elle peut être vue comme un PoS modifié à la différence qu'à la place des intérêts, les participants utilisent leur propre identité. En d'autres termes, c'est un consensus où le vote est utilisé par les participants afin de choisir les validateurs sur la base de leur identité. [17]	• Possibilité de prédire le temps de validation et de création des blocs [59] • Elle a besoin de moins d'énergie et de puissance de traitement par rapport à d'autres mécanismes tels que le PoW .[58]	• Partiellement décentralisé, car basé sur des personnes dites de confiance • Les nœuds administrateurs ont leur identité connue par les autres rendant possible, la manipulation par une tierce personne ou organisation [59]

En plus des avantages et des limites, des auteurs comme [58], définissent cinq caractéristiques d'un mécanisme de consensus idéal dont la mission première est de rendre la chaîne de blocs plus sûrs et augmenter le nombre de transactions traitées par seconde. Ces caractéristiques sont les suivantes :

- l'extensibilité : il s'agit de la capacité du mécanisme à continuer la fourniture d'un service optimal malgré une croissance du nombre de transactions. En comparaison, le PoW est plus extensible que le PoS et le PoA ;
- l'intégrité : il s'agit de veiller à ce que les actions des nœuds malicieux n'aient aucune incidence sur la chaîne de blocs ;
- la résilience : Tous les nœuds légitimes doivent pouvoir décider de toute action sur la chaîne de blocs [60] ;
- la Coopération : le consensus d'un maximum possible de nœuds doit être atteint ;
- l'égalité : un mécanisme de consensus doit pouvoir permettre l'égalité des votes entre les nœuds.

À la vue de tout ce qui précède, il sera utilisé dans le cadre de cette étude, la preuve d'autorité. Trois arguments peuvent expliquer ce choix. D'abord, la preuve d'autorité permet d'assurer une validation plus rapide des transactions par rapport au PoW. Cette dernière reste plus sécurisée, mais le choix d'une blockchain privée permet d'améliorer le niveau de sécurité du processus de validation avec la PoA. Ensuite, le choix d'une PoA au détriment d'une PoS s'explique par le désir d'assurer plus d'égalité entre les acteurs en évitant le contrôle exclusif du système par les plus nantis. Enfin, d'après les auteurs [47], la PoA ne présente aucun inconvénient pour ce type d'application.

3.1.4 La plateforme

Le choix de la plateforme de blockchain dépend des tâches qu'elle doit effectuer, car tout comme les algorithmes de consensus, chaque plateforme possède des forces et des limites. Afin de réaliser le choix de la plateforme dans cette étude, le champ des possibilités a été réduit aux plateformes les plus retrouvées dans la littérature. Les études présentées dans le tableau 2.2 du Chapitre précédent ont servi de base d'analyse. Les plateformes sont : Etherum et l'Hyperledger Fabric. De plus, ces dernières donnent la possibilité d'implémenter le type de blockchain et le mécanisme de consensus retenus plus haut. Plusieurs arguments soutenant l'emploi de chacune de ses plateformes peuvent être trouvés dans la littérature. À titre d'exemple, les auteurs [38] utilisent l'Hyperledger Fabric dans leur étude et justifie son emploi par la possibilité d'écrire des contrats intelligents dans des langages génériques tels que Java, Go et Node.Js. De plus, Fabric permet un choix varié de mécanismes de consensus.

La plateforme Etherum quant à elle, est plus connue que la précédente. Elle se présente comme l'une des plateformes les plus populaires, notamment grâce aux possibilités qu'elle offre de permettre la rédaction de contrats intelligents en plus de la possibilité d'utilisation de sa cryptomonnaie appelée Ether pour la rémunération des parties prenantes. Cependant, Etherum utilise le PoS comme algorithme de consensus par défaut. Afin de pouvoir utiliser le PoA, l'une des possibilités est la création d'une blockchain locale ou privée avec un mécanisme du choix.

Dans le cadre de cette étude, la plateforme Etherum a été choisie. La popularité du langage de programmation Solidity dans la rédaction du contrat intelligent a justifié ce choix.

Les caractéristiques de la blockchain à utiliser étant présentées, le matériel et les méthodes qui devront mener à son implémentation de la blockchain feront l'objet de la section suivante.

3.2 Matériel et méthodes

3.2.1 Matériel

3.2.1.1 Hardhat

Hardhart [61] est un environnement de développement, utilisant une Machine Virtuelle Etherum (EVM) avec pour objectif d'apporter des outils permettant de simplifier le développement, la compilation, le test et le déploiement des contrats intelligents. Dans le cadre de ce travail, la version 2.22.19 de Hardhat sera utilisée.

Afin de parfaitement accomplir toutes ses fonctions, Hardhat a besoin d'un compilateur de contrat intelligent et de Node.js [62] pour le fonctionnement de la console Hardhat permettant l'exécution des commandes en Javascript. Dans cette étude, la version du compilateur Solidity utilisé sera la version 0.8.28 et celle de Node.js sera la version 18.20.6.

3.2.1.2 Go-Etherum (Geth)

Go Etherum [63] est un outil aussi appelé logiciel client, facilitant l'interaction entre le réseau Etherum et ses utilisateurs. Il est l'implémentation en langage Golang du protocole Etherum. Dans le cadre de cette étude, il doit permettre la création d'une chaîne de blocs privée. De plus, Go Etherum permet, à travers le mécanisme Clique d'utiliser la preuve d'autorité, consensus qui a été retenu. Afin de mener à bien les travaux de cette étude, la version 1.13.15 de Go Etherum sera utilisée.

3.2.1.3 Microsoft Visual Studio Code

C'est un Environnement de Développement Intégré (IDE) qui permet le développement de programmes informatiques notamment avec la possibilité qu'il offre d'intégrer des extensions telles que le langage de programmation Solidity. Dans le cadre de cette étude, il a été utilisé avec l'extension du langage Solidity pour rédiger le contrat intelligent.

3.2.2 Méthodes

Dans cette section sera développée la méthode d'implémentation. Elle sera subdivisée en trois parties : d'abord, l'architecture du système sera présentée ; ensuite, les objectifs du contrat intelligent seront expliqués ; et enfin, la structure du contrat intelligent sera décrite.

3.2.2.1 Présentation de l'architecture du système

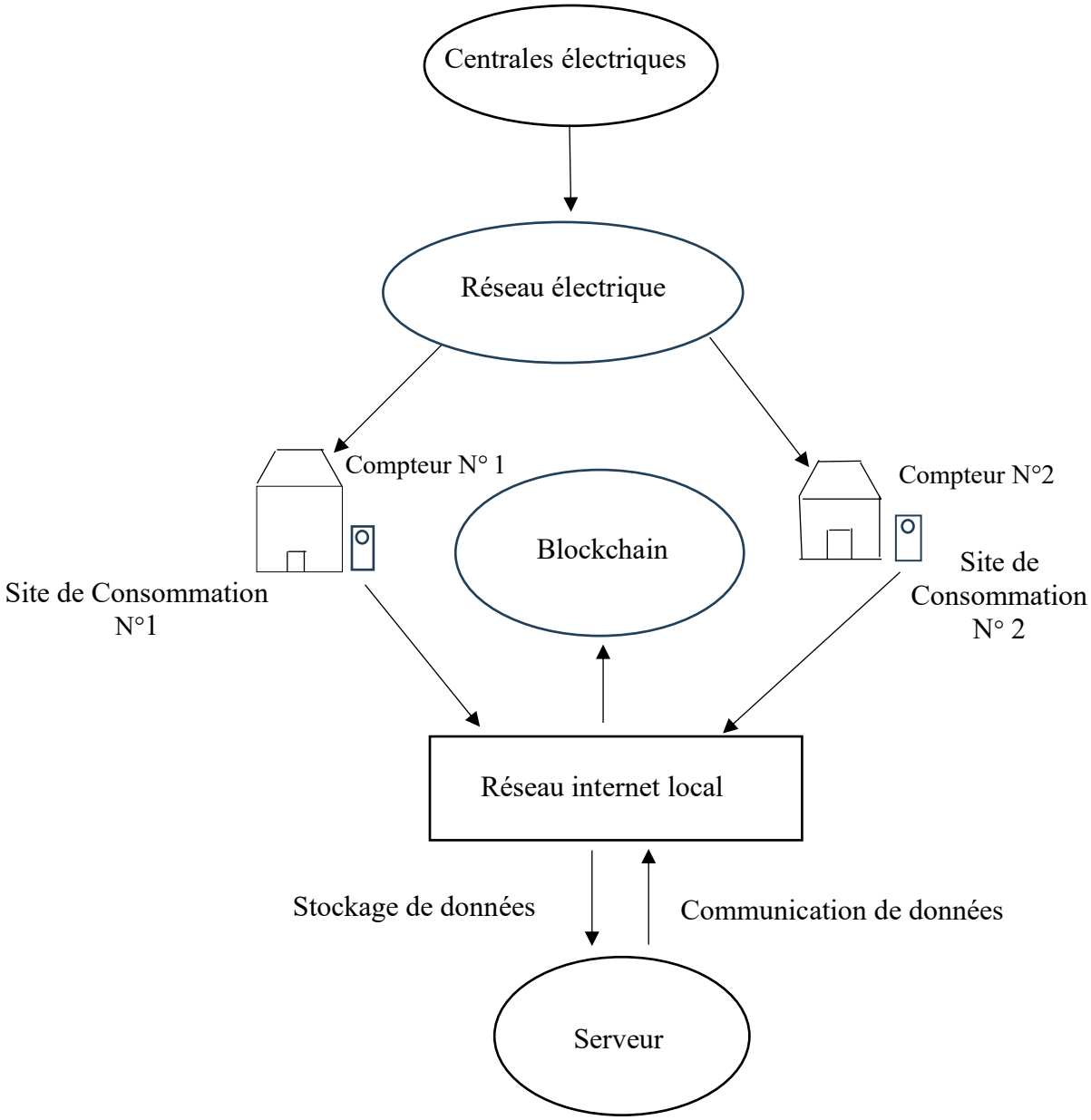
L'architecture du système peut être subdivisée en deux grandes parties :

- l'architecture physique ;
- l'architecture virtuelle comprend le contrat intelligent qui exécute le programme retenu, ainsi que la blockchain.

3.2.2.1.1 Architecture physique

L'architecture physique comprend les infrastructures électriques telles que les appareils de mesure, de stockage et de communication de données (aussi appelés oracles), le réseau électrique et les centrales électriques, etc. La configuration des différents éléments dans l'architecture physique relève d'une grande importance, car l'objectif de cette architecture est de permettre l'acheminement des données électriques des compteurs au contrat intelligent. Pour se faire, les données sont d'abord stockées dans un serveur informatique

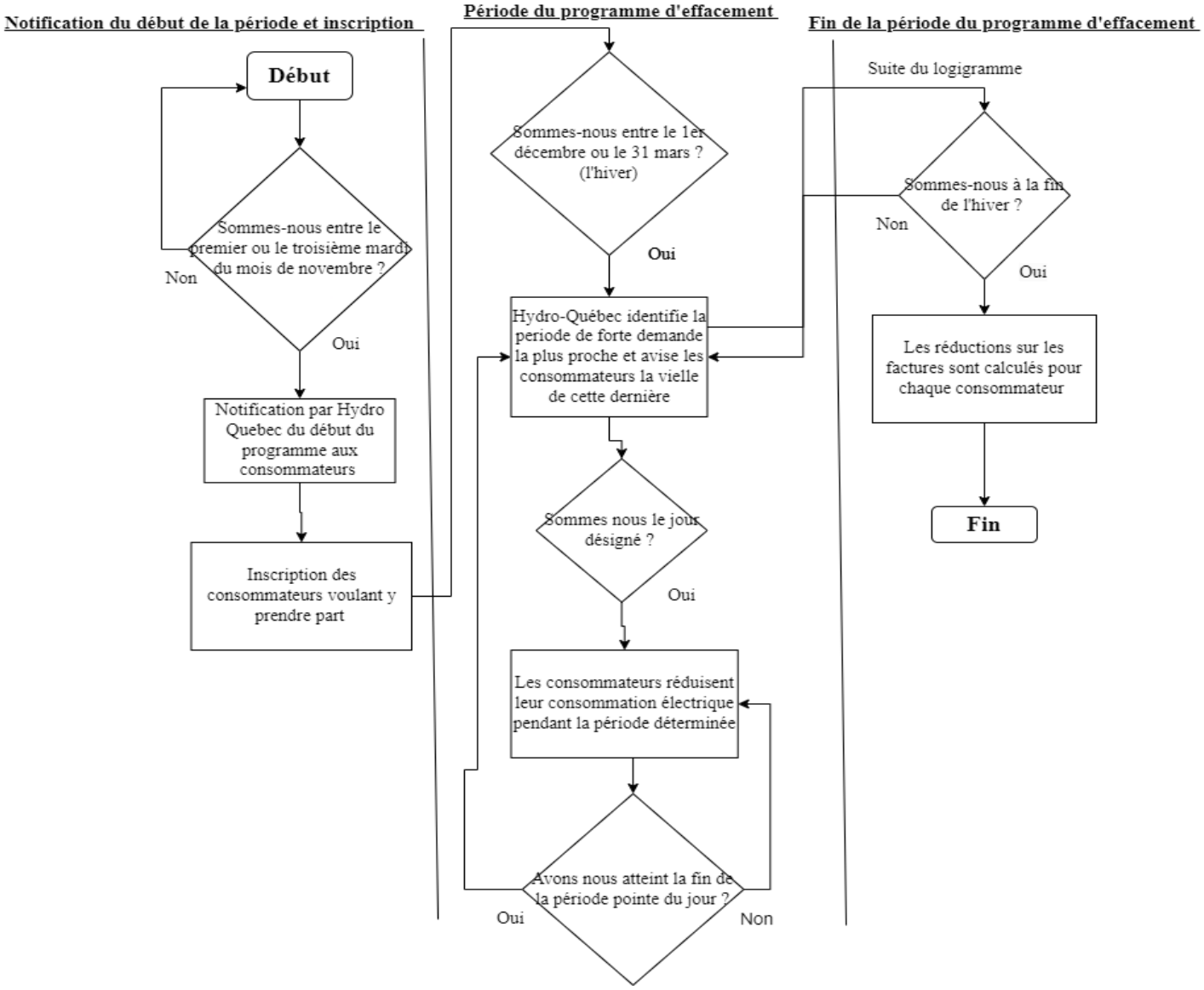
avant d'être communiquées. La configuration des infrastructures choisie est présentée dans la figure 3-1 ci-dessous. Il est à noter que les interactions entre les consommateurs et/ou participants ne sont pas représentées dans cette figure.



3.1 ***Schéma de l'architecture physique (Source : S. Franck A. Coulibaly)***

3.2.2.1.2 Architecture virtuelle (Contrat intelligent et blockchain)

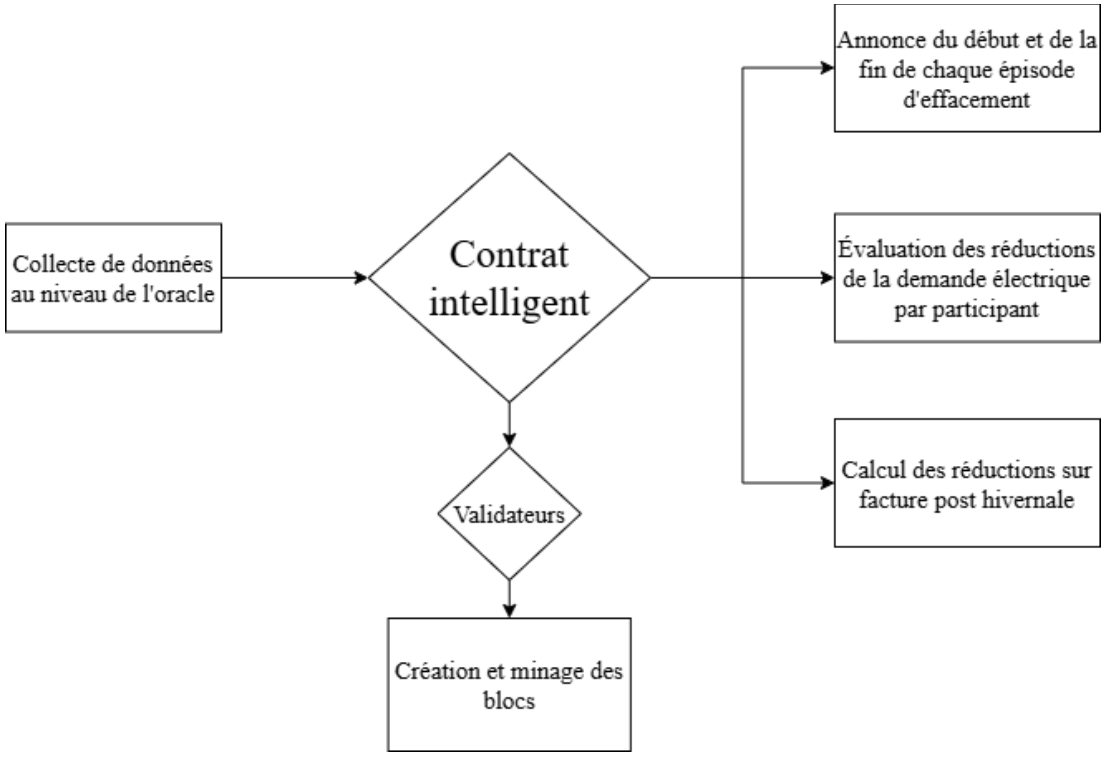
Le programme d'effacement de la consommation électrique à implémenter suivra les grandes étapes de celui actuellement utilisé par la société Hydro-Québec. L'objectif de ce choix est de permettre à terme une comparaison objective. Les informations de ce programme sont présentées dans cette référence [14] et résumées dans la section 2.1 du chapitre précédent. Le logigramme à la figure 3.1.1 en résume également les grandes étapes.



3.2 ***Logigramme montrant le programme d'effacement de la consommation électrique utilisé par Hydro-Québec (Source : S. Franck A. Coulibaly)***

À l'image de ce qui est présenté dans la figure 3.3, la contribution apportée au programme d'effacement de la consommation actuellement en vigueur, est l'usage d'un contrat intelligent afin de :

- permettre la double annonce des dates de réduction de la consommation tant par voie électronique (courriel) que sur la chaîne de blocs ;
- permettre la collecte des données au niveau du serveur informatique;
- évaluer les réductions de demande électrique par client participant au programme ;
- calculer les réductions sur les factures post hivernales sur la base de l'évaluation des réductions de puissance réalisées lors du programme.



3.3 ***Illustration de l'action du contrat intelligent proposé dans la chaîne de blocs (Source : S. Franck A. Coulibaly)***

3.2.2.2 Structure du contrat intelligent proposé

Afin de réaliser l'objectif fixé dans la section précédente, le contrat en langage Solidity, présenté en Annexe A, a été subdivisé en plusieurs fonctions, évènements, constructeurs, et modificateurs afin d'en faire une imitation du programme actuellement utilisé par Hydro-Québec tout en incluant la technologie blockchain. Pour se faire, il a été divisé en trois grandes parties ou étapes de fonctionnement :

La première étape est celle marquée par l'inscription de nouveaux participants aux programmes par Hydro-Québec (ici désigné par le nom propriétaire ou « Owner »). L'étape se caractérise par un évènement nommé Participantenregistre, dont la syntaxe présentée à la figure 3-4, informe tous les participants du réseau pair-à-pair du début des nouvelles inscriptions. Une fois l'évènement émis, la fonction du nom ajoutparticipant, présentée à la figure 3-5 réalise l'action d'inscription. Par la suite, à compter du 1er décembre, le propriétaire ne dispose plus du pouvoir d'intégrer de nouveaux participants. Cette action est matérialisée par un modificateur de fonction, du nom avantDécembre, qui modifie la fonction inscription comme présentée dans la figure 3.6.

Une autre variante de l'action de la fonction inscription peut être réalisée en créant un autre contrat intelligent enregistrant les données des participants directement dans l'oracle. Les résultats obtenus avec cette variante peuvent, théoriquement, être semblables à ceux présentés plus haut.

```
44
45      event Participantenregistre(
46          address indexed participant,
47          string profilConso,
48          uint256 timestamp
49      );
50
```

3.4 ***Syntaxe de l'évènement Participantenregistre (Source : S. Franck A. Coulibaly)***

```
function ajoutparticipant(address participant, string memory profilConso) public onlyOwner avantDecembre {
    require(!participants[participant].enregistre, "Deja inscrit");
    require(participant != address(0), "Adresse invalide");

    participants[participant] = Participant({
        enregistre: true,
        profilConso: profilConso,
        totalReduction: 0,
        totalCredits: 0,
        chargeEvents: new uint256 [](0)
});

    participantAddresses.push(participant);
    emit ParticipantEnregistre(participant, profilConso, block.timestamp);
}
```

3.5 ***Syntaxe de la fonction ajoutparticipant montrant l'émission de l'évènement Participantenregistre (Source : S. Franck A. Coulibaly)***

```
// Modificateur pour limiter l'ajout de participants après le 1er décembre
modifier avantDecembre() {
    uint256 month = (block.timestamp / 30 days) % 12 + 1; // Récupère le mois actuel (1 = janvier, 12 = décembre)
    require(month < 12, "Ajout de participants interdit apres le 1er decembre");
    _;
}
```

3.6 ***Syntaxe du modificateur de fonction avantDecembre (Source : S. Franck A. Coulibaly)***

La seconde étape qui s'étend du 1er décembre au 1er avril est marquée par le déroulement du programme d'effacement. Pour se faire, chaque fois que le besoin se présente, Hydro-Québec peut émettre l'évènement appelé DebutEffacementActif, avec la syntaxe présentée dans la figure 3.7, pour informer d'un épisode d'effacement à venir. La fonction debutEffacement, présentée à la figure 3.8, permet de lancer l'épisode d'effacement. Au cours de ce dernier, la puissance électrique de chaque participant est enregistrée et stockée sur le serveur. Lorsque les objectifs de l'effacement sont atteints, Hydro-Québec peut à tout moment mettre fin à l'évènement DebutEffacementActif, en utilisant la fonction finEffacement dont la syntaxe est présentée à la figure 3.8. Un évènement nommé FinEffacement est alors lancé afin d'informer les parties prenantes au programme.

3.7 ***Syntaxe des évènements DebutEffacementActif et FinEffacement (Source : S. Franck A. Coulibaly)***

```
// Déclencher un effacement (par le propriétaire)
function debutEffacement() public onlyOwner {
    require(!effacementActif, "Un effacement est deja en cours");
    effacementActif = true;
    emit DebutEffacementActif(msg.sender, block.timestamp);
}

// Mettre fin à un effacement (par le propriétaire)
function finEffacement() public onlyOwner effacementEstActif {
    effacementActif = false;
    emit FinEffacement(msg.sender, block.timestamp);
}
```

3.8 ***Syntaxe des fonctions debutEffacement et finEffacement pour respectivement lancer ou mettre fin à l'effacement (Source : S. Franck A. Coulibaly)***

La troisième et dernière étape est le calcul des crédits post hivernal lancé le 1er avril, et marquant la fin du programme. Pour se faire, la puissance électrique réduite et effective est calculée. Pour rappel, la puissance électrique réduite et effective est la moyenne des puissances électriques réduites de chaque participant enregistrées durant le programme. Afin de pouvoir la calculer, une fonction du contrat intelligent nommée calculerCredits, présentée à la figure 3.11, peut être utilisée. De plus, la période de calcul des crédits est marquée par l'évènement CreditsCalcules, dont la syntaxe est présentée à la figure 3.10. À la suite de ce premier calcul, la fonction nommée _calculCredits donnant les crédits sur la base de l'analyse de la puissance réduite, moyenne et effective est présentée à la figure 3.9.

```
    function _calculEffectiveReduce(address participant) internal view returns (uint256) {
    }
function _calculCredits(uint256 effectiveReduce) internal pure returns (uint256) {
    uint256 credit = 0; // Initialisation de la valeur credit
    uint256 creditPr = 78825;     // Le montant Maximale de la première tranche (multipliée par 10)
    uint256 creditPr_Cons = creditPr / 10 ; // Le montant Maximale de la première tranche à considérer
    uint256 creditSc = 273260;  // Le montant Maximale de la seconde tranche
    uint256 creditTr= 75672;    // Le montant Maximale de la troisième tranche

    if (effectiveReduce > 10 && effectiveReduce <= 100) {
        credit = (78825 * effectiveReduce) / 1000;

    } else if (effectiveReduce > 100 && effectiveReduce <= 400) {
        credit = ((68315 * (effectiveReduce - 100)) / 1000)  + creditPr_Cons;

    } else if (effectiveReduce > 400 && effectiveReduce <= 1200) {
        credit = ((63060 * (effectiveReduce - 500)) / 1000)  + creditSc + creditPr_Cons;

    } else if (effectiveReduce > 1200) {
        credit = ((57805 * (effectiveReduce - 1700)) / 1000) + creditSc+ creditPr_Cons + creditTr ;

    }

    return credit;
}
```

3.9 ***Syntaxe de la fonction _calculCredits (Source : S. Franck A. Coulibaly)***

```
event CreditsCalcules(
    address indexed participant,     // Adresse du participant pour lequel les crédits sont calculés
    uint256 effectiveReduce,         // Réduction effective calculée
    uint256 totalCredits,            // Total des crédits après le calcul
    uint256 timestamp                // Horodotage du calcul
);
```

3.10 ***Syntaxe de l'évènement CreditsCalcules (Source : S. Franck A. Coulibaly)***

```
function calculerCredits(address participant) internal view returns (uint256) {
    uint256[] storage chargeEvents = participants[participant].chargeEvents;
    uint256 total = 0;
    uint256 count = chargeEvents.length;

    for (uint256 i = 0; i < count; i++) {
        total += chargeEvents[i];
    }

    return count > 0 ? total / count : 0;
}
```

3.11 ***Syntaxe de la fonction calculerCredits (Source : S. Franck A. Coulibaly)***

Ces trois étapes décrites plus haut permettent de mettre en œuvre le programme d'effacement. Toutefois, d'autres commandes telles que le décryptage des données fournies par le serveur, un constructeur permettant la communication avec le serveur et les syntaxes d'évènements importants peuvent être également retrouvées au sein du contrat intelligent.

La présentation de la structure du contrat intelligent ainsi terminée, la section suivante portera sur la présentation des conditions et des étapes de simulation de ce contrat dans une chaîne de blocs privée.

3.3 Simulations

L'objectif principal de cette simulation est d'apprécier la fonctionnalité des contrats intelligents, à défaut de moyens pour pouvoir le réaliser dans des conditions réelles. Les objectifs spécifiques attendus quant à eux, sont dans un premier temps, la création d'une blockchain privée, fonctionnant avec la preuve d'autorité comme mécanisme de consensus et deuxièmement, le pouvoir d'interagir avec les différentes fonctions du contrat d'effacement. À défaut d'avoir un serveur ou oracle à disposition; un deuxième contrat intelligent dont la syntaxe est disponible en Annexe C, a été rédigé afin de feindre le rôle du

serveur informatique. Le rôle de celui-ci sera de stocker et de communiquer des données. La simulation sera subdivisée en deux grandes étapes. Premièrement, il s'agira de concevoir une blockchain privée, sur laquelle il sera possible, dans un second temps, de tester les différentes fonctionnalités du contrat intelligent au sein d'un réseau privé.

La première étape peut être subdivisée en deux points principaux. D'abord, il s'agira d'initialiser le bloc de genèse sur la base du programme en javascript qui peut être trouvé en Annexe B et de la commande (1) en Annexe D. Le programme en javascript possède en son sein plusieurs types de données telles que le numéro d'identification du réseau privé, les adresses des validateurs, le mécanisme de consensus utilisé, etc. Pour terminer, Go Etherum est démarré avec un réseau PoA grâce à la commande (2) en Annexe D, qui possède les mêmes données que celles présentées en Annexe B. Le démarrage du réseau PoA correspond à la création de chaînes de blocs privées et marque la fin de la première étape.

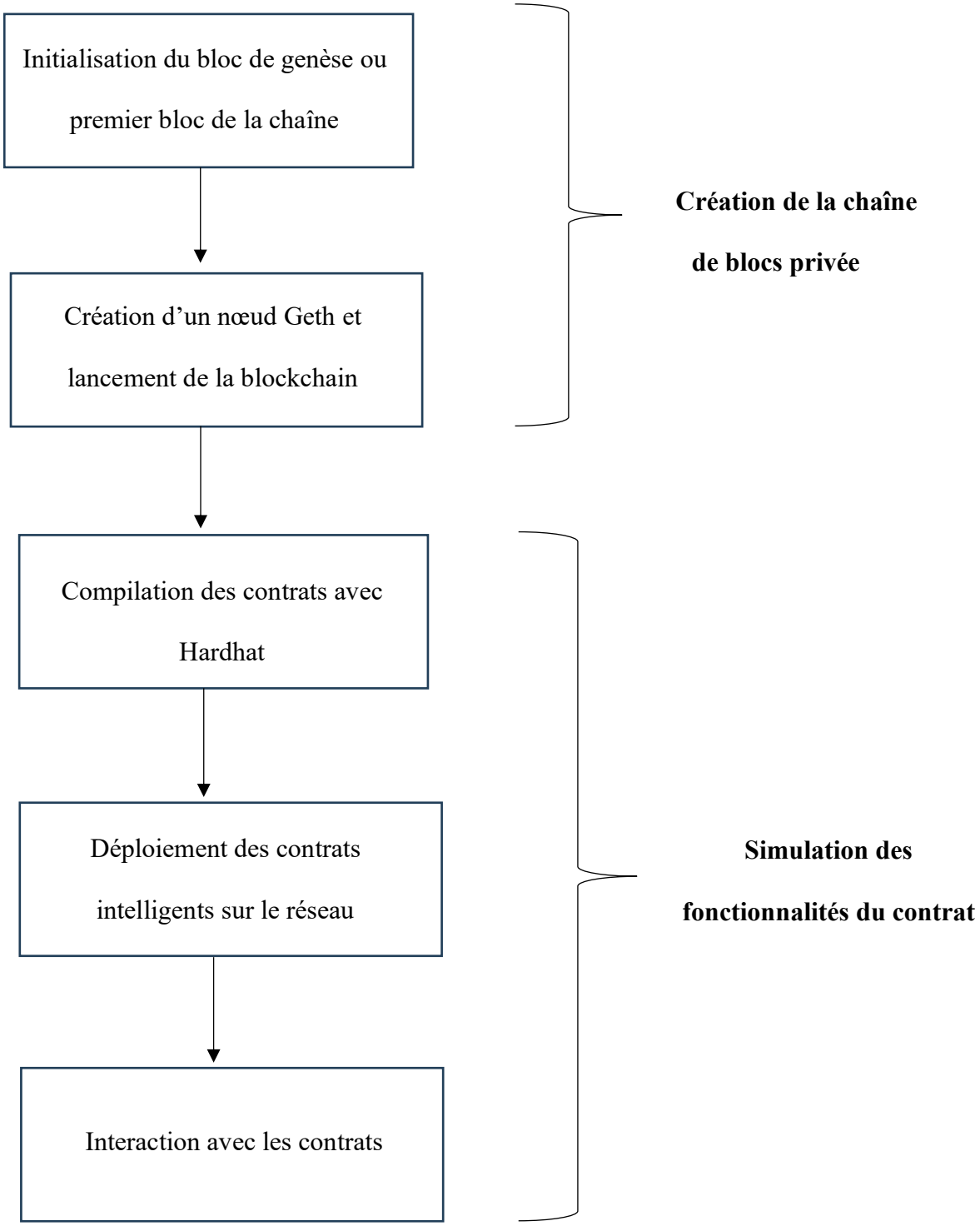
La seconde et dernière étape est celle de la simulation des fonctionnalités du contrat. Cette étape possède trois grands points. D'abord, il s'agira de compiler les contrats intelligents en langage Solidity avec Hardhat. La compilation a pour objectif de convertir les contrats en artefacts, c'est-à-dire en format Bytecode ou Abi, des formats lisibles pour le réseau. La compilation est réalisée grâce à la commande (3) en Annexe D. À la suite de la compilation, les contrats doivent être migrés ou déplacés sur la blockchain afin d'être simulable. La migration du contrat faisant office d'oracle sera d'abord réalisée dans un premier temps et enfin sera réalisée la migration du contrat régissant le programme d'effacement de la consommation électrique. La commande (4) en Annexe D permettra cette action. Enfin, lorsque la compilation et la migration sont parfaitement réalisées, il est possible d'interagir avec les contrats à partir de la commande (5) donnant accès à la console de

Hardhat. L'ensemble des commandes en Javascript permettant d'interagir avec les contrats est donné en Annexe D à la suite de la commande (5). La figure 3.12 ci-dessous résume l'ensemble des étapes et des points abordés et le tableau 3.2 présente les conditions de simulation. Dans cette étude, quatre fonctions seront testées, il s'agit de la fonction permettant l'inscription de nouveaux participants par le propriétaire, la fonction lançant un effacement et celle y mettant fin et la fonction de calcul des crédits. Par conséquent, il en résulte un fichier test rédigé en javascript et présenté en Annexe E. Il est à noter que les modificateurs de fonctions avantdecembre et celle avantavril ont été retirées afin de pouvoir simuler les contrats sans prendre le risque de recevoir des messages d'erreur, car les tests sont réalisés en février 2025. Les contrats intelligents du programme et de l'oracle ont été réécrits afin de permettre des simulations sans erreur. Ces contrats sont respectivement présentés en Annexes G et H. Les principales différences rencontrées avec les contrats précédemment présentés, sont les suivants : d'abord, l'absence des modificateurs de fonctions avantdecembre et avantavril au sein du nouveau contrat du programme, ensuite l'ensemble des données sont stockées au niveau de l'oracle pour le nouveau contrat et enfin l'ajout de participants est réalisé depuis l'oracle. À des fins de simulation, l'ajout d'un participant et les données en lien avec ledit participant se feront manuellement étant donnée l'absence de compteurs d'énergie.

À la suite de la présentation des conditions et des étapes de simulation des contrats intelligents, la section suivante présentera les résultats obtenus et les discutera.

3.2 ***Résumé des conditions de simulation (Source : S. Franck A. Coulibaly)***

Caractéristiques	Conditions retenues pour la simulation
Type de blockchain	Privée
Mécanisme de consensus	Preuve d'autorité (PoA)
Plateforme Blockchain	Etherum (Go Etherum)
Nombre de validateurs	01 (adresse : 0x550c3335681388d27daa7812153ba3ad131c806b)
Nombre de contrats à déployer	02 (OracleMock qui fait objet d'oracle et EffacementP2P pour la gestion du programme d'effacement de la consommation)



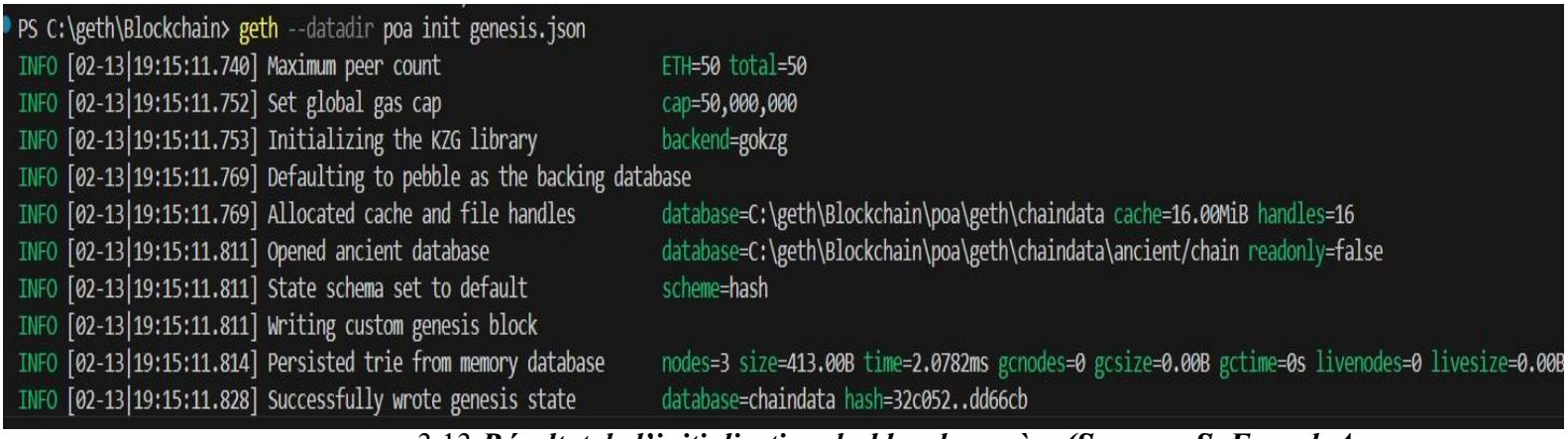
3.12 ***Les grandes étapes de simulation d'un réseau privé (Source : S. Franck A. Coulibaly)***

3.4 Résultats et discussions

Suivant les différentes étapes de simulation présentées dans la section précédente, les résultats se présentent comme suit :

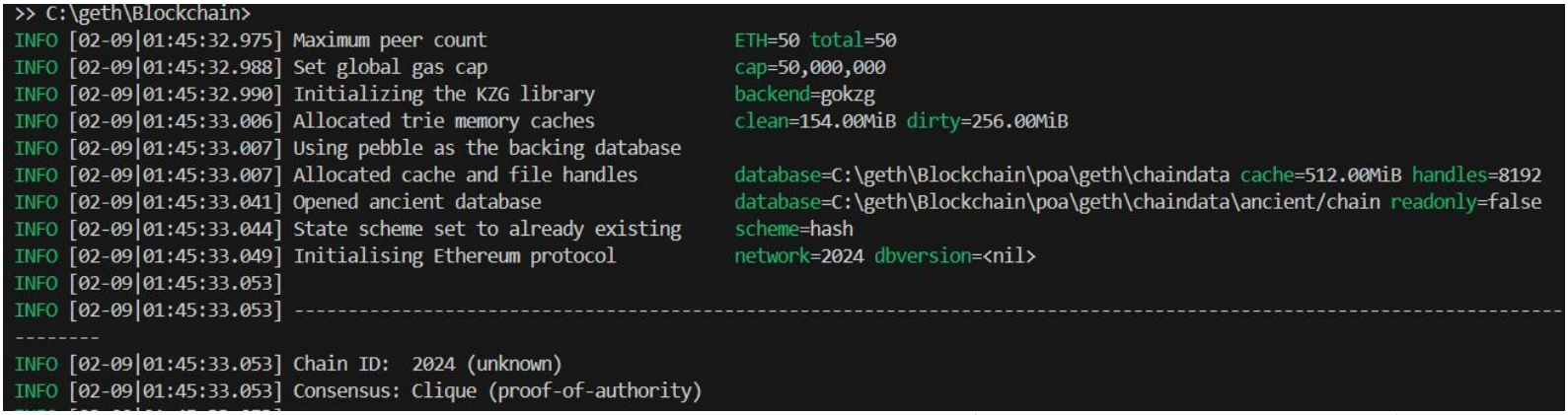
3.4.1 Résultats sur la création de la chaîne de blocs

Dans un premier temps, l'initialisation du bloc de genèse a donné les résultats présents dans la figure 3.13 ci-dessous. Il peut être constaté, à la dernière ligne que l'initialisation du bloc a rencontré un succès et celui-ci a pour haché de bloc: 32c052…dd66cb.



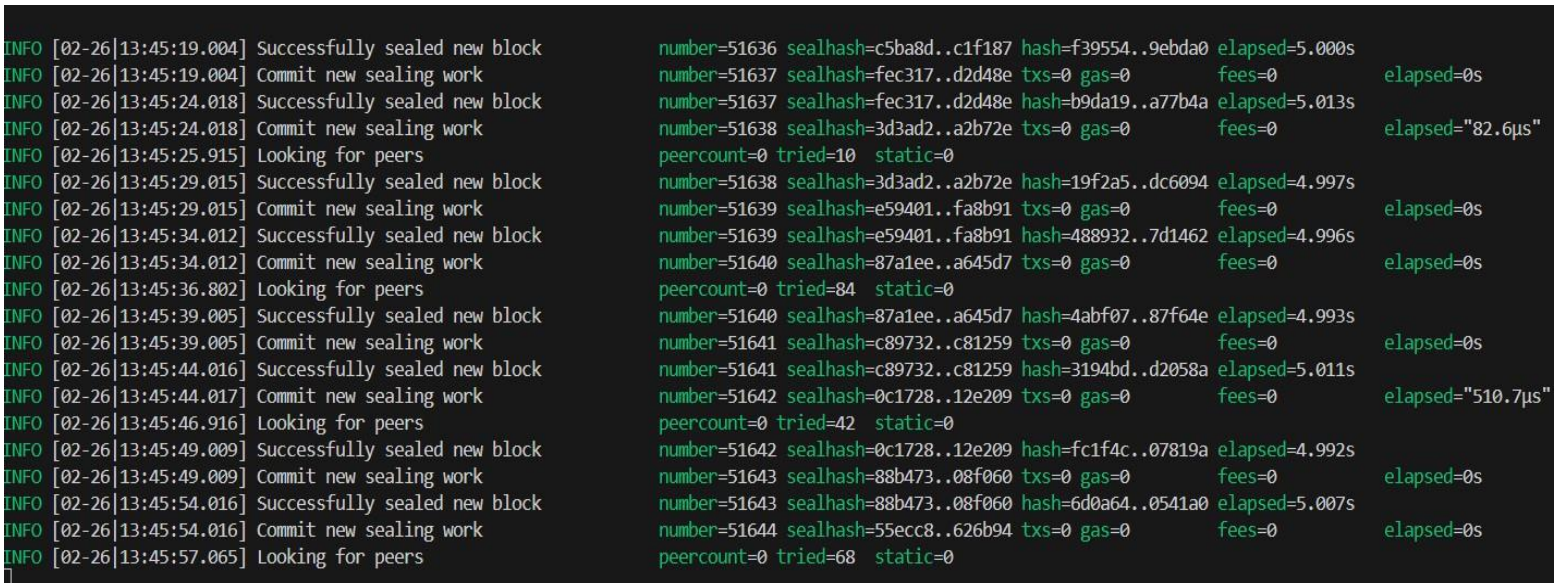
3.13 ***Résultat de l'initialisation du bloc de genèse (Source : S. Franck A. Coulibaly)***

Par la suite, le nœud Geth a été lancé avec la clé privée du validateur. La figure 3-14 montre les résultats du lancement de la blockchain et la figure 3.15 montre la preuve de création des blocs.



3.14 ***Mise en marche de la chaîne de blocs (Source : S. Franck A. Coulibaly)***

3.15 ***Création de blocs (Source : S. Franck A. Coulibaly)***

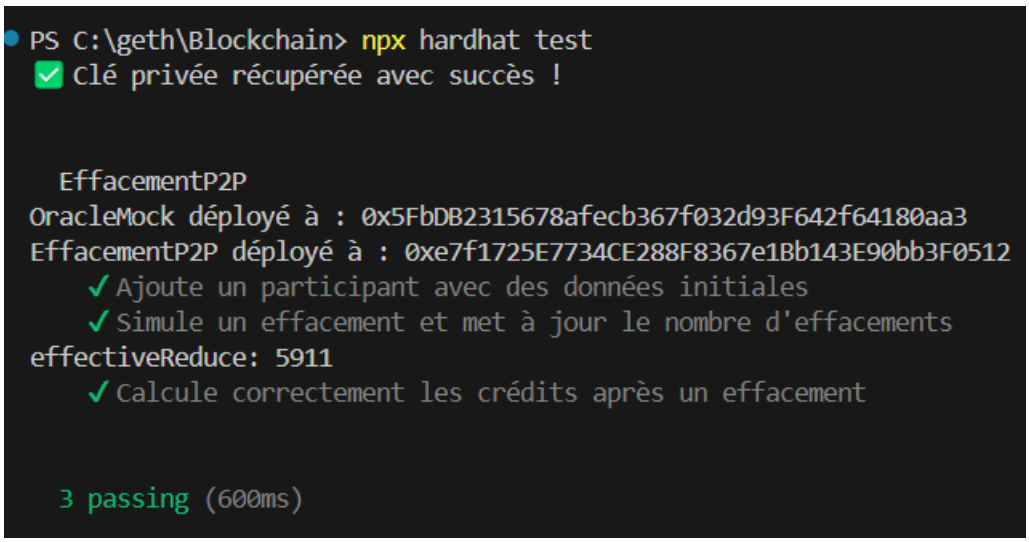


La figure 3-15 montre que la création des blocs se fait sans difficulté prenant pour preuve la mention « Successfully sealed new block » suivi des numéros des blocs, de leur haché et de leur haché de création par le validateur.

La blockchain pleinement fonctionnelle, la section suivante portera sur les résultats obtenus lors des tests effectués sur celle-ci.

3.4.2 Résultats des tests de simulation

Les résultats des tests peuvent être présentés sous deux grands points. Premièrement, il s'agira de présenter les résultats d'un test global afin d'évaluer la fonctionnalité de l'ensemble des fonctions couvertes par le fichier de test. La figure 3.17 présente les résultats obtenus. Les contrats sont déployés sur la blockchain à la même occasion et leurs adresses sont également présentées sur la figure 3.17.



3.16 Résultats du contrôle des fonctionnalités du fichier test (Source : S. Franck A. Coulibaly)

Par la suite, un test par fonction du contrat intelligent a été réalisé. Pour rappel, les fonctions à tester dans le contrat intelligent du programme d'effacement et celui mimant l'oracle sont : d'abord, l'ajout d'un nouveau participant, ensuite, l'activation et la désactivation d'un épisode d'effacement et enfin, le calcul de crédit. L'avantage que présente ces tests est de permettre l'obtention d'un plus grand nombre d'informations notamment les données sur les nouveaux blocs créés. Ces fonctions sont simulées depuis la console Hardhat, et les différentes commandes sont présentées en Annexe F.

Le premier test a porté sur le déploiement du contrat OracleMock, qui fait office d'oracle dans cette étude. Les résultats de son déploiement sont présentés dans la figure 3-16 ci-dessous. Ladite figure fournit les informations sur le bloc créé à la suite du déploiement du contrat. Il est possible d'y trouver le numéro du bloc (Block #1), l'adresse du contrat, le coût de la transaction en gas, le haché du bloc….

```
Contract deployment: OracleMock
Contract address:    0x5fbdb2315678afecb367f032d93f642f64180aa3
Transaction:         0xe4d19dd006d266b09e6e1349832c7bd09df812aefb0756d169d3067f04231ed0
From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
Value:               0 ETH
Gas used:            887326 of 30000000
Block #1:            0x5c9dd429ec90eeacbe3bdd02234c9e3987a9233f97d93f83fe75957a597c0596
```

3.17 ***Bloc généré lors du déploiement du contrat OracleMock (Source : S. Franck A. Coulibaly)***

Par la suite, le contrat intelligent du programme nommé EffacementP2P a été déployé et les informations du bloc généré sont présentées dans la figure 3.17. Les données retrouvées sont semblables à celles présentées dans le Bloc #1 portant sur le déploiement d'OracleMock.

```
Contract deployment: EffacementP2P
Contract address:    0xcf7ed3acca5a467e9e704c703e8d87f634fb0fc9
Transaction:         0x2312ad33a78de4fa897a30975e4afc62787a2348f226c719db71719c57aa502d
From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
Value:               0 ETH
Gas used:            923840 of 30000000
Block #4:            0xa97a1ee7c8e67413930d78b6dff6c9155cc5992b5100ffa7bcd5ad63ee895751
```

3.18 ***Bloc généré lors du déploiement du contrat EffacementP2P (Source : S. Franck A. Coulibaly)***

Les deux contrats étant déployés, les fonctions propres à chacun sont testées. D'abord, la fonction ajout de participant, propre au contrat OracleMock, a été testée et a généré le bloc

dont les informations sont résumées dans la figure 3.18. La présentation des données est également identique au bloc généré précédemment.

```
Contract call:      OracleMock#ajouterParticipant
Transaction:        0x27bf2c3688731c642426f4739248e761492c40e87c3019ac990c659dea06defe
Transaction:        0x27bf2c3688731c642426f4739248e761492c40e87c3019ac990c659dea06defe
From:               0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
From:               0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
To:                 0x9fe46736679d2d9a65f0992f2272de9f3c7fa6e0
Value:              0 ETH
To:                 0x9fe46736679d2d9a65f0992f2272de9f3c7fa6e0
Value:              0 ETH
Value:              0 ETH
Gas used:           118912 of 30000000
Block #5:           0xa49ecc9f397d5fa17f4c74ebcfc6b47b94f1973a4c6a9ac9b6f1f2bf7f69de71
```

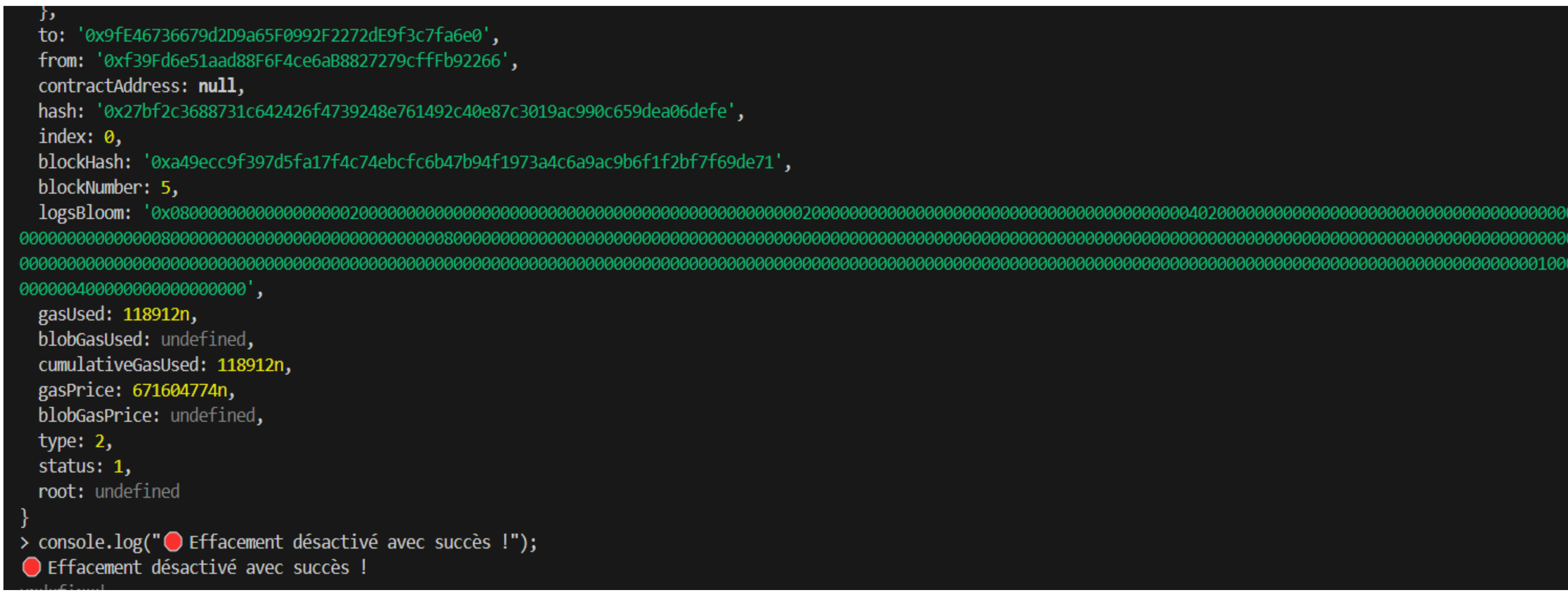
3.19 ***Bloc généré lors de l'exécution de la fonction ajouterParticipant (Source : S. Franck A. Coulibaly)***

Les prochaines fonctions testées sont l'activation et la désactivation de l'effacement. Les informations sur l'activation d'un effacement sont présentées dans la figure 3.19 ci-dessous.

```
  _networkName: 'localhost',
  _blockListeners: [],
  _transactionHashListeners: Map(0) {},
  _eventListeners: []
},
to: '0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0',
from: '0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266',
contractAddress: null,
hash: '0x27bf2c3688731c642426f4739248e761492c40e87c3019ac990c659dea06defe',
index: 0,
blockHash: '0xa49ecc9f397d5fa17f4c74ebcfc6b47b94f1973a4c6a9ac9b6f1f2bf7f69de71',
blockNumber: 5,
logsBloom: '0x080000000000000002000000000000000000000000000000000020000000000000000000000000000000000000004020000000000000000000000000000000000000000
000000000000008000000000000000000000000008000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000
00004000000000000000000',
gasUsed: 118912n,
blobGasUsed: undefined,
cumulativeGasUsed: 118912n,
gasPrice: 671604774n,
blobGasPrice: undefined,
type: 2,
status: 1,
```

3.20 ***Informations sur l'activation de l'effacement (Source : S. Franck A. Coulibaly)***

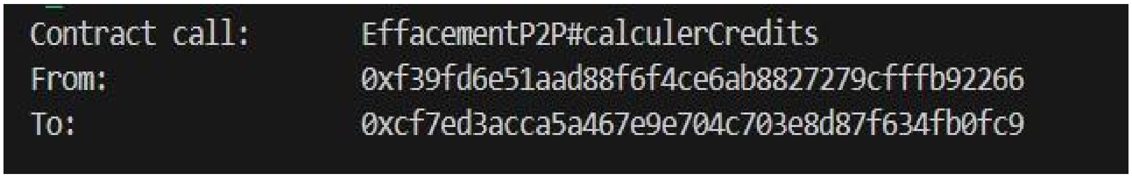
Les informations importantes qu'apportent cette simulation sont dans un premier temps, la réussite de l'activation de l'évènement lié à la fonction, avec pour preuve la dernière ligne sur la figure « status; 1 ». Enfin, les informations telles que le haché de bloc, le numéro du bloc (cette transaction est enregistrée dans le bloc N°5 avec la fonction ajoutParticipant) et le coût de la transaction peuvent y être retrouvés. Dans la figure 3.20, des données identiques sont également présentées, mais cette fois-ci pour la désactivation de l'effacement.



3.21 ***Informations sur la désactivation de l'effacement (Source : S. Franck A. Coulibaly)***

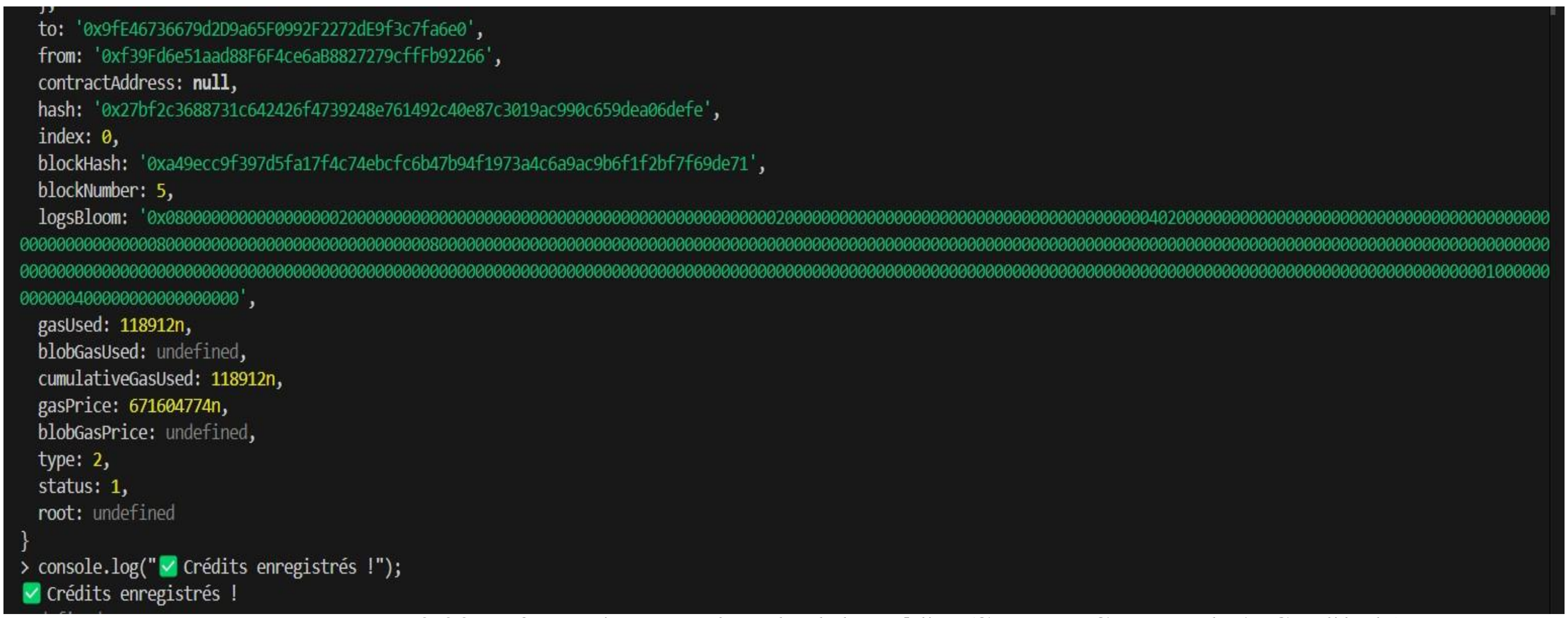
Les fonctions _calculcrédits, enregistrecredit et calculerCrédit sont appelées en dernier. calculerCrédit est d'abord appelée comme présenté dans la figure 3.21, elle permet de calculer la réduction effective moyenne à partir du nombre d'effacements et de la valeur totale des réductions au cours du programme. À la suite de cette opération, les fonctions _calculCrédits et enregistrecredit sont appelés respectivement pour calculer les crédits obtenus lors de la période d'effacement et enregistrer le crédit calculé précédemment. La figure 3.22 montre que l'évènement annonçant le calcul de crédit (évènement lié à la fonction

enregistrecredit) a été parfaitement émis et qu'un crédit a été calculé et enregistré dans l'oracle. Les données de la transaction ont été stockées dans le bloc N°5.



```
Contract call:       EffacementP2P#calculerCredits
From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
To:                  0xcf7ed3acca5a467e9e704c703e8d87f634fb0fc9
```

3.21 Appel de la fonction calculerCredits (Source : S. Franck A. Coulibaly)



```
  to: '0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0',
  from: '0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266',
  contractAddress: null,
  hash: '0x27bf2c3688731c642426f4739248e761492c40e87c3019ac990c659dea06defe',
  index: 0,
  blockHash: '0xa49ecc9f397d5fa17f4c74ebcfc6b47b94f1973a4c6a9ac9b6f1f2bf7f69de71',
  blockNumber: 5,
  logsBloom: '0x0800000000000000002000000000000000000000000000000000000000000200000000000000000000000000000000000000000000040200000000000000000000000000000000000000000000
000000000000000800000000000000000000000000000800000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000000
000000400000000000000000',
  gasUsed: 118912n,
  blobGasUsed: undefined,
  cumulativeGasUsed: 118912n,
  gasPrice: 671604774n,
  blobGasPrice: undefined,
  type: 2,
  status: 1,
  root: undefined
}
> console.log("✅ Crédits enregistrés !");
✅ Crédits enregistrés !
```

3.22 ***Informations sur le calcul de crédits (Source : S. Franck A. Coulibaly)***

3.5 Conclusion

À l'issue de ce chapitre, un modèle de gestion du programme d'effacement de la consommation électrique, utilisant la technologie blockchain et mimant celui de la société Hydro-Québec, a été proposé et testé. Les tests ont montré que le modèle proposé est capable d'ajouter de nouveaux participants au programme, d'émettre des évènements marquant le début et la fin de chaque épisode d'effacement et de calculer les réductions et les crédits sur facture.

Fort de ce modèle, le chapitre suivant présentera une étude comparative des modèles, celui en cours d'utilisation et celui proposé.

Chapitre 4 - Étude comparative du programme d'effacement de la consommation électrique d'Hydro-Québec et celui proposé

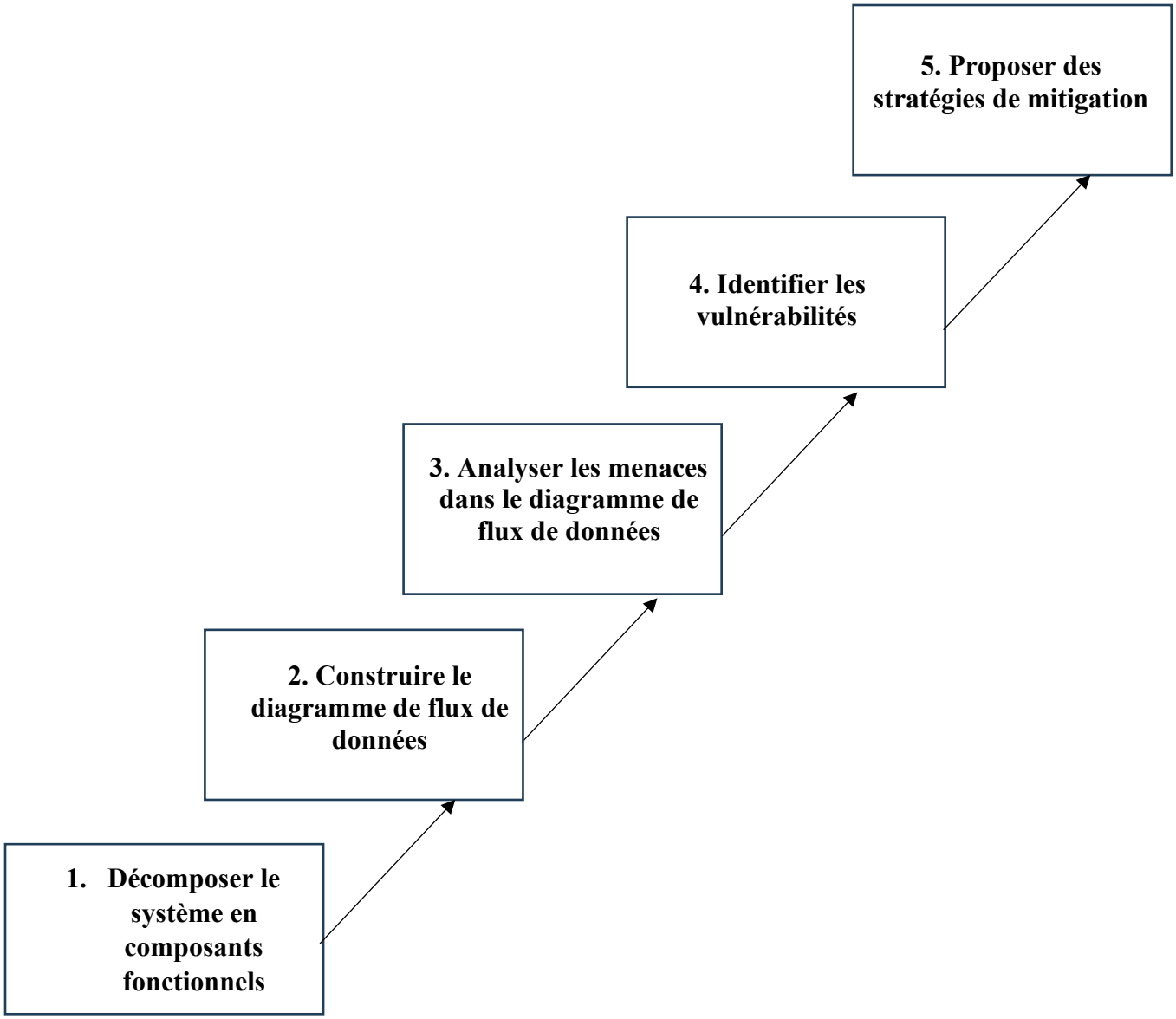
L'objectif de ce chapitre sera de proposer une comparaison objective entre le modèle proposé et celui utilisé par la compagnie Hydro-Québec, d'une part, et, d'autre part, de présenter les perspectives potentielles pour ces modèles. Dans un souci d'harmonisation des appellations dans ce chapitre, le modèle en cours d'utilisation par la société Hydro-Québec sera désigné par modèle classique et celui utilisant la technologie blockchain sera désigné par modèle proposé.

4.1 Répertoire des risques liées aux modèles

Comme point de départ pour ce chapitre, le répertoire des risques liés à l'utilisation des deux modèles sera présenté. Le but de cette partie est de montrer que le modèle proposé apporte des réponses aux risques liés à l'utilisation du modèle classique. Afin de parvenir à cela, l'analyse Spoofing (Usurpation), Tampering (Altération des données), Repudiation (Négation d'action), Denial of service (Indisponibilité de service) et Elevation of Privilege (Élévation des privilèges) ou analyse STRIDE sera utilisée.

L'analyse STRIDE est une analyse de risques en cybersécurité développée par Loren Khonfelder et Praerit Garg [64] et adoptée par Microsoft dans l'implémentation de son cycle de vie de développement logiciel sécurisé [65]. Cette analyse met l'accent sur cinq aspects

de sécurité primordiaux et peut se subdiviser en cinq étapes selon les travaux des auteurs [66] : Premièrement, il s'agira de décomposer le système étudié en composants fonctionnels. Deuxièmement, produire le diagramme de flux de données résultant de cette décomposition. Troisièmement, identifier les menaces dans le diagramme. Quatrièmement, identifier les vulnérabilités et enfin proposer des stratégies de mitigation. La figure 4-1 présente les étapes de la méthodologie STRIDE.



4.1 ***Les étapes de l'analyse STRIDE selon les auteurs [66]***

Le prochain point de ce chapitre portera sur l'application de cette analyse sur le modèle classique.

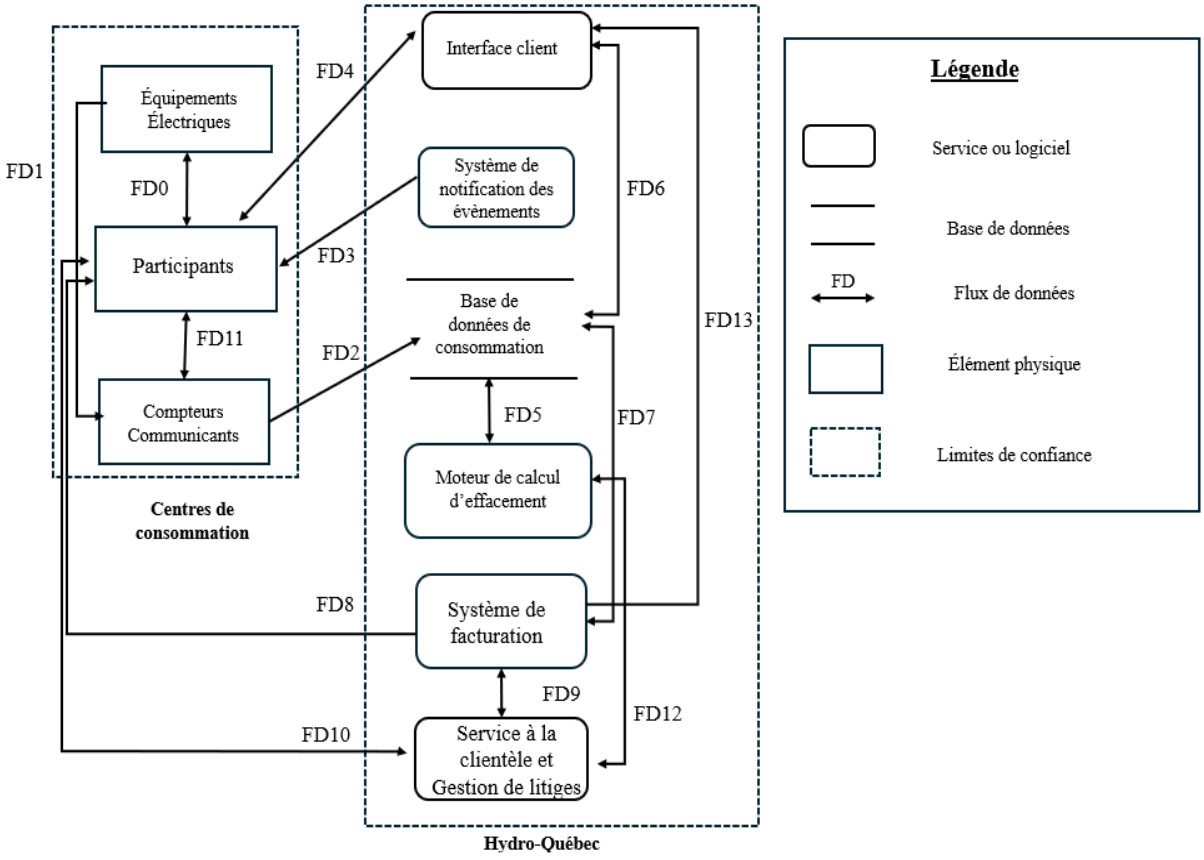
4.1.1 L'analyse STRIDE appliquée au modèle classique

Comme présenté dans la figure 4-1, la première étape de l'analyse choisie consiste à décomposer le système étudié en composants fonctionnels. La décomposition suivante peut être proposée pour le modèle classique :

- Les équipements électriques et les participants au programme
- Les infrastructures de comptage avancées, qui sont des appareils mesurant et communiquant les données de consommation et d'identification des clients.
- La base de données de consommation où sont stockées les données de consommation et les informations des clients.
- Le système de notification des événements permet de prévenir les clients de l'occurrence d'événements de pointe.
- Le moteur de calcul de l'effacement est un logiciel intégré qui permet de calculer les réductions réelles et les crédits sur la base d'une consommation de référence. Il faut noter qu'entre la base de données et le moteur de calcul, il existe une étape de validation et de correction des données effectuée au sein d'Hydro-Québec. En effet, les données envoyées par les infrastructures de comptage peuvent être inexactes ou manquantes. Cette étape ne sera pas présentée dans le diagramme de flux de données à venir, mais l'analyse du flux de données entre les deux éléments tiendra en compte les conséquences liées à cette étape.

- Le système de facturation et d'attribution des crédits permet d'attribuer des crédits aux clients à partir des résultats du moteur.
- Le support client pour la gestion d'éventuels litiges.

Tous les éléments cités plus haut sont répertoriés dans le diagramme de flux de données présenté dans la figure ci-dessous.



4.2 ***Diagramme de flux de données du modèle classique (Source : S. Franck A. Coulibaly)***

La figure 4-2 montrant le diagramme de flux de données du modèle classique présente deux limites de confiance qui sont les centres de consommation et Hydro-Québec. Il peut également être observé sur la figure, les différents flux de données dans le système. Quatorze flux de données peuvent être recensés. La connaissance de ces flux permet d'effectuer

l’analyse STRIDE et de développer des mesures de mitigation. Le tableau 4.1 ci-dessous énumère les catégories STRIDE et donne pour chacune de ces catégories, le risque identifié, les conséquences possibles et les mesures de mitigation. Ces derniers restent non exhaustifs.

4.1 ***Catégories STRIDE et mesures de mitigation du modèle classique (Source : S. Franck A. Coulibaly)***

Catégories STRIDE	Flux concernés	Risque identifié	Conséquences possibles	Mesures de mitigation
Usurpation (Spoofing)	FD3, FD10, FD13	Campagne d'Hameçonnage sur les participants	Usurpation d'identité, fraude client, vol de données.	Signature numérique et meilleur authentification
Altération (Tampering)	FD2, FD5, FD7, FD12	Altération des données via des attaques comme le Man in the middle ou des erreurs de correction (FD5)	Facturation erronée, perte d'intégrité des données de calcul.	Conserver les traces horodatées des opérations et vérifier la cohérence des données
Négation d'action (Repudiation)	FD9, FD10	Risque qu'un employé ou client nie une action.	Difficulté à tracer les actions et prouver les opérations.	Journalisation infalsifiable, horodatage, et traçabilité complète des opérations.
Divulgation d'information (Information Disclosure)	FD8, FD9, FD10, FD13	Fuite non autorisée des informations	Atteinte à la vie privée	Chiffrement des données et anonymisation, contrôle d'accès
Indisponibilité (Denial of Service)	FD8, FD10, FD13	Attaque par déni de service distribué	Interruption des services	Redondance des serveurs
Élévation des privilège	FD5, FD7, FD9, FD12	Employé obtenant des pouvoirs étendus	Manipulation des données	Amélioration du principe de moindre privilège et la segmentation réseau

L'analyse STRIDE du modèle classique étant terminée, la section suivante portera sur l'analyse STRIDE du modèle proposé.

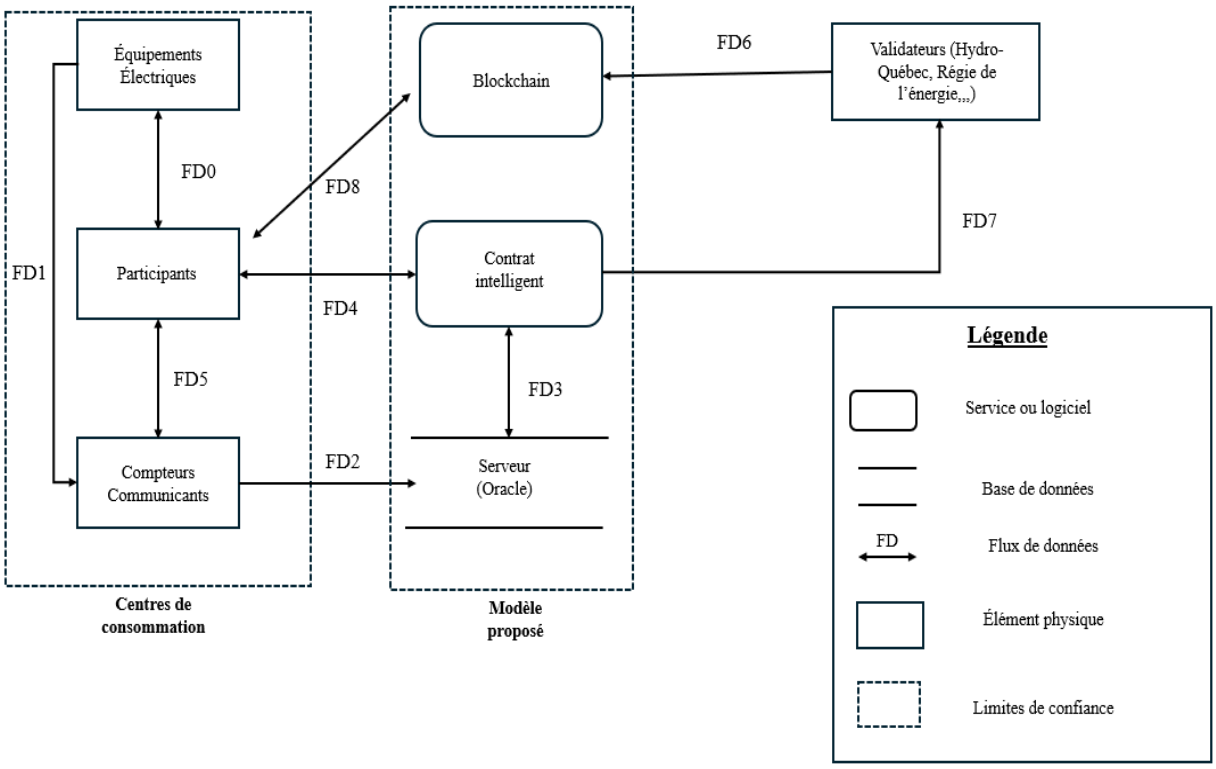
4.1.2 L'analyse STRIDE appliquée au modèle proposé

À la différence du modèle classique, l'étude du modèle proposé a mis l'accent sur l'évaluation de la gestion du programme « demand response option » avec la technologie blockchain. Les services tels que le service clientèle et la gestion des litiges n'ont pas été développés et ne seront donc pas abordés dans l'analyse du modèle proposé.

Suivant l'ordre chronologique des étapes de l'analyse STRIDE proposé les auteurs [66], les composants fonctionnels du modèle proposé suivants peuvent être proposés :

- Les composants tels que équipements électriques, participants et compteurs communicants formant la limite de confiance centres de consommation lors de l'analyse du modèle classique.
- La blockchain permettant de stocker les informations essentielles du programme
- Le contrat intelligent chargé de la gestion des étapes essentielles du programme
- Les validateurs s'assurant de l'authenticité des transactions
- Un serveur à l'image de la base de données de consommation qui stocke les donnes de consommations durant le programme afin d'éviter une surcharge de la chaine de blocs.

La figure 4-3 ci-dessous présente le diagramme de flux de données du modèle proposé, montrant ses différents composants et les flux de données.



4.3 ***Diagramme de flux de données du modèle proposé (Source : S. Franck A. Coulibaly)***

Par la suite, l'analyse du diagramme de flux de données permet d'identifier les échanges d'informations entre les différentes entités du système, ainsi que les vulnérabilités et risques associés à ces échanges. Le tableau 4.2 à l'image du 4.1 résume les risques, les conséquences et les mesures de mitigation associés à aux différentes catégories STRIDE. Il est à noter que les risques, les conséquences et mesures de mitigation proposés sont non exhaustifs.

4.2 Catégories STRIDE et mesures de mitigation du modèle proposé (Source : S. Franck A. Coulibaly)

Catégories STRIDE	Flux concernés	Risques identifiés	Conséquences possibles	Mesures de mitigation
Usurpation (Spoofing)	FD6, FD7, FD8	Usurpation d'identité d'un participant ou d'un validateur par vol de clé privée,	Crédit accordé à tort, Fraude.	Module de sécurité matérielle pour les clés privées. Oracle certifié
Altération (Tampering)	FD6, FD7	Falsification des données envoyés à l'oracle (serveur) et/ou au contrat Bloc modifié par un validateur compromis.	Litige, calculs erronés	Redondance de serveur (validation croisée) Journalisation immuable des validations
Négation d'action (Repudiation)	FD8, FD6	Participant ou validateur nie une transaction.	Litige notamment sur la validation Perte de confiance des participants au programme	Archivage sécurisé des transactions Signatures infalsifiables
Divulgation d'information (Information Disclosure)	FD8	Exposition de données sensibles : profils de charge, habitudes par une dé anonymisation d'adresse blockchain	Violation de la loi 25 du Québec (atteinte à la vie privée)	Aucune donnée sensible sur la blockchain Pseudonymisation des participants
Indisponibilité (Denial of Service)	FD6, FD7, FD8	Cyberattaque de l'Oracle Mauvaise conception du contrat intelligent	Dysfonctionnement du réseau Calculs retardés	Monitoring réseau continu
Élévation des privilège	FD6, FD7	Contrat intelligent manipulé par une mise à jour non consentie Participant qui tente de devenir validateur	Contrôle du réseau par un acteur malveillant	Réseau PoA fermé avec un nombre restreint de validateurs Impossibilité d'une mise à jour sans vote ou multisignature

À l'issue de l'analyse STRIDE des deux modèles, il ressort que chacun d'eux comporte des risques liés à son utilisation. Cependant, le modèle proposé apporte quelques réponses à certains risques de sécurité inhérents au modèle classique. En effet, les principaux avantages du modèle proposé sont son auditabilité et sa décentralisation du stockage des données, à l'opposé du modèle classique, où les données sont centralisées et conservées au sein d'Hydro-Québec. En outre, le modèle proposé prévient la survenue des risques STRIDE notamment l'élévation des privilèges et la négation d'action par la présence d'une chaine de blocs quasi immuable. Par conséquent, ce modèle permet une transparence et une confiance accrues des acteurs.

Le tableau 4.3 ci-dessous résume les avantages apportés par le modèle proposé à chacune des grandes étapes du programme :

4.3 ***Réponses apportées à quelques risques du modèle classique (Source : S. Franck A. Coulibaly)***

Grandes étapes du programme	Risques associés au modèle classique	Réponses du modèle proposé
Inscription	Données stockées au sein d'Hydro-Québec uniquement, situation pouvant entraîner un problème de confiance	Données des participants enregistrées sur la blockchain qui est immuable et décentralisée
Période d'effacement		
Fin d'une période d'effacement		
Calcul de crédits	Réaliser à l'interne avec un processus connu qu'au sein d'Hydro-Québec	Contrat intelligent transparent et audit possible
Litiges	Données enregistrées et traitées par Hydro-Québec	Données facilement consultables sur la blockchain notamment ceux de l'inscription des participants et des périodes d'effacement

À l'issue de l'analyse STRIDE des deux modèles, la section suivante portera sur la recherche de critères pertinents afin de poursuivre la comparaison entre les deux modèles au-delà des risques.

4.2 Proposition de critères d'évaluation

Afin de réaliser une comparaison objective, il est nécessaire d'identifier des critères pertinents d'évaluation. Dans la littérature, il ressort de l'étude des auteurs [1] le critère d'identification des points de vulnérabilité des systèmes . À ce sujet, les auteurs ont mis l'accent sur les systèmes dits « centralisés », tels que le modèle classique, où le traitement et le stockage des données sont réalisés au niveau du gestionnaire de réseau. Cette analyse peut également être réalisée avec les systèmes utilisant la technologie blockchain, car des points de vulnérabilité peuvent être également identifiés pour ce type de modèle. En plus du critère d'analyse des vulnérabilités, d'autres tels que la transparence et la sécurité des données sont abordées notamment par les auteurs [51], [67] et [68] pour le critère de transparence et les auteurs [69] pour la sécurité des données. Enfin, les critères tels que les modes de fonctionnement et la complexité d'implémentation des deux modèles ont également été jugés pertinents.

Le tableau 4.4 résume les critères de comparaison des deux modèles.

4.4 ***Critères d'évaluation des deux modèles (Source : S. Franck A. Coulibaly)***

Critères d'évaluation	Modèle classique	Modèle proposé
Mode de fonctionnement	Centralisé	Moins centralisé avec une réduction des pouvoirs du propriétaire
Points de vulnérabilité	Le gestionnaire de réseau	Serveur informatique (oracle)
Transparence	Données manipulables par des personnes tierces et accessibilité des données soumises à l'approbation du gestionnaire de réseau	Plus grande transparence avec l'immutabilité de la blockchain et plus grande accessibilité des données par les consommateurs
Sécurité des données	Historique de cyberattaques avéré (1224 pour l'année 2024) au sein d'Hydro-Québec notamment des attaques par déni de service distribué [70]	La nature décentralisée de la chaine de blocs peut permettre de lutter contre le déni de service distribué
Complexité d'implémentation	Plus facile à implémenter	Difficulté à l'implémenter avec des infrastructures traditionnelles

À la vue du tableau 4.4, il est possible d'observer dans un premier temps, des avantages et de limites pour chacun des modèles étudiés. Cependant, le modèle proposé se démarque par une augmentation du niveau de sécurité, de transparence et de confidentialité des informations. Cependant, il reste toujours vulnérable aux cyberattaques et plus difficile à implémenter que le modèle classique. Quant à ce dernier, il reste plus facile à implémenter, mais présente un niveau de transparence inférieur à celui du modèle proposé et un faible niveau de protection des données.

Dans l'objectif de pousser encore plus loin l'analyse afin d'entrevoir les perspectives qu'offrent les deux modèles, l'analyse SWOT a été utilisée dans la section suivante.

4.3 Analyse SWOT

L'analyse SWOT peut être définie comme un outil utilisé pour la planification et le management stratégique dans une organisation [71]. Dans le cadre d'une entreprise ou d'une organisation, elle peut permettre une redéfinition ou une clarification de la vision stratégique en fonction des réalités rencontrées. L'analyse SWOT comprend quatre parties : les forces, les faiblesses, les opportunités et les menaces. Les forces et les faiblesses sont des faits perceptibles à court et moyen terme. Les opportunités et menaces quant à elles sont des faits perceptibles à plus long terme. En d'autres mots, les opportunités et les menaces peuvent être désignées comme les conséquences sur le long terme de respectivement les forces et les faiblesses.

Dans cette étude, l'analyse SWOT sera utilisée pour analyser les deux modèles présentés

4.3.1 Analyse SWOT du modèle classique

Le tableau 4.5 ci-dessous présente la matrice de l'analyse SWOT du modèle classique. À noter que les arguments présentés dans le tableau sont non exhaustifs.

4.5 ***Matrice de l'analyse SWOT du modèle classique (Source : S. Franck A. Coulibaly)***

Forces	Modèle maîtrisé, car utilisé pendant une assez longue période, il a donc fait ses preuves. Il permet d'encourager l'usage des énergies renouvelables ou des sources décentralisées.
Faiblesses	Centralisation du modèle le rendant dépendant du gestionnaire de réseau. Les délais de traitement des données peuvent être longs.
Opportunités	Amélioration des performances avec les réseaux intelligents. Contribue à l'atteinte de la carboneutralité de la province.
Menaces	Risque de cyberattaques pouvant paralyser le fonctionnement du modèle (attaque par déni de service distribué). Difficultés à faire face à un nombre élevé d'informations à traiter.

4.3.2 Analyse SWOT du modèle proposé

Le tableau 4.6 ci-dessous présente la matrice d'analyse SWOT du modèle proposé. Ici également, les arguments présentés ne sont pas exhaustifs.

4.6 ***Matrice de l'analyse SWOT du modèle proposé (Source : S. Franck A. Coulibaly)***

Forces	Automatisation des tâches avec les contrats intelligents Une transparence accrue avec l'immutabilité de la chaîne de blocs
Faiblesses	Peut-être difficilement interfaçable avec les infrastructures énergétiques traditionnelles Possibilité d'avoir un système paralysé ou inopérant si l'oracle est corrompu
Opportunités	Réduction des coûts et de la bureaucratie lors de l'exécution du programme Renforcement de la confiance entre les acteurs du programme
Menaces	Augmentation des coûts pour la mise à niveau des infrastructures énergétiques Augmentation possible des cyberattaques sur le serveur car point de faiblesse centrale

Sur son site officiel [72], Hydro-Québec donne un délai de trois cycles de facturation pour l'émission des montants des crédits. Trois cycles de facturation correspondent à trois mois (2160 heures) ou à six mois (4320 heures), selon le système de facturation (mensuel ou

bimensuel). À l'opposé, les temps de latence des chaînes de blocs à permission sont faibles, de l'ordre de quelques secondes, inférieurs à la minute, comme le montrent les études des auteurs [73] et [74]. En y ajoutant le temps de traitement des données issues des infrastructures de comptage, qui est de 24 heures (Hydro-Québec actualise les données de consommation des clients chaque jour avec les données corrigées de la veille), le modèle proposé pourrait consacrer environ 90 ou 180 fois moins de temps (mensuelle ou bimensuelle) au calcul du crédit.

4.4 Discussions

À l'issue de la présentation des matrices d'analyse SWOT des deux modèles, il est possible d'affirmer, à la lumière du tableau 4-3, que le modèle classique est un modèle bien connu, car il est utilisé depuis une période assez longue. C'est également un modèle dont l'implémentation a été affinée afin de répondre aux objectifs recherchés. Par conséquent, les perspectives pour ce modèle pourraient consister en une amélioration des performances avec l'arrivée des réseaux intelligents, qui permettront de meilleures analyses et décisions grâce à l'accessibilité à des données complémentaires sur le système électrique. De plus, le programme d'effacement, demandant aux participants de réduire leur consommation durant les périodes de forte consommation, pourrait pousser à des investissements dans des sources d'énergie autonomes ou décentralisées, comme les énergies renouvelables, contribuant à l'atteinte des objectifs de carboneutralité de la province. Par contre, ce modèle reste vulnérable aux cyberattaques, comme l'atteste l'historique de la compagnie Hydro-Québec à ce sujet. En effet, sa nature non distribuée constitue un point de faiblesse majeur.

D'autre part, le tableau 4-3 montre que le modèle proposé automatiserait les tâches à travers des contrats intelligents, ce qui aurait pour conséquence une éventuelle réduction des

coûts de fonctionnement (notamment moins de litiges). De plus, le modèle proposé permet d'augmenter la transparence, car les informations sur la chaîne de blocs sont accessibles à tous et non modifiables une fois mises en ligne sur la blockchain. Cet état de fait pourrait, à terme, renforcer la confiance entre les acteurs du programme. Cependant, des limites peuvent être identifiées dans le modèle proposé. Son implémentation reste actuellement très dépendante de nouvelles infrastructures ou appareils de communication. Cette situation pourrait entraîner des coûts supplémentaires notamment lors de la mise à niveau des infrastructures énergétiques. De plus, le fonctionnement du modèle proposé dépend du bon fonctionnement et de l'exactitude des données provenant de l'oracle, ce qui place le point de vulnérabilité de ce modèle au sein de l'oracle. Ceci pourrait entraîner, à long terme, une augmentation des cyberattaques ciblant l'oracle.

4.5 Conclusion

Dans ce chapitre, une comparaison entre le modèle classique et celui proposé a été effectuée. L'objet de cette comparaison était de comprendre l'impact des deux modèles sur le programme d'effacement de la consommation. Pour ce faire, l'analyse STRIDE des deux modèles a été réalisée afin de mettre en évidence les risques liés à leur utilisation respective. Ensuite, une proposition de critères pertinents de comparaison a été établie, puis une comparaison a été réalisée sur la base de ces critères. Enfin une analyse SWOT des deux modèles a été présentée afin d'entrevoir les potentielles perspectives. Il a donc été montré que le modèle proposé garantissait une meilleure protection des données, automatisait les tâches et accroissait la transparence entre les acteurs. Cependant, il reste vulnérable à l'égard de l'oracle et difficile à implémenter par rapport au modèle classique.

Chapitre 5 - Conclusion générale

La finalité de cette étude était d'évaluer l'impact de la technologie blockchain sur la réponse à la demande, plus spécifiquement sur l'un des programmes d'effacement de la consommation électrique dans la province du Québec : le « demand response option ». Pour se faire, le travail a été subdivisé en plusieurs points.

Dans un premier temps, il a consisté à réaliser une revue de la littérature sur les mots-clés suivants : l'effacement de la consommation électrique et la technologie Blockchain, afin de pouvoir les appréhender pleinement. La revue de littérature visait également à mettre en évidence des études similaires afin de comprendre l'état de l'art sur la question.

Dans un second temps, l'implémentation d'un programme d'effacement de la consommation électrique, utilisant la technologie blockchain et suivant les méthodes de celui actuellement utilisé au Québec, a été présentée. Pour se faire, la revue de littérature a été analysée afin d'identifier les caractéristiques les plus rencontrées et essentielles au bon fonctionnement du programme dans ce type d'étude. Ces caractéristiques sont : le type de blockchain, le mécanisme de consensus et la plateforme. À l'issue de l'analyse, il a été retenu pour cette étude, une chaîne de blocs privée utilisant la preuve d'autorité pour la validation des transactions et fonctionnant sur la plateforme Etherum. Par la suite, le programme a été simulé. Les simulations ont visé à démontrer que le nouveau modèle est capable d'ajouter de nouveaux participants au programme, d'activer ou de désactiver un épisode d'effacement, et enfin de calculer les crédits.

Pour terminer, une étude comparative a été réalisée afin de mettre en évidence les différences entre les deux modèles et d'aborder les atouts et les limites du modèle proposé. À cette fin, cette partie de l'étude a été scindée en trois parties avec une première partie visant à faire le répertoire des risques en lien avec les deux modèles, une deuxième partie portant sur une comparaison des modèles avec pour base des critères d'évaluation sélectionnés et enfin une partie utilisant l'analyse SWOT dans le but de mettre en avant les perspectives.

À l'issue de cette étude, plusieurs pistes d'amélioration peuvent être suggérées. Il y a d'abord l'utilisation de testnet comme Sepolia ou Goerli, qui ont pour avantage des conditions de simulation, proches du réel. Ensuite, la possibilité d'explorer une option incluant l'usage de tokens ou de cryptomonnaies pour rémunérer les acteurs du programme. Enfin, l'implémentation d'infrastructures adéquates pourrait permettre d'améliorer le modèle proposé, grâce notamment à l'évaluation de l'engouement des participants pour la technologie blockchain et à la correction du modèle sur des cas pratiques et concrets. C'est dans cette optique qu'il est proposé comme suite de cette étude, son implémentation dans la ville de Gaspé dans la province du Québec en collaboration avec le centre de recherche appliquée Nergica. Pour terminer; l'analyse du document par le logiciel anti-plagiat Grammarly est disponible en Annexe I.

Bibliographie

Mots-clés utilisés lors de la recherche bibliographique : DSM, Blockchain, Bitcoin, Blockchain Demand Response Program, Hydro-Québec Demand Response Option, Québec Demand Response, SHA-256 explanation, Ethereum, Go Ethereum, SWOT Analysis. STRIDE Analysis

Banque de données : IEEE Xplore, Elsevier, ScienceDirect, Bibliothèque de l'Université du Québec à Trois-Rivières, Bibliothèque de l'Université de Carleton.

[1] C. Pop, T. Cioara, M. Antal, I. Anghel, I. Salomie, and M. Bertoncini, "Blockchain based decentralized management of demand response programs in smart energy grids," *Sensors,* vol. 18, no. 1, p. 162, 2018.

[2] J. D. Khazzoom, "Economic implications of mandated efficiency in standards for household appliances," ed: SAGE Publications Sage CA: Los Angeles, CA, 1980.

[3] L. Brookes, "Energy policy, the energy price fallacy and the role of nuclear energy in the UK," *Energy Policy,* vol. 6, no. 2, pp. 94-106, 1978.

[4] M. M. Eissa, "Demand side management program evaluation based on industrial and commercial field data," *Energy Policy,* vol. 39, no. 10, pp. 5961-5969, 2011.

[5] D. Stanelyte, N. Radziukyniene, and V. Radziukynas, "Overview of demand-response services: A review," *Energies,* vol. 15, no. 5, p. 1659, 2022.

[6] C. W. Gellings and K. E. Parmenter, "Demand-side management," in *Energy management and conservation Handbook*: CRC Press, 2016, pp. 399-420.

[7] Y. Yan, J. Huang, X. Chen, Z. Zhang, T. Zhang, and Z. Lin, "Blockchain-based framework of power demand response in China," *IET Renewable Power Generation,* vol. 16, no. 4, pp. 781-791, 2022.

[8] X. Mao *et al.*, "Centralized bidding mechanism of demand response based on blockchain," *Energy Reports,* vol. 8, pp. 111-117, 2022.

[9] C. W. Gellings and J. H. Chamberlin, "Demand-side management: concepts and methods," 1987.

[10] V. Dufresne, "The Value of Demand Response in a Hydro-Dominated Power Grid-The Example of Quebec, Canada," Carleton University, 2016.

[11] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 1, pp. 152-178, 2014.

[12] M. E. Honarmand, V. Hosseinnezhad, B. Hayes, M. Shafie-Khah, and P. Siano, "An overview of demand response: From its origins to the smart energy community," *IEEE Access,* vol. 9, pp. 96851-96876, 2021.

[13] A. Conchado and P. Linares, "The economic impact of demand-response programs on power systems. A survey of the state of the art," *Handbook of networks in power systems I,* pp. 281-301, 2012.

[14] Hydro-Québec."Demand Response Offer." https://www.hydroquebec.com/business/customer-space/rates/demand-response-option.html (accessed march 27, 2023).

[15] Hydro-Québec "Demand Response Eligbility." https://www.hydroquebec.com/business/customer-space/rates/demand-response-option-eligibility.html (accessed August 03, 2023).

[16] Hydro-Quebec."Demand Response Option." https://www.hydroquebec.com/business/customer-space/rates/demand-response-option-credit.html (accessed march 05, 2025).

[17] M. Andoni *et al.*, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and sustainable energy reviews,* vol. 100, pp. 143-174, 2019.

[18] S. Haber and W. S. Stornetta, *How to time-stamp a digital document*. Springer, 1991.

[19] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review,* p. 21260, 2008.

[20] R. C. Merkle, "One way hash functions and DES," in *Conference on the Theory and Application of Cryptology*, 1989: Springer, pp. 428-446.

[21] I. B. Damgård, "A design principle for hash functions," in *Conference on the Theory and Application of Cryptology*, 1989: Springer, pp. 416-427.

[22] L. Tables. "ASCII Table." https://www.lookuptables.com/text/ascii-table (accessed 23 septembre, 2024).

[23] K. Devika and R. Bhakthavatchalu, "Parameterizable FPGA implementation of SHA-256 using blockchain concept," in *2019 International conference on communication and signal processing (ICCSP)*, 2019: IEEE, pp. 0370-0374.

[24] Z. Al-Odat, A. Abbas, and S. U. Khan, "Randomness analyses of the secure hash algorithms, SHA-1, SHA-2 and modified SHA," in *2019 International Conference on Frontiers of Information Technology (FIT)*, 2019: IEEE, pp. 316-3165.

[25] H. Delfs, H. Knebl, and H. Knebl, *Introduction to cryptography*. Springer, 2002.

[26] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 2018.

[27] M. Go. "What is blockchain ?" https://www.youtube.com/watch?v=XaTqdCgbk7Y (accessed 23 septembre 2023, 2023).

[28] R. P. Naik and N. T. Courtois, "Optimising the sha256 hashing algorithm for faster and more efficient bitcoin mining," *MSc Information Security Department of Computer Science UCL,* pp. 1-65, 2013.

[29] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Conference on the theory and application of cryptographic techniques*, 1987: Springer, pp. 369-378.

[30] R. C. Merkle, "Method of providing digital signatures," ed: Google Patents, 1982.

[31] O. Kuznetsov, A. Rusnak, A. Yezhov, K. Kuznetsova, D. Kanonik, and O. Domin, "Merkle Trees in Blockchain: A Study of Collision Probability and Security Implications," *Internet of Things,* p. 101193, 2024.

[32] V. Buterin, "On public and private blockchains," 2015.

[33] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services,* vol. 14, no. 4, pp. 352-375, 2018.

[34] S. Zhou, K. Li, L. Xiao, J. Cai, W. Liang, and A. Castiglione, "A systematic review of consensus mechanisms in blockchain," *Mathematics,* vol. 11, no. 10, p. 2248, 2023.

[35] K. Mammadzada, M. Iqbal, F. Milani, L. García-Bañuelos, and R. Matulevičius, "Blockchain oracles: A framework for blockchain-based applications," in *Business Process Management: Blockchain and Robotic Process Automation Forum: BPM 2020 Blockchain and RPA Forum, Seville, Spain, September 13–18, 2020, Proceedings 18*, 2020: Springer, pp. 19-34.

[36] Z. Tu and C. Xue, "Effect of bifurcation on the interaction between Bitcoin and Litecoin," *Finance Research Letters,* vol. 31, 2019.

[37] V. Buterin, "Ethereum white paper," *GitHub repository,* vol. 1, pp. 22-23, 2013.

[38] M. L. Di Silvestre, P. Gallo, E. R. Sanseverino, G. Sciume, and G. Zizzo, "Aggregation and remuneration in demand response with a blockchain-based framework," *IEEE Transactions on Industry Applications,* vol. 56, no. 4, pp. 4248-4257, 2020.

[39] E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1-15.

[40] R. G. Brown, "The corda platform: An introduction," *Retrieved,* vol. 27, p. 2018, 2018.

[41] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," University of Guelph, 2016.

[42] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee, "Performance evaluation of the quorum blockchain platform," *arXiv preprint arXiv:1809.03421,* 2018.

[43] A. Dolgui, D. Ivanov, and B. Sokolov, "Ripple effect in the supply chain: an analysis and recent literature," *International journal of production research,* vol. 56, no. 1-2, pp. 414-430, 2018.

[44] T. Laha, A. Bandyopadhyay, K. Deb, and S. Koley, "A Methodical Study on Blockchain Technology," in *International Conference on Network Security and Blockchain Technology*, 2021: Springer, pp. 391-403.

[45] V. Buterin, J. Coleman, and M. Wampler-Doty, "Notes on Scalable Blockchain Protocols (verson 0.3)," ed, 2015.

[46] V. Buterin, "A next-generation smart contract and decentralized application platform," *white paper,* vol. 3, no. 37, pp. 2-1, 2014.

[47] B. Li, I. Banimenia, L. Chuan, H. Zhansheng, and J. Zhao, "Incentive-based demand response program with self-reported baseline supported by blockchain technology," *IET Smart Grid,* vol. 6, no. 2, pp. 205-218, 2023.

[48] J. Gao *et al.*, "GridMonitoring: Secured sovereign blockchain based monitoring on smart grid," *IEEE access,* vol. 6, pp. 9917-9925, 2018.

[49] A. Aoun, H. Ibrahim, M. Ghandour, and A. Ilinca, "Blockchain-enabled energy demand side management cap and trade model," *Energies,* vol. 14, no. 24, p. 8600, 2021.

[50] M. Di Silvestre, P. Gallo, E. R. Sanseverino, G. Sciume, and G. Zizzo, "A new architecture for Smart Contracts definition in Demand Response Programs," in *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, 2019: IEEE, pp. 1-5.

[51] C. D. Pop, M. Antal, T. Cioara, I. Anghel, and I. Salomie, "Blockchain and demand response: Zero-knowledge proofs for energy transactions privacy," *Sensors,* vol. 20, no. 19, p. 5678, 2020.

[52] V. Deshpande, L. George, H. Badis, and A. A. Desta, "Blockchain based decentralized framework for energy demand response marketplace," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*, 2020: IEEE, pp. 1-9.

[53] O. Kanwhen, J. Jain, and A. Mohamed, "Optimization and Scalability of Blockchain Enabled Demand Response Smart Contracts using Sharding and Neural Networks," in *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, 2023: IEEE, pp. 1-5.

[54] A. Porat, A. Pratap, P. Shah, and V. Adkar, "Blockchain Consensus: An analysis of Proof-of-Work and its applications," ed: Stanford University: Stanford, CA, USA, 2017.

[55] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021: IEEE, pp. 279-283.

[56] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities," *IEEE access,* vol. 7, pp. 85727-85745, 2019.

[57] Y. Shifferaw and S. Lemma, "Limitations of proof of stake algorithm in blockchain: A review," *Zede Journal,* vol. 39, no. 1, pp. 81-95, 2021.

[58] S. Joshi, "Feasibility of proof of authority as a consensus protocol model," *arXiv preprint arXiv:2109.02480,* 2021.

[59] M. A. Manolache, S. Manolache, and N. Tapus, "Decision making using the blockchain proof of authority consensus," *Procedia Computer Science,* vol. 199, pp. 580-588, 2022.

[60] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "Distributed consensus protocols and algorithms," *Blockchain for Distributed Systems Security,* vol. 25, p. 40, 2019.

[61] N. Foundation. "Hardhat." https://hardhat.org/ (accessed 01 march, 2025).

[62] OpenJS-Foundation. "Node.js Website." https://nodejs.org/fr (accessed march 08, 2025).

[63] T. g.-e. Authors. "Go Etherum." https://geth.ethereum.org/ (accessed 1 march, 2025).

[64] L. Kohnfelder and P. Garg, "The threats to our products," *Microsoft Interface, Microsoft Corporation,* vol. 33, pp. 1-8, 1999.

[65] M. Da Silva, M. Puys, P.-H. Thevenon, S. Mocanu, and N. Nkawa, "Automated ICS template for STRIDE Microsoft threat modeling tool," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, 2023, pp. 1-7.

[66] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017: IEEE, pp. 1-6.

[67] Y. Qu *et al.*, "Towards privacy-aware and trustworthy data sharing using blockchain for edge intelligence," *Big Data Mining and Analytics,* vol. 6, no. 4, pp. 443-464, 2023.

[68] A. Lucas, D. Geneiatakis, Y. Soupionis, I. Nai-Fovino, and E. Kotsakis, "Blockchain technology applied to energy demand response service tracking and data sharing," *Energies,* vol. 14, no. 7, p. 1881, 2021.

[69] S. Jiang, J. Li, X. Zhang, H. Yue, H. Wu, and Y. Zhou, "Secure and privacy-preserving energy trading with demand response assistance based on blockchain," *IEEE Transactions on Network Science and Engineering,* vol. 11, no. 1, pp. 1238-1250, 2023.

[70] Radio Canada. "Le site web d'Hydro-Québec paralysé." https://ici.radio-canada.ca/nouvelle/1971255/hydro-quebec-panne-cyberattaque (accessed november 22, 2025).

[71] E. Gurl, "SWOT analysis: A theoretical review," 2017.

[72] Hydro-Quebec."Demand Response Option Credit." https://www.hydroquebec.com/business/customer-space/rates/demand-response-option-credit.html (accessed november 22, 2025).

[73] M. Mazzoni, A. Corradi, and V. Di Nicola, "Performance evaluation of permissioned blockchains for financial applications: The ConsenSys Quorum case study," *Blockchain: Research and applications,* vol. 3, no. 1, p. 100026, 2022.

[74] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *CEUR workshop proceedings*, 2018, vol. 2058: CEUR-WS.

Annexe A – Contrat intelligent du programme d'effacement de la consommation électrique (.sol)

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.13;



// Importer OracleMock

import "./OracleMock_version.sol";



contract EffacementP2P {

    address public owner;

    address public oracleAddress;

    bool public effacementActif;



    struct Participant {

        bool enregistre;

        string profilConso;

        uint256 totalReduction;
```

```
        uint256 totalCredits;
        uint256[] chargeEvents;
    }

    mapping (address => Participant) public participants;
    address[] public participantAddresses;

    event DebutEffacementActif(address indexed initiateur, uint256 timestamp);
    event FinEffacement(address indexed initiateur, uint256 timestamp);
    event ChargeEventEnregistre(address indexed participant, uint256 chargeEvent, uint256 timestamp);
    event CreditsCalcules(address indexed participant, uint256 effectiveReduce, uint256 totalCredits, uint256 timestamp);
    event ParticipantEnregistre(address indexed participant, string profilConso, uint256 timestamp);

    modifier onlyOwner() {
        require(msg.sender == owner, "Seul le proprietaire peut executer cette action");
        _;
    }
```

```
modifier onlyRegistered() {
    require(participants[msg.sender].enregistre, "Adresse non inscrite au registre");
    _;
}

modifier effacementEstActif() {
    require(effacementActif, "Aucun effacement n'est actif actuellement");
    _;
}

modifier avantDecembre() {
    uint256 month = (block.timestamp / 30 days) % 12 + 1;
    require(month < 12, "Ajout de participants interdit apres le 1er decembre");
    _;
}

modifier enavril() {
    uint256 year = block.timestamp / 365 days + 1970;
```

```
        uint256 aprilFirst = (year - 1970) * 365 days + 91 days;
        require(block.timestamp >= aprilFirst && block.timestamp < aprilFirst + 1 days,
"Action autorisee uniquement le 1er avril");
        _;
    }

    constructor(address _oracleAddress) {
        require(_oracleAddress != address(0), "Adresse Oracle invalide");
        oracleAddress = _oracleAddress;
        owner = msg.sender;

        participants[msg.sender] = Participant({
            enregistre: true,
            profilConso: "Owner",
            totalReduction: 0,
            totalCredits: 0,
            chargeEvents: new uint256       });

        participantAddresses.push(msg.sender);
```

```
        emit ParticipantEnregistre(msg.sender, "Owner", block.timestamp);
    }

    function ajoutparticipant(address participant, string memory profilConso) public onlyOwner avantDecembre {
        require(!participants[participant].enregistre, "Deja inscrit");
        require(participant != address(0), "Adresse invalide");

        participants[participant] = Participant({
            enregistre: true,
            profilConso: profilConso,
            totalReduction: 0,
            totalCredits: 0,
            chargeEvents: new uint256
        });

        participantAddresses.push(participant);
        emit ParticipantEnregistre(participant, profilConso, block.timestamp);
    }
```

```
function debutEffacement() public onlyOwner {
    require(!effacementActif, "Un effacement est deja en cours");
    effacementActif = true;
    emit DebutEffacementActif(msg.sender, block.timestamp);
}

function finEffacement() public onlyOwner effacementEstActif {
    effacementActif = false;
    emit FinEffacement(msg.sender, block.timestamp);
}

function enregistrerChargeEvent() public onlyRegistered effacementEstActif {
    (bool success, bytes memory data) = oracleAddress.call(
        abi.encodeWithSelector(bytes4(keccak256("getdata(address)")), msg.sender)
    );
    require(success, "Echec lors de la recuperation des donnees de l'Oracle");
```

```
        (string memory profilConso, uint256 seuil, uint256 chargeEvent) = abi.decode(data,
(string, uint256, uint256));
        require(keccak256(abi.encodePacked(profilConso))                    ==
keccak256(abi.encodePacked(participants[msg.sender].profilConso)), "Profil non valide");
        require(chargeEvent <= seuil, "chargeEvent non valide");

        participants[msg.sender].chargeEvents.push(chargeEvent);
        participants[msg.sender].totalReduction += chargeEvent;
        emit ChargeEventEnregistre(msg.sender, chargeEvent, block.timestamp);
    }

    function calculerCredits() public onlyOwner enavril {
        for (uint256 i = 0; i < participantAddresses.length; i++) {
            address participantAddr = participantAddresses[i];
            Participant storage participant = participants[participantAddr];

            uint256 effectiveReduce = _calculEffectiveReduce(participantAddr);
            uint256 credits = _calculCredits(effectiveReduce);
            participant.totalCredits += credits;
```

```
            emit CreditsCalcules(participantAddr, effectiveReduce, participant.totalCredits,
block.timestamp);
        }
    }

    function _calculEffectiveReduce(address participant) internal view returns (uint256) {
        uint256[] storage chargeEvents = participants[participant].chargeEvents;
        uint256 total = 0;
        uint256 count = chargeEvents.length;

        for (uint256 i = 0; i < count; i++) {
            total += chargeEvents[i];
        }

        return count > 0 ? total / count : 0;
    }

     function _calculCredits(uint256 effectiveReduce) internal pure returns (uint256) {
        uint256 credit = 0;
```

```
        uint256 creditPr = 78825;
        uint256 creditPr_Cons = creditPr / 10;
        uint256 creditSc = 273260;
        uint256 creditTr = 75672;

        if (effectiveReduce > 10 && effectiveReduce <= 100) {
            credit = (78825 * effectiveReduce) / 1000;
        } else if (effectiveReduce > 100 && effectiveReduce <= 400) {
            credit = ((68315 * (effectiveReduce - 100)) / 1000) + creditPr_Cons;
        } else if (effectiveReduce > 400 && effectiveReduce <= 1200) {
            credit = ((63060 * (effectiveReduce - 500)) / 1000) + creditSc + creditPr_Cons;
        } else if (effectiveReduce > 1200) {
            credit = ((57805 * (effectiveReduce - 1700)) / 1000) + creditSc + creditPr_Cons
+ creditTr;
        }

        return credit;
    }
```

```
    function isRegistered(address participant) public view returns (bool) {
        return participants[participant].enregistre;
    }
}
```

Annexe B – Programme en Javascript créant le premier bloc de la chaîne de blocs (genesis.json)

```
{
 "config": {
  "chainId": 2024,
  "homesteadBlock": 0,
  "eip150Block": 0,
  "eip155Block": 0,
  "eip158Block": 0,
  "byzantiumBlock": 0,
  "clique": {
   "period": 5,
   "epoch": 30000
  }
 },
 "alloc": {
  "550C3335681388d27DAA7812153bA3AD131C806B": {
   "balance": "100000000000000000000"
  }
```

```
    },
    "difficulty": "1",
    "gasLimit": "30000000",
    "coinbase": "0x550C3335681388d27DAA7812153bA3AD131C806B",
   "extraData":
"0x0000000000000000000000000000000000000000000000000000000000000000550C33
35681388d27DAA7812153bA3AD131C806B0000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000",
    "mixHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
    "nonce": "0x0",
    "timestamp": "0x0"
  }
```

Annexe C – Contrat intelligent de l'oracle (.sol)

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.13;


contract OracleMock {

    struct Data {

        string profilConso;

        uint256 seuil;

        uint256 chargeEvent;

    }


    mapping(address => Data) public dataStore;


    // Event pour tracer les mises à jour de données

    event DataUpdated(address indexed participant, string profilConso, uint256 seuil, uint256
chargeEvent);


    // Fonction permettant d'ajouter ou modifier les données d'un participant
```

```
    function setData(address participant, string memory profilConso, uint256 seuil, uint256 chargeEvent) public {
        require(participant != address(0), "Adresse participant invalide");

        dataStore[participant] = Data(profilConso, seuil, chargeEvent);
        emit DataUpdated(participant, profilConso, seuil, chargeEvent);
    }

    // Fonction appelée par le contrat EffacementP2P pour récupérer les données du participant
    function getdata(address participant) external view returns (string memory profilConso, uint256 seuil, uint256 chargeEvent) {
        Data memory d = dataStore[participant];
        return (d.profilConso, d.seuil, d.chargeEvent);
    }
}
```

Annexe D – Arguments de simulation

Initialisation du bloc de genèse (1)

geth --datadir poa init genesis.json

Lancement du nœud Geth (2)

geth --datadir poa --networkid 1337 --syncmode full --mine ` --allow-insecure-unlock --unlock "0x550C3335681388D27DAA7812153BA3AD131C806B" ` --password "password1.txt" --http --http.api "web3,eth,net,personal,miner" ` --http.addr "0.0.0.0" --http.port 8545 --authrpc.port 8551 ` --http.corsdomain "*" --http.vhosts "*" ` --miner.etherbase "0x550C3335681388D27DAA7812153BA3AD131C806B"

Lancement de la compilation des contrats intelligents (3)

npx hardhat compile --network poa

Lancement du déploiement sur le réseau (4)

npx hardhat run scripts_deploy_Oracle.js --network poa

npx hardhat run migrations/EffacementP2P.js --network poa

Lancement de la console Hardhat (5)

npx hardhat console --network poa

Commande de lancement du test global Hardhat (6)

npx hardhat test

Annexe E – Fichier test de simulation en Javascript (.js)

```
const { expect } = require("chai");
const { ethers } = require("hardhat");
describe("EffacementP2P", function () {
  let effacementP2P, oracleMock, owner, participant;
  beforeEach(async function () {
    [owner, participant] = await ethers.getSigners();
    //  Déploiement de OracleMock
    const OracleMock = await ethers.getContractFactory("OracleMock");
    oracleMock = await OracleMock.deploy();
    await oracleMock.waitForDeployment();
    console.log("OracleMock déployé à :", await oracleMock.getAddress());
    //  Déploiement de EffacementP2P
    const EffacementP2P = await ethers.getContractFactory("EffacementP2P");
    effacementP2P = await EffacementP2P.deploy(await oracleMock.getAddress());
    await effacementP2P.waitForDeployment();
    console.log("EffacementP2P déployé à :", await effacementP2P.getAddress());
  });
```

```
it("Ajoute un participant avec des données initiales", async function () {
    const tx = await oracleMock.ajouterParticipant(
        participant.address,
        10,  // seuil
        300, // totalReduction
        3,   // nombreEffacements
        "D"  // profilConso
    );
    await tx.wait();
    // Vérifie que les données sont bien enregistrées
    const participantData = await oracleMock.getData(participant.address);
    expect(participantData.seuil).to.equal(10);
    expect(participantData.totalReduction).to.equal(300);
    expect(participantData.nombreEffacements).to.equal(3);
    expect(participantData.profilConso).to.equal("D");
});
it("Simule un effacement et met à jour le nombre d'effacements", async function () {
    //  Ajouter le participant avant de tester l'effacement
    await oracleMock.ajouterParticipant(
```

```
        participant.address,
        10,
        300,
        3,
        "D"
    );
    // Activation de l'effacement
    await expect(effacementP2P.debutEffacement())
        .to.emit(effacementP2P, "EffacementActive");
    // Fin de l'effacement
    await expect(effacementP2P.finEffacement(participant.address))
        .to.emit(effacementP2P, "EffacementDesactive");
    //  Vérification que le nombre d'effacements a bien augmenté
    const participantData = await oracleMock.getData(participant.address);
    expect(participantData.nombreEffacements).to.equal(4); // Incrémenté
});
it("Calcule correctement les crédits après un effacement", async function () {
    //  Ajouter le participant avec `nombreEffacements = 4`
    await oracleMock.ajouterParticipant(
```

```
            participant.address,
            10,
            300,
            4,
            "D"
        );
        //  Vérification du calcul des crédits
        const tx = await effacementP2P.enregistrerCredits(participant.address);
                                        const [effectiveReduce] = await
effacementP2P.calculerCredits(participant.address);
        console.log("effectiveReduce:", effectiveReduce.toString());
        await expect(tx)
          .to.emit(effacementP2P, "CreditsCalcules")
          .withArgs(participant.address, ethers.toBigInt(5911), "D"); // 300/4 = 75 -> crédits
= 5911
      });
    });
```

Annexe F – Commandes du test détaillé en Javascript (.js)

Déployer l'oracle

```
const OracleMock = await ethers.getContractFactory("OracleMock");

const oracle = await OracleMock.deploy();

await oracle.waitForDeployment();
```

Ajouter un participant

```
await effacementP2P.ajoutparticipant(newParticipant.address, "residential");

console.log(`Participant ajouté dans EffacementP2P: ${newParticipant.address}`);
```

Vérifier les données du participant ajouté

```
const isRegistered = await effacementP2P.isRegistered(newParticipant.address);

console.log("Le participant est-il enregistré ?", isRegistered);
```

Démarrer un effacement

```
await effacementP2P.debutEffacement();
```

Fin d'effacement

```
await effacementP2P.finEffacement(participant.address);

console.log("Effacement désactivé avec succès!");
```

Enregistrer et calculer les crédits

```
await effacementP2P.enregistrerCredits(participant.address);
```

```
console.log("Credits enregistrés!");
```

Annexe G – Variante du contrat intelligent du programme utilisé dans la simulation (.sol)

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.13;


// Interface de l'oracle qui fournit les données sur les participants

interface IOracleMock {

    function getData(address participant) external view returns (

        uint256 seuil,

        uint256 totalReduction,

        uint256 nombreEffacements,

        string memory profilConso

    );


    function incrementerEffacements(address participant) external;

}


contract EffacementP2P {

    // Référence à l'oracle
```

```
    IOracleMock private oracle;

    // Adresse du propriétaire du contrat
    address public owner;

    // Indique si un effacement est en cours
    bool public effacementActif;

    // Événement déclenché lorsque les crédits sont calculés
    event CreditsCalcules(
        address indexed participant,
        uint256 credits,
        string profilConso
    );

    // Événements pour activer et désactiver l'effacement
    event EffacementActive();
    event EffacementDesactive(address indexed participant, uint256
nouveauNombreEffacements);
```

```
// Modificateur permettant de restreindre certaines actions au propriétaire du contrat
modifier onlyOwner() {
    require(msg.sender == owner, "Acces refuse");
    _;
}

// Constructeur : initialise l'adresse de l'oracle et définit le propriétaire du contrat
constructor(address _oracleAddress) {
    oracle = IOracleMock(_oracleAddress);
    owner = msg.sender;
}

// Fonction permettant de démarrer un effacement (uniquement pour le propriétaire)
function debutEffacement() external onlyOwner {
    effacementActif = true;
    emit EffacementActive();
}
```

```
    // Fonction permettant d'arrêter un effacement et d'incrémenter le nombre d'effacements du participant
    function finEffacement(address participant) external onlyOwner {
        require(effacementActif, "Aucun effacement en cours");
        effacementActif = false;

        // Incrémentation du nombre d'effacements pour ce participant via l'oracle
        oracle.incrementerEffacements(participant);

        // Récupération du nouveau nombre d'effacements après mise à jour
        (, , uint256 nouveauNombreEffacements, ) = oracle.getData(participant);

        // Émission d'un événement indiquant la fin de l'effacement
        emit EffacementDesactive(participant, nouveauNombreEffacements);
    }

    // Fonction qui calcule les crédits d'un participant en fonction de ses réductions de consommation
    function calculerCredits(address participant) public view returns (uint256 credits, string memory profilConso) {
```

```
        // Récupération des données du participant depuis l'oracle
        (uint256 seuil, uint256 totalReduction, uint256 nombreEffacements, string memory
_profilConso) = oracle.getData(participant);

        // Vérification que le seuil et le nombre d'effacements sont valides
        require(seuil > 0, "Aucune donnee pour ce participant");
        require(nombreEffacements > 0, "Nombre d'effacements invalide");

        // Calcul de la réduction effective
        uint256 effectiveReduce = totalReduction / nombreEffacements;

        // Calcul des crédits en fonction de la réduction effective
        credits = (effectiveReduce >= seuil) ? _calculCredits(effectiveReduce) : 0;

        return (credits, _profilConso);
    }

    // Fonction permettant d'enregistrer les crédits et d'émettre un événement
    function enregistrerCredits(address participant) external {
```

```
        (uint256 credits, string memory profilConso) = calculerCredits(participant);
        emit CreditsCalcules(participant, credits, profilConso);
    }

    // Fonction interne qui applique un barème de calcul des crédits en fonction de la réduction effective
    function _calculCredits(uint256 effectiveReduce) internal pure returns (uint256) {
        uint256 credit = 0;
        uint256 creditPr = 78825;
        uint256 creditPr_Cons = creditPr / 10;
        uint256 creditSc = 273260;
        uint256 creditTr = 75672;

        if (effectiveReduce > 10 && effectiveReduce <= 100) {
            credit = (78825 * effectiveReduce) / 1000;
        } else if (effectiveReduce > 100 && effectiveReduce <= 400) {
            credit = ((68315 * (effectiveReduce - 100)) / 1000) + creditPr_Cons;
        } else if (effectiveReduce > 400 && effectiveReduce <= 1200) {
            credit = ((63060 * (effectiveReduce - 500)) / 1000) + creditSc + creditPr_Cons;
```

```
            } else if (effectiveReduce > 1200) {

                credit = ((57805 * (effectiveReduce - 1700)) / 1000) + creditSc + creditPr_Cons
+ creditTr;

            }



            return credit;

        }

        }

    }
```

Annexe H – Variante du contrat intelligent faisant office d'oracle utilisé dans la simulation (.sol)

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.13;


contract OracleMock {

    struct Participant {

        uint256 seuil;

        uint256 totalReduction;

        uint256 nombreEffacements;

        string profilConso;

    }


    mapping(address => Participant) private participants;

    address public owner;


    modifier onlyOwner() {

        require(msg.sender == owner, "Acces refuse");

        _;
```

```
}

event DonneesMisesAJour(
    address indexed participant,
    uint256 seuil,
    uint256 totalReduction,
    uint256 nombreEffacements,
    string profilConso
);

constructor() {
    owner = msg.sender;
}

function ajouterParticipant(
    address participant,
    uint256 seuil,
    uint256 totalReduction,
    uint256 nombreEffacements,
```

```
        string memory profilConso
    ) external onlyOwner {
        participants[participant] = Participant(seuil, totalReduction, nombreEffacements, profilConso);
        emit DonneesMisesAJour(participant, seuil, totalReduction, nombreEffacements, profilConso);
    }
    function getData(address participant) external view returns (
        uint256 seuil,
        uint256 totalReduction,
        uint256 nombreEffacements,
        string memory profilConso
    ) {
        Participant memory p = participants[participant];
        return (p.seuil, p.totalReduction, p.nombreEffacements, p.profilConso);
    }

    function incrementerEffacements(address participant) external {
        require(participants[participant].seuil > 0, "Participant non existant");
        participants[participant].nombreEffacements += 1;
```

```
        emit DonneesMisesAJour(
            participant,
            participants[participant].seuil,
            participants[participant].totalReduction,
            participants[participant].nombreEffacements,
            participants[participant].profilConso
        );
    }
}
```

Annexe I – Rapport du logiciel anti-plagiat Grammarly