Une architecture de référence basée sur la chaine de blocs pour la numérisation des services publics centrés sur le citoyen

par

Sion Israel Sion

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE DES TECHNOLOGIES DE L'INFORMATION
M. Sc. A.

MONTRÉAL, LE 28 FÉVRIER 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Kaiwen Zhang, directeur de mémoire
Département de génie logiciel et TI, École de technologie supérieure

M. Alain April, codirecteur
Département de génie logiciel et TI, École de technologie supérieure

M. Julien Gascon-Samson, président du jury
Département de génie logiciel et TI, École de technologie supérieure

M. François Coallier, membre du jury
Département de génie logiciel et TI, École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 7 FÉVRIER 2023

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à remercier le directeur de cette recherche, le professeur Kaiwen Zhang, par le fait qu'il m'a volontiers accepté sous sa tutelle, pour que la rédaction de ce travail soit faite. Les orientations qu'il n'a jamais cessé de me donner ont été d'une valeur inestimable.

Je tiens à témoigner ma profonde gratitude au professeur Alain April, codirecteur de cette recherche, pour ses conseils concernant la méthodologie de recherche, la rédaction scientifique et le perfectionnement de ce texte de mémoire.

Mes remerciements vont aussi à l'endroit du Dr. Thomas Maketa pour sa disponibilité et sa collaboration. Son expérience a été d'un profit et d'une importance indéniables.

Mes pensées sont également portées vers ma feu mère, Joelle Lukama Nabintu, pour l'héritage de la foi, ainsi que la fratrie Boanerges pour le soutien infaillible.

Enfin, je remercie les personnes-ressources à Revenu Québec pour la contribution à cette recherche ainsi qu'à tous les miens pour leur foi en moi.

« La partie la plus difficile de la spécification des besoins n'est pas de documenter ce que les utilisateurs veulent ; il s'agit plutôt de l'effort pour les aider à déterminer ce dont ils ont besoin qui peut leur être fourni avec succès. »

Steve McConnell

Une architecture de référence basée sur la chaine de blocs pour la numérisation des services publics centrés sur le citoyen

Sion Israel Sion

RÉSUMÉ

L'avènement de la technologie de la chaine de blocs a inspiré un nouveau type d'infrastructure d'échange d'information sur un modèle pair à pair qui rend possible l'émergence d'un ensemble de cas d'utilisation dans des secteurs où l'amélioration de l'efficacité, de la fiabilité, de la sécurité et des couts autour de la transparence est recherchée. L'adoption de la chaine de blocs offre des opportunités innovantes pour le secteur public et peut améliorer les interactions avec les citoyens et les entreprises. Cette tendance est alimentée par les changements technologiques qui modifient l'environnement quotidien des citoyens et des entreprises privées, et influencent leurs attentes en matière de prestation de services publics. Parmi les développements indiquant des exemples de changement liés à la tendance, il y a la mise en œuvre des nouvelles politiques et de nouveaux services centrés sur le citoyen améliorant l'utilisation des données dans l'intérêt public. Toutefois, l'intégration de la technologie de la chaine de blocs se heurte à de nombreux défis techniques relatifs à la coordination interorganisationnelle, notamment l'interopérabilité, la protection de la vie privée, la gouvernance et les répercussions sur les processus opérationnels. Pour surmonter ce problème, des infrastructures sont nécessaires afin de guider l'intégration de la chaine de blocs dans les services publics.

Dans ce mémoire, les limitations d'intégration de la chaine de blocs dans les services publics sont abordées en proposant une architecture de référence qui identifie les exigences relatives aux défis rencontrés. Dans un effort conjoint avec Revenu Québec, une preuve de concept qui utilise différentes technologies a été développée en combinaison avec un contrat intelligent qui orchestre l'exécution d'un service public et la délégation d'un traitement privé dans un composant indépendant de la chaine de blocs tout en centrant les interactions autour du citoyen. Pour évaluer l'architecture conçue, la méthode ATAM a été utilisée sur la preuve de concept en développant un prototype autour des scénarios du processus d'immatriculation des véhicules entre Revenu Québec et la Société d'Assurance Automobile du Québec.

Mots-clés: Chaîne de blocs, Numérisation, Service public, Intégration, Preuve de concept, Service centré sur le citoyen

A blockchain-based reference architecture for the digitalization of citizen-centric public services

Sion Israel Sion

ABSTRACT

The advent of blockchain technology has inspired a new type of information exchange infrastructure on a peer-to-peer model that makes possible the emergence of a set of use cases in sectors where improved efficiency, reliability, security, and costs around transparency are sought. Blockchain adoption offers innovative opportunities for the public sector and can improve interactions with citizens and businesses. This trend is fueled by technological changes that are altering the daily environment of citizens and private businesses, and influencing their expectations for public service delivery. Developments indicating examples of change related to the trend include the implementation of new citizen-centric policies and services improving the use of data in the public interest. However, the integration of blockchain technology faces many technical challenges related to interorganizational coordination, including interoperability, privacy, governance, and business process impacts. To overcome this problem, infrastructure is needed to guide the integration of blockchain into public services.

In this thesis, the limitations of blockchain integration in public services are addressed by proposing a reference architecture that identifies the requirements for the challenges faced. In a joint effort with Revenue Quebec, a proof of concept that uses different technologies was developed in combination with a smart contract that orchestrates the execution of a public service and the delegation of a private treatment in a blockchain-independent component while centering the interactions around the citizen. To evaluate the designed architecture, the ATAM method was used on the proof of concept by developing a prototype around the scenarios of the vehicle registration process between Revenu Québec and the Société d'Assurance Automobile du Québec.

Keywords: Blockchain, Digitization, Public service, Integration, Proof of concept, Citizen-centric service

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	Application Programming Interface
ATAM	Architecture Tradeoff Analysis Method
BFT	Byzantine Fault Tolerant
BPM	Business Process Management
BPMN	Business Process Model and Notation
CA	Certificate Authority
CFT	Crash Fault Tolerance
CSS	Cascading Style Sheets
DLT	Distributed Ledger Technology
ERC	Ethereum Request for Comment
HLF	Hyperledger Fabric
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IPFS	InterPlanetary File System
PU	Processus Unifié
REST	Representational State Transfer
RQ	Revenu Québec
SDK	Software Development Kit
TVQ	Taxe de vente du Québec
UML	Unified Modeling Language

LISTE DES SYMBOLES ET UNITÉS DE MESURE

CAD	Dollar canadien
GB	Gigabytes
Mo/s	Mégaoctets par seconde
vCPU	Virtual processor

INTRODUCTION

L'amélioration des services du secteur public, par la numérisation, est de plus en plus incontournable. Un organisme public efficace doit, en effet, être en mesure de soutenir les citoyens et les entreprises avec des services de plus en plus performants et accessibles à tous, comblant le fossé entre les citoyens et les institutions, diminuant ainsi les déplacements physiques et le temps de passage dans les bureaux (Battilani *et al.*, 2022). Ces activités d'amélioration visent la conception de services numériques, dédiés aux citoyens et soulevant une série de décisions technologiques fondamentales et essentielles concernant : les infrastructures, l'interopérabilité, les plateformes, les services, la sécurité, l'accessibilité, la transparence, et bien d'autres aspects.

Initialement proposée comme un grand livre public, décentralisé et sans confiance pour les monnaies numériques, la technologie de la chaine de blocs a été largement adoptée dans de nombreux domaines (Allessie, Sobolewski & Vaccari, 2019). Avec la fiabilité, la sécurité des données, l'exactitude et la réduction des coûts, la technologie de la chaine de blocs offre un bon potentiel pour la transformation des organisations et des services publics (Verma & Sheel, 2022).

De manière générale, les grands livres distribués ont le potentiel de devenir une nouvelle source d'infrastructure d'information soutenant l'échange d'informations entre les organismes publics, les citoyens et les entreprises (Allessie *et al.*, 2019). Les deux principaux groupes d'avantages liés à l'utilisation de la technologie de la chaine de blocs sont : une sécurité accrue (c.-à-d. le renforcement de l'intégrité, de l'immuabilité et de la cohérence des données entre les organisations utilisatrices) et des gains d'efficacité (c.-à-d. la réduction des délais de traitement et des coûts associés). Les organismes publics sont en mesure d'utiliser la technologie de la chaine de blocs pour mettre en place des systèmes d'administration en ligne (Verma & Sheel, 2022). Il s'agit notamment d'exemples de son utilisation en gestion des dossiers, de soins de santé, de commerce international et douanes, de vote, de protection de l'environnement, de marchés publics, de sécurité alimentaire, des identités numériques, d'énergie, de protection

sociale, d'engagement communautaire, d'éducation, de comptabilité publique, de système fiscal, de sécurité publique, de loisirs (Cagigas, Clifton, Diaz-Fuentes & Fernández-Gutiérrez, 2021).

Toutefois, son adoption se heurte à de nombreuses contraintes techniques, administratives et légales (Le Vallée, 2018). Le service public est principalement caractérisé par des processus opérationnels qui font intervenir plusieurs institutions publiques pour un objectif donné. Mais la plupart de ces institutions fonctionnent en silos, sans une bonne collaboration avec les institutions impliquées dans les autres activités d'un même processus (Punia & Saxena, 2004). Si les contraintes d'intégration sont résolues, la technologie de la chaine de blocs pourrait jouer un rôle plus central dans la numérisation des services publics, non seulement en termes de rationalisation administrative et de réduction des coûts et du temps qui en résulte, mais aussi parce qu'elle permet d'améliorer considérablement les niveaux de confiance, de transparence, de sécurité, de fiabilité et d'accessibilité des services (Reply, 2022).

Pour surmonter ce problème, des infrastructures sont nécessaires afin de guider l'intégration de la chaine de blocs dans les services publics. Ce mémoire propose une architecture de référence pour aborder les limitations d'intégration de la chaine de blocs en se basant sur une étude de cas du service public.

Ce mémoire est composé de six chapitres : le chapitre 1 présente la problématique, les objectifs et la méthodologie adoptée pour mener à terme la recherche. Le chapitre 2 présente une revue de la littérature qui synthétise le contexte pertinent pour l'objectif de cette recherche. Le chapitre 3 analyse l'étude de cas pour en extraire les exigences du système et les mesures de succès de l'expérimentation et présente le processus ayant guidé le choix de la plateforme utilisée. Le chapitre 4 présente notre contribution principale qui est l'architecture proposée et définit les considérations conceptuelles relatives à notre travail. Le chapitre 5 présente la conception du système et le chapitre 6 présente les résultats de l'implémentation et de l'évaluation de

notre proposition. Finalement, une conclusion et des recommandations pour les travaux futurs clôturent le mémoire.

CHAPITRE 1

PRÉSENTATION DE LA RECHERCHE

1.1 Problématique et motivation de recherche

Les services publics évoluent dans un paysage en mutation rapide qui transforme la prestation de services et l'élaboration des politiques. Afin d'assurer des résultats positifs pour les citoyens, les gouvernements ont besoin du soutien de services publics performants ayant la capacité de suivre les nouvelles tendances et l'évolution des attentes (Mergel, Edelmann & Haug, 2019). Cette évolution des attentes offre aux services publics l'occasion d'envisager une approche différente sur la manière dont ils peuvent adapter leurs activités aux besoins modernes des citoyens. On parle d'une approche centrée sur le citoyen. Il s'agit basiquement d'aligner les services publics sur les besoins et les attentes des citoyens pour améliorer la satisfaction. Les décisions techniques et conceptuelles sont prises en fonction des besoins des citoyens et non en fonction des contraintes des structures publiques (Eggers, 2018).

Pour comprendre les raisons de l'insatisfaction des citoyens à l'égard des services publics, il faut considérer le « parcours du citoyen ». Le parcours du citoyen fait référence à l'expérience complète qu'une personne traverse lorsqu'elle sollicite un service public (Dudley, Lin, Mancini & Ng, 2015). Le service public offert à un citoyen est principalement caractérisé par des processus opérationnels qui font intervenir plusieurs services pour un objectif donné, on parle de processus interorganisationnel (Punia & Saxena, 2004). Le parcours du citoyen implique donc généralement de nombreuses interactions, ou points de contact, avec plusieurs services différents, avant que la tâche ne soit accomplie. Un service public centré sur le citoyen revient à considérer un processus interorganisationnel mettant le citoyen au centre.

De nombreux services publics disposent des portails en ligne pour améliorer la satisfaction, mais ces services ne communiquent généralement pas, chaque service distinct mettant en place son propre système (Punia & Saxena, 2004). Il en résulte que les citoyens doivent saisir les mêmes informations dans le système plusieurs fois au lieu de les saisir une seule fois dans un

formulaire en ligne consolidé. Au lieu d'améliorer chaque point de contact individuellement, il serait plus judicieux d'adopter une approche holistique, en examinant le parcours dans son ensemble et en rationalisant tout le processus.

Une facette importante de l'évolution vers des services centrés sur le citoyen est d'une part, la transparence vis-à-vis des citoyens, et d'autre part, la collaboration entre les différents services participant à un processus interorganisationnel.

L'avènement de la technologie de la chaine de blocs a inspiré un nouveau type d'infrastructure d'échange d'information versé sur un modèle pair-à-pair qui rend possible l'émergence d'un ensemble de cas d'utilisation dans des secteurs où l'amélioration de l'efficacité, de la fiabilité, de la sécurité et des couts autour de la transparence et de la collaboration est recherchée (Baset *et al.*, 2018).

Toutefois, l'intégration de la technologie de la chaine de blocs dans un processus interorganisationnel se heurte à de nombreux défis relatifs à la coordination des organisations, notamment l'interopérabilité, la confidentialité, la flexibilité, la gouvernance, et les répercussions sur les processus opérationnels propres à une organisation (Le Vallée, 2018).

Étant générique, la technologie de la chaine de blocs permet de nombreuses conceptions de systèmes différents. Pour relever le défi de l'intégration de la chaine de blocs dans le service public dans une approche centrée sur le citoyen, cette recherche étudie une approche qui intègre la chaine de blocs aux processus opérationnels interorganisationnels des services publics.

1.2 Question des recherches

Cette recherche propose une architecture de référence basée sur la chaine de blocs pour les services publics centrés sur le citoyen. Afin d'atteindre cet objectif général, cette recherche vise à répondre aux questions suivantes :

1. Quelles sont les exigences d'un système intégrant la chaine de blocs pour le traitement des services publics centrés sur le citoyen ?
2. Quelle est l'architecture de référence d'un système basé sur la chaine de blocs pour un service public centré sur le citoyen ?
3. Comment l'architecture du système peut-elle être mise en œuvre dans un prototype ?

1.3 Méthodologie

Le cadre de Basili décrit par Abran, Laframbroise & Bourque (2003) propose une approche de la planification d'une recherche empirique en génie logiciel. Il y décrit quatre phases de recherche : la définition, la planification, le développement et l'interprétation des résultats. Cette méthodologie guide l'approche de planification de cette recherche dans la recherche de résolution des questions de recherche et est détaillée dans les sous-sections qui suivent.

1.3.1 Définition de la recherche

Il est question ici de définir de manière précise ce qui s'inscrit dans le cadre de cette recherche et ce qui ne l'est pas. On y présente ici la portée de la recherche. Cette première phase de la planification de la recherche, présentée sur le tableau 1.1, comprend la définition de la motivation de la recherche, ses objectifs précis, une proposition d'innovation précise, ainsi que les utilisateurs potentiels des résultats de cette recherche.

1.3.2 Planification de la recherche

Il est question ici d'identifier les activités de recherche et les résultats attendus pour atteindre les objectifs fixés et répondre aux questions de recherche. Le tableau 1.2 présente les activités liées à cette phase.

Tableau 1.1 Phase de définition de la recherche

Motivation	**Objectif**	**Proposition/but**	**Utilisateurs**
Intégrer la chaine de blocs dans le service public dans une approche centrée sur le citoyen.	• Proposer une architecture assurant la coordination des organisations dans une approche centrée sur le citoyen ; • Évaluer le degré auquel la proposition satisfait les objectifs de conception ; • Développer un prototype fonctionnel qui implémente l'architecture proposée.	Concevoir une architecture de référence basée sur la chaine de blocs pour les services publics centrés sur le citoyen.	Les étudiants, les chercheurs, les architectes de solution et les responsables des services de l'innovation et des technologies de l'information dans les services publics.

Tableau 1.2 Phase de planification

Étapes	**Entrées**	**Résultats**
Revue de littérature	Revue de littérature des éléments suivants : • Numérisation des services publics ; • Processus interorganisationnel ; • Chaine de blocs ; • Méthode de conception et d'évaluation d'une architecture.	• Identification des défis relatifs à la numérisation des processus interorganisationnels ; • Les considérations de conception clé d'un système basé sur la chaine de blocs ; • Section de la revue de la littérature de ce mémoire.
Activités de recherche	1. Résultats de la revue de littérature ; 2. Étude de cas proposé par Revenu Québec ; 3. Choix de la plateforme de la chaine de blocs.	• Définition des objectifs de conception ; • Définition d'une approche et d'une architecture de référence d'un système basé sur la chaine de blocs pour le service public centré sur le citoyen ; • Application de l'architecture à travers une preuve de concept ; • Publication d'articles de conférence.
Vérification et évaluation des résultats	Choix d'une stratégie d'évaluation de l'architecture et d'expérimentation	Résultats finaux • Évaluation de l'architecture basée sur les scénarios ; • Dépôt du document de mémoire.

1.3.3 Développement de la recherche

À l'aide des constats de la revue littéraire, la phase de développement, présentée sur le tableau 1.3, comprend les étapes de proposition d'une architecture de référence ainsi que la définition d'un cadre d'évaluation de la contribution et de l'expérimentation.

Tableau 1.3 Phase de développement de la recherche

Développement	**Validation**	**Analyse**
Proposer une architecture de référence d'un système basé sur la chaine de blocs pour le service public centré sur le citoyen.	• Justifier les composants de l'interface pour chaque élément de l'architecture proposée ; • Identifier une étude de cas et réaliser une expérience pour valider la proposition ; • Publier l'orientation proposée dans une revue scientifique et discuter avec les pairs.	Recueillir les commentaires des lecteurs afin de juger de la validité de l'approche et du modèle.
Définir un cadre d'évaluation de l'architecture et de l'expérimentation	• Organiser des séances d'analyse avec les organisations participantes ; • Exécuter un scénario basé sur l'étude de cas.	• Clarifier les exigences en dressant un arbre d'utilité en fonction des questions posées par les organisations participantes ; • Analyser les résultats de l'expérience afin d'obtenir des conclusions et d'en tirer des améliorations.

1.3.4 Interprétation et extrapolation

Il est question ici d'évaluer la solution proposée, d'analyser le résultat de l'expérimentation et comment elle peut être utilisée plus largement. Des travaux futurs sont aussi discutés. Le tableau 1.4 présente les activités de cette phase.

Tableau 1.4 Phase d'interprétation

Interprétation	**Extrapolation**	**Travaux futurs**
Interpréter les résultats obtenus lors de l'évaluation de l'architecture basée sur les scénarios afin de valider la contribution originale de la recherche.	Généraliser les enseignements pour une utilisation dans un domaine plus large des services publics.	• Limitations constatées dans les objectifs de conception non résolus ; • Avenues futures pour la faisabilité d'une adaptation du modèle à un domaine plus large des services publics.

1.3.5 Contribution

À ce jour, il n'y a pas beaucoup d'études publiées qui décrivent le développement d'applications orientées chaine de blocs dans des contextes organisationnels (Faruk *et al.*, 2022). Bien que les preuves de concept soient des démonstrations importantes de faisabilité, la principale contribution de cette recherche réside dans les leçons tirées de leur mise en œuvre, à savoir le choix de la plateforme de chaine de blocs et la définition d'une architecture pour les services publics centrés sur le citoyen. Plus précisément, les contributions de ce mémoire s'alignent sur les objectifs définis et s'énoncent de la manière suivante :

1. Une architecture à trois couches intégrant la chaine de blocs et les composants autour d'une couche réseau, d'une couche interaction et d'une couche application ; suivant une approche

conceptuelle rendant possible l'application de la chaine de blocs dans le service public centré sur le citoyen ;

2. Une évaluation de l'architecture proposée en utilisant la méthode d'évaluation d'architecture basée sur les scénarios ATAM ;
3. Une implémentation basée sur l'architecture de référence proposée d'une étude de cas réel retraçant le processus d'immatriculation de véhicule à cheval entre Revenu Québec et la Société de l'Assurance Automobile du Québec.

L'architecture proposée a fait l'objet d'un article intitulé « Towards Citizen-Centric Services using Blockchain-Powered Digitalization of Public Sector Processes » qui a été soumis à la conférence « IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2023, Dubai, United Arab Emirates) ».

CHAPITRE 2

REVUE DE LITTÉRATURE

Dans ce chapitre, le contexte qui est pertinent pour cette recherche est passé en revue. En premier lieu, ce chapitre présente une analyse de la littérature sur les services publics dans son aspect numérisation et collaboration dans le processus interorganisationnel. En deuxième lieu, ce chapitre présente les notions de base encadrant la technologie de la chaine de blocs. La chaine de blocs est présentée de manière générale suivi d'une présentation de plateformes de registres distribués populaires. En troisième lieu, un état de lieu sur les méthodes de conception et d'évaluation des solutions basées sur la chaine de blocs est présenté. En dernier lieu, des travaux connexes portant sur l'utilisation de la chaine de blocs dans la numérisation des services publics, dans le processus interorganisationnel et dans l'adoption d'une approche centrée sur le citoyen sont présentés.

2.1 Services publics

2.1.1 Numérisation des services publics

Le service public désigne une activité d'intérêt général relevant du secteur public. Le secteur public est utilisé par opposition au secteur privé pour désigner l'ensemble des activités détenu par l'État ayant pour rôle de fournir un service aux citoyens. Il s'agit d'un terme générique qui comprend les administrations publiques, les institutions publiques et les entreprises publiques.

L'objectif des programmes et des politiques du secteur public est de créer de la valeur publique (Davoudi & Johnson, 2022). La simplification de ce secteur par l'utilisation de la technologie est un sujet qui s'inscrit dans l'ordre des tendances actuelles des organismes publics dans la poursuite de l'optimisation de la valeur publique des services fournis (Dobrolyubova, 2021). La valeur publique peut être définie au sens de Scupola & Mergel (2022) comme un ensemble comprenant la valeur économique exprimée par l'économie de coûts, la valeur administrative

exprimée par l'amélioration de l'efficacité administrative, la valeur citoyenne exprimée par la transparence et la vie privée et la valeur sociétale entendue au sens large comme la contribution du secteur public à la société et exprimée en termes d'état de droit.

Le secteur public évolue dans un environnement de plus en plus complexe qui exige une plus grande capacité d'adaptation, d'innovation et de collaboration (Forum, 2016). L'adoption des nouvelles technologies offre des possibilités innovantes au secteur public et peut améliorer les interactions avec les citoyens et les entreprises. Cette tendance est impulsée par les changements technologiques qui modifient l'environnement quotidien des citoyens et des entreprises privées, et qui influent sur les attentes de ce dernier dans la prestation des services publics (Mergel *et al.*, 2019; Scupola & Mergel, 2022).

Cette reconstruction des modèles existants en fonction des besoins des clients en utilisant les nouvelles technologies est désignée par le terme « transformation numérique ». S'il est vrai que ce terme englobe généralement, dans la littérature, les notions de numérisation et de digitalisation, il est important de différencier le premier qui caractérise la transition des processus analogiques vers les processus numériques, du second qui se concentre sur les changements potentiels dans les processus au-delà de la simple numérisation des processus et formulaires existants (Dobrolyubova, 2021).

Cette transformation numérique en utilisant les nouvelles technologies se concentre en particulier sur l'amélioration des processus, des relations et des services (Mergel *et al.*, 2019). Toutefois, ce mélange de refonte, de normalisation des processus opérationnels et d'interopérabilité avec les autres services est un objectif délicat (Battilani *et al.*, 2022). Les processus, les personnes, les politiques et la gouvernance doivent, en effet, être fondamentalement modifiés pour accomplir la transformation numérique dans le secteur public (Armenia, Casalino, Gnan & Flamini, 2021; Mergel *et al.*, 2019).

Parmi les développements indiquant des exemples de changement liés à la tendance, il y a le principe *« Once Only »* pour la fourniture des services aux citoyens où ces derniers ne doivent fournir leurs données qu'une seule fois, avant qu'elles ne soient partagées entre les services avec

les protections appropriées en matière de vie privée (Commission, 2017), la mise en œuvre de nouvelles politiques et de nouveaux services centrés sur le citoyen améliorant l'utilisation des données dans l'intérêt public (Sullivan, Bellman, Sawchuk & Mariani, 2021), une gouvernance des données plus démocratique au service de l'intérêt public avec un intérêt croissant pour les approches alternatives de gestion, de contrôle et d'utilisation des données permettant aux acteurs de contrôler les données personnelles et de décider de leur utilisation (Carballa, 2019; Micheli, Ponti, Craglia & Berti Suman, 2020), ou encore la chaine de blocs pour établir la confiance et accroître la transparence comme une nouvelle infrastructure d'information qui pourrait soutenir et garantir l'échange sécurisé d'informations entre les administrations publiques, les citoyens et les entreprises (De Filippi, Mannan & Reijers, 2020; IBM, 2017).

Un exemple remarquable de numérisation des services publics est représenté par l'Estonie, où pratiquement toutes les transactions gouvernementales peuvent être effectuées de manière numérique. Selon Martinson (2019), 99% des services publics sont accessibles aux citoyens sous forme de services en ligne, et chaque citoyen dispose d'une identité numérique délivrée par l'État, ce qui leur permet d'accéder à tous les services électroniques. La numérisation de l'Estonie utilise les dernières technologies, notamment la technologie de la chaîne de blocs, qui est utilisée pour de nombreux registres, tels que les registres de santé, de propriété, d'entreprises, de successions et le système judiciaire numérique. La mise en place d'une stratégie de développement numérique ambitieuse visant à moderniser l'ensemble des services gouvernementaux a rendu possible cette numérisation des services publics.

En conclusion, la réalité de l'administration publique étant encore très variée : des centres de décision qui se chevauchent, des règlementations adaptées au fonctionnement en silo ou encore le manque de véritable modèle de prise de décision basé sur les données (Battilani *et al.*, 2022), la transformation numérique ne consiste pas simplement à remplacer l'existant par une version électronique, mais à concevoir et à gérer tous les processus organisationnels de manière intégrée et collaborative, en modifiant les modèles, les processus opérationnels et les expériences des utilisateurs (Armenia *et al.*, 2021).

2.1.2 Processus interorganisationnel

Un processus interorganisationnel désigne un processus dans lequel plusieurs organisations travaillent conjointement à la fourniture d'un service (Putnik & Cruz-Cunha, 2008). C'est un modèle d'organisation plébiscité par les organisations privées ou commerciales, qui, en fonction du besoin, forment une entreprise virtuelle avec d'autres pour atteindre divers objectifs commerciaux (Fridgen, Radszuwill, Urbach & Utz, 2018). Avec l'intégration de tels processus, les organisations s'attendent à de nombreux bénéfices : réduction de coût, transparence, facilité d'accès, rapidité de traitement (Aubert, Babin & Mignerat, 2001). Les organisations du secteur public recourent fréquemment à la collaboration pour effectuer leurs opérations. Elles ont besoin pour cela d'une coordination interorganisationnelle pour garantir la réalisation de résultats communs (Lie, 2011). Impliquant plusieurs niveaux allant de l'élaboration des politiques à la prestation de services, la coordination est une question omniprésente et complexe de l'administration publique (Davoudi & Johnson, 2022; Vignieri & Zeinali, 2022).

Selon le rapport de Reply (2022), plusieurs institutions ont lancé divers essais pour remédier aux inefficacités des processus administratifs qui ont un faible niveau de numérisation et de collaboration, et ainsi corriger la tendance à la formation des propres « silos » de protocoles de données et de procédure de gestion de l'information qui les caractérisent (Rot, Sobińska, Hernes & Franczyk, 2020). Pour assurer cette collaboration, les organisations concernées doivent s'accorder sur un processus interorganisationnel commun réalisé par un ensemble d'activités interdépendantes, qui réalisent collectivement un objectif métier commun (Rondanini, Carminati, Daidone & Ferrari, 2020). L'orchestration et l'automatisation en tout ou en partie de ce processus au cours duquel des documents, des informations ou des tâches sont transmis d'un participant à un autre pour action, selon un ensemble de règles procédurales est appelé flux de travaux ou plus communément workflow, de son terme en anglais (Punia & Saxena, 2004; Rondanini *et al.*, 2020). L'intégration de divers participants et l'interopérabilité avec les processus privés propres à chaque organisation rendent les processus interorganisationnels et les flux de travaux qui en résultent très complexes (Fridgen *et al.*, 2018).

Pour exécuter le processus interorganisationnel, le flux de travaux interorganisationnel offre aux organisations privées la possibilité de remodeler les processus opérationnels au-delà des limites de leur propre organisation. Les organisations créent, par exemple, des modèles de processus interorganisationnel représentés numériquement sous BPMN ou utilisent des moteurs de workflow BPM pour réaliser une orchestration complète de tous les flux de travaux (Morales-Sandoval, Molina, Marin-Castro & Gonzalez-Compean, 2021).

Toutefois, si un processus a une portée interorganisationnelle, un défi majeur pour l'efficience et l'efficacité d'un tel processus est de gérer la coordination entre les activités réalisées par différentes organisations sans que les participants perdent leur autonomie (Punia & Saxena, 2004). L'une des solutions pour résoudre le défi de coordination tout en sauvegardant la flexibilité et le respect de la confidentialité pour les organisations participantes est d'utiliser une approche publique-privée (Evermann & Kim, 2019; Punia & Saxena, 2004). L'approche publique-privé, inspirée des principes de l'architecture orientée services, considère la définition d'un flux de travaux public comme un contrat entre les acteurs. Les acteurs peuvent fournir des implémentations privées pour leurs parties d'un processus, tant que celles-ci sont compatibles avec le contrat public (Evermann & Kim, 2019). Les processus privés sont internes et uniques à chaque organisation et sont gérés de manière indépendante, tandis que les processus publics sont communs et standardisés pour être utilisés par tous (Punia & Saxena, 2004). Toutefois, la conception suivant cette approche peut impliquer la connaissance et le partage des connaissances organisationnelles sur les processus privés de chaque organisation ainsi que la réingénierie des processus d'entreprise pour modifier et intégrer ces processus et doit être conforme aux principes des organisations autonomes et homologues au sens de Ludwig cités dans Punia & Saxena (2004) : orientation vers un objectif, confidentialité, flexibilité et indépendance.

2.2 Registre distribué

2.2.1 Chaine de blocs

La chaine de blocs est, de manière simple, une technologie de registre distribué apparue avec la cryptomonnaie Bitcoin. Un registre distribué est un type de base de données réparti sur plusieurs sites, régions ou participants. Les registres distribués ont attiré davantage l'attention avec l'introduction de la chaine de blocs et des transactions financières (El Ioini & Pahl). De manière détaillée, une chaine de blocs est un registre immuable, partagé sur un réseau distribué entre plusieurs nœuds. Chaque nœud possède une copie du registre. Pour assurer la confiance entre les nœuds, un mécanisme de consensus est mis en place pour valider les transactions et s'assurer que le consensus est intégral. Apparu avec Bitcoin, aujourd'hui on compte les chaines de blocs publiques, dans lesquelles la participation est ouverte à toutes personnes et où le consensus est généralement assuré par une preuve de travail, et les chaines de blocs privées dans lesquelles la participation est restreinte et qui répondent à la préoccupation mettant en œuvre un nombre réduit d'entités ayant un objectif commun, mais ne se faisant pas mutuellement confiance (Androulaki *et al.*, 2018).

La chaine de blocs promet de résoudre fondamentalement les problèmes de temps et de confiance afin d'améliorer l'efficacité et les coûts dans plusieurs secteurs. Les principales caractéristiques de la chaine de blocs sont l'immuabilité et la décentralisation. La décentralisation se résume dans un grand livre partagé où les mises à jour transactionnelles sont effectuées par un système de confiance basé sur le consensus, rendant possible une interaction véritablement numérique entre plusieurs parties (Baset *et al.*, 2018). L'immutabilité signifie qu'une fois que les données regroupées dans un bloc, que ce dernier est enchainé aux blocs existants et qu'un nombre suffisant de participants se sont mis d'accord sur cet état, les informations sont stockées de manière permanente et immuable. Pour modifier les informations contenues dans un bloc particulier, il faudrait parvenir à modifier tous les blocs précédents jusqu'au dernier étant donné que chaque nouveau bloc créé contient une référence du bloc précédent.

Une caractéristique importante de la chaine de blocs est la possibilité de construire des applications décentralisées implémentant une logique programmable dans lequel il est possible de définir des règles arbitraires personnalisées pour la propriété, les formats de transaction et les fonctions de transition d'état (Buterin, 2014). Il est ainsi possible de représenter la logique qui régit les transactions sur le réseau et réguler les accords entre les organisations. Ceux-ci sont stockés et exécutés sur la chaine de blocs. En tant que telle, la chaine de blocs peut être considérée comme un registre distribué stockant les résultats (c'est-à-dire les transactions) des contrats intelligents dont l'évaluation correcte a été validée par les participants au réseau (Rondanini *et al.*, 2020).

L'autre élément à considérer est qu'on peut opter pour une chaine de blocs publique (« entièrement décentralisée »), une chaine de blocs de consortium («partiellement décentralisée») ou une chaine de blocs privée (« centralisée »), mais en vrai, d'un point de vue de l'utilisateur, il y a deux éléments à considérer pour catégoriser le type d'une chaine de blocs : la permission d'utiliser le réseau et la permission de participer au processus de consensus. Le premier élément renvoie au critère Public ou Privé d'une chaine de blocs et le second renvoi au critère Permissionné ou non-permissionné (Allessie *et al.*, 2019).

Une chaine de blocs typique se compose généralement de plusieurs couches (Wang, Wang, Cao, Li & Xiong, 2019), mais d'un point de vue plus général, elle peut être décomposée en trois couches différentes comme le présente la figure 2.1

La couche 1 présente le mécanisme de consensus. Les mécanismes de consensus d'une structure de la chaine de blocs peuvent varier considérablement et dépendent de son caractère public ou privé (Valenta & Sandner, 2017). Mais ils sont généralement basés sur une structure incitative qui récompense les participants qui contribuent au réseau. Les mécanismes de consensus peuvent être classés en deux catégories : le consensus basé sur la preuve et le consensus basé sur le vote. Les mécanismes de consensus basés sur les preuves exigent que les nœuds rejoignant le réseau de la chaine de blocs résolvent un puzzle mathématique pour montrer qu'ils sont plus éligibles que les autres pour effectuer le travail d'ajout ou de minage (Pahlajani, Kshirsagar & Pachghare,

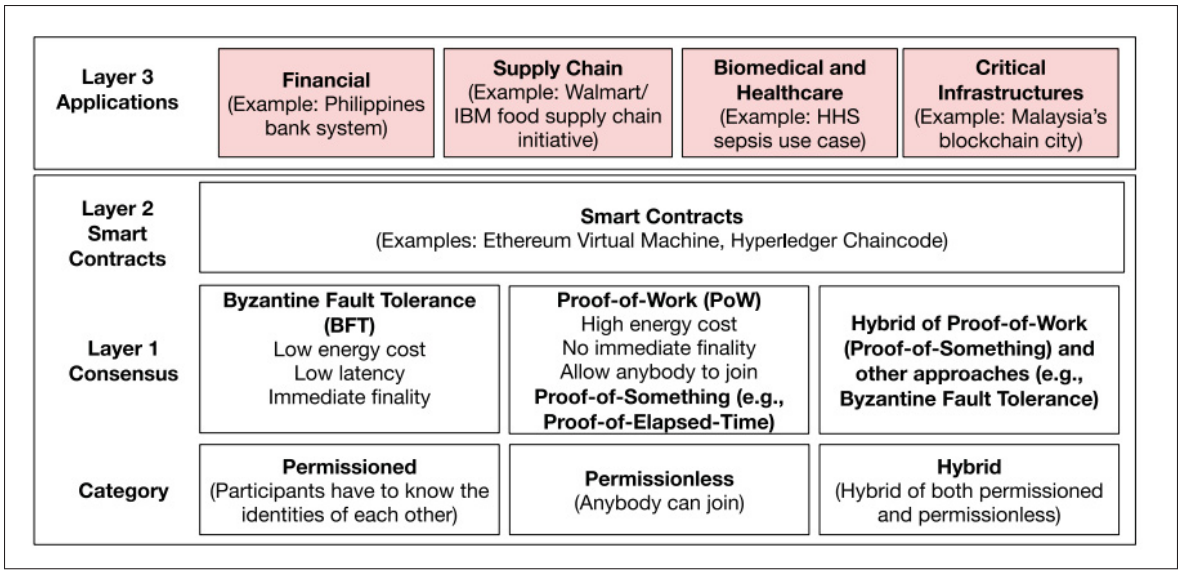

Figure 2.1 Architecture en couche d'une blockchain
Tirée de Clavin et al.(2020, p.5)

2019). Les mécanismes de consensus par vote exigent que les nœuds du réseau de la chaîne de blocs diffusent les résultats de l'extraction d'un nouveau bloc ou d'une transaction, avant d'ajouter le bloc à la chaîne de blocs, elles sont préférables dans les chaines de blocs privées, où les nœuds sont connus (Pahlajani *et al.*, 2019).

La couche contrat (couche 2) fournit une interface permettant aux développeurs de mettre en œuvre de nouvelles fonctions (Clavin *et al.*, 2020). Elle encapsule divers types de codes de script, d'algorithmes et de contrats intelligents sophistiqués, et constitue ainsi la base d'une programmation et d'une manipulation flexibles des systèmes de la chaine de blocs (Wang *et al.*, 2019). Cette couche relie les protocoles de consensus sous-jacent de la couche 2 aux applications et cas d'utilisation de la couche 3.

Les trois éléments de base qui composent la technologie de la chaine de blocs, à savoir le réseau distribué d'ordinateurs qui conserve une base de données chronologique de toutes les transactions (le grand livre), l'utilisation de clés cryptographiques et le mécanisme de consensus peuvent être conçus de différentes manières selon le type de chaine de blocs nécessaire à un objectif particulier (Eenmaa-Dimitrieva & Schmidt-Kessen, 2019; Valenta & Sandner, 2017).

Toutefois, il demeure essentiel de choisir le bon registre distribué (DLT) pour une application particulière, car chaque DLT s'accompagne de son propre ensemble de caractéristiques et de limitations. L'arbre DLT de Kaiwen peut être utilisé à cet effet (Maketa, 2020).

La plupart des solutions basées sur la chaine de blocs sont construites sur les plateformes Ethereum et Hyperledger Fabric (Franciscon *et al.*, 2019).

2.2.2 Ethereum

Ethereum peut être défini comme une plateforme de contrats intelligents et d'applications décentralisées. C'est une technologie publique et sans permission, distribuée, décentralisée et construite pour créer un protocole alternatif pour la construction d'applications décentralisées implémentant une chaine de blocs avec un langage de programmation Turing complet intègre, permettant à quiconque d'écrire des contrats intelligents et des applications décentralisées où il est possible de définir des règles arbitraires personnalisées pour la propriété, les formats de transaction et les fonctions de transition d'état (Buterin, 2014). Le code des contrats intelligents est déterministe et immuable une fois déployé. Un contrat intelligent est un programme informatique traduisant un contrat écrit en code qui se déclenche automatiquement lorsque certaines conditions sont remplies. Il peut détenir et transférer des actifs numériques entre les parties au contrat et invoquer d'autres contrats intelligents stockés sur la chaine de blocs (Xu, Weber & Staples, 2019). Le code écrit dans un langage de haut niveau, est compilé dans un langage *« bytecode »* de bas niveau qui est exécuté par la machine virtuelle Ethereum (EVM) incluse dans chaque nœud du réseau chaine de blocs Ethereum (Xu *et al.*, 2019). Le code est déterministe afin de garantir la cohérence et immuable c'est-à-dire qu'une fois déployé, le code ne peut plus être modifié. Ethereum est une chaine de blocs publique et sans permission qui distribue un registre entre tous les nœuds du réseau. Afin de garantir le consensus entre ces derniers, un mécanisme de preuve de travail *(proof-of-work)* fonctionne sur l'ensemble de la chaine de blocs. Il permet aux nœuds participants du réseau décentralisé de parvenir à un consensus ou de se mettre d'accord sur des éléments tels que les soldes de comptes et l'ordre des transactions, ce qui empêche les utilisateurs

d'effectuer de fausses transactions et de dépenser deux fois leurs pièces (Xu *et al.*, 2019). Il est aussi possible de construire des chaines de blocs privées basées sur la technologie Ethereum en utilisant Enterprise Ethereum Alliance et Hyperledger Besu et Burrow qui sont basés sur la couche Ethereum ou encore Quorum qui est doté de fonctionnalités avancées de niveau « entreprise » permettant le respect de la vie privée, l'octroi d'autorisations et la performance (Ethereum, 2022).

2.2.3 Hyperledger Fabric

Hyperledger Fabric est une plateforme libre conçue pour construire des solutions de grands livres distribués, avec une architecture modulaire qui offre des degrés élevés de confidentialité, de flexibilité, de résilience et d'évolutivité (Group, 2018). Hyperledger Fabric maintient un contrôle strict sur la participation au réseau. Il fait fonctionner la chaine de blocs parmi un ensemble de participants connus et identifiés et fournit un moyen de sécuriser les interactions entre un groupe d'entités qui ont un objectif commun, mais qui ne se font pas entièrement confiance, comme les organisations qui échangent des fonds, des biens ou des informations (Androulaki *et al.*, 2018).

Seuls les membres autorisés et les nœuds sélectionnés par les membres autorisés peuvent utiliser la plateforme Hyperledger Fabric et ses outils grâce aux services d'adhésion ou services aux membres. Cela permet de cacher des informations précieuses et confidentielles aux parties externes et les empêche de les manipuler. Le contrôle d'accès aux données est appliqué au niveau du réseau et des canaux. Le concept de canal permet de répondre à des scénarios où la vie privée et la confidentialité sont importantes et où une transparence réduite est acceptable. Un canal permet à un groupe de participants de créer un registre distinct des transactions, partagé uniquement avec un ensemble de membres de ce canal. Il peut couvrir l'ensemble du réseau de la chaine de blocs, comme dans le cas d'un système de chaine de blocs public, ou ne comprendre que quelques participants de l'ensemble du réseau. Les canaux sont importants pour les systèmes où les participants peuvent être des concurrents et ne veulent pas divulguer leurs informations (Xu *et al.*, 2019).

Hyperledger Fabric permet également aux organisations membres d'exécuter un certain code sur les nœuds qui créent les transactions sur une condition spécifique. Ces codes sont connus sous le nom de *« chaincode »*.

Dans une chaine de blocs classique comme Ethereum, chaque nœud de la chaine de blocs remplit trois fonctions : conserver une copie redondante du grand livre, exécuter tout code du contrat intelligent demandé, maintenir toutes les copies du grand livre en synchronisation (mêmes données, même ordre). Dans Hyperledger Fabric, chaque fonction est assurée par son propre type de nœud. Les nœuds de commande (de l'anglais *« Orderer node »*) conservent toutes les copies du grand livre en synchronisation (mêmes données, dans le même ordre), les nœuds pairs (de l'anglais *« peer node »*) sont divisés en deux sous-groupes à savoir les nœuds d'approbation (de l'anglais *« endorsing peer »*) qui exécute tout code de contrat intelligent demandé et le nœud de validation (de l'anglais *« committing peer »*) qui conserve une copie redondante du grand livre.

Il n'est pas aisé de déterminer quels attributs de la chaine de blocs HLF sont importants pour un cas d'utilisation donné. Le tableau 2.1, inspiré des recherches de Scriber (2018) et de Alketbi, Nasir & Abu Talib (2020) donne un aperçu des principes directeurs à considérer pour HLF.

2.3 Conception et évaluation d'une solution basée sur la chaine de blocs

La variété des plateformes de chaine de blocs existantes pouvant être utilisée en entreprise et la variété des modèles complexifient le processus d'analyse, de modélisation et de développement d'un projet sur la chaine de blocs dans les organisations (Nanayakkara, Rodrigo, Perera, Weerasuriya & Hijazi, 2021).

Les méthodes d'analyse et de conception sont surtout développées pour le développement et l'intégration des applications classiques en ingénierie logicielle. Pour maitriser la cohérence et garantir la pertinence d'une technologie dans la mise en place d'un système, l'utilisation d'une méthodologie de conception est importante. Elle permet de formaliser les étapes du

Tableau 2.1 Principes directeurs de HLF

Caractéristiques	**Considérations**
Immutabilité	Une chaine de blocs est intrinsèquement permanente. Elle ne rend pas possible l'exécution des commandes avec mise à jour ou suppression sémantique.
Écosystème	Une architecture basée sur la chaine de blocs est indiquée pour une intégration dans un écosystème de participants diversifiés.
Distribution	La distribution des nœuds et la validation des transactions sont essentielles à une chaine de blocs, car elles garantissent la confiance distribuée dans la chaine.
Transparence	La transparence entre les acteurs est une exigence dans l'architecture du réseau de la chaine de blocs.
Registre immuable et partagé	Le registre de HLF est conçu pour être immuable et partagé pour d'encoder l'historique complet des transactions pour chaque canal avec une capacité d'interrogation.
Chaincode (Contrat intelligent)	L'évolutivité et les performances du réseau sont optimisées, car HLF sépare l'exécution des contrats intelligents de l'ordonnancement des transactions.
Services aux membres	HLF étant un réseau permissionné, il dispose d'un mécanisme qui permet aux participants de prouver leur identité au reste du réseau afin d'effectuer des transactions sur ce dernier.
Confidentialité avec les canaux	HLF intègre la notion de Canal qui offre un haut degré de confidentialité en permettant des transactions multilatérales. Les canaux offrent un moyen de créer plusieurs registres, chacun d'entre eux pouvant avoir un ensemble unique de participants et de permissions.
Actifs	Les définitions d'actifs dans HLF permettent l'échange d'actifs tangibles ou immatériels représentés comme une collection de paires clé-valeur dont les changements d'état sont enregistrés sous forme de transactions sur un grand livre de dans les canaux.
Collection de données privées	HLF permet de garder les données privées pour un groupe d'organisations et de pairs et non accessibles pour les autres organisations dans un canal. C'est une alternative possible au stockage décentralisé qui peut être effectué hors chaine en utilisant un système distribué pour le stockage et l'accès aux fichiers comme IPFS.

développement d'un système en définissant et en décrivant une démarche de mise en œuvre accompagnant les phases d'analyse, de conception et de développement d'une application tout

en facilitant le suivi de toutes les parties impliquées dans le but de s'assurer de sa conformité aux besoins identifiés.

Les méthodes de conception couramment utilisées en ingénierie logicielle telle que MERISE, PU (utilisant la notation UML) ou encore le cadre i*, rendent possible l'analyse et la conception de la solution à un problème dans un format standard, de sorte que tous les paramètres puissent être discutés avec les parties prenantes, indépendamment de la mise en œuvre, à l'aide de représentations visuelles des composants de la conception.

Il n'existe à ce jour aucune technique de modélisation standard, disciplinée et organisée, pour la modélisation organisationnelle pour les applications basées sur la chaine de blocs (Faruk *et al.*, 2022; Vingerhouts, Heng & Wautelet, 2020). De nombreux travaux existants (Górski, 2021; Vingerhouts *et al.*, 2020; Wessling, Ehmke, Hesenius & Gruhn, 2018) utilisent des méthodologies adaptées qui leur sont propres, chacune d'entre elles ayant ses spécificités dans son contexte, montrant ainsi l'absence d'unanimité autour de la question de la méthodologie de conception.

L'un des modèles de conception qui a fait ses preuves dans l'industrie est le modèle de vue « 4 + 1 » de Kruchten (1995) qui est utilisé pour décrire l'architecture (la conception) de systèmes en utilisant plusieurs vues simultanées et les perspectives des différentes parties prenantes : Les utilisateurs finaux, les développeurs, les ingénieurs système et les chefs de projet. Le modèle comporte ainsi quatre points de vue adaptés aux perspectives des parties prenantes : la vue logique, la vue développement, la vue processus et la vue physique. De plus, le modèle prend en compte des cas d'utilisation ou des scénarios comme vue supplémentaire « plus un » pour illustrer la conception.

Toutefois, afin de valider la proposition d'une architecture, l'évaluation de l'architecture est nécessaire pour déterminer si l'architecture proposée répond aux objectifs de conception définis. Il existe plusieurs méthodes pour évaluer une architecture telle que présentée par Arshad, Townend & xu (2011). Ces méthodes diffèrent les unes des autres en fonction du stade de développement du logiciel et du niveau d'information disponible sur les différents

composants de l'architecture. Les modèles mathématiques nécessitent des données spécifiques à l'implémentation et utilisent des systèmes mathématiques pour évaluer une architecture. Ces méthodes conviennent mieux aux systèmes logiciels dont les données d'implémentation sont connues. Les modèles basés sur les scénarios utilisent des scénarios pour l'évaluation architecturale. Un scénario est une brève description d'une utilisation anticipée ou souhaitée d'un système. Ces modèles ne nécessitent pas de données spécifiques à la mise en œuvre. Une analyse comparative des méthodes d'évaluation de l'architecture basée sur des scénarios a révélé que l'ATAM est la seule méthode d'évaluation de l'architecture basée sur des scénarios qui fournit un soutien complet au processus d'évaluation de l'architecture (Babar & Gorton, 2004). ATAM est utilisé pour évaluer les architectures logicielles à l'aide de scénarios. L'objectif d'ATAM est d'évaluer le degré auquel un système est susceptible de pouvoir satisfaire ses objectifs de conception attendus en l'analysant d'un point de vue architectural.

2.4 Travaux connexes

2.4.1 Numérisation des services publics basée sur la chaine de blocs

Avec la fiabilité, la sécurité des données, l'exactitude et la réduction des coûts offerts par la chaine de blocs, plusieurs organisations expérimentent de multiples cas d'utilisation de la chaine de blocs allant des gouvernements, aux institutions financières en passant par des entreprises. Ses caractéristiques de technologie distribuée, traçable et sécurisée, sont applicables à la construction de nouveaux modèles de ressources d'informations des services publics basés sur le numérique (Zhang, 2019).

Les domaines d'application de la technologie chaine de blocs dans les services publics portent sur la monnaie numérique, le vote électronique, l'économie partagée, le transfert de titres de propriété, le système de gestion de l'identité, l'interaction directe entre le gouvernement et les citoyens, l'établissement de valeurs publiques, la protection de la vie privée, le dossier de santé publique ou encore le partage des données (Verma & Sheel, 2022). Selon Rot *et al.* (2020), le

nombre de mises en œuvre réussie, de projets en cours, de test et d'études analytiques font des services publics un segment particulièrement important et actif dans l'évolution et la propagation des solutions de la chaine de blocs.

La chaine de blocs fournit une solution de consentement permettant d'établir la confiance entre une partie consentante et une partie destinataire, elle permet également aux contrats intelligents d'adhérer aux statuts règlementaires, régissant quand et comment le consentement doit être appliqué (Delacruz, 2018), elle facilite l'accès à des documents, ce qui permet de réduire ou d'éliminer les retards dans des activités qui exigeaient auparavant du temps, elle réduit les coûts d'enregistrement des informations et garantit que les dossiers sont mis à jour en temps quasi réel pour tous les utilisateurs de la chaine de blocs (Cagigas *et al.*, 2021). Elle peut, comme statué par l'Observatoire et le Forum Blockchain de l'Union européenne cité dans Reply (2022), jouer un rôle central dans la numérisation du secteur public, non seulement en termes de rationalisation administrative et de réduction des coûts et du temps qui en résulte, mais aussi parce qu'elle permet d'améliorer considérablement les niveaux de confiance, de transparence, de sécurité, de fiabilité et d'accessibilité des services, réduisant ainsi les marges d'erreur et, surtout, comblant le fossé entre les citoyens et les institutions.

La technologie de la chaine de blocs peut être abordée non seulement comme une solution innovante, mais aussi comme un outil permettant de créer efficacement de nouvelles pratiques de gestion et de nouveaux modèles de fonctionnement dans divers types d'organisations et d'institutions (Rosado, Vasconcelos & Correia, 2019; Rot *et al.*, 2020).

À l'instar de l'Estonie, dont le gouvernement a mis en œuvre des services utilisant la chaine de blocs depuis 2012 (Martinson, 2019), du Royaume-Uni (Ojo & Adebayo, 2017) et de l'Italie (Treiblmaier & Sillaber, 2020), certains pays mettent en place des systèmes modernes remplaçant les grands livres et les registres traditionnels par des applications basées sur la chaine de blocs pour stocker en toute sécurité tous les types d'informations (Rot *et al.*, 2020) prouvant l'utilité des solutions de grands livres distribués en tant qu'instrument de soutien opérationnel dans la réalisation des services de l'administration publique dans de nombreux domaines. De plus,

selon le rapport de Reply (2022), plusieurs institutions ont lancé divers essais pour remédier aux inefficacités des processus administratifs qui ont un faible niveau de numérisation, et ainsi corriger la tendance à la formation des propres « silos » de protocoles de données et de procédure de gestion de l'information qui les caractérisent (Rot *et al.*, 2020).

En se basant sur les recherches de Rosado *et al.* (2019), l'utilisation de la chaine de blocs pour l'administration publique offrent un grand nombre d'avantages tel que le confort du service pour l'utilisateur final, la réduction de la bureaucratie, la réduction des coûts de fonctionnement, la réduction des coûts liés à la protection des données personnelles, l'échange sécurisé de données et d'informations entre les acteurs à tous les niveaux ou encore des améliorations en termes d'efficacité des processus et de rapidité des transactions, bénéfiques tant pour l'utilisateur final que pour les prestataires de services.

2.4.2 Chaine de blocs et processus interorganisationnel

Du point de vue des chercheurs, la littérature scientifique sur la gestion de processus interorganisationnel basée sur la chaine de blocs et focalisée sur le secteur public est rare. Par contre, l'idée d'exploiter la chaine de blocs pour l'exécution de processus métier interorganisationnels dans les organisations privées a été étudiée dans plusieurs articles. L'absence de confiance mutuelle consécutive au fait que le contrôle est géré de manière collaborative par les participants, généralement issus de différentes organisations non fiables, identifie un besoin qui peut être résolu à l'aide de la chaine de blocs (Morales-Sandoval *et al.*, 2021). La chaine de blocs a été proposée pour gérer les processus métiers pour l'exécution et la surveillance des processus interorganisationnels où un contrat intelligent est généré spécifiquement pour chaque processus et est utilisé comme une mise en œuvre directe du médiateur de la logique du contrôle de processus (Weber *et al.*, 2016). Une étape supplémentaire a été franchie dans d'autres travaux qui proposent une approche où un seul contrat intelligent générique qui ne nécessite qu'une configuration au moment de l'exécution est déployé pour la gestion de n'importe quel processus interorganisationnel (Sturm, Szalanczi, Schönig & Jablonski, 2019), ou encore un accent est

placé sur la collecte des données d'exécution confiantes pour l'exploration des processus (Morales-Sandoval *et al.*, 2021).

Une grande partie des travaux récents portant sur l'utilisation de la chaine de blocs sur les processus interorganisationnels se concentre sur l'utilisation de contrats intelligents pour jouer le rôle du moteur de workflow et coordonner les activités entre les participants dans les organisations privées (Evermann & Kim, 2019). Cette recherche prend une direction différente en mettant un accent sur l'architecture exécutant la coordination des organisations publiques tout en conservant leur autonomie et la confidentialité de leurs processus locaux pour tirer profit des avantages de l'utilisation de la chaine de blocs dans le secteur public.

2.4.3 Service centré sur le citoyen basé sur la chaine de blocs

L'adoption d'une approche centrée sur le citoyen consiste essentiellement à aligner les services publics sur les besoins et les attentes des citoyens pour améliorer leur satisfaction (Eggers, 2018). Une facette importante de l'évolution vers des services centrés sur le citoyen est, d'une part, la transparence envers les citoyens et, d'autre part, la coordination entre les différents services impliqués dans un processus. Cette coordination soulève le défi de l'interopérabilité des données. Les principaux problèmes d'interopérabilité des données auxquels sont confrontées les organisations sont les suivants : Les données résidant à plusieurs endroits, la gestion de l'accès et des politiques à travers divers ensembles de données, l'intégration des données dans les systèmes privés des organisations participantes (Prabhakar, 2021).

La façon dont le service est fourni, comme l'utilisation des données personnelles, et la façon dont elles sont stockées, ont une influence sur la confiance des citoyens dans les services publics (Tolbert & Mossberger, 2006). La notion de confiance identifie un cas d'utilisation de la chaine de blocs. Une façon courante de résoudre ce problème est de proposer une architecture qui répond à cette problématique. La littérature scientifique traitant des services centrés sur le citoyen et basé sur la chaine de blocs comprend plusieurs propositions à l'instar de Lo *et al.* (2022), où un modèle de partage de données distribué centré sur les citoyens qui renforce la confiance

entre les citoyens et réduit les processus bureaucratiques est proposé à l'aide d'une architecture intégrant Hyperledger Fabric avec IPFS.

Parry (2019) propose une architecture centrée sur le citoyen et basé sur la chaine de blocs dans le système de soins de santé en utilisant une approche qui permet aux citoyens de contrôler et d'auditer l'utilisation et le partage de leurs données. L'auteur affirme que l'exigence clé d'une architecture de partage viable est de soutenir la confiance dans le système et entre les parties prenantes. Upadhyaya, Upadhyay, Subedi, Subedi & Gaire (2018) ont proposé une architecture pour un système de soins de santé centralisé et fondé sur une chaine de blocs privée, qui utilise le principe *« Once Only »* lors de la saisie des informations du patient, afin que ce dernier n'ait pas à fournir des informations de base à chaque fois qu'il se rend dans un établissement de soins. Biswas & Roy (2022) ont proposé une chaine de blocs pour l'authentification des utilisateurs lors d'une transaction entre un citoyen et un gouvernement, qui peut être conçue pour la fourniture de services électroniques en utilisant une approche centrée sur le citoyen .Cependant, aucune de ces approches ne prend en compte la question de la coordination et de l'interopérabilité des données dans le cadre de leur solution.

Domalis, Karacapilidis, Tsakalidis & Giannaros (2021) présente une solution décentralisée fiable et interopérable utilisant la chaine de blocs et l'intelligence artificielle qui vise à rendre les administrations publiques ouvertes, efficaces et inclusives sur une approche centrée sur le citoyen. Les auteurs utilisent l'apprentissage automatique pour reconnaître les structures de données afin de résoudre le problème de l'interopérabilité. Même si cet article présente des recherches intéressantes, il manque une évaluation pratique de l'architecture afin de comprendre les risques et les défis liés à l'interopérabilité.

2.5 Conclusion

Dans ce chapitre, les notions de base et les travaux connexes applicables dans le contexte de notre recherche ont été présentées. En résumé, ce chapitre a présenté un état de lieux de la numérisation dans le secteur public et l'utilisation des processus interorganisationnels, suivi de

la technologie des registres distribuée et plus précisément la technologie des chaines de blocs avant de présenter un état de lieu sur les méthodes de conception et d'évaluation des solutions basées sur la chaine de blocs et les travaux connexes sur l'utilisation de la chaine de blocs dans les services publics dans une approche centrée sur le citoyen.

La revue de littérature a montré que l'utilisation d'une plateforme basée sur la chaine de blocs permet non seulement de rationaliser et d'automatiser différents services, mais aussi de rendre les informations et les processus plus transparents pour les citoyens, qui peuvent alors participer plus activement grâce au contrôle de leurs données personnelles. Toutefois, il y a nécessité de repenser et d'adapter les procédures administratives pour tirer pleinement parti de l'innovation technologique de la chaine de blocs, car la numérisation des procédures administratives ne peut être réalisée uniquement en greffant de nouveaux outils dans les procédures préexistantes avec le risque inévitable de maintenir inchangée la structure traditionnelle de la procédure. En dernier, il en ressort que l'utilisation d'une approche publique-privé peut permettre de résoudre le défi de coordination tout en sauvegardant la flexibilité et le respect de la confidentialité pour les organisations participantes dans un processus interorganisationnel.

Les travaux connexes ont permis de clarifier les contributions de cette recherche par rapport aux travaux existants. Les résultats de ce chapitre ont guidé les choix de cette recherche et ont ainsi permis d'utiliser le modèle de vue 4+1 de Kruchten et la méthode d'évaluation de l'architecture basée sur les scénarios ATAM a été choisie pour l'évaluation de l'architecture proposée.

CHAPITRE 3

ANALYSE

Dans ce chapitre, il est présenté en premier une analyse de l'étude de cas qui a été utilisé pour extraire les exigences du système suivi des mesures de succès de l'expérimentation de cette étude de cas. Un aperçu comparatif entre les deux plateformes de chaine de blocs populaires est ensuite présenté dans le but de fixer le choix de la plateforme utilisée pour cette recherche.

3.1 Etude de cas

3.1.1 Processus d'immatriculation

La Société d'Assurance Automobile du Québec (SAAQ) assure l'immatriculation des véhicules routiers au Québec. Lors d'une acquisition d'un véhicule routier par un citoyen, ce dernier doit l'immatriculer pour avoir le droit de circuler sur les routes du Québec. La SAAQ perçoit au moment de l'immatriculation du véhicule routier la taxe de vente du Québec qui est déterminé par le certificat VDE-23 qui est le certificat de détermination de la TVQ, fournit par Revenu Québec (RQ), lors de l'immatriculation d'un véhicule routier. La SAAQ se base sur ce document pour percevoir le montant calculé de la TVQ.

Le processus d'immatriculation se déroule tel que présenté sur la figure 3.1 : l'acquéreur du véhicule remplit une demande du certificat VDE- 23, RQ analyse et traite la demande puis octroie le certificat VDE-23 papier à l'acquéreur. Ce dernier dépose la demande d'immatriculation à la SAAQ avec le certificat VDE-23 ainsi obtenu et d'autres documents, La SAAQ analyse et traite la demande puis perçoit la TVQ indiquée sur le certificat VDE-23 avant d'émettre le certificat d'immatriculation.

L'effort de coordination est absent. Les citoyens doivent se rendre d'un service à l'autre pour coordonner leurs efforts et obtenir l'immatriculation. Ce problème n'est pas spécifique à ce

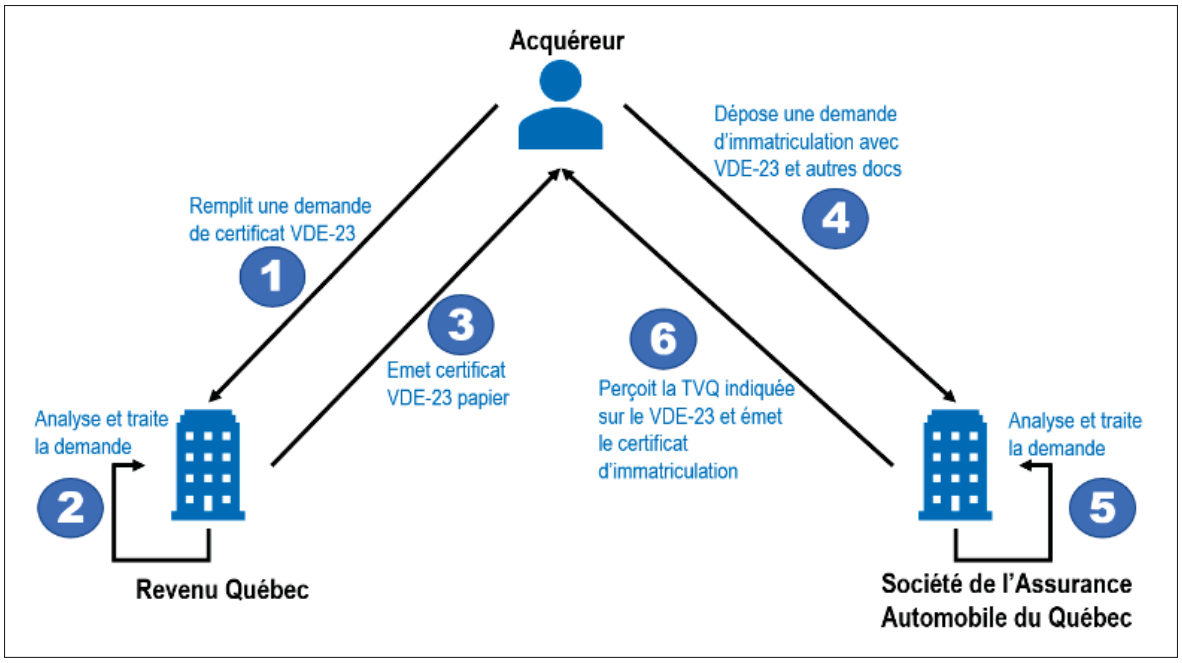

Figure 3.1 Processus de l'existant

seul service. Les services publics coopèrent pour fournir divers services aux citoyens, mais la coordination est un problème majeur : la plupart des services fonctionnent comme des silos verticalement rigides qui se consacrent à leur travail sans aucune collaboration avec les autres agences impliquées dans les autres activités du même processus.

Les principaux problèmes dans le traitement de cette demande sont repris dans le tableau 3.1.

L'objectif est d'éliminer le citoyen comme vecteur d'informations et faciliter la coordination entre RQ et SAAQ dans le processus de demande d'immatriculation.

3.1.2 Mesure de succès

Pour que la mise en œuvre de la proposition soit réellement transformatrice, l'étude doit commencer par les objectifs de conception puis appliquer la proposition basée sur la chaine de blocs lorsqu'elle peut tirer parti de ses caractéristiques clés. L'alignement entre les résultats technologiques et les objectifs de conception est le facteur clé de la réussite de la proposition.

Tableau 3.1 Domaine d'amélioration

Challenge	**Considérations**
Confiance et Sécurité	Il est techniquement possible de modifier ou de falsifier un VDE-23 sous sa forme actuelle.
Transparence limitée	Le citoyen est utilisé comme vecteur d'informations entre RQ et la SAAQ. Aucune transparence existante entre les deux organisations.
Processus lent	Le processus est lent (4-5 jours) lorsqu'on fait une demande par courrier (région éloignée) et prend généralement 30 minutes lorsqu'on fait la demande en personne, mais implique un déplacement physique.
Utilisation des papiers	La réduction des papiers dans les processus est un des objectifs des organisations.

À partir des domaines d'amélioration, des objectifs de conception ont été déduits pour la proposition. Ces objectifs sont définis suivant un cadre d'évaluation à deux niveaux, tels que présentés dans la figure 3.2 : organisationnel et applicatif. Le niveau applicatif mesure la faisabilité de l'étude de cas sur l'architecture proposée. Le niveau organisationnel identifie son applicabilité dans le secteur public. Le tableau 3.2 donne un aperçu descriptif sur les objectifs définis.

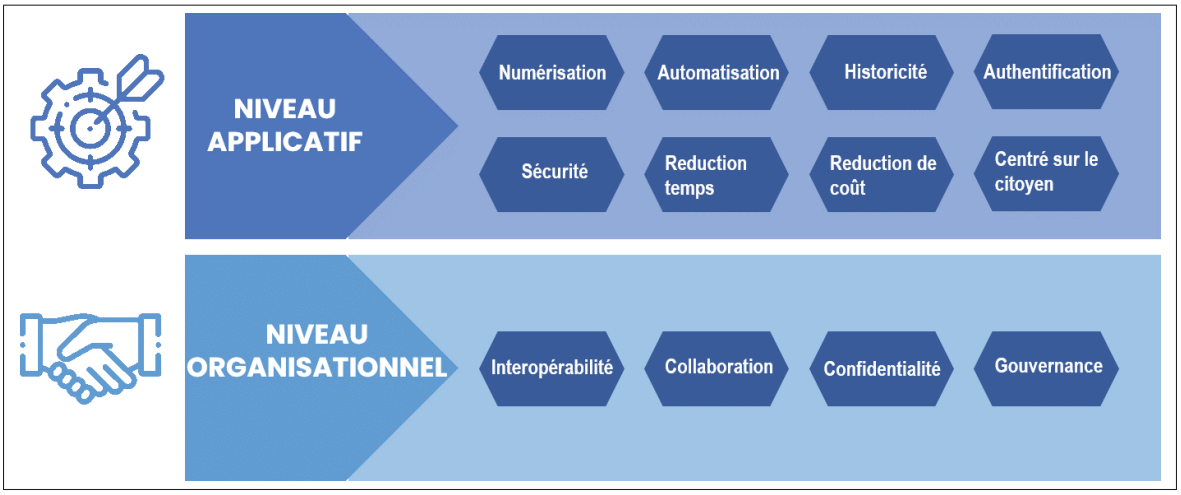

Figure 3.2 Cadre d'évaluation

Tableau 3.2 Objectifs de conception

	Objectif de conception	**Description**
DG1	Processus basé sur la numérisation	Réduire l'utilisation du papier dans le processus.
DG2	Automatiser les contrôles manuels	Simplifier les procédures d'échange de données.
DG3	Fournir un historique du processus	Rendre le processus traçable et transparent.
DG4	Authentification	Approuver l'identification des participants avant chaque action.
DG5	Sécurité	Protéger le processus contre la fraude et la falsification.
DG6	Réduction du temps	Réduire la durée globale du processus.
DG7	Réduction des coûts	Réduire le coût d'exécution du processus.
DG8	Services centrés sur le citoyen	Accorder le contrôle et l'utilisation des données personnelles au citoyen.
DG9	Interopérabilité	Résoudre le problème du fonctionnement en silos en assurant la communication tout en maintenant l'autonomie des organisations.
DG10	Collaboration	Assurer une collaboration transparente et claire en temps réel.
DG11	Confidentialité	Privatiser le processus privé de chaque organisation.
DG12	Gouvernance	Définir un mécanisme de gouvernance correct.

3.2 Choix de la plateforme de chaine de blocs

La plupart des solutions basées sur la chaine de blocs étant construites sur les plateformes Ethereum et Hyperledger Fabric, une comparaison orientée vers son utilisation dans le service public est nécessaire pour justifier le choix.

D'un point de vue technique, la chaine de blocs publique Ethereum prend environ 14 secondes pour construire un nouveau bloc. Un bloc est ainsi créé toutes les 14 secondes. Ceci définit le temps d'attente minimum entre l'envoi d'une transaction et sa confirmation. Toutefois, pour les organisations qui ont besoin d'une chaine de blocs pour leur activité, mais qui ne veulent pas

exposer leurs données au public, il existe une instance privée de l'infrastructure Ethereum. Ce système est appelé « Enterprise Ethereum » (Ethereum, 2022).

Enterprise Ethereum apporte des fonctions d'autorisation ainsi que la gestion des identités, pour offrir un réseau à autorisation. En outre, il augmente le niveau de confidentialité avec des transactions pair à pair privées pour accroître l'évolutivité et les performances et introduit également différents algorithmes de consensus autres que la preuve de travail (Anwar, 2019). Ce point présente une comparaison générale portant sur les spécificités techniques ainsi que les coûts qui entourent l'installation de Enterprise Ethereum et Hyperledger Fabric.

3.2.1 Comparaison technique

Les organisations et les règlementations ont de fortes exigences en matière d'identités et d'autorisation, ce qui tend à les conduire vers un protocole de la chaine de blocs avec autorisation. Hyperledger Fabric, Enterprise Ethereum et R3 Corda sont les trois protocoles de la chaine de blocs très utilisés selon Zhang (2019). Toutefois, Corda présente la limitation d'être spécifiquement conçu pour le secteur financier tandis que Hyperledger Fabric et Enterprise Ethereum sont des outils à usage général. Le tableau 3.3 dresse une comparaison de haut niveau entre Hyperledger Fabric et Enterprise Ethereum.

3.2.2 Comparaison du coût

Les coûts de déploiement, d'infrastructure et de maintenance des nœuds sont à prévoir dans l'installation d'une chaine de blocs privée. En se basant sur un estimateur de coûts d'un fournisseur de service infonuagique public où la tarification est basée sur l'utilisation et où il n'y a pas de frais par transaction ou pour le volume global ou encore les appels à l'API, HLF est plus couteux étant donné son architecture plus modulaire. Considérant les offres proposant les limites de ressources en fonction de la taille du nœud provisionné proposé par Kal (2021), les forfaits obtenus sont présentés sur les tableaux 3.4 et 3.5 :

Tableau 3.3 Comparaison technique entre Hyperledger Fabric et Enterprise Ethereum
Adapté de Zhang (2019)

Propriété	**Hyperledger Fabric**	**Enterprise Ethereum**
Autorisation des noeuds	Configurable au niveau du nœud, du canal et du consortium.	Règles basées sur des contrats intelligents, avec des règles par nœud basées sur des fichiers, qui peuvent être remplacées localement.
Identité	Basé sur PKI avec identité organisationnelle native. L'identité de l'organisation plutôt que les identités individuelles sont utilisées pour le consensus et l'autorisation.	Clés publiques - distribuées, et interopérables entre les chaînes basées sur Ethereum. Couplé à l'ICP via des preuves.
Consensus de transaction	Executer -> Commander -> Valider	Commander -> Executer
Responsabilité de l'application	Coordination directe avec tous les autres participants pour obtenir l'aval, gestion du verrouillage concurrentiel optimisé sur l'état, la signature et la soumission.	Envoi des transactions signées à un nœud du réseau.
Algorithmes de consensus appliqués	Tendermint Kafka/Zab (CFT avec leader de confiance), Raft (CFT avec leader de confiance).	Preuve d'autorité (BFT), Raft (CFT avec leader de confiance), Istanbul BFT (BFT avec rotation déterministe du leader).
Moteur de contrats	Virtualisation avec Docker	Machine virtuelle Ethereum
Langages de contrats intelligents	Langages complets (Go, Node.js, Java), le non-déterminisme est toléré.	Langage à déterminisme garanti (Solidity, Serpent).
Cycle de vie des contrats intelligents	Nécessite un processus élaboré pour le déploiement ou la modification. Stocké hors chaîne.	Immuable. Facile à déployer. Stocké sur la chaîne.
Mise à niveau des contrats intelligents	Remplacement du code hors chaîne par une procédure administrative et des transactions de mise à niveau.	Modèles de programmation pour étendre ou migrer le code et les données.
Tokenisation des actifs	Possible avec une solution personnalisée.	Fonctionnalité native de nombreux standards : ERC20/ERC721/ERC777...
Transactions privées	Le hachage public représente l'entrée et l'état final privé.	Le hachage public représente l'entrée.

Tableau 3.4 Limites des ressources de noeuds

Formule	**Starter**	**Small**	**Medium**	**Large**
Mémoire (Mensuel)	1 GB	1 GB	2 GB	4 GB
vCPU	0,5	0,5	1	2
Connexions simultanées	5	5	10	50
Requêtes par seconde	5	5	100	100

Tableau 3.5 Tarification à l'usage pour déploiement de chaine de blocs en préproduction

Plateforme	**Noeud**	**Prix**
Ethereum Enterprise	Noeud Ethereum : 0,15 $/ heure	0,15 $/heure /nœud
Hyperledger Fabric	Peer Node : 0,22$/heure Fabric CA : 0,03 $/heure Ordering Node :0, 07 $/heure	0,32 $/heure/nœud

En se basant sur les tableaux 3.4 et 3.5, et en considérant un écosystème autour de deux organisations avec un nombre maximal de 1000 transactions par mois, correspondant à la formule Starter, deux nœuds Ethereum Enterprise fonctionnant 730 heures (environ 1 mois) à 0,15 $ de l'heure constitueront un cout total de 220 $/mois. Pour HLF, la même configuration nécessitera deux ensembles de nœuds Fabric (Nœud pair, nœud de commande et CA) à 0,32 $ par heure par nœuds, soit un cout total de 470 $.

Dans un environnement de production, les couts sont plus élevés et pour la même configuration, Ethereum Enterprise coutera 0,55 $/heure pour un total de 803 $/mois et Hyperledger Fabric coutera 1168 $/mois.

3.2.3 Validation des principes directeurs

Les principes directeurs tels que présentés à la figure 3.3, résument les différents aspects importants à prendre en compte lors du choix de la plateforme de chaine de blocs entre HLF et Enterprise Ethereum. Globalement, le parallélisme entre les deux plateformes se résume sur la

question du niveau de configurabilité et le coût déploiement et d'entretien. HLF offrant plus de flexibilité, mais ayant un coût déploiement élevé par rapport à Enterprise Ethereum.

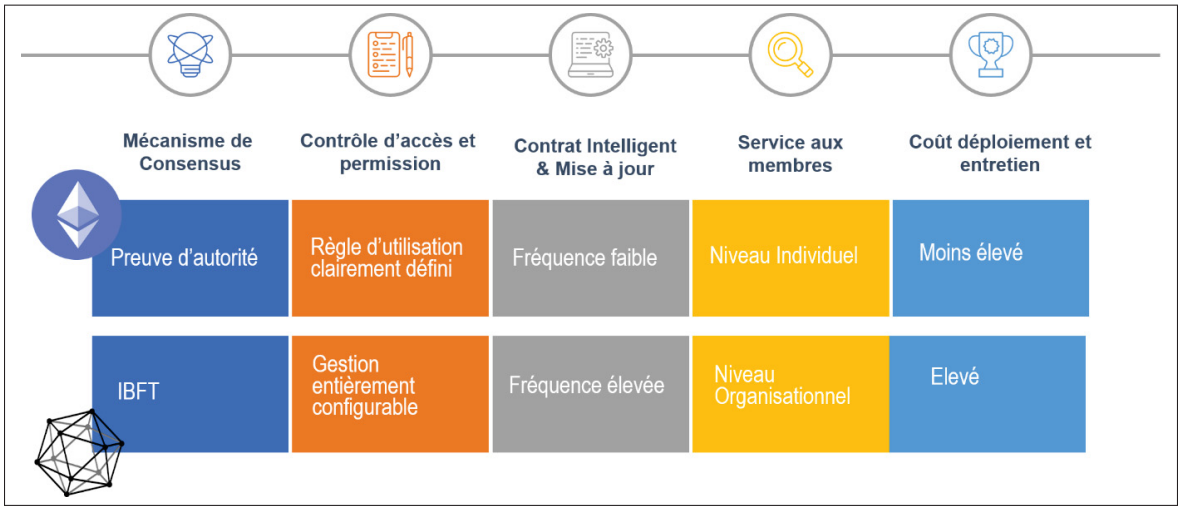

Figure 3.3 Comparaison des principes directeurs

Le choix de HLF dans l'analyse de cette recherche est donc finalement un compromis entre le niveau de configurabilité souhaité et le coût associé au projet.

3.3 Conclusion

Ce chapitre a présenté l'étude de cas sélectionné avec le concours de Revenu Québec dans la résolution de l'effort de coordination entre les organisations participantes à un processus interorganisationnel. Une étude comparative entre les deux plateformes de chaine de blocs populaires pour effectuer un choix pour la suite a ensuite été présentée. Du fait d'une plus grande flexibilité accordée, HLF est le choix effectué pour la suite de cette recherche.

CHAPITRE 4

ARCHITECTURE DU SYSTÈME

Ce chapitre présente la contribution principale de cette recherche qui est l'architecture expérimentale de processus interorganisationnel qui permet d'améliorer la collaboration par la numérisation et de mettre en pratique les concepts de chaine de blocs et d'interfaçage avec les processus privés internes de chaque organisation. À la fin, les considérations de conception entourant la proposition sont détaillées.

4.1 Architecture proposée

L'architecture proposée est basée sur la plateforme de chaine de blocs privée Hyperledger Fabric à laquelle les organisations participantes à un processus interorganisationnel peuvent se connecter. Cette architecture tire parti de la fonctionnalité des services aux membres pour maintenir une couche de contrôle d'accès permettant de gérer les participants et les actions autorisées et utilise les principes d'immutabilité pour offrir une transparence dans la collaboration par transfert de cas où tous les acteurs partagent une définition de processus et où chaque acteur exécute différentes activités pour un cas. Afin de se conformer aux principes des organisations autonomes et homologues, l'architecture s'organise autour de trois couches comme montré dans la figure 4.1 : une couche application, une couche interaction et une couche réseau.

De manière détaillée, comme montré sur la figure 4.2, l'architecture proposée dessine une organisation opérationnelle des citoyens en communication avec les services publics. Les points de vue de ces deux entités seront considérés dans la suite de cette recherche.

4.1.1 Couche réseau

La couche réseau est composée d'une chaine de blocs. Une chaine de blocs est constituée d'un ensemble de nœuds qui forment un réseau. Chaque nœud possède une copie du registre des

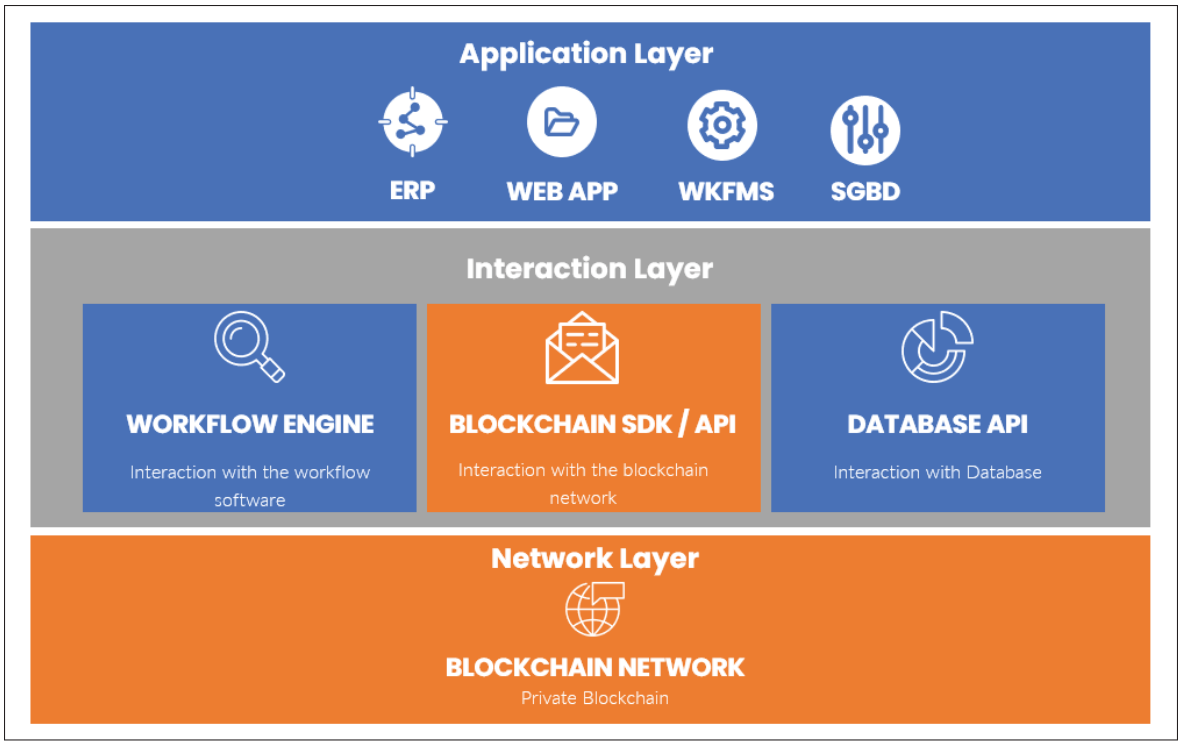

Figure 4.1 Architecture générale

transactions et des actifs. À travers un nœud, plusieurs processus mettent en œuvre les protocoles et les fonctions de la chaine de blocs. Quatre organisations seront considérées pour permettre une vue générique de la proposition : une organisation qui regroupe les nœuds de commande du réseau et trois organisations partenaires organisées autour d'un canal pour effectuer des transactions. Chaque organisation partenaire a deux nœuds pairs, dont un nœud d'approbation *(endorsing node)* qui héberge le contrat intelligent. Chaque nœud embarque un registre qui contient l'historique des transactions passées sur la chaine de blocs et une base de données d'état fournissant l'état courant. Chaque organisation exécute sa propre autorité de certification (CA). Elles sont en outre chargées de configurer leurs nœuds et créer les certificats pour les participants. Les nœuds multiples rendent l'application et la capacité d'exécution des règles hautement disponibles.

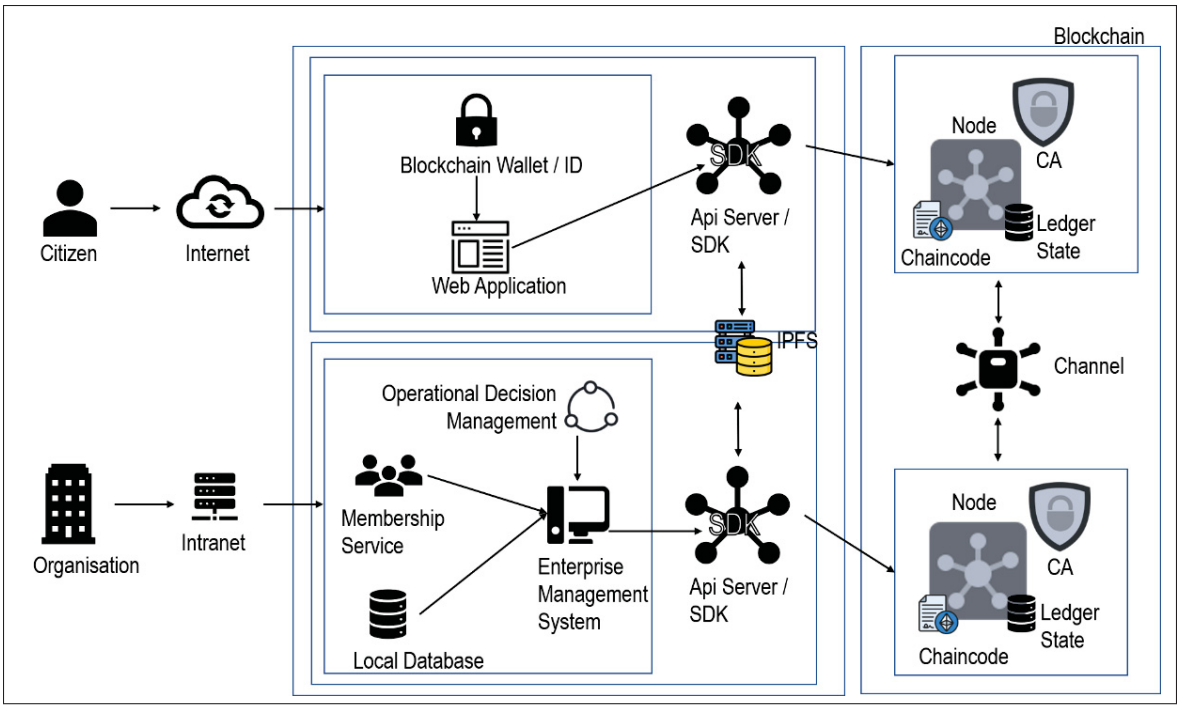

Figure 4.2 Architecture détaillée

4.1.2 Couche interaction

La couche d'interaction fournit les interfaces chargées de permettre la communication entre la couche réseau et la couche application. Elle assure l'interopérabilité technique des données, comme la structure et le format. Pour interagir avec la chaine de blocs, l'envoi et la réception de transactions se font par l'intermédiaire de nœuds de chaine de blocs qui peuvent être communiqués à l'aide d'un SDK pour permettre l'enregistrement des transactions. En outre, des événements personnalisés peuvent être déclenchés à partir des contrats intelligents, ce qui permet à la chaine de blocs d'initier des interactions et de s'intégrer au monde extérieur.

La figure 4-3 présente le flux de transaction dans la couche interaction. Les données peuvent être générées à travers une application métier personnalisée, ou un moteur de flux de travaux (de l'anglais *« workflow »*) privé à une organisation. Ces données sont ingérées à travers le SDK qui fournit un constructeur de transactions API qui cible le contrat intelligent. Le contrat intelligent définit les différents états d'un objet métier et régit les processus qui déplacent l'objet entre

ces différents états. Sur la base de la transaction, la validité de la logique du contrat intelligent s'exécute et dans le cas où elle est valide, elle est envoyée sur la chaine de blocs qui met à jour le grand livre. Dans HLF, le grand livre se compose de deux éléments essentiels, l'historique des transactions de la chaine de blocs et la base des données d'état qui est un lieu de stockage de l'état au format clé-valeur lorsque la transaction est exécutée par des contrats intelligents. Un gestionnaire de données de la chaine de blocs s'assure de renvoyer à l'application initiatrice à travers un lien de rappel HTTP (de l'anglais *« webhook »*) les données générées dans le cadre d'un évènement valide au sein de la chaine de blocs. Le flux de travaux privé pourra déclencher un ensemble d'action suivant le flux basé sur l'évènement.

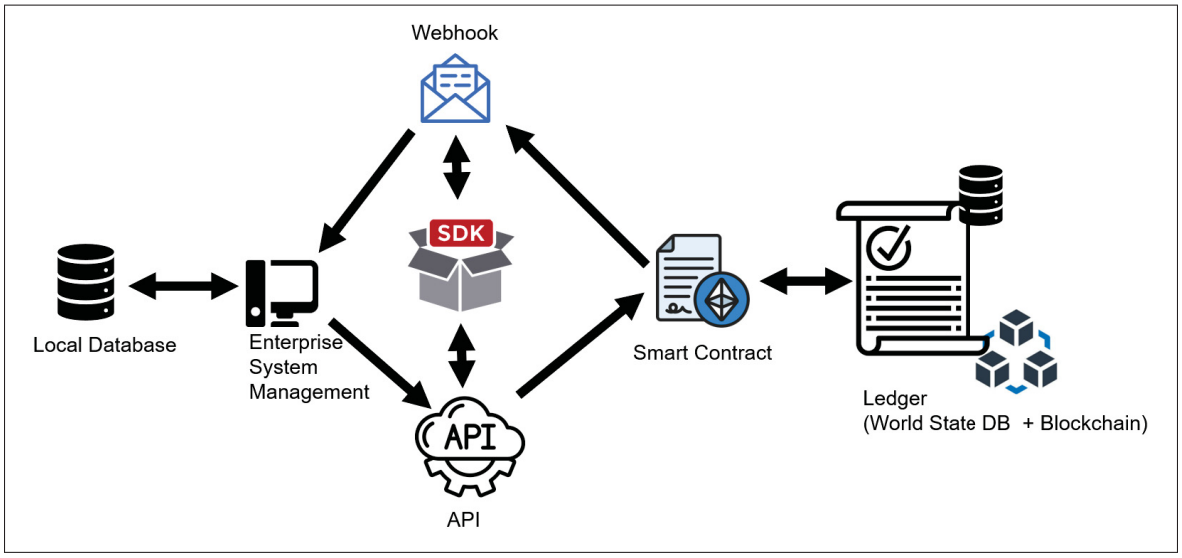

Figure 4.3 Flux de transaction sur la couche interaction

4.1.3 Couche application

Cette couche est constituée d'application ou portail spécifique servant des activités spécifiques de l'organisation auxquelles chaque groupe d'utilisateurs finaux accède avec des fonctions différentes selon la logique définie dans le contrat intelligent. Les utilisateurs finaux exécutent la logique de l'application sur le dispositif approprié (application web ou tableau de bord de

bureau). La proposition de valeur pour les utilisateurs finaux est que les informations qu'ils voient sont dignes de confiance et qu'ils peuvent ne pas être au courant des activités sur la chaine de blocs.

4.2 Considérations de conception

L'introduction d'une solution basée sur la chaine de blocs dans les procédures administratives du secteur public nécessite une analyse détaillée et doit être précédée d'un examen des procédures sur lesquelles elles fonctionnent et d'une réflexion sur leur nécessité.

4.2.1 Approche publique-privée

La technologie de la chaine de blocs étant générique, elle permet de nombreuses conceptions de systèmes différentes. Afin de s'orienter vers une architecture axée sur la flexibilité et le respect de la confidentialité où les détails des processus privés ne doivent pas nécessairement être visibles publiquement, la conception s'est faite suivant l'approche publique-privée qui considère la définition d'un flux de travaux public comme un contrat entre les acteurs participants. Et ces derniers peuvent fournir des implémentations privées pour leurs parties d'un processus, tant que celles-ci sont compatibles avec le contrat public. La figure 4.4 présente un modèle de processus public-privé pour l'étude de cas. Les étapes présentées retracent le cycle de vie de la demande d'immatriculation jusqu'à son traitement. La demande d'immatriculation sera référencée par le document VDX.

Les processus locaux de RQ, tels que l'analyse et vérification des pièces, la détermination d'admissibilité sont isolés du service de SAAQ. Seuls les détails des citoyens à vérifier sont transmis par le biais d'un processus public. De même, les processus internes de la SAAQ sont isolés de RQ et seule la livraison du certificat est partagée par un processus public. Le citoyen dispose d'une interface unique pour demander et vérifier l'état de sa demande. Ce modèle

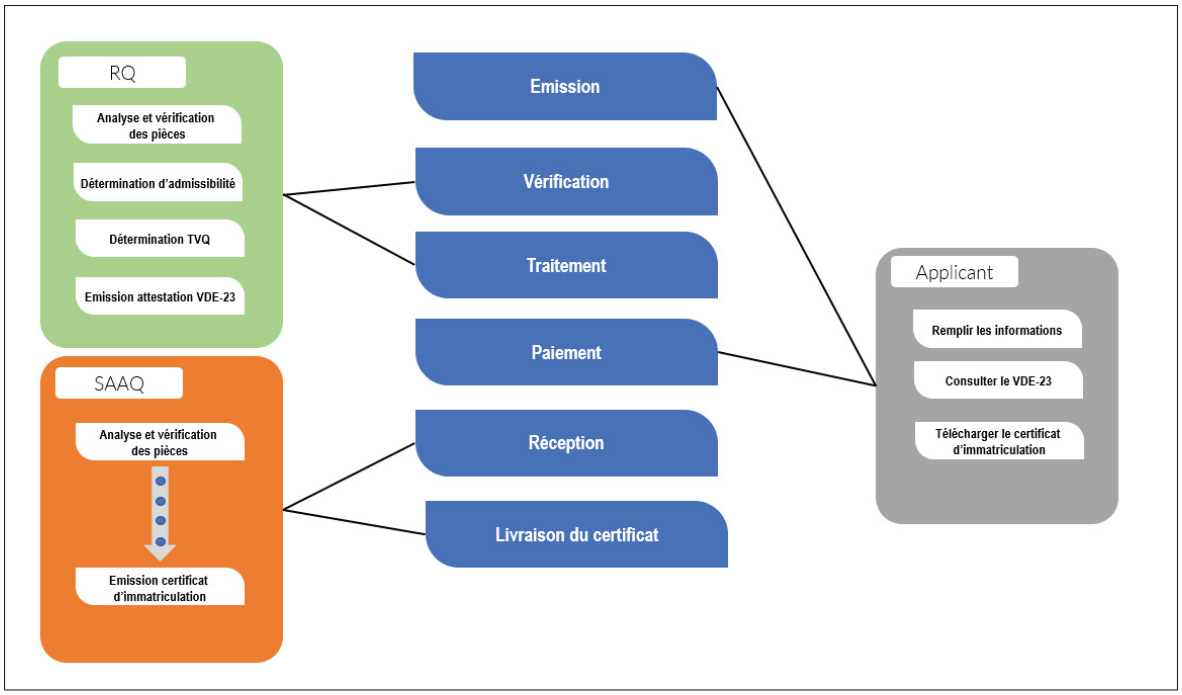

Figure 4.4 Approche publique-privée sur l'étude de cas

donne très clairement des informations concernant la propriété du processus à tout moment. Les processus privés sont la propriété des services respectifs.

4.2.2 Spécifications conceptuelles

La conception d'une chaine de blocs passe par la prise en compte d'un certain nombre des paramètres importants étant donné que les ajustements peuvent être difficiles, voire impossibles une fois le déploiement effectué.

4.2.2.1 Interactions humaines

L'identification des interactions humaines et du système est essentielle pour tracer les limites d'une solution basée sur la chaine de blocs. La figure 4.5 illustre une vue du contexte des

interactions entre les utilisateurs archétypiques (de l'anglais *« personas »*) et les systèmes externes avec la solution basée sur la chaine de blocs.

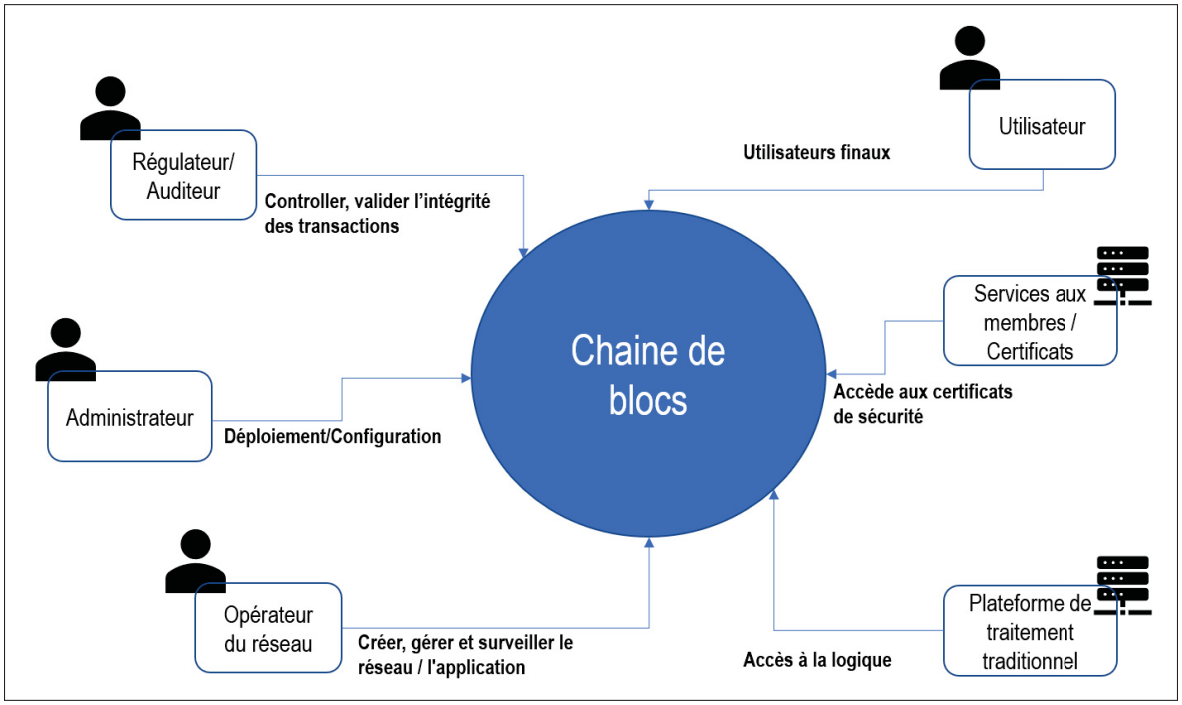

Figure 4.5 Contexte des interactions

Le contexte des interactions rend possible l'identification des participants et des rôles qui leur sont accordés. En outre, le choix du type de participation des organisations à une incidence sur la conception architecturale. Il existe différents segments d'une organisation qui peuvent être intégrés à un réseau de la chaine de blocs. Certaines d'entre elles sont de grands acteurs qui peuvent posséder un nœud pair, tandis que des organisations plus petites utilisent des services fournis par un nœud pair d'une autre organisation via une interface utilisateur de la couche d'application. Le tableau 4.1 donne un aperçu des participants et rôles selon la spécification de l'étude de cas.

Tableau 4.1 Participants et rôles

Participants	Rôles & Type	Description
Revenu Québec	Administrateur, Opérateur, Régulateur | Participant direct.	Gestion des profils des utilisateurs, administration du réseau, enregistrement des organisations et des membres, gestion des données de configuration, surveillance des réseaux de la chaine de blocs.
SAAQ	Opérateur | Participant direct.	Organisation participante du réseau, gérer et surveiller le réseau, personnaliser les personas, identifier la liste des participants membres à intégrer au réseau.
Citoyen	Utilisateur | Participant direct via RQ	Visualiser les données et permettre des transactions mineures par le biais des interfaces utilisateur.
Systèmes traditionnels, moteurs de flux de travaux privé.	Plateforme de traitement traditionnel.	Des systèmes informatiques appartenant aux organisations partenaires et utilisés par la chaine de blocs à des fins d'intégration avec les processus locaux.
Autorités de certification par défaut	Services aux membres	Fournir la validation d'identité et l'authentification.

4.2.2.2 Canal

Le canal est un élément clé pour la communication et la confidentialité des données. Les transactions dans un canal ne sont visibles que pour les participants inscrits à ce canal. Un canal global unique est l'approche qui sera utilisée étant donné que l'étude de cas ne mentionne pas de communication privée spécifique entre les participants. Dans cette approche, les participants d'un canal ont une visibilité totale des données qui sont stockées dans le grand livre du canal.

4.2.2.3 Base des données d'état

La base de données d'état influence le modèle de données, les modèles d'accès, la stratégie de déploiement, la complexité opérationnelle et la maintenabilité. CouchDB est le format de stockage de données choisi du fait qu'il permet de stocker au mieux les données sous forme de documents JSON, ce qui permet d'avoir un modèle de données plus riche, ainsi que de fournir

un support de requête riche. Il permet également d'améliorer la sécurité pour la conformité et la protection des données dans la chaine de blocs et prend également en charge la sécurité au niveau du champ par le filtrage et le masquage des entités individuelles dans une transaction.

4.2.2.4 Politique d'approbation

La politique d'approbation est déterminée en fonction de la nature de la transaction et des règles logiques impliquées dans les contrats intelligents, et détermine quelles organisations dans le canal doivent approuver la transaction. La configuration par défaut considère que tous les participants du canal doivent endosser une transaction. Cette approche a certes un impact négatif sur l'évolutivité et les performances dans le cas où le réseau dispose de plusieurs participants, mais l'étude de cas présente un nombre réduit des participants, ce qui justifie notre choix pour la configuration par défaut. En d'autres cas, il faudrait s'assurer que seul un ensemble minimal de parties participants requis pour approuver une transaction est spécifié dans la politique d'approbation qu'on peut mentionner sur le contrat intelligent.

4.2.2.5 Gouvernance

La gouvernance est considérée comme un défi majeur pour l'adoption et la mise en œuvre de la chaine de blocs dans les organismes gouvernementaux. En général, une fois qu'un contrat intelligent est déployé, les mécanismes de sa modification et de sa correction ne sont pas triviaux. Avec la version 2.2, HLF introduit un mécanisme de cycle de vie du contrat intelligent qui permet à plusieurs organisations de se mettre d'accord sur la manière dont un contrat intelligent sera exploité avant de pouvoir être utilisé sur un canal. L'approche utilisée est une gouvernance centralisée hors chaine coercitive. Le caractère centralisé fait référence à l'organisation de la gouvernance autour d'un groupe spécifique d'organisations qui prend les décisions de gouvernance. Le caractère hors chaine fait référence au mécanisme de prise de décision dont les processus de délibérations sont faits en dehors de la chaine et le caractère de

coercition fait référence à la responsabilité de la gouvernance stipulant que les règles d'échanges sont préalablement définies et sont des codes intégrés dans les contrats intelligents.

4.2.3 Spécifications fonctionnelles

Les fonctionnalités de base qui ont été prises en compte dans la phase de conception concernent l'élaboration du contrat intelligent, l'intégration avec les systèmes existants, le stockage sur chaine ou hors chaine ainsi que le processus d'authentification.

4.2.3.1 Contrat intelligent

Les contrats intelligents sont des codes personnalisés qui encapsulent les règles et les processus logiques qui régissent les transactions. La manière de gérer la logique de décision pour équilibrer la flexibilité, l'opérabilité et la complexité de la solution est une décision essentielle de conception. L'approche utilisée est celle d'encapsuler la logique de décision dans le code du contrat intelligent, ce qui facilite sa gestion du point de vue du temps même si cette approche est moins flexible, car elle nécessite l'intervention du service informatique pour modifier les contrats intelligents à chaque fois qu'un changement doit être effectué.

L'interaction au niveau des processus locaux est gérée par des moteurs de flux de travaux propre à chaque organisation, celle au niveau des processus publics est définie sur le contrat intelligent. Les deux concepts importants qu'il sied de soulever lorsqu'il s'agit de processus sur une chaine de blocs sont les états et les transactions. Il existe des objets conceptuels de valeur, modélisés comme des états, dont les transitions du cycle de vie sont décrites par des transactions. Le cycle de vie d'un papier VDX peut être représenté à l'aide d'un diagramme de transition d'état présenté sur la figure 4.6.

Le diagramme d'état, présenté sur la figure 4.7, décrit l'évolution des VDX au fil du temps et comment des transactions spécifiques régissent les transitions du cycle de vie. Le contrat intelligent défini met en œuvre une logique de transaction qui fait passer les VDX d'un état à

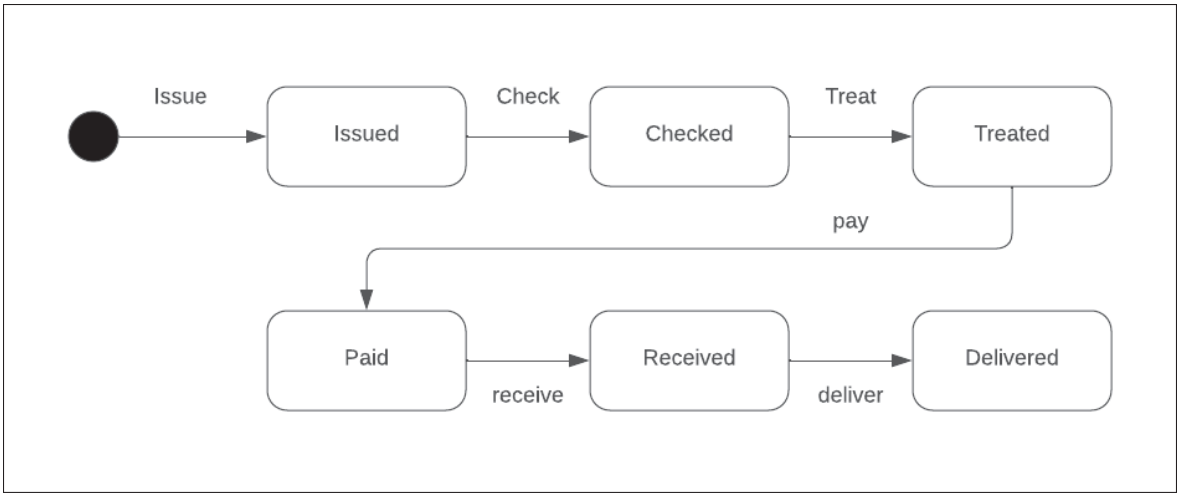

Figure 4.6 Diagramme de transition d'état

l'autre comme le montre la figure 4-7. Les états des VDX sont en fait conservés dans la base de données d'état.

4.2.3.2 Données sur chaine ou hors chaine

La recherche de l'équilibre approprié entre les données sur chaine et hors chaine est une question importante pour une solution basée sur la chaine de blocs. Il est important de savoir quelles données doivent être stockées sur chaine, car le nombre et la taille des objets ont un impact sur les performances, et l'évolutivité du réseau est affectée puisque l'objet fera partie de la charge utile échangée au cours du processus de consensus. L'approche utilisée considère l'enregistrement hors chaine des données telles que les documents échangés en utilisant le système de stockage décentralisé hors chaine IPFS. IPFS distribue les fichiers sur le réseau et chaque fichier est adressé par son contenu. Lorsqu'ils sont stockés dans le système, un hachage cryptographique leur est attribué en fonction de leur contenu, contrairement à l'adressage traditionnel utilisé par HTTP, où un seul serveur héberge de nombreux fichiers et où les informations doivent être récupérées en accédant à ce serveur. Avec cette approche, le réseau est évolutif, mais cela augmente la complexité de l'application en dirigeant les opérations à la fois vers la chaine de blocs et vers le stockage hors chaîne.

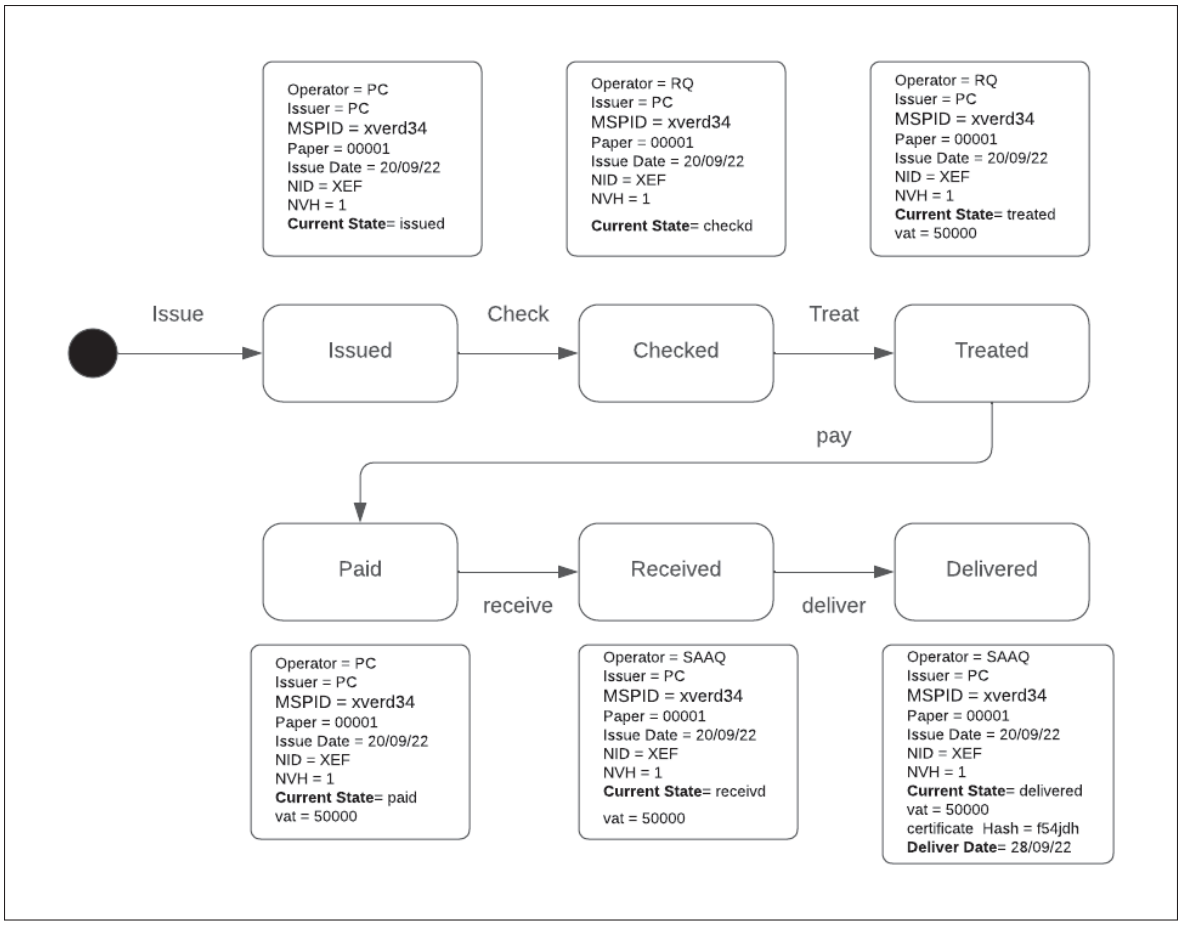

Figure 4.7 Diagramme d'état avec aperçu de l'évolution du VDX

4.2.3.3 Intégration avec les systèmes existants

Les organisations disposent de systèmes existants, désignés par plateforme de traitement traditionnel, qui sont des systèmes assurant l'exécution des traitements au sein de l'organisation. La chaine de blocs est utilisée dans notre contexte pour suivre le cycle de vie d'un document sur un processus public ainsi que les interactions entre les participants du réseau. L'objectif étant de partager des données entre plusieurs entreprises d'une manière vérifiable et immuable. Il est nécessaire de transformer les données entre la chaine de blocs et les formats des systèmes existants. L'approche utilisée consiste à utiliser un modèle d'intégration de service API pour établir un pont entre les données des deux formats différents (dans la chaine de blocs et les

systèmes existants). La communication de services hors chaine avec la chaine de blocs se fait par la couche du kit de développement logiciel (SDK). Le service cible exécute une requête sur la couche de la chaine de blocs par le biais du SDK en passant par les points d'accès API. Cela nécessite la configuration des certificats et la configuration de la liste de contrôle d'accès. La figure 4.8 donne un aperçu de l'interconnectivité entre la chaine de blocs et un service externe. Le service API permet ainsi la connexion entre une application cliente d'une organisation et les nœuds HLF dont dispose cette organisation.

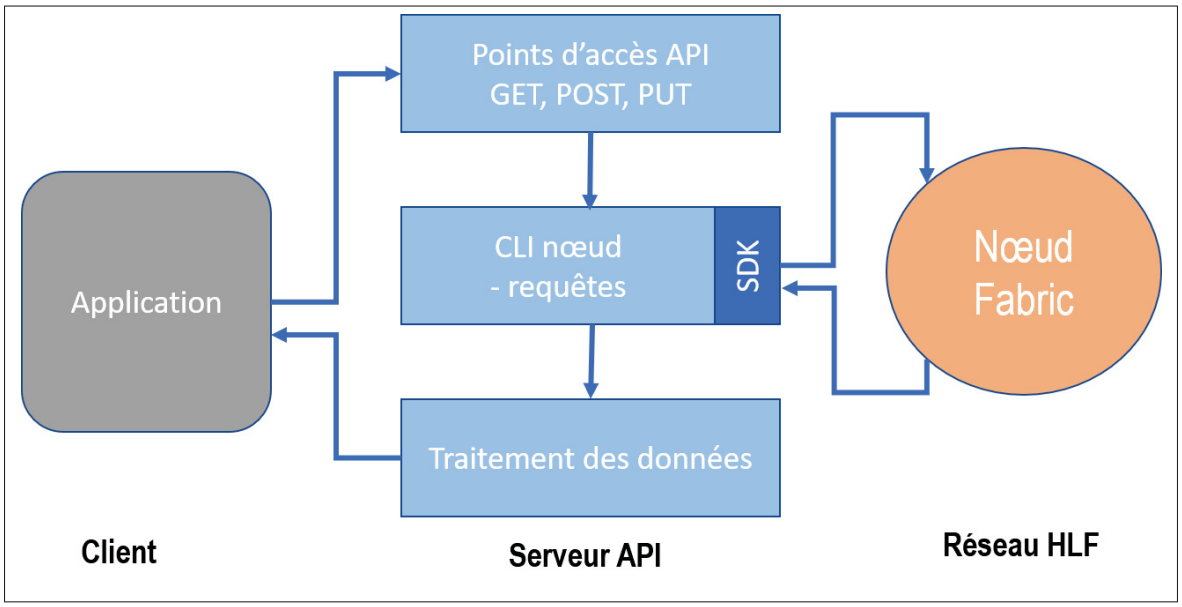

Figure 4.8 Interconnectivité avec la chaine de blocs

4.2.3.4 Authentification

Les applications accèdent au code de la chaîne ou l'invoquent par l'intermédiaire de la couche d'interaction qui intègre les points d'accès API. Le couche d'interaction est responsable de la mise en correspondance de l'identité de l'organisation avec l'identité sur la chaine de blocs (à l'aide du SDK) qui est nécessaire pour accéder au code de la chaîne fourni par la solution et l'invoquer. Dans HLF, l'identité est représentée par un certificat numérique X.509. Lorsqu'il

interagit avec un réseau HLF, l'acteur spécifie et présente le certificat numérique, et le réseau HLF accepte ou refuse en fonction de la politique. Comme illustré sur la figure 4.9, l'utilisateur reçoit un certificat numérique avec les informations appropriées qui lui sont associées. Le certificat numérique est généré par l'enregistrement et l'inscription auprès d'une autorité de certification. Dans la mesure où le certificat numérique présenté par un utilisateur est émis par une autorité de certification à laquelle le réseau HLF fait confiance, l'opération de l'utilisateur sera acceptée et traitée par le réseau HLF.

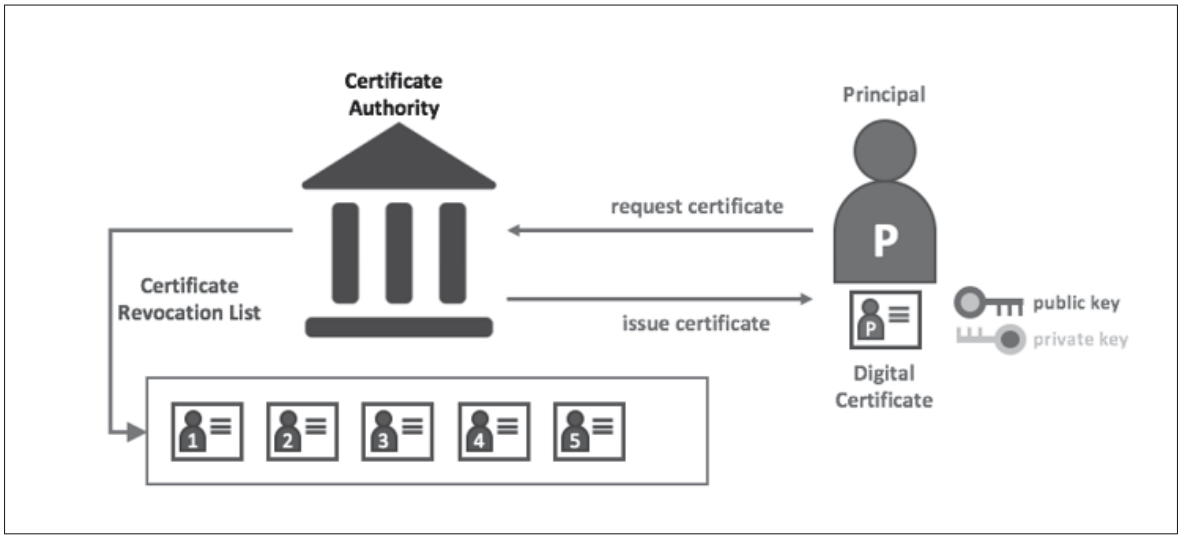

Figure 4.9 Mécanisme d'authentification de HLF
Tirée de HLF (2022)

Toutefois, pour établir une uniformité entre tous les services publics dans le cas où cette approche devrait être dupliquée dans plusieurs cas d'utilisation, il serait judicieux de centraliser le processus d'authentification dans une organisation générale comme montré dans la figure 4.10. De cette façon, un citoyen pourrait créer un compte sécurisé et une identité numérique qui stockerait ses informations personnelles en un seul endroit et remplirait automatiquement les informations précédemment saisies dans les formulaires de nombreux services publics différents. La mise en place de ce mécanisme d'authentification augmenterait le niveau de satisfaction des citoyens.

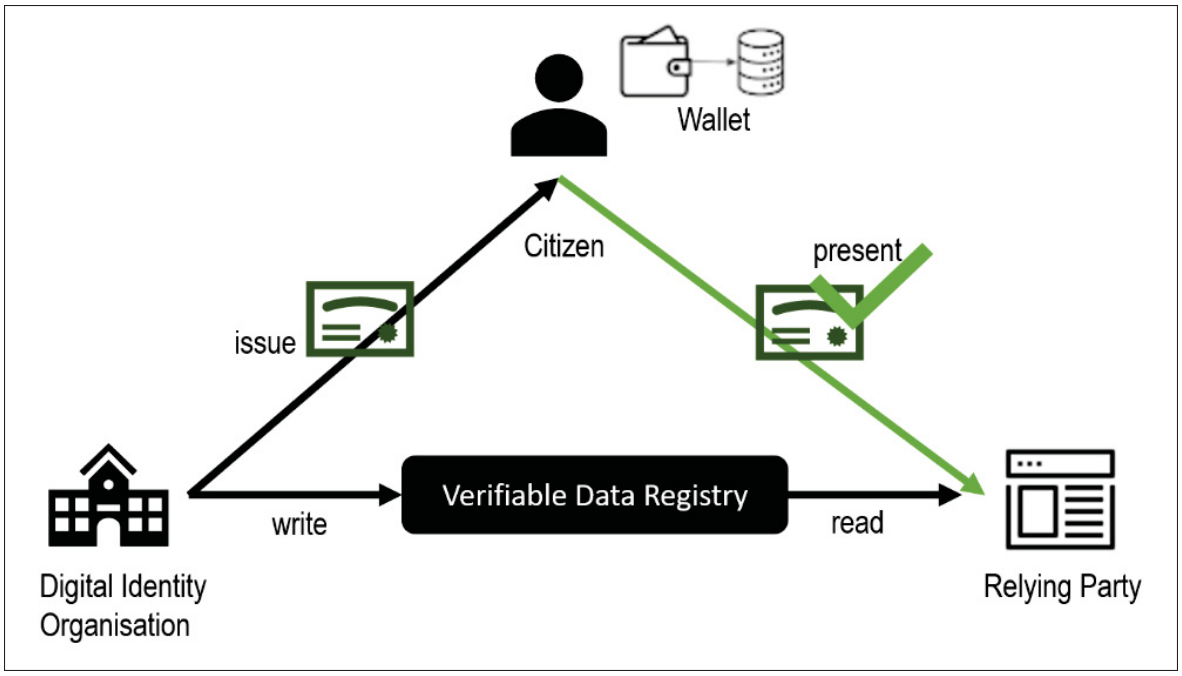

Figure 4.10 Mécanisme d'authentification préconisé
Adaptée de Affinidi (2021)

4.2.4 Spécifications non fonctionnelles

En plus des spécifications fonctionnelles, certaines caractéristiques non fonctionnelles à savoir la sécurité des contrats et la traçabilité des processus ont fait l'objet d'une attention particulière dans la conception de la solution attendue.

4.2.4.1 Sécurité

Avec l'utilisation d'un canal et du mécanise de collection des données privées d'une part, la vie privée et la confidentialité sont assurées. Toutes les données du canal défini, y compris les informations sur les transactions et les membres, sont invisibles et inaccessibles aux membres du réseau qui n'ont pas accès à ce canal. D'autre part, l'utilisation de clés de cryptage avec PKI fournit aux organisations un niveau de sécurité plus élevé. De plus, un mécanisme de cryptage des données est utilisé avant la manipulation des données sur la chaine de blocs. En résumé, les

divers aspects supplémentaires qui doivent être pris en compte dans la conception comportent divers aspects de sécurités :

- Séparation du réseau de la chaine de blocs en canaux, où chaque canal représente un sous-ensemble de participants autorisés à voir les données des contrats intelligents qu'ils déploient sur ce canal ;
- Hachage des données avant d'appeler un contrat intelligent et définition d'un mécanisme pour partager les données sources ; et
- Utilisation des listes de contrôle d'accès pour restreindre l'accès aux données à certains rôles dans la logique du contrat intelligent.

4.2.4.2 Traçabilité

L'exécution d'un processus interorganisationnel nécessite la confiance entre les différentes parties. Grâce à sa vérifiabilité, la chaine de blocs permet d'établir cette confiance. Chaque transaction ajoutée à la chaine de blocs possède une signature et un horodatage. Cela signifie que les organisations peuvent retracer chaque transaction à un moment précis et identifier la partie correspondante sur la chaine de blocs (via son identifiant sur le réseau). Cette fonctionnalité garantit la non-répudiation, c'est-à-dire la capacité de s'assurer qu'une partie à un contrat ou à une communication ne peut pas nier l'authenticité de sa signature sur un document ou l'envoi d'un message dont elle est à l'origine et assurer un audit efficace des données.

4.3 Conclusion

Ce chapitre a présenté une architecture en trois couches conçues autour d'une intégration entre les divers systèmes de traitement des demandes propres à chaque institution. En dernier lieu, la conception de la proposition a été détaillée. Un accent particulier a été mis sur les composants conceptuels étant donné que les ajustements peuvent être difficiles, voire impossibles une fois qu'une solution basée sur la chaine de blocs est déployée. En dernier, ce chapitre a présenté les spécifications fonctionnelles et non fonctionnelles de notre proposition.

CHAPITRE 5

CONCEPTION DU SYSTÈME

Ce chapitre décrit l'architecture logicielle de la solution proposée en suivant le modèle de vue « 4 + 1 » de Kruchten (1995). L'architecture est présentée du point de vue logique, développement, processus, et physique en fournissant une analyse de chaque cas suivi d'un diagramme UML. Le scénario a été décrit au chapitre 3.

5.1 Vue logique

La vue logique concerne la fonctionnalité du système d'un point de vue des utilisateurs finaux en se focalisant sur le contrat intelligent. Le contrat intelligent définit la partie de l'architecture qui est responsable de l'écriture et de la lecture sur la chaine de blocs. Le contrat intelligent reprend l'enchainement des processus publics tel que défini par l'approche utilisée. Il utilise un identifiant unique constitué du numéro du document traité et de la signature de l'utilisateur effectuant la demande. Il conditionne l'exécution d'une fonction par la validation de l'opérateur en cours en fonction du statut du document traité et la conformité de l'identité de l'opérateur effectuant l'action. Il garde la trace des résultats et n'accepte les appels que des entités autorisées. Une description détaillée du contrat est donnée dans la figure 5.1 en utilisant un diagramme de classe et son code source se trouve à l'annexe II.

L'API de contrat Hyperledger Fabric fournit une interface pour le développement de contrats intelligents et une interface pour le contexte transactionnel pour une exécution donnée. Chaque contrat intelligent développé doit étendre la classe Contrat de cet API, puis mettre en œuvre la logique requise. La classe Contexte peut être sous-classée pour fournir un comportement fonctionnel supplémentaire afin de soutenir l'exécution de contrats intelligents. Elle est étendue pour fournir une aide supplémentaire en mappant les identifiants des objets de l'application aux clés composites de la base des données d'état et permet ainsi d'accéder facilement à la liste de tous les objets.

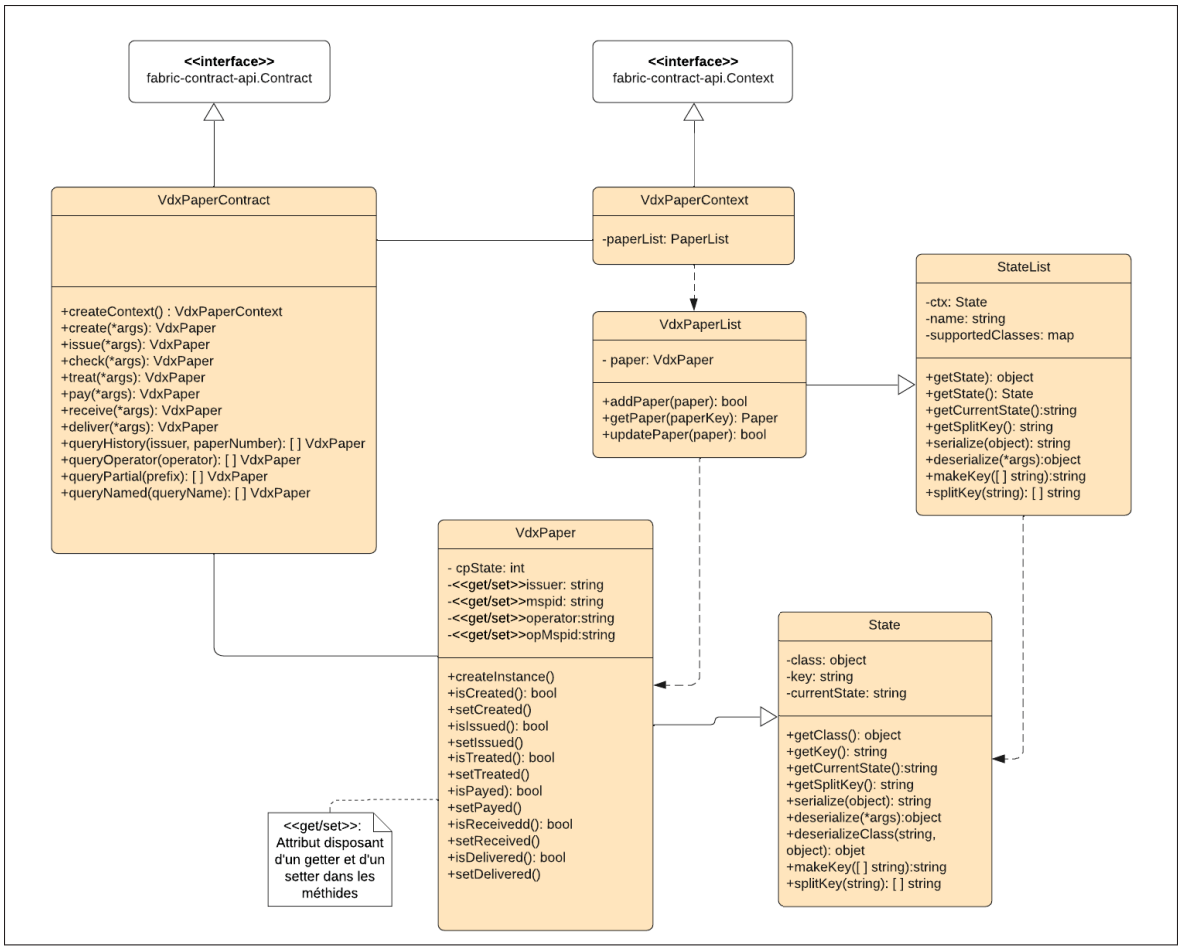

Figure 5.1 Vue logique à l'aide du diagramme de classe

Les principales classes présentées sont :

- La classe *State* : Les états ont une classe, une clé unique et un état actuel du cycle de vie. L'état actuel est déterminé par la sous-classe spécifique ;
- La classe *StateList* fournit un conteneur virtuel nommé pour un ensemble d'états de grand livre. Chaque état a une clé unique qui l'associe au conteneur, plutôt que le conteneur contenant un lien vers l'état. Cela minimise les collisions pour les transactions parallèles sur différents états ;
- La classe *VdxPaper* étend la classe State, c'est la classe qui sera utilisée par l'application et le contrat intelligent pour définir un papier ; et

- La classe *VdxPaperContract* définit le contrat intelligent VdxPaper en étendant la classe HLF Contract. Elle regroupe la méthode exécutant les changements d'état et dispose en outre des requêtes pour les fonctions d'interrogation telles que l'historique et la récupération des valeurs.

5.2 Vue développement

La vue du développement dépeint un système du point de vue d'un programmeur et décrit les composants et les modules utilisés pour assembler l'application. Le diagramme de composant, présenté sur la figure 5.2, est l'un des diagrammes UML utilisés pour représenter la vue de développement. Dans notre vue, deux approches différentes sont considérées : l'une ayant comme point focal l'entité Citoyen et l'autre, l'entité Organisation, faisant référence à tous les organismes publics. À noter l'utilisation du prototype « EXT » pour justifier les dépendances externes incluses dans notre de notre solution. Le tableau 5.1 résume ces éléments.

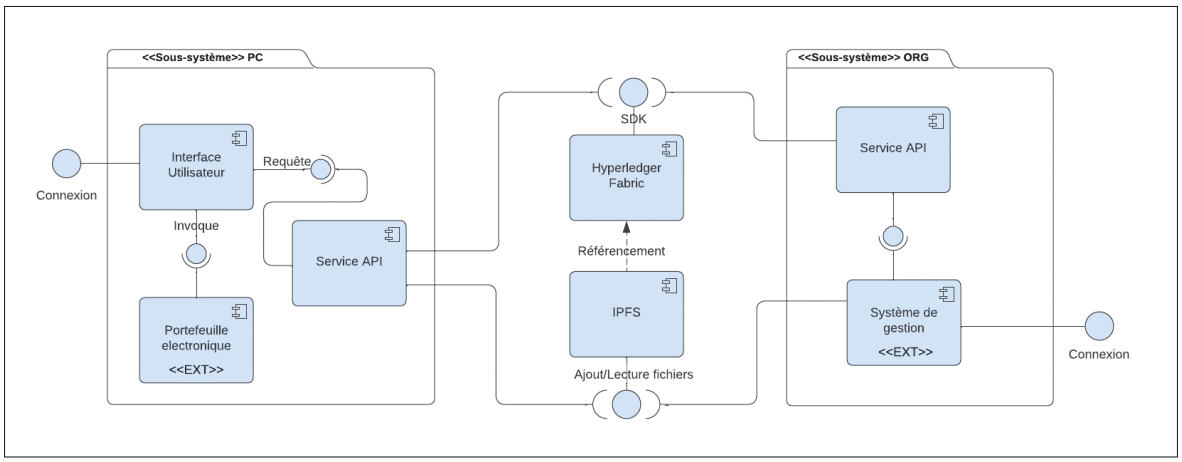

Figure 5.2 Vue développement à l'aide du diagramme de composant

Tableau 5.1 Composants du système

Composant	**Observation**
Interface utilisateur (Navigateur web)	Composant utilisé pour fournir une interface conviviale permettant d'interagir avec l'application et la chaine de blocs.
Service API	Deux composants différents utilisés pour interfacer l'échange avec les contrats intelligents suivant les dispositions d'échanges relatives à l'organisation regroupant les citoyens (pour le 1er), et les dispositions d'échanges relatives à un organisme public.
Portefeuille électronique	Composant gérant l'identité numérique des citoyens sur la chaine de blocs. « EXT ».
Système de gestion	Composant d'interaction propre à une organisation et permettant d'interfacer le processus public au processus privé interne à un organisme public.
Chaine de blocs HLF	Composant utilisé pour déployer des contrats intelligents et stocker des informations sur les processus publics et l'état de document.
IPFS	Module utilisé pour héberger des fichiers joints au processus public par le réseau IPFS.

5.3 Vue processus

La vue processus se concentre sur le comportement du système en cours d'exécution et traite des éléments dynamiques du système. Elle présente un point de vue pour les chefs de projets et explique les interactions entre les acteurs et les différents composants du système. Le diagramme d'activité est utilisé pour présenter notre vue processus. Cette section décrit le processus pour deux entités différentes, les citoyens qui sont donc des utilisateurs individuels et les organisations, qui représentent les organismes du secteur public. La figure 5.3 illustre le processus d'interaction avec le système du point de vue d'un citoyen afin d'offrir des ressources en matière de données, de logiciels ou de conteneurs. À l'étape 1, les citoyens se connectent sur la plateforme web en utilisant un mécanisme d'authentification connecté à un portefeuille électronique. À ce niveau, cette recherche recommande que le composant portefeuille électronique soit géré par une entité tierce qui s'occupe des identités numériques des citoyens. La problématique étant ici la nécessité de rattacher une demande à une signature numérique représentant en unique un citoyen. Les

citoyens stockent leurs fichiers sur IPFS. Ces fichiers peuvent être des ensembles de données ou des fichiers. Les ensembles de données doivent être chiffrés par le citoyen avant d'être ajoutés sur IPFS. Une fois que ces fichiers sont ajoutés au réseau IPFS, leurs métadonnées ainsi que l'adresse IPFS retournée sont rattachées à la requête soumise par le citoyen et stockée sur la chaine de blocs avec la référence de la requête.

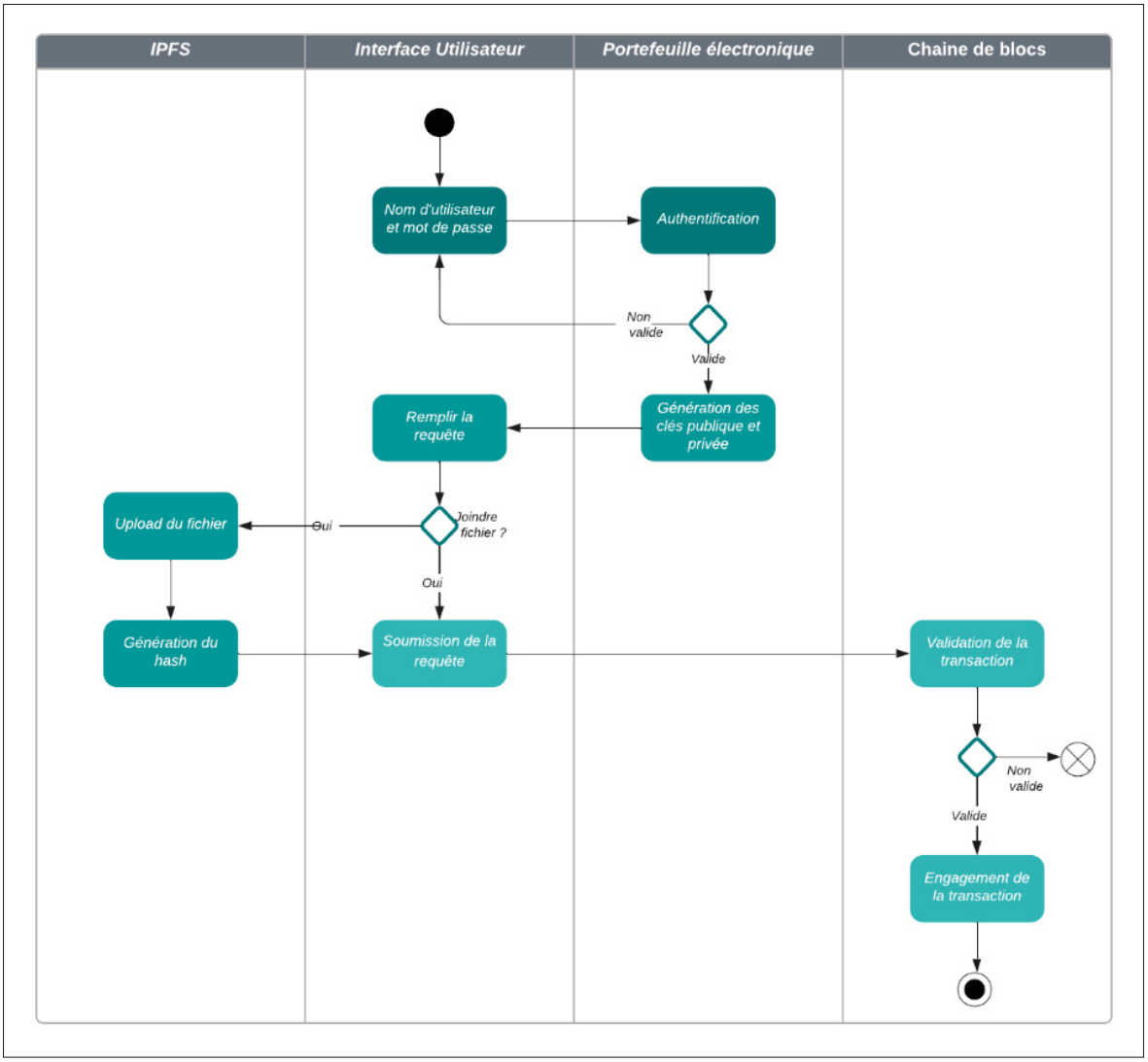

Figure 5.3 Diagramme d'activité : Vue Citoyen

D'un point de l'entité Organisation, tel que présenté dans la figure 5.4, l'élément principal et l'élément déclencheur de tout, c'est le système de gestion interne à une organisation. Les utilisateurs de l'organisation se connectent en passant par un mécanisme interne. Le lien entre le processus privé dans le système de gestion interne et le processus public exécuté sur la chaine de blocs est représenté par le service API qui fait le pont avec la chaine de blocs.

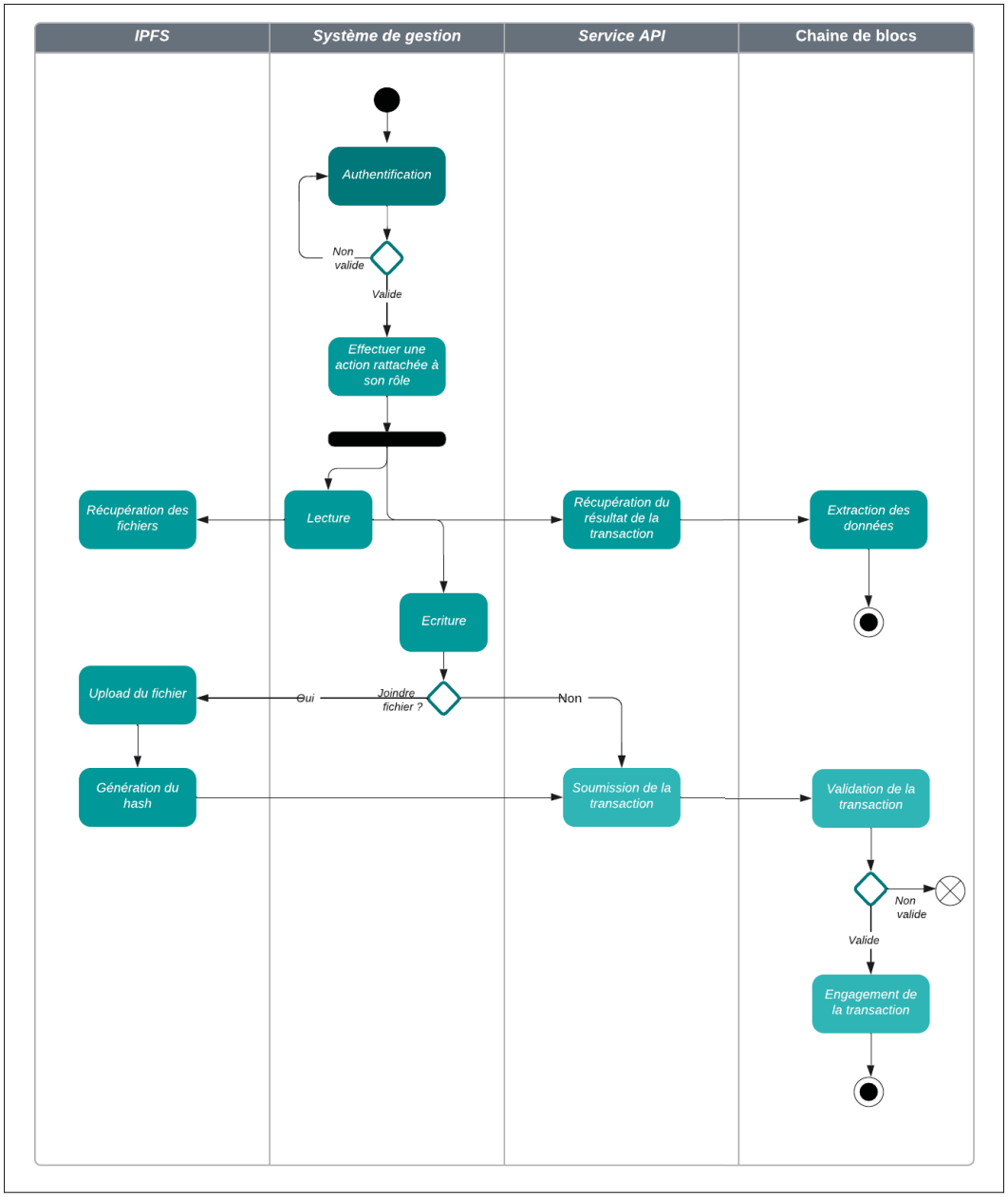

Figure 5.4 Diagramme d'activité : Vue Organisation

En fonction du statut de traitement dans le processus public, le contrôle est envoyé à un utilisateur spécifique d'une organisation et le traitement se poursuit jusqu'à atteindre un niveau en local qui correspond à un changement d'état dans le processus public. Toutes les opérations de lecture et d'écritures se font au travers du service API. Le serveur de stockage décentralisé IPFS est aussi utilisé pour récupérer ou joindre les données.

5.4 Vue physique

La vue physique dépeint le système du point de vue d'un ingénieur système. Elle s'intéresse à la topologie des composants logiciels ainsi qu'aux connexions physiques entre ces composants. Le diagramme de déploiement est l'un des diagrammes UML utilisés pour représenter la perspective physique. Il décrit comment le logiciel et le matériel de l'application sont connectés et comment le système est installé lors de son fonctionnement. La figure 5.5 donne un aperçu à large échelle du diagramme de déploiement UML de l'application. Elle présente trois périphériques : le premier est un ordinateur portable sous Windows, MacOs, ou Linux ayant comme environnement d'exécution le navigateur, il exécute l'interface utilisateur. Le deuxième périphérique est un serveur d'applications s'exécutant sur Node.js et intégrant le SDK de HLF pour la communication avec la chaine de blocs, un composant IPFS pour le lien avec le système de stockage décentralisé et un composant Express pour la création d'un service API. Le troisième périphérique est celui qui contient notre réseau de la chaine de blocs. Ses composants sont chacun déployés sur un environnement d'exécution virtuelle utilisant Docker. Chaque composant, à savoir les nœuds, le certificat d'autorité ou encore la base de données d'état est déployée sur un conteneur Docker spécifique. Il faudra dupliquer ce périphérique en fonction du nombre d'organisations des participants directs et le connecter ensemble en un seul réseau en utilisant Docker *Swarm*.

5.5 Conclusion

Dans ce chapitre, le modèle de vue « 4+1 » de Krutchen a été utilisé pour organiser les représentations de l'architecture de l'application en différentes vues qui répondent aux besoins

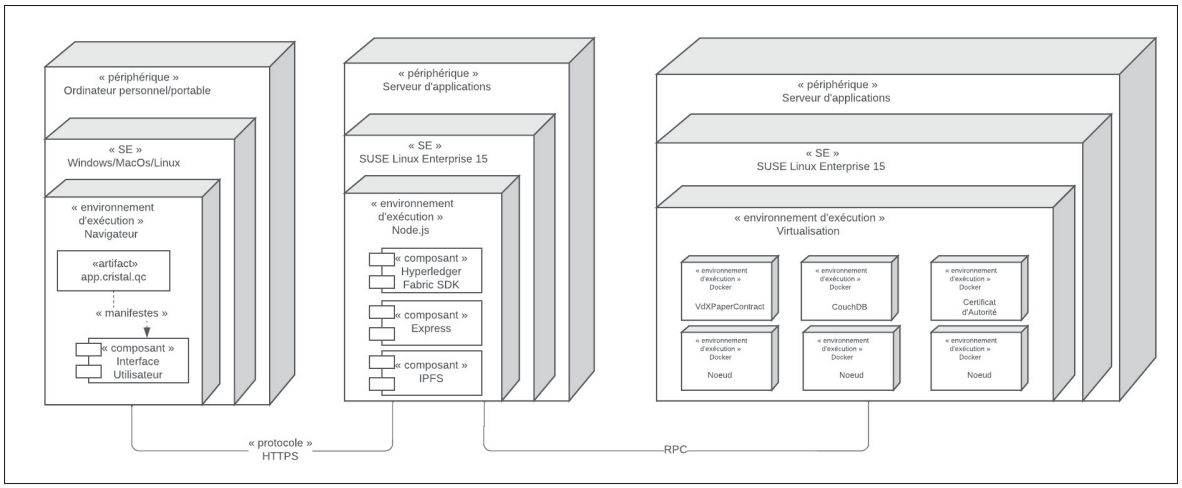

Figure 5.5 Vue physique à l'aide du diagramme de déploiement

des utilisateurs en décrivant un point de vue unique pour les utilisateurs finaux, les développeurs, les ingénieurs système et les chefs de projet. Les résultats de ce chapitre permettront de détailler la présentation de l'architecture à d'autres participants à des fins d'évaluation.

CHAPITRE 6

IMPLÉMENTATION ET ÉVALUATION

Ce chapitre présente l'environnement expérimental qui a été utilisé pour la mise en œuvre de notre preuve de concept. Une évaluation du système basé sur l'architecture proposée en utilisant la méthode ATAM et une observation basée sur les scénarios de l'interaction avec le système sont ensuite présentées. Enfin, les résultats obtenus sont discutés est les enseignements généralisables en sont tirés.

6.1 Configuration expérimentale

Dans l'objectif de tirer profit de la décentralisation et du caractère distribué, le réseau HLF a été déployé sur plusieurs machines virtuelles, chaque organisation étant placée dans une machine virtuelle séparée dans Microsoft Azure et connectée les uns aux autres en utilisant Docker *Swarm* pour les faire communiquer. La configuration suivante a été utilisée pour la mise en œuvre et le développement de notre proposition : la version 2.2 de Hyperledger Fabric, le langage JavaScript sur un environnement d'exécution Node.js version 1.0 pour le script du contrat intelligent et le service API, le script Bash version 5.0.17 pour l'automatisation des commandes réseaux, Docker 20.10.10 pour la virtualisation, CouchDB 3.1.1 pour la base des données d'état, le *framework* Django version 3.2 pour l'application web utilisant le langage Python pour le *back-end*, ainsi que le langage HTML et CSS pour la partie *front-end*. La configuration matérielle en local était basé sur un ordinateur portable de 8vCPU, 16GB, équipé d'un processeur 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz fonctionnant sur Ubuntu 20.04.4. Les machines virtuelles sur Microsoft Azure ont été créés suivant la spécification Standard_B8ms.

Le réseau comprend trois organisations participantes (PC, RQ et SAAQ) et une organisation de commande. Deux nœuds par organisations sont considérés : un nœud pair pour la redondance, et un nœud d'approbation qui contient le contrat intelligent.

La figure 6.1 donne un aperçu de haut niveau de la topologie définie. En plus de la chaine de blocs HLF et de ses composants, l'architecture comprend une application cliente et un service API spécifiques à chaque organisation respective. Chaque service API fait ainsi le pont entre l'application cliente et le nœud d'une organisation donnée et dispose des fonctions qui correspondent spécifiquement à l'activité d'une organisation sur la chaine de blocs.

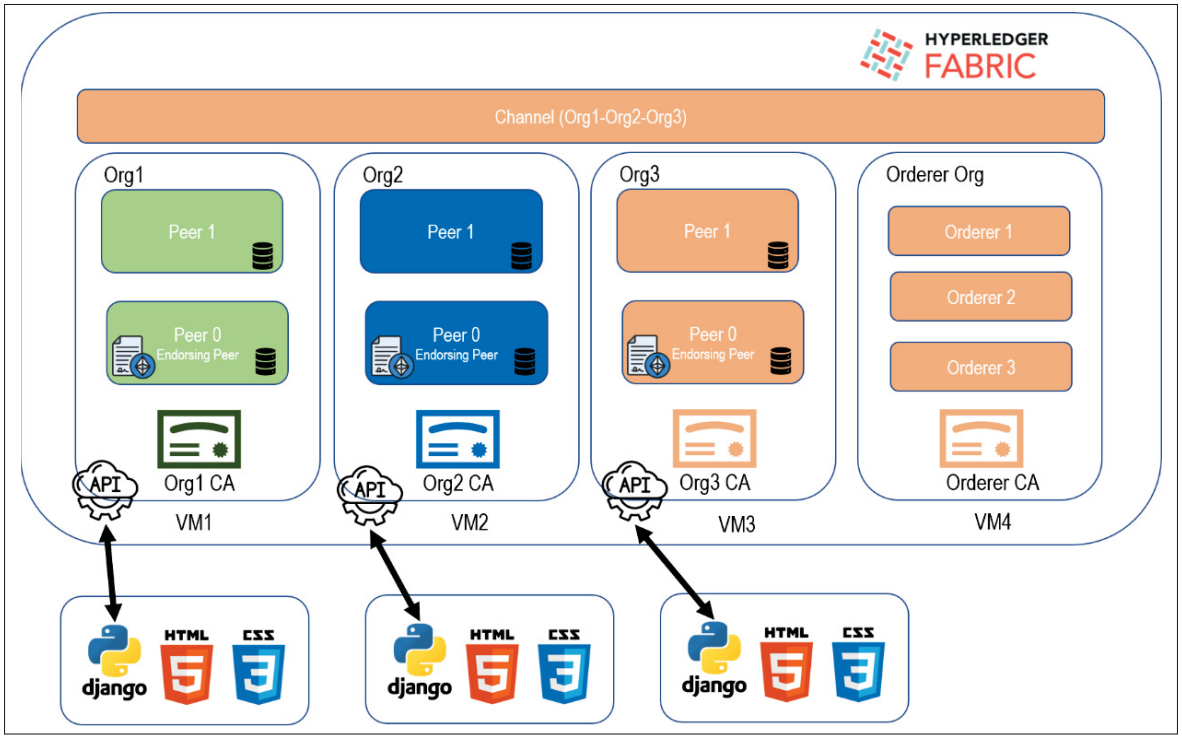

Figure 6.1 Aperçu de la topologie

Sur la base de cette configuration, lorsqu'un utilisateur souhaite exécuter une fonction de contrat intelligent, le *back-end* envoie une requête au service API qui convertit cette demande en une transaction qui sera signée et envoyée sur la chaine de blocs. Ces transactions interagissent avec le contrat écrit en JavaScript, compilé et déployé sur HLF. Chacun des acteurs (citoyen, agents RQ et SAAQ) interagit avec le système par le biais d'une interface web ressemblant aux systèmes de gestion contemporains. Les six (6) étapes de mise en œuvre sont les suivantes :

1. **Création des matériaux cryptographiques :** La génération d'identité numérique se fait par des certificats d'autorité dédiés pour les membres de chaque organisation. La première étape est donc la création des matériaux cryptographiques en créant les instances CA de toutes les organisations ;
2. **Génération d'une identité numérique :** Chaque entité dans le réseau de la chaine de blocs privée (nœuds d'approbation, services de commande) doit avoir une identité numérique pour une communication sécurisée entre les entités. Pour cette raison, un certificat numérique X.509 est créé, qui est un modèle d'infrastructure à clé publique (PKI). Pour chaque organisation, les participants sont créés en utilisant leur propre autorité de certification ;
3. **Construction du réseau de la chaine de blocs :** En suivant la topologie présentée dans la figure 6-1, le réseau est créé et simulé par la virtualisation au niveau du système d'exploitation en utilisant Docker. Le réseau construit est composé de trois organisations Org1, Org2 et Org3 regroupé autour d'un canal, et d'une organisation Orderer. Chaque composant s'exécute sous la forme d'un conteneur Docker. Chaque organisation exécute sa propre autorité de certification (CA). Les nœuds, les bases de données d'état, les nœuds de commande et l'autorité de certification de chaque organisation fonctionnent chacun dans leur propre conteneur Docker. Les trois organisations du réseau permettent d'interagir avec un registre de la chaine de blocs comme trois organisations qui exploitent des nœuds séparés ;
4. **Installation des contrats intelligents :** Implémenté en utilisant le langage JavaScript et installé dans chaque nœud d'approbation, le contrat intelligent se place au centre des interactions entre les différents participants : l'utilisateur (Citoyen), le fournisseur de service (RQ) et le service partenaire (SAAQ). La restriction du contrôle d'accès et de l'utilisation des données est spécifiée dans le contrat intelligent afin que l'auteur soit toujours en possession du document qu'il a initié ;
5. **Construction des services API :** Afin de rendre possible une interconnexion entre les applications clientes et la chaine de blocs, des services API sont créés en utilisant le SDK de HLF écrit en Node.js qui permet aux applications d'interagir avec un réseau HLF. Il

fournit une API simple pour soumettre des transactions ou interroger le contenu du registre HLF avec un code minimal. Le service utilise ExpressJS pour implémenter une API REST ;

6. **Construction des prototypes d'application utilisateur :** La construction des prototypes s'est faite avec Django, un cadriciel Web écrit en python et ayant pour but de rendre le développement d'applications web simple et basé sur la réutilisation de code. Ces applications clientes interagissent avec la chaine de blocs en passant par le service API.

6.2 Évaluation du système

Cette section présente l'évaluation du système basé sur la façon dont l'architecture développée répond aux exigences des parties prenantes, suivies de l'évaluation basée sur la preuve de concept développé en utilisant un scénario reprenant l'interaction avec le système.

6.2.1 Évaluation ATAM basée sur l'architecture

L'évaluation ATAM de l'architecture proposée a été réalisée en deux phases. La première phase a été organisée avec les coordonnateurs du projet. : Un groupe de cinq personnes a été constitué à cet effet, comprenant (i) un architecte de solutions de systèmes d'information (ii) un personnel informatique avec une expérience dans les cas d'utilisation de la chaine de blocs (iii) un chef de projet coordonnateur recherche et développement (v) une personne non experte des technologies de l'information et (iv) un professeur associé expert en technologie de chaine de blocs. La deuxième phase a impliqué dix participants supplémentaires comme partie prenante pour vérifier les résultats de la première phase lors d'une séance de démonstration. Le tableau 6.1 donne un aperçu des activités effectuées.

L'objectif de l'arborescence d'utilité est d'identifier, de hiérarchiser et d'affiner les objectifs d'attributs de qualité les plus importants comme le montre la figure 6.2. Les nœuds de haut niveau représentent les attributs de qualité les plus importants qui, dans ce cas, sont : l'autonomie, la collaboration, l'adéquation fonctionnelle. En outre, les feuilles de cet arbre représentent les

scénarios permettant d'atteindre les attributs de qualité respectifs. Le tableau 6.2 présente les scénarios spécifiques élaborés pour évaluer ces attributs ;

Tableau 6.1 Activités de l'évaluation ATAM

Activité	**Description**
Présentation de ATAM	Un aperçu du processus, des techniques et des résultats attendus d'ATAM a été présenté au groupe de coordonnateurs. Les techniques utilisées sont les suivantes : La génération d'un arbre d'utilité ; l'approche architecturale ; l'élicitation ou analyse ; et ; le mappage de scénarios. Les résultats d'ATAM sont les suivants : • Scénarios obtenus et classés par ordre de priorité ; • Les questions utilisées pour comprendre/évaluer l'architecture afin de produire un arbre d'utilité ; • Arbre d'utilité décrivant et hiérarchisant les exigences architecturales critiques ; • Ensemble des risques et des non-risques découverts ; • Ensemble des points de sensibilité et des compromis découverts.
Présentation des objectifs de conception	Une description des divers objectifs de conception telle que mentionnée au point 3.1.2 a été présentée aux participants afin de leur permettre de mieux comprendre le système.
Présentation de l'architecture	Le système a été présenté aux participants en utilisant le modèle de vue architecturale 4 + 1 de Krutchen, qui prescrit une vue logique, une vue de développement, une vue de processus, une vue physique, ainsi que des descriptions de scénarios. • La vue logique est illustrée par un diagramme de classe UML ; • La vue du processus est illustrée par un diagramme d'activité UML ; • La vue développement est illustrée par un diagramme de composants UML ; • La vue physique est illustrée par un diagramme de déploiement UML.

Table 6.1 Activités de l'évaluation ATAM (suite)

Activité	**Description**
Identification des approches architecturales	Les approches architecturales qui ont été adoptées et les raisons de leur adoption ont été expliquées. Le résumé des objectifs de conception et de l'approche/composant qui cherche à les traiter est présenté comme suit : • Portefeuille d'identification : Service centré sur le citoyen, Sécurité ; • Interaction humaine : Service centré sur le citoyen ; • Canal : Confidentialité ; • Base des données d'État : Historicité ; • Politique d'approbation : Gouvernance ; • Cycle de vie du contrat intelligent : Gouvernance ; • Contrat intelligent : Automatisation, Historicité ; • Données sur chaine ou hors chaine : Collaboration ; • Intégration avec l'existant : Interopérabilité ; • Chaine de blocs : Sécurité ; • Cryptage à clé publique/privée : Authentification, Sécurité ; • Immutabilité : Sécurité ; • Registre distribué : Traçabilité, Historicité, Collaboration ; • API REST : Fiabilité, sécurité, performance.
Génération de l'arbre utilitaire des attributs de qualité	L'arborescence d'utilité des attributs de qualité, présentée sur la figure 6.2, a été générée à partir des questions posées par les participants dans leur tentative de mieux comprendre la conception et le processus de l'architecture, ainsi que des objectifs de conception et de l'élicitation des scénarios. L'arborescence d'utilité a permis de classer par ordre de priorité les exigences spécifiques des attributs de qualité, en fonction des scénarios.
Analyse des approches architecturales	La liste de scénarios classés par ordre de priorité, qui est le résultat de l'étape précédente, est utilisée pour sonder les approches architecturales permettant de réaliser les attributs de qualité importants. Cette étape a été utilisée pour identifier les risques, les points sensibles et les compromis dans l'architecture de conception.
Démonstration	Les scénarios ont été mis en correspondance avec l'interaction du système et les approches architecturales pour une démonstration fonctionnelle du prototype réalisé.
Présentation des résultats	C'est à cette étape que les résultats de l'évaluation sont présentés. Les tableaux 6.3 et 6.4 présentent un ensemble de risques et de non-risques ainsi qu'un ensemble de points de sensibilité et de compromis identifiés comme résultats de l'évaluation ATAM.

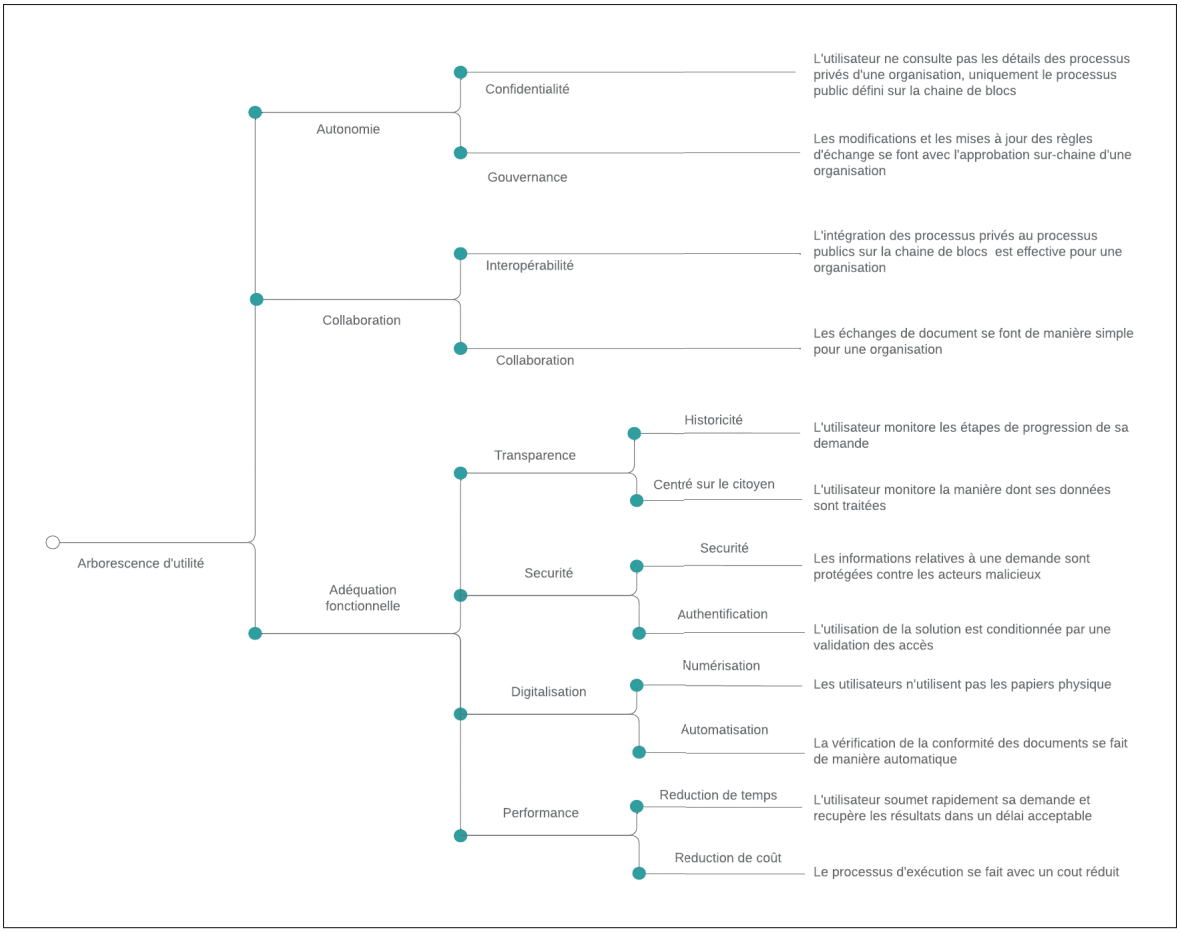

Figure 6.2 Arborescence d'utilité

6.2.2 Observation basée sur les scénarios

En se conformant à notre conception, les organisations exécutives, Revenu Québec (fournisseur de service) et la SAAQ (service partenaire), se réunissent et créent de nouveaux processus à intégrer à leurs processus existants. Ces nouveaux processus sont communs à tous les participants et liés aux processus privés internes de chaque organisation.

Afin de dégager les points attendus par ATAM, une exploration de l'interaction du système présenté sur la figure 6.3 permet d'épingler les scénarios définis et d'identifier les risques potentiels, les non-risques, les points de sensibilité et les points de compromis.

Tableau 6.2 Attributs et scénarios ATAM

Attributs	**Scénarios**
Autonomie	SC1 : L'utilisateur ne consulte pas les détails des processus privés d'une organisation, uniquement le processus public défini sur la chaine de blocs
	SC2 : Les modifications et les mises à jour des règles d'échange se font avec l'approbation sur chaine d'une organisation
Collaboration	SC3 : L'intégration des processus privés au processus public sur la chaine de blocs est effective pour une organisation
	SC4 : Les échanges de document se font de manière simple pour une organisation
Adéquation fonctionnelle	SC5 : L'utilisateur surveille les étapes de progression de sa demande
	SC6 : L'utilisateur surveille la manière dont ses données sont traitées
	SC7 : Les informations relatives à une demande sont protégées contre les acteurs malicieux
	SC8 : L'utilisation de la solution est conditionnée par une validation des accès
	SC9 : Les utilisateurs n'utilisent pas les papiers physiques
	SC10 : La vérification de la conformité des documents se fait de manière automatique
	SC11 : L'utilisateur soumet rapidement sa demande et récupère les résultats dans un délai acceptable
	SC12 : Le processus d'exécution se fait avec un cout réduit

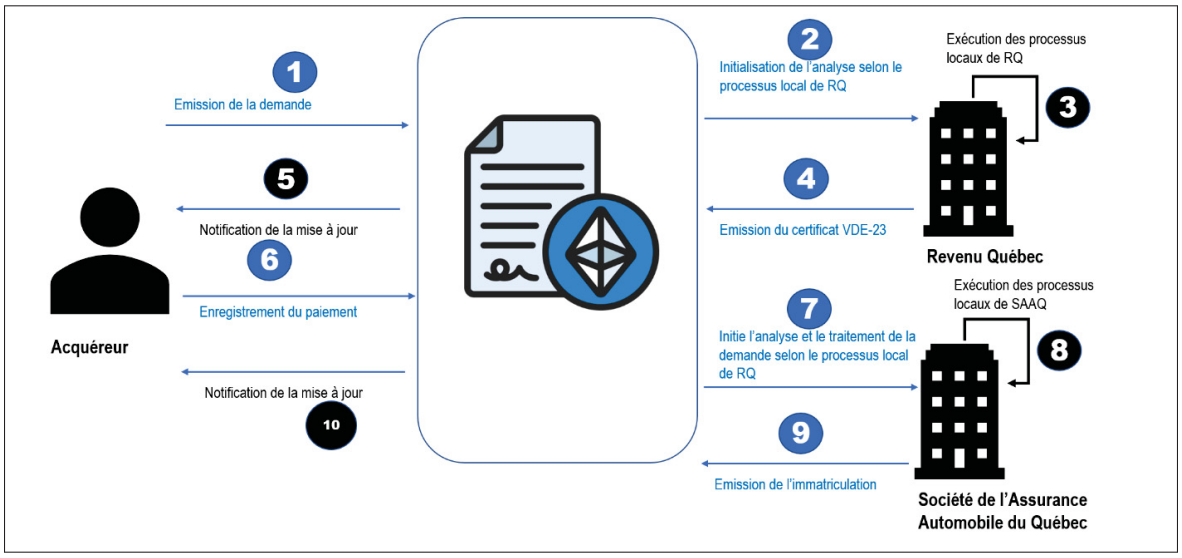

Figure 6.3 Interaction du système

Le processus public est donc écrit sur un contrat intelligent pour rendre possible le flux de travaux repris dans la figure 4.4. Au préalable, chaque participant dans le réseau se voit fournir une identité numérique via un certificat X.509 de son organisation pour une communication sécurisée entre les entités participantes (SC8) et les processus se déroulent de la manière suivante :

1. Le citoyen crée et émet la demande. La demande est enregistrée sur la chaine de blocs à l'aide du contrat intelligent. Cette étape suppose que tous les documents devant être joints sont disponibles sous forme numérique ou peut être numérisée avant cette étape (SC9). Pour garantir la vérifiabilité, toutes les informations de chaque document sont stockées de manière permanente et inviolable dans la chaine de blocs (SC5). De plus, en fournissant les documents sous forme numérique, la vérification simultanée des documents par les organisations participantes est rendue possible (SC4) ;
2. L'enregistrement de la demande sur le contrat intelligent initie un évènement permettant à l'organisation fournisseur de service d'initier son processus local (SC3). Le processus de vérification est divisé en deux parties. D'une part, les conditions qui peuvent être vérifiées automatiquement sont renseignées à l'aide de contrats intelligents (SC10). Par exemple, il peut être spécifié, dans le cas où la demande est incomplète, que l'envoi des documents doit être réalisé avant une certaine date. Ainsi, un contrat intelligent vérifie si tous les documents ont été déposés avant cette date. D'autre part, il existe des conditions qui ne peuvent pas être mises en œuvre pour un contrôle automatisé, car elles nécessitent une expérience humaine, par exemple les restrictions tarifaires. Ces conditions doivent être vérifiées manuellement et sont reléguées au traitement privé d'une organisation ;
3. Le fournisseur de service effectue le traitement local en fonction de son flux de travaux privé. Ces processus internes ne sont pas visibles pour les autres (SC1) et sont indépendants par rapport aux changements internes des processus des autres organisations ;
4. La fin de l'exécution du processus local rend possible l'enregistrement du document émis par le fournisseur de service sur la chaine de blocs et met à jour le statut de la demande en utilisant les points d'accès API (SC3). Dans le cas contraire, le citoyen doit soumettre à nouveau les documents ;

5. La mise à jour du statut de la demande notifie le citoyen que le contrôle est à son niveau, et qu'il doit enregistrer la preuve de paiement de la taxe de vente.
6. Le processus de paiement se passe hors système, le citoyen enregistre sur la chaine de blocs la preuve de paiement effectué ;
7. Similaire à (2), l'enregistrement de la preuve de paiement sur le contrat intelligent initie un évènement permettant à l'organisation partenaire d'initier son processus local via une notification (SC3).
8. Similaire à (3), le service partenaire effectue le traitement local en fonction de son flow de travaux privé (SC1) ;
9. Similaire à (4), la fin de l'exécution du processus local rend possible l'enregistrement du document émis par le service partenaire sur la chaine de blocs et met à jour le statut de la demande (SC3) ; et
10. Le citoyen est notifié de la fin de traitement de sa demande et peut télécharger le document au besoin.

Pour l'ensemble du processus, chaque action concernant l'état du processus est stockée dans la chaine de blocs, c'est-à-dire que chaque participant au processus signe et approuve également la progression du processus à l'aide de sa clé privée (SC8). Cela permet d'avoir un état du processus en temps réel accessible à tous les participants (SC4) et un historique complet du processus (SC5). L'utilisation de la chaine de blocs permet en outre une approche centrée sur le citoyen où celui-ci surveille la manière dont les données sont utilisées (SC6). Quant à la question de la gouvernance (S6), une gouvernance centralisée hors chaine et coercitive est utilisée.

Le cycle de vie du contrat intelligent introduit dans la version 2.2 de HLF conditionne la mise à jour de la logique écrite dans un contrat intelligent à la validation de toutes les organisations participantes. L'utilisation d'un mécanisme de stockage des données hors chaine garantie des meilleurs résultats sur la performance et l'évolutivité du réseau et la numérisation des processus manuels augmente la transparence et potentiellement les délais de traitement (SC11), le citoyen n'étant plus obligé de se déplacer pour faire valider sa requête.

L'interaction avec le système se fait de deux points de vue différents correspondants à nos entités : Citoyen et Organisation. La figure 6.4 présente quelques captures d'écrans de la vue Citoyen et la figure 6.5 présente celles de la vue Organisation. Les fonctionnalités développées dépendent ainsi des interactions d'une entité donnée avec la solution développée sur notre architecture.

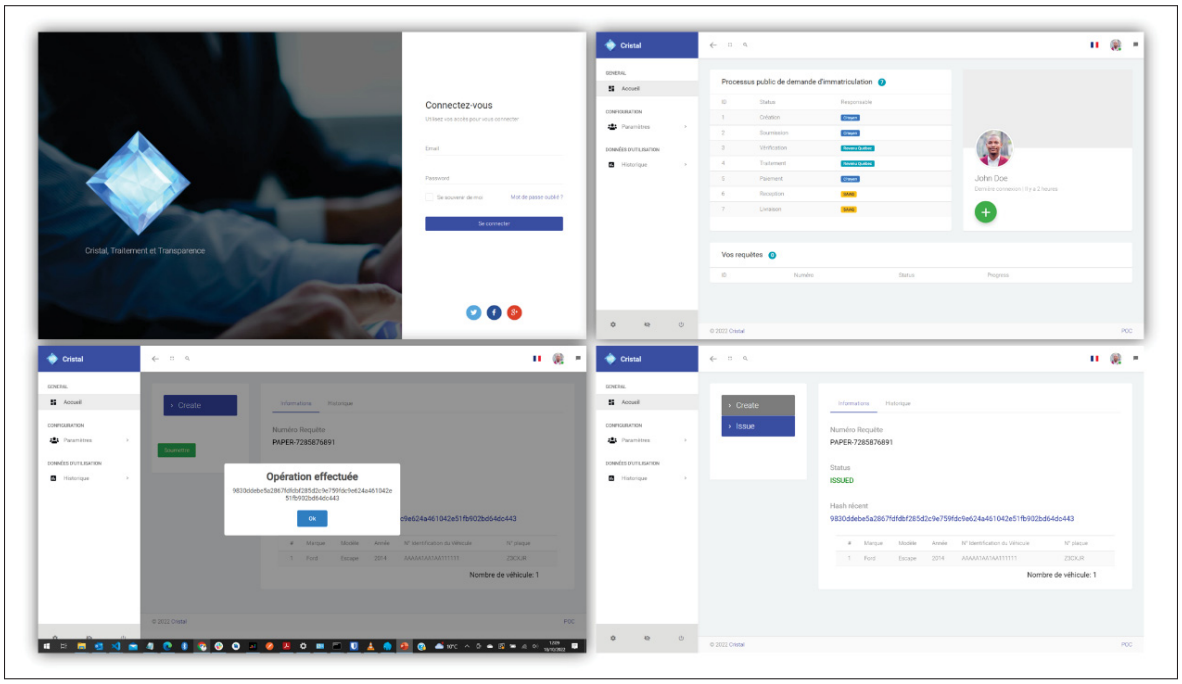

Figure 6.4 Vue de l'application côté Citoyen

6.2.3 Performance du réseau

Les performances de la plateforme HLF sont évaluées en termes de débit, de latence et d'évolutivité. Afin d'identifier les capacités du réseau, deux principales mesures de performance ont été étudié sur les opérations d'écriture (*createVdx*) et de lecture (*readVdx*) dans le registre en mesurant le débit qui représente les transactions réussies par seconde et la latence qui représente le temps de réponse par transaction en secondes. Hyperledger Caliper a été utilisé pour mettre en place l'outil de test du référencé dont la configuration a été définie pour alterner les taux,

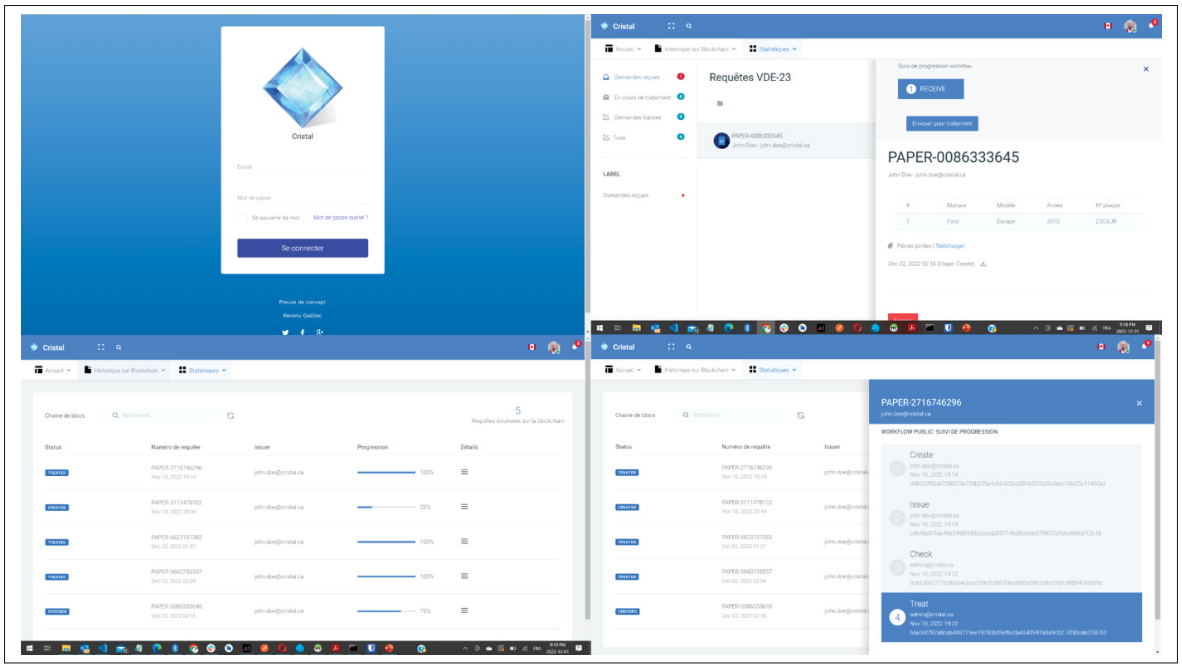

Figure 6.5 Vue de l'application côté Organisation

nombres et types de transactions. Le taux linéaire a été utilisé comme configuration afin de trouver les taux de charge de travail qui affectent la performance du système.

Dans l'environnement de test de notre configuration expérimentale, le réseau pouvait supporter jusqu'à 100 tps. La figure 6.6 montre une relation inversement proportionnelle entre la latence et le débit. Pour les opérations de lecture, les performances sont similaires même pour un grand nombre de transactions : 0,1 seconde de latence en moyenne et une moyenne de 100 transactions/seconde. En revanche, pour les opérations d'écriture, au-delà de 5000 transactions, la latence devient significative. Ces résultats peuvent varier en fonction des environnements de test mais permettent de préparer une configuration matérielle appropriée pour une mise en œuvre à grande échelle.

Une autre information importante est la bande passante qui est une mesure de la capacité de transmission de données dans un réseau. La figure 6.7 montre la manière dont la bande passante est utilisée en fonction du nombre de transactions. Cette information est obtenue à partir des métriques *Traffic IN* et *Traffic OUT* obtenus des requêtes *createVdx* sur des nœuds spécifiques

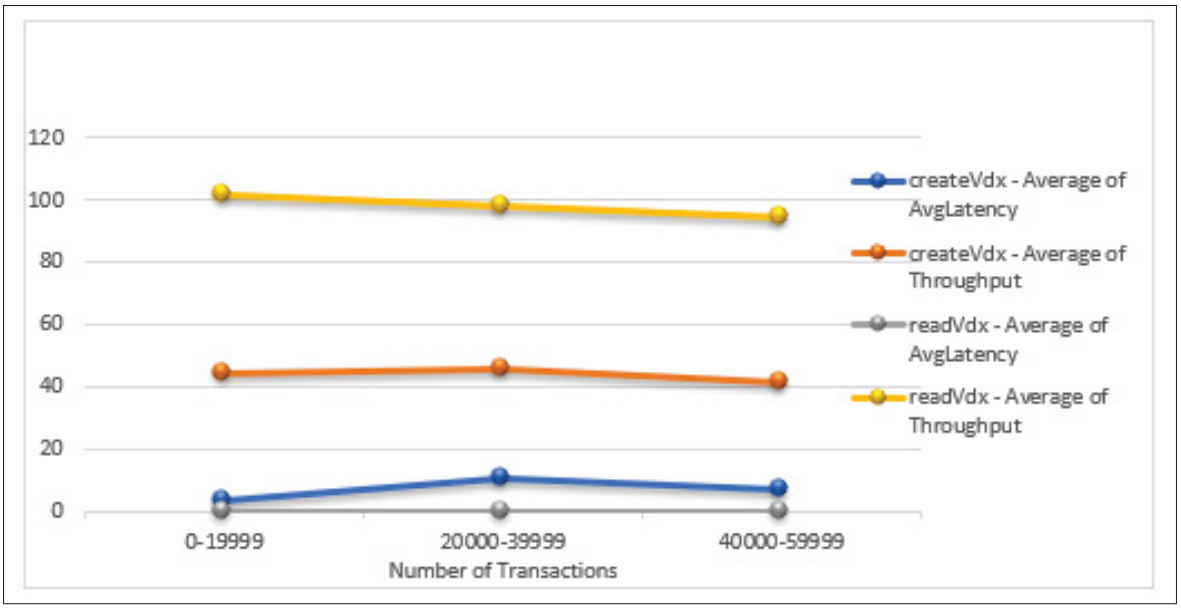

Figure 6.6 Moyenne de la latence et du débit par rapport au nombre de transactions

du réseau. Les moments où la bande passante est fortement utilisée indiquent des moments de congestion en fonction du nombre de transaction. Il en ressort que le nœud Orderer est celui qui utilise le plus la bande passante car son rôle nécessite la transmission de grandes quantités de données pour garantir la cohérence et la sécurité du réseau. En moyenne les autres nœuds tournent autour 0,7 Mégaoctets/seconde tandis que le nœud Orderer a une moyenne 1,519 Mégaoctets/seconde.

6.2.4 Coût

Les machines virtuelles créées pour la preuve de concept sur Microsoft Azure ont fait l'objet d'un monitoring sur le coût associé à leurs fonctionnements. Les quatre machines ont été créées suivant la spécification Standard_B8ms et ont eu un coût moyen équivalent à 12,5$/ jour, tel que présenté sur la figure 6.8. Ces résultats peuvent varier en fonction des spécifications de la machine créée, mais permettent d'avoir une estimation financière pour une mise en œuvre à grande échelle.

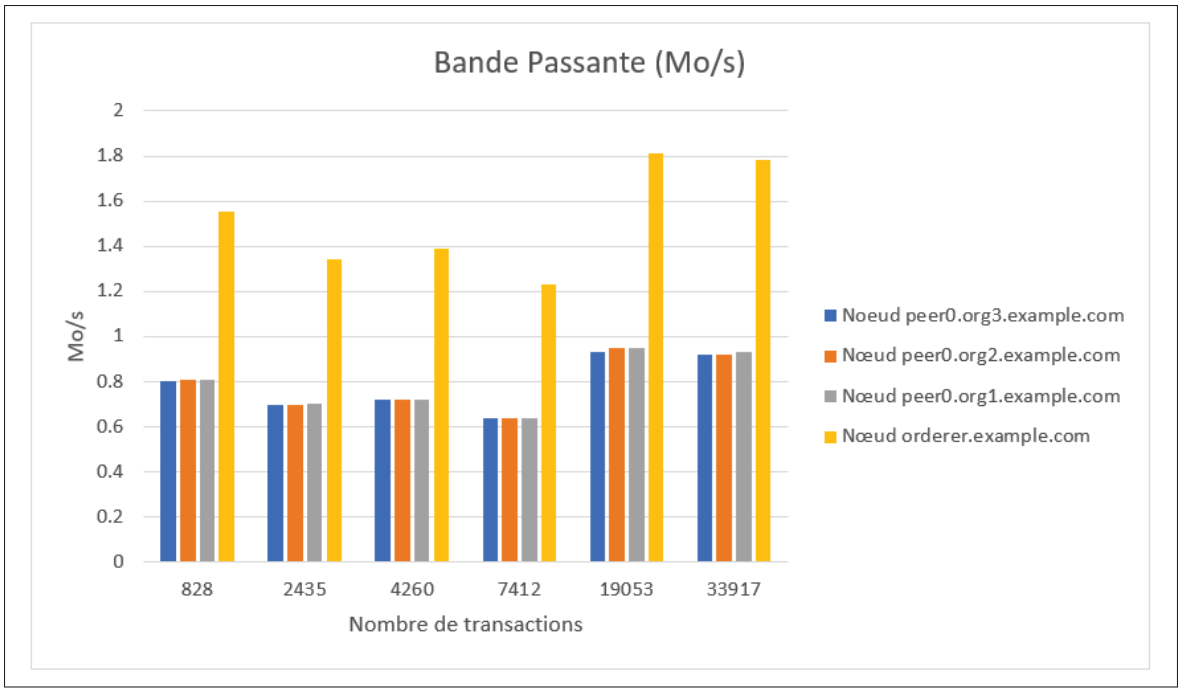

Figure 6.7 Bande passante du réseau

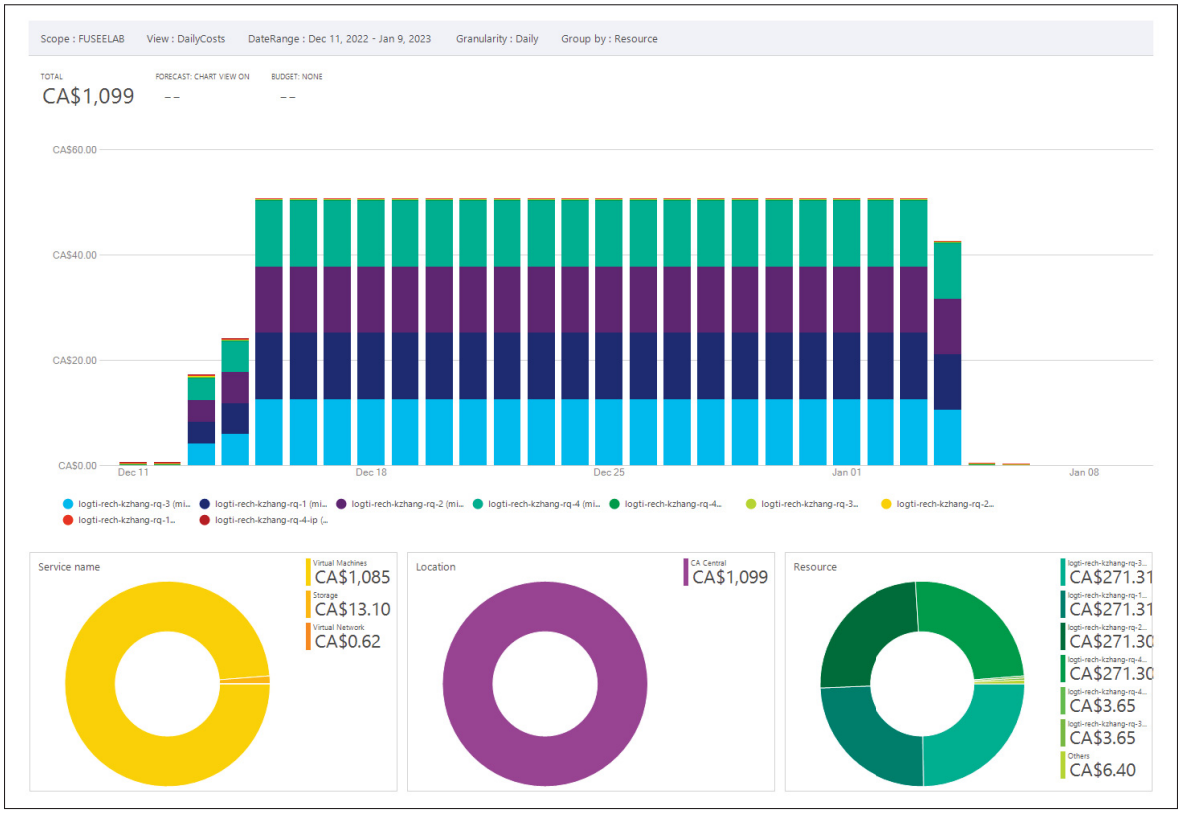

Figure 6.8 Évaluation de coût

6.2.5 Résultats de l'ATAM

À la suite des activités décrites ci-dessus, un ensemble de risques, de non-risques, de points de sensibilité et de points de compromis a été généré. Un ensemble de points de sensibilité et de compromis fait référence aux décisions architecturales qui ont un effet marqué sur un ou plusieurs attributs de qualité. Un risque est défini dans l'ATAM comme une décision architecturale qui peut entraîner des conséquences indésirables à la lumière des exigences d'attributs de qualité énoncées. Un non-risque fait référence à une décision architecturale qui, après analyse, est considérée comme sûre. Les tableaux 6.3 et 6.4 présentent ces résultats.

6.3 Discussion des résultats et recommandations

En fonction des résultats obtenus avec ATAM, un ensemble de recommandations a été proposé afin de favoriser l'utilisation de l'architecture proposée pour atteindre les objectifs de conception établis dans cette recherche. Les risques identifiés peuvent constituer la base d'un plan d'atténuation des risques architecturaux. Ces recommandations se concentrent sur des composants spécifiques de l'architecture et peuvent être utilisées comme des guides de bonnes pratiques pour mettre en œuvre l'architecture proposée.

- Une analyse de la sécurité de HLF et une présentation des menaces possibles aux différentes couches du cas d'étude afin d'étudier la manière dont les fonctionnalités intégrées de HFL peuvent les atténuer ;
- Une organisation centralisée du processus d'authentification, tel que proposé sur la figure 4-10 afin d'établir une uniformité entre tous les services publics dans le cas où cette architecture devrait être dupliquée dans plusieurs cas d'utilisation ;
- Une conception « multicanal » permettant d'exécuter plusieurs cas d'utilisation autour des organisations sur le même réseau dans le but d'effectuer des transactions privées et confidentielles ;

Tableau 6.3 Risques et non-risques de l'architecture proposée

Risque	**Non-risque**
• IPFS est public et est généralement utilisé avec une chaine de blocs publique. Les trafics et le contenu sont publics. Il est donc un risque par rapport à la confidentialité même si les données peuvent être cryptées ; • L'utilisation d'un service API présente le risque de sécurité si le service peut être amené à exécuter des commandes imprévues ; • L'utilisation des points d'intégration est un risque si la transformation des données entre la chaine de blocs et les formats des systèmes existants n'est pas standardisée ; • Un canal global unique est un risque si d'autres organisations ne faisant pas partie du cas d'utilisation rejoignent le canal ; • La politique d'approbation utilisée suppose que tous les participants du canal doivent endosser une transaction, c'est un risque sur l'évolutivité et les performances dans le cas où le réseau dispose de plusieurs participants ; • L'implémentation de la logique de décision dans le contrat intelligent est un risque si ce dernier fait l'objet des mises à jour fréquentes.	• L'utilisation d'une chaine de blocs est un non-risque pour l'autonomie étant donné qu'elle est utilisée pour exécuter uniquement un processus public n'interférant pas avec les processus privés des organisations ; • L'authentification avec le certificat X.509 géré par le certificat d'autorité par défaut de HLF est un non-risque pour l'authentification ; • Le fonctionnement automatisé des différents composants de l'architecture est un non-risque en ce qui concerne une intervention humaine minimale ; • La séparation de la conception basée sur l'approche publique-privée afin d'améliorer la flexibilité de l'architecture n'est pas un risque ; • Le fait de se passer de l'utilisation d'une liste de contrôle d'accès n'est pas un risque si les vérifications nécessaires sont effectuées au niveau de la couche d'interaction ; • L'utilisation des mécanismes de sécurité proposée mis en œuvre par HLF n'est pas un risque si une analyse de la sécurité par rapport au cas d'utilisation est menée.

- Un déploiement du réseau HLF à hôtes multiples pour garantir la décentralisation et la transparence, chaque organisation déployant ses propres machines virtuelles pour organiser ses composants ;

Tableau 6.4 Points de sensibilité et de compromis de l'architecture proposée

Risque	**Non-risque**
• L'utilisation d'une chaine de blocs privée garantit la sécurité, la transparence, la collaboration et permet une approche centrée sur le citoyen ; • L'utilisation d'un modèle de stockage hors chaine influe positivement sur la collaboration ; • L'utilisation d'un service API implémentant la connexion avec le contrat intelligent garantit la communication avec le système privé et assure l'autonomie des organisations ; • L'utilisation de CouchDB comme base des données d'état a des impacts sur la performance dans la fourniture d'un support de requête riche et une amélioration de sécurité pour la conformité et la protection des données ; • Le déploiement à hôtes multiples est une garantie de la décentralisation et de la transparence ; • L'utilisation de la V2 de HLF rend possible la définition d'un mécanisme de gouvernance permettant aux organisations de se mettre d'accord sur la manière sur les mises à jour.	• Une chaine de blocs définit un mode de fonctionnement qui va à l'encontre de l'autonomie des organisations ; • IPFS est un réseau externe, il ramène de la charge de travail pour l'intégrabilité ; • La participation directe des citoyens est un point de compromis entre la transparence et la performance ; • L'encapsulation de la logique de décision dans le code du contrat intelligent facilite sa gestion, mais limite la flexibilité au niveau de sa mise à jour ; • L'utilisation d'un canal ramène la confidentialité des transactions et des données et oblige un effort supplémentaire pour la collaboration ; • L'implémentation de la logique de décision dans le contrat intelligent réduit le cout global du projet, mais empêche une gouvernance flexible.

- Une uniformisation des formats d'échange et une élaboration des normes interservices entre la chaine de blocs et les systèmes de gestion privée des organisations afin d'étendre la mise en œuvre au sein des services ayant des structures similaires ;
- Une externalisation de la logique de décision vers un moteur de règles externe plutôt qu'un codage de la logique de décision dans le code du contrat intelligent. L'externalisation de la

logique offrira aux organisations la flexibilité de modifier la logique de décision sans avoir à modifier le code du contrat intelligent ;
- Une solution de stockage décentralisée avec un réseau privé IPFS interfacée avec un client HLF pour l'échange de données ;
- Un hachage des données avant d'appeler un contrat intelligent et la définition d'un mécanisme pour partager les données sources ; et
- Une utilisation des listes de contrôle d'accès pour restreindre l'accès aux données à certains rôles dans la logique du contrat intelligent.

En prenant en compte les recommandations, l'architecture proposée rend possible l'interopérabilité entre le flux de travaux local et public dans les processus interorganisationnels dans le secteur public et est axée sur la transparence et la confidentialité pour un déploiement efficace. L'utilisation de la chaine de blocs comme solution décentralisée augmente les niveaux de confiance, de transparence et d'accessibilité des services et permet une approche centrée sur le citoyen, dans laquelle le citoyen peut surveiller la façon dont ses données sont utilisées. La technologie de la chaine de blocs est générique et admet de nombreuses conceptions de systèmes différents, de sorte que le problème aurait pu être mis en œuvre de différentes manières. En optant pour l'architecture proposée, les objectifs de conception mentionnés dans le tableau 3.2 sont pris en compte, comme le résume le tableau 6.5.

Cependant, les objectifs de conception portant sur la réduction du temps global du processus (DG7) et la réduction des coûts (DG8) n'ont pas pu être totalement évalués et sont considérés comme un défi ouvert. La précision sur la réduction de cout d'exécution du processus est tributaire d'une estimation du cout actuel de l'ensemble de processus d'immatriculation et la réduction du temps global du processus est tributaire de la numérisation des processus locaux. Ces deux éléments ne sont pas couverts dans cette recherche.

Tableau 6.5 Tableau d'évaluation

	Objectif de conception	**Description**
DG1	Processus basé sur la numérisation	Toutes les étapes pertinentes du processus peuvent être réalisées sous forme numérique.
DG2	Automatiser les contrôles manuels	Un contrat intelligent permet à la procédure publique d'être menée automatiquement.
DG3	Fournir un historique du processus	En utilisant HLF, toutes les transactions historiques dans le grand livre de la chaine de blocs peuvent être récupérées ainsi que l'état actuel d'un document spécifique dans la base de données mondiale.
DG4	Authentification	HLF est un réseau à autorisation, il dispose d'un mécanisme qui permet aux participants de prouver leur identité au reste du réseau afin d'y effectuer des transactions.
DG5	Sécurité	La sécurité dépend amplement de HLF, l'infrastructure de la chaine de blocs utilisée. De plus, l'organisation des échanges autour d'un canal assure la confidentialité des données et des transactions.
DG8	Services centrés sur le citoyen	L'approche publique-privée utilisée permet aux citoyens d'avoir accès aux données et de contrôler le traitement qui en est fait.
DG9	Interopérabilité	L'approche publique-privée permet la mise en place d'un système qui assure une meilleure coordination et un meilleur suivi des processus publics tout en laissant aux organisations l'autonomie et la confidentialité de leurs processus locaux. En utilisation le SDK de HLF dans notre couche d'interaction, l'intégration avec les systèmes privés de gestion des processus de chaque organisation est rendue possible en assurant l'interopérabilité des données.
DG10	Collaboration	Le registre partagé permet de partager les données en temps réel pour tous les participants de manière transparente.
DG11	Confidentialité	Le processus interne d'une organisation n'est pas visible pour les autres et est indépendant par rapport aux changements internes des processus des autres.
DG12	Gouvernance	Une gouvernance centralisée hors chaine et coercitive scelle l'accord entre les participants et définit un processus de gouvernance clairement défini, publié et accessible.

CHAPITRE 7

CONCLUSION

L'objectif principal de cette recherche était de concevoir une architecture de référence basée sur la chaine de blocs pour les services publics centrés sur le citoyen. Le système devait permettre d'intégrer la chaine de blocs dans le service public dans une approche centrée sur le citoyen en relevant les défis relatifs à la coordination des organisations.

Les principaux défis rattachés à cet objectif, et présentés comme sous-objectifs spécifiques de cette recherche, étaient la proposition d'une architecture assurant la coordination des organisations dans une approche centrée sur le citoyen, l'évaluation du degré auquel la proposition satisfait les objectifs de conception et le développement d'un prototype fonctionnel qui implémente l'architecture proposée. Pour atteindre cet objectif, trois questions de recherche ont été formulées dans la section introduction de ce mémoire :

1. Quelles sont les exigences d'un système intégrant la chaine de blocs pour le traitement des services publics centrés sur le citoyen ?

   Pour répondre à cette question, une revue de la littérature a été menée sur la base des défis relatifs à la numérisation de services publics en général et des processus interorganisationnels en particulier, ainsi que sur la base des considérations de conception clé d'un système basé sur la chaine de blocs. Ensuite, une analyse de l'étude de cas a été utilisée pour extraire les exigences du système ainsi que les objectifs de conception, alignés sur les défis relatifs à la coordination des organisations, permettant de mesurer le succès de l'expérimentation.
2. Quelle est l'architecture de référence d'un système basé sur la chaine de blocs pour un service public centré sur le citoyen ?

   L'architecture de référence est basée sur les exigences du système et a été conçue de manière à s'adapter aux objectifs de conception définis. Une analyse des plateformes de chaine de blocs populaires a été menée afin de choisir la plateforme adéquate et le modèle de vue de Krutchen a permis de décrire les entités, les modèles et les couches du système et de définir leur fonctionnalité et les relations entre eux.

3. Comment l'architecture du système peut-elle être mise en œuvre dans un prototype ?
   Pour répondre à cette question, une preuve de concept a été développée sur la base de l'architecture proposée en utilisant la technologie de la chaine de blocs. Elle a été testée en utilisant l'étude de cas sélectionnée et évaluée en utilisant la méthode ATAM.

L'architecture proposée est basée sur la plateforme de chaine de blocs privée HLF à laquelle les organisations participantes à un processus interorganisationnel peuvent se connecter. Afin de se conformer aux exigences identifiées, cette architecture s'organise autour de trois couches : une couche réseau qui organise les organisations participantes autour d'une chaine de blocs ; une couche application organisant les applications spécifiques de chaque organisation et une couche interaction fournissant les interfaces chargées de permettre la communication entre la couche réseau et la couche application en utilisant un contrat intelligent exécutant un processus public partagé entre tous les participants du réseau. L'architecture a été évaluée en utilisant la méthode d'évaluation basée sur le scénario ATAM qui est une procédure standard d'évaluation préliminaire des architectures permettant la participation des parties prenantes. Les résultats ont montré que le système développé satisfait aux objectifs de conception.

La contribution de ce mémoire s'aligne sur les sous-objectifs spécifiques et comprend une architecture à trois couches intégrant la chaine de blocs et les composants autour d'une couche réseau, d'une couche interaction et d'une couche application suivant une approche conceptuelle rendant possible l'application de la chaine de blocs dans le service public centré sur le citoyen, une évaluation de l'architecture proposée en utilisant la méthode d'évaluation d'architecture basée sur les scénarios ATAM et une implémentation basée sur l'architecture de référence proposée d'une étude de cas réel retraçant le processus d'immatriculation de véhicule à cheval entre Revenu Québec et la Société de l'Assurance Automobile du Québec.

L'approche choisie a permis de rationnaliser le processus d'immatriculation de véhicule, améliorant le parcours du citoyen en diminuant les déplacements physiques et le temps de passage dans les bureaux et en réduisant les points de contact avec les différentes organisations sur une seule plateforme.

Bien que les mesures de succès, faisant référence aux objectifs de conception alignés sur les défis relatifs à la coordination des organisations, aient été remplies, le système proposé est perfectible. L'évaluation ATAM a permis de proposer, dans la discussion des résultats, des recommandations identifiant les limites qui peuvent être utilisées comme des guides de bonnes pratiques et des points d'amélioration pour la mise en œuvre de l'architecture proposée.

De plus, la conception de ce modèle peut impliquer la connaissance et le partage des connaissances organisationnelles sur les processus locaux de chaque organisation, la réingénierie des processus pour modifier et intégrer ces processus. Cette recherche ne couvre pas la restructuration nécessaire des organisations et des nouveaux processus. Elle n'examine pas non plus les aspects économiques de la mise en œuvre de l'écosystème technologique proposé. En outre, il existe des limites dans la mesure où, pour une meilleure praticabilité, les processus locaux doivent être préalablement numérisés. L'autre limite est la construction d'un composant intégré pour reconnaître automatiquement les structures de données afin de résoudre le problème de l'interopérabilité de manière automatique et permettre ainsi une infrastructure commune standardisée au niveau de la couche interaction. Ce mémoire ne présente pas les menaces possibles au niveau des différentes couches de l'architecture et la manière dont HLF permet de les atténuer. Des travaux futurs dans cette direction permettrait de définir, de manière précise, les mesures de sécurité supplémentaires qui seraient nécessaires pour sécuriser le système.

Le choix de la plateforme HLF était un compromis entre le niveau de configurabilité souhaité et le coût associé au projet. La preuve de concept a montré qu'il était possible de mettre en œuvre avec succès l'architecture proposée en utilisant HLF. La mise en œuvre de l'architecture proposée avec une plateforme de chaine de blocs différente, comme Enterprise Ethereum, et la comparaison des résultats avec la mise en œuvre HLF décrite ici pourraient être instructives. Cela pourrait définitivement montrer quelle plateforme est plus adaptée au développement d'applications pour le secteur public pour ce cas d'utilisation.

Bien que cette recherche ait montré que l'exécution d'un processus interorganisationnel sur la chaine de blocs dans une approche centrée sur le citoyen est possible aux niveaux opérationnel et technique, les aspects administratif et légal soulèvent de nombreuses questions :

- La définition des processus publics : La définition d'un processus public suppose un cadre d'échange administratif où les organisations se mettent d'accord sur le processus public officiel pour un service donné et adaptent leurs processus privés en fonction de cela ;
- Le contrôle des processus publics : Le choix du garant des processus publics est à définir. Ces processus publics peuvent être contrôlés par une agence indépendante tierce ou par le consortium des organisations concernées ;
- Le contrôle des données personnelles par l'utilisateur : le citoyen peut se voir accorder la possibilité de gérer la manière dont ses données sont exploitées sur une chaine de blocs, les termes d'exploitation de cette possibilité doivent être étudiés ;
- La protection des renseignements personnels : L'utilisation d'une technologie de stockage distribuée revient à considérer la question de la protection des renseignements personnels comme une base importante de cette infrastructure technologique.

Ces aspects ne sont pas abordés dans cette recherche et pourraient être des domaines futurs à explorer. Quant à l'évaluation ATAM, l'utilisation d'un plus grand nombre de parties prenantes aurait pu fournir une base plus solide pour des conclusions plus robustes. Toutefois, les résultats obtenus offrent une bonne base d'évaluation pour une architecture de référence à un stade précoce.

Cette recherche reconnaît que l'aspect de la numérisation des processus locaux et celui de la connexion à une plateforme traditionnelle de gestion d'entreprise n'ont pas été étudiés en profondeur, ce qui n'a pas permis de mesurer combien la proposition réduit réellement le temps et le coût total. Fournir une analyse complète de l'ensemble de processus serait un ajout intéressant à cette recherche.

ANNEXE I

TOWARDS CITIZEN-CENTRIC SERVICES USING BLOCKCHAIN-POWERED DIGITALIZATION OF PUBLIC SECTOR PROCESSES

Sion Israel Sion[1] , Kaiwen Zhang[1] , Alain April[1]

[1] Département de génie logiciel et des TI, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article soumis à la conférence « IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2023, Dubai, United Arab Emirates) »

1. Abstract

Public sector institutions operate in a rapidly changing landscape that is transforming the way services are delivered. To ensure positive outcomes for citizens, governments need the support of efficient public institutions with the capacity to keep track with emerging trends and changing expectations. The adoption of new technologies offers innovative opportunities for the public sector and can improve interactions between citizens and institutions and enabling citizen-centric services. The emergence of blockchain technology has inspired a new type of information exchange infrastructure on a peer-to-peer model that makes possible a set of use cases in sectors where improved efficiency, reliability, security and costs around transparency are intended. The public sector is mainly characterized by administrative processes that involve several institutions. If blockchain can promote various types of integration in an organization, the implications around its applicability and its interoperability in the complex integration of an inter-organizational public process remain to be demonstrated.

In this paper, we investigate a real-life use case in public administration related to the vehicle registration process involving multiple institutions to develop an architecture that makes possible a blockchain-based digitalization of the inter-organizational process.

We describe a case study and used it to extract the system requirements, analyze an approach that identifies data interoperability in public sector inter-organizational processes, and propose an

architecture focused on transparency and privacy for effective deployment. We then evaluate the designed architecture using the ATAM method around the scenarios of the vehicle registration, positioning blockchain both as an innovative alternative and as a vehicle for creating new management practices and models in the public sector digitalization.

keyword : Blockchain, Digitalization, Public sector, Interorganizational process, Interoperability, Citizen-centric

2. Introduction

Initially introduced as a public, decentralized, and untrusted ledger for digital currencies, blockchain technology has been widely adopted in many fields (Rosado *et al.*, 2019; Rot *et al.*, 2020). In general, distributed ledgers have the potential to become a new source of information infrastructure promoting the exchange of information between public institutions, citizens and enterprises (Allessie *et al.*, 2019). New technologies have redefined citizens' expectations of the public sector and its services, making digitalization a must. An efficient public institution has to be able to provide citizens and enterprises with efficient services that are accessible to all, thus bridging the gap with concerned people and reducing physical travel and time spent in offices.

This digitalization aims to design digital services focused on citizens and to raise fundamental technological decisions concerning infrastructure, interoperability, platforms, services, security, accessibility, transparency, and many other aspects (Battilani *et al.*, 2022). Blockchain technology can therefore play a more central role in the digitalization of public sector processes, not only in terms of administrative streamlining and the resulting reduction in costs and time, but also because it can significantly improve levels of trust, transparency, security, reliability and accessibility of services (Reply, 2022).

The public sector is mainly characterized by inter-organizational processes that involve several public institutions for a specific purpose but most of these institutions operate in silos, without any collaboration with institutions involved in other activities of the same process (Punia & Saxena, 2004). The integration of various participants and the interoperability with local processes

specific to each institution makes inter-organizational processes very complex (Fridgen *et al.*, 2018). Blockchain is a decentralized transaction and data management technology (Potty Yu, 2020). With its transparency, reliability and distributed nature, blockchain offers potential for managing inter-organizational processes (Evermann & Kim, 2019).

The aim of this paper is to explore the applicability and integrability of blockchain in the execution of inter-organizational processes in a citizen-centric way. In a joint effort with a public institution in country X, we explored the application of blockchain technology in the digitalization of inter-organizational processes on a citizen-centred way, proposed an architecture considering the integration of local processes specific to each organization, and developed a prototype around the scenarios of the vehicle registration process.

In this paper, the main objective is to investigate whether digitalization through blockchain can solve the main challenges of inter-organizational processes in the public sector, mainly the challenge of coordination, transparency of information, confidentiality of local processes, non-citizen-centred approach, and to what extent. The main contributions of this paper are :

1. A design approach allowing blockchain's integrability and applicability in managing the coordination of institutions without affecting their autonomy ;
2. A three-layer architecture in line with the design approach integrating component interoperability around a network layer, an interaction layer, and an application layer ;
3. The development of the proposed architecture on a real-life use case in public administration about the vehicle registration process involving multiple institutions ;
4. An evaluation of the proposed architecture using the ATAM scenario-based architecture evaluation method.

The rest of the paper is organized as follows : Section 2 provides the background about digitalization in public sector and blockchain technology and related work. Section 3 analyzes the vehicle registration case study. Section 4 presents the proposed approach and architecture. The evaluation is conducted in Section 5 whilst discussions and conclusion are presented in Section 6.

3. Background and related work

In this section, we provide a state-of-art relative to opportunities and challenges in the digitalization of the public sector, and the role hold by blockchain then a review of related work is presented.

3.1 Blockchain technology

Blockchain is, basically, a distributed ledger technology that emerged with the cryptocurrency Bitcoin. A distributed ledger is a type of database distributed across multiple sites, regions, or participants. Interest on distributed ledgers increased with the advent of blockchain and its use in financial transactions (El Ioini & Pahl). In a specific way, a blockchain is an immutable ledger, shared over a distributed network among multiple nodes. Each node has a copy of the ledger. To ensure trust between nodes, a consensus mechanism is put in place to validate transactions and ensure that the consensus is complete (Androulaki *et al.*, 2018).

The main characteristics of blockchain are immutability and decentralization. Decentralization is achieved in a shared ledger where transactional updates are made through a consensus mechanism, making truly digital interaction between multiple parties possible (Baset *et al.*, 2018). Immutability means that once the data is aggregated into a block, the block is chained to existing blocks and the information is stored permanently and immutably (Knirsch, Unterweger & Engel, 2019).

An important feature of blockchain is the ability to build decentralized applications implementing programmable logic in which it is possible to define arbitrary custom rules for ownership, transaction formats and state transition functions (Buterin, 2014). It is thus possible to represent the logic that governs transactions on the network and regulate agreements between organizations. These are called smart contracts and are stored and executed on the blockchain. As such, the blockchain can be seen as a distributed ledger storing the results (i.e., transactions) of (smart) contracts whose correct valuation has been validated by the network participants (Rondanini *et al.*, 2020).

The other element to consider is that there are public blockchains, in which participation is open to all and transactions are validated by group consensus like proof of work, and private blockchains in which participation is restricted and transactions are validated by participant consensus which address the concern of a small number of entities with a common goal, yet do not trust each other. Most blockchain-based solutions are built on the Ethereum and Hyperledger Fabric platforms (Franciscon *et al.*, 2019).

3.2 Public sector digitalization

The public sector is used in contrast to the private sector to refer to the set of activities owned by the state whose role is to provide a public service to citizens. It is a generic term that includes public administrations, public institutions and public enterprises. The simplification of this sector through the use of technology is a topic that is in line with the current trends of public institutions in the pursuit of improving the services provided (Dobrolyubova, 2021; Gieske, George, van Meerkerk & van Buuren, 2020).

The adoption of new technologies offers innovative opportunities for the public sector and can improve interactions with citizens and enterprises. This trend is driven by technological changes that are altering the daily environment of citizens and private enterprises, and influencing their expectations in the delivery of public services (Mergel *et al.*, 2019; Scupola & Mergel, 2022).

Developments indicating examples of change related to the trend include the implementation of new citizen-centric policies and services improving the use of data for public interest (Sullivan *et al.*, 2021), more democratic data governance for public benefit with a growing interest in alternative approaches to data management, control and use that allow stakeholders to control personal data and decide how it is used (Carballa, 2019; Micheli *et al.*, 2020), and the use of blockchain to build trust and increase transparency as a new information infrastructure that could support and guarantee the secure exchange of information between public administrations, citizens and enterprises (De Filippi *et al.*, 2020; IBM, 2017).

Digitalization in the public sector goes far beyond the implementation of new technologies and the automating processes. It aims to offered citizens better services by taking a citizen-centric approach to design and deliver new services (Garcia-Garcia, Gil-Garcia & Gómez, 2014).

3.3 Related work

The related work review focused mainly on citizen-centric services and coordination approach on public sector using blockchain, the category to which this paper fits. Taking a Citizen-centric approach is basically about aligning public services with the needs and expectations of citizens to improve satisfaction. Technical and conceptual decisions are made according to the needs of citizens and not the constraints of public structures (Eggers, 2018).

An important facet of the evolution towards citizen-centric services is, on the one hand, transparency towards citizens, and on the other hand, coordination between the different services involved in a process. The coordination raised the data interoperability challenge. Key data interoperability issues facing organizations include : Data residing in multiple locations, managing access and policies across diverse datasets, data integration across sub-organizations (Prabhakar, 2021). The way the service is delivered, such as the use of personal data and the way they are stored, has an influence on the trust of citizens in public services (Tolbert & Mossberger, 2006). This identifies a blockchain use case.

One common way of resolving this problem is to propose an architecture that is solving this issue. Examples of Blockchain-based citizen centric so far reported include (Lo *et al.*, 2022), where a citizen-centric distributed data-sharing model that enhances trust among citizens and reduce bureaucratic processes is proposed using an architecture integrating Hyperledger Fabric with IPFS, a distributed system for storing and accessing files.

In (Parry, 2019) and (Upadhyaya *et al.*, 2018), blockchain-based architectures are proposed using a citizen-centric approach in the delivery of healthcare services while (Biswas & Roy, 2022) described an application of permissioned blockchain for user authentication using a citizen-

centric approach. However, none of these approaches are taking into account the coordination and data interoperability issue as a part of their solution.

In (Domalis *et al.*, 2021), a trustable and interoperable decentralized solution for Citizen-Centric using blockchain and Artificial Intelligence is proposed. The authors use machine learning to automatically recognizes data structures in order to solve the interoperability issue. Even if this paper has interesting research, it lacks practical evaluation of the architecture in order to understand the risks and challenges related.

The review presented on (Evermann & Kim, 2019) gives the current state of understanding on the use of blockchain on inter-organizational processes to achieve collaboration issues. Much of the recent work focuses on the use of "smart contracts" to act as the workflow engine and coordinate activities between participants. This paper takes a different direction by focusing on managing the coordination of institutions while maintaining their autonomy and the confidentiality of their local processes and enabling a citizen-centric approach that designs interactions around citizens.

4. Case study : vehicle registration

We will use Org1 and Org2 terms to identify the two organizations that are involved in the registration process we have selected. Org1 is responsible for the registration of road vehicles in Country X. When a citizen acquires a road vehicle, it should be registered in order to be used on the roads of Country X. Org1 collects at the time of the registration, the sales tax which is determined by a certificate, called VD0, provided by Org2. This certificate is used by Org1 to collect the calculated amount of tax fees in the time of the registration of a road vehicle. If completed, Org1 delivers the registration certificate, we will refer as VDX, to the citizen. The registration process, as shown in Figure I-1, is performed as follows : The purchaser of the vehicle (the citizen) fills out an application for VD0 to Org2, Org2 analyzes and processes the application, according to its internal procedure, and then grants to the purchaser the certificate VD0, where the tax fees to be collected are specified. The latter submits the vehicle registration application to the Org1 with the VD0 certificate and other documents. The Org1 analyzes and

processes the application, according to its internal procedure, and collects the tax fees indicated on the VD0 certificate before issuing the registration certificate VDX. The coordination effort is absent. Citizens have to go from one institution to another to coordinate their efforts to obtain registration. This problem is common in the public sector. Public institutions collaborate to provide diverse services to citizens, but coordination is a major problem (Punia & Saxena, 2004).

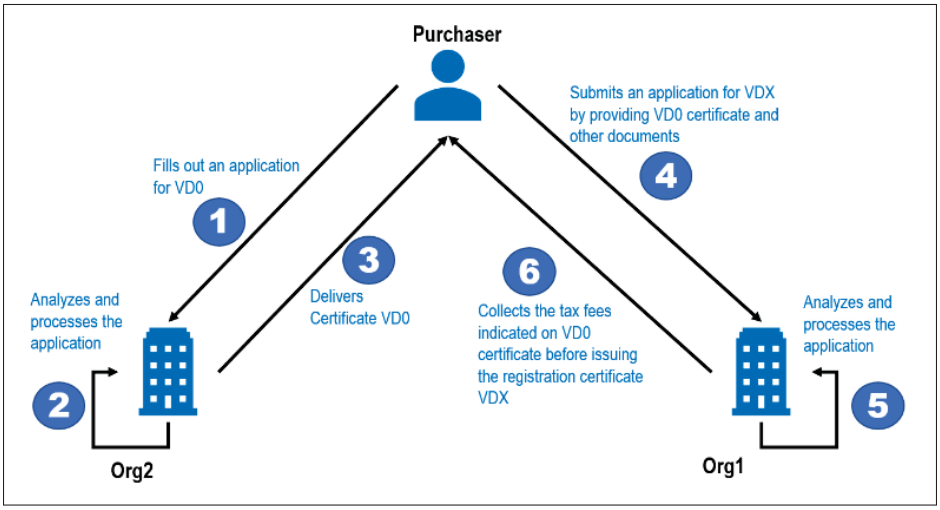

Figure-A I-1 Existing process

The main problems in the processing of this application are shown in Table I-1. From the improvement areas, we derive concrete design goals for our proof of concept and summarize them in Table I-2.

5. Proposed reference design

This section presents the design overview and considerations for an inter-organizational process architecture model that explores and practises the concepts of blockchain on a citizen-centric approach, and interfacing with local processes of each organization.

5.1 Design overview

Tableau-A I-1 Areas of improvement—status quo

Challenge	**Considerations**
Trust and Security	It is technically possible to modify or falsify a VDX in its current form.
Limited transparency	The citizen is used as a medium of information between Org1 and Org2. No transparency exists between the two organizations.
Slow process	The process is slow (4–5 days) when applying by mail (remote area) and usually takes 30 minutes when applying in person but involves physical travel.
Use of papers	The reduction of paper in processes is one of the objectives of organizations.
Non-citizen centred service	The interaction is not designed around the citizen.
Vertical silo	Organization is unable to communicate or interoperate with others.

Tableau-A I-2 Design goals

	Design goals	**Description**
DG1	Digitized-based process	Reduce paper use in the process.
DG2	Automate manual checking	Simplify data exchange procedures.
DG3	Provide Process History	Make the process traceable and transparent.
DG4	Authentication	Participant identification must be approved prior to each action.
DG5	Collaboration	Enable transparent and clear real-time collaboration.
DG6	Security	Protect the process against fraud and tampering.
DG7	Reduce time	Shorten overall process time.
DG8	Reduce cost	Reduce the cost of process execution.
DG9	Citizen-centric services	Granting control and use of personal data to citizen.
DG10	Interoperability	Solve the issue of operating in silos by ensuring communication while maintaining organization autonomy.

The proposed architecture is based on the Hyperledger Fabric (HLF) private blockchain platform to which organizations participating in an interorganizational process can connect. This architecture leverages member services functionality to maintain an access control layer

to manage participants and authorized actions and uses immutability principles to provide transparency in case transfer collaboration where all actors share a process definition and each actor performs different activities for a case. In order to conform to the design objectives, the architecture is organized around three layers as shown in Figure I-2 : an application layer, an interaction layer, and a network layer.

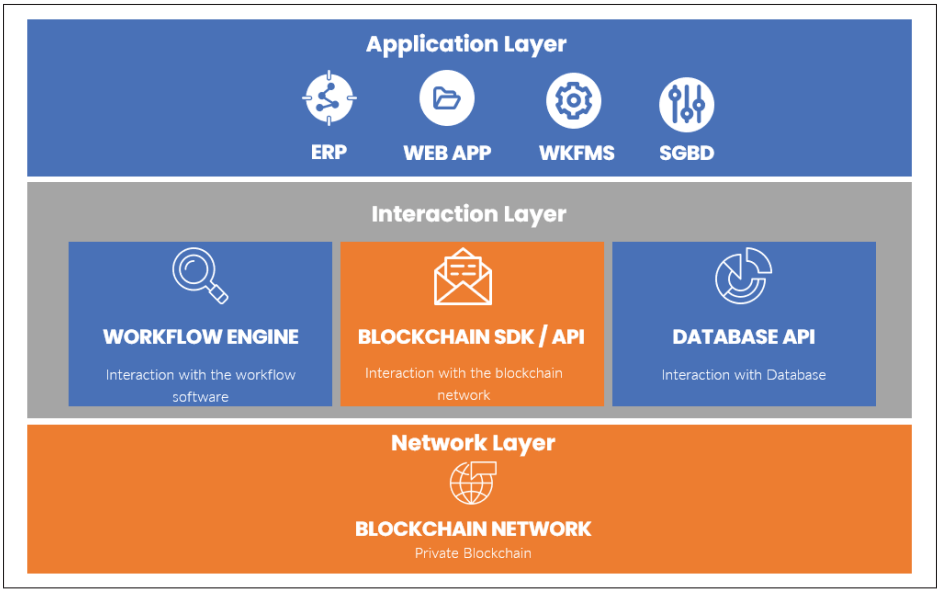

Figure-A I-2 Architecture overview

In detail, as shown in Figure I-3, the proposed architecture draws an operational organization of citizens in communication with public services.

5.1.1 Network layer

The network layer is composed of a single blockchain. A blockchain consists of a set of nodes belonging to organizations that form a network. Each node has a copy of the transaction and asset ledger.

5.1.2 Interaction layer

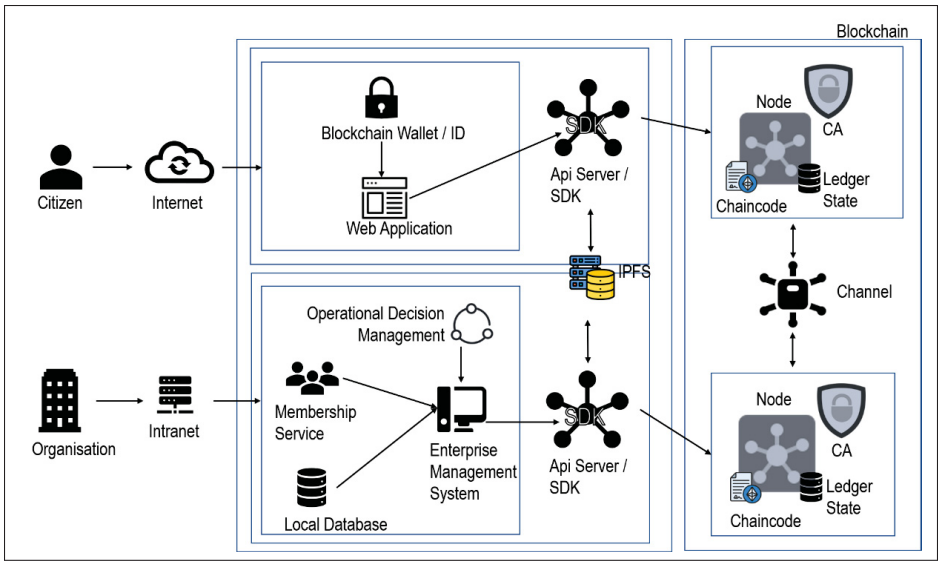

Figure-A I-3 Detailed architecture

The interaction layer provides the interfaces responsible for enabling communication between the network layer and the application layer and its components. It provides technical interoperability for data, such as structure and format. It has a component that links to the blockchain (Software Development Kit—SDK), a component that links to the legacy systems. To interact with the blockchain, sending and receiving transactions are done through blockchain nodes that can be communicated with the help of an SDK to allow the recording of transactions. In addition, custom events can be triggered from the smart contracts which allows the blockchain to initiate interactions and integrate with the outside world (Buch, 2020).

Figure I-4 shows the transaction flow in the interaction layer. Data can be generated through a custom business application, or a workflow engine private to an organization. This data is ingested through the SDK which provides an API transaction builder that targets the smart contract. The smart contract specifies the states of an object and controls the events that make the object move from one state to another. Based on the transaction, the validity of the smart contract logic is executed and in case it is valid, it is sent to the blockchain that updates the ledger. A blockchain data manager ensures that the data generated in a valid event within the

blockchain is returned to the initiating application through an HTTP callback link (web hook). The legacy system will be able to trigger a set of actions following the event-based workflow.

The architecture sets an API Service for each organization in order to make possible the implementation of a data access and transformation layer that formats and recognizes data structures and access.

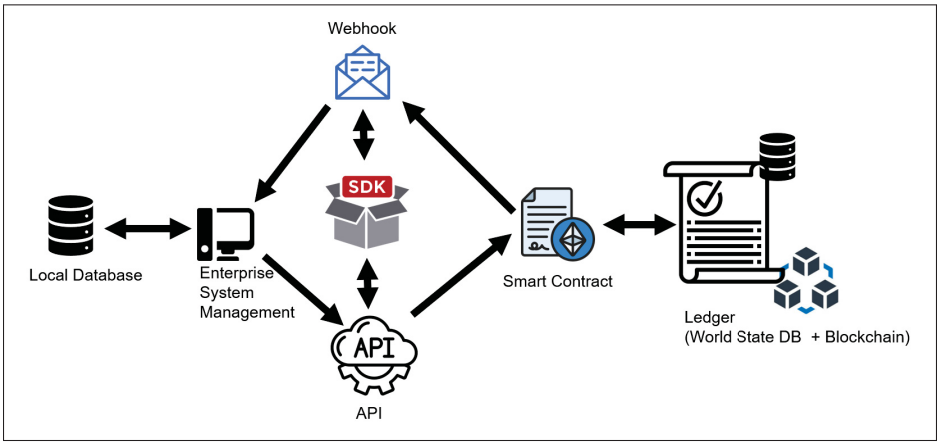

Figure-A I-4 Detailed architecture

5.1.3 Application layer

This layer consists of specific applications or portal serving specific activities of the organization that each group of end-user access with different functions according to the logic defined in the smart contract. The end users execute the application logic on the appropriate device (web application or desktop dashboard). The value proposition for end users is that the information they see is trustworthy and they may not be aware of the activities on the blockchain.

5.2 Design consideration

The introduction of a blockchain-based solution in the administrative procedures of the public sector requires a detailed analysis and must be preceded by a review of the procedures on which they operate and a reflection on their necessity. In addition, the design of a blockchain

requires the consideration of a number of important parameters as adjustments may be difficult or impossible once deployed.

5.2.1 Public-private approach

Given its generic scope, blockchain technology allows for many different system designs. In order to move towards an architecture focused on flexibility and privacy where the details of private processes do not need to be publicly visible, we have opted for a public-private approach.

The public-private approach, inspired by the principles of service-oriented architecture, considers the definition of a public workflow as a contract between the participating actors. Actors can provide private implementations for their parts of a process, as long as these are compatible with the public contract (Evermann & Kim, 2019). Public processes are common and standardized for every organization while private processes are specific to each organization and managed independently (Peristeras, Tarabanis & Goudos, 2009). The citizen is then considered as a participant in the public process, belonging to a specific organization (Org3), giving him the possibility to monitor all the processes and managing his own data.

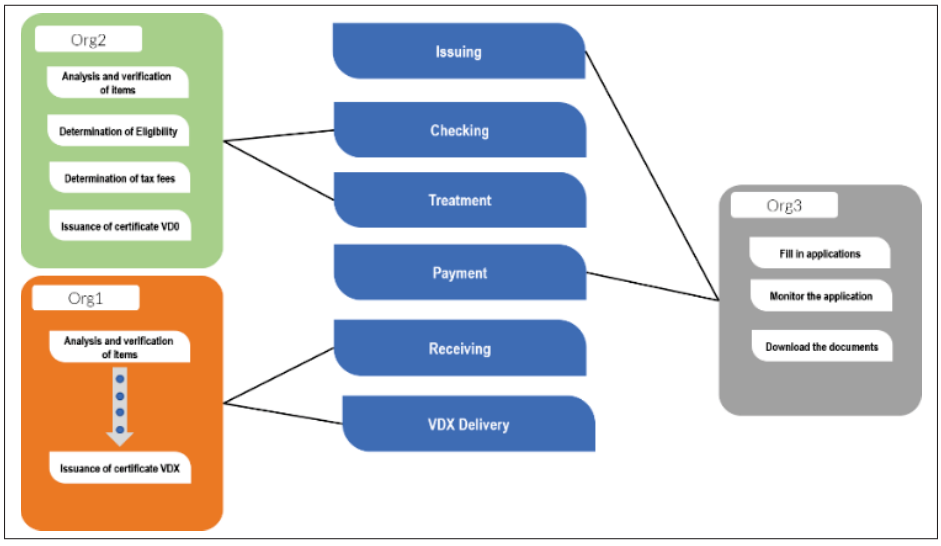

Figure-A I-5 Public private approach applies on the case study

Figure I-5 presents a public-private process model for the case study. The steps shown trace the life cycle of the registration application through to its processing. We will refer to this registration request as the VDX document.

5.2.2 Channel

The channel is a key element for communication and data confidentiality. Transactions in a channel are only visible to the participants registered to that channel. A single global channel is the approach that will be used since the case study does not mention specific private communication between participants. In this approach, participants in a channel have full visibility of the data that is stored in the channel ledger.

5.2.3 World state database

In HLF, the ledger consists of two essential elements, the blockchain transaction history and the state database which is a place to store the state in key-value format when the transaction is executed by smart contracts. The state database influences the data model, access patterns, deployment strategy, operational complexity and maintainability. CouchDB is the chosen data storage format due to the fact that it stores data as JSON documents, which allows for a richer data model, as well as providing rich query support. It also provides improved security for data protection and compliance in the blockchain.

5.2.4 Smart contract

Smart contracts are custom codes that encapsulate the logical rules and processes that govern transactions. How to handle the decision logic to balance flexibility, operability, and complexity of the solution is a key design decision. The approach used is to encapsulate the decision logic in the smart contract code, which makes it easier to manage from a time perspective even though this approach is less flexible, as it requires IT intervention to modify the smart contracts each time a change needs to be made. Interaction at the local process level is therefore managed internally with a workflow engine specific to each organization, we will define how the interaction is

done on the blockchain. There are two important concepts that should be raised when it comes to processes on a blockchain : states and transactions. There are conceptual objects of value, modelled as states, whose life cycle transitions are described by transactions. The life cycle of a VDX paper can be represented using a state transition diagram describing how VDXs evolve and the way transactions drive life cycle transitions. Smart contracts implement transaction logic that moves VDXs from one state to another as shown in the figure I-6. The states of the VDXs are maintained in the world state ledger.

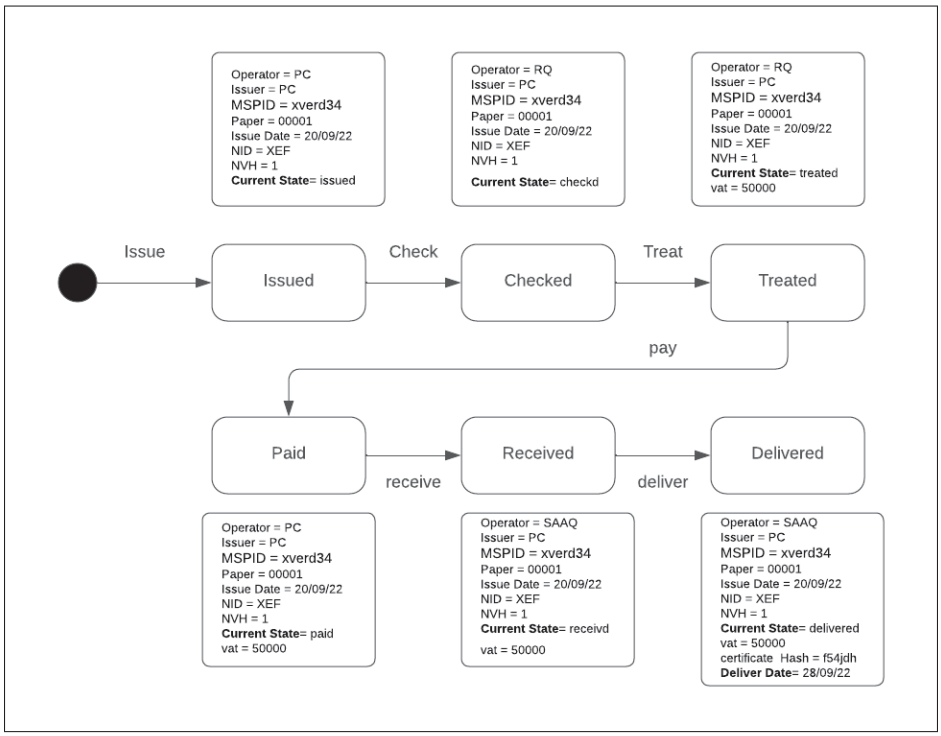

Figure-A I-6 State Diagram with VDX evolvement overview

5.2.5 On-chain and off-chain data

Finding the appropriate balance between on-chain and off-chain data is an important issue for a blockchain-based solution. It is important to know which data should be stored on-chain, as the number and size of objects has an impact on performance, and the scalability of the network is affected since the object will be part of the payload exchanged during the consensus process.

The approach used considers off-chain storage of data such as exchanged documents using the decentralized off-chain storage system IPFS. IPFS distributes files over the network and each file is addressed by its content. When they are stored in the system, they are assigned a cryptographic hash based on their content, unlike the traditional addressing used by HTTP, where a single server hosts many files and the information must be retrieved by accessing that server. With this approach, the network is scalable, but it increases the complexity of the application by directing operations to both blockchain and off-chain storage.

5.2.6 Integration with existing systems

Organizations have existing systems, referred to as traditional processing platforms, which are systems that perform processing within the organization. Blockchain is used in our context to track the life cycle of a document on a public process as well as the interactions between the participants of the network. The objective is to share data between several organizations in a verifiable and immutable way. It is necessary to transform the data between the blockchain and the formats of existing systems. The approach used is to use an API service integration model to bridge data from two different formats (in the blockchain and legacy systems). The communication of off-chain services with the blockchain is done through the SDK layer. The target service executes a request on the blockchain layer through the SDK via the API access points. This requires certificate configuration and access control list configuration. Figure 4.8 gives an overview of the interconnectivity between the blockchain and an external service. The API service thus enables the connection between an organization's client application and the HLF nodes available to that organization.

6. Evaluation

This section presents the evaluation of the system based on how well the developed architecture meets the requirements of the stakeholders, followed by the evaluation based on the proof of concept developed using a scenario representing the interaction with the system.

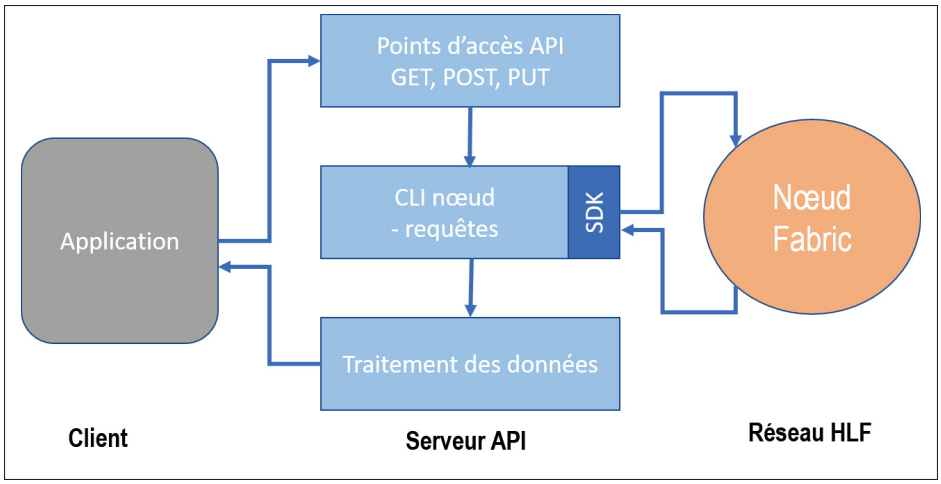

Figure-A I-7 Interconnectivity with blockchain

6.1 Architecture-based ATAM evaluation

The Architecture Tradeoff Analysis Method (ATAM) is one of the best scenario-based architecture methods that is used to evaluate software architectures using scenarios by evaluating the degree to which a system is likely to be able to satisfy its expected design goals by analyzing it from an architectural perspective (Daramola & Thebus, 2020).

The ATAM evaluation of the proposed architecture was conducted with the project coordinators. A group of five people was formed for this purpose, including (i) an information system solution architect (ii) IT staff with experience in blockchain use cases (iii) a project manager research and development coordinator (v) a non-IT expert and (iv) an associate professor expert in blockchain technology. The table I-3 presents an overview of the activities performed during ATAM activities.

6.2 Scenario-based observation

By conforming to our design, the executive organizations, Org2 (service provider) and Org3 (partner service), meet and create new processes to integrate with their existing processes. These new processes are common to all participants and linked to the internal private processes of each

Tableau-A I-3 Overview of ATAM activities

Activity	**Description**
Presentation of ATAM	An overview of the ATAM process, techniques and deliverables was presented to the group of coordinators using utility tree generation, architectural approach, elicitation or analysis and scenario mapping.
Presentation of design objectives	A description of the various design objectives as mentioned in Table I-2 was presented to the participants to establish a common understanding.
Presentation of the architecture	The system was presented to the participants using Krutchen's 4 + 1 architectural view model, as defined in (Daramola & Thebus, 2020), which prescribes a logical view, a development view, a process view, a physical view, and scenario descriptions. Figure I-8 present the development view using UML component diagram.
Identification of architectural approaches	The synthesis of the design objectives and the approach used to attempt their achievement were presented : Identification wallet(Citizen-centric service, Security), Channel (Privacy), World State Database(Historicity) ; Smart Contract (Automation, Historicity), On-chain and off-chain data (Collaboration), Integration with existing (Interoperability), Blockchain (Security), Public/private key encryption (Authentication, Security), Distributed registry (Traceability, Historicity, Collaboration), REST API (Reliability, Security).
Generation of the utility tree of quality attributes	A quality attribute utility tree was generated from questions asked by participants when trying to understand the architecture as well as the design goals and scenario elicitation. The utility tree prioritized specific quality attribute requirements based on the scenarios.
Analysis of architectural approaches	This step used the prioritized list of scenarios from previous step to identify risks, pain points, and trade-offs in the proposed architecture based on system interaction according to the scenario.
Presentation of the results	At this step, the results of the evaluation were presented.

organization. In order to identify the points expected by ATAM, an exploration of the system interaction presented in Figure I-9 allows the defined scenarios to be pinpointed and potential risks, non-risks, sensitivity points and trade-off points to be identified.

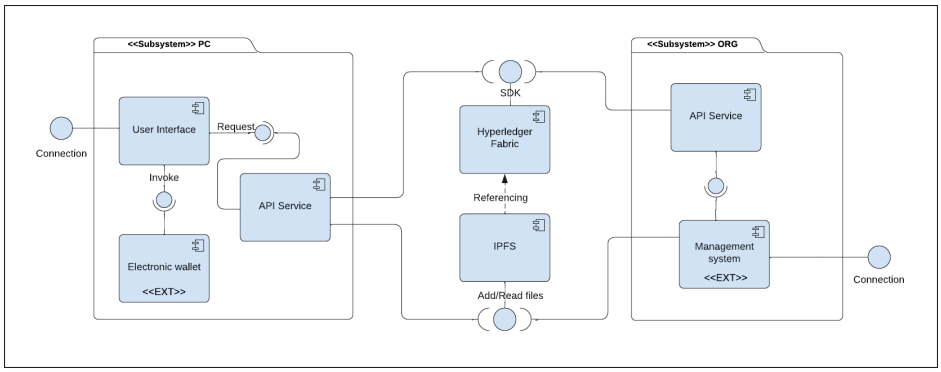

Figure-A I-8 Development View of the case study

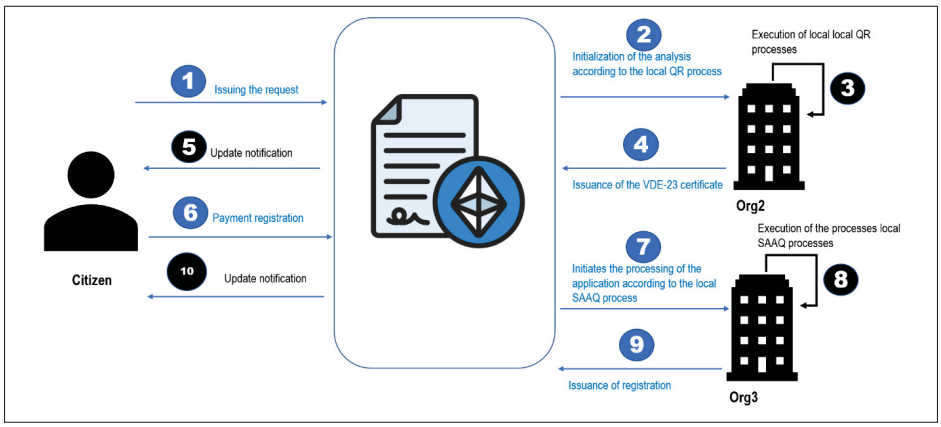

Figure-A I-9 System interaction

The public process is thus written on a smart contract to make the workflow shown in Figure 5 possible. We have considered the Fabric authentication model for this scenario. Beforehand, each participant in the network is provided with a digital identity via an X.509 certificate from their organization for secure communication between the participating entities (DG4) and the processes proceed as follows :

1. The citizen creates and issues the request. The request is registered on the blockchain using the smart contract. This step assumes that all documents to be attached are available in digital form or can be scanned before this step (DG1). To ensure verifiability, all information of each document is permanently and tamper-proof stored in the blockchain (DG3). In addition,

by providing the documents in digital form, simultaneous verification of the documents by the participating organizations is made possible (DG2) ;

2. The registration of the request on the smart contract initiates an event allowing the service provider organization to initiate its local process (DG10). The conditions that can be checked automatically are filled in using smart contracts (DG2).
3. The service provider performs local processing based on its private workflow. These internal processes are not visible to others (DG6) and are independent of the internal process changes of other organizations ;
4. The completion of the local process makes it possible to save the document issued by the service provider on the blockchain and update the status of the request using the API access points (DG10). Otherwise, the citizen must resubmit the documents ;
5. The application status update notifies the citizen that the check is at their level, and that they need to register the proof of payment of the sales tax ;
6. The payment process takes place outside the system, the citizen registers on the blockchain the proof of payment made.

Similar respectively to (2), (3) and (4), Steps 7, 8 et 9 are conducted with the service partner. In step 10 the citizen is notified of the end of the processing of his request and can download the document if needed.

For the entire process, each action regarding the status of the process is stored in the blockchain, i.e., each participant in the process also signs and approves the progress of the process using their private key (DG4). This provides a real-time process status accessible to all participants (DG5) and a complete process history (DG3). The use of blockchain also allows for a citizen-centric approach where the citizen monitors how the data is used (DG9). The use of an off-chain data storage mechanism guarantees better results on network performance and scalability, and the digitization of manual processes increases transparency and potentially processing times (DG7, DG8), as the citizen is no longer required to travel to make his request validated.

6.3 Network performance

The performance of HLF platform is evaluated regarding throughput, latency and scalability. In order to identify network capabilities, we investigated two main performance metrics on write (createVdx) and read (readVdx) operations in the ledger by measuring successful transactions per second (throughput) and response time per transaction in seconds (latency). Hyperledger Caliper was used to set up the test benchmark tool, the benchmark configuration has been set to alternate the transaction rates, numbers and types. Linear rate was used as configuration in order to find the workload rates that affect the system performance in an interesting way.

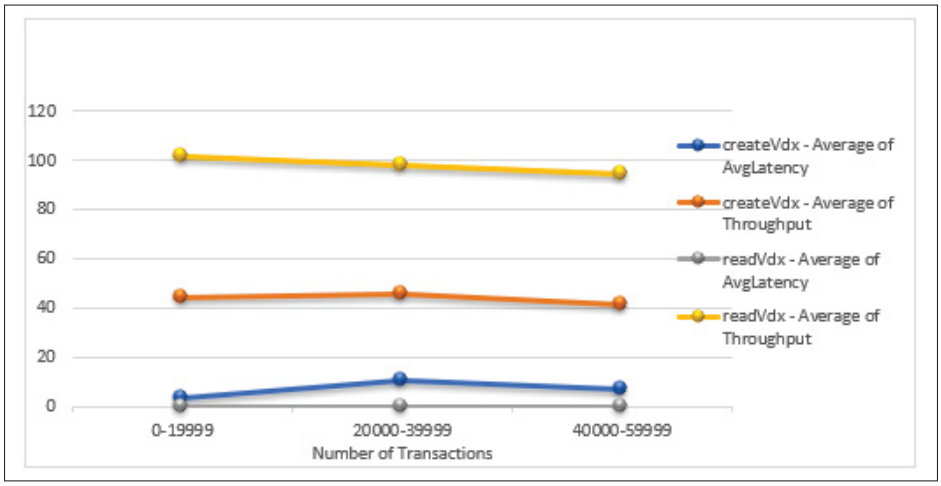

Figure-A I-10 Average of latency and throughput to the number of transactions

Under the test environment having 8 vCPUs, 3 GHz 11th Gen Intel Core and 16GB RAM, the network could support up to 100tps. Fig I-10 shows an inversely proportional relationship between latency and throughput. For read operations, the performance is similar even for a large number of transactions : 0.1 second latency on average and an average of 100 transactions/second. On the other hand, for write operations, above 5000 transactions, latency became significant. These results may differ depending on test environments but allow to prepare an appropriate hardware configuration for a large scale implementation.

7. Discussions and conclusion

Based on the results obtained with ATAM, a set of recommendations was proposed to support the use of the proposed architecture to achieve the design goals established in this research. These recommendations focus on specific components of the architecture and can be used as best practice guides to implement the proposed architecture.

- A security analysis of HLF and a presentation of possible threats to the different layers of the study case to explore how the integrated HFL functionality can mitigate them ;
- A “multi-channel” design that allows multiple use cases to run around organizations on the same network in order to conduct confidential and private transactions ;
- A multi-host HLF network deployment to ensure decentralization and transparency, with each organization deploying its own virtual machines to organize its components ;
- Standardization of exchange formats and development of cross-departmental standards between blockchain and organizations’ private management systems to extend implementation across departments with similar structures ;
- An outsourcing of the decision logic to an external rule engine rather than encoding the decision logic in the smart contract code ;
- A decentralized storage solution with a private IPFS network interfaced with an HLF client for data exchange ;
- A hash of the data before calling a smart contract and the definition of a mechanism to share the source data ; and a use of access control lists to restrict access to data to specific roles in the smart contract logic.

Taking the recommendations into account, the proposed architecture makes interoperability between local and public workflow possible in inter-organizational processes in the public sector and focuses on transparency and privacy for effective deployment. The use of blockchain as a decentralized solution increases the levels of trust, transparency and accessibility of services and enables a citizen-centric approach, where the citizen can monitor how their data is used. By opting for the proposed architecture, the design goals mentioned in Table A2 are addressed, as summarized in Table I-4.

Tableau-A I-4 Evaluation table

	Objectif de conception	**Description**
DG1	Digitized-based process	All relevant process step can be conducted in digital form.
DG2	Automate manual checking	Smart contract allows public process to be conducted automatically.
DG3	Provide Process History	Using HLF, we can retrieve all transactions history in the ledger of the blockchain and retrieve the current state of a specific document on the world state database.
DG4	Authentication	HLF is a permissioned network, it has a mechanism that allows participants to prove their identity before conducting transactions on the network.
DG5	Collaboration	The shared registry makes it possible to share data in real time for all participants in a transparent way.
DG8	Security	The security depends largely on HLF, the infrastructure of the blockchain used. Moreover, with the use of a channel and private data collection mechanism, privacy and confidentiality are ensured.
DG9	citizen-centric services	The public-private approach used allows citizens to have access to the data and to monitor the processing done on it.
DG10	Interoperability	The public-private approach allows for the implementation of a system that ensures better coordination and monitoring of public processes while leaving organizations with the autonomy and confidentiality of their local processes. By using the HLF blockchain SDK in our interaction layer, integration with the private process management systems of each organization is made possible by ensuring data interoperability.

However, the design objectives on reducing the overall process time (DG7) and reducing the cost (DG8) could not be fully evaluated and are considered as an open challenge. The accuracy on the process execution cost reduction is dependent on an estimation of the current cost of the registration process set and the overall process time reduction is dependent on the digitization of local processes. Both of these elements are not covered in this paper. In addition, the design of our approach may involve the knowledge and sharing of organizational knowledge about each organization's private processes, re-engineering of business processes to modify and integrate

these processes. This paper does not cover the necessary restructuring of organizations and new processes. As future work, an Architecture-Centric Evaluation for blockchain-based applications for the public sector will be published to present deeper detail about the evaluation process of a blockchain architecture using ATAM. Our exploration supports the conclusion that the integration of blockchain and smart contracts into the public process can solve the coordination problem between organizations. This indicates that there is significant potential for current blockchain-based application and integration initiatives, positioning blockchain both as an innovative alternative and as a vehicle for creating new management practices and models in the public sector digitalization.

ANNEXE II

CODE SOURCE DU CONTRAT INTELLIGENT

```
/*
 * PaperContract based on Hyperledger Fabric of IBM Corp
 * Ce contrat intelligent est bas sur les fichiers de base de IBM Corp disponibles
 * sur le lien: https://github.com/hyperledger/fabric-samples/blob/main/commercial-paper/organization/digibank/contract/lib/papercontract.js
* Il est tendu pour intgrer le traitement de la preuve de concept.
*/

'use strict';

// Fabric smart contract classes
const { Contract, Context } = require('fabric-contract-api');

// CopNet specifc classes
const VdxPaper = require('./paper.js');
const PaperList = require('./paperlist.js');
const QueryUtils = require('./queries.js');

/**
 * A custom context provides easy access to list of all vdx papers
 */
class VdxPaperContext extends Context {

    constructor() {
        super();
        // All papers are held in a list of papers
        this.paperList = new PaperList(this);
    }

}

/**
 * Define vdx paper smart contract by extending Fabric Contract class
 *
 */
class VdxPaperContract extends Contract {

    constructor() {
        // Unique namespace when multiple contracts per chaincode file
        super('org.copnet.vdxpaper');
    }

    /**
     * Define a custom context for vdx paper
    */
    createContext() {
        return new VdxPaperContext();
    }

    /**
     * Instantiate to perform any setup of the ledger that might be required.
     * @param {Context} ctx the transaction context
     */
    async instantiate(ctx) {
        console.log('Instantiate the contract');
```

```
55      }
56
57      /**
58       * Create vdx paper
59       *
60       * @param {Context} ctx the transaction context
61       * @param {String} issuer vdx paper issuer
62       * @param {Integer} paperNumber paper number for this issuer
63       * @param {String} createDateTime paper create date
64       * @param {String} docHash paper hash linked // transaction input - not written to asset
65       * @param {String} nid identification number of issuer
66       * @param {String} nvh number of vehicle
67       * @param {String} fullname Fullname of Issuer
68       * @param {String} nas Assurance
69       * @param {String} lines lines of paper
70      */
71      async create(ctx, issuer, paperNumber, createDateTime, nid, nvh, fullname, nas, lines, docHash) {
72
73          // create an instance of the paper
74          let paper = VdxPaper.createInstance(issuer, paperNumber, createDateTime, nid, nvh, fullname, nas, lines);
75
76          // Smart contract, rather than paper, moves paper into CREATED state
77          paper.setCreated();
78
79          // save the issuer's MSP
80          let mspid = ctx.clientIdentity.getMSPID();
81          paper.setIssuerMSP(mspid);
82
83          // Newly created paper is operated by the issuer to begin with (recorded for reporting purposes)
84          paper.setOperator(issuer);
85          paper.setOperatorMSP(mspid);
86
87          // Add the paper to the list of all similar vdx papers in the ledger world state
88          await ctx.paperList.addPaper(paper);
89
90          // Must return a serialized paper to caller of smart contract
91          return paper;
92      }
93
94      /**
95       * Issue vdx paper
96       *
97       * @param {Context} ctx the transaction context
98       * @param {String} issuer vdx paper issuer
99       * @param {Integer} paperNumber paper number for this issuer
100      * @param {String} issueDateTime paper issue date
101      * @param {String} currentOperator current operator of paper
102      * @param {String} newOperator new operator of paper
103      * @param {String} docHash paper hash linked // transaction input - not written to asset
104     */
105     async issue(ctx, issuer, paperNumber, issueDateTime, currentOperator, newOperator) {
106
107         // Retrieve the current paper using key fields provided
108         let paperKey = VdxPaper.makeKey([issuer, paperNumber]);
109         let paper = await ctx.paperList.getPaper(paperKey);
110
111         // Validate current operator (Current Operator is the old one)
112         if (paper.getOperator() !== currentOperator) {
113             throw new Error('\nPaper ' + issuer + paperNumber + ' is not operated by ' + currentOperator);
114         }
```

```
115
116         // First check transactions moves state from CREATED to ISSUED
117         if (paper.isCreated()) {
118             paper.setIssued();
119         }
120
121         // Check paper is not already ISSUED or another forward state level
122         if (paper.isIssued()) {
123             paper.setOperator(newOperator);
124             // save the operator's MSP
125             let opMspid = ctx.clientIdentity.getMSPID();
126             paper.setOperatorMSP(opMspid);
127             paper.issueDateTime = issueDateTime;
128         } else {
129             throw new Error('\nPaper ' + issuer + paperNumber + ' is not checked. Current state = ' + paper.getCurrentState());
130         }
131
132         // Update the paper
133         await ctx.paperList.updatePaper(paper);
134         return paper;
135     }
136
137
138     /**
139      * Check vdx paper
140      *
141       * @param {Context} ctx the transaction context
142       * @param {String} issuer vdx paper issuer
143       * @param {Integer} paperNumber paper number for this issuer
144       * @param {String} currentOperator current operator of paper
145       * @param {String} newOperator new operator of paper
146       * @param {String} checkDateTime time paper was checked // transaction input - not written to asset
147       * @param {String} comment A bit comment if needed // transaction input - not written to asset
148      */
149     async check(ctx, issuer, paperNumber, currentOperator, newOperator, checkDateTime, comment) {
150
151         // Retrieve the current paper using key fields provided
152         let paperKey = VdxPaper.makeKey([issuer, paperNumber]);
153         let paper = await ctx.paperList.getPaper(paperKey);
154
155         // Validate current operator
156         if (paper.getOperator() !== currentOperator) {
157             throw new Error('\nPaper ' + issuer + paperNumber + ' is not operated by ' + currentOperator);
158         }
159
160         // First check transactions moves state from ISSUED to CHECKED
161         if (paper.isIssued()) {
162             paper.setChecked();
163         }
164
165         // Check paper is not already TREATED or another forward state level
166         if (paper.isChecked()) {
167             paper.setOperator(newOperator);
168             // save the operator's MSP
169             let opMspid = ctx.clientIdentity.getMSPID();
170             paper.setOperatorMSP(opMspid);
171         } else {
172             throw new Error('\nPaper ' + issuer + paperNumber + ' is not checked. Current state = ' + paper.getCurrentState());
173         }
174
```

```
175         // Update the paper
176         await ctx.paperList.updatePaper(paper);
177         return paper;
178     }
179
180     /**
181      * Treat vdx paper
182      *
183       * @param {Context} ctx the transaction context
184       * @param {String} issuer vdx paper issuer
185       * @param {Integer} paperNumber paper number for this issuer
186       * @param {String} currentOperator current operator of paper
187       * @param {String} newOperator new operator of paper
188       * @param {String} treatDateTime time paper was treated // transaction input - not written to asset
189       * @param {String} docHash paper hash linked // transaction input - not written to asset
190       * @param {Number} vat face value of paper // transaction input - not written to asset
191      */
192     async treat(ctx, issuer, paperNumber, currentOperator, newOperator, treatDateTime, docHash, vat) {
193
194         // Retrieve the current paper using key fields provided
195         let paperKey = VdxPaper.makeKey([issuer, paperNumber]);
196         let paper = await ctx.paperList.getPaper(paperKey);
197
198         // Validate current operator
199         if (paper.getOperator() !== currentOperator) {
200             throw new Error('\nPaper ' + issuer + paperNumber + ' is not operated by ' + currentOperator);
201         }
202         // First check transactions moves state from CHECKED to TREATED
203         if (paper.isChecked()) {
204             paper.setTreated();
205         }
206
207         // Check paper is not already PAID or another forward state level
208         if (paper.isTreated()) {
209             paper.setOperator(newOperator);
210             // save the operator's MSP
211             let opMspid = ctx.clientIdentity.getMSPID();
212             paper.setOperatorMSP(opMspid);
213             paper.vat = parseFloat(vat);
214         } else {
215             throw new Error('\nPaper ' + issuer + paperNumber + ' is not treated. Current state = ' + paper.getCurrentState());
216         }
217
218         // Update the paper
219         await ctx.paperList.updatePaper(paper);
220         return paper;
221     }
222
223     /**
224      * Paid vdx paper
225      *
226       * @param {Context} ctx the transaction context
227       * @param {String} issuer vdx paper issuer
228       * @param {Integer} paperNumber paper number for this issuer
229       * @param {String} currentOperator current operator of paper
230       * @param {String} newOperator new operator of paper
231       * @param {String} payDateTime time paper was paid // transaction input - not written to asset
232       * @param {String} docHash proof of payment doc hash linked // transaction input - not written to asset
233       * @param {String} reference payment reference // transaction input - not written to asset
234      */
```

```
235     async pay(ctx, issuer, paperNumber, currentOperator, newOperator, payDateTime, docHash, reference) {
236
237         // Retrieve the current paper using key fields provided
238         let paperKey = VdxPaper.makeKey([issuer, paperNumber]);
239         let paper = await ctx.paperList.getPaper(paperKey);
240
241         //Validation more strict
242         // Validate current issuer's MSP in the paper === invoking Payer's MSP id - can only transfer if you are the owning org.
243         if (paper.getIssuerMSP() !== ctx.clientIdentity.getMSPID()) {
244             throw new Error('\nPaper ' + issuer + paperNumber + ' is not owned by the current invoking Organisation, and not authorised to
                    paid');
245         }
246
247         // First check transactions moves state from TREATED to PAID
248         if (paper.isTreated()) {
249             paper.setPaid();
250         }
251
252         // Check paper is not already RECEIVED or another forward state level
253         if (paper.isPaid()) {
254             paper.setOperator(newOperator);
255             // save the operator's MSP
256             let opMspid = ctx.clientIdentity.getMSPID();
257             paper.setOperatorMSP(opMspid);
258         } else {
259             throw new Error('\nPaper ' + issuer + paperNumber + ' is not treated. Current state = ' + paper.getCurrentState());
260         }
261
262         // Update the paper
263         await ctx.paperList.updatePaper(paper);
264         return paper;
265     }
266
267     /**
268      * Received vdx paper
269      *
270       * @param {Context} ctx the transaction context
271       * @param {String} issuer vdx paper issuer
272       * @param {Integer} paperNumber paper number for this issuer
273       * @param {String} currentOperator current operator of paper
274       * @param {String} newOperator new operator of paper
275       * @param {String} receiveDateTime time paper was received // transaction input - not written to asset
276       * @param {String} comment A bit comment if needed // transaction input - not written to asset
277      */
278     async receive(ctx, issuer, paperNumber, currentOperator, newOperator, receiveDateTime, comment) {
279
280         // Retrieve the current paper using key fields provided
281         let paperKey = VdxPaper.makeKey([issuer, paperNumber]);
282         let paper = await ctx.paperList.getPaper(paperKey);
283
284         // Validate current operator
285         if (paper.getOperator() !== currentOperator) {
286             throw new Error('\nPaper ' + issuer + paperNumber + ' is not operated by ' + currentOperator);
287         }
288
289         // First check transactions moves state from PAID to RECEIVED
290         if (paper.isPaid()) {
291             paper.setReceived();
292         }
293
```

```
294         // Check paper is not already DELIVERED or another state level
295         if (paper.isReceived()) {
296             paper.setOperator(newOperator);
297             // save the operator's MSP
298             let opMspid = ctx.clientIdentity.getMSPID();
299             paper.setOperatorMSP(opMspid);
300         } else {
301             throw new Error('\nPaper ' + issuer + paperNumber + ' is not received. Current state = ' + paper.getCurrentState());
302         }
303 
304         // Update the paper
305         await ctx.paperList.updatePaper(paper);
306         return paper;
307     }
308 
309     /**
310      * Deliver vdx paper
311      *
312       * @param {Context} ctx the transaction context
313       * @param {String} issuer vdx paper issuer
314       * @param {Integer} paperNumber paper number for this issuer
315       * @param {String} currentOperator current operator of paper
316       * @param {String} newOperator new operator of paper
317       * @param {String} deliverDateTime time paper was delivered // transaction input - not written to asset
318       * @param {String} docImma paper certificate immatriculation hash linked // transaction input - not written to asset
319      */
320     async deliver(ctx, issuer, paperNumber, currentOperator, newOperator, deliverDateTime, fileNumber, docImma) {
321 
322         // Retrieve the current paper using key fields provided
323         let paperKey = VdxPaper.makeKey([issuer, paperNumber]);
324         let paper = await ctx.paperList.getPaper(paperKey);
325 
326         // Validate current operator
327         if (paper.getOperator() !== currentOperator) {
328             throw new Error('\nPaper ' + issuer + paperNumber + ' is not operated by ' + currentOperator);
329         }
330 
331         // Check paper is not alread in a state of DELIVERED
332         if (paper.isDelivered()) {
333             throw new Error('\nPaper ' + issuer + paperNumber + ' has already been delivered');
334         }
335         // First check transactions moves state from RECEIVED to DELIVERED
336         if (paper.isReceived()) {
337             paper.setDelivered();
338         }
339 
340         // Check paper is on anoter wrong state level
341         if (paper.isDelivered()) {
342             paper.setOperator(newOperator);
343             // save the operator's MSP
344             let opMspid = ctx.clientIdentity.getMSPID();
345             paper.setOperatorMSP(opMspid);
346             paper.fileNumber = fileNumber
347             paper.deliverDateTime = deliverDateTime; // record delivering date against the asset (the complement to 'issue date')
348             paper.docImma = docImma; // record the hash of the final doc Immatriculation delivered
349         } else {
350             throw new Error('\nPaper ' + issuer + paperNumber + ' is not delivered. Current state = ' + paper.getCurrentState());
351         }
352 
353         // Update the paper
```

```
354         await ctx.paperList.updatePaper(paper);
355         return paper;
356     }
357
358     // Query transactions
359
360     /**
361      * Query history of a vdx paper
362      * @param {Context} ctx the transaction context
363      * @param {String} issuer vdx paper issuer
364      * @param {Integer} paperNumber paper number for this issuer
365     */
366     async queryHistory(ctx, issuer, paperNumber) {
367
368         // Get a key to be used for History query
369
370         let query = new QueryUtils(ctx, 'org.copnet.paper');
371         let results = await query.getAssetHistory(issuer, paperNumber); // (cpKey);
372         return results;
373
374     }
375
376     /**
377     * queryOperator vdx paper: supply name of operating org, to find list of papers based on operator field
378     * @param {Context} ctx the transaction context
379     * @param {String} operator vdx paper operator
380     */
381     async queryOperator(ctx, operator) {
382
383         let query = new QueryUtils(ctx, 'org.copnet.paper');
384         let operator_results = await query.queryKeyByOperator(operator);
385
386         return operator_results;
387     }
388
389     /**
390     * queryPartial vdx paper - provide a prefix eg. "RQ" will list all papers _issued_ by RQ etc etc
391     * @param {Context} ctx the transaction context
392     * @param {String} prefix asset class prefix (added to paperlist namespace) eg. org.copnet.paperRQ asset listing: papers issued by RQ.
393     */
394     async queryPartial(ctx, prefix) {
395
396         let query = new QueryUtils(ctx, 'org.copnet.paper');
397         let partial_results = await query.queryKeyByPartial(prefix);
398
399         return partial_results;
400     }
401
402     /**
403     * queryAdHoc vdx paper - supply a custom mango query
404     * eg - as supplied as a param:
405     * ex1: ["{\"selector\":{\"vat\":{\"$lt\":100}}}"]
406     * ex2: ["{\"selector\":{\"vat\":{\"$gt\":500000}}}"]
407     *
408     * @param {Context} ctx the transaction context
409     * @param {String} queryString querystring
410     */
411     async queryAdhoc(ctx, queryString) {
412
413         let query = new QueryUtils(ctx, 'org.copnet.paper');
```

```
        let querySelector = JSON.parse(queryString);
        let adhoc_results = await query.queryByAdhoc(querySelector);

        return adhoc_results;
    }


    /**
     * queryNamed - supply named query - 'case' statement chooses selector to build (pre-canned for demo purposes)
     * @param {Context} ctx the transaction context
     * @param {String} queryname the 'named' query (built here) - or - the adHoc query string, provided as a parameter
     */
    async queryNamed(ctx, queryname) {
        let querySelector = {};
        switch (queryname) {
            case "delivered":
                querySelector = { "selector": { "currentState": 6 } }; // 6 = delivered state
                break;
            case "received":
                querySelector = { "selector": { "currentState": 5 } }; // 5 = received state
                break;
            case "paid":
                querySelector = { "selector": { "currentState": 4 } }; // 4 = paid state
                break;
            case "treated":
                querySelector = { "selector": { "currentState": 3 } }; // 3 = treated state
                break;
            case "checked":
                querySelector = { "selector": { "currentState": 2 } }; // 2 = checked state
                break;
            case "issued":
                querySelector = { "selector": { "currentState": 1 } }; // 1 = issued state
                break;
            case "value":
                // may change to provide as a param - fixed value for now in this sample
                querySelector = { "selector": { "vat": { "$gt": 1000 } } }; // to test, issue CommPapers with vat <= or => this figure.
                break;
            default: // else, unknown named query
                throw new Error('invalid named query supplied: ' + queryname + '- please try again ');
        }

        let query = new QueryUtils(ctx, 'org.copnet.paper');
        let adhoc_results = await query.queryByAdhoc(querySelector);

        return adhoc_results;
    }

}

module.exports = VdxPaperContract;
```

ANNEXE III

CODE SOURCE DU SERVICE API

Cette section présente le code source du service API de l'organisation PC, les autres services API ont été construits de manière similaire dépendamment des interactions de leurs organisations respectives avec le contrat intelligent. La documentation complète peut être trouvée sur https://documenter.getpostman.com/view/3607507/2s8YsozEma

```
const express = require('express');
const bodyParser = require('body-parser');
const fs = require('fs');
const yaml = require('js-yaml');
const path = require('path');


// Setting for Hyperledger Fabric
const { Wallets, Gateway } = require('fabric-network');


// const ccpPath = path.resolve(__dirname, '.', 'connection-org3.json');
const ccpPath = '../gateway/connection-org3.yaml'
const userName = 'appUser';
const walletPath = '../identity/user/appUser/wallet';
const port = 8080

const app = express();
app.use(bodyParser.json())

async function customizeTransaction(contract, fcn, ...args){
    //This function allow us to retrieve TXID and Result (It replace contract.submitTransaction)
    const transaction = contract.createTransaction(fcn);
    const result = await transaction.submit(...args);
    return {
        TxID : transaction.getTransactionId(),
        result: result.toString()
    }
}


console.log(`****API Service for Citizen running on port ${port}****`)

app.post('/api/create', async function (req, res) {
    try {

        //1. Connection Using Wallet and Access to the Gateway

        // 1.1. Loading of Wallet Identity
        // A wallet stores a collection of identities for use
        const wallet = await Wallets.newFileSystemWallet(walletPath);

        //1.2. Check to see if we've already enrolled the user.
        // The user have to be created au prealable
        const identity = await wallet.get(userName);
```

```
46        if (!identity) {
47           console.log(‘An identity for the user "${userName}" does not exist in the wallet‘);
48           console.log('Ask Admin to Run the EnrollUser application before retrying');
49           return;
50        }
51        //1.3. Load info from connection-orgX.yaml (see ccpPath)
52        let connectionProfile = yaml.safeLoad(fs.readFileSync(ccpPath, 'utf8'));
53
54        //1.4 Set connection options; identity and wallet
55        let connectionOptions = {
56           identity: userName,
57           wallet: wallet,
58           discovery: { enabled:true, asLocalhost: true }
59        };
60        // Connect to gateway using application specified parameters
61        console.log('****Connect to Fabric gateway.****');
62
63        //1.5 Create a new gateway for connecting to our peer node.
64        const gateway = new Gateway();
65        await gateway.connect(connectionProfile, connectionOptions);
66
67        //2. Access PaperNet network
68        console.log('****Use network channel: mychannel.****');
69
70        //2.1 Get the network (channel) our contract is deployed to.
71        const network = await gateway.getNetwork('mychannel');
72
73        //2.2. Get addressability to vdx paper contract
74        console.log('Use org.copnet.vdxpaper smart contract.');
75        const contract = network.getContract('papervdx');
76
77        //3. SubmitTransaction
78        //3.1 create vdx paper
79        console.log('Submit vdx paper create transaction.');
80
81        //create transaction - requires 10 arguments, ex: ('create', 'PC', '00001', '2022-11-21', '35464646', '2', 'John Doe', 'NAS003', 'Ford
                Escape 2014, XX', '')
82        let data = req.body
83        //console.log(req)
84        const response = await customizeTransaction(contract, 'create', 'PC', data.paperNumber, data.createDateTime,
85        data.nid, data.nvh, data.fullname, data.nas, lines = data.lines, docHash = data.docHash);
86
87        //3.2 process response
88        console.log(‘Process create transaction id. , TxID is: ${response.TxID}‘);
89        console.log(‘Process create transaction response. , Result is: ${response.result}‘);
90
91        res.status(200).json({response});
92
93        // Disconnect from the gateway.
94        await gateway.disconnect();
95
96     } catch (error) {
97        console.error(‘Failed to process Create transaction: ${error}‘);
98        res.status(500).json({error: error});
99        process.exit(1);
100    }
101 });
102
103 app.post('/api/issue', async function (req, res) {
104    try {
```

```
105
106         //1. Connection Using Wallet and Access to the Gateway
107
108         // 1.1. Loading of Wallet Identity
109         // A wallet stores a collection of identities for use
110         const wallet = await Wallets.newFileSystemWallet(walletPath);
111
112         //1.2. Check to see if we've already enrolled the user.
113         // The user have to be created au prealable
114         const identity = await wallet.get(userName);
115         if (!identity) {
116             console.log(`An identity for the user "${userName}" does not exist in the wallet`);
117             console.log('Ask Admin to Run the EnrollUser application before retrying');
118             return;
119         }
120         //1.3. Load info from connection-orgX.yaml (see ccpPath)
121         let connectionProfile = yaml.safeLoad(fs.readFileSync(ccpPath, 'utf8'));
122
123         //1.4 Set connection options; identity and wallet
124         let connectionOptions = {
125             identity: userName,
126             wallet: wallet,
127             discovery: { enabled:true, asLocalhost: true }
128         };
129         // Connect to gateway using application specified parameters
130         console.log('****Connect to Fabric gateway.****');
131
132         //1.5 Create a new gateway for connecting to our peer node.
133         const gateway = new Gateway();
134         await gateway.connect(connectionProfile, connectionOptions);
135
136         //2. Access PaperNet network
137         console.log('****Use network channel: mychannel.****');
138
139         //2.1 Get the network (channel) our contract is deployed to.
140         const network = await gateway.getNetwork('mychannel');
141
142         //2.2. Get addressability to vdx paper contract
143         console.log('Use org.copnet.vdxpaper smart contract.');
144         const contract = network.getContract('papervdx');
145
146         //3. SubmitTransaction
147         //3.1 issue vdx paper
148         console.log('Submit vdx paper issue transaction.');
149
150         //issue transaction - requires 6 argument, ex: ('issue', 'PC', '00001', '2022-11-21', 'PC', 'RQ')
151         let data = req.body
152         const response = await customizeTransaction(contract, 'issue', 'PC', data.paperNumber, data.issueDateTime,
153         'PC', 'RQ');
154
155         //3.2 process response
156         console.log(`Process create transaction id. , TxID is: ${response.TxID}`);
157         console.log(`Process create transaction response. , Result is: ${response.result}`);
158
159         res.status(200).json({response});
160
161         // Disconnect from the gateway.
162         await gateway.disconnect();
163
164     } catch (error) {
```

```
        console.error(`Failed to process issue transaction: ${error}`);
        res.status(500).json({error: error});
        process.exit(1);
    }
});

app.post('/api/pay', async function (req, res) {
    try {

        //1. Connection Using Wallet and Access to the Gateway

        // 1.1. Loading of Wallet Identity
        // A wallet stores a collection of identities for use
        const wallet = await Wallets.newFileSystemWallet(walletPath);

        //1.2. Check to see if we've already enrolled the user.
        // The user have to be created au prealable
        const identity = await wallet.get(userName);
        if (!identity) {
            console.log(`An identity for the user "${userName}" does not exist in the wallet`);
            console.log('Ask Admin to Run the EnrollUser application before retrying');
            return;
        }
        //1.3. Load info from connection-orgX.yaml (see ccpPath)
        let connectionProfile = yaml.safeLoad(fs.readFileSync(ccpPath, 'utf8'));

        //1.4 Set connection options; identity and wallet
        let connectionOptions = {
            identity: userName,
            wallet: wallet,
            discovery: { enabled:true, asLocalhost: true }
        };
        // Connect to gateway using application specified parameters
        console.log('****Connect to Fabric gateway.****');

        //1.5 Create a new gateway for connecting to our peer node.
        const gateway = new Gateway();
        await gateway.connect(connectionProfile, connectionOptions);

        //2. Access PaperNet network
        console.log('****Use network channel: mychannel.****');

        //2.1 Get the network (channel) our contract is deployed to.
        const network = await gateway.getNetwork('mychannel');

        //2.2. Get addressability to vdx paper contract
        console.log('Use org.copnet.vdxpaper smart contract.');
        const contract = network.getContract('papervdx');

        //3. SubmitTransaction
        //3.1 pay vdx paper
        console.log('Submit vdx paper pay transaction.');

        //pay transaction - requires 8 argument, ex: ('pay', 'PC', '00001', 'PC', 'SAAQ', '2022-11-21', 'dochashTestfeeedkdkdkd','RDRCF');
        let data = req.body
        const response = await customizeTransaction(contract, 'pay', 'PC', data.paperNumber, 'PC',
        'SAAQ', data.payDateTime, data.docHash, data.reference);

        //3.2 process response
        console.log(`Process create transaction id. , TxID is: ${response.TxID}`);
```

```
225         console.log(`Process create transaction response. , Result is: ${response.result}`);
226
227         res.status(200).json({response});
228
229         // Disconnect from the gateway.
230         await gateway.disconnect();
231
232     } catch (error) {
233         console.error(`Failed to process pay transaction: ${error}`);
234         res.status(500).json({error: error});
235         process.exit(1);
236     }
237 });
238
239 app.post('/api/queryhistory', async function (req, res) {
240     try {
241
242         //1. Connection Using Wallet and Access to the Gateway
243
244         // 1.1. Loading of Wallet Identity
245         // A wallet stores a collection of identities for use
246         const wallet = await Wallets.newFileSystemWallet(walletPath);
247
248         //1.2. Check to see if we've already enrolled the user.
249         // The user have to be created au prealable
250         const identity = await wallet.get(userName);
251         if (!identity) {
252             console.log(`An identity for the user "${userName}" does not exist in the wallet`);
253             console.log('Ask Admin to Run the EnrollUser application before retrying');
254             return;
255         }
256         //1.3. Load info from connection-orgX.yaml (see ccpPath)
257         let connectionProfile = yaml.safeLoad(fs.readFileSync(ccpPath, 'utf8'));
258
259         //1.4 Set connection options; identity and wallet
260         let connectionOptions = {
261             identity: userName,
262             wallet: wallet,
263             discovery: { enabled:true, asLocalhost: true }
264         };
265         // Connect to gateway using application specified parameters
266         console.log('****Connect to Fabric gateway.****');
267
268         //1.5 Create a new gateway for connecting to our peer node.
269         const gateway = new Gateway();
270         await gateway.connect(connectionProfile, connectionOptions);
271
272         //2. Access PaperNet network
273         console.log('****Use network channel: mychannel.****');
274
275         //2.1 Get the network (channel) our contract is deployed to.
276         const network = await gateway.getNetwork('mychannel');
277
278         //2.2. Get addressability to vdx paper contract
279         console.log('Use org.copnet.vdxpaper smart contract.');
280         const contract = network.getContract('papervdx');
281
282         // Evaluate the specified transaction.
283         // queryHistory transaction - requires 2 argument, ex: ('queryHistory', 'PC', '00001')
284         data = req.body
```

```
285         console.log(`Query VDX Paper History of data ${data.paperNumber}`);
286         const result = await contract.evaluateTransaction('queryHistory', 'PC', data.paperNumber);
287         console.log(`Transaction History has been evaluated, result is: ${result.toString()}`);
288         res.status(200).json({response: result.toString()});
289 
290         // Disconnect from the gateway.
291         await gateway.disconnect();
292 
293     } catch (error) {
294         console.error(`Failed to evaluate transaction History: ${error}`);
295         res.status(500).json({error: error});
296         process.exit(1);
297     }
298 });
299 
300 
301 app.listen(port);
```

ANNEXE IV

CONFIGURATION DE RÉFÉRENCE POUR LE TEST DE PERFORMANCE

Dans le but de tester la performance du réseau déployé, cette recherche a utilisé l'outil Hyperledger Caliper dans lequel des fichiers de configuration ont été écrits et ont permis de réaliser plus de 200000 transactions. Le script présenté ici est un modèle reprenant la structure utilisé pour effectuer ces différents tests (*createVdx* et *readVdx*).

```
test:
  name: papervdx-contract-benchmark
  description: A test benchmark
  workers:
    type: local
    number: 5
  rounds:
    - label: createVdx
      description: Create Vdx benchmark
      txNumber: 1000
      rateControl:
        { type: "linear-rate", opts: { startingTps: 50, finishingTps: 200 } }
      workload:
        module: workload/createVdx.js
        arguments:
          contractId: papervdx
    - label: readVdx
      description: Read Vdx benchmark
      txDuration: 60
      rateControl: { type: "fixed-load", opts: { transactionLoad: 5 } }
      workload:
        module: workload/readVdx.js
        arguments:
          assets: 10
          contractId: papervdx

monitors:
  resource:
    - module: docker
      options:
        interval: 4
        containers:
          [
            "peer0.org1.example.com",
            "peer0.org2.example.com",
            "peer0.org3.example.com",
            "orderer.example.com",
          ]
```

LISTE DE RÉFÉRENCES

(2021). *Web Page n°2022-06-13*. Resource Limits on Kaleido. Repéré à https://docs.kaleido.io/using-kaleido/resource-limits/.

(2022). Hyperledger Fabric, A Blockchain Platform for the Enterprise [Web Page]. Repéré à https://hyperledger-fabric.readthedocs.io/en/latest/index.html.

Abran, A., Laframbroise, L. & Bourque, P. (2003). *A risk assessment method and Grid for Software Engineering Measurement Programs*.

Affinidi. (2021). What are Verifiable Credentials (VCs), Demystified [Web Page]. Repéré à https://academy.affinidi.com/what-are-verifiable-credentials-79f1846a7b9.

Alketbi, A., Nasir, Q. & Abu Talib, M. (2020). Novel blockchain reference model for government services : Dubai government case study. *International Journal of System Assurance Engineering and Management*, 11(6), 1170-1191. doi : 10.1007/s13198-020-00971-2.

Allessie, D., Sobolewski, M. & Vaccari, L. (2019). *Blockchain for digital government An assessment of pioneering implementations in public services*. doi : 10.13140/RG.2.2.34874.85449.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., Caro, A. D., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W. & Yellick, J. (2018). Hyperledger fabric : a distributed operating system for permissioned blockchains [Conference Paper]. Association for Computing Machinery. Repéré à https://doi.org/10.1145/3190508.3190538.

Anwar, H. (2019). *Web Page n°2022-03-25*. Blockchain For Government : Decentralization At The Core. Repéré à https://101blockchains.com/blockchain-for-government/.

Armenia, S., Casalino, N., Gnan, L. & Flamini, G. (2021). A systems approach to the Digital Transformation of Public Administration. *PROSPETTIVE IN ORGANIZZAZIONE*. Repéré à https://prospettiveinorganizzazione.assioa.it/a-systems-approach-to-the-digital-transformation-of-public-administration/.

Arshad, J., Townend, P. & xu, J. (2011). An Abstract Model for Integrated Intrusion Detection and Severity Analysis for Clouds. *IJCAC*, 1, 1-16. doi : 10.4018/ijcac.2011010101.

Aubert, B., Babin, G. & Mignerat, M. (2001). Panorama des Systèmes d'Intégration Inter-Organisationnels. *CIRANO, CIRANO Project Reports*.

Babar, M. A. & Gorton, I. (2004). *Comparison of scenario-based software architecture evaluation methods*. Conference Proceedings présentée à 11th Asia-Pacific Software Engineering Conference (pp. 600-607). doi : 10.1109/APSEC.2004.38.

Baset, S. A., Desrosiers, L., Gaur, N., Novotny, P., O'Dowd, A. & Ramakrishna, V. (2018). *Hands-On Blockchain with Hyperledger : Building Decentralized Applications with Hyperledger Fabric and Composer*. Birmingham, UNITED KINGDOM : Packt Publishing, Limited. Repéré à http://ebookcentral.proquest.com/lib/etsmtl-ebooks/detail.action?docID=5434978.

Battilani, C., Galli, G., Arecco, S., Casarino, B., Granero, A., Lavagna, K., Varna, R., Ventura, M., Revetria, R. & Damiani, L. (2022). Business Process Re-engineering in Public Administration : The case study of Western Ligurian Sea Port Authority. *Sustainable Futures*, 4, 100065. doi : https://doi.org/10.1016/j.sftr.2022.100065.

Biswas, A. & Roy, A. (2022). *Blockchain-Based User Authentication in Cloud Governance Model* (pp. 815-825).

Buch, H. (2020). *Web Page n°2022-08-30*. Enterprise Integration and Interoperability of Blockchain. Repéré à https://www.networkcomputing.com/network-security/enterprise-integration-and-interoperability-blockchain.

Buterin, V. (2014). Ethereum Whitepaper. Repéré à https://ethereum.org/en/whitepaper/#ethereum-accounts.

Cagigas, D., Clifton, J., Diaz-Fuentes, D. & Fernández-Gutiérrez, M. (2021). Blockchain for Public Services : A Systematic Literature Review. *IEEE Access*, 9, 13904-13921. doi : 10.1109/ACCESS.2021.3052019.

Carballa, B. (2019). Alternative Data Governance Models : Moving Beyond One-Size-Fits-All Solutions. *Intereconomics*, 54(4), 222-227. doi : 10.1007/s10272-019-0828-x.

Clavin, J., Duan, S., Zhang, H., Janeja, V. P., Joshi, K. P., Yesha, Y., Erickson, L. C. & Li, J. D. (2020). Blockchains for Government : Use Cases and Challenges. *Digit. Gov. : Res. Pract.*, 1(3), Article 22. doi : 10.1145/3427097.

Commission, E. (2017). *Once-Only Principle for citizens and businesses : Policy options and their impacts*. Repéré à https://digital-strategy.ec.europa.eu/en/library/eu-wide-digital-once-only-principle-citizens-and-businesses-policy-options-and-their-impacts.

Daramola, O. & Thebus, D. (2020). Architecture-Centric Evaluation of Blockchain-Based Smart Contract E-Voting for National Elections. *Informatics*, 7(2), 16. Repéré à https://www.mdpi.com/2227-9709/7/2/16.

Davoudi, S. & Johnson, M. (2022). Public Management Review ISSN : (Print) (Preconditions of coordination in regional public organizations Preconditions of coordination in regional public organizations. *Public Management Review*. doi : 10.1080/14719037.2022.2134915.

De Filippi, P., Mannan, M. & Reijers, W. (2020). Blockchain as a confidence machine : The problem of trust challenges of governance. *Technology in Society*, 62, 101284. doi : https://doi.org/10.1016/j.techsoc.2020.101284.

Delacruz, J. (2018). *Web Page n°2022-03-24*. Blockchain is tackling the challenge of data sharing in government. Repéré à https://www.ibm.com/blogs/blockchain/2018/02/blockchain-is-tackling-the-challenge-of-data-sharing-in-government/.

Dobrolyubova, E. (2021). Measuring Outcomes of Digital Transformation in Public Administration : Literature Review and Possible Steps Forward. *NISPAcee Journal of Public Administration and Policy*, 14(1), 61-86. doi : doi:10.2478/nispa-2021-0003.

Domalis, G., Karacapilidis, N., Tsakalidis, D. & Giannaros, A. (2021). *A Trustable and Interoperable Decentralized Solution for Citizen-Centric and Cross-Border eGovernance : A Conceptual Approach* (pp. 259-270).

Dudley, E., Lin, D.-Y., Mancini, M. & Ng, J. (2015). Implementing a citizen-centric approach to delivering government services [Electronic Article]. Repéré à https://www.mckinsey.com/industries/public-and-social-sector/our-insights/implementing-a-citizen-centric-approach-to-delivering-government-services.

Eenmaa-Dimitrieva, H. & Schmidt-Kessen, M. J. (2019). Creating markets in no-trust environments : The law and economics of smart contracts. *Computer Law Security Review*, 35(1), 69-88. doi : https://doi.org/10.1016/j.clsr.2018.09.003.

Eggers, W. (2018, 2022-11-26). Taking A Citizen-Centric Approach to State Government [Electronic Article]. Deloitte. Repéré à https://deloitte.wsj.com/articles/taking-a-citizen-centric-approach-to-state-government-1521000134.

El Ioini, N. & Pahl, C. *A Review of Distributed Ledger Technologies* (pp. 277-288).

Ethereum. (2022). *Web Page n°2022-03-25*. Private Ethereum for enterprise. Repéré à https://ethereum.org/en/enterprise/private-ethereum/.

Evermann, J. & Kim, H. (2019). *Workflow Management on the Blockchain — Implications and Recommendations*.

Faruk, M. J. H., Subramanian, S., Shahriar, H., Valero, M., Li, X. & Tasnim, M. (2022). *Software Engineering Process and Methodology in Blockchain-Oriented Software Development : A Systematic Study*. Conference Proceedings présentée à 2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA) (pp. 120-127). doi : 10.1109/SERA54885.2022.9806817.

Forum, C. P. P. (2016). *Optimizing government : A White Paper on Public Sector Modernization*. Repéré à https://ppforum.ca/wp-content/uploads/2018/03/Opt_jan28_EN.pdf.

Franciscon, E. A., Nascimento, M. P., Granatyr, J., Weffort, M. R., Lessing, O. R. & Scalabrin, E. E. (2019). *A Systematic Literature Review of Blockchain Architectures Applied to Public Services*. Conference Proceedings présentée à 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD) (pp. 33-38). doi : 10.1109/CSCWD.2019.8791888.

Fridgen, G., Radszuwill, S., Urbach, N. & Utz, L. (2018). *Cross-Organizational Workflow Management Using Blockchain Technology – Towards Applicability, Auditability, and Automation*. doi : 10.24251/HICSS.2018.444.

Garcia-Garcia, L. M., Gil-Garcia, J. R. & Gómez, V. (2014). Citizen-centered e-government : towards a more integral aproach [Conference Paper]. Association for Computing Machinery. Repéré à https://doi.org/10.1145/2612733.2612791.

Gieske, H., George, B., van Meerkerk, I. & van Buuren, A. (2020). Innovating and optimizing in public organizations : does more become less ? *Public Management Review*, 22(4), 475-497. doi : 10.1080/14719037.2019.1588356. doi : 10.1080/14719037.2019.1588356.

Group, H. W. P. W. (2018, 2022-02-24). Hyperledget Whitepaper : An Introduction to Hyperledger [Electronic Article]. Repéré à https://www.hyperledger.org/wp-content/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf.

Górski, T. (2021). The 1+5 Architectural Views Model in Designing Blockchain and IT System Integration Solutions. *Symmetry*, 13(11), 2000. Repéré à https://www.mdpi.com/2073-8994/13/11/2000.

IBM. (2017). *Building trust in government : Exploring the potential of blockchains*. Repéré à https://www.ibm.com/downloads/cas/WJNPLNGZ.

Knirsch, F., Unterweger, A. & Engel, D. (2019). Implementing a blockchain from scratch : why, how, and what we learned. *EURASIP Journal on Information Security*, 2019(1), 2. doi : 10.1186/s13635-019-0085-3.

Kruchten, P. B. (1995). The 4+1 View Model of architecture. *IEEE Software*, 12(6), 42-50. doi : 10.1109/52.469759.

Le Vallée, J.-C. (2018). Cautious Optimism : Adopting Blockchain to Improve Canadian Government Digital Services [Electronic Article]. The Conference Board of Canada. Repéré à https://www.conferenceboard.ca/e-library/abstract.aspx?did=9634.

Lie, A. (2011). Coordination processes and outcomes in the public service : The challenge of inter-organizational food safety coordination in norway. *Public Administration*, 89(2), 401-417. doi : https://doi.org/10.1111/j.1467-9299.2010.01845.x.

Lo, O., Buchanan, W. J., Sayeed, S., Papadopoulos, P., Pitropakis, N. & Chrysoulas, C. (2022). GLASS : A Citizen-Centric Distributed Data-Sharing Model within an e-Governance Architecture. *Sensors*, 22(6), 2291. Repéré à https://www.mdpi.com/1424-8220/22/6/2291.

Maketa, T. (2020). *Trusted remittance services in decentralized ledger technology using a three-layered stack : a proposed model*. (Thesis). Repéré à https://espace.etsmtl.ca/id/eprint/2544/.

Martinson, P. (2019). Estonia–the Digital Republic Secured by Blockchain. *Pricewaterhouse-Coopers : London, UK*, 1-12.

Mergel, I., Edelmann, N. & Haug, N. (2019). Defining digital transformation : Results from expert interviews. *Government Information Quarterly*, 36(4), 101385. doi : https://doi.org/10.1016/j.giq.2019.06.002.

Micheli, M., Ponti, M., Craglia, M. & Berti Suman, A. (2020). Emerging models of data governance in the age of datafication. *Big Data Society*, 7(2), 2053951720948087. doi : 10.1177/2053951720948087. doi : 10.1177/2053951720948087.

Morales-Sandoval, M., Molina, J. A., Marin-Castro, H. M. & Gonzalez-Compean, J. L. (2021). Blockchain support for execution, monitoring and discovery of inter-organizational business processes. *PeerJ Comput Sci*, 7, e731. doi : 10.7717/peerj-cs.731. 2376-5992 Morales-Sandoval, Miguel Orcid : 0000-0003-1702-8467 Molina, José A Marin-Castro, Heidy M Gonzalez-Compean, Jose Luis Journal Article 2021/10/30 PeerJ Comput Sci. 2021 Sep 29 ;7 :e731. doi : 10.7717/peerj-cs.731. eCollection 2021.

Nanayakkara, S., Rodrigo, M. N. N., Perera, S., Weerasuriya, G. T. & Hijazi, A. A. (2021). A methodology for selection of a Blockchain platform to develop an enterprise system. *Journal of Industrial Information Integration*, 23, 100215. doi : https://doi.org/10.1016/j.jii.2021.100215.

Ojo, A. & Adebayo, S. (2017). Blockchain as a Next Generation Government Information Infrastructure : A Review of Initiatives in D5 Countries. Dans Ojo, A. & Millard, J. (Éds.), *Government 3.0 – Next Generation Government Technology Infrastructure and Services : Roadmaps, Enabling Technologies Challenges* (pp. 283-298). Cham : Springer International Publishing. doi : 10.1007/978-3-319-63743-3_11.

Pahlajani, S., Kshirsagar, A. & Pachghare, V. (2019). *Survey on Private Blockchain Consensus Algorithms*. Conference Proceedings présentée à 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT) (pp. 1-6). doi : 10.1109/ICIICT1.2019.8741353.

Parry, D. (2019). From Fax to Blockchain : Sharing Health Information Democratically and Safely. *Volume 264 : MEDINFO 2019 : Health and Wellbeing e-Networks for All*, 1747 - 1748. doi : 10.3233/SHTI190628.

Peristeras, V., Tarabanis, K. & Goudos, S. K. (2009). Model-driven eGovernment interoperability : A review of the state of the art. *Computer Standards Interfaces*, 31(4), 613-628. doi : https://doi.org/10.1016/j.csi.2008.09.034.

Prabhakar, A. (2021). Enterprise architecture and the importance of data interoperability [Web Page]. Repéré à https://www.intertrust.com/blog/enterprise-architecture-and-the-importance-of-data-interoperability/.

Punia, D. K. & Saxena, K. B. C. (2004). Managing inter-organisational workflows in eGovernment services [Conference Paper]. Association for Computing Machinery. Repéré à https://doi.org/10.1145/1052220.1052283.

Putnik, G. & Cruz-Cunha, M. M. (2008). Characterization and Classification of Cross-Organizational Business Processes. Dans *Encyclopedia of Networked and Virtual Organizations* (ch. 22). doi : 10.4018/978-1-59904-885-7.

Reply. (2022). *Blockchain Solutions in Public Administration*. Repéré à https://www.reply.com/en/Shared%20Documents/blockchain-in-pa.pdf.

Rondanini, C., Carminati, B., Daidone, F. & Ferrari, E. (2020). *Blockchain-based controlled information sharing in inter-organizational workflows*. Conference Proceedings présentée à 2020 IEEE International Conference on Services Computing (SCC) (pp. 378-385). doi : 10.1109/SCC49832.2020.00056.

Rosado, T., Vasconcelos, A. & Correia, M. (2019). A Blockchain Use Case for Car Registration (pp. 205-234). doi : 10.1201/9780429674457-10.

Rot, A., Sobińska, M., Hernes, M. & Franczyk, B. (2020). Digital Transformation of Public Administration Through Blockchain Technology. Dans Hernes, M., Rot, A. & Jelonek, D. (Éds.), *Towards Industry 4.0 — Current Challenges in Information Systems* (pp. 111-126). Cham : Springer International Publishing. doi : 10.1007/978-3-030-40417-8_7.

Scriber, B. A. (2018). A Framework for Determining Blockchain Applicability. *IEEE Software*, 35(4), 70-77. doi : 10.1109/MS.2018.2801552.

Scupola, A. & Mergel, I. (2022). Co-production in digital transformation of public administration and public value creation : The case of Denmark. *Government Information Quarterly*, 39(1), 101650. doi : https://doi.org/10.1016/j.giq.2021.101650.

Sturm, C., Szalanczi, J., Schönig, S. & Jablonski, S. (2019). *A Lean Architecture for Blockchain Based Decentralized Process Execution* (pp. 361-373).

Sullivan, M., Bellman, J., Sawchuk, J. & Mariani, J. (2021). Accelerated digital government [Electronic Article]. Deloitte. Repéré à https://www2.deloitte.com/xe/en/insights/industry/public-sector/government-trends/2021/digital-government-transformation-trends-covid-19.html.

Tolbert, C. J. & Mossberger, K. (2006). The Effects of E-Government on Trust and Confidence in Government. *Public Administration Review*, 66(3), 354-369. doi : https://doi.org/10.1111/j.1540-6210.2006.00594.x.

Treiblmaier, H. & Sillaber, C. (2020). A Case Study of Blockchain-Induced Digital Transformation in the Public Sector. Dans Treiblmaier, H. & Clohessy, T. (Éds.), *Blockchain and Distributed Ledger Technology Use Cases : Applications and Lessons Learned* (pp. 227-244). Cham : Springer International Publishing. doi : 10.1007/978-3-030-44337-5_11.

Upadhyaya, P., Upadhyay, S. K., Subedi, B., Subedi, B. & Gaire, A. (2018). *Revolutionizing healthcare systems of a developing country using Blockchain*. Conference Proceedings présentée à 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-6). doi : 10.1109/ICCIC.2018.8782417.

Valenta, M. & Sandner, P. G. (2017). *Comparison of Ethereum, Hyperledger Fabric and Corda*.

Verma, S. & Sheel, A. (2022). Blockchain for government organizations : past, present and future. *Journal of Global Operations and Strategic Sourcing*, ahead-of-print(ahead-of-print). doi : 10.1108/JGOSS-08-2021-0063.

Vignieri, V. & Zeinali, Z. (2022). *The role of digital transformation for inter-departmental policy coordination : assessing the contribution of the NRRP horizontal reform "public administration digitalization" to public sector dynamic capabilities innovation*.

Vingerhouts, A. S., Heng, S. & Wautelet, Y. (2020). *Organizational Modeling for Blockchain Oriented Software Engineering with Extended-i* and UML.* Conference Proceedings présentée à PoEM Workshops.

Wang, H., Wang, Y., Cao, Z., Li, Z. & Xiong, G. (2019). *An Overview of Blockchain Security Analysis*. Conference Proceedings présentée à Cyber Security (pp. 55-72).

Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A. & Mendling, J. (2016). *Untrusted Business Process Monitoring and Execution Using Blockchain* (pp. 329-347).

Wessling, F., Ehmke, C., Hesenius, M. & Gruhn, V. (2018). How much blockchain do you need ? towards a concept for building hybrid DApp architectures [Conference Paper]. Association for Computing Machinery. Repéré à https://doi.org/10.1145/3194113.3194121.

Xu, X., Weber, I. & Staples, M. (2019). *Architecture for Blockchain Applications*. Springer Publishing Company, Incorporated.

Zhang, J. (2019). *Web Page n°2022-06-06*. Enterprise Blockchain Protocols : A Technical Analysis of Ethereum vs Fabric vs Corda. Repéré à https://www.kaleido.io/blockchain-blog/enterprise-blockchain-protocols-a-technical-analysis-of-ethereum-vs-fabric-vs-corda.