Intégration de solution blockchain dans un système global de traçabilité d'une usine opérationnelle

Valentin Mullet

Thèse de Doctorat

Mention : Sciences et Technologie de l'Information et de la Communication
Spécialité : Informatique et Applications

Présentée à l'Ecole Doctorale en Sciences Technologie et Santé (ED 585)

de L'Université du Littoral Côte d'Opale

par

Valentin Mullet

Pour obtenir le grade de Docteur en Informatique

Intégration de solution blockchain dans un système global de traçabilité d'une usine opérationnelle

Soutenue publiquement le **25/11/2022**, après avis des rapporteurs, devant le jury d'examen :

Président du Jury	M. Antoine Gallais, Professeur, Univ. Polytechnique Hauts de France
Rapporteur	Mme Nathalie Mitton, Directrice de Recherche, Inria
Rapporteur	Mme, Virginie Goepp, Maître de Conférences (HDR), INSA Strasbourg
Examinateur	M. Christophe Gransart, Chargé de Recherche, Univ. Gustave Eiffel
Directeur de thèse	M. Eric Ramat, Professeur, Univ. du Littoral Côte d'Opale
Co-Directeur de thèse	M. Patrick Sondi, Maître de Conférences (HDR), Univ. du Littoral Côte d'Opale
Invité Valeo EEM	M. Jean-François Sury, Responsable suivi projet ETAPLES 4.0, Valeo EEM
Invité Région Hauts-de-France	M. Jean-Denis Collé, Responsable suivi projet ETAPLES 4.0, Région Hauts-de-France

Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution.

– Albert Einstein

À la mémoire de mon père.

Intégration de solution blockchain dans un système global de traçabilité d'une usine opérationnelle
Par ***Valentin Mullet***

Résumé :

L'industrie 4.0 implique des changements majeurs dans la gestion des processus de fabrication. L'Internet des objets et le cloud computing permettent des interactions en ligne entre des tiers, tels que des clients et des fournisseurs, avec le système de traçabilité d'une usine. La blockchain industrielle offre un paradigme qui permet de mettre en place un registre infalsifiable de transactions impliquant plusieurs partenaires, individuellement libres d'y inscrire des actions et de vérifier les actions des autres de manière ad hoc. En ce sens, elle permet de mettre en place une traçabilité fiable, transparente et directement vérifiable par chaque partenaire (fournisseur/usine, usine/sous-traitant, usine-client, etc). A l'ère des usines connectées, elle constitue une solution qui peut être mise en œuvre entre le système d'information de l'usine et les différents partenaires pour automatiser la gestion de la traçabilité. Néanmoins, le système global de traçabilité de l'usine peut comporter des composantes internes avec des implications fortes sur la confidentialité et dont le lien avec la traçabilité n'incombe qu'à l'usine et pas à ses partenaires. Dans ce contexte, la transparence apportée se ferait au détriment de la confidentialité des données. De plus, les évaluations proposées sur la blockchain portent en général sur ses performances en tant que technologie, plutôt que sur son impact global sur le système industriel, ce qui peut être un frein à son adoption.

Ce travail de thèse vise à développer une architecture et proposer des évolutions des systèmes de traçabilité pour permettre l'intégration d'une solution blockchain destinée à garantir la traçabilité entre l'usine et les partenaires sans qu'il n'y ait compromission de la confidentialité au motif de la transparence. Dans un premier temps, un état des lieux de l'usine sera présentée sous l'angle de la cybersécurité avec pour contribution une division de l'usine sous forme de périmètres en vue d'aboutir à une synthèse de bonnes pratiques adaptées à chaque périmètre en matière de sécurité informatique. Ensuite, nous définissons une approche de traçabilité centrée sur le produit soulignant l'implication des partenaires ainsi qu'une catégorisation des données selon leur criticité, ce qui permet de gérer leur accès et leur stockage au niveau des différents partenaires ainsi que dans l'infrastructure de l'usine. La présentation de l'architecture blockchain viendra dans un troisième temps avec une implémentation via la plateforme Multichain ainsi qu'une discussion sur la consommation en termes énergétique mais également en termes de stockage de la blockchain. Enfin, une modélisation à évènement discret pouvant être adaptée à n'importe quelle usine de production industrielle est également proposée afin d'analyser l'impact d'une solution blockchain sur une usine opérationnelle en fonction de différentes métriques telles que les besoins en capacité de stockage, la consommation d'énergie et l'impact environnemental.

Mots-Clés : *traçabilité ; blockchain ; usine 4.0; big data ; sécurité ; simulation*

Abstract :

Industry 4.0 involves major changes in manufacturing process management. Both the Internet of Things and cloud computing allow online interactions between third parties, such as providers, customers and suppliers, with the traceability system of a factory. The industrial blockchain offers a paradigm that makes it possible to set up an unfalsifiable register of transactions involving several partners, individually free to register actions and to verify the actions of others on an ad hoc basis. In this sense, it makes it possible to set up reliable, transparent and directly verifiable traceability by each partner (supplier/factory, factory/subcontractor, factory-customer, etc.). In the era of connected factories, it is a solution that can be implemented between the factory's information system and the various partners to automate traceability management. However, the plant's overall traceability system may include internal components with strong implications on confidentiality, and whose link with traceability lies only with the plant and not with its partners. In this context, the transparency would be obtained to the detriment of data confidentiality. In addition, the evaluations proposed on the blockchain often focus on its own performance rather than on its impact on the industrial system, which is one of the obstacles to its massive adoption.

This thesis work aims to develop an architecture and propose changes to traceability systems in order to allow the integration of a blockchain solution intended to guarantee traceability in full transparency between the factory and the partners without compromising the confidentiality. Firstly, an inventory of the factory will be presented regarding cybersecurity, thus introducing our contribution to a division of the factory in the form of perimeters which will guide our synthesis of good practices regarding the manufacturing system security. Then, we define a product-centered traceability approach underlining the implication of the partners as well as a categorization of the data according to their criticality. The presentation of the blockchain architecture will come in a third step along with its implementation using the Multichain platform as well as a discussion on its resource consumption, such as energy and storage volume related to the blockchain solution. Finally, a discrete event modeling that can be adapted to any manufacturing factory plant will be proposed in order to analyze the impact of a blockchain solution on the overall factory plant, according to different metrics such as storage volume needs, energy consumption, and environmental impact.

Keywords : *traceability ; blockchain ; factory 4.0; big data ; security ; simulation*

Remerciements

Je souhaite exprimer tous mes remerciements...

...À ma famille, plus particulièrement mes parents ainsi que ma soeur pour m'avoir encouragé et avoir été à mon écoute durant cette aventure.

...À Éric Ramat, mon directeur de thèse, pour toute l'aide apportée.

...À Patrick Sondi, mon co-directeur de thèse, pour ses conseils et sa pédagogie m'ayant permis d'avancer et de rester positif malgré les difficultés ainsi que son implication tout au long de l'élaboration de ce projet dont fait partie la thèse.

...Aux membres du jury, Nathalie Mitton et Virginie Goepp-Thiebaud pour avoir accepté de rapporter ce manuscrit ; ainsi qu'Antoine Gallais et Christophe Gransart pour avoir accepté d'examiner mes travaux de recherche.

...À Jean-François Sury, responsable du suivi du projet ETAPLES 4.0 chez Valeo EEM, pour sa participation dans la mise en place de ce projet ainsi que ses conseils et sa disponibilité tout au long de son déroulement.

...À Valeo EEM, l'Université du Littoral Côte d'Opale, la Région des Hauts-de-France ainsi que le Fond Européen de Développement Régional pour leur contribution au financement du projet ainsi que de la thèse.

...À mes collègues, qui ont pu avoir un jour une parole ou une action en faveur de mes recherches.

...À mes amis, pour leur présence et soutien à mon égard.

Merci à tous !

– Valentin Mullet

Table des matières

Table des figures

Liste des tableaux

Extraits de Code

Introduction générale 1

1.1 Contexte

Le secteur industriel a traversé plusieurs révolutions au cours des derniers siècles. La mécanisation portée par la machine à vapeur en 1765 fut la première. Puis vint la production de masse poussée par l'électricité et la production pétrolière en 1870, annonçant le développement des industries de pointe comme l'automobile et l'aéronautique. La troisième en 1969 fut celle de la production automatisée soutenue par l'électronique avec l'avènement du transistor et du microprocesseur amorçant ainsi l'arrivée des automates, robots et technologies informatiques dans les usines.

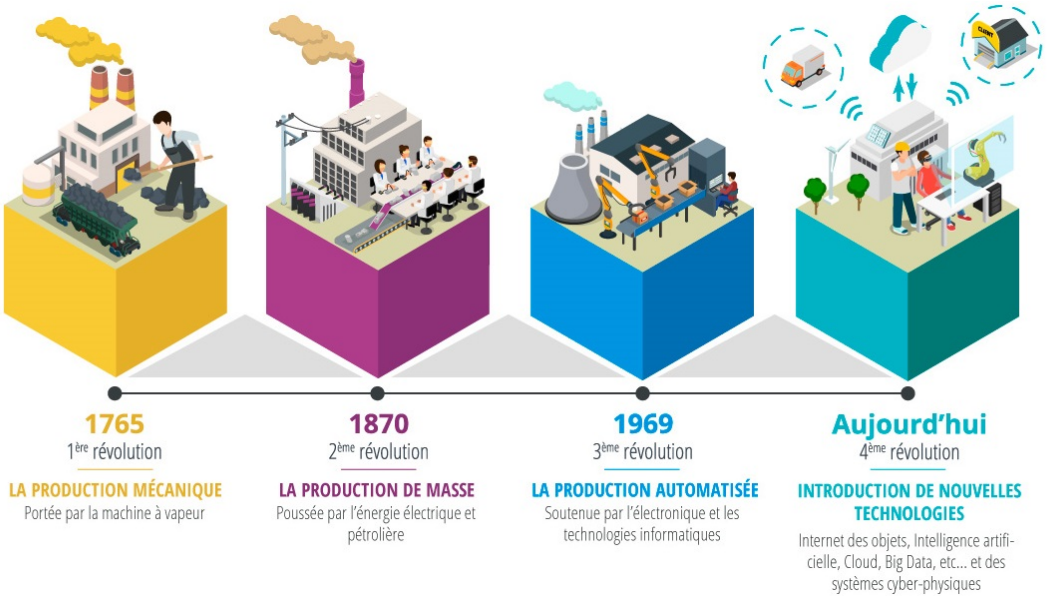


Figure 1.1 – Historique des révolutions industrielles [1]

En 2011, le gouvernement allemand a défini le concept qui représenterait la quatrième étape de l'évolution des usines traditionnelles [2] pour les rendre plus flexibles et plus adaptées aux environnements de production en constante évolution : le paradigme de l'industrie 4.0, également appelé Internet des objets industriels [3] ou Internet industriel. Techniquement, l'Industrie 4.0 vise à connecter les exploitations agricoles et les usines de fabrication à Internet afin d'améliorer leur efficacité et leur productivité (environ 15% à 20%). Cette hyperconnectivité permettra la collecte d'un grand volume de données (« Big Data ») issues de la chaîne de valeur pour de multiples usages tels que :

— échange d'informations en temps réel entre les appareils appartenant aux usines, aux fournisseurs ou aux clients
— acquisition et stockage de données pour la traçabilité et la gestion de la performance numérique
— traitement des données pour la maintenance prédictive ou la télésurveillance afin de réduire les temps d'arrêt des machines
— automatisation et réduction des stocks
— amélioration des niveaux de service et de la qualité des produits

La première étape de la mise en oeuvre de l'industrie 4.0 consiste à récupérer le plus d'informations possibles afin de les numériser, pour ensuite pouvoir les traiter et les agréger avec d'autres informations. Cette transformation numérique s'exprime à travers des objets hautement innovants. L'idée est de connecter ensemble les opérateurs, les machines ou équipements dans le but de s'adapter aux demandes des clients et d'apporter une réponse immédiate à la moindre variation. L'objectif est donc de parvenir à personnaliser une commande malgré une production de masse.

De plus, le travail s'entend différemment en raison de la mise en œuvre de processus combinés entre les machines et les opérateurs. Les enjeux vont porter sur la formation de ces salariés aux nouvelles technologies autant que sur les moyens techniques eux-mêmes.

Les efforts devront également porter sur la sécurité telle que la protection des données, les dispositifs anti-intrusion ou encore la lutte contre les cyberattaques. Les ressources devront ainsi se répartir équitablement entre la formation des opérateurs, l'investissement dans les machines ainsi que la protection des données.

Longtemps en réponse à des contraintes réglementaires ou commerciales, la traçabilité est devenue un vecteur d'amélioration de la performance dans un contexte de transformation digitale. Les nouvelles technologies de l'industrie 4.0 sont fondées sur l'exploitation des données collectées et historisées par la traçabilité industrielle.

Avec la digitalisation croissante des industries, la traçabilité des données intervient dorénavant dans toutes les unités de production de l'usine. Toutes les étapes, de la conception au suivi, en passant par les processus de production, sont étroitement liées à la récupération, au traitement et à l'analyse des données. Autant d'indicateurs émanant des cycles de développement et devant être interprétés, exploités. En d'autres termes, la traçabilité devient un outil de gestion des risques permettant d'aider à la résolution des litiges, l'analyse des produits défectueux et de l'inefficacité tout au long de la production ainsi qu'à la répartition des responsabilités. Il est important de souligner qu'elle ne tombe pas sous la responsabilité d'une seule personne ou entreprise, elle inclut plusieurs acteurs de la chaîne d'approvisionnement notamment les fournisseurs et les clients qui ne font pas directement partie de l'usine.

La transition vers l'usine 4.0 mène à un objectif clair : mieux comprendre les données générées, savoir les exploiter pour innover et améliorer la productivité finale.

Les outils de traçabilité, développés notamment grâce à l'automatisation industrielle, répondent aux besoins liés à un processus de production, tels que :

— localiser les en-cours (rôle habituellement joué par les fiches suiveuses) ou les équipements pour déclencher des demandes automatiques de réapprovisionnement, assurer des flux tendus de produits et de marchandises
— automatiser l'apport de l'information directement sur le terrain
— identifier des contenants sur des chaînes d'assemblage
— suivre une flotte de véhicules, l'acheminement de matériel ou le parcours entre deux points géographiques
— suivre et tracer des productions dans l'industrie agroalimentaire

Les volumes de données générés par ces processus de traçabilité sont souvent au-delà des capacités de stockage de l'usine en raison de la production de masse d'où la nécessité d'externaliser le stockage vers le cloud en connectant les systèmes informatiques de l'usine à Internet. Dans un rapport publié en 2018 [4], l'entreprise IDC (« International Data Corporation ») , en partenariat avec Seagate, proposait dans son livre blanc une étude comparant la quantité de données (appelées « datasphère ») générées par différents secteurs professionnels (Figure 1.2).

La part de l'industrie manufacturière est la plus importante avec *3584 Eo (*10^{18} *octets)* grâce à sa maturité, à l'investissement de l'industrie dans l'Internet des Objets et à sa capacité à produire 7j/7 & 24h/24.

L'ensemble de ces processus et leur gestion, identifiés en tant que « système de traçabilité de l'usine » sera donc au coeur des problématiques et des contributions qui seront formulées dans le cadre de cette thèse.

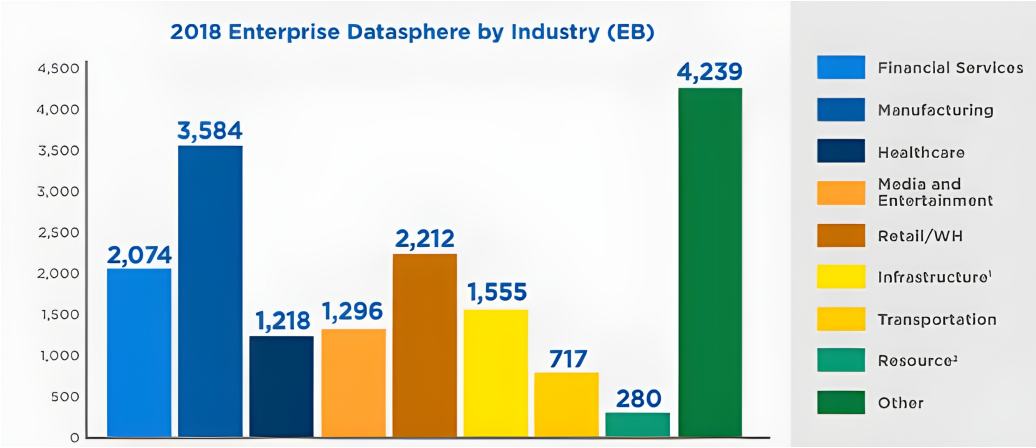


Figure 1.2 – Datasphère des entreprises classées par industrie en 2018 [4]

1.2 Problématique

Dans cette thèse, nous nous intéressons au système de traçabilité de l'usine et à son organisation.

A l'ère de l'industrie 4.0, alors que les moyens de production sont de plus en plus digitalisés, la sécurité des données prend une importance inédite. Cette industrie du futur, dans laquelle équipements physiques, systèmes de contrôle et d'information sont interconnectés, aspire à l'excellence opérationnelle. Entre économies d'énergie et de matières, maintenance prédictive, gain de productivité, optimisation des processus et automatisation des tâches à faible valeur ajoutée, les bénéfices de cette évolution sont indiscutables.

Mais comme toute autre opportunité, l'avènement de l'industrie 4.0 s'accompagne de risques, tels que l'exposition de l'usine aux cyberattaques, ce qui compromet donc par extension la sécurité des données notamment celles de traçabilité.

Pour les entreprises, la « supply chain » (ou chaîne d'approvisionnement) est un domaine essentiel qui consiste à suivre la provenance et le parcours des produits de bout en bout. Cependant, cette dernière souffre d'opacité et d'un manque d'enregistrement des données. Pourtant, la qualité ou l'origine des produits sont des indicateurs mis en lumière par les entreprises pour garantir l'image de leurs marques et répondre aux besoins de leur clients.

Le besoin de visibilité étant plus prégnant, les entreprises sont dans l'obligation d'améliorer la transparence et la traçabilité de leur chaîne d'approvisionnement. Dans ce contexte, la notion de transparence se concentre sur la cartographie de l'ensemble de la chaîne d'approvisionnement et la traçabilité examine les lots individuels de composants ou les bons de commande au fur et à mesure de leur progression dans la chaîne d'approvisionnement.

Un autre point de vigilance pour ces entreprises est la lutte contre la contrefaçon, qui trouve dans ce système de traçabilité totale un outil efficace pour garantir l'authenticité des produits. L'opacité présente dans les chaînes d'approvisionnement provient aussi de la vétusté du système, qui repose encore trop sur des registres papier imposant des manipulations manuelles, sources d'erreurs et de perte de temps.

La digitalisation des processus sur les chaînes d'approvisionnement génère de nombreuses données à stocker, à certifier et à exposer aux clients, partenaires, consommateurs et régulateurs. Dans ce modèle, l'intégrité des données ne saurait être remise en cause et leur accès libre doit être garanti. L'objectif est de maintenir un registre certifié avec l'intégralité des données sur toute la chaîne, ce qui constitue précisément le service rendu par la blockchain. En effet, la blockchain est une base de données décentralisée et infalsifiable, même par un administrateur. Ce concept permet de supprimer les intermédiaires de confiance et les services d'audit, ce qui améliore encore plus le retour sur investissement à envisager. La blockchain apporte aussi aux utilisateurs une réponse outillée aux besoins de transparence sur les produits : provenance, constitution, transformation, transport.

Ce besoin de transparence ne doit cependant pas se faire au détriment d'un autre aspect qui est la confidentialité. En effet, les informations générées par les processus de production peuvent contenir des informations à caractère critique relevant du savoir-faire de l'usine et ne peuvent donc être exposées sous peine de mettre en péril la compétitivité de l'entreprise.

Un équilibre doit donc être trouvé afin de pouvoir garantir au sein du système de traçabilité la transparence, l'intégrité, ainsi que le respect de la confidentialité.

De plus, la blockchain étant une technologie innovante, cette dernière possède également ses propres contraintes et défis à relever tels que la consommation énergétique ainsi que l'impact environnemental, auxquels s'ajoutent les coûts liés à sa mise en place ainsi que les volumes importants en termes de données de traçabilité à stocker. Ces aspects ne peuvent être négligés et amènent à la nécessité pour l'usine de disposer d'un outil d'évaluation permettant de simuler, en amont de tout investissement, les impacts de l'évolution de son système de traçabilité vers une telle solution.

En somme, les travaux relatés tout au long de mon mémoire viseront à répondre aux objectifs suivants :

1. Faire évoluer le système de traçabilité de l'usine vers la blockchain en couvrant la chaîne d'approvisionnement, du fournisseur jusqu'au client
2. Garantir au sein de ce système les aspects d'intégrité, la transparence, ainsi que la confidentialité des données critiques
3. Parvenir à évaluer les impacts de cette évolution du système de traçabilité vers la blockchain au moyen d'un outil de simulation qui pourrait servir également d'outil d'évaluation pour la planification des évolutions

1.3 Organisation du mémoire

Afin de mettre en évidence les contributions issues de l'ensemble de mes travaux au regard des problématiques susmentionnées, la présentation de ce mémoire sera déclinée sous la forme de plusieurs chapitres.

Le Chapitre 2 abordera la question de la cybersécurité et de son impact sur la traçabilité en industrie 4.0. Cette vision globale de l'usine du point de vue de la cybersécurité permettra de mieux appréhender les concepts détaillés dans les chapitres ultérieurs. Une synthèse des bonnes pratiques relatives à la cybersécurité en industrie 4.0 sera proposée en seconde partie de ce chapitre.

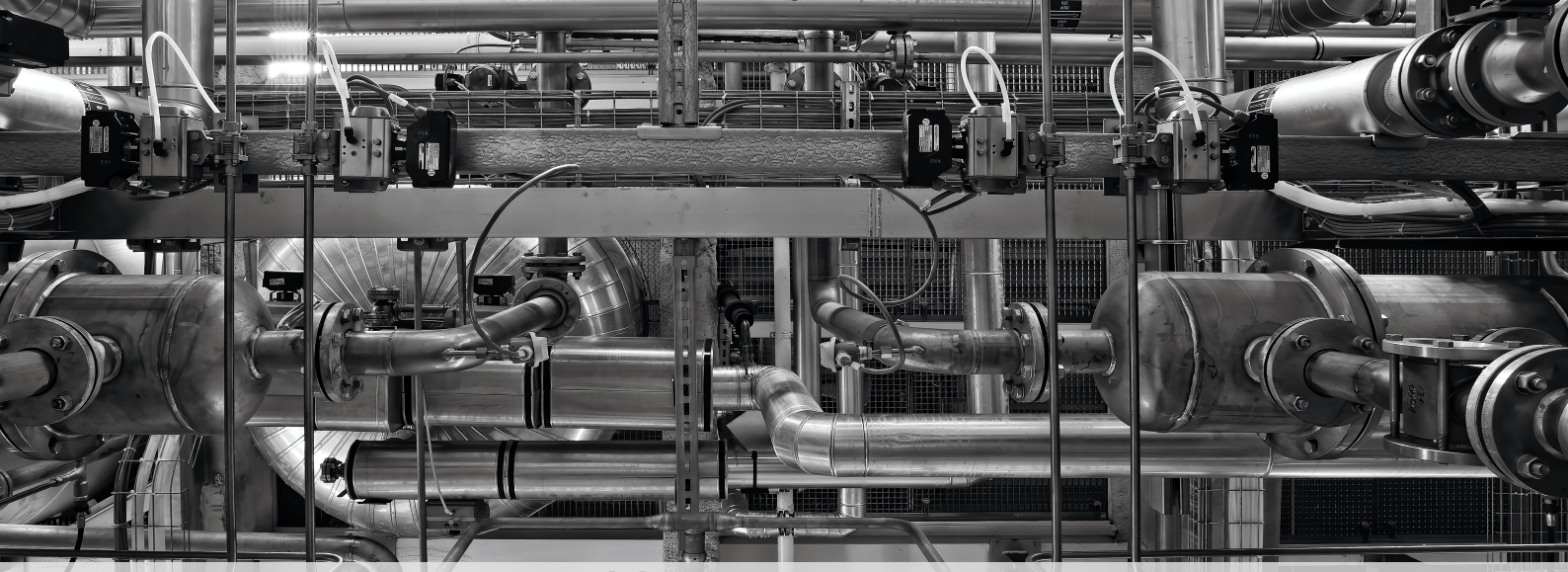
Afin d'affiner ce concept large que représente la traçabilité, le Chapitre 3 se concentrera sur la notion de traçabilité orientée produit. Un panorama des normes dans l'usine 4.0 sera proposé, ainsi que l'approche que nous proposons pour la caractérisation des données de traçabilité, leur collecte ainsi que leur traitement.

Au terme de cette entrée en matière, le moment sera venu de présenter la technologie blockchain dans le Chapitre 4, ainsi que la manière dont elle pourrait être intégrée à la mise en oeuvre de la traçabilité dans une usine ayant vocation à suivre les préconisations de l'industrie 4.0. Cette présentation s'accompagnera également d'une description de l'architecture que nous proposons à cette fin, ainsi que son implémentation avec un outil dédié à la réalisation de blockchain, en l'occurrence Multichain.

Dans le but d'évaluer les performances des solutions proposées, le Chapitre 5 présentera une série de modèles basés sur la simulation à évènements discrets permettant de créer un simulateur reproduisant le fonctionnement d'une usine qui intègre nos propositions. L'évaluation de différents scénarios de production à partir d'un certain nombre de configurations d'usine permettront de mettre en évidence l'impact des mécanismes basés sur la blockchain proposés pour la traçabilité sur le fonctionnement de l'usine au regard de critères d'intérêt tels que la volumétrie des données de traçabilité, l'impact de leur stockage, l'impact sur la consommation énergétique de l'usine ou encore l'impact environnemental de l'usine résultant du recours à ces solutions.

Enfin, le Chapitre 6 formulera une synthèse des travaux réalisés ainsi que les perspectives ouvertes par ces derniers afin de clôre ce manuscrit.

Etat des lieux de l'usine 4.0



2 Cybersécurité et Traçabilité dans l'usine 4.0 : État des lieux et perspectives

L'industrie 4.0 représente une révolution dans le monde industriel en introduisant des technologies de rupture telles que l' Internet des Objets (IoT) et le « cloud computing » au sein de l'usine. L'automatisation accrue qui en résulte et l'amélioration de la synergie de production entre les stocks, les chaînes d'approvisionnement et les demandes des clients s'accompagnent des menaces et des attaques provenant d'Internet. Malgré une littérature abondante sur le sujet de la cybersécurité, de nombreux acteurs industriels commencent à peine à prendre conscience de l'impact de la cybersécurité sur la préservation de leur activité.

Ce chapitre sera dédié à la présentation des concepts et des aspects pratiques liés à la cybersécurité dans une usine de fabrication de l'industrie 4.0.

2.1 Etat de l'art de la cybersécurité dans l'industrie 4.0

Pour créer cet environnement de production intelligent, des technologies innovantes sont nécessaires pour gérer les communications autonomes entre tous les appareils industriels de l'usine et Internet. Ces technologies [5] incluent l'IoT, le cloud computing, le big data, le jumeau numérique, la réalité augmentée, l'impression 3D, l'intelligence artificielle, les Systèmes Cyber-Physiques (CPS) de nouvelle génération comme l'illustre la Figure 2.1.

De plus, l'industrie 4.0 encourage l'application de ces technologies pour permettre des architectures de communications distribuées (de type Peer-To-Peer (P2P)), au lieu de s'appuyer uniquement sur des solutions cloud standards ou d'autres architectures centralisées [6].

L'intégration de ces équipements hétérogènes dans le cyberenvironnement industriel rend la prise en compte de la cybersécurité obligatoire dans la stratégie de conception des entreprises qui cherchent à adopter le paradigme de l'Industrie 4.0. Malgré les nombreuses améliorations apportées par cette révolution dans l'efficacité des usines de fabrication, les failles de cybersécurité entraîneraient des impacts critiques sur le modèle commercial et une perte de compétitivité [7].

Plusieurs enquêtes montrent que seulement 16% des entreprises sont prêtes à relever les défis de la cybersécurité[8]. Parmi les raisons invoquées, il y a le manque de normes de référence précises et le manque de

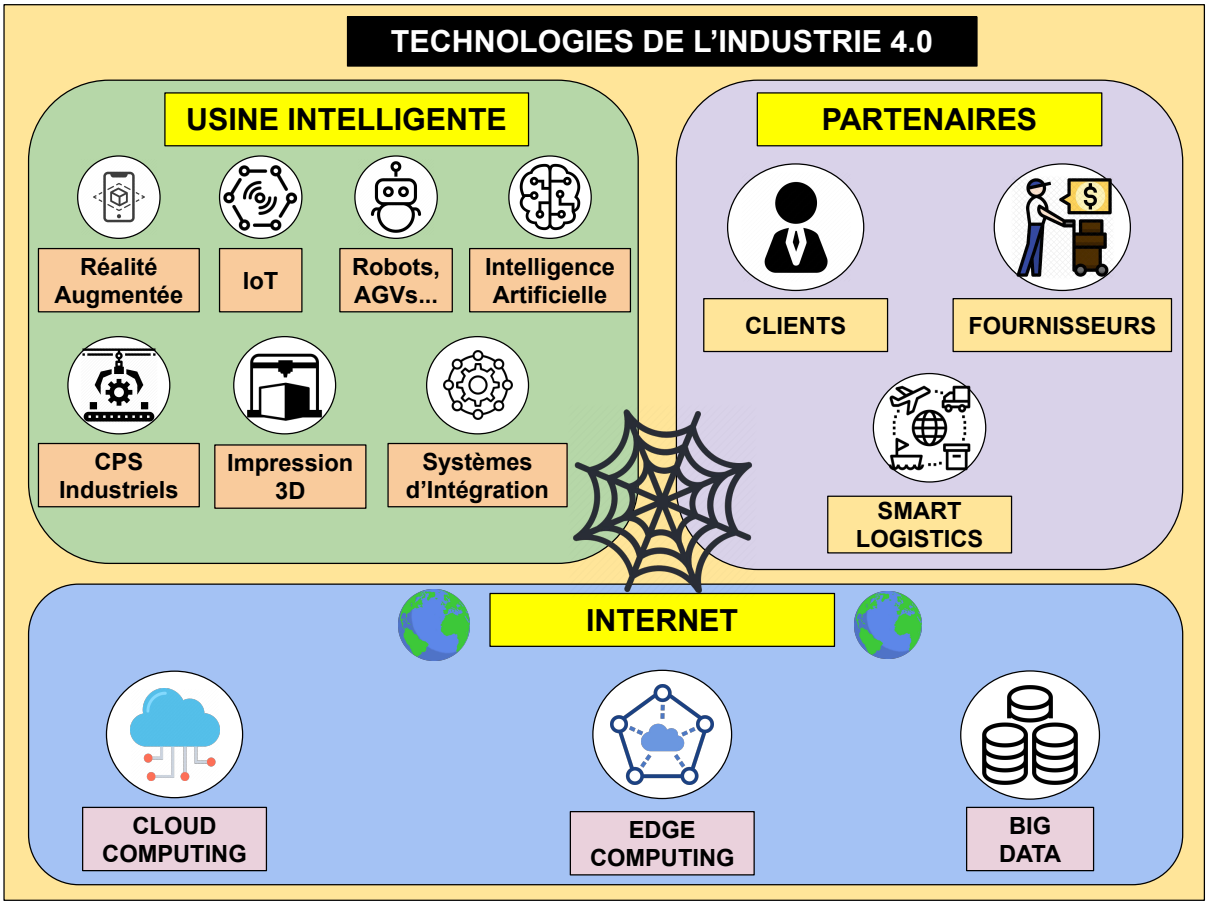


Figure 2.1 – Technologies de l'Industrie 4.0

compétences managériales et techniques pour les comprendre et les mettre en œuvre. Plusieurs organisations travaillant sur des lignes directrices et des normes aident les entreprises à comprendre quel schéma elles doivent utiliser pour renforcer leur sécurité et la rendre conforme. Parmi ces organismes, pour n'en citer que quelques-uns, on peut trouver : ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information) en France, ENISA (Agence de l'Union européenne pour la cybersécurité) dans l'Union européenne et NIST (National Institute of Standards and Technology) aux États-Unis d'Amérique.

Une description plus détaillée des concepts et applications de l'industrie 4.0 est proposée dans [9], ainsi que certaines attaques de cybersécurité récentes observées dans plusieurs usines de fabrication dans le monde. Les auteurs proposent également des contre-mesures pour faire face à un large éventail de risques de cybersécurité. Ces risques doivent être dans un premier lieu identifiés ainsi que les facteurs qui en sont à l'origine [10]. Une autre enquête proposée dans [11] porte sur la cybersécurité où les auteurs analysent en particulier le passage d'un système de contrôle industriel ICS (Industrial Control Systems) d'une usine autonome à un environnement basé sur le cloud, tout en se concentrant sur les solutions d'apprentissage automatique. Une enquête récente plus spécifique [12] a passé en revue 262 articles concernant tous les aspects de la sécurité de l'industrie 4.0. Outre une revue systématique de la littérature, leurs principales propositions portent sur les opportunités apportées par le Fog computing dans ce domaine. Toutes ces références sont des enquêtes classiques qui visent à fournir une vue complète de la littérature au lecteur.

2.2 Etat des lieux de l'usine

Afin de mieux comprendre la notion de cybersécurité dans un environnement industriel, il est important de se rappeler que l'usine contient plusieurs départements avec des besoins et des processus de travail

différents. Nous proposons d'abord une subdivision de l'usine en six zones génériques que nous appelons les périmètres.

2.2.1 Définition des périmètres

Manufacturing-Production Il s'agit de la zone principale d'une usine où se trouvent les lignes de production, chacune dédiée à certaines des nombreuses étapes nécessaires à la fabrication des produits finaux. Les appareils présents dans ce périmètre appartiennent à deux groupes : Systèmes de Contrôle Industriels (ICS) et Systèmes Cyber-Physiques (CPS). ICS rassemble principalement les composants de contrôle qui agissent ensemble pour atteindre un objectif industriel [8].

Ils sont utilisés depuis la seconde moitié du 20^{e} siècle [13]. A l'intérieur de cette catégorie, on peut citer Automate Programmable (PLC), Unité terminale distante (RTU)[14], IED (Intelligent Electronic Devices). Pour interagir avec le contrôleur matériel et obtenir les données recueillies par l'environnement ICS, les administrateurs disposent d'une Interface Homme Machine (IHM), qui est également utilisée pour afficher l'état des appareils[15].

Les CPS sont liés à tout ce qui intègre le calcul, la mise en réseau ou les processus physiques. Ils permettent une interaction entre le monde numérique et le monde physique. A titre d'exemple, une ligne de fabrication peut être considérée comme un CPS [8]. En effet, ils utilisent des capteurs et autres systèmes embarqués pour collecter des données issues de processus physiques. Plusieurs auteurs [8, 16], montrent que l'ICS est un domaine d'application du CPS.

Zones logistiques La logistique industrielle se différencie de la logistique traditionnelle de par son adaptation à la gestion des flux de production, élargissant ainsi son champ d'action. Son périmètre est constitué de multiples structures telles que des ateliers et des entrepôts où sont utilisés des chariots élévateurs. Dans ce domaine où la mobilité est un point clé, les appareils sans fil sont majoritairement utilisés (lecteurs de codes-barres, tablettes).

Supervision active La supervision active représente l'activité tertiaire, qui couvre la partie bureau de l'usine. Il regroupe des services tels que la comptabilité, les ventes, les ressources humaines (RH), le système informatique (SI) local, etc. Les appareils les plus courants sont les ordinateurs de bureau, les smartphones, les écrans, les imprimantes, etc.

Recherche et Développement La recherche et le développement désignent les laboratoires et les bureaux liés à l'innovation et au développement de nouveaux produits ou services. Il s'agit de la première étape du processus de développement, ce qui signifie que certains équipements sont nouveaux ou expérimentaux. Les ingénieurs utilisent des appareils plus puissants que les ordinateurs de bureau courants et nécessitent généralement une configuration spécifique.

Zones de vie L'espace de vie correspond aux lieux de rassemblement des salariés (cantine, salles de réunion, salles de repos, etc.). Dans ces zones, on trouve des téléphones VoIP, des tablettes, des écrans, etc.

Zones externes La zone externe fait référence à tout ce qui se trouve à l'extérieur de l'usine. Elle comprend les lieux physiques tels que les parkings, ainsi que les lieux virtuels comme Internet (cloud, etc.).

2.2.2 Intéractions entre les périmètres

Les périmètres jouent un rôle essentiel dans le processus de fabrication industrielle et leur interaction au niveau du SI est obligatoire comme on peut le voir sur le Tableau 2.1.

Tableau 2.1 – Intéractions informatiques entre les périmètres de l'usine

Présence d'intéractions informatiques						
	Manufacturing-Production	**Logistique**	**Supervision active**	**Recherche et Développement**	**Zone de vie**	**Zone externe**
Manufacturing-Production	X	X	X	X		
Logistique	X	X	X	X	X	
Supervision active	X	X	X	X	X	X
Recherche et Développement	X		X	X		
Zones de vie			X		X	X
Zones externes			X		X	X

2.2.3 Périmètres réseaux dans l'usine

Dans le domaine du manufacturing, il existe plusieurs types de systèmes de contrôle industriels ICS [15]. Ces ICS sont classés en deux couches : la couche de contrôle physique et la couche de contrôle logique. Trois sous-réseaux existent dans ce périmètre : Système de contrôle, supervision et d'acquisition de données (SCADA), Systèmes de Contrôle Distribués (DCS) et automates (PLC - Programmable Logic Controller)/Capteurs/Protocoles. La Figure 2.2 montre ces différents réseaux avec les équipements et les protocoles de communication utilisés.

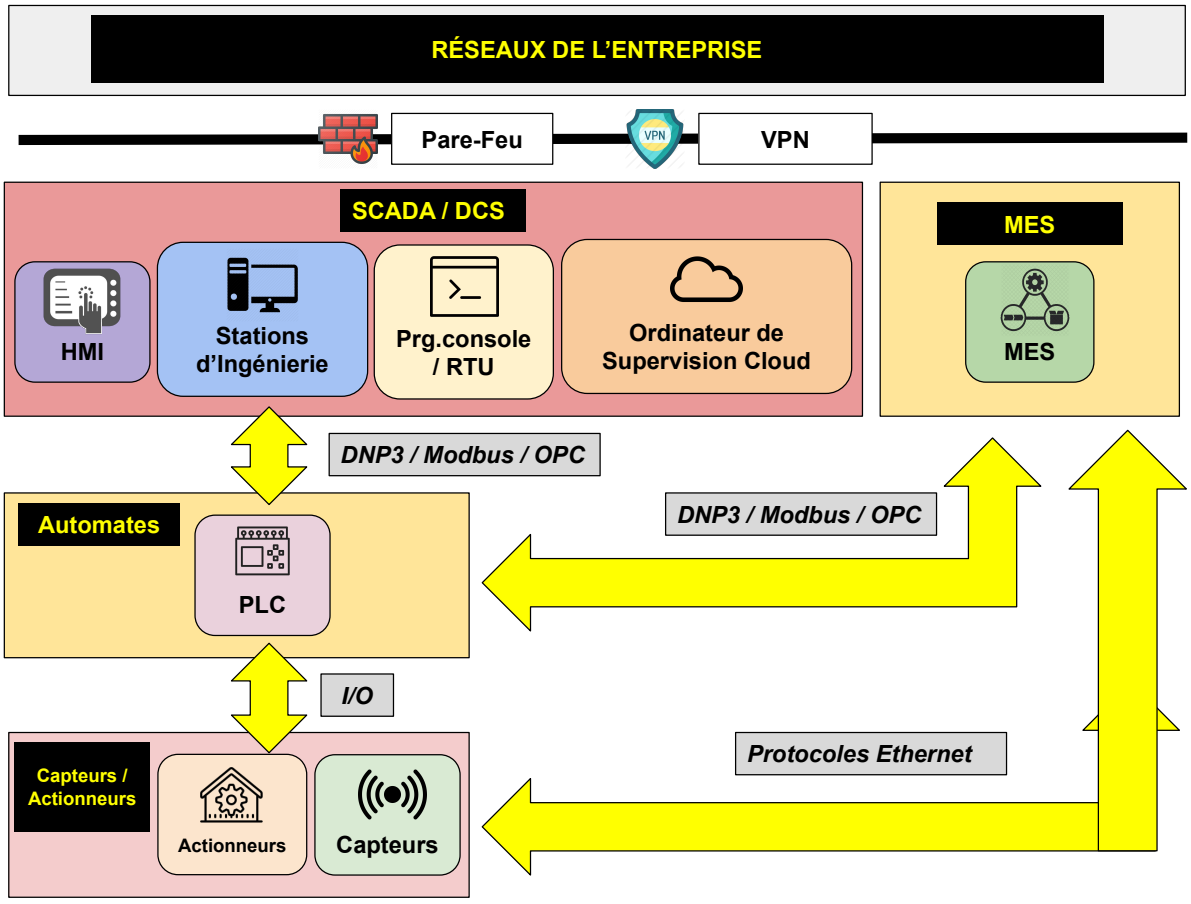


Figure 2.2 – Vue d'ensemble des réseaux du périmètre Manufacturing-Production

SCADA Le SCADA permet la supervision de l'acquisition des données et la surveillance du système de production. Il est également utilisé pour le contrôle à distance des sites par les administrateurs adoptant un système de contrôle centralisé.

DCS Un DCS est composé de contrôleurs autonomes installés dans une unité de fabrication ou de production. Le système DCS utilise ces contrôleurs pour superviser et surveiller une unité à distance. La différence entre DCS et SCADA est que le SCADA peut gérer des systèmes à plusieurs endroits contrairement au DCS qui est limité à un seul endroit.

PLC / Capteurs / Protocoles Tous les dispositifs de contrôle tels que les automates programmables PLC, les capteurs ou les protocoles (protocoles de réseau distribué DNP3 / Modbus) se trouvent dans ce dernier sous-réseau, qui fait partie de la couche de contrôle physique.

Un automate est l'interface logique entre les systèmes SCADA et DCS. L'automate est censé recevoir des commandes de contrôle et renvoyer l'état des capteurs. Pour établir la connexion entre l'automate et le SCADA [17], des protocoles de communication spécifiques ont été conçus par les fabricants de systèmes industriels (DNP3/Modbus/OPC).

2.2.4 Contrôles des accès et surveillance des périmètres

Dès lors qu'il y a des interactions entre périmètres, il est nécessaire d'effectuer des contrôles d'accès précis en fonction des informations échangées, qui peuvent être très sensibles. Un autre aspect important de la sécurité est la surveillance, qui permet aux utilisateurs d'être avertis lorsqu'une faille de sécurité apparaît ou lorsque le système est sous une attaque. Le Tableau 2.2 montre les différents types de contrôles d'accès disponibles dans la littérature pour les périmètres définis ainsi que certaines solutions de surveillance applicables aux usines de fabrication.

Tableau 2.2 – Périmètres de l'usine en industrie 4.0

Périmètres de l'usine	*Equipements utilisés*	*Méthodes d'accès*	*Méthodes de surveillance*	*Types de réseaux*
Manufacturing-Production	PLC,RTU, IED, IHM	identifiant / mot de passe ou acces badge (humain), identification périphérique via clés publiques & privées (machines)	Surveillance vidéo, vérification traçabilité, logging, administration à distance (SNMP ou intégré)	SCADA, DCS
Logistique	Terminaux mobiles (tablettes, lecteurs code-barres, portables)	accès badge (pour humain), clés publiques / privées (machines)	Surveillance vidéo, contrôle des informations sytèmes, logging, administration à distance (IDS/IPS probes)	Sans-Fil
Supervision Active	Bureautique, imprimantes, portables, smartphones	identifiant/ mot de passe, accès badge (for human), identification réseau ou système (pour terminaux)	contrôles systèmes, logging, administration à distance (SNMP)	Bureau
Recherche et Développement	Stations de travail, périphériques expérimentaux	accès badge ou biométrique (humain), réseau et vérifications sytèmes (terminaux)	vérifications sytèmes, logging, administration à distance (SNMP)	LAB
Zones de vie	Téléphonie VoIP, écrans sans fil, tablettes	pas de contrôle spécifique (pour humain), contrôle des informations sytèmes (pour terminaux)	Surveillance vidéo, administration à distance (SNMP or built-in)	Bureau
Zones externes physiques	Caméras	Accès badge (humain), contrôles systèmes et réseaux (terminaux)	Logging, surveillance vidéo, administration à distance (SNMP, etc.)	CAM
Zones externes virtuelles	Fournisseurs de services Cloud	login/password (human), system & Règles pare-feu (périphériques)	Logging, surveillance du traffic, Sondes des systèmes de prévention (IPS) ou de détection (IDS) des intrusions	Internet

2.3 Vulnérabilités, risques et menaces de cybersécurité en industrie 4.0

Les usines de l'industrie 4.0 présentent des failles de sécurité, comme la plupart des systèmes organisationnels. L'interconnexion entre les équipements rend la sécurité plus complexe, et apporte des vulnérabilités inattendues[18, 19]. Dans le secteur industriel, la fabrication est la plus ciblée par les attaques de sécurité et le nombre de menaces augmente chaque année [8]. La première partie de cette section portera sur la définition et la description des vulnérabilités, menaces et risques.

2.3.1 Définitions

Vulnérabilités

Dans le domaine informatique, les vulnérabilités sont définies par [8] comme des faiblesses qui pourraient être exploitées par des pirates pour compromettre le système. Plus précisément, ces faiblesses peuvent être soit dans le système, soit dans les procédures de sécurité, soit dans les contrôles internes. Elles peuvent être classées en trois catégories qui sont : les vulnérabilités d'accès à distance, les vulnérabilités logicielles et les vulnérabilités du Réseau local (LAN) ou Réseau local sans fil (WLAN).

Menaces

Une menace de cybersécurité (aussi appelée cybermenace) est définie par [8] comme toute circonstance ayant un impact sur les opérations organisationnelles, les actifs, les individus, d'autres organisations ou la Nation par le biais d'un système d'information via un accès non autorisé, la destruction, la divulgation, la modification d'informations. Plusieurs paramètres doivent être considérés pour analyser une cybermenace :

— L'origine de l'attaque (interne ou externe)
— L'objectif
— La couche cyber affectée, comprenant la couche d'exécution (capteurs, actionneurs), la couche de transport de données (réseau) et la couche applicative (stockage de données utilisateur)

Dans le cadre de l'Industrie 4.0, les principales catégories de cybermenaces suivantes ont été identifiées [8], pour n'en citer que quelques-unes :

— Attaques directes sur les accès externes
— Attaques indirectes avec un fournisseur de services auquel un accès externe a été accordé
— Vecteurs d'attaque inconnus (ou « zero-day exploits »)
— Logiciels malveillants
— Intrusion dans les réseaux voisins

Risques

Selon [8], un risque est le niveau d'impact sur les opérations organisationnelles, les actifs ou les individus résultant de la conséquence potentielle d'une menace et de sa probabilité d'occurrence. En cybersécurité, les risques sont identifiés par la perte de certaines caractéristiques que sont la disponibilité, l'intégrité, la confidentialité et l'authentification.

Disponibilité Les attaques ciblant la disponibilité visent à rendre le système incapable d'effectuer ses tâches habituelles en le surchargeant. Leur cible peut être l'équipement ou l'accès au réseau associé en le perturbant. Les types les plus courants sont les attaques Déni de Service distribué (DDoS), qui tentent de saturer la bande passante ou d'autres ressources du système, le rendant incapable de réagir. Certaines attaques affectent également le réseau, et plus précisément les opérations de routage (« grey hole », « black hole », attaques relais).

Intégrité L'intégrité consiste à maintenir l'exactitude et l'exhaustivité de données. Les menaces associées sont similaires au sabotage. Elles visent à modifier les protocoles de communication industriels ou le trafic réseau. Un problème majeur est que la plupart de ces protocoles sont hérités, ce qui souligne que leur conception n'incluait pas de considérations de sécurité.

Certaines attaques courantes contre l'intégrité sont les attaques « Man-In-the-Middle », qui consistent à altérer et relayer les communications entre deux entités alors que ces dernières pensent qu'elles sont directement connectées.

Confidentialité Les menaces liées à la confidentialité consistent à accéder ou à voler des données sensibles liées à des processus industriels, configurations, clients ou à l'administration. Elles peuvent être qualifiées de cyberespionnage et réalisées dans différents contextes tels que l'analyse passive du trafic réseau, l'injection active de code dans les applications opérationnelles pour obtenir des informations d'identification de sécurité ou la corruption des mesures de contrôle.

Authentification Cela concerne les menaces qui tirent parti des défauts de conception ou des vulnérabilités logicielles pour élever les privilèges et accéder aux ressources protégées. Ces attaques utilisent des techniques d'ingénierie sociale telles que le phishing ou des chaînes de courriers indésirables pour collecter des informations stratégiques. Les erreurs de configuration conduisant à des accès inadaptés tant au niveau physique que logique peuvent conduire aux mêmes problèmes de sécurité.

2.3.2 Impacts métiers

Les menaces liées à la cybersécurité sont un sujet sérieux, et elles mettent à l'épreuve les capacités des entreprises les plus avancées. L'année 2017 a été une année charnière car trois attaques à grande échelle se sont produites dans le monde [20] :

— « NotPetya » s'est répandu dans 65 pays et a causé 892 millions de dollars de dommages
— « Bad Rabbit » qui ciblait les infrastructures critiques
— « WannaCry » s'est propagé à 150 pays et a causé 8 milliards de dollars de dommages

Les entreprises infectées par ces attaques ont vu leur production interrompue pendant une longue période et n'ont pas pu mener à bien leurs activités de manière efficace. L'impact de ces menaces n'est pas seulement technique ou financier. Il peut affecter les relations avec les partenaires et entraîner des conséquences judiciaires. Le Tableau 2.3 rapporte l'analyse trouvée dans [21] des impacts commerciaux filtrés par risques. Tous les risques sont couverts par une gamme de menaces et on peut observer que le nombre d'impacts sur l'entreprise est supérieur au nombre de menaces. Pour chaque menace unique, plusieurs impacts métiers se produisent ce qui suggère qu'aucune menace ne doit être sous-estimée et qu'une politique en termes d'investissements au niveau de la cybersécurité est nécessaire [22].

Tableau 2.3 – Impacts métiers liés aux menaces de sécurité et filtrés par risques [21]

Risques / Perte de	*Attaques / Menaces*	*Impacts métiers*
Disponibilité	Déni de service DDoS	Perte de productivité
		Violation des accords commerciaux avec les clients
		Dégradation de la qualité des pièces
		Vol de service
Intégrité	Sabotage des infrastructures critiques, machines ou composants	Endommagement des machines de travail
		Dégradation de la qualité des produits
		Violation des standards et de la règlementation dans le domaine de la sûreté
		Violation des accords commerciaux avec les clients au sujet des produits
Confidentialité	Vol de secrets industriels, cyberespionnage	Diminution de l'avantage compétitif de l'entreprise
		Dommage envers l'image et la réputation de l'entreprise
		Violation des accords commerciaux avec les partenaires au sujet des données

2.3.3 Menaces de cybersécurité majeures en industrie 4.0

Les principales menaces de sécurité auxquelles est confrontée une usine de l'industrie 4.0 peuvent être classées dans les catégories suivantes [18] :

Cyberespionnage

En raison de processus commerciaux intelligents et connectés, l'industrie 4.0 est vulnérable au cyberespionnage. Des groupes bien organisés de cybercriminels ont fait de l'industrie 4.0 leur cible favorite pour voler des informations sensibles et de la propriété intellectuelle. L'un de ces groupes est le groupe « Black Vine » qui

cible principalement les industries de l'aérospatiale, de l'énergie et de la santé. Le vol de données d'entreprise et de produits devient très courant, en particulier les logiciels et les fonctionnalités faciles à copier. Dans l'industrie 4.0, la coopération de plusieurs partenaires, tels que les fournisseurs du réseau, facilite la tâche de ces criminels car leurs attaques peuvent emprunter de nombreuses voies et se propager très rapidement.

Déni de Service

Le Déni de Service distribué ou DDoS est une cyberattaque qui vise à rendre le système indisponible. Il peut être réalisé notamment par :

— Lancer des vagues de requêtes sur un serveur pour consommer toutes ses ressources
— Envoyer des données d'entrée malformées pour planter un processus
— Infiltration de virus
— Détruire ou désactiver les capteurs d'un système

La plupart des appareils sont interconnectés dans une usine, et par extension sont interdépendants. Par conséquent, l'indisponibilité de certains appareils peut être très critique pour un environnement de production, rendant ainsi ces attaques très populaires. De plus, avec le développement du cloud computing, de nouvelles façons de lancer des attaques DDoS apparaissent, poussant ainsi les entreprises à y accorder une attention accrue [23].

Contrairement au cyberespionnage où la perte porte sur des données virtuelles, les attaques DDoS ont des impacts physiques avec des dommages matériels tels que les serveurs qui pourraient devoir être remplacés (surcharge), reconfigurés ou repensés.

Un autre problème avec ces attaques est qu'elles sont imprévisibles et très difficiles à contrôler.

Chaîne d'approvisionnement et systèmes étendus

Pour rendre la chaîne d'approvisionnement plus efficace, le paradigme de l'industrie 4.0 propose la connexion entre plusieurs environnements organisationnels. Cependant, cette dernière présente des vulnérabilités système inhérentes qui peuvent être exploitées par des attaquants [24].

Une vulnérabilité peut provenir d'un fournisseur victime d'une attaque de phishing ou d'un vol d'informations d'identification, entraînant une exposition massive des données pour l'usine.

Sécurité intelligente et usine intelligente

La plupart des entreprises manufacturières ne sont pas pleinement conscientes des risques de sécurité liés au paradigme de l'industrie 4.0. Habituellement, ils traitent principalement les problèmes de sécurité lorsqu'un incident grave se produit.

Cependant, les produits techniques seuls ne suffisent pas à gérer ces risques [25]. Le facteur humain est un point important. La sensibilisation des employés à la sécurité est également importante, des opérateurs de machines qualifiés aux ingénieurs en logiciels sécurisés et en planification. Cette prise de conscience peut être obtenue de plusieurs manières, telles que :

— Campagnes de sensibilisation impliquant l'ensemble de l'environnement de fabrication
— Groupes de recherche dans les établissements d'enseignement supérieur qui étudient les sujets de cybersécurité et fournissent des lignes directrices aux professionnels de l'industrie

Menaces persistantes avancées (APT)

Les Menaces persistantes avancées (APT) appartiennent à une classe spécifique de cyberattaques. Elles sont perpétrées par certains groupes ayant une expérience et des ressources importantes. Le concept est de profiter des vulnérabilités pour infiltrer le réseau de la victime et passer inaperçu pendant une longue période [14].

La première APT identifiée dans l'industrie était *Stuxnet* en 2010. Stuxnet a été conçue comme une plate-forme pour cibler les PLC et SCADA afin d'automatiser les processus électromécaniques et de provoquer la destruction de matériel. Elle exploitait les vulnérabilités « zero-day » du système d'exploitation Microsoft Windows et des logiciels Siemens. D'autres exemples sont Duqu, DragonFly, BlackEnergy et ExPetr. Le processus d'attaque suivi par ces APT est généralement divisé en cinq étapes :

— Reconnaissance du réseau pour trouver les vulnérabilités
— Communication pour démarrer la première intrusion en envoyant des « exploits » à la victime. Cela peut se faire directement avec l'ingénierie sociale (hameçonnage, etc.), ou indirectement en compromettant un tiers tel que le fournisseur
— Suivi des vulnérabilités « zero-day » pour exécuter des actions à distance en utilisant les « backdoors » des étapes précédentes
— Propagation vers d'autres zones du réseau pour infiltrer de nouveaux appareils afin de collecter des informations ou de modifier le comportement du matériel existant
— Filtrage des informations obtenues pour les transférer vers le domaine de l'attaquant

La première étape est possible en raison des fuites de métadonnées provenant des serveurs, des automates (PLC) et des capteurs. Ces problèmes sont hérités des paradigmes cloud et IoT [26], et ils doivent être résolus [27].

2.3.4 Vulnérabilités, menaces et risques par périmètre

Dans cette section, nous analysons les failles de sécurité dans les différents périmètres prédéfinis de l'usine. Le Tableau 2.4 servira de support dans les sections suivantes.

Manufacturing-Production

Les équipements critiques dans la zone de fabrication peuvent être réduits aux ICS. D'une architecture propriétaire et isolée, les ICS sont devenus une plate-forme standard et ouverte hautement interconnectée avec les réseaux d'entreprise et publics[15]. De nouvelles fonctionnalités tels que l'accès à distance aux réseaux et aux terminaux sont apparues, rendant ainsi possible un large éventail de cyberattaques. De plus, ces systèmes sont maintenant disponibles sur Internet.

De 1997 à 2015, le nombre de vulnérabilités est passé de 2 à 189 selon le rapport Kaspersky en 2015[15]. La plupart des vulnérabilités en production sont dites « zero-day » car les développeurs viennent de découvrir l'existence de la faille alors qu'un patch visant à corriger n'a pas encore été publié.

Les raisons habituelles derrière cela sont mises en évidence dans [8] :

— Dans les usines, les appareils fonctionnent pendant des semaines ou des mois sans aucune mise à jour de sécurité ni déploiement d'antivirus
— Multiples voies d'intrusion (ordinateurs portables transportés, clés USB, etc.)
— Pas d'isolement entre les différents réseaux

Dans ce domaine, les menaces les plus courantes visent à compromettre la disponibilité et l'intégrité, notamment par la destruction physique, les attaques DDoS, les logiciels malveillants et les vers, les attaques « zero-day ».

La destruction physique peut entrer dans la catégorie des sabotages typiques d'équipements industriels. Dans le domaine de la fabrication, les attaques DDoS se concentrent principalement sur le routage avec des attaques de relais, un transfert sélectif, un trou gris, un trou noir ou des botnets qui affecteront la disponibilité.

Les logiciels malveillants et les vers ralentiront les performances opérationnelles afin d'obtenir des informations sensibles ou modifier le comportement de l'équipement afin d'en compromettre l'intégrité. Les modifications de comportement peuvent prendre plusieurs formes, comme l'altération des protocoles de communication en exploitant leurs faiblesses en matière d'authentification et d'intégrité des données.

L'analyse passive du trafic peut également être utilisée pour voler des informations confidentielles. La méthode utilisée peut être l'injection de code dans des applications opérationnelles dans le but de corrompre les mesures de contrôle, effectuer un piratage de l'utilisateur final, puis accéder aux données.

L'élévation des privilèges peut être obtenue en tirant parti des failles de sécurité présentes dans les logiciels. En 2015, la recherche IBM X-Force a rapporté que 45% de toutes les attaques se concentraient sur l'accès non autorisé.

La mobilité aisée des opérateurs en usine et leurs nombreuses interactions avec les terminaux mobiles (ordinateurs portables, smartphones, tablettes) augmentent les risques liés à ces menaces.

Les configurations et les contrôles d'accès des applications et des appareils doivent être vérifiés rigoureusement dans ce domaine.

Logistique

Dans le domaine de la logistique, de nombreux appareils sans fil présentent certaines vulnérabilités, telles que [28] :

- Réseaux Wifi, surtout lorsqu'ils ne sont pas cryptés
- Les applications métiers qui utilisent encore le protocole HTTP
- Installation d'applications malveillantes

Les vulnérabilités du réseau (wifi non crypté, HTTP, etc.) exposent presque toutes les informations envoyées par les appareils aux pirates. Pour les terminaux portables, toutes les données scannées avec le lecteur de codes à barres telles que les références, les numéros de série, les destinations des produits ainsi que les informations sur l'infrastructure informatique telles que les serveurs et les bases de données sont concernées.

L'installation d'applications malveillantes pourrait être la source d'attaques de logiciels malveillants. À titre d'exemple, « HummingBad » et « HummingWale » affectent les appareils Android et déploient des applications de collecte de données personnelles qui sont vendues en cours de route.

Supervision active

La zone de supervision active appartient au réseau de l'entreprise et comprend principalement du matériel de bureau. Certaines failles possibles dans ce domaine sont [15] [28] :

- L'absence de logiciel antivirus ou de signatures non mises à jour pourrait infecter l'ensemble du Systèmes de Contrôle Industriels (ICS) via Internet et le rendre indisponible
- Les personnes non sensibilisées à la sécurité qui cliquent sur des liens malveillants
- Ordinateur non verrouillé en quittant le bureau
- Accès à des fichiers/sites Web/données non autorisés
- Connexion d'appareils externes
- Installation de programmes sans licence / piratés

Ces vulnérabilités pourraient être à l'origine de plusieurs menaces telles que :

- Attaques d'ingénierie sociale (hameçonnage)
- Attaques sur le réseau
- Virus, logiciels malveillants, vers et rançongiciels
- Vol, perte ou bris de matériel et de données
- Transfert de données depuis et vers des appareils non autorisés

Recherche et Développement

Cette zone contient des équipements innovants qui peuvent devenir source de menaces potentielles par manque de recul sur les technologies utilisées. Cette zone peut également être la cible d'attaques en raison d'informations confidentielles et précieuses que les attaquants pourraient trouver.

Zone de vie

Dans la zone de vie, les principaux équipements récupérés sont les téléphones VoIP. Même si la VoIP permet aux entreprises de réaliser des économies sur les coûts de réseau, elle apporte également de nouvelles menaces et de nouveaux risques pour la sécurité [29]. Certains de ces risques sont courants, tels que les attaques DDoS qui submergent le serveur VoIP avec des messages de signalisation d'appel SIP (Session Initiation Protocol). De telles attaques sont dangereuses, car elles ne nécessitent pas une pénétration de l'ensemble du réseau.

Les virus et les logiciels malveillants peuvent également affecter les téléphones VoIP car les configurations VoIP utilisent des « softphones » (téléphones logiciels). Avec les réseaux VoIP, les logiciels malveillants mobiles sont également un problème car de nombreux utilisateurs passent des appels VoIP avec leurs smartphones. Cela signifie qu'une fois que le malware s'est infiltré dans le smartphone, il peut accéder et voler des informations précieuses.

Une autre menace est le « vishing » qui est la contrepartie vocale du phishing par e-mail. Les employés, les fournisseurs et les clients sont amenés à partager des informations sensibles.

Le « phreaking » se produit lorsqu'un pirate accède au réseau VoIP de l'entreprise et l'utilise pour voler des données d'entreprise, modifier le plan du réseau ou passer des appels coûteux, entraînant ainsi des factures élevées pour les fournisseurs de services.

L'écoute clandestine est également une menace de cybersécurité courante qui est très difficile à surmonter. Les pirates réussissent à accéder aux appels VoIP et à les écouter en capturant le trafic VoIP non crypté. De cette façon, ils sont capables d'effectuer des vols d'identité, ainsi que des vols de services VoIP.

La dernière menace est « Spam Over Internet Technology » (SPIT). Pour les pirates, cela consiste à capturer des milliers d'adresses IP VoIP, puis à envoyer des messages vocaux à un système VoIP.

Zones externes

Les espaces extérieurs sont divisés en deux catégories qui sont physiques et virtuelles.

Zone externe physique La zone externe physique est constituée de caméras de surveillance IP qui relèvent du domaine de l'Internet des objets. Les menaces pour ces types d'appareils sont :

- Influence du mode de fonctionnement du protocole de routage avec brouillage et interférence afin de perturber les communications
- Épuisement des ressources en utilisant des vulnérabilités dans les logiciels qui contrôlent les appareils ou avec un code malveillant (malware)
- Manipulation des informations de routage pour influencer le trafic, comme dans une attaque Sybil[14]. Ces attaques sont des passerelles vers d'autres, telles que le trou noir ou le déni de service.
- Attaques par canal latéral pour exposer les informations sur les appareils (batterie, mémoire) ou les informations de routage et la topologie pour identifier les équipements vulnérables dans l'infrastructure
- Injection de noeuds factices capables d'exécuter du code ou d'injecter du trafic illégitime afin de contrôler de vastes zones du réseau ou d'effectuer des écoutes clandestines.
- Mauvais contrôle d'accès entraînant un accès non autorisé aux ressources protégées

Zone externe virtuelle La deuxième catégorie est l'espace externe virtuel qui regroupe tout ce qui touche au cloud computing. Ce dernier est intéressant de par son faible coût d'investissement et la facilité de déploiement qu'il offre aux entreprises. De nombreuses organisations utilisent le cloud computing pour stocker leurs données mais aussi pour héberger certains processus. Il peut même être utilisé dans l'IoT pour acquérir des données de capteurs mais aussi comme un moyen pour les clients de fabriquer un produit via un réseau partagé de fournisseurs tout au long de son cycle de vie. Derrière ces innovations apparaissent ces menaces :

— Attaques DDoS en utilisant des vulnérabilités à l'intérieur du composant planificateur de certains hyperviseurs pour facturer le service et le rendre indisponible [14]
— Injection de logiciels malveillants pour remplacer un service d'instance cloud légitime comme une machine virtuelle par un service malveillant pour accéder aux données échangées.
— Attaques par canaux secondaires qui obligent les machines à étudier les émanations électromagnétiques et à accéder à leurs ressources
— Attaques de mémoire partagée qui analysent le cache ou la mémoire principale pour obtenir des informations techniques sur l'infrastructure, les processus en cours d'exécution ou pour accéder au vidage mémoire des machines virtuelles
— Attaques d'ingénierie sociale qui capturent les informations des clients des différentes applications. L'objectif est d'obtenir des données sensibles telles que des comptes, des mots de passe afin d'héberger des services malveillants dans le cloud

La plupart de ces attaques font partie des menaces persistantes avancées (APT), qui peuvent être exécutées principalement par des attaquants expérimentés et dotés de ressources. Il sied de remarquer que nous n'avons pas considéré ici les risques liés à la souveraineté des données et les risques géopolitiques, lorsque les services cloud ne relèvent pas de la législation du pays où se trouve l'usine.

Tableau 2.4 – Risques et menaces par périmètre dans l'industrie 4.0

	Menaces	*Présence des menaces dans les périmètres*				
		Manufacturing Production	*Logistique*	*Supervision active*	*Zone de vie*	*Zone externe*
Disponibilité	Soustraction de périphériques	X				
	Attaques DDoS	X	X	X	X	X
	Attaques sur le chemin	X	X	X		
	Épuisement des ressources des noeuds	X	X	X		
	Vol de services	X	X			X
Intégrité	Configuration incorrecte	X	X			X
	Injection de malware	X	X	X	X	X
	Injection de fausses données	X	X			
	Usurpation d'identité	X		X	X	
	Manipulation des informations de routage	X	X			

	Menaces	*Présence des menaces dans les périmètres*				
		Manufacturing Production	*Logistique*	*Supervision active*	*Zone de vie*	*Zone externe*
Confidentialité	Vol de données sensibles	X		X		X
	Exposition de l'état des noeuds	X	X	X	X	X
	Analyse passive du traffic	X	X	X	X	X
	Exposition d'informations liées à l'infrastructure	X				X
Authentification	Élévation de privilèges	X		X		X
	Ingénierie sociale	X		X	X	X
	Contrôle d'accès déficient	X	X	X		X
	Emprunt d'identité de noeuds	X	X			

2.4 Revue des solutions

Cette section proposera un examen détaillé de certaines solutions existantes contre les menaces de cybersécurité. Ces solutions ont été sélectionnées afin de présenter les principales tendances concernant les solutions classiques et innovantes pouvant s'appliquer dans les différents périmètres définis précédemment.

Les deux notions suivantes sont nécessaires pour une meilleure compréhension des solutions présentées : les contre-mesures et les solutions à long terme issues de la littérature.

2.4.1 Définitions

Contre-mesures Les contre-mesures peuvent être considérées comme le moyen à court terme de se défendre contre une menace ou une attaque. Elles représentent un ensemble d'actions et de techniques pour éliminer une menace ou la prévenir afin de limiter les dommages qu'elles pourraient causer. Elles peuvent également aider à signaler la menace afin que des mesures correctives puissent être prises [8].

Solutions Les solutions sont le moyen à long terme de faire face aux menaces. Elles sont souvent définies avec des noms différents tels que approche, méthodologie ou architecture. Elles représentent un ensemble complet d'actions pour se protéger contre différents types de cyberattaques. Ces solutions peuvent être classées en trois catégories [15] : les outils d'évaluation de la sécurité, les technologies de détection et de prévention des intrusions ainsi que la gestion des risques ICS.

Les outils d'évaluation de la sécurité sont capables de fournir une expérimentation sécurisée avec des scénarios de test réalistes (attaques, infections, etc.) afin de détecter les problèmes de sécurité avant la mise en production.

Les technologies de détection et de prévention des intrusions mettent en évidence les approches de sécurisation des ICS en introduisant de nouveaux composants ou en mettant à niveau les architectures existantes. La gestion des risques ICS propose des lignes directrices, des normes et des mesures pour assurer la protection de la sécurité mise en œuvre contre l'évolution des menaces.

2.4.2 Contre-mesures en cybersécurité

Selon [8], l'approche suivante à trois niveaux peut être utilisée pour garantir la sécurité des systèmes de contrôle industriels :

— Renforcement du périmètre en isolant le réseau de l'usine du réseau bureautique à l'aide de pare-feu et de Zone démilitarisée (DMZ)
— Défense multicouche sur le réseau pour contenir les attaques
— Isoler les utilisateurs distants dans une zone/un réseau séparé

Tableau 2.5 – Contremesures pour la cybersécurité en industrie 4.0 [8]

Contremesures	*Protège de / Prévient*
Chiffrement des communications	Altération de données Divulgation d'informations
Chiffrement des données stockées (signature)	Attaques de répudiation
Chiffrement des flux de données	Falsification des données Attaques de répudiation Menaces DDoS
Pare-feu / passerelle et proxy	Accès réseaux non voulus (depuis/vers)
Contrôle d'accès / Autorisation multiple	Accès non autorisé (informations, zones physiques)
Mises à jour logicielles	Failles (Système, logiciel, métiers, firmware)
Sécurisation des communications (VPN, WI-FI et IP)	Écoute clandestine Connexion non autorisée Analyse de données
Antivirus et anti-malware	Logiciel malveillant

Ces mesures permettent une protection contre les accès indésirables depuis et vers Internet, mais sont également utilisées pour séparer les services et les zones entre les systèmes de l'usine. Le chiffrement est la plus populaire de ces mesures et est disponible à plusieurs niveaux :

— Chiffrement des communications pour éviter la falsification et la divulgation d'informations
— Chiffrement des données stockées pour éviter les attaques de répudiation
— Chiffrement des flux de données pour éviter tous les problèmes précédents

Pour être efficaces, toutes les contre-mesures doivent être continuellement mises à jour, y compris les protocoles de cryptage, les signatures de pare-feu, les contrôles de sécurité (correctifs), la surveillance (analyse des journaux). Tableau 2.5 décrit les contre-mesures les plus populaires résumées à partir de [8] et comment elles peuvent aider à prévenir les cybermenaces.

2.4.3 Propositions de solutions de cybersécurité dans la littérature

Cette section mettra en évidence et présentera en détail certaines des solutions de cybersécurité actuellement disponibles dans la littérature Tableau 2.6. Pour chaque solution, nous rapportons son objectif et sa structure de conception, ainsi que la valeur que l'on peut donner à cette solution.

I. Protection des environnements physiques

Contexte et objectif Les entreprises modernes ont besoin de solutions à la fois logicielles et physiques pour la sécurité de leurs informations. Malgré l'efficacité des contrôles logiciels, ces derniers sont virtuels et il existe une interface qui peut sortir du contrôle du système et le compromettre. Cette interface peut être

humaine, et la brèche apparaît lorsque des informations sont échangées verbalement. De plus, la croissance explosive des réseaux sans fil dans les architectures des systèmes d'information a encore accru la possibilité de fuites d'informations [30]. Pour ces raisons, [20] a proposé de créer une sécurité physique de l'information. Cette solution prend la forme de « Zones Protégées » (voir Figure 2.3), qui sont des espaces où des données sensibles et précieuses peuvent être échangées en toute sécurité, aussi bien acoustiquement que visuellement.

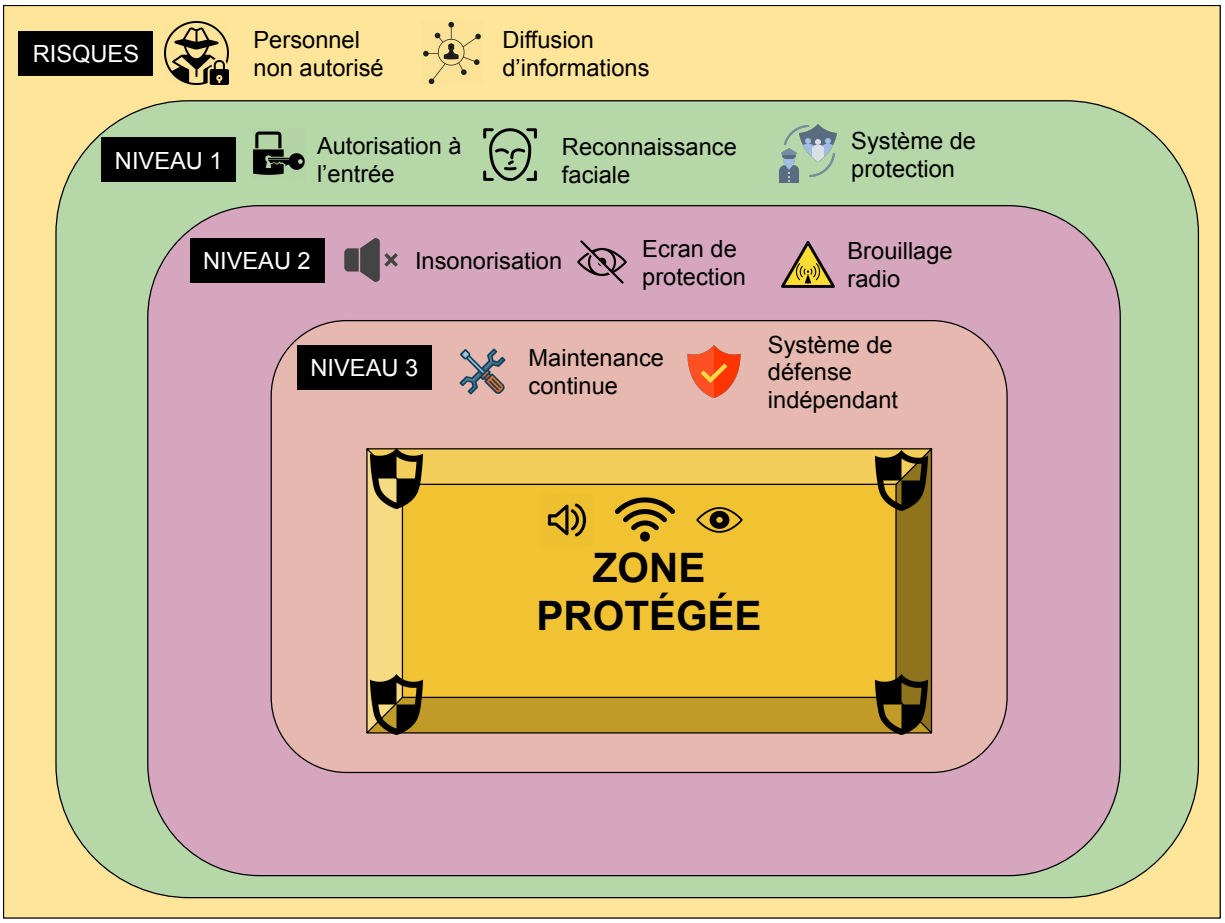


Figure 2.3 – Protection des zones physiques

Conception et approche Cette solution poursuit les objectifs suivants :

- Empêcher l'accès aux personnes non autorisées ;
- Limiter la propagation d'informations aux murs des aires protégées ;
- Environnement homogènement protégé et contrôlé

Pour atteindre ces objectifs, trois couches de solutions sont proposées.

La première couche est constituée des moyens classiques utilisés pour la défense des installations protégées : authentification à l'entrée, filtrage des personnes, définition des autorisations et systèmes de sécurisation.

Avec la première couche, nous pouvons être sûrs que seules les personnes autorisées se trouvent dans la zone, mais il y a toujours un risque car les informations peuvent toujours être « vues » de l'extérieur.

La deuxième couche permet d'empêcher la diffusion d'informations hors de la zone protégée par des caractéristiques physiques (visuelles, sonores, radio) en utilisant un blindage acoustique, un blindage visuel et un blindage radio.

La dernière couche est dédiée à la maintenance dans le cadre de cette proposition de sécurité, qui passe par une maintenance continue et un système de défense indépendant.

L'idée derrière la maintenance est de maintenir l'homogénéité de l'environnement, de rechercher des corps étrangers et de surveiller les signaux radio. Un système autonome de sauvegarde et de surveillance a également été conseillé pour compléter la sécurisation de ces aires protégées.

Résultats et mise en valeur L'auteur a suggéré que même si les nouvelles technologies introduisent de nouvelles menaces, la préparation et la création de ces zones contrôlées entre autres actions pourraient permettre le maintien de l'intégrité, de la confidentialité et de la disponibilité à long terme.

II. Pare-Feu pour l'industrie 4.0

Contexte et objectif Le « Manufacturing Execution System (MES) » est le système intermédiaire entre l'ICS et les applications d'entreprise telles que le progiciel de gestion intégré (ERP). Il améliore la transparence des données de fabrication. Les données des capteurs peuvent être utilisées pour calculer des indicateurs de performance en temps réel ou pour surveiller l'état de la machine et la qualité des processus de fabrication. L'amélioration de l'interconnectivité des systèmes informatiques expose les appareils comme les PLC à des cyberattaques, qui pourraient perturber la production ou infecter d'autres systèmes. Les attaques les plus courantes dans le cas du MES sont basées sur le balayage/sondage du réseau où la défense en profondeur est une contre-mesure efficace. Les normes de cybersécurité proposaient généralement des architectures de réseau divisées en plusieurs segments avec des pare-feu entre eux pour minimiser les risques de sécurité. Compte tenu de la nécessité de définir des règles de configuration de manière flexible et sécurisée, le réseau défini par logiciel (SDN) est probablement une technologie clé à cet égard [31] [32]. Le système d'exécution de la fabrication (MES) est le système intermédiaire entre le SCI et les applications d'entreprise telles que le progiciel de gestion intégré (ERP). Il améliore la transparence des données de fabrication. Les données des capteurs peuvent être utilisées pour calculer des indicateurs de performance en temps réel ou pour surveiller l'état de la machine et la qualité des processus de fabrication. L'amélioration de l'interconnectivité des systèmes informatiques expose les appareils comme les CPL à des cyberattaques, qui pourraient perturber la production ou infecter d'autres systèmes. Les attaques les plus courantes dans le cas du MES sont basées sur le balayage/sondage du réseau où la défense en profondeur est une contre-mesure efficace. Les normes de cybersécurité proposaient généralement des architectures de réseau divisées en plusieurs segments avec des pare-feu entre eux pour minimiser les risques de sécurité. Compte tenu de la nécessité de définir des règles de configuration de manière flexible et sécurisée, le réseau défini par logiciel (SDN) est probablement une technologie clé à cet égard [31] [32]. Une autre proposition connexe récente axée sur les performances du réseau est présentée dans [33]. Le « Software Defined Network (SDN) » est une technologie capable d'alterner le réseau, de libérer des informations critiques et de fournir de nouveaux services pour exécuter des applications à la demande. Il donne un aperçu clair de l'architecture du réseau aux administrateurs et permet aux utilisateurs de contrôler l'architecture du réseau par programmation. Cela signifie qu'il est possible de modifier l'accès au réseau à la demande et de minimiser l'exposition des réseaux ICS aux attaquants. Dans ce contexte, [34] a proposé une structure de réseau de protection basée sur un pare-feu SDN spécifiquement conçu pour les réseaux industriels, sans compromettre la flexibilité du réseau (Figure 2.4).

Conception et approche La solution proposée vise trois objectifs :

1. Création de segments sans reconfigurer les réseaux existants, tout en utilisant une intégration verticale DMZ (Zone Démilitarisée) ;
2. Développement de mécanismes d'accès unidirectionnels (accès au serveur uniquement lorsqu'un client a besoin d'une connexion) ;
3. Réduction des failles dans les règles d'accès en raison de l'insertion ou du remplacement fréquent d'appareils dans certaines zones

Le pare-feu SDN a deux fonctions principales qui sont le filtrage de paquets basé sur l'application des règles d'accès à un groupe d'appareils automatiquement ainsi que le fait de servir de passerelle entre les interfaces réseau pour éviter toute modification de la configuration réseau existante.

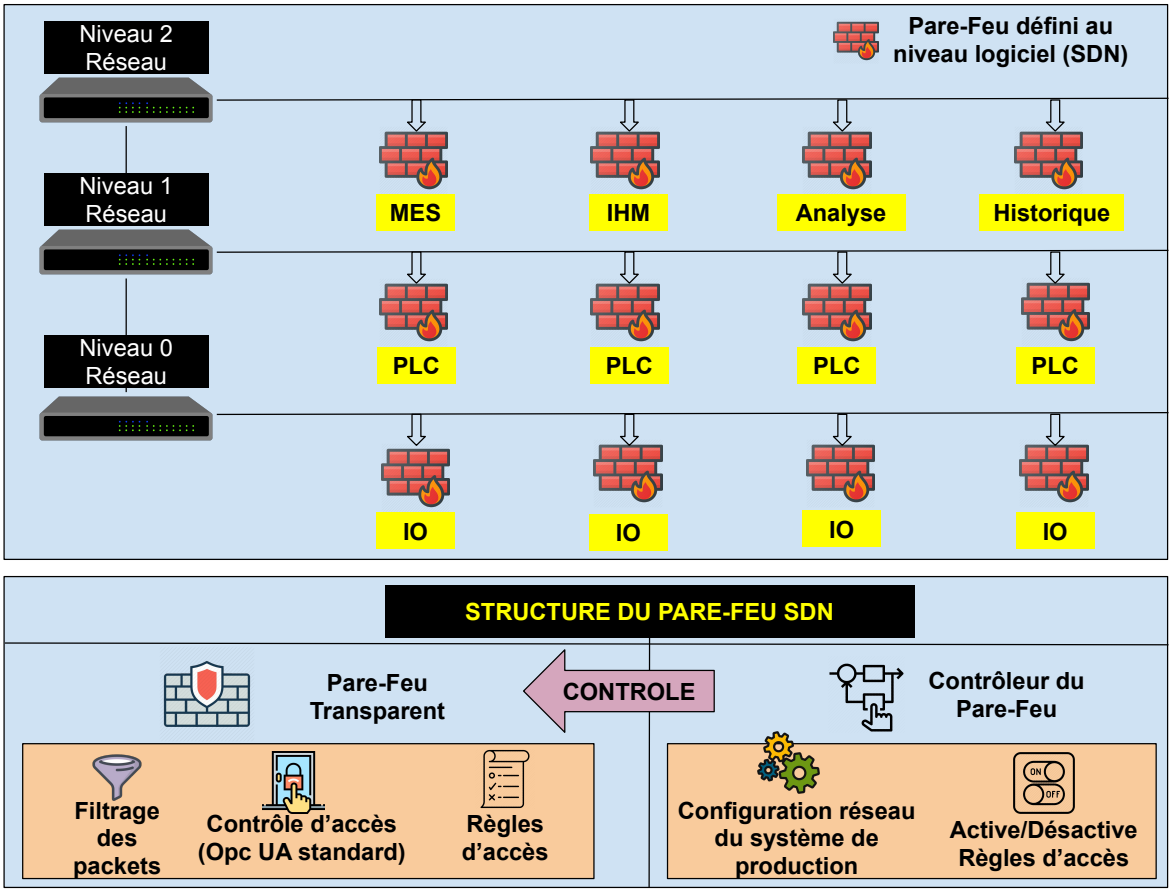


Figure 2.4 – Pare-Feu défini au niveau logiciel (SDN)

Cette fonction de filtrage de paquets signifie que seules les applications figurant dans une liste blanche peuvent accéder aux dispositifs de contrôle industriels. La liste blanche est gérée par les administrateurs réseau, dont la tâche de gestion est grandement simplifiée, car il s'agit de la seule chose à faire pour se conformer aux normes de sécurité des ICS. Concernant les composants, le pare-feu contient :

— Le pare-feu transparent qui applique les règles d'accès et implémente un système de contrôle d'accès basé sur la norme OPC UA
— Le contrôleur de pare-feu qui conserve la configuration du système de production et gère les règles

Le standard « Open Platform Communication Unified Architecture (OPC UA) » utilisé dans le contrôle d'accès du pare-feu transparent est un protocole de communication machine à machine pour l'automatisation industrielle.

Il s'agit d'une évolution du protocole OPC original qui est mieux adapté pour répondre aux besoins émergents de l'automatisation industrielle dans l'Industrie 4.0. Ses innovations les plus notables sont :

— Implémentation multiplateforme ;
— Évolutivité (capteurs intelligents, actionneurs intelligents, etc.) ;
— Multi-thread ou mono-thread ;
— Amélioration de la sécurité (nouvelles normes) ;
— Délais d'attente configurables ;
— Regroupement de datagrammes volumineux

La sécurité dans OPC UA consiste en l'authentification, l'autorisation, le cryptage et l'intégrité des données via des signatures. De plus, la pile de communication utilise des transmissions compatibles avec le pare-feu, ce qui explique pourquoi cette norme a été utilisée pour concevoir le pare-feu SDN.

Résultats et mise en valeur L'auteur a confirmé que la solution a été testée dans le réseau virtuel d'un environnement complet (client/serveur OPC avec échange de données machine) et que le pare-feu a pu empêcher les scanners de sécurité d'acquérir le port d'application et d'autres détails au niveau du système d'exploitation du serveur OPC.

La mise en œuvre du prototype a permis de compléter les fonctionnalités de sécurité de la norme OPC UA et a fourni une solution de sécurité globale pour les réseaux ICS.

III. Sécurisation des équipements IoT

Contexte et objectif L'Internet des objets (IoT) est un terme utilisé pour la première fois par Kévin Ashton en 1999 [35]. Il n'existe pas de définition universelle du terme, cependant il induit l'idée que tout objet du quotidien peut être équipé avec la capacité de mesurer toute chose liée à son environnement, communiquer avec d'autres objets et envoyer les données détectées via internet [36]. Au sein d'une usine, il associe plusieurs technologies liées au développement de capteurs et au contrôle des machines. Dans le contexte de l'industrie 4.0, ils deviennent populaires en raison de l'interconnexion entre les données des ateliers industriels et de la possibilité de fournir un retour d'information sur l'exécution des systèmes. Leur utilisation conduit au nouveau concept de systèmes cyber-physiques (CPS), qui sont associés à la mise en œuvre de l'IoT, et font référence à l'utilisation de capteurs pour collecter des données, les traiter et les utiliser dans le monde cybernétique. En termes de connectivité des équipements IoT, il existe des réseaux émergents appelés Low Power Wide Area Network (LPWAN), tels que : Sigfox, LoRa et Weightless [37]. Ces réseaux sont conçus spécifiquement pour l'IoT car ce sont des protocoles à faible consommation d'énergie. En outre, ils permettent une communication bidirectionnelle entre les objets et le monde extérieur, en utilisant le réseau LPWAN et le cloud du fournisseur de réseau pour la connectivité Internet.

Le principal inconvénient qui entrave l'adoption complète est leurs faiblesses en matière de cybersécurité [38], qui sont dues notamment à la connectivité hétérogène et aux menaces qui en découlent (violation de la vie privée, etc.), et peuvent entraîner des conséquences majeures pour les utilisateurs de la technologie IoT[39]. Selon [40], ces systèmes doivent être conçus et exploités selon une vision unifiée des caractéristiques de sûreté et de sécurité [41]. Dans le cadre de l'usine intelligente, l'une de ces menaces pourrait résulter des faiblesses liées à l'utilisation du cloud computing [42]. Compte tenu de cette situation, un framework de cybersécurité basé sur une ontologie pour l'IoT a été proposé par [43].

Conception et approche Le framework proposé se concentre sur :

— Surveillance côté entreprise ;
— Analyse et classification de la sécurité dans une base de connaissances ;
— Conception de services de sécurité ;
— Amélioration des mécanismes de sécurité concernant les processus métiers et les actifs technologiques

Figure 2.5 est une représentation de l'architecture du framework. Il est divisé en trois couches : deux couches traitant de la cybersécurité à la conception et à l'exécution, et la couche d'intégration utilisée aux deux étapes. Le concept de la couche de conception est le suivant : on suppose qu'une entreprise a besoin de mettre en place à tout moment des services spécifiques en plus de ceux déjà utilisés, ce qui nécessite une adaptation pour répondre aux contraintes de l'appareil.

Le « Model-Driven Service Engineering Architecture (MDSEA) » est une méthodologie de développement logiciel qui se concentre sur la création et l'exploitation de modèles conceptuels (dans le cas du framework : modèles commerciaux, modèles indépendants en termes de technologie et modèles spécifiques) liés à un problème spécifique. En utilisant cette méthodologie, la couche de conception est capable de générer du code à partir d'un haut niveau d'abstraction pour accélérer la conception et l'adaptation du service, ainsi que le temps de déploiement. Ensuite, les managers peuvent collaborer avec les développeurs pour participer à la création des nouvelles fonctionnalités.

La couche d'exécution a deux objectifs : surveiller et mettre à jour la base de connaissances. La surveillance consiste à détecter les intrusions, les vols de données, les virus et autres tentatives d'exploitation de failles de

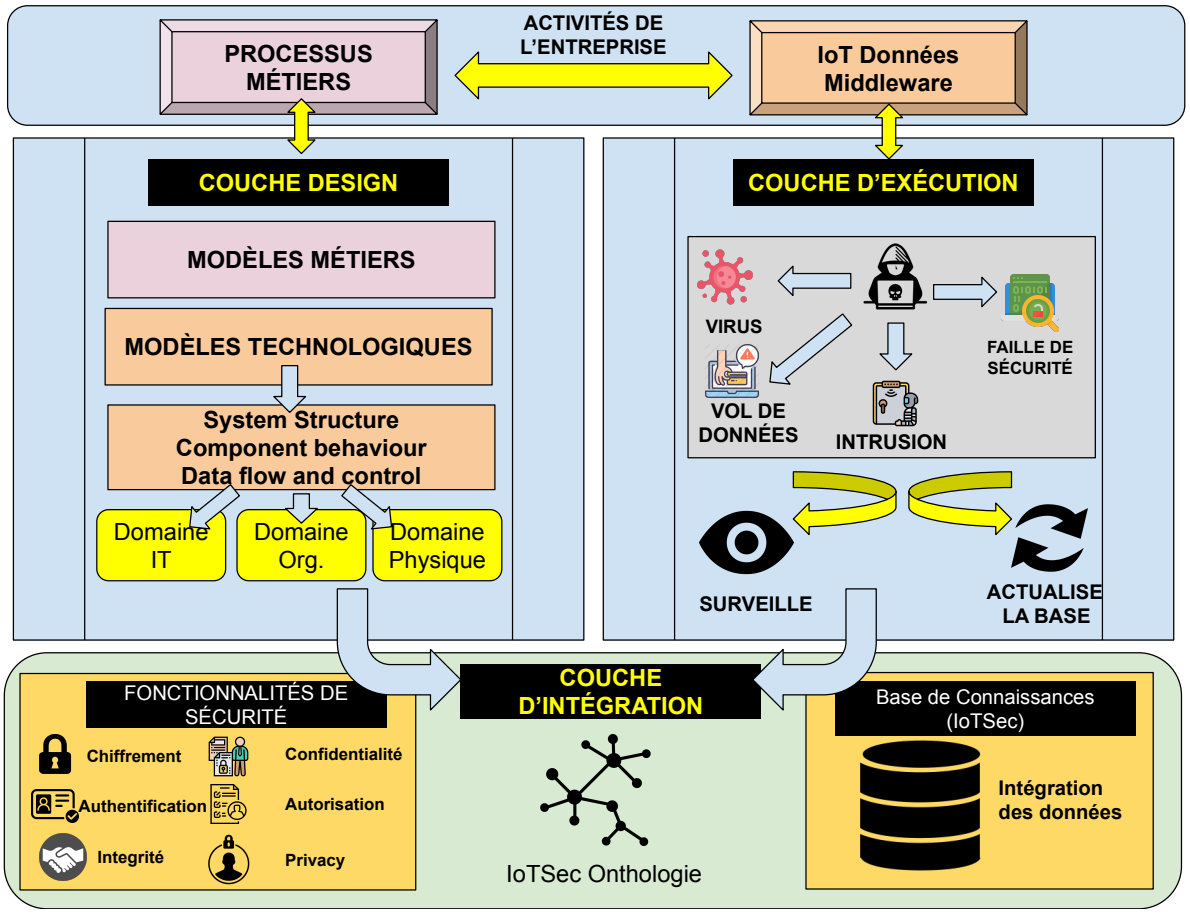


Figure 2.5 – Framework d'ontologie pour la cybersécurité de l'IoT

sécurité. Toutes ces situations sont analysées pour identifier les solutions adaptées parmi le pool de services de sécurité d'IoTSec afin de restaurer le système et d'améliorer la cybersécurité. La mise à jour consiste à ajouter les menaces détectées et l'analyse de sécurité à la base de connaissances, empêchant ainsi que ces menaces ne réapparaissent. La couche d'intégration de données fournit des informations sur les menaces et les vulnérabilités à l'aide de l'ontologie IoTSec, qui est un travail continu de [44].

Résultats et mise en valeur Enfin, les auteurs considèrent que l'ontologie en cybersécurité améliore l'efficacité des opérations de sécurité et aide les analystes à extraire des informations pertinentes pour caractériser les vulnérabilités. Cependant, certaines questions ouvertes subsistent avant d'obtenir une ontologie de renseignement de cybersécurité multicouche capable de comprendre les menaces potentielles dans le contexte de la cybersécurité, qui est un domaine en constante évolution.

IV. Authentification des systèmes cyber-physiques

Contexte et objectif Les chaînes de valeur longues sont l'un des principaux problèmes de sécurité de l'industrie 4.0. Les paradigmes de la Information Technology (technologie de l'information) (IT) ne reflètent pas les circonstances particulières rencontrées dans la technologie de l'exploitation (OT).

De nombreux types de données sont accumulées dans la partie production, et sont utilisées pour les contrôles qualité et la maintenance prédictive. Cependant, seuls certains d'entre eux sont critiques pour la protection. Une solution OT doit se concentrer sur : la nomenclature, les informations de conception et les paramètres de contrôle [45].

Dans l'environnement de fabrication, l'architecture actuellement utilisée est généralement entièrement séparée des environnements de production, avec des dispositifs de fabrication isolés. Il n'y a aucun contrôle sur les sous-traitants dans la production en plusieurs étapes lorsque les données quittent le serveur.

De la même manière, les opérateurs ne sont pas supervisés une fois les données reçues et de nombreux problèmes surviennent, tels que la corruption des données. Comme il a été établi par [46], les menaces exposées dans la fabrication additive s'étendent aux appareils de fabrication numérique de toutes formes et tailles. Par conséquent, un nouveau paradigme est proposé par [45]. Il vise à limiter et à protéger les flux d'informations dans les dispositifs d'étapes sous-traitées pour compléter la sécurité périmétrique.

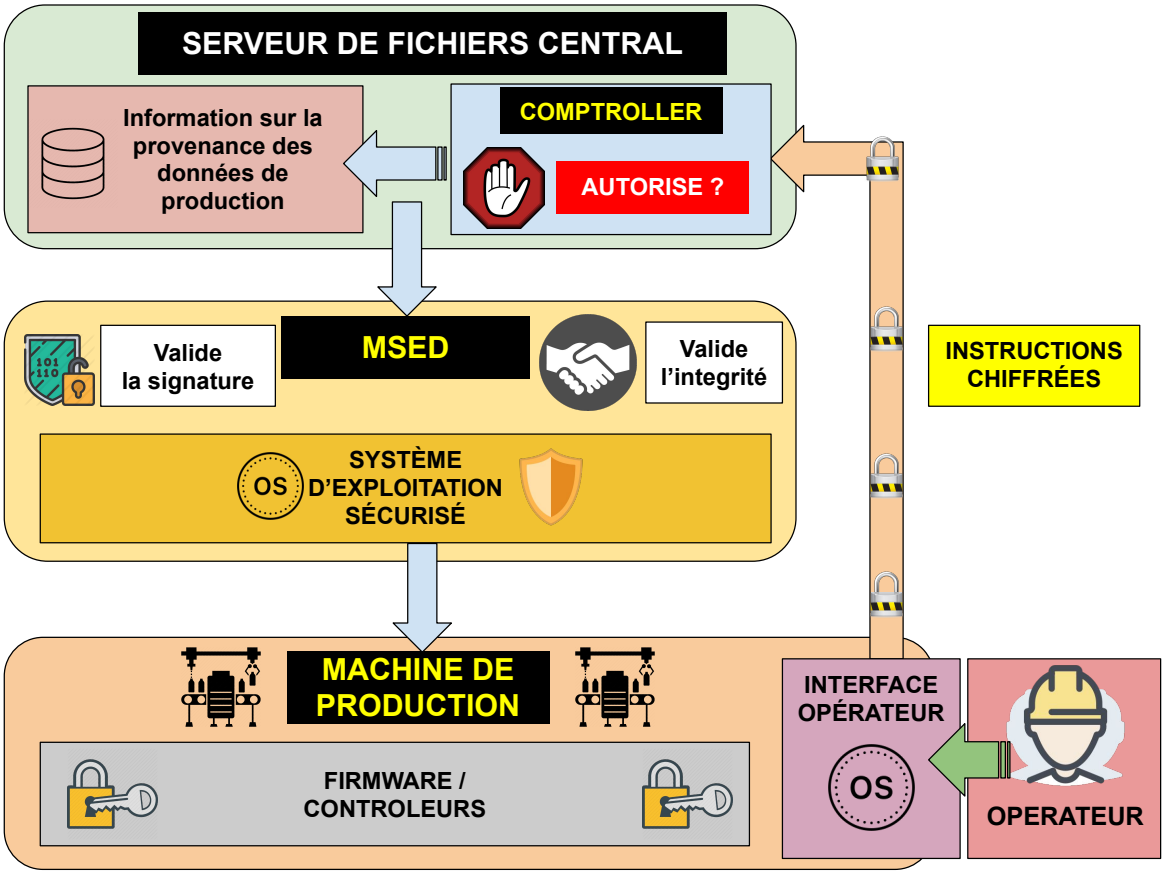


Figure 2.6 – Approche Machine Directe pour la protection des systèmes cyberphysiques dans le manufacturing

Conception et approche Cette nouvelle architecture d'informations de fabrication est basée sur l'approche holistique selon laquelle les données doivent être envoyées directement au dispositif de fabrication concerné. Cette approche est également appelée communication « Direct-to-Machine » (Figure 2.6). Le problème de sécurité fondamental derrière cela concerne l'authentification et l'autorisation :

— La demande envoyée à l'appareil est-elle authentique ?
— L'émetteur est-il autorisé à envoyer cette demande ?

Pour résoudre ce problème, l'architecture proposée est composée de deux composants. Le premier est le cryptage avec un dispositif d'application de la sécurité de fabrication (MSED). Il repose sur la cryptographie asymétrique. La clé publique est utilisée pour vérifier que l'acteur ou l'entité dispose bien de la clé privée appropriée, l'authentifiant ainsi.

Le second est un logiciel appelé « comptroller » qui s'exécute sur un réseau de production et autorise chaque action entreprise. Il prend les données d'entrée, fournit une clé et stocke les données de sortie. Les données de sortie sont ajoutées à la fin d'un document virtuel qui est l'enregistrement de provenance d'une pièce

produite. Ensuite, les données transmises entre le « comptroller » et le dispositif de production seront traitées par un MSED, dont l'aperçu complet peut être trouvé dans [47]. Le MSED se trouve en amont de l'équipement de fabrication et authentifie les instructions de fabrication provenant du cloud en vérifiant à la fois l'intégrité des données d'instruction, ainsi que l'identité et les autorisations du « comptroller ». À cette fin, la meilleure méthode consiste à utiliser le cryptage et des signatures de données uniques garantissant que la source de données est authentique. Une autre exigence pour le MSED est d'avoir un système d'exploitation sécurisé (OS). Les systèmes d'exploitation courants pour les systèmes embarqués sont Microsoft WindowsTM ou LinuxTM. Ils fournissent une large gamme d'outils, mais sont vulnérables aux attaques zero-day qui peuvent être d'une efficacité dévastatrice. Les auteurs [45] ont mentionné des systèmes d'exploitation plus petits tels que SeL4, dont la sécurité a été formellement vérifiée.

Résultats et mise en valeur Un prototype de dispositif MSED utilisant le micro-noyau SeL4 a été complété avec la société « True Secure SCADA, LLC », et a été confirmé compatible pour le contrôle industriel général ainsi que pour la fabrication.

La communication « Direct-To-Machine » dispose d'arguments solides pour surmonter les défis de sécurité cyber-physique, notamment :

— ses caractéristiques (autorisation, surveillance et contrôle de l'opérateur, prise en charge de l'appareil, responsabilité répartie, indépendant de l'emplacement) ;
— son contraste avec les solutions existantes (sécurité intégrée, cryptage en couches, toujours à jour, protection de l'intégrité, partage minimum) ;
— son rôle dans la construction d'un environnement de production réactif (suivi détaillé avec le contrôleur, collaboration, fabrication distribuée, automatisation)

L'un des défis concernant l'adoption de cette approche est la volonté de gérer l'élévation du niveau de sécurité à l'échelle cyber-physique, ou que les services informatiques reconnaissent l'équipement de production comme un autre composant numérique à intégrer dans la liste des éléments à protéger. Dans l'industrie 4.0, ces appareils ne sont plus séparés du flux de données, et l'approche « Direct-To-Machine » reconnaît ce fait.

V. Détection prédictive des cyberattaques

Contexte et objectif Les architectures de cybersécurité traditionnelles se concentrent sur des mécanismes qui assurent la confidentialité, l'authentification, l'intégrité, le contrôle d'accès et la non-répudiation afin de prévenir les intrusions et les attaques sur le réseau. Cependant, le paysage sécuritaire actuel se caractérise par des attaques en constante évolution, volumineuses, rapides, persistantes et hautement sophistiquées.

Pour les systèmes critiques appartenant à l'industrie 4.0, le besoin de détection et de réponse autonomes aux cyberattaques est nécessaire afin d'obtenir une cybersécurité robuste avec une défense en profondeur. Un algorithme de détection de cyberattaques a été proposé par [48] pour défendre les systèmes de l'industrie 4.0, ainsi que d'autres systèmes basés sur Internet. Cet algorithme est basé sur l'intelligence d'ensemble avec des réseaux de neurones pour exploiter une sortie de classification fournissant une rétroaction aux mécanismes de réponse actifs. L'objectif sous-jacent est de montrer comment les approches d'intelligence computationnelle peuvent être utilisées dans la cybersécurité de l'Industrie 4.0.

Conception et approche Habituellement, les systèmes de détection de cyberattaques nécessitent des algorithmes qui collectent et analysent les données générées par divers événements se produisant dans un cyberenvironnement. L'un de leurs principaux problèmes est le manque de précision, et des résultats inexacts peuvent avoir un impact négatif sur les performances du système et entraîner des problèmes de sécurité (faux signalements, intrusions inaperçues, etc.)

Les Systèmes d'intelligence computationnelle (CIS) sont des systèmes adaptatifs dotés de capacités de prise de décision, et ils sont spécifiquement conçus pour gérer de gros volumes de données (avec bruit) dans leur processus de décision. Par conséquent, ils semblent être un choix logique lors de la conception de nouveaux algorithmes pour les systèmes de détection. Ces systèmes utilisent des technologies telles que le machine

learning et le deep learning qui sont capables de découvrir avec précision les différences essentielles entre les données normales et les données anormales [49] [50].

L'algorithme de classification proposé est nommé « Neural Network Oracle (NNO) », et il est composé de trois composants : réseaux de neurones, algorithme génétique et le NNO.

Figure 2.7 est une représentation du framework d'intelligence d'ensemble qui utilise l'algorithme de classification NNO dans le cadre de la détection prédictive des cyberattaques pour le manufacturing avancé.

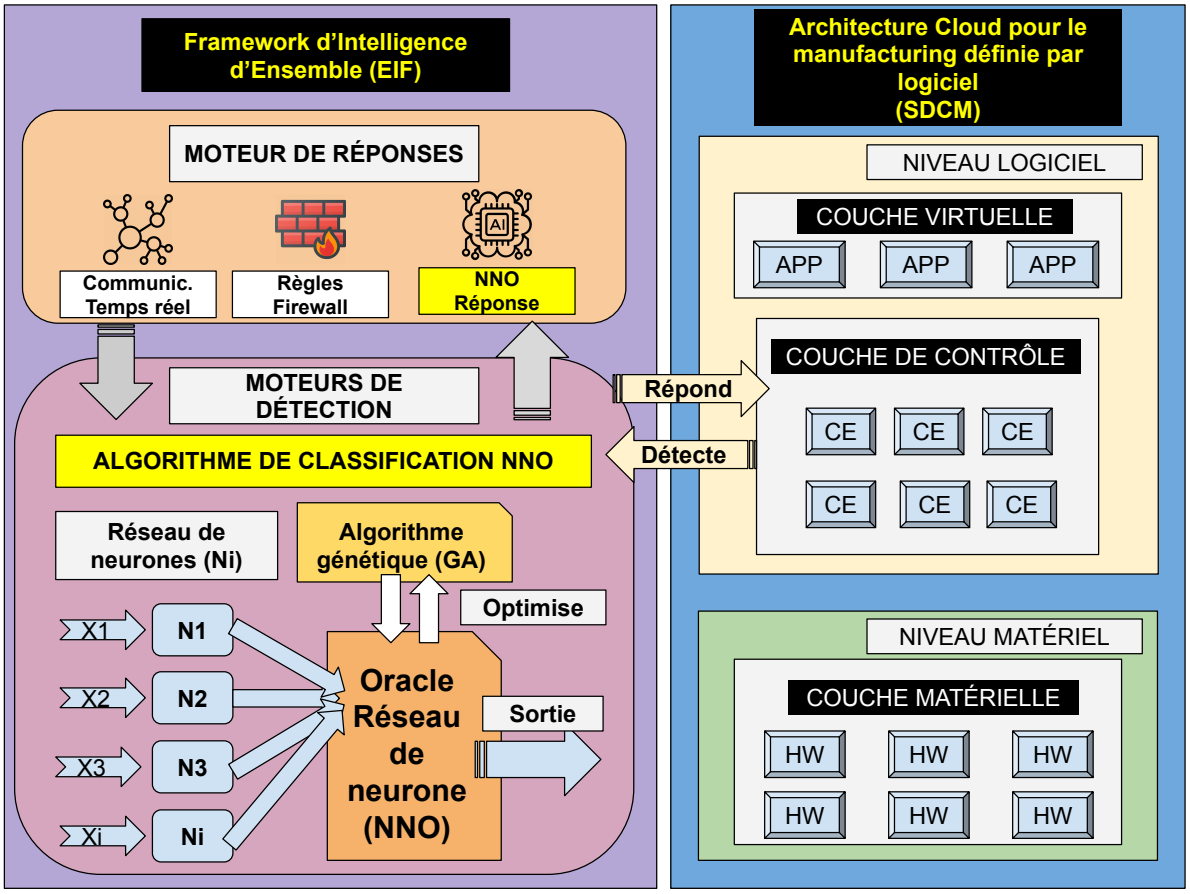


Figure 2.7 – Détection prédictive des cyberattaques avec intelligence d'ensemble dans le manufacturing avancé

Les réseaux de neurones aussi appelés Réseaux de Neurones Artificiels (ANN) s'inspirent des processus biologiques, et sont utilisés pour résoudre des problèmes d'intelligence artificielle. L'une de leurs caractéristiques les plus intéressantes est l'auto-apprentissage basé sur l'entraînement avec des ensembles de données. Dans le NNO, une collection de réseaux de neurones s'entraînera avec un premier ensemble de données d'audit. Ensuite, le résultat de cette première classification sera envoyé au NNO.

L'algorithme génétique, en anglais Genetic Algorithm (GA), s'inspire du processus d'évolution biologique et de la capacité à s'adapter au fil du temps dans des environnements changeants. Dans l'algorithme NNO, le GA est chargé de trouver les paramètres les plus optimaux afin de réduire le taux d'erreur et d'augmenter la précision. Pour obtenir ces paramètres optimaux, une fonction de fitness évaluée avec les réponses du réseau de neurones est utilisée.

L'oracle du réseau de neurones est entraîné avec un ensemble secondaire de données composé des sorties des réseaux de neurones précédents et de l'ensemble de données d'origine. Pour minimiser les erreurs, il utilise les paramètres optimaux fournis par l'algorithme génétique.

Résultats et mise en valeur L'auteur a intégré le framework d'intelligence d'ensemble basé sur la classification NNO dans une architecture cloud définie par logiciel pour le manufacturing (SDCM) [48]. Le SDCM est divisé en 3 couches : virtuelle, de contrôle et matérielle distribuée.

Comme la couche de contrôle est celle qui a la meilleure connaissance des activités et des communications, elle sera utilisée comme point de prélèvement de données. La couche de contrôle alimentera l'Ensemble Intelligence Framework (EIF) avec des données en continu, et l'EIF sera responsable de l'analyse des données détectées et de la réponse aux anomalies détectées.

En termes de performances, le NNO a été formé et testé avec l'ensemble de données de détection d'intrusion *CUP99*, et il a montré de bonnes performances de classification. Il a été conclu qu'il pourrait être couplé à des mécanismes de réponse active dans le cadre de l'industrie 4.0 pour arrêter les cyberattaques.

VI. Protection des systèmes par la maintenance prédictive

Contexte et objectif Les Systèmes de Contrôle Industriels (ICS) sont de plus en plus ciblés par les pirates depuis le début du 21^{e} siècle en raison des dommages potentiellement importants qu'ils pourraient infliger au système et à son environnement en cas d'attaques réalisées avec succès. Le réseau ICS peut être divisé en trois niveaux [51] [52] :

— Niveau 0 : Partie opérationnelle avec capteurs/actionneurs ;
— Niveau 1 : Partie contrôle avec PLC/IHM ;
— Niveau 2 : Supervision avec salle de contrôle/SCADA ;

Les solutions informatiques telles que le pare-feu/DMZ sont généralement utilisées au niveau 2 et supérieur pour se protéger des attaques DDoS ou des attaques « Man in the Middle » (MITM). Les solutions de cette couche fonctionnent car elles sont très similaires à l'infrastructure informatique traditionnelle.

Cependant, ces solutions ne fonctionnent pas sur les couches temps réel 0 et 1, qui ont leurs propres attaques inhérentes (attaques aléatoires, fausse injection de données). Quatre types d'attaques sont considérées, à savoir directes, séquentielles, temporelles et sursollicitées.

Une méthodologie innovante a été proposée par [13], et repose sur le concept de « connaissance de l'automatisation ». Ceci est un approfondissement de [51] du même auteur. La solution sera dédiée à la protection des éléments bas niveau (niveau 0-1) comme les automates, les capteurs de l'architecture « Computer-Integrated Manufacturing » (CIM) en prenant en compte les aspects sûreté et sécurité. L'objectif final est de pouvoir détecter les ordres malveillants émis par l'automate.

Cette approche peut être considérée comme un dernier bouclier pour protéger le système des cyberattaques en supposant que les pirates ont déjà franchi les niveaux de défense précédents.

Conception et approche : modèles et filtres La première étape est basée sur des modèles comportementaux [53], qui représentent le fonctionnement normal du système. Elle est divisée en quatre parties :

— Évaluation des risques pour déterminer les zones critiques à protéger (Prérequis) ;
— Identification des paramètres (hors ligne) ;
— Contrôler la génération du modèle de filtre (hors ligne) ;
— Exécution des mécanismes de détection durant le fonctionnement (en ligne).

L'évaluation des risques est un préalable. La méthodologie suppose qu'une analyse de risque a été effectuée sur le système. Cette analyse doit mettre en évidence les événements redoutés pour le système et induire quelle partie de l'ICS doit être protégée en priorité. Pour cela, il est nécessaire de modéliser la liste des entrées-sorties (E/S) de l'automate.

L'identification des paramètres consiste à créer les états à partir de la liste d'E/S issue de l'évaluation des risques. Cela aidera à définir à la fois le périmètre et les états utilisés dans les modèles comportementaux. Mathématiquement, ces états sont définis avec des contraintes combinatoires basées sur les valeurs des capteurs et des actionneurs. Les types d'états suivants sont pris en compte dans un système de contrôle industriel :

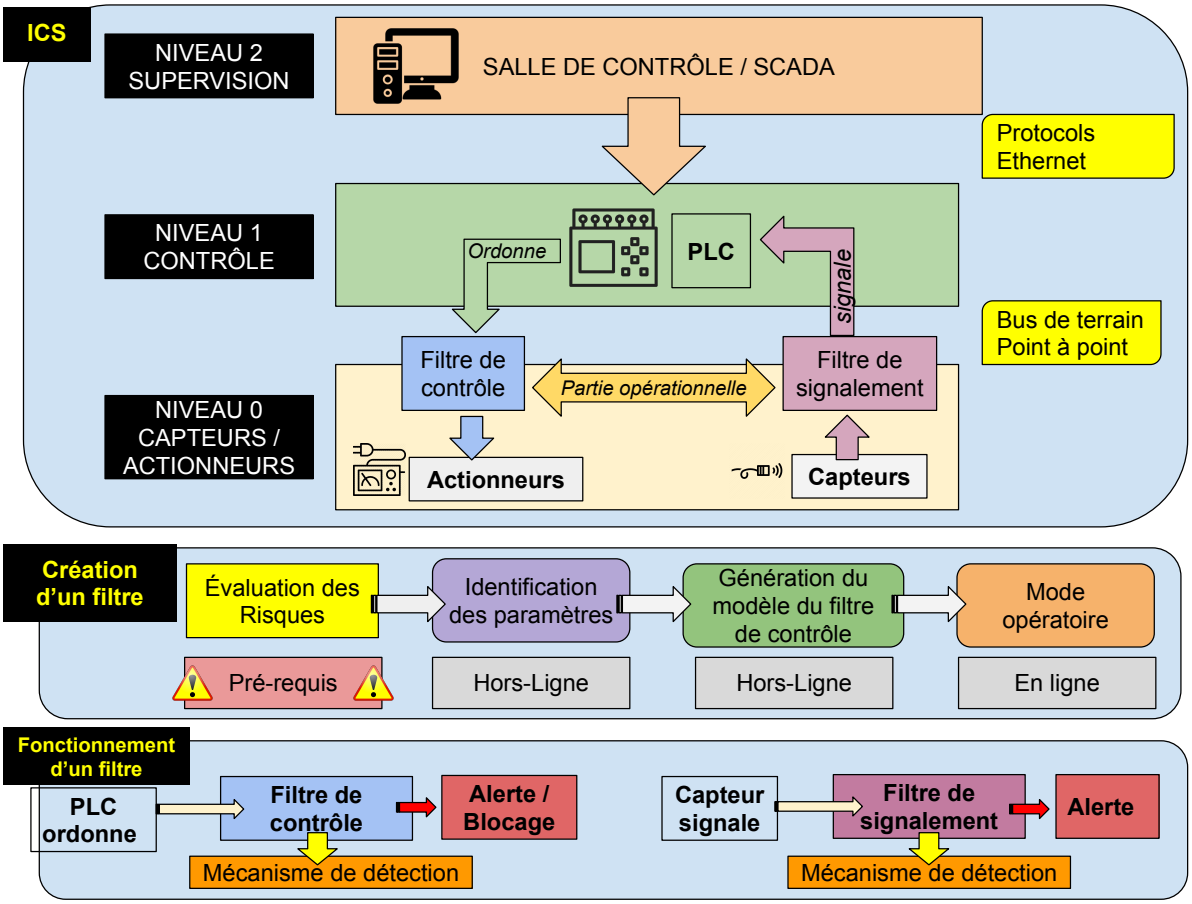


Figure 2.8 – Implémentation de filtres dans l'approche du modèle comportemental pour la protection des systèmes industriels

— Optimal (Respecte la loi de commande ainsi que les contraintes physiques du système) ;
— Dangereux (Respecter uniquement les contraintes physiques du système) ;
— Interdit (Endommage le système physique).
— Accessible (Tous les états tolérables).
— Inaccessible (Tous les états à éviter).

Ensuite, des contraintes temporelles doivent être déterminées pour identifier les attaques temporelles. Ces spécifications temporelles ajoutent une autre dimension à la caractérisation du comportement du système ainsi qu'un autre niveau de protection pour l'ICS. L'utilisation de contraintes combinatoires et temporelles est nécessaire pour assurer l'efficacité des algorithmes de détection.

La génération du modèle de filtre de contrôle vise à représenter le système industriel avec le modèle de processus et de contrôle. L'étape de fonctionnement correspond à la disponibilité des filtres de contrôle et de rapport dans l'ICS. Ces filtres doivent être situés juste après les capteurs pour garantir l'intégrité des commandes, et pour détecter tout comportement malveillant (Figure 2.8).

La question suivante est de déterminer le niveau de protection souhaité avec ces filtres. Trois niveaux de sécurité sont pris en compte :

— Sécurité des biens et des personnes : garantir un ensemble d'états où le système peut évoluer sans danger ;
— Qualité : garantit la bonne exécution de la loi de commande, et surveille le système ;
— Protection des équipements : surveille la sollicitation des actionneurs avec des commandes trop intenses ou trop fréquentes (afin de réduire l'usure, le risque de panne, et la maintenance des équipements).

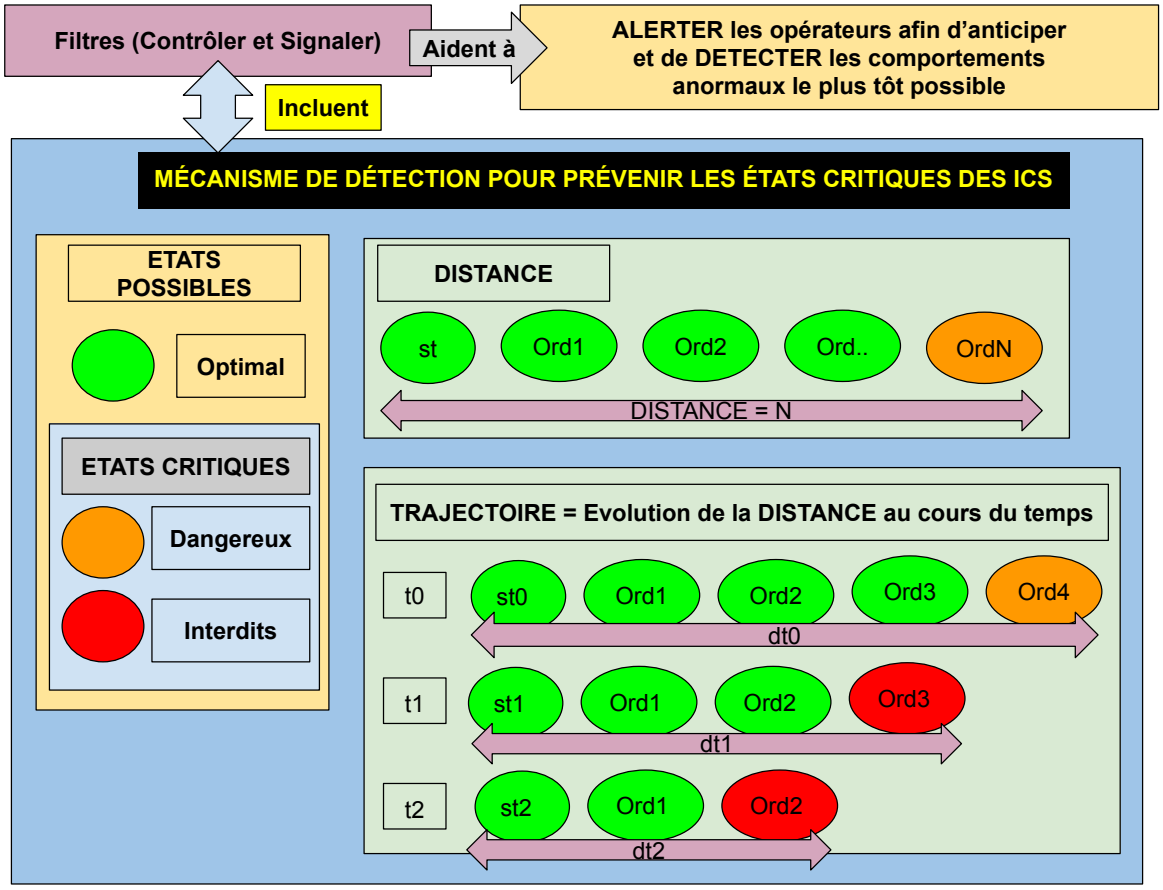


Figure 2.9 – Mécanisme de détection des états critiques basé sur la distance et la trajectoire pour protéger les systèmes industriels

Conception et approche : détection basée sur la notion de distance et de trajectoire Les filtres décrits dans le paragraphe précédent permettent la mise en place de règles, mais ils ne peuvent stopper les attaques qu'au moment où l'état critique est sur le point d'être atteint. Un mécanisme de détection des écarts par rapport au comportement normal est nécessaire pour compléter la méthodologie. Ce mécanisme repose sur trois notions, qui sont la distance, le plus court chemin vers l'état critique et la trajectoire. Figure 2.9 est une représentation simplifiée des mécanismes de détection proposés.

La notion de « distance » est liée aux états de l' ICS, et a été introduite dans [54]. Elle représente l'écart entre l'état actuel et un ensemble d'états critiques, qui sont tous des états que le système ne doit pas atteindre. Ce concept est très intéressant car il donne des indications aux opérateurs sur la proximité avec la zone critique. Le « chemin le plus court vers l'état critique » est le plus petit nombre d'ordres qui doivent être appliqués avant d'atteindre un état critique. L'objectif général est de calculer, pour chaque état atteignable, la distance la plus proche avec un état critique. Ces calculs sont effectués hors ligne et intégrés dans les filtres pour effectuer l'algorithme de détection, tout en tenant compte des contraintes de temps réel. La dernière notion est celle de « trajectoire » qui est complémentaire de la notion de distance. La distance ne donne que des informations ponctuelles pour le système, mais la trajectoire est définie comme « l'évolution de la distance en fonction de la séquence d'états ou du temps ».

Les mécanismes de détection suivants peuvent être mis en œuvre avec l'approche distance/trajectoire :

— Détection de contexte ;
— Détection d'anomalies (combinatoire et temporelle) ;
— Dégradation du matériel

Le principal avantage des mécanismes proposés est d'alerter les opérateurs, d'anticiper les déviations et de détecter les comportements anormaux à leur stade le plus précoce.

Résultats et mise en valeur L'auteur a confirmé que l'approche a montré de bons résultats pour la détection des cyberattaques qui affectent les systèmes physiques. L'amélioration notable a été le blocage des commandes avant de conduire le système dans un état critique, et la détection d'attaques séquentielles utilisant des contraintes combinatoires et temporelles. Cependant, certaines améliorations ont été suggérées, telles que :

— L'utilisation d'une approche de type sécurité/sûreté pour améliorer l'efficacité et par conséquent le nombre d'attaques prises en compte ;
— Considération de l'évolution du système entre deux états stables

Tableau 2.6 – État de l'art des solutions de cybersécuritié en industrie 4.0

Solutions de cybersécurité	*Équipement(s) concerné(s)*	*Protège de*	*Périmètre(s) de l'usine concerné(s)*
Protection des environnements physiques	Tous	Accès physique non autorisé, Information spreading (acoustic, visual, radio)	Tous
Pare-Feu défini par logiciel	Manufacturing Execution System (MES)	Sondage de réseau	Manufacturing-Production
Framework d'onthologie pour IoT	Périphériques IoT	Attaques externes (virus, vol de données, failles de sécurité)	Manufacturing-Production / Logistique
Authentification directe des systèmes cyber-physiques	Systèmes Cyber-Physiques (CPS)	Instructions non-autorisées vers les machines de production	Manufacturing-Production
Framework d'apprentissage ensembliste pour la fabrication avancée	Systèmes de Contrôle Industriels (ICS)	Détection prédictive des cyberattaques en utilisant les réseaux de neurones	Manufacturing-Production
Maintenance prédictive des ICS via l'établissement de modèles comportementaux	Systèmes de Contrôle Industriels (ICS)	Détection prédictive des cyberattaques en utilisant la notion de distance et les états critiques (ex : détection de sabotage)	Manufacturing-Production

2.4.4 Honeypots et Digital Twins

Vue d'ensemble

Plusieurs solutions innovantes sont disponibles dans la littérature, comme le système immunitaire pour la cybersécurité dans l'Industrie 4.0 [55], pour n'en citer que quelques-unes. Cependant, dans ce manuscrit, nous nous concentrerons sur deux approches populaires : les « honeypots » et les « digital twins » (jumeaux numériques). Les systèmes Honeypot sont des systèmes de surveillance passifs avec des capacités d'alerte précoce pour les environnements de production et les infrastructures critiques. Leur fonctionnalité principale est d'émettre des alertes si l'infrastructure a été violée par des pirates ou des activités liées à des logiciels malveillants.

Techniquement, un honeypot se compose de données qui semblent être une partie légitime du système avec des ressources précieuses pour les attaquants. Cependant, en fait, le honeypot est isolé et surveillé pour bloquer ou analyser les attaquants. Concrètement, le concept consiste à appâter les attaquants. Les principaux avantages de cette solution sont :

— Avoir conscience du fait que quelqu'un ou quelque chose essaie d'exploiter les systèmes critiques de votre entreprise
— L'attaquant perd un temps précieux à attaquer un faux système, au lieu de la véritable infrastructure
— L'équipe de sécurité a plus de temps pour arrêter l'attaque
— L'entreprise ne perdra pas de temps sur de fausses alertes positives car il n'y a aucune raison pour qu'il y ait la moindre communication depuis ou vers le système honeypot

I. Solutions basées sur les « honeypots »

Les solutions existantes basées sur des honeypots sont généralement des systèmes distribués capables de collecter et d'analyser les informations liées aux menaces ou aux attaques [14]. Le but de cette analyse est de déterminer le type d'attaque, l'existence d'appareils infectés, ainsi que les activités menées sur le système.

ThreatMatrix par Attica Networks ThreatMatrix est la principale plate-forme de détection basée sur les honeypots proposée par *Attica Networks* et capable de détecter les intrusions en temps réel dans les systèmes ICS/SCADA ainsi que dans les environnements IoT [14]. Son produit phare, BOTsink, est capable de détecter les APT sans être détecté par les attaquants. Certaines autres fonctionnalités sont également incluses, telles que des images logicielles pour simuler des dispositifs SCADA ainsi que leurs protocoles, dans le but de les rendre impossibles à distinguer des vrais dispositifs.

ICS Honeypot par Industrial Defenica En 2018, une étude mondiale menée par le Ponemon Institute pour le compte d'IBM a révélé que le temps moyen nécessaire pour identifier une violation de données est de 197 jours. De plus, les attaques les plus connues en milieu industriel ont impliqué des pirates présents dans le réseau depuis au moins trois mois avant d'être repérés. Sur la base de ces résultats, *Industrial Defenica* a proposé un honeypot à haute intéraction pour les systèmes industriels ICS/SCADA.

Cette solution utilise une technologie de tromperie avancée capable de présenter de fausses unités basées sur des modèles (PLC, périphériques Ethernet-série) sur le réseau. Plus de 3500 appareils différents peuvent être falsifiés (protocoles, services, ports ouverts, etc.), et ces appareils falsifiés peuvent communiquer avec de vrais équipements ICS. De nombreuses étapes sont nécessaires pour que les attaquants déterminent s'il s'agit d'un appareil réel, et ces étapes alerteraient l'équipe de sécurité que quelqu'un s'immisce dans l'infrastructure.

Pour fournir les meilleures données possibles sur les menaces, un réseau mondial « ICS Industrial Honeypot » a été créé pour obtenir des retours d'informations sur les solutions déployées, ainsi que pour améliorer les données, le support des équipements, de meilleurs profils d'équipement afin de les maintenir en tant que simulateurs crédibles.

« II. Digital twins » (jumeaux numériques) pour les systèmes cyberphysiques

Les jumeaux numériques ont été reconnus comme l'une des principales tendances technologiques stratégiques en 2019 par Gartner [56]. Ici, nous nous concentrons sur quelques cas d'utilisation qui montrent comment ils peuvent renforcer la sécurité des systèmes cyber-physiques.

Définition La définition standard des jumeaux numériques est qu'ils sont des répliques virtuelles d'objets physiques qui permettent de surveiller, de visualiser et de prédire les états des systèmes cyber-physiques [56]. Dans le contexte de la sécurité de l'information, [56] a proposé une définition uniforme basée sur la littérature : un jumeau numérique est « une réplique virtuelle d'un système qui accompagne son homologue physique pendant les phases de son cycle de vie, consomme du temps réel et de l'historique si nécessaire et est suffisamment fidèle pour permettre la mise en œuvre de la mesure de sécurité souhaitée ». Un autre terme est également évoqué à propos des jumeaux numériques : le fil numérique. Dans [56], le fil numérique est défini comme « la liaison de données ininterrompue tout au long du cycle de vie d'un système qui peut être utilisée pour générer et fournir des mises à jour à un jumeau numérique ».

Cas d'utilisation dans le domaine du manufacturing Plusieurs cas d'utilisation ont été identifiés pour le rôle des jumeaux numériques dans la sécurisation des systèmes de fabrication :

— Conception sécurisée des CPS ;
— Détection d'intrusion ;
— Détection de mauvaise configuration (matérielle et logicielle) ;

— Tests de sécurité ;
— Intimité ;
— Essais et formation du système ;
— Démantèlement sécurisé ;
— Sécurité et conformité légale ;

Pour concevoir des CPS plus sécurisés, l'idée est d'utiliser des jumeaux numériques en combinaison avec un environnement virtuel afin d'analyser le comportement du système face aux attaques. Ainsi, les ingénieurs pourraient estimer les dommages potentiels, facilitant ainsi le processus de conception des mécanismes de sécurité et de sûreté pour produire des architectures CPS plus robustes et tolérantes aux pannes. Les jumeaux numériques pourraient également aider à révéler des points faibles dans l'architecture ou des fonctionnalités inutiles dans les appareils qui pourraient les exposer à une intrusion.

Les jumeaux numériques peuvent également aider à mettre en place des systèmes de détection d'intrusion (IDS). En effet, [56] a présenté une approche de réplication d'état passive qui visait à répliquer un état d'un appareil physique vers un jumeau numérique. Cela permet au jumeau numérique de refléter le comportement du CPS réel pendant son fonctionnement. Ensuite, il est nécessaire d'implémenter un IDS basé sur des spécifications de comportement où le comportement normal du CPS est correctement défini pour détecter toute modification. Cette technique produit un faible taux de faux négatifs et peut détecter certaines attaques qui n'étaient pas connues au moment où le comportement légitime a été défini. Enfin, les intrusions peuvent être simplement détectées en comparant les entrées et les sorties des dispositifs physiques et leurs jumeaux numériques associés.

Comme les jumeaux numériques sont le résultat d'une émulation matérielle et logicielle d'appareils, ils imitent des fonctionnalités similaires. La détection d'une mauvaise configuration matérielle et logicielle consiste à observer différents comportements entre les jumeaux numériques et leur homologue physique. Si une différence est observée, cela pourrait indiquer des actions malveillantes. Contrairement à l'analyse traditionnelle des données de configuration où seul le logiciel est vérifié, ce cas d'utilisation s'applique également au matériel.

Les tests de sécurité dans les environnements opérationnels (OT) sont essentiels car ils sont effectués sur des systèmes en direct et peuvent causer de graves dommages ou des interruptions d'activité. Normalement, les bancs d'essai sont utilisés pour éviter les interférences, mais leur maintenance est coûteuse en temps et en efforts. Les jumeaux numériques permettent d'effectuer virtuellement des tests de sécurité au lieu de les mener sur les systèmes réels. Bien sûr, la fidélité du jumeau numérique est un point critique, et ce cas d'utilisation pourrait également s'appliquer à la phase d'ingénierie pour corriger les vulnérabilités tôt dans le système cyber-physique. Dans les phases opérationnelles, l'utilisation d'un jumeau numérique comme système honeypot pourrait également permettre de tester la sécurité du CPS face à de vrais attaquants, et contribuer à renforcer la sécurité réelle du CPS.

Le concept de jumeaux numériques peut également contribuer à protéger la vie privée, par exemple en aidant les responsables du traitement ou les sous-traitants à respecter les exigences du Règlement Général sur la Protection des Données (RGPD). Par exemple, un assureur proposant un produit d'assurance basé sur les données obtenues à partir des jumeaux numériques des voitures intelligentes. Comme les jumeaux numériques utilisent certaines méthodes pour classer les données qui peuvent être rendues anonymes avant le transfert des données au propriétaire, les droits à la vie privée peuvent être préservés.

Comme les jumeaux numériques sont virtuels et fonctionnent dans un environnement isolé, ils peuvent être utilisés comme plateforme de test et de formation. Cette plate-forme pourrait servir à tester de nouvelles défenses ou à s'entraîner à répondre aux cyberattaques. L'idée principale est de lancer des attaques contre les jumeaux numériques à partir de l'environnement virtuel à des fins de test et de formation.

Lorsque la fin de vie de l'ICS et du CPS est atteinte, les composants doivent être éliminés de manière sécurisée. De multiples aspects doivent être pris en compte, tels que les exigences de confidentialité des données, ainsi que les coûts associés à la désinfection. Les jumeaux numériques pourraient faciliter l'élimination sécurisée des appareils physiques. Cependant, ils peuvent être affectés par un accès non autorisé. Il est donc important que le fil numérique soit coupé et archivé correctement.

Les exigences réglementaires pour les opérateurs de CPS semblent augmenter, c'est pourquoi les jumeaux numériques pourraient aider en fournissant une réflexion précise sur les CPS tout au long de leur cycle de vie pour permettre une surveillance et une documentation continues des aspects de sécurité.

2.5 Contribution à la synthèse sur la cybersécurité

2.5.1 Bonnes pratiques techniques

La sécurité se doit également d'être abordée par le biais de capacités techniques et d'environnements appropriés où elles sont déployées [57] [58]. Cette section fera la promotion de ce que nous considérons comme des lignes directrices pour la cybersécurité dans l'Industrie 4.0.

Le Tableau 2.7 synthétise ces pratiques proposées par l'ANSSI. Pour chaque pratique, les détails suivants sont exposés :

— La raison pour laquelle elle doit être prise en compte ;
— La méthode pour l'appliquer ;
— Son périmètre (matériel, réseaux, infrastructure, etc.) ;
— Les contraintes liées à son application ;
— Comment gérer ces contraintes.

Tableau 2.7 – Bonnes pratiques en cybersécurité pour l'industrie 4.0

Pratique	*Motivation*	*Méthode*	*Périmètre*	*Contraintes*	*Moyens de gestion des contraintes*
Contrôle des points d'accès physiques	Déterminer les points d'entrée dans le système	Identifier les accès (qui ? pourquoi ? à quelle fréquence ?), Protéger les accès aux serveurs, équipements, câbles (salles informatiques, armoires verrouillées)	Postes de travail, serveurs, périphériques réseau, machines, écrans tactiles	Taille du système, maintenir l'accès même en cas d'urgence	Porte « contact sec » avec alarmes, procédures bris de glace
Ségrégation réseau	Limiter la propagation des attaques et contenir les vulnérabilités	Établir une carte de flux, filtrer, tracer et séparer le réseau avec les VLAN	Réseaux (SCADA, CPL, développement...)	Contraintes temps réel (réseaux de processus)	Filtrage appliqué en amont, contrôle d'accès physique aux équipements du réseau
Gestion des appareils portables	Réduiser les risques d'attaques de logiciels malveillants via des supports portables (usb, hdd...)	Définir la politique, les restrictions logicielles, restreindre l'utilisation de ce média, les ports USB	Postes de travail, serveurs, consoles, écrans tactiles	Nécessité d'échanger des données entre réseaux non interconnectés	Machines propres (renforcées, sécurisées) dédiées aux transferts de données
Gestion de compte	Protège des accès non autorisés	Politique de comptes (utilisateurs/applications), pas d'identifiants par défaut, mots de passe forts changés régulièrement	OS, bases de données, applications (SCADA/-PLC), périphériques réseau, machines	Comptes génériques, accès d'urgence	Actions de traçabilité, procédures strictes pour déterminer l'identité avec des comptes génériques (instant par instant)

Pratique	*Motivation*	*Méthode*	*Périmètre*	*Contraintes*	*Moyens de gestion des contraintes*
Durcissement de la configuration	Limiter les zones exposées aux attaques	Installer uniquement les logiciels, protocoles et services nécessaires, éviter les options par défaut, désactiver les protocoles vulnérables	OS, applications (SCADA/PLC), périphériques réseau, écrans tactiles	Impact des modifications sur les applications de production	Analyse documentée, gestion des exceptions pour les fonctionnalités non sécurisées nécessaires
Surveillance, avertissements et alarmes	Détection des intrusions, traçage des actions, interventions de maintenance	Activer les fonctions de traçabilité (syslog, événements windows...), filtrer et générer des alertes pour les événements pertinents	OS, bases de données, équipements réseau, automate...	Volume élevé de journaux générés	Outils de gestion des événements (filtrer, limiter, supprimer...)
Gestion de la configuration	S'assurer qu'il n'y a pas de modifications malveillantes entre les versions	Comparaison entre les applications exécutées et la configuration de l'application de référence, identification des variations avant déploiement	Apps (SCADA/PLC), périphériques réseau (fichiers de configuration)	Complexité et hétérogénéité des ICS	Outils de gestion de configuration pour identifier les variations entre deux versions
Sauvegarde et restauration	Posséder des données en cas de redémarrage complet après une attaque ou un sinistre	Politique de sauvegarde (quelles données sauvegarder pour les utilisateurs ou en fonction des exigences réglementaires ?)	Codes sources, bases de données, historiques, firmwares des automates, configs des équipements réseaux (switchs, routeurs...)	La sauvegarde ne peut pas être effectuée automatiquement pour certains appareils (capteurs, actionneurs pour automates)	Tracer les modifications de réglages, pilotage, ajustements, alarmes des capteurs/actionneurs
Documentation	Avoir une représentation exacte des systèmes et maîtriser la diffusion des informations	Politique de documentation (processus de mise à jour, durée de conservation, stockage...)	Usine, schémas architecturaux, emplacements, manuels d'administration et de maintenance, analyse du système...	Les copies papier des documents peuvent contenir des mots de passe (astreinte), leur contrôle est compliqué	Sensibilisation des utilisateurs aux risques avec documentation, aucun document en vue sur le bureau...
Détection de code malveillant	Protection avancée contre les attaques de virus	Politique de protection, priorité aux matériels / applications en contact direct avec l'extérieur et les utilisateurs	Applications SCADA, stations d'ingénierie, consoles de programmation et de maintenance	Incompatibilité avec les anciennes applications, pas de mise à jour antivirus, problèmes contractuels (perte de garantie)	Déployer l'antivirus sur les machines portables et de maintenance, renforcement de la configuration

Pratique	*Motivation*	*Méthode*	*Périmètre*	*Contraintes*	*Moyens de gestion des contraintes*
Gestion des mises à niveau et des correctifs	Protection préventive contre les attaques, les pannes liées aux bugs, les vulnérabilités	Politique de gestion des correctifs (systématiques ou périodiques) adaptés aux contraintes, risques et matériels identifiés	OS, applications, firmware, machines opérateurs, serveurs, automates, équipements télécoms, écrans tactiles...	Les correctifs doivent être évalués avant le déploiement, certains appareils ne sont pas faciles à arrêter	Identifier les vulnérabilités, planifier les mises à jour, surveiller le trafic, renforcer les configurations et isoler les appareils
Protection des automates	Protection des programmes automates	Accès protégé (mots de passe) à l'automate, code source, accès en lecture seule pour la maintenance de premier niveau, verrouillage des armoires de l'automate	Automate de production, programmes automates	...	...
Stations d'ingénierie et de développement	Points de vulnérabilité et vecteurs de contamination (portables, connexion à d'autres réseaux)	Correctifs, logiciels antivirus, pas de connexion hors SCADA, surveillance de l'utilisation, arrêt lorsqu'il n'est pas utilisé	Stations de développement SCADA, console automate, appareils portables pour configurer capteurs et actionneurs	...	...

Cependant, certaines mesures liées à l'IoT et à la fabrication intelligente font défaut. C'est pour cette raison qu'elles seront détaillées dans les sous-sections suivantes.

Confiance et intégrité Cette pratique permet de garantir l'intégrité et la fiabilité des données et des appareils. Les mesures suivantes doivent être appliquées :

- Vérifiez l'intégrité et la source du logiciel avant de l'exécuter ;
- Pour les appareils IoT, autorisez-les au sein du réseau en utilisant des certificats numériques/PKI ;
- Définition des canaux d'échange de données sécurisés pour les appareils IoT (liste blanche) ;
- Mise en place de listes blanches d'applications et révision périodique (annuelle) de la liste en cas de changement ;
- Utilisation de mécanismes de cryptographie pour assurer l'intégrité des données de production ;
- Surveillez les données au repos et en transit pour détecter les modifications non autorisées.

Sécurité du cloud L'aspect sécurité du cloud computing est concerné par cette pratique. Les mesures suivantes doivent être appliquées :

- Choix du type de cloud en fonction de l'entreprise, de l'impact sur la vie privée, des lois, du pays du fournisseur de cloud ;
- Inclusion des aspects de sécurité et de disponibilité dans les accords avec les fournisseurs de cloud ;
- Éviter le point de défaillance unique avec les applications cloud et les systèmes centralisés ;
- Détermination des systèmes et applications critiques lors de l'utilisation du cloud public ;

— Pour réduire les risques d'attaques dans le cloud, utilisez une approche sans connaissance et protégez toutes les données dans le cloud et pendant le transfert.

Sécurité machine à machine Le concept de sécurité machine à machine est lié au stockage des clés, au chiffrement, à la validation des entrées et à la protection lors des communications machine à machine. Les mesures suivantes doivent être appliquées :

— Utilisation d'un serveur-HSM dans l'infrastructure pour stocker les clés de la couche de service à long terme ;
— Association de sécurité entre les entités communicantes et les algorithmes de cryptographie pour assurer l'authentification mutuelle, l'intégrité et la confidentialité ;
— Utilisation de protocoles de communication capables de détecter la répétition non autorisée de messages antérieurs ;
— Pour vous protéger contre les scripts intersites et l'injection de commandes, utilisez la validation des entrées de la liste blanche.

Protection des données La protection des données est un concept très général, mais dans ce cas, il est lié à la garantie de la confidentialité des données à différents niveaux au sein d'une entreprise et à la gestion de l'accès aux données. Les mesures suivantes doivent être appliquées :

— Protection des données au repos (mémoire volatile et non volatile), en transit et en cours d'utilisation.
— Catégorisation des données basée sur l'analyse des risques, l'évaluation de la criticité et la définition des mesures de sécurité.
— Accordez l'accès à certaines données à des tiers disposant du moindre privilège et documentez cet accès.
— Pour des données hautement confidentielles, utilisez des solutions de chiffrement, de gestion des clés et de prévention des pertes de données.
— Sécurisez et anonymisez les données personnelles directes et indirectes traitées par le biais de contrôles d'accès, de rôles et de cryptage.

Logiciel / Mises à jour du micrologiciel Comme tout autre appareil, les mises à jour logicielles sur les solutions IoT doivent suivre une méthodologie spécifique, dont les mesures sont :

— Assurer un contrôle serré de la mise à jour (aucune altération entre la source et la destination).
— Effectuer le déploiement des correctifs uniquement après avoir testé et prouvé qu'il n'y a pas de conséquences négatives.
— Pour les systèmes qui ne peuvent pas être mis à jour, des mesures compensatoires doivent être appliquées.

Contrôle d'accès Les appareils IoT sont des appareils critiques dans l'industrie 4.0 et peuvent être accessibles physiquement ou à distance. C'est pourquoi les mesures de contrôle d'accès suivantes doivent être appliquées :

— Niveau minimum d'authentification et d'autorisation pour un certain segment du système.
— Implémentation de l'authentification multi-facteurs.
— Applique le principe du moindre privilège.
— Implémentation d'une fonctionnalité de verrouillage de compte.
— Ségrégation de l'accès à distance.
— Prise en compte de la sécurité des accès physiques.

Réseaux, protocoles et cryptage La mise en œuvre appropriée du protocole, le chiffrement et la segmentation du réseau sont des éléments clés pour construire un réseau IoT solide. Les mesures suivantes sont liées à cet objectif :

— Utilisation de canaux de communication sécurisés et cryptage lorsque cela est possible ;
— Utilisez des protocoles éprouvés basés sur des normes (TLS 1.3) et évitez les protocoles vulnérables (Telnet, SNMP v1/v2) ;
— Limite le nombre de protocoles dans un environnement donné et désactive les services inutilisés.
— Garantit les capacités de sécurité et l'interopérabilité entre les protocoles.

Suivi et audit Plusieurs mesures sont suggérées pour l'environnement IoT concernant le trafic réseau, la surveillance de la disponibilité et l'examen des journaux :

— Implémentation d'une solution de surveillance passive pour créer une base de référence du trafic réseau industriel ;
— Analyse des journaux de sécurité en temps réel à l'aide de solutions SIEM (Security Information and Event Management) ;
— Revue périodique des logs, des privilèges d'accès et des configurations ;
— Surveillez la disponibilité des appareils IoT en temps réel.

Gestion de la configuration L'IoT comprend une large gamme d'appareils, il est important de garder une trace des changements dans les configurations, le renforcement des appareils et les méthodes de sauvegarde. Les mesures suivantes vont dans le sens de cet objectif :

— Configurations de sécurité de base adaptées aux différents types d'actifs ;
— Documenter tout changement de configuration conformément à la politique de gestion des changements de l'organisation ;
— Implémentation d'outils de support permettant la gestion de la configuration ;
— Création d'un plan de sauvegarde complet, adapté aux différents types d'actifs.

2.6 Synthèse entre industrie 4.0, cybersécuritié et traçabilité

Dans ce chapitre, nous avons exploré les concepts et les solutions de cybersécurité dans le contexte de l'industrie 4.0 à travers la littérature scientifique et normative. Dans les articles analysés, certains aspects de la cybersécurité n'étaient pas toujours évoqués, comme l'aspect managérial de la cybersécurité ou les impacts business [59]. Pour cette raison, notre étude a été structurée comme une approche étape par étape pour donner une vue complète du sujet. En plus d'être un examen introductif des travaux scientifiques concernant la cybersécurité, il peut également être utilisé comme guide pour présenter la cybersécurité dans un environnement de « smart manufacturing ».

Après avoir introduit le concept d'Industrie 4.0, nous avons montré la complexité induite par les mécanismes de cybersécurité pour assurer la sécurité des systèmes, en raison des nombreuses technologies innovantes impliquées.

Nous avons ensuite proposé une caractérisation de la cybersécurité sous ses aspects à la fois techniques et managériaux pour montrer que chaque niveau d'une organisation a un rôle à jouer. Etant donné que la cybersécurité doit être considérée en fonction des zones physiques, nous avons proposé un découpage d'une usine en plusieurs périmètres, que nous avons caractérisés en fonction de leurs interactions, équipements et réseaux. Nous avons ensuite rapporté les vulnérabilités, menaces et risques de cybersécurité rencontrés dans les usines de l'Industrie 4.0 pour chaque périmètre, tout en prenant en compte les impacts métiers au niveau organisationnel (Figure 2.10)).

D'une part, l'ouverture de l'usine à l'internet afin de permettre la transmission et le partage de l'information en temps réel avec l'extérieur va conduire à la création d'une passerelle entre les réseaux ce qui peut être

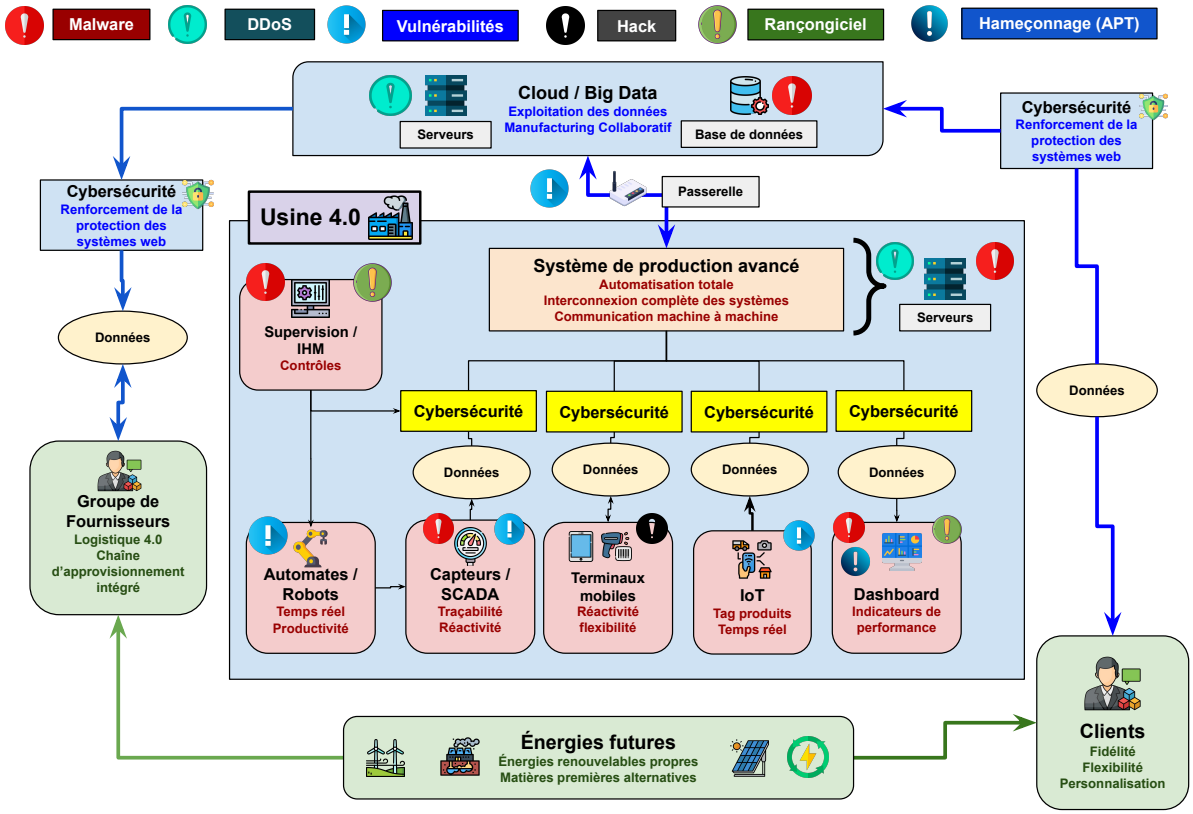


Figure 2.10 – Synthèse des risques de cybersécurité dans une usine 4.0

à l'origine de nouvelles failles de sécurité. D'autres part, les périphériques de l'usine présentent chacun à leur manière des risques en terme de cybersécurité (hack pour les terminaux mobiles, vulnérabilités pour l'IoT, DDoS pour les serveurs etc...) Parmi ces nouvelles technologies, certaines sont étroitement liées avec les notions d'identification et de traçabilité des produits dans l'usine (capteurs, IoT, tags RFID...).

On peut voir que l'usine 4.0 sera gérée par des données d'origine différentes, internes comme externes. Avec l'aide de l'externalisation du stockage via le cloud et le Big Data, l'usine pourra prendre en charge l'ensemble des données liées à un produit, de sa conception à sa fin de vie, de sa production à sa commercialisation.

Point encore plus important, l'usine devra être en mesure de récupérer, d'intégrer et d'exploiter des données provenant de son secteur d'activité en général et de ses clients en particulier. Une fois l'ensemble de ces données compilées, elle devra être en mesure de fournir une analyse prédictive cohérente.

L'objectif étant de permettre aux industriels de prendre en temps réel les meilleures décisions.

Traçabilité orientée produit et Blockchain

3 Approche globale de la traçabilité orientée produit dans l'usine 4.0

La traçabilité est un concept large qui fait référence à la pratique consistant à identifier un objet ou un élément de travail et à accéder à tout ou partie des informations le concernant, n'importe où dans son cycle de vie. Avec l'avènement de l'industrie 4.0 et les changements majeurs dans le fonctionnement des processus industriels, le concept de traçabilité inclut un aspect stratégique qui va au-delà de la simple identification des produits et de leur emplacement dans l'usine. Les technologies telles que le cloud ou l'IoT permettent des échanges entre le système de traçabilité de l'usine et les partenaires telles que les clients et les fournisseurs afin de pouvoir améliorer la confiance entre ces derniers en garantissant la qualité et la sécurité des produits. Cependant, la transparence apportée peut se faire au détriment des risques d'exposition des données confidentielles de l'usine. Ce chapitre sera donc consacré à la présentation de notre approche concernant la traçabilité orientée produit et sur la façon d'améliorer la confiance entre les partenaires sans pour autant compromettre la confidentialité des données.

3.1 Principes de base

Dans un monde où qualité et sécurité sont les maîtres mots, la traçabilité revêt une importance capitale et gagne de nombreuses industries, notamment automobile, composants électroniques, alimentaire et pharmaceutique. Cette section décrit les principes de base de ce concept en plein essor.

Bien qu'il soit possible d'envisager la traçabilité de diverses manières, elle peut généralement être abordée sous deux angles : traçabilité de la chaîne et traçabilité interne.

3.1.1 Traçabilité de la chaîne

Le concept général de traçabilité dans le monde fait référence à la traçabilité de la chaîne. La traçabilité de la chaîne signifie qu'il est possible de tracer l'historique en amont ou en aval depuis l'approvisionnement en matières premières et en pièces jusqu'à la vente en passant par l'usinage, l'assemblage et la distribution (Figure 3.2).

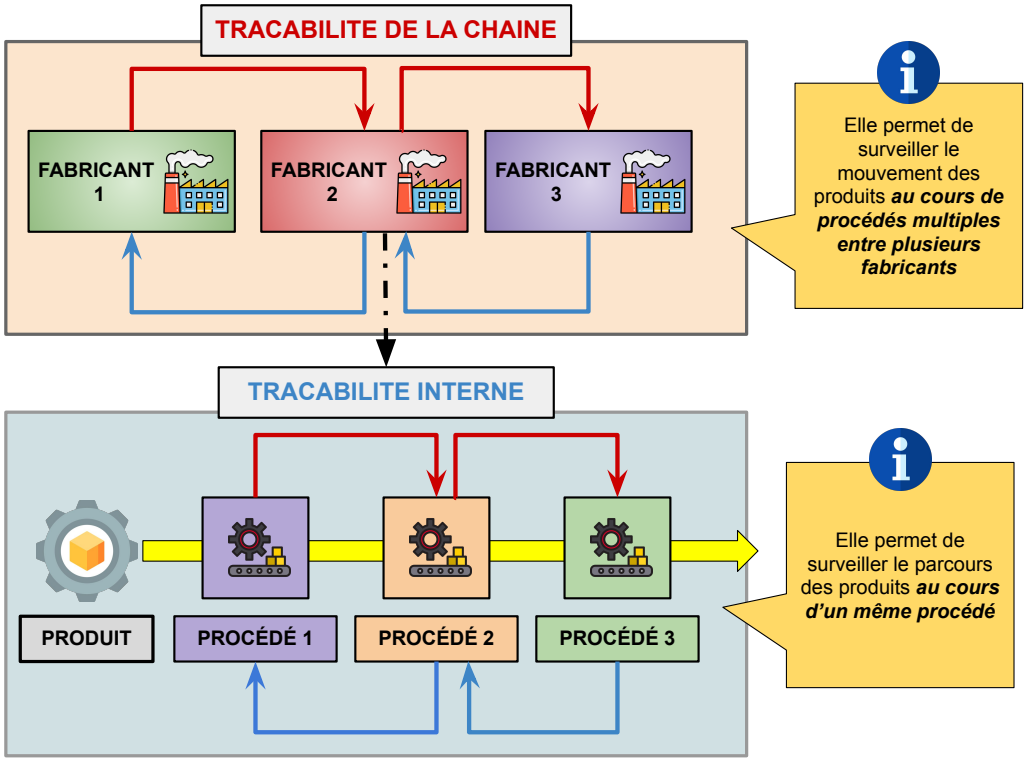


Figure 3.1 – Comparaison entre la traçabilité de la chaîne et la traçabilité interne

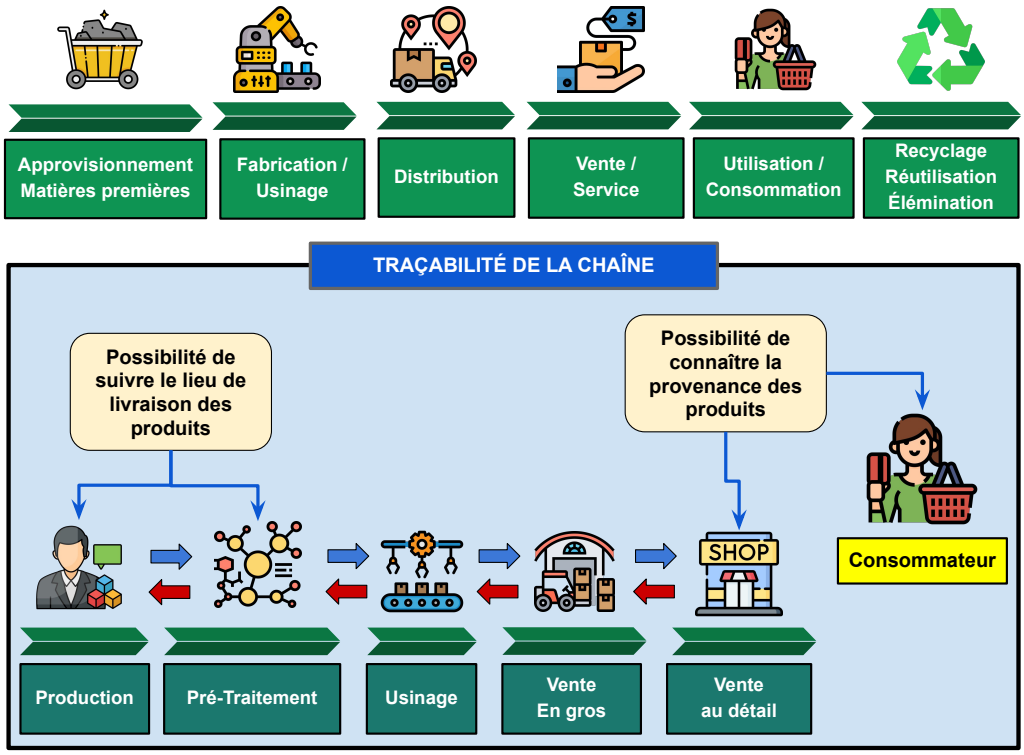


Figure 3.2 – Vue d'ensemble de la traçabilité de la chaîne

Les fabricants peuvent suivre « où les produits ont été livrés (= les tracer en aval) » tandis que les sociétés et les consommateurs situés en aval peuvent connaître « la provenance des produits qu'ils ont entre les mains ». Cela offre aux fabricants l'avantage de faciliter la recherche des causes et le rappel de leurs produits quand

ces derniers connaissent des problèmes imprévus. Les consommateurs peuvent également s'en servir de référence pour sélectionner des produits hautement fiables, sans ennuis tels que les erreurs d'étiquetage.

3.1.2 Traçabilité interne

La traçabilité interne désigne le suivi du parcours des pièces/produits au sein d'un espace spécifique limité d'une chaîne d'approvisionnement globale, telle qu'une société ou une usine.

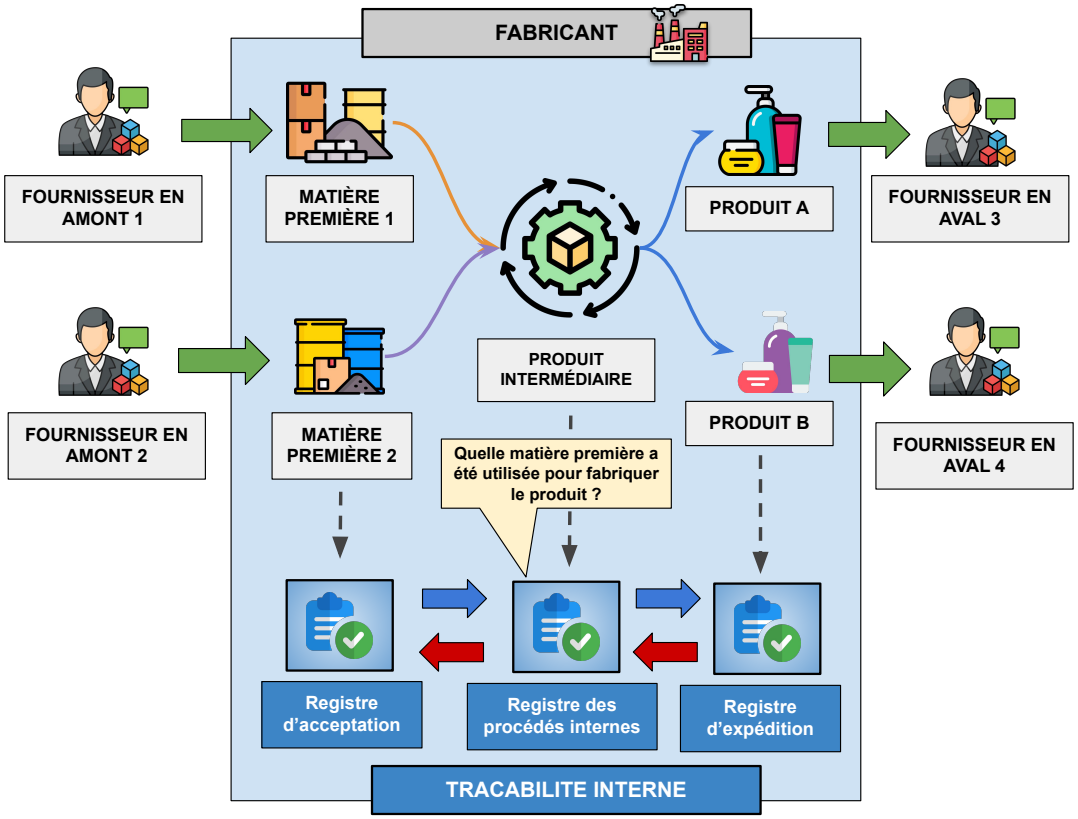


Figure 3.3 – Vue d'ensemble de la traçabilité interne

Par exemple, une usine d'assemblage de moteurs se procure des pièces de moteurs comme des arbres à cames et des pistons auprès de ses fournisseurs et les assemble. La gestion et l'utilisation de l'historique de fabrication ainsi que des résultats d'inspection de ces pièces par l'usine peuvent également être assimilées à la traçabilité interne.

3.1.3 Exemples de traçabilité interne

Traçabilité des procédés de fabrication

La traçabilité des procédés de fabrication se réfère à la collecte et au traitement des informations relatives à ce qui a été fait au cours des procédés de fabrication, depuis la réception des matières premières et des pièces jusqu'à l'expédition des produits.

Pour la traçabilité des procédés de fabrication, il est attribué à chaque produit ou lot un numéro d'identification, auquel sont rattachées, à chaque procédé, des informations telles que les détails des tâches, les résultats d'inspection et les dimensions, à des fins d'exploitation lors d'un procédé ultérieur (Figure 3.4). L'exploitation de ces informations lors de l'usinage contribue à améliorer le rendement et la qualité.

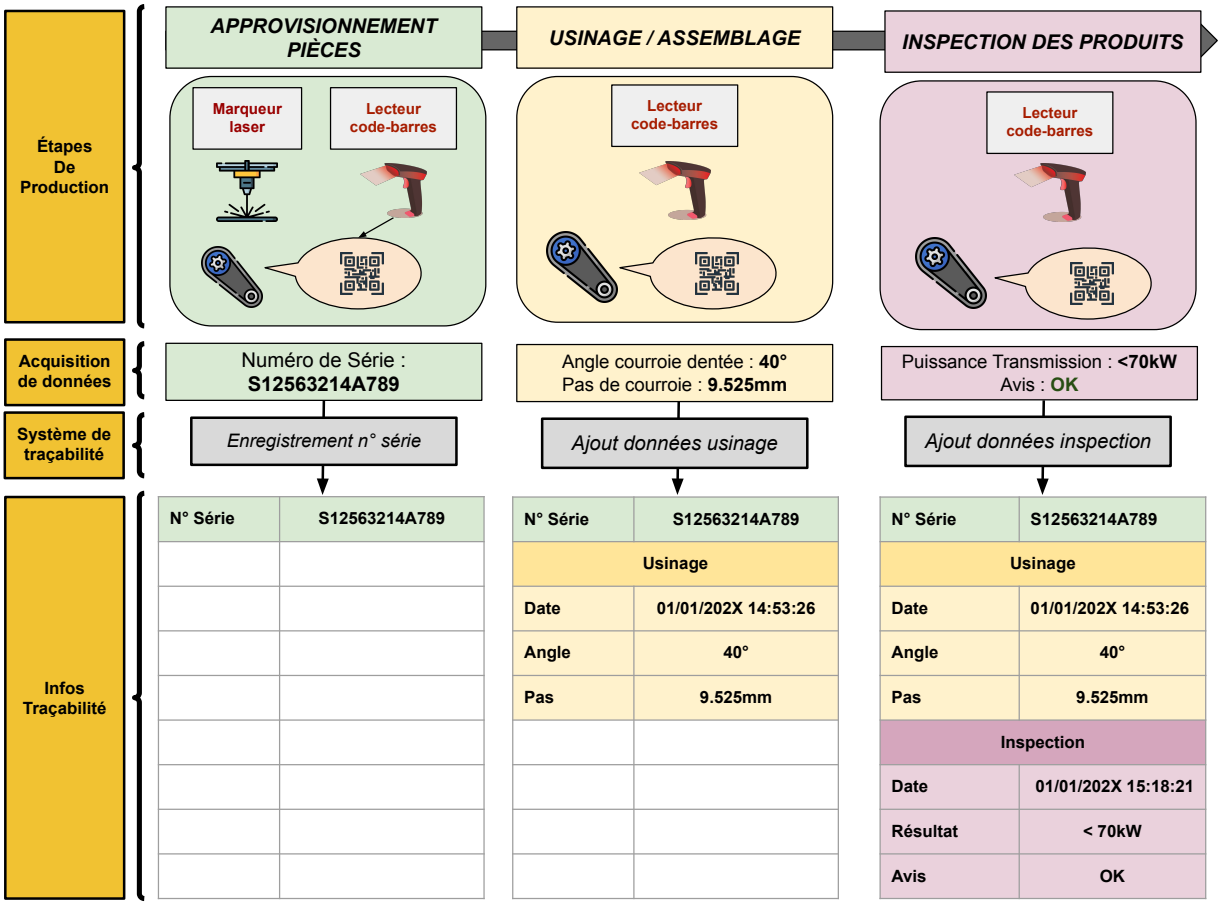


Figure 3.4 – Exemple de traçabilité interne au cours de multiples procédés de fabrication

Traçabilité du contrôle des pièces

La traçabilité de gestion des pièces facilite le contrôle et l'analyse de l'usure récurrente de pièces telles que des outils et gabarits. Un numéro de série servant à l'identification individuelle (tel qu'un code 2D) est marqué sur chaque outil pour la gestion de son état, notamment sa durée d'utilisation et ses limites d'usure. Comme pour l'ensemble du flux, les outils et autres pièces entreposés sont marqués de codes 2D de gestion et des informations leur sont affectées, telles que le nom de l'usine, le numéro de rayonnage et le numéro de série, pour la gestion des sorties et des retours. Des informations, telles que le nombre de meulages et leur date/heure, sont collectées et gérées pour garantir et uniformiser la qualité des produits.

3.1.4 Traçabilité en aval et en amont

Dans la traçabilité, les produits sont identifiés individuellement ou par lots pour accumuler des informations à chaque procédé. Tracer en aval signifie utiliser les informations accumulées pour suivre le parcours des produits et tracer en amont signifie suivre les registres en remontant la chronologie. Toutefois, il ne suffit pas d'identifier les pièces et produits et d'accumuler des informations. La traçabilité n'est assurée que si ces informations sont accessibles et peuvent être tracées en aval comme en amont à tout moment.

Traçabilité en aval

La traçabilité en aval consiste à tracer un produit en suivant la chronologie. Lorsqu'un défaut est détecté sur certaines pièces, par exemple, il est possible d'identifier les produits contenant ces pièces pour effectuer

un rappel précis. Il s'agit ainsi d'une mesure de prévention efficace des rappels et produits défectueux (Figure 3.5).

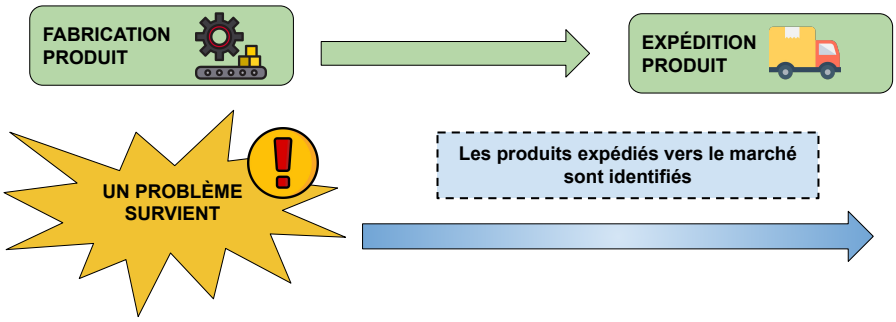


Figure 3.5 – Représentation visuelle de la traçabilité en aval

Traçabilité en amont

La traçabilité en amont consiste à tracer un produit en remontant la chronologie. Lorsqu'un problème survient sur des produits reçus, par exemple, il est possible d'identifier le lot et le procédé concernés en retraçant le registre de fabrication pour enquêter rapidement sur la cause. L'identification d'un lot ou d'un procédé permet de prendre rapidement les mesures nécessaires pour résoudre les problèmes et ainsi assurer une qualité supérieure et constante (Figure 3.6).

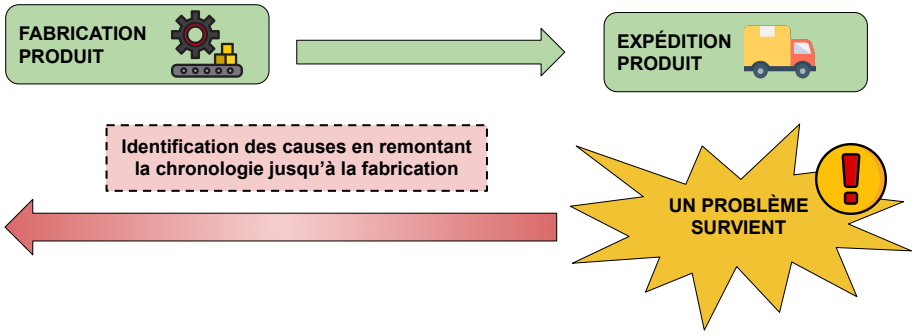


Figure 3.6 – Représentation visuelle de la traçabilité en amont

3.1.5 Données de traçabilité et formats d'identification

La mise en œuvre de la traçabilité en amont et en aval repose sur la collecte et l'enregistrement des informations appropriées à chaque procédé.

Les informations requises dépendent de plusieurs facteurs, notamment de l'industrie, du produit et des procédés. Dans l'industrie manufacturière, par exemple, il est important de connaître l'expéditeur, les détails de fabrication, la date et l'heure de fabrication, la date et l'heure d'inspection, le responsable, la ligne de production et le destinataire. Dans l'industrie alimentaire, il s'agit plutôt du producteur, du lieu de production, de la date et de l'heure d'expédition, de la date et de l'heure de traitement, de la date d'expiration et du destinataire. Le Tableau 3.1 présente une liste d'informations de traçabilité classée par origine de l'information.

Définition des symboles d'identification

Afin de garantir un traitement efficace des informations, il est indispensable d'établir des règles de définition des symboles d'identification. Si les règles établies ne sont pas respectées et que les symboles utilisées par ces

Tableau 3.1 – Exemple d'informations à collecter dans le cadre de la traçabilité

Origine de l'information	*Exemples d'information de traçabilité*
Réception	Quantité arrivée Date et heure d'arrivée N° de série / lot Informations sur le producteur, fabricant...
Fabrication / Usinage	Quantité fabriquée / traitée Date et heure de fabrication / traitement N° de série / lot Historique de fabrication Historique d'inspection...
Expédition	Quantité expédiée Date et heure d'expédition N° de série / lot Informations sur le client Informations sur l'usine de production

formats se confondent, les informations ne peuvent être identifiées avec précision, compromettant ainsi la traçabilité en aval et en amont.

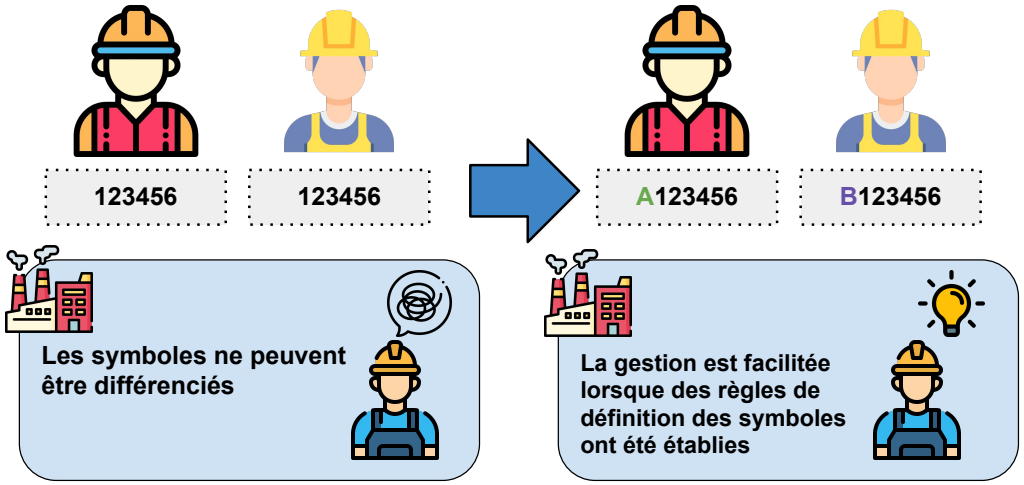


Figure 3.7 – Définition des formats d'identification

Lorsque plusieurs fournisseurs ou fabricants sont impliqués dans la réalisation d'un produit, il devient indispensable d'établir une règle de définition des symboles d'identification commune à toutes les entreprises concernées.

Formats de représentation et support de transfert

La palette des symboles d'identification utilisés pour transférer des informations est très vaste, allant des chiffres et caractères reconnaissables par l'oeil humain aux codes-barres, codes 2D et données électroniques. Ces symboles et méthodes de représentation sont appelés « formats de représentation ». Les formats de représentation se réfèrent uniquement aux symboles tels que les caractères ou codes 2D. Ils sont associés à un support de transfert (support de stockage) tel qu'une étiquette papier ou une étiquette à puce apposée sur le produit ou la pièce.

Par exemple, supposons que l'on colle une étiquette sur un carton contenant une pièce reçue. On y inscrit la date de livraison et le nom du produit au stylo. Dans cet exemple, l'étiquette est le support de transfert et les

caractères indiquant la date de livraison ainsi que le nom du produit sont le format de représentation. Si ces informations sont directement inscrites sur le carton, celui-ci devient le support de transfert. En revanche, si on inscrit les données au laser directement sur la pièce, les caractères inscrits sont le format de représentation et le support de transfert est inexistant car il s'agit du produit lui-même.

3.1.6 Collecte des données

Il est possible d'adjoindre des informations à une pièce ou un produit au moyen d'un format de représentation et d'un support de transfert. Cependant, pour assurer la traçabilité, ces informations ne doivent pas seulement être apposées, elles doivent être collectées et gérées.

Supports d'enregistrement

De ce fait, il est nécessaire de lire le format de représentation sur le support de transfert, collecter les informations qu'il contient et les gérer. Les registres papier, les ordinateurs ou les serveurs cloud sont autant de supports permettant d'enregistrer et de gérer les informations. En traçabilité, ils sont appelés **supports d'enregistrement**.

Les informations peuvent être collectées et gérées soit en les écrivant à la main sur un registre, soit en les saisissant ou en les lisant et en les transférant avec un lecteur de codes sur un PC ou un serveur. Une fois que les règles facilitant la récupération des informations lorsque cela est nécessaire ont été établies, la traçabilité en amont et en aval peut être mise en oeuvre.

Cas d'utilisation de collecte de données de traçabilité

Chaîne de caractères et support papier Dans ce contexte, on suppose l'utilisation de des caractères et des chiffres pouvant être reconnus par l'œil humain en tant que format de représentation et des étiquettes ou des autocollants en tant que support de transfert. Par exemple, des chiffres tels que *20XX0101* pour afficher la date d'expiration, des caractères tels que *LOT000001* pour indiquer le numéro de lot et des caractères tels que *ABC0001* pour préciser le numéro de série. Les supports papier employés en tant que support de transfert se divisent en deux catégories : les supports intégrés au produit, notamment les étiquettes, les emballages et les supports inclus avec le produit, par exemple les factures et bons de livraison. Concernant le support d'enregistrement, il est fréquent de recourir au registre papier. L'enregistrement direct sur PC via un terminal portatif avec fonction Reconnaissance optique des caractères (OCR) gagne cependant en popularité.

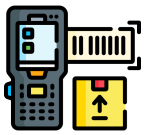
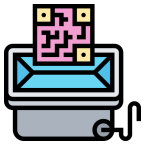
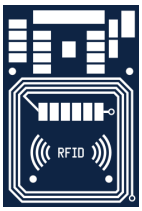
Code-barres et support papier Dans ce scénario, on utilise des codes-barres en tant que format de représentation et on les imprime sur des étiquettes ou emballages servant de support de transfert. Les codes-barres n'étant pas lisibles par l'œil humain, un lecteur de codes ou tout autre dispositif de lecture est requis. Une imprimante et un logiciel sont également nécessaires pour imprimer le code-barres sur une étiquette ou un autre support de transfert. Le support d'enregistrement est un ordinateur ce qui permet une gestion aisée des informations, une lecture sans contact et une mise en oeuvre avec un coût relativement faible. Cette combinaison est utilisée dans diverses industries notamment manufacturière, alimentaire, pharmaceutique et distribution.

Code 2D et marquage direct L'utilisation des codes 2D en tant que format de représentation se répand depuis quelques années. Les codes 2D permettent d'accumuler davantage d'informations sur une plus petite surface. De plus, la précision de lecture ne cesse d'être améliorée. Le marquage direct consiste à inscrire le code directement sur la pièce. L'absence de support de transfert garantit une réduction des coûts d'exploitation.

Données électroniques et étiquette à puce (étiquette RFID) Les données électroniques sont utilisées en tant que format de représentation et une petite étiquette à puce (étiquette RFID) en tant que support de transfert. Les étiquettes à puce communiquent les informations par ondes radio. Elles offrent l'avantage de permettre l'acquisition des informations même à distance ou sur plusieurs étiquettes simultanément. L'inconvénient demeure toutefois leur coût élevé.

Le Tableau 3.2 est une synthèse des différentes méthodes de collecte présentées ci-dessus mettant en avant leurs caractéristiques ainsi que leurs avantages et inconvénients.

Tableau 3.2 – Synthèse des cas d'utilisation de collecte de données de traçabilité

Cas d'utilisation	*Format de représentation*	*Support de transfert*	*Support d'enregistrement*	*Avantages +*	*Inconvénients -*
Code-barres et support papier				+ Gestion aisée des infos + Lecture sans contact + Coût faible	- Résistance / durée de vie - Ne peut être lu qu'avec un lecteur - Capacité de stockage restreinte
Code 2D et marquage direct				+ Capacité de stockage élevée + Précision de lecture + Absence de support de transfert (réduction de coût)	- Dépendant de la méthode de marquage (non permanence, détérioration ou réduction de la surface exploitable...)
Données électroniques et étiquette à puce (étiquette RFID)				+ Acquisition des données à distance + Lecture simultanée de plusieurs étiquettes	- Coût élevé

3.1.7 Enjeux et importance de la traçabilité

Si un produit connaît un problème de qualité, le fabricant de ce produit doit prendre des mesures efficaces et rapides. Une réponse lente ou inefficace de ce fabricant crée un climat de méfiance chez les consommateurs ou les partenaires commerciaux, susceptible de compromettre l'existence de la société. De plus, une législation protégeant les consommateurs étant mise en œuvre, le nombre de sociétés qui doivent rapidement rappeler leurs produits en raison de problèmes augmente chaque année.

Depuis longtemps vigoureusement recommandée dans l'industrie automobile, la traçabilité est largement utilisée à des fins de prévention des rappels, de minimisation des dommages, d'identification/résolution des problèmes et de gestion de la qualité. Il est cependant difficile de suivre, de la fabrication à la mise au rebut, les données de plusieurs dizaines de milliers de composants, tout en respectant des lois et réglementations en constante évolution. Pourtant la traçabilité est un enjeu majeur dans nos sociétés mondialisées, régies par une féroce concurrence des prix et des délais. La nécessité de mettre en œuvre un système de gestion de l'historique global, qui couvrirait la traçabilité interne et externe, n'a jamais été si urgente.

3.2 Principes de mise en oeuvre

Cette section décrit étape par étape les procédures de mise en oeuvre de la traçabilité, en se basant sur un exemple type de flux de production dans l'industrie manufacturière.

Afin d'assurer la traçabilité, la première étape consiste à identifier les pièces et produits, puis à les gérer en y rattachant des objets et informations. Les concepts clés sont ici l'identification et la gestion des liens. On présentera tout d'abord la méthode de formalisation du flux des objets, première étape pour assurer la traçabilité, puis on fera le point sur l'identification.

3.2.1 Formalisation du flux des objets

La traçabilité peut être appréhendée sous divers angles comme on l'a vu dans la section précédente, par exemple *traçabilité interne* et *traçabilité de la chaîne*. L'exemple de traçabilité interne que nous allons décrire ici sera à une échelle limitée telle que celle de l'usine. Lors de la mise en oeuvre de la traçabilité, il est nécessaire de clarifier le flux des procédés de production en vérifiant les points suivants :

1. Les matières premières et pièces sont-elles fournies par une seule ou plusieurs sociétés ?
2. Qui vérifie la correspondance entre le bon de livraison, les matériaux et pièces reçus et de quelle manière ?
3. Quel est le mode de gestion du stock (individuel ou par lots) ?
4. Comment les rapports d'usinage, d'assemblage et d'inspection sont-ils enregistrés et gérés ?
5. Quelles informations sont nécessaires et quelles informations doivent être collectées ?

3.2.2 Identification

L'identification consiste uniquement à reconnaître les matières premières, pièces et produits individuellement ou par lot. L'identification a pour fin de permettre la reconnaissance de chaque produit, même lorsque des produits similaires dans leur forme et leur matériau sont fabriqués en masse. L'identification peut être réalisée de deux manières : soit par l'attribution de symboles d'identification individuels à chaque pièce ou produit, soit par la définition d'un groupe spécifique en tant que lot et l'attribution d'un symbole d'identification au lot. Les éléments individuels ou groupes auxquels un symbole d'identification a été attribué sont appelés unités d'identification.

Identification individuelle Chaque pièce ou produit se voit attribué un numéro de série unique et est géré individuellement (Figure 3.8). En cas de rappel, les produits posant problème peuvent être localisés en toute fiabilité. Cette méthode est également appelée gestion par numéro de série car elle exploite des numéros de série qui sont des numéros uniques.

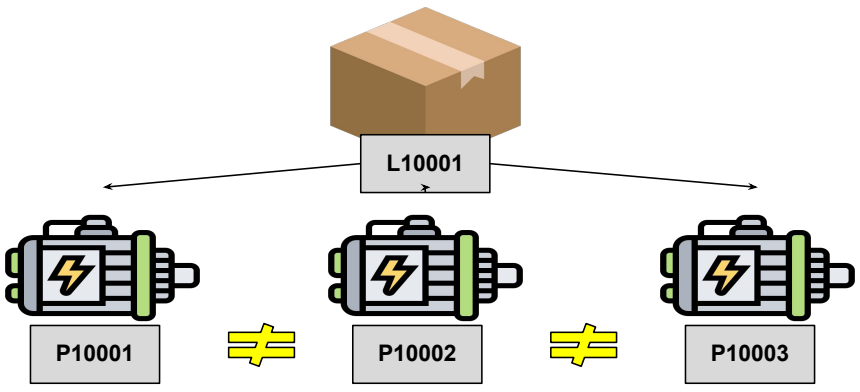


Figure 3.8 – Représentation graphique de l'identification individuelle

Identification par lot Les produits fabriqués dans des conditions identiques sont considérés comme appartenant à un même groupe appelé *lot* et à chaque lot est attribué un numéro d'identification. Cette méthode de gestion exige de vérifier que tous les produits soient de qualité uniforme. Tous les produits du lot portant le même numéro d'identification, il est impossible de les identifier individuellement (Figure 3.9).

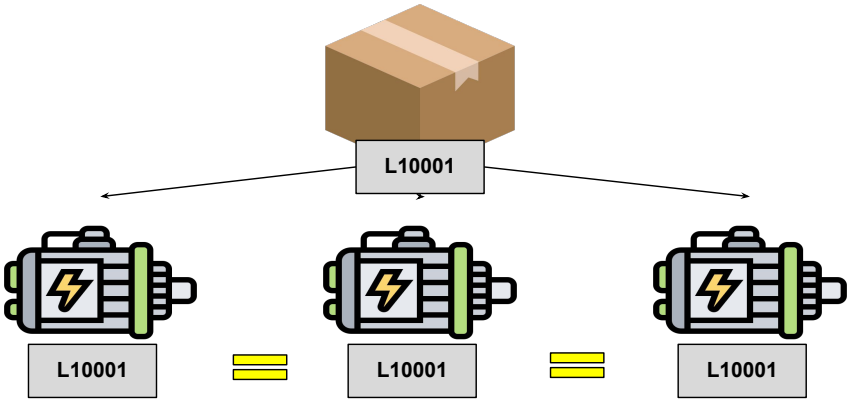


Figure 3.9 – Représentation graphique de l'identification par lot

La gestion par lot permet de mettre en oeuvre la traçabilité à moindre coût grâce à l'élargissement de l'échelle de l'identification. À l'inverse, pour une précision élevée, il est préférable de constituer des lots plus restreints ou de choisir la gestion individuelle.

En gestion par lots, la précision du suivi varie en fonction des critères de séparation des lots. La manière la plus simple de créer un lot est de regrouper des produits identiques fabriqués la même journée. Par exemple, si 1000 pièces sont fabriquées en une journée, regroupez-les en un lot unique et utilisez le même numéro de lot pour chacune d'elles. Les pièces ou produits d'un même lot doivent être fabriqués avec la même machine, le même jour et présenter la même qualité.

Qu'il s'agisse des réglages de la machine, du fabricant fournissant les vis ou de la personne chargée de la fabrication, tout changement des conditions en cours de production risque d'affecter la qualité du produit. Dans ce cas, il est important de modifier également le numéro de lot (réduction du lot). Plus un lot est réduit, plus l'étendue d'un potentiel rappel de produits défectueux est limitée. Cette méthode de gestion par division des lots est appelée gestion par ségrégation. Cependant, la tâche est alors plus ardue, en raison de la ségrégation et les coûts plus élevés. Il est donc primordial d'étudier le rapport coût-efficacité lors de la création de lots.

3.2.3 Gestion des liens / Association

La gestion des liens consiste à rattacher un objet à un autre, un objet à des informations ou des informations entre elles. Supposons, par exemple, qu'une usine fabrique un moteur au sein duquel des pièces de différentes tailles, telles que des vilebrequins et des pistons, sont assemblées. La gestion des liens désigne l'ensemble des règles régissant l'accumulation, l'organisation et la récupération des informations, établies pour permettre l'obtention de détails, tels que le lieu de fabrication du piston utilisé ou le procédé d'usinage de la bielle à partir du symbole d'identification. Également appelée association, elle est indispensable à la mise en oeuvre de la traçabilité en aval et en amont.

Assurer la traçabilité une étape en amont ou en aval

Dans cette partie, nous entrons dans les détails de la gestion des liens à travers un exemple de procédés de production.

Nous supposons que la traçabilité jusqu'aux fournisseurs de pièces (une étape en amont) et jusqu'aux clients (une étape en aval) est assurée. Afin d'assurer la traçabilité en amont, les étapes suivantes doivent être respectées :

— Vérification de la date, de l'expéditeur et de la quantité de chaque pièce reçue sur les bons de livraison des fournisseurs
— Attribution aux pièces de numéros d'identification liés à ces informations
— Enregistrement et stockage des informations au sein de l'usine

Grâce à cette méthode, lorsqu'une pièce est défectueuse, il suffit de remonter les informations enregistrées jusqu'au fournisseur.

Concernant le client et la traçabilité en aval, il est nécessaire de rattachez au numéro d'identification du produit à livrer, la date et l'heure d'expédition par votre société et les informations sur le produit. Il est ainsi possible de tracer la date d'expédition, l'expéditeur et la quantité de chaque produit expédié.

Pour assurer la traçabilité, il est important de clarifier le flux externe lié à l'usine. En reprenant l'exemple de l'usine fabriquant un moteur, on peut supposer que cette dernière achète des composants à plusieurs fournisseurs. Si l'un des composants est de mauvaise qualité, la société peut aisément identifier le fournisseur concerné à condition que la traçabilité une étape en amont ait été assurée. Si la traçabilité une étape en aval est également assurée, la société peut rappeler les moteurs fabriqués avec les composants d'apport du même lot en identifiant avec précision les destinataires afin de minimiser les pertes.

Ajout des informations de la traçabilité interne

Ensuite, il est essentiel de comprendre le parcours des matières premières, pièces et produits au sein de l'usine. Dans l'exemple issu de l'industrie automobile cité ci-dessus, la traçabilité interne est assurée en ajoutant et en enregistrant avant et après chaque procédé les informations pertinentes telles que le fabricant de chaque composant utilisé (poulie, courroie, durite, alternateur...) et les composants assemblés (Figure 3.10).

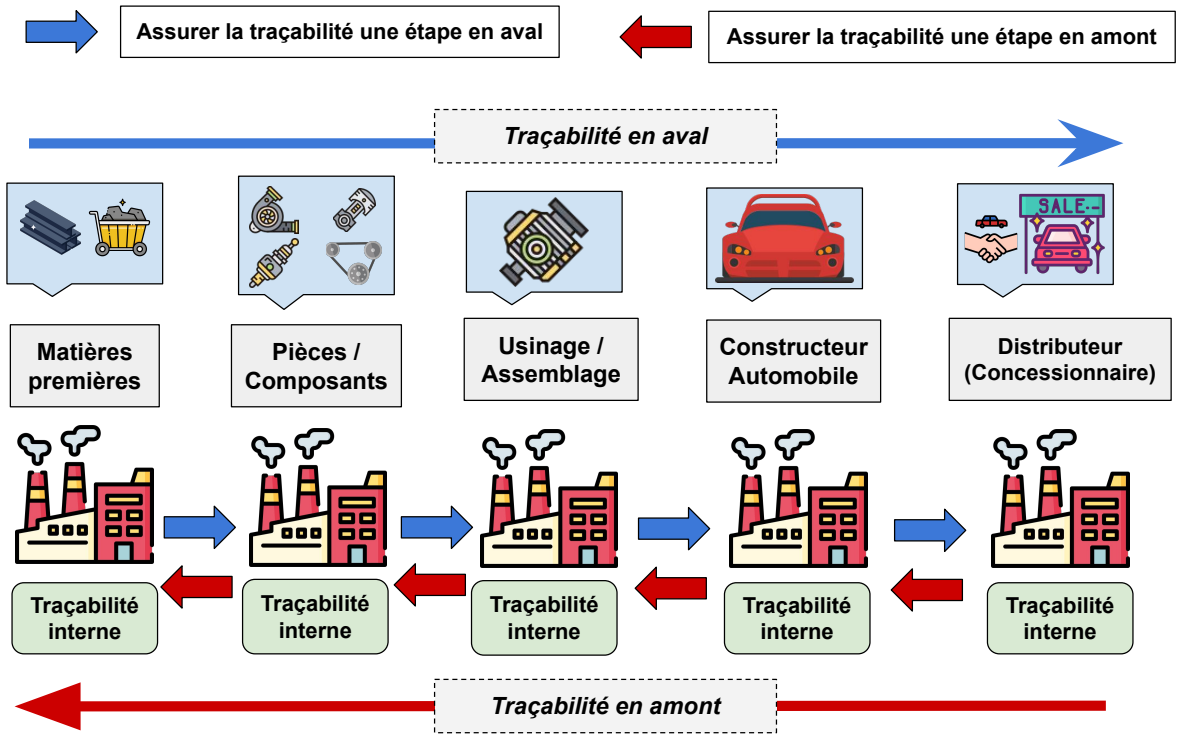


Figure 3.10 – Vue d'ensemble de la gestion des liens dans la traçabilité interne

Le principal objectif de la gestion des liens est de relier les procédés un à un tout en maintenant la traçabilité interne et la traçabilité jusqu'aux opérateurs une étape en amont et en aval. Si le lien est établi entre chaque procédé, par exemple entre le fournisseur de la courroie de ventilation et de la durite de radiateur, l'usine chargée de l'assemblage du moteur et le constructeur automobile installant le moteur sur ses véhicules, la traçabilité de la chaîne est naturellement assurée.

3.3 Règlementations, lois et normes

Cette section présente les réglementations, lois et règles de base en matière de traçabilité et de gestion de la qualité applicables à toute industrie, notamment automobile, composants électroniques, alimentaire et pharmaceutique.

3.3.1 Règlementation générale

Certaines normes et lois, telles que l'International Organization for Standardization (ISO) 9000/9001 et la loi japonaise sur la précision de la mesure, sont bien connues en gestion de la qualité et en contrôle de production. Elles sont tout aussi importantes dans le domaine de la traçabilité.

ISO 9000

Dans la norme ISO 9000, l'International Organization for Standardization (ISO) définit la traçabilité comme suit : "La capacité à retracer l'historique, l'utilisation ou l'emplacement d'un article ou d'une activité au moyen d'une identification enregistrée."

Cette définition implique deux prérequis essentiels :

- Spécifier clairement l'unité d'identification (individuelle ou lot) de l'objet concerné (matière première, pièce ou produit)
- Identifier l'objet concerné et enregistrer les informations nécessaires à la traçabilité en aval/amont

ISO 9001

Les normes relatives au système de management de la qualité de l'ISO sont regroupées sous le nom de série ISO 9000 ou famille ISO 9000 et la plus importante d'entre elles est l'ISO 9001. Un système de management de la qualité se réfère à « la partie d'un système de gestion destinée à la gestion et au contrôle de la qualité dans une organisation ».

L'ISO 9001 est une norme internationale basée sur l'étude de nombreux cas d'améliorations. Son objectif est de maximiser les bénéfices, en guidant les sociétés via un ensemble de règles cohérentes, applicables non seulement aux procédés de fabrication et aux produits mais à toute la chaîne, de l'achat au service après-vente, en passant par la fabrication et la distribution. Référence de choix, cette norme contribue à optimiser les procédés en éliminant des problèmes tels que les procédures inefficaces ou les erreurs répétées. La certification ISO 9001 procure aux entreprises une crédibilité sociale et la confiance de leurs clients.

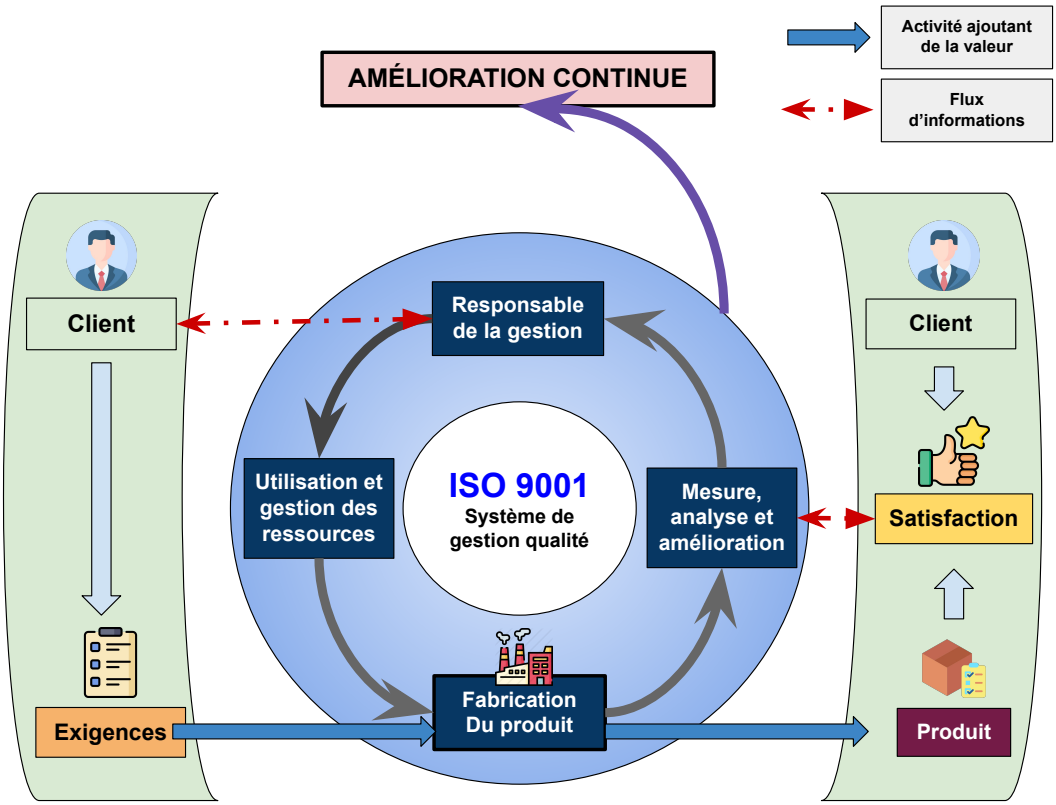


Figure 3.11 – Caractérisation de la norme ISO 9001

Identification et traçabilité L'ISO 9001 spécifie le concept de traçabilité dans deux sections distinctes : Traçabilité de la mesure et Identification et traçabilité. Cette page traitant de la traçabilité tout au long de la chaîne de commercialisation, nous nous concentrerons sur la section Identification et traçabilité.

En matière d'identification et de traçabilité, l'ISO 9001 stipule les exigences suivantes :

- Sélectionner les moyens d'identification appropriés (format et méthode de représentation) à chaque étape de la chaîne, de la production à la distribution
- Grâce aux moyens sélectionnés, identifier les sorties, telles que les produits finis, et les informations relatives à chaque procédé
- Enregistrer et stocker les informations de façon adaptée

Dans ces exigences, l'identification est un concept particulièrement important. Elle permet une reconnaissance commune et partagée à travers tous les procédés de production et tout au long de la chaîne de commercialisation. Prenons l'exemple d'une puce utilisée pour la fabrication d'une télévision. Le principe fondamental de l'identification est de garantir que cette puce soit reconnue de la même manière qu'elle soit seule ou intégrée à un produit et de permettre à chacun de connaître son parcours, c'est-à-dire son état, son historique d'inspection et les procédés qu'elle a subi.

À cette fin, le système de traçabilité identifie les objets au moyen d'un format de représentation, tel qu'un numéro de série ou de lot, et d'un support de transfert, tel qu'une étiquette papier ou une étiquette RFID.

Lorsque toutes les parties impliquées dans la chaîne de commercialisation, incluant les consommateurs, sont en mesure de connaître l'état du produit et que les liens entre objets et informations ont été établis, le suivi en aval/amont est aisé. La qualité et la sécurité du produit sont alors garanties.

Norme de sécurité relative aux systèmes de commande électroniques : CEI 61508 Les sites de production regorgent de systèmes de commande électroniques, à l'instar des automates programmables indispensables à l'automatisation industrielle. La Commission Électrotechnique Internationale (CEI) a établi la norme

CEI 61508 qui est une norme de sécurité fonctionnelle destinée aux systèmes électriques et électroniques. Elle regroupe les normes de sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables applicables à toute industrie.

L'idée principale est de garantir la sécurité et de prévenir les accidents et sinistres en ajoutant des équipements fonctionnels pour protéger la vie, la santé et l'environnement (région) de tout événement dangereux. La norme CEI 61508 constitue le fondement de toutes les normes de sécurité fonctionnelle en vigueur. L'ISO 26262, norme de sécurité fonctionnelle relative aux systèmes électriques/électroniques automobiles, découle également de la norme CEI 61508.

3.3.2 Règlementation dans l'industrie automobile

Cette partie présente les lois et réglementations relatives à la traçabilité que tout acteur de l'industrie automobile se doit de connaître, des constructeurs automobiles et de motos aux entreprises d'assemblage de moteurs et de transmissions et leurs fournisseurs de pistons, de vilebrequins et d'autres pièces, en passant par les fournisseurs de systèmes de navigation, de pneus et d'autres équipements périphériques.

Norme de management de la qualité pour les fournisseurs : IATF 16949

Les constructeurs qui produisent et commercialisent des automobiles ou des motos sont fournis en pièces par des sous-traitants (fournisseurs). Ces constructeurs sont tenus responsables de la supervision des systèmes de gestion de la qualité de leurs fournisseurs et lorsque des dizaines de fournisseurs sont impliqués, il devient essentiel de définir des critères d'évaluation unifiés.

La norme internationale IATF 16949 est là pour les guider. Elle spécifie les directives de gestion de la qualité que les constructeurs automobiles et de motos doivent imposer à leurs fournisseurs afin de satisfaire les exigences de qualité, de délai et autres. Il s'agit d'une spécification technique anciennement nommée ISO TS 16949 puis renommée en 2016 en tant que *IATF16949 :2016* par l'International Automotive Task Force (IATF) . Elle est notamment inspirée de normes européennes et américaines et a été basée sur l'ISO 9001.

Norme de sécurité des systèmes électroniques : ISO 26262

Les automobiles modernes sont équipées de nombreux dispositifs électriques/électroniques, incluant une unité de commande électronique (UCE), des capteurs et des actionneurs (moteurs). L'ISO 26262 est une norme internationale relative à la sécurité fonctionnelle des automobiles dérivée de la norme CEI 61508, applicable aux équipements et produits industriels en général.

L'ISO 26262 couvre la totalité du cycle de vie, de la définition des exigences à la mise au rebut, en passant par le développement, la production, la maintenance et l'utilisation. Par conséquent, elle exige une gestion de la sécurité fonctionnelle tout au long de la chaîne de commercialisation et est étroitement liée à la traçabilité. Face à la montée en puissance de l'électrique et de l'hybride, l'industrie automobile doit intégrer de plus en plus de dispositifs électriques/électroniques. De ce fait, les constructeurs automobiles sont en constante recherche de fournisseurs et fabricants de ces dispositifs et la certification ISO 26262 est en tête de liste des critères de sélection.

Obligation de rappel : « Title 49 USC Chapter 301 » et « TREAD Act »

Le « Title 49 USC Chapter 301 - MOTOR VEHICLE SAFETY » est un système de rappel lancé en septembre 1996 aux États-Unis. Lorsqu'une automobile ou un de ses équipements est supposé présenter un défaut de sécurité ou lorsqu'un nouveau véhicule n'est pas conforme aux normes de sécurité, le système de rappel des États-Unis exige des constructeurs qu'ils notifient ce défaut ou cette non-conformité aux autorités compétentes et aux utilisateurs. Il impose également l'obligation de rappeler ou de réparer gratuitement les modèles de véhicule concernés.

Règlementations liées aux rappels dans le monde Partout dans le monde, des lois et réglementations régissent les rappels et autres obligations (Tableau 3.3).

Tableau 3.3 – Vue d'ensemble des règlementations relatives aux rappels liées à l'industrie automobile dans le monde

Pays	*Règlementation*
Etats-Unis	« Title 49 USC Chapter 301 - MOTOR VEHICLE SAFETY » (sécurité des véhicules motorisées)
Japon	Loi sur les véhicules de transport routier
Union Européenne	Directive 2001/95/CE relative à la sécurité générale des produits
Allemagne	« Gerate und Produktsicherheitsgesetz » (loi sur la sécurité des produits et des équipements)
Royaume-Uni	« Code of Practice on Vehicle Safety Defects » (code des pratiques relatives aux défauts de sécurité des véhicules)
Australie	« Competition and Consumer Act 2010 » (loi sur les pratiques commerciales)

Un système de rappel plus strict : « TREAD Act » aux Etats-Unis

Le « Title 49 USC Chapter 301 - MOTOR VEHICLE SAFETY » (sécurité des véhicules motorisés) exige l'émission d'un rappel dans tout cas de non satisfaction des normes de sécurité, même lorsque la cause est inconnue. Outre ces exigences déjà très strictes imposées aux constructeurs automobiles et fabricants de pièces, les États-Unis ont renforcé leur système de rappel à la suite du problème de décollement de la bande de roulement sur des véhicules Ford Explorer équipés de pneus Bridgestone/Firestone en 2000, qui a engendré de nombreux accidents mortels et blessures.

De cet événement est né le « Transportation Recall Enhancement, Accountability, and Documentation Act » (Loi sur la documentation, la responsabilité et le renforcement des rappels dans le domaine des transports) ou *TREAD Act*, entré en vigueur en 2000. Le *TREAD Act*, qui rend obligatoire le signalement sans faille de toute information relative à la sécurité dans les moindres détails, a soudainement mis la traçabilité sous le feu des projecteurs.

3.3.3 Règlementation dans les industrie alimentaires et pharmaceutiques

Outre les réglementations internationales telles que l'ISO 9000/9001 relative au management de la qualité dans le monde industriel de façon générale, les industries alimentaires et pharmaceutiques possèdent également dans chaque pays une législation spécifique en matière de traçabilité.

Dans cette partie, nous présenterons les lois et réglementations applicables aux industries alimentaire et pharmaceutique dans divers pays, incluant les « Principes généraux de la législation alimentaire » dans l'UE et le « Bioterrorism Act » (Loi sur le bioterrorisme) aux États-Unis.

HACCP

L'« Hazard Analysis Critical Control Points (système d'analyse des risques et points critiques pour leur maîtrise) (HACCP) » est une méthode de contrôle d'hygiène alimentaire, basée sur le rapport publié en 1993 par la Commission du Codex Alimentarius, une organisation commune à l'Organisation des Nations Unies pour l'alimentation et l'agriculture (FAO) et l'Organisation Mondiale de la Santé (OMS), appliquée partout dans le monde.

L'HACCP désigne une méthode de contrôle de la sécurité qui consiste à analyser scientifiquement les risques potentiels relatifs à la salubrité d'un aliment, tels que le risque d'empoisonnement, d'un bout à l'autre de la chaîne alimentaire, de la réception des matières premières à l'expédition, en passant par la fabrication et le traitement, puis à identifier les points de contrôle critiques pour réduire ou éliminer ces risques. Au Japon, l'HACCP a été instituée à travers l'amendement *Gestion intégrée d'hygiène au processus de la fabrication*, publié en 1995.

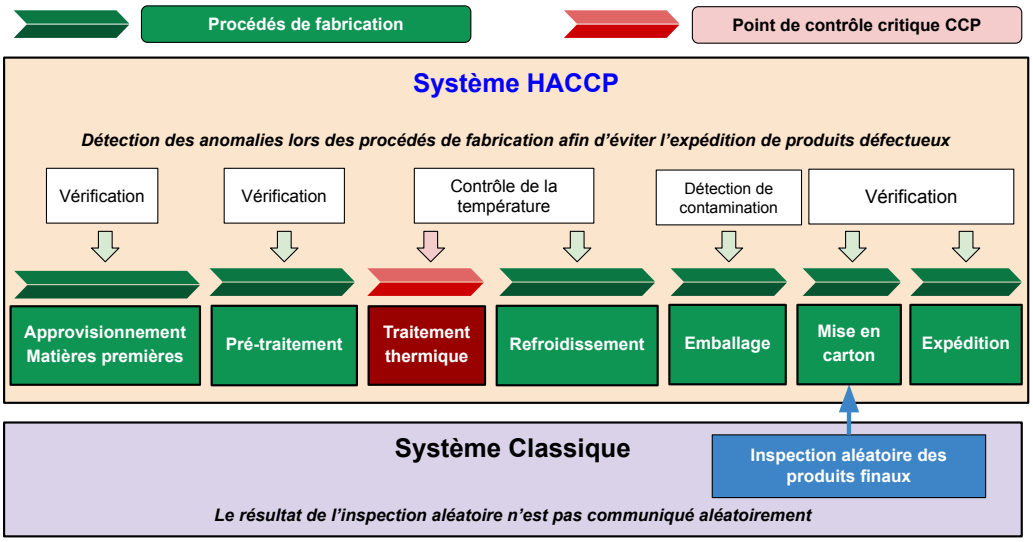


Figure 3.12 – Caractérisation de l'HACCP

Dans le cadre d'un contrôle d'hygiène classique, les normes applicables aux installations/équipements et aux méthodes de traitement des aliments sont spécifiées puis les produits finaux sont contrôlés. Le système HACCP préconise d'analyser les risques à chaque procédé de fabrication, de réaliser des vérifications intensives lors des procédés importants et d'effectuer un contrôle/une inspection sur la base d'une norme spécifiquement dédiée à chaque procédé. Les contre-mesures prises tout au long de la chaîne permettent de prévenir les empoisonnements alimentaires et contaminations par des particules étrangères avant que ne se produise un problème, renforçant ainsi la sécurité.

Principes de traçabilité alimentaire dans l'Union Européenne

Avant le problème de l'ESB (maladie de la vache folle) survenu en Europe en 1996, l'Union Européenne ne disposait pas de principes unifiés concernant la législation alimentaire. Il n'existait alors qu'un ensemble de lois établies individuellement pour chaque cas. Cet état de fait a été mis en lumière par la crise de l'ESB, qui a conduit à l'établissement d'une réglementation unifiée appelée *Principes généraux de la législation alimentaire* et promulguée en 2002. Cette dernière pose les bases des principes généraux et prescriptions générales de la législation alimentaire, conduisant ainsi à la création de l'Autorité européenne de sécurité des aliments par le règlement N°178/2002.

Spécifications dans les registres de traçabilité En matière de traçabilité, l'article 18 des Principes généraux de la législation alimentaire stipule le principe suivant : la traçabilité des denrées alimentaires, des aliments pour animaux, des animaux producteurs de denrées alimentaires et de toute autre substance destinée à être incorporée ou susceptible d'être incorporée dans des denrées alimentaires ou des aliments pour animaux est établie à toutes les étapes de la production, de la transformation et de la distribution.

Il est spécifié que les acteurs de la chaîne alimentaire doivent tenir les informations suivantes à disposition des autorités compétentes, à la demande de celles-ci :

1. Nom

2. Lieu
3. Type de produit fourni
4. Date de transaction et de livraison

Les principes recommandent également de collecter et conserver des informations plus détaillées sur le produit, telles que la quantité.

« Bioterrorism Act » (loi traçabilité alimentaire - États-Unis)

Aux États-Unis, la traçabilité alimentaire est régie par le Bioterrorism Act (The Public Health Security and Bioterrorism Preparedness and Response Act of 2002 (Loi américaine de 2002 sur la sûreté de la santé publique et sur la prévention du bioterrorisme)).

Le Bioterrorism Act impose la mise en œuvre obligatoire de mesures telles que l'enregistrement des installations pour produits agro-alimentaires, la notification préalable des importations, l'établissement et la tenue de registres et la détention administrative. Les mesures de la loi relatives à la traçabilité sont liées à la tenue de registres.

Il est exigé des opérateurs qui fabriquent, transforment, conditionnent, transportent, distribuent, reçoivent, stockent ou importent des produits alimentaires aux États-Unis qu'ils établissent et tiennent des registres. La gestion des liens entre les procédés une étape en aval et une étape en amont, telle que décrite dans la section « Mettre en place un système d'identification », est requise par la loi. De plus, si un produit est soupçonné de présenter une menace pour la santé humaine ou animale, les opérateurs concernés devront mettre tous les registres et autres informations à la disposition de la Food and Drug Administration (FDA - administration américaine chargée des aliments et des médicaments).

Durée de conservation des registres Le Bioterrorism Act définit la durée de conservation en fonction de la nature périssable du produit alimentaire :

Tableau 3.4 – Définition des durées de conservation des registres de traçabilité par le Bioterrorism Act

Catégorie	*Non-Transporteurs*	*Transporteurs*
Produits présentant un risque élevé de détérioration dans une période de moins de 60 jours	6 mois	6 mois
Produits présentant un risque élevé de détérioration dans une période de 60 jours à 6 mois	1 an	1 an
Produits présentant un risque élevé de détérioration après 6 mois	2 ans	1 an
Produits alimentaires pour animaux y compris les animaux domestiques	1 an	1 an

Traçabilité de l'industrie pharmaceutique en Chine : « Drug Electronic Administration Code »

Le code de gestion des médicaments est un système administré par la China Food and Drug Administration (CFDA), autorité chargée de garantir la sécurité des aliments et des médicaments, pleinement en vigueur depuis 2016. Les opérateurs souhaitant commercialiser des médicaments en Chine doivent obtenir au préalable une autorisation et un code de gestion des médicaments auprès de la CFDA, et intégrer ces informations sous la forme d'un code-barres.

Le code de gestion des médicaments a été introduit en 2012 à des fins de prévention de l'utilisation incorrecte de médicaments, de mise en oeuvre de la traçabilité et d'amélioration de la distribution. Bien que son application soit jusqu'ici limitée, toutes les sociétés, incluant les opérateurs étrangers, sont tenues de mettre en pratique le système depuis 2016. Il est obligatoire d'inscrire le code-barres sur une étiquette ou directement sur l'extérieur de l'emballage de chaque produit et le cas échéant de chaque dose individuelle.

Code de gestion des médicaments Les codes sont composés de 20 caractères numériques, intégrés sous la forme de codes-barres Code128C dont voici un exemple :

$$8 - 123456 - 123456789 - 1234$$

1. Nombre fixe, 1 caractère
2. Code du produit, 6 caractères, attribué par la CFDA
3. Numéro de série assigné à chaque produit
4. Code de vérification, 4 caractères

Synthèse des normes de traçabilité dans les différentes industries

Tableau 3.5 – Synthèse des normes et règlementations en terme de traçabilité

Secteur industriel	*Norme*	*Zone d'application*	*Date création*	*Description*
Général	ISO 9000	**Monde**	1987 (révisée en 1994, 2000, 2008 et 2015)	Norme donnant une définition générale de la traçabilité
	ISO 9001	**Monde**	1987 (révisée en 1994, 2000, 2008 et 2015)	Norme majeure fournissant un ensemble de règles visant à améliorer les procédés tels que la traçabilité
	CEI 61508	**Monde**	2002	Norme de sécurité fonctionnelle destinée aux systèmes électriques et électroniques
Automobile	IATF 16949	**Monde**	2016	Norme guidant les constructeurs automobiles dans les exigences à adresser à leurs fournisseurs en terme de qualité
	ISO 26262	**Monde**	2011 (révisée en 2018)	Norme dérivée de la CEI 61508 pour la sécurité des systèmes électroniques dans les équipements automobiles
	49 US Code Chapter 301 - MOTOR VEHICLE SAFETY	**Etats-Unis**	1994	Norme visant à règlementer le rappel de véhicules lorsqu'une non-conformité est avérée
	TREAD ACT	**Etats-Unis**	2000	Loi rendant obligatoire le signalement de toute information relative à la sécurité des véhicules
	Loi sur les véhicules de transport routier	**Japon**	1960	
	Directive 2001/95/CE relative à la sécurité générale des produits	**Union Européenne**	2001	Directive permettant de protéger les consommateurs de la mise sur le marché européen de produits non-sûrs en définissant notamment ce qu'est un produit sûr.
	Gerate und Produktsicherheitsgesetz	**Allemagne**	2004	Loi sur la sécurité des produits et des équipements
	Code of Practice on Vehicle Safety Defects	**Royaume-Uni**	2019	Code des pratiques relatives aux défauts de sécurité des véhicules

Secteur industriel	*Norme*	*Zone d'application*	*Date création*	*Description*
	Competition and Consumer Act 2010	 **Australie**	2011	Loi sur les pratiques commerciales
Alimentaire & Pharmaceutique	HACCP	 **Monde**	1993	Méthode de contrôle d'hygiène visant à identifier les risques potentiels des procédés de fabrication ainsi que les points de contrôle critique d'un bout à l'autre de la chaîne alimentaire
	Principes de la législation alimentaire	 **Union Européenne**	2002	Règlementation unifiée des principes liés à la législation alimentaire dans l'Union Européenne.
	Bioterrorism Act	 **États-Unis**	2002	Loi régissant les principes de la traçabilité alimentaire aux États-Unis tels que les durées de conservation des registres, l'enregistrement des installations...
	Code de gestion des médicaments (DEA Code)	 **Chine**	2012	Méthode de traçabilité des médicaments en Chine permettant d'identifier le produit (voire chaque dose individuelle) ainsi que son numéro d'autorisation de mise sur le marché

3.4 Etat de l'art sur la traçabilité

Le concept de traçabilité existe depuis le 13e siècle et trouve son origine dans l'industrie agro-alimentaire [60]. À ce jour, la traçabilité se trouve principalement dans cette industrie, principalement pour la prévention des épidémies [61]. Dans la législation nationale et internationale, des exigences de traçabilité alimentaire sont en place pour tenter d'atténuer ces problèmes [62].

La technologie trouvée dans les systèmes de traçabilité a été utilisée dans la lutte contre la contrefaçon de produits, principalement pour les médicaments, qui est devenue un problème majeur dans l'industrie pharmaceutique . Pour y parvenir, des microtagants inimitables peuvent être incorporés dans le médicament. Cela peut être fait en étiquetant le récipient, la capsule de médicament ou les particules à l'intérieur du médicament lui-même. Dans l'industrie médicale, des techniques similaires sont utilisées dans le suivi des dispositifs et instruments médicaux.

Outre le médical, les industries manufacturières aérospatiales ont également vu la nécessité et les avantages de la mise en œuvre de la traçabilité de leurs produits et procédés. Airbus trace électroniquement les pièces à l'aide d'étiquettes RFID depuis 2009 afin de suivre certaines pièces tout au long de leur cycle de vie. La traçabilité a également été utilisée dans le traçage des opérations de machines telles que la fabrication CNC à l'aide du langage de commande de machine-outil STEP-NC. En intégrant un système de traçabilité à ces machines, davantage de données deviennent disponibles sur le produit et ses processus de production.

3.4.1 Terminologie et définitions

Il existe de nombreuses définitions relatives à la traçabilité, dont beaucoup se retrouvent dans les normes et réglementations de l'industrie agroalimentaire. Une définition universelle moderne pour la traçabilité a été

proposée par [63] comme étant « la possibilité d'accéder à tout ou partie des informations se rapportant à ce qui est considéré, tout au long de son cycle de vie, au moyen d'informations d'identification enregistrées ».

Une revue de la terminologie, des techniques et technologies liées à l'implémentation d'un système de traçabilité est notamment proposée par [64]. Bien que les définitions et termes utilisés dans la littérature puissent différer, certains concepts reviennent systématiquement tels que la notion d' Unité de Ressource Traçable (TRU) [63] qui correspond au nom de l'élément que l'on cherche à tracer.

La traçabilité peut être qualifiée de passive ou active. Un système de traçabilité passif cherche uniquement à fournir une visibilité aux données tandis qu'un système actif cherche à optimiser et mieux contrôler les processus au cours de la chaîne d'approvisionnement.

Grâce à la notion d'unité de ressource traçable, les définitions liées à la traçabilité en amont et en aval peuvent êtra précisées : la traçabilité en amont fait référence au fait de suivre le TRU au cours du temps (de la création à la destruction) tandis que la traçabilité en aval permet de remonter à l'origine du TRU. En anglais, les notions d'amont et d'aval sont distinguées respectivement par les termes « tracking » et « tracing » [65].

La traçabilité peut être mise en œuvre à différents niveaux d'exhaustivité. La norme IPC-1782 pour l'industrie de l'électronique décompose la traçabilité en quatre niveaux : basique, standard, avancé et complet [66].

L'entreprise privée GS1, spécialiste des standards industriels définit un « niveau de traçabilité » comme étant la combinaison entre la précision de l'identification du TRU et le niveau dans la hiérarchie logistique [67]. Dans cette définition, le plus haut niveau de traçabilité correspond à une identification unique et individuelle des TRU. Par exemple, un fournisseur de jouets pourrait adopter une traçabilité bas niveau en groupant les produits par numéro d'expédition. A l'inverse, un fournisseur de matériel médical utilisera plutôt une traçabilité à haut niveau pour tracer les produits individuellement en fonction de leur utilisation mais aussi pour assurer le recyclage ou l'élimination appropriée à la fin de vie de l'instrument.

3.4.2 Données de traçabilité

Une description des trois principaux composants d'un système de traçabilité est proposée par [62] : identifier les Unité de Ressource Traçable (TRU), documenter leurs transformations, leurs connexions entre elles et enregistrer leurs attributs. Une Unité de Ressource Traçable (TRU) peut être une unité commerciale (caisse, sac, carton...), ou une unité logistique (palette) ou une unité de production (IoT, lot).

De multiples aspects des données de traçabilité sont mis en évidence dans [68], tels que leur nature hétérogène ou les défis liés à l'interopérabilité et à l'intégration. Les aspects réglementaires et d'exigences font également partie du sujet car ils pilotent en fait les politiques de traçabilité. Tous ces aspects sont pris en compte dans [68] dans lequel les auteurs ont proposé une description de modélisation basée sur une ontologie d'un système de traçabilité intelligent. Afin d'évaluer le modèle suggéré, ils ont proposé une application basée sur le cloud dont la description se concentre sur la précision et les avantages du système. Un autre travail remarquable [69] introduit le concept d'Objets de fabrication intelligents (SMO) dans une plate-forme de traçabilité afin d'obtenir une production en temps réel au sein d'une usine intelligente. Une combinaison de technologies telles que l'IoT[3], l'identification par radiofréquence (RFID) [70] et le scanner laser permet de convertir n'importe quelle ressource en SMO. De cette manière, les opérations et le comportement de production peuvent être surveillés en temps réel, permettant ainsi une traçabilité détaillée et à jour. Un examen des avantages de la traçabilité concernant la résilience de la chaîne d'approvisionnement est proposé dans [71] tandis qu'une analyse globale de la cybersécurité de la traçabilité dans la chaîne d'approvisionnement est rapportée dans [72].

La plupart de ces travaux envisagent la confidentialité en matière de divulgation des données ce qui conduit à des solutions qui se concentrent principalement sur leur protection, les retirant ainsi des données de traçabilité échangées entre les partenaires. De notre côté, nous considérons que les données confidentielles doivent être incluses, d'une manière appropriée qui reste à définir dans les données de traçabilité échangées lorsqu'elles peuvent contribuer à accélérer les investigations rétrospectives.

3.5 Etat de l'art sur la traçabilité orientée produit

Un produit est un ensemble d'attributs, appelés caractéristiques, composés de manière identifiable [73]. Ces caractéristiques déterminent la valeur de ce qui est produit. La traçabilité des produits est définie par la norme ISO 8402 comme "la possibilité d'accéder à l'historique, à l'utilisation ou à la localisation d'un article ou d'une activité, ou d'articles ou d'activités similaires au moyen d'une identification enregistrée" [74]. En fabrication, cela revient à enregistrer les mouvements et modifications spécifiques d'un produit tout au long de son cycle de vie dans la chaîne d'approvisionnement. Une ontologie orientée produit développée dans [75] propose une méthodologie qui formalise les concepts techniques et les données embarquées dans le produit lui-même, favorisant ainsi l'interopérabilité avec d'autres applications. La conceptualisation du modèle d'information est basée sur des normes telles que ISO 10303 et IEC 62264. Le domaine de la traçabilité des produits s'avère être hétérogène et nécessite de ce fait de multiples technologies (QR Code, code-barres, RFID) qui rendent donc son intégration dans un système déjà existant très complexe. Dans ce contexte, une approche systématique de la traçabilité des produits a été conçue par [74], avec un aperçu de l'industrie de la céramique. Cette approche combine plusieurs technologies à différentes étapes du cycle de vie d'un produit et prend en charge l'intégration de la traçabilité aux différents niveaux requis par l'Industrie 4.0.

Une approche méthodologique est proposée par [73] afin de caractériser les produits, les exigences et les acteurs impliqués dans un système de traçabilité. Afin de tracer les pièces de rechange agrégées dans un produit manufacturé, [76] présente une approche basée sur la blockchain qui répond à certains problèmes traditionnels rencontrés par une gestion de la chaîne d'approvisionnement, tels que la transparence et le suivi actif des opérations qui affectent la conformité et sécurité des composants des pièces de rechange. De par sa caractéristique immuable, la technologie blockchain permet non seulement d'identifier les pièces défectueuses ou contrefaites, mais également de déterminer l'origine des actifs dans le cadre de la traçabilité. Le système proposé dans [76] comprend un stockage décentralisé de fichiers basé sur le système de fichiers interplanétaire (IPFS) pour les données des pièces de rechange, les algorithmes de contrats intelligents, la sécurité et l'analyse des coûts. Une évaluation est proposée afin de démontrer la fiabilité de la solution de suivi de propriété des pièces détachées. Cependant, cette solution se concentre uniquement sur les interactions au sein d'une même entreprise. Par conséquent, les défis liés au maintien de la confidentialité des données critiques ainsi qu'à la transparence dans un contexte d'interopérabilité avec des partenaires externes n'ont pas été abordés.

Un autre exemple de traçabilité des produits basée sur la blockchain est proposé par [77] pour la fabrication additive. L'objectif annoncé était d'assurer une traçabilité sécurisée et fiable, ainsi que l'accessibilité et l'immuabilité. L'architecture proposée utilise les contrats intelligents Ethereum et IPFS comme solution de stockage de fichiers distribués. Les auteurs proposent une évaluation axée sur les exigences de sécurité telles que l'intégrité, la responsabilité, la non-répudiation et l'autorisation. Cependant, les questions liées à la gestion des données critiques, à la transparence et à la confiance entre les parties prenantes ne sont pas évoquées dans l'étude. Pourtant, le coût introduit par l'utilisation d'IPFS comme solution de stockage complémentaire à la blockchain n'est pas analysé. À propos des évaluations, une enquête sur les solutions blockchain pour la fabrication durable et l'autonomisation de la gestion du cycle de vie des produits est proposée dans [78]. Diverses mesures liées à la confiance dans l'utilisation des solutions de blockchain dans la fabrication sont introduites, concernant la transparence, la décision décentralisée, la réputation et la relation client. Du point de vue de la fabrication, la blockchain pourrait être utilisée comme catalyseur pour piloter des systèmes déjà existants pour les ateliers, tels que la planification des ressources d'entreprise (ERP) et le système d'exécution de la fabrication (MES).

Du point de vue de la gestion des produits, la blockchain pourrait offrir une base de données unifiée pour partager les informations sur les produits. Le fait que de nouvelles propositions soient encore présentées dans divers secteurs manufacturiers [79], y compris le textile [80], souligne l'importance des solutions pratiques de blockchain pour la traçabilité dans l'Industrie 4.0. En effet, d'autres défis techniques tels que la satisfaction des besoins de protection de la vie privée, la consommation d'énergie et les problèmes d'évolutivité du stockage sont parmi les principaux problèmes qui empêchent les fabricants d'élever leurs solutions de blockchain au-delà des preuves de concept [81].

3.5.1 Innovations technologiques dans les systèmes de traçabilité

Assurer la traçabilité est aujourd'hui une obligation. Outre l'identification au moyen d'étiquettes et de marquages, toute entreprise doit être en mesure de gérer les liens entre informations et objets.

L'essor de la traçabilité est très certainement dû au perfectionnement des technologies informatiques, qui permettent aujourd'hui le traitement complet des informations au moyen de systèmes de marquage et de lecture plus précis que jamais.

Même si les systèmes de traçabilité utilisés dans les industries diffèrent, un élément précis est toujours présent au coeur de ces derniers et il s'agit de la méthode d'identification du produit [64] [82]. Cette identification se fait en général en liant au TRU un identifiant unique permettant à la fois de différencier les parties physiques du produit mais aussi de lier les dites parties avec leurs données respectives.

Une des techniques les plus employées en termes d'identification est le marquage produit qui se décline en deux formes : marquage direct et indirect. [64]

Marquage indirect

Le marquage indirect fait référence à l'utilisation d'un support d'identification séparé, généralement un autocollant avec un chiffre, un alphanumérique ou un code barre 2D. Cette méthode n'est généralement utilisée que sur des pièces plus grandes.

Le défaut majeur lié aux étiquettes est qu'elles peuvent être sujets aux erreurs de lecture lorsqu'elles sont fixées sur des objets présentant une surface non places (tuyau par exemple) ou bien sur des matériaux spécifiques (métallique, brossée, moulée).

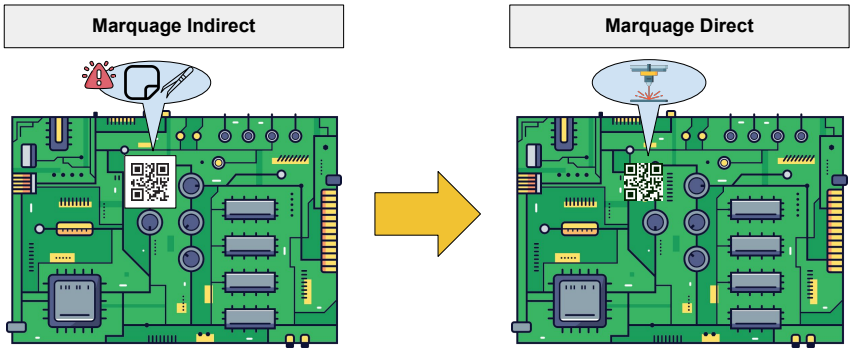


Figure 3.13 – Comparaison entre marquage produit indirect et direct

Marquage direct

Le marquage direct (DPM), qui permet d'inscrire les numéros de série, numéros de lot et codes 2D directement sur les pièces et produits, a également grandement contribué à l'évolution des systèmes de traçabilité. Sur un site de fabrication de cartes de circuit imprimé, les fausses puces sont devenues un problème majeur. Le marquage direct des pièces a été appliqué en tant que contre-mesure. La combinaison du marquage direct et des codes 2D permet d'inscrire des symboles d'identification même sur des composants trop petits pour y coller une étiquette (Figure 3.13).

Le marquage direct convient également aux produits soumis à de hautes températures et utilisés en extérieur durant de longues périodes, tels que les moteurs et transmissions. L'emploi d'étiquettes sur ces produits est fortement déconseillé dû au risque de décollement et de décoloration. Le marquage direct élimine tout problème d'échec de lecture.

Codes-barres et stockage d'informations

La traçabilité exige de rattacher aux produits une multitude d'informations, notamment numéros de pièce, quantité, numéro de série et volume d'expédition. Malheureusement, les codes-barres classiques dits linéaires (ou 1D) [83] ne peuvent contenir toutes ces informations (Figure 3.14).

Les codes 2D sont capables de stocker davantage d'informations sur une plus petite surface et sont, de ce fait, utilisés dans divers processus de gestion des pièces et produits, incluant la traçabilité [84]. Ces codes peuvent même être marqués sur de minuscules composants, tels que les cartes de circuit imprimé miniaturisées haute densité, garantissant ainsi une fiabilité de gestion individuelle accrue.

Les types de codes 2D les plus couramment utilisés sont le QR-Code, le datamatrix et le PDF417 dont la taille physique est bien plus grande que les deux précédents [85]. Le datamatrix est le plus petit code 2D de par sa taille de 10x10 modules. Le QR-code quant à lui mesure au minimum 21x21 modules.

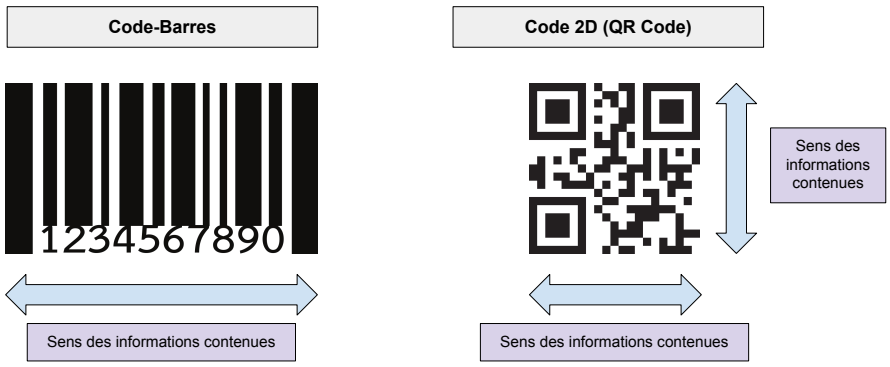


Figure 3.14 – Stockage d'informations entre code barres et code 2D

Ces codes sont également régis par des normes concernant la méthode d'encodage tels que la norme ECC200 qui demande à utiliser l'algorithme reed-Solomon. L'avantage de cet algorithme est qu'il permet de réduire le risque d'erreurs de lecture d'un code datamatrix à 1 chance sur 613 millions dans le meilleur des cas. Pour comparaison, un code barre standard a une probabiblité d'erreur de lecture de 1 chance sur 4.5 millions dans le meilleur des cas.

Des bonnes pratiques en termes d'implémentation des codes datamatrix ont été formulées par l'entreprise GS1 dans [86] en se basant sur la norme ECC200. Ce document est à ce jour utilisé par de nombreuses industries dans les secteurs industriels de l'automobile, de l'aéronautique et du médical.

Les différents types de code-barre sont appelés symbologies, une synthèse des différentes symbologies est disponible dans le Tableau 3.6.

Reconnaissance optique (OCR) et automatisation

L'OCR est une technologie de reconnaissance des caractères contenus dans une image. Le système de vision capture une image de la cible au moyen d'une caméra, reconnaît les caractères de l'image capturée et compare leur forme aux données enregistrées dans le dictionnaire (polices de caractères) pour les lire en tant que caractères. L'OCR est devenue une technologie standard de l'identification et de l'enregistrement des dates d'expiration ou numéros de pièce. Elle permet, en outre, d'éviter les erreurs inhérentes à la lecture et à la saisie manuelle [87] [88].

Son évolution permet aujourd'hui de lire en d'enregistrer avec précision des caractères, tels que des dates d'expiration et des numéros de pièce. Parfait exemple de la sensibilisation croissante à la sécurité, les dates d'expiration des aliments comportent à présent non seulement une date d'expiration mais également une date de production. Plus la quantité d'informations est grande, plus la charge de travail est lourde sur le site

Tableau 3.6 – Symbologies des codes-barre utilisés pour l'identification des produits dans la traçabilité

Symbologie	*Symbole*	*Type*	*Date création*	*Normes liées*	*Caractéristiques*	*Cas d'utilisation*
EAN / UPC		1D	1970 (EAN), 1976 (UPC)	ISO/IEC 15420 :2009	Stockage de 13 caractères (standard EAN-13 le plus répandu) ou 12 pour UPC-A, stockage de données numériques uniquement	Présent sur presque tous les produits de consommation dans le monde (UPC aux Etats-Unis et EAN pour les produits internationaux)
Code 128		1D	1981 par Computer Identics Corporation (US)	ISO/IEC 15417 :2007	L'une des symbologies linéaires les plus denses, encode le jeu de caractères ASCII complet de 128 caractères	Suivi d'expédition
Code 39		1D	1974	ISO/IEC 16388 :2007	Moins dense que le code 128, nécessitant ainsi plus de place sur l'étiquette	Identification, inventaire et suivi
Data Matrix		2D	1994 par Data Matrix.Inc	ISO/CEI 16022	Stockage jusqu'à 2335 caractères alphanumériques, idéal pour l'encodage de volumes importants de données sur un espace réduit	Produits pharmaceutiques, circuits imprimés, instruments chirurgicaux, numéros de série dans le manufacturing
Aztec		2D	1997 par AIM.Inc	ISO/IEC 24778 :2008	Stocke environ jusqu'à 3000 caractères, taille du symbole dynamique en fonction de la quantité de données	Tickets de transport, bracelets d'identification des patients, étiquettes pour les médicaments à usage unique, produits sanguins et les échantillons
QR Code (Quick Response Code)		2D	1994 par Denso.Inc	ISO/IEC 18004 :2015	Peut stocker entre 4000 et 7000 caractères	Applications mobiles (confirmation de commande, coupons) et encodage de volume de données importants
PDF417		2D	1989 (Symbol Technologies)	ISO 15438	Peut stocker entre 1850 et 2725 caractères	Permis de conduire américain, cartes d'identité, passeports, inventaires

de production. L'automatisation par OCR du procédé de lecture résout les problèmes de sécurité et de charge de travail.

L'OCR, les codes 2D (QR code, Datamatrix) et le marquage direct ont été adoptés par de nombreuses sociétés afin d'assurer la traçabilité. D'autres nouvelles technologies font leur entrée en scène, telles que les imprimantes à jet d'encre et l'identification radio, permettant de lire/écrire des informations en exploitant des étiquettes à puce et des ondes radio.

Internet des objets (IoT)

Dans le contexte de la chaîne logistique, l'identification par radiofréquence a été utilisée pour identifier et suivre les expéditions. Dans la chaîne du froid, des enregistreurs de données ont également été utilisés pour surveiller les conditions de transport des expéditions. Dans les deux cas, la visibilité des données était limitée et les utilisateurs ne pouvaient obtenir aucune donnée sur les expéditions lorsque l'opération de transport était en cours. Il existe donc un réel besoin d'utiliser de nouvelles solutions connectées pour améliorer les

mécanismes de collecte des données de traçabilité. L'intégration de l'IoT dans la chaîne logistique améliore la visibilité de l'acteur, la le suivi et la traçabilité des envois transportés tout au long de la chaîne logistique. Cette intégration vise le fait d'améliorer les capacités de détection des évènements mais aussi à « créer du sens » au niveau des données récoltées [89] tout cela dans le but d'améliorer les prises de décision. De nombreux travaux récents proposent d'utiliser cette technologie notamment pour l'amélioration de la traçabilité [90] [91] [92].

L'évolution des technologies informatiques favorise la coordination entre les systèmes de traçabilité, les systèmes industriels de production ainsi que les Progiciel de Gestion Intégré (PGI), ne profitant pas seulement à la production mais à chaque activité de la société. De plus, les informations peuvent être partagées par réseau entre les filiales et usines, nationales comme situées à l'étranger. À l'avenir, il est certain que la traçabilité deviendra l'un des vecteurs clés de la mondialisation.

La portée de la traçabilité est appelée à croître en raison de la diversification des méthodes d'identification et du perfectionnement des technologies d'enregistrement et de transfert des informations.

3.6 Présentation de l'approche proposée

3.6.1 Vision du produit

Le concept de traçabilité intelligente assure la surveillance du produit parallèlement à son traçage et son suivi afin d'améliorer la qualité et la sécurité du produit [68]. L'importance de la traçabilité des produits est soulignée dans la norme de gestion de la qualité ISO 9001 :2015 [74]. Différentes technologies et différents types d'informations sont impliqués dans les différentes phases du cycle de vie d'un produit, à savoir le développement, la production, l'utilisation et l'élimination. Dans ce travail, notre approche se concentre sur la traçabilité d'un produit fabriqué dans une chaîne d'approvisionnement, qui peut inclure certains composants fournis par des entités externes telles que des fournisseurs. Une approche similaire a été adoptée dans [76] concernant l'intégration des différentes pièces de rechange qui composent un moteur de voiture.

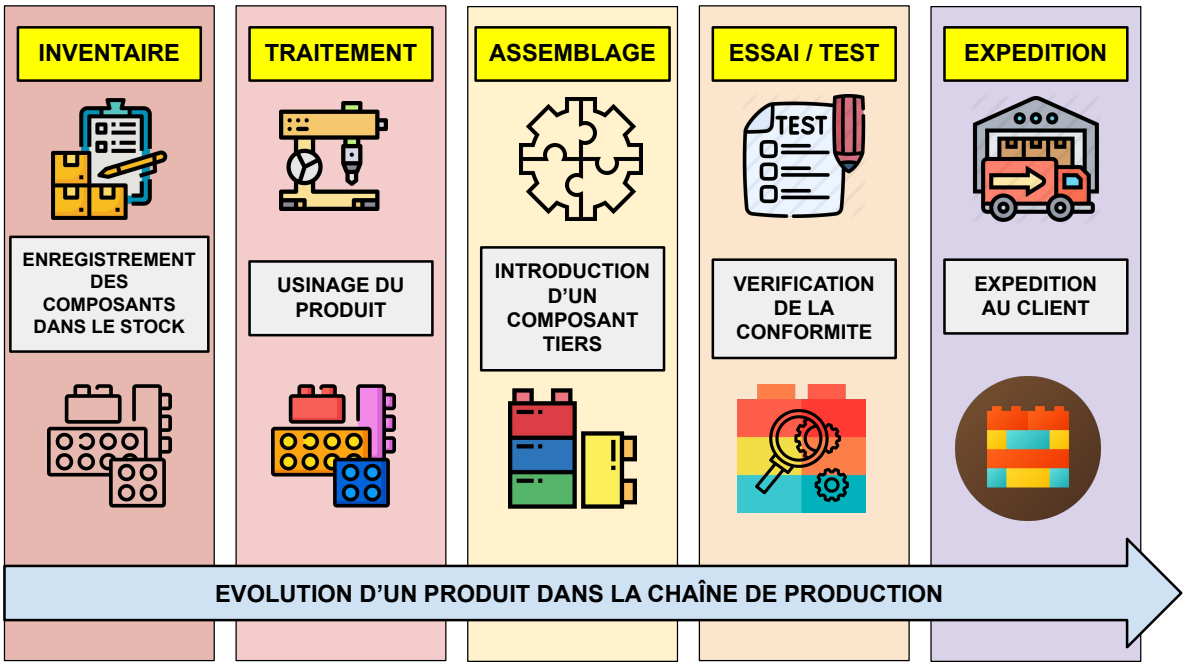


Figure 3.15 – Evolution d'un produit dans la chaîne de production

Notre approche se concentre plutôt sur la traçabilité du produit tout au long du processus de fabrication, et l'intégration de composants de rechange n'est qu'une des étapes. Figure 3.15 illustre une évolution type d'un produit manufacturé dans la supply chain. Cette évolution considère cinq étapes principales, détaillées ci-après :

Inventaire Moment où différents composants du produit sont enregistrés dans le stock. A cette étape, les données de traçabilité peuvent être les références des composants, des numéros de série ou toute autre information qui aiderait à tracer les produits ou composants.

Process Etape lorsque le produit entre sur une ligne de production où il subira des opérations de transformation à travers différentes sous-étapes. A ce niveau, les paramètres clés liés au process ou à la machine sont des données de traçabilité et leur niveau de confidentialité peut s'avérer critique.

Assemblage Il s'agit du moment où un composant fabriqué par un tiers est intégré dans le produit. Les caractéristiques techniques liées au composant et la façon dont il a été utilisé lors du processus d'assemblage constitue les données de traçabilité à ce niveau. Elles peuvent s'avérer critiques pour un audit post-assemblage.

Test Moment où la conformité du produit est vérifiée. Les données de traçabilité peuvent être n'importe quelle information résultant des tests (contexte, mesures, etc.), qui pourrait aider à établir ou non la conformité du produit.

Expédition Moment où le produit est prêt à être expédiée au client. Les exemples de données de traçabilité sont : numéro de lot, d'expédition et numéro de série du produit afin de pouvoir continuer à le suivre après qu'il ait quitté l'usine.

3.6.2 Caractérisation des données de traçabilité

D'après la norme ISO 9001, la traçabilité vise deux objectifs : mesurer et identifier. Concernant l'identification, il est demandé d'utiliser des moyens adaptés pour permettre l'identification des sorties telles que le produit fini ou les informations en fin de chaque étape de production. Dans ce but, un système de traçabilité identifie les objets en utilisant un format de représentation (numéros de série, numéros de lot) ou bien des supports de transfert (étiquettes avec code barres, RF tags)

De notre côté, nous mettons l'accent également sur les aspects décrits ci-dessous :

Type des données Les données de traçabilité étant hétérogènes, elles peuvent être de type divers. Dans notre approche, nous les séparons en deux catégories :

- les données simples brutes (RAW) qui peuvent être stockées dans une base de données
- les données plus complexes tels que des fichiers. Ces derniers sont plus volumineux que les données simples.

Confidentialité Pour l'ensemble des données de traçabilité, il est important de connaître leur niveau de confidentialité afin de pouvoir évaluer les risques et les coûts liés à leur sécurité.

Rétention La rétention permet de définir la durée de conservation. Ce paramètre est crucial notamment pour évaluer les coûts liés au stockage des données.

3.6.3 Enjeux de l'approche

Les enjeux de l'approche sont multiples et peuvent être représentés par les termes définis ci-dessous :

Sécurité Dans le secteur industriel, les données de traçabilité peuvent être associées à des contraintes de confidentialité donc il est important de maintenir cet aspect. De plus, une usine doit pouvoir contrôler les accès des personnes et entités pouvant accéder ou non à la traçabilité.

Transparence De façon générique, la transparence est un état où un système est dénué de toute tentative de cacher quelque chose. Dans notre approche, il s'agit de pouvoir permettre aux différentes entités de s'assurer qu'elles ont toutes accès à la même information à un instant donné.

Intégrité L'intégrité implique le maintien de la cohérence, de l'exactitude et de la fiabilité des données tout au long de leur cycle de vie.

Confidentialité La confidentialité a été définie par l'Organisation internationale de normalisation (ISO) comme le fait de s'assurer que l'information n'est accessible qu'à ceux dont l'accès est autorisé.

Non-repudiation Dans le domaine de la sécurité des systèmes d'information, la non-répudiation signifie la possibilité de vérifier que l'envoyeur et le destinataire sont bien les parties qui disent avoir respectivement envoyé ou reçu le message.

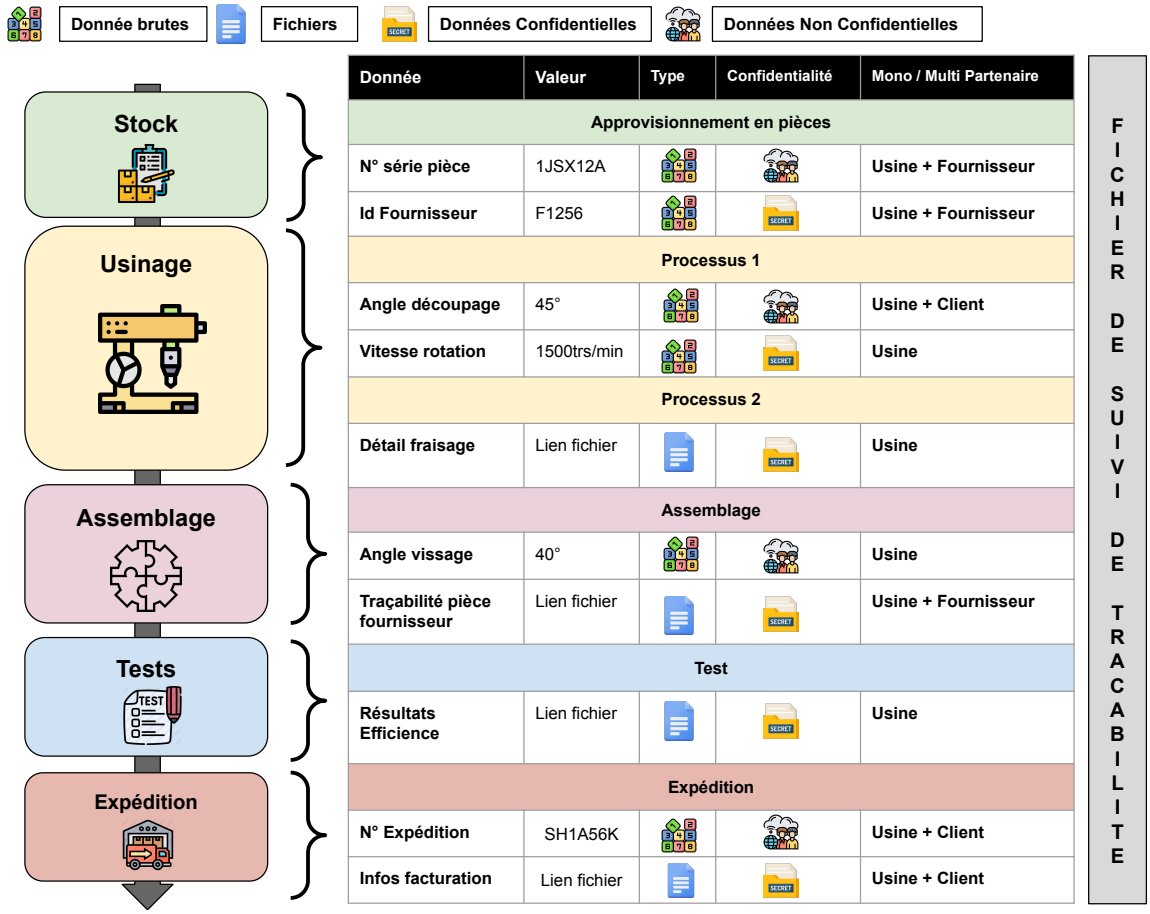
3.7 Synthèse des points forts et points faibles de la contribution

Dans ce chapitre, nous avons vu les différents aspects que peut prendre la traçabilité au sein d'une usine. Tout d'abord, la présentation des bases de la traçabilité a permis d'introduire les différents types et notions inhérentes à ce domaine. Ensuite, la description des principes de mise en oeuvre d'un système de traçabilité a montré comment s'établissent les liens et les associations entre les données de traçabilité tout au long des processus de fabrication du produit. A travers les règlementations, normes mais également l'état de l'art, on a constaté le fait qu'il était critique de pouvoir obtenir une traçabilité fiable que ce soit d'un point de vue sanitaire dans les industries agoralimentaires et pharmaceutiques mais également sécuritaire (industries automobiles). Ces différents éléments permettent de comprendre la nature complexe du sujet dans le sens où la traçabilité du produit ne se limite pas à un seul acteur mais à une chaîne d'acteurs qui constituent ce qu'on appelle la chaîne d'approvisionnement. Les technologies à l'origine de ces données de traçabilité s'avèrent hétérogènes et conduisent par conséquent à la génération de données qui le sont également.

Dans notre approche, nous nous concentrons sur la traçabilité du produit au sein de l'usine. Par conséquent, nous avons réparti cette dernière sous la forme de cinq étapes pouvant impliquer chacune à leur façon des acteurs externes notamment l'assemblage qui implique un fournisseur et l'expédition qui implique le client. Une synthèse montrant l'accumulation des données de traçabilité au fur et à mesure de la fabrication du produit est présentée sous la forme de la Figure 3.16.

En plus de cette classification en différentes étapes, nous attribuons aux différentes données des tags permettant de les qualifier sous différents aspects :

- type
- confidentialité
- périmètre (mono / multi partenaire)
- rétention

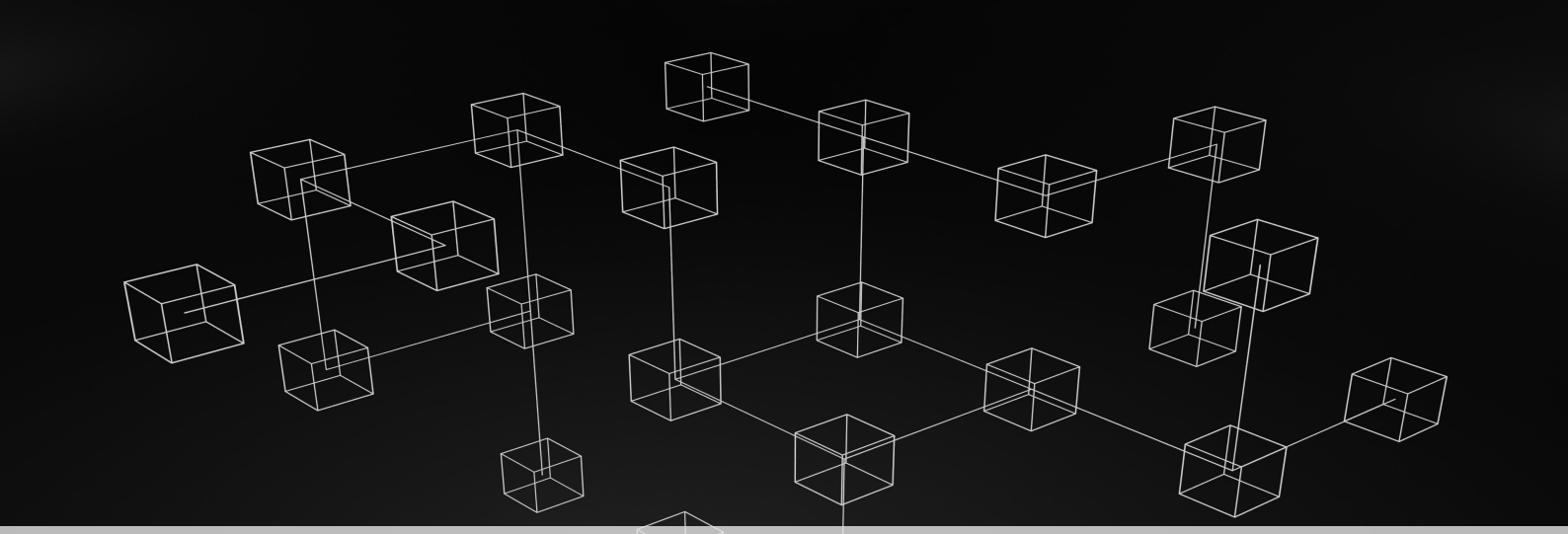


Donnée	Valeur	Type	Confidentialité	Mono / Multi Partenaire
Approvisionnement en pièces				
N° série pièce	1JSX12A			Usine + Fournisseur
Id Fournisseur	F1256			Usine + Fournisseur
Processus 1				
Angle découpage	45°			Usine + Client
Vitesse rotation	1500trs/min			Usine
Processus 2				
Détail fraisage	Lien fichier			Usine
Assemblage				
Angle vissage	40°			Usine
Traçabilité pièce fournisseur	Lien fichier			Usine + Fournisseur
Test				
Résultats Efficience	Lien fichier			Usine
Expédition				
N° Expédition	SH1A56K			Usine + Client
Infos facturation	Lien fichier			Usine + Client

Figure 3.16 – Synthèse d'une fiche de traçabilité d'un produit

L'intérêt de ces tags est d'avoir une meilleure visibilité sur les données stockées dans le système de traçabilité de l'usine afin de faciliter les prises de décision relatives au choix de l'emplacement de stockage (local / cloud), aux précautions à prendre quant à la sécurisation des données ou plus globalement quant aux choix technologiques relatifs au système de traçabilité.

À mesure que les usines deviennent de plus en plus numériques, il y a une demande grandissante afin de pouvoir garantir la fiabilité des données critiques sur les produits, principalement entre les différentes usines durant la chaîne de production. Récemment, la blockchain (la technologie derrière le bitcoin) a été utilisée comme moyen d'inviolabilité des grandes quantités de données générées par les systèmes de traçabilité [93]. La blockchain est une technologie prometteuse qui pourrait, conjointement avec notre approche, répondre à la demande en assurant la transparence, permettant ainsi à toutes les parties prenantes de vérifier l'intégralité de l'historique du cycle de vie d'un produit de façon fiable.



4 Intégration de solution blockchain dans la traçabilité de l'usine 4.0

L'industrie 4.0 apporte des changements majeurs dans la gestion des processus de fabrication. Les tiers, tels que les fournisseurs, les clients et autres prestataires sont plus enclins à interagir avec le système de traçabilité d'une usine afin d'obtenir une transparence totale sur le processus de fabrication d'un produit. Plusieurs solutions, y compris des approches basées sur la blockchain, ont été proposées afin de renforcer la confiance dans de tels systèmes. Cependant, de nombreux propriétaires d'usines sont encore sceptiques quant au déploiement de telles solutions, notamment en raison de menaces à la confidentialité des données de l'usine qui pourraient être incluses dans les données de traçabilité.

Ce chapitre présentera notre approche visant à inclure les données critiques d'une usine dans un système de traçabilité basé sur la blockchain et de valider les transactions associées avec des tiers sans compromettre la confidentialité. Nous présenterons en premier lieu les grands axes du paradigme de la blockchain, un état de l'art de son utilisation en milieu industriel ainsi que notre proposition de système de traçabilité basé sur la plateforme Multichain. En plus des propriétés de préservation de la confidentialité, nous montrons comment d'autres problèmes tels que la consommation d'énergie et le volume de stockage pourraient être contrôlés grâce aux fonctionnalités de Multichain.

4.1 Présentation du paradigme blockchain

La blockchain représente une innovation technologique majeure définie en premier lieu comme une technologie de stockage et de transmission d'informations. Elle offre de hauts standards de transparence et de sécurité car elle fonctionne sans organe central de contrôle. Plus concrètement, elle permet à ses utilisateurs connectés en réseau de partager des données sans intermédiaire.

La première blockchain fut initialisée à partir de 2008 par Satoshi Nakamoto [94] et donna naissance au réseau blockchain Bitcoin [95] qui est utilisée dans le secteur bancaire via les crypto-monnaies.

Même si historiquement, cette dernière s'est développée pour soutenir des transactions réalisées via les cryptomonnaies, son champ d'application s'avère bien plus large et présente des intérêts dans les secteurs

industriels et logistiques notamment pour assurer une traçabilité des produits ainsi que pour établir l'historique des différentes interventions réalisées sur une chaîne de production et de distribution.

Afin de mieux apréhender le voculaire inhérent à la blockchain, la Figure 4.1 suivante est proposée en tant que support aux définitions détaillées tout au long de cette section.

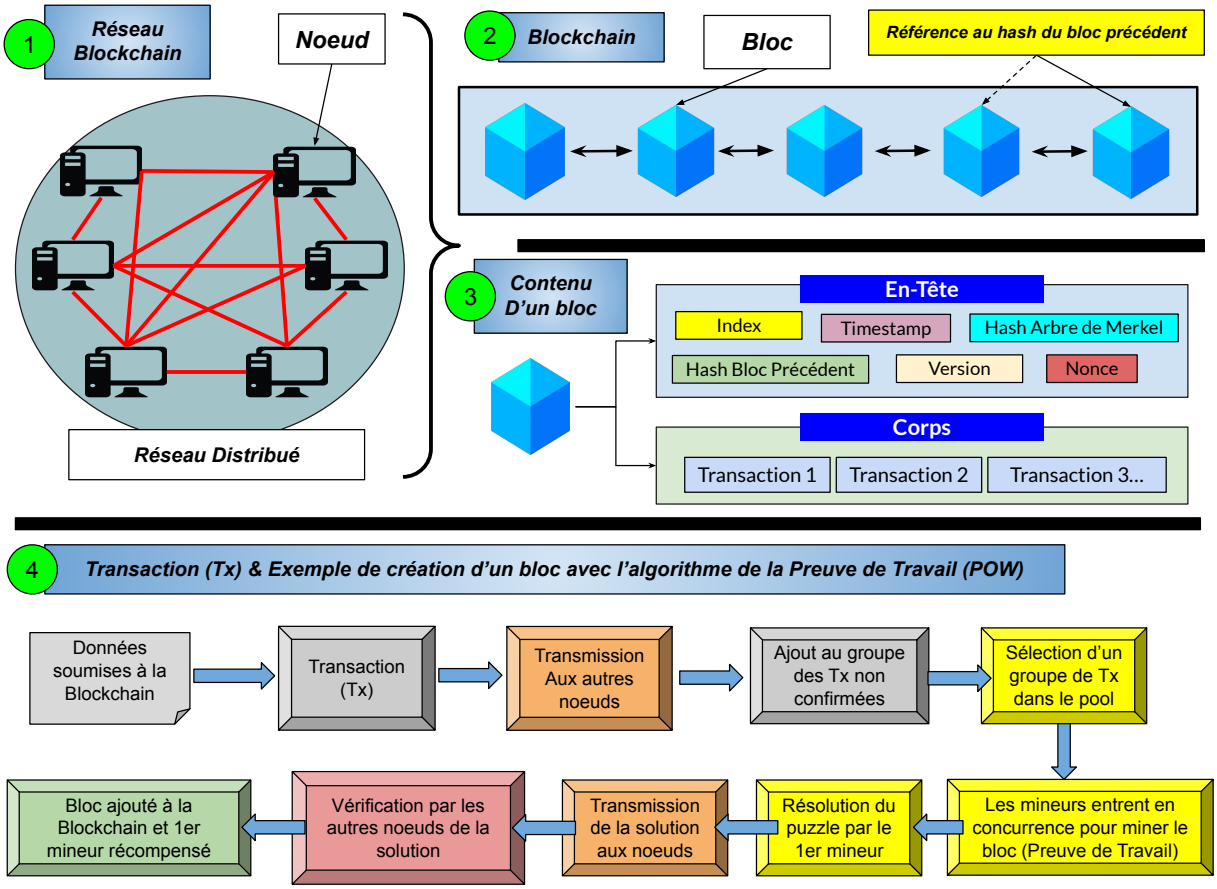


Figure 4.1 – Vue d'ensemble des principaux concepts liés à la blockchain

Blockchain La blockchain est un registre distribué qui contient l'historique de tous les échanges effectués entre les utilisateurs depuis sa création. Elle est constitué d'une liste croissante de blocs où chaque bloc est lié au précédent par un procédé cryptographique. Cette liaison entre les blocs fut introduit en 1991 par [95] et est à l'origine du concept de chaîne. Le point clé de cette liaison réside dans le fait que chaque bloc contient le hash du bloc précédent ce qui permet d'éviter toute modification des données postérieures à leur ajout garantissant ainsi le principe d'immutabilité [77]. Grâce à cette caractéristique fondamentale, un bloc ne peut être altéré rétroactivement sans avoir à modifier l'ensemble des blocs suivants or une telle modification révèlerait une tentative de manipulation des données.

Réseau Blockchain Nom donné au réseau formé par l'ensemble des machines exécutant la blockchain. Chaque membre de ce réseau est appelé un "Noeud". Il est considéré comme un réseau distribué aussi appelé réseau Peer-To-Peer (P2P) dans le sens où il ne dépend pas d'une autorité centrale.

Noeud Il s'agit d'une machine au sein du réseau blockchain. Concrètement, on peut donc considérer qu'un noeud représente une partie, un membre ou un utilisateur. Chaque machine est identifiée via une ou plusieurs adresses uniques dont la génération est basée sur un procédé cryptographique. Ces adresses sont donc la

garantie de pouvoir tracer avec certitude les auteurs des différentes actions réalisées. Chaque noeud héberge localement une copie de la blockchain

Bloc Le bloc est le premier niveau de décomposition dans la structure sous-jacente de la blockchain. Il est divisé en deux partie distinctes : l'en-tête et le corps. L'en-tête contient l'index (position du bloc dans la chaîne), l'horodatage [78] qui rend la preuve d'existence, le hash de l'arbre de Merkel qui correspond au hash de l'ensemble des transactions, le hash du bloc précédent créant ainsi la relation entre les blocs, la version du protocole cryptographique utilisé ainsi que le nonce qui est une valeur représentant la solution de l'algorithme de consensus donc la définition est détaillée plus bas. Le corps du bloc quant à lui contient une liste de transactions. Le hash d'un bloc correspond au hash de l'ensemble des informations contenues dans le header.

Transaction La transaction est le second niveau de décomposition de la blockchain. Elle représente l'ensemble des données soumises et stockées dans la blockchain. Les transactions sont enregistrées dans les blocs, en un nombre limité par un paramètre limitant leur nombre dans le bloc. Lorsqu'un utilisateur soumet des données à la blockchain, cette dernière génère une transaction ainsi que son hash qui permettra l'identifier de manière singulière dans la blockchain.

Algorithme de Consensus En tant que réseau distribué, la blockchain ne possède pas d'autorité centrale permettant d'assurer qu'à tout instant que chaque noeud possède la même version de la blockchain. Ce rôle permettant d'assurer faibilité et confiance entre les différentes parties du réseau blockchain est justement assuré par l'algorithme de consensus. Il peut avoir plusieurs objectifs tels que décider de la validité d'une transaction, synchroniser des états de machine pour assurer leur cohérence ou encore désigner un noeud dirigeant pour une tâche précise. Les algorithmes de consensus partent du principe que certains processus et systèmes seront indisponibles, ils doivent donc être tolérants aux pannes. Selon cette hypothèse, seule une partie des noeuds répondra et parmi ceux-là un minimum de réponses est exigé, par exemple 51%. Différents algorithmes existent, dont la célèbre Preuve de travail (POW) que l'on retrouve dans le bitcoin [94].

« Smart contract » (contrat intelligent) Le smart contract désigne le codage en dur de toutes les clauses du contrat dans un matériel ou un logiciel à exécuter automatiquement dans un environnement sécurisé et distribué [96]. Dans le contexte de la blockchain, il s'agit d'un code informatique stocké dans cette dernière et définissant les règles des transactions entre les différentes parties. Une des premières implémentations des smarts contracts dans la blockchain fut proposée par Ethereum [97] en 2013. Les contrats intelligents automatisent l'exécution des interactions entre les acteurs sans nécessiter d'intermédiaire de confiance. En exprimant les relations juridiques en code plutôt qu'en mots, ils promettent de permettre aux transactions d'avoir lieu directement et sans erreur, délibérée ou non. Ces règles peuvent être exprimées, exécutées et validées de bien des manières dans les plateformes blockchain existantes. Une comparaison de l'implémentation des smart contracts entre différentes technologies blockchain est disponible dans l'annexe A.4.

4.1.1 Types d'implémentations et environnements

Il existe de nombreuses implémentations du paradigme de la blockchain et ces dernières peuvent être classées dans différentes catégories. Certaines blockchains sont implémentées sans système d'autorisation tel que Bitcoin tandis que d'autres sont dites autorisées telles que Hyperledger.

Les blockchains sans autorisation n'ont aucun contrôle sur les accès en lecture/écriture des transactions tandis que les blockchain autorisées ont des mécanismes intégrés pour contrôler les accès en lecture/écriture des transactions soumises par les parties prenantes, et ces accès sont accordés à un nombre limité et connu d'utilisateurs.

Les environnements dans lesquelles ces implémentations peuvent être déployées sont classées en trois catégories : public, privé ou de consortium [98].

Blockchain publique Les accès en lecture et en écriture dans la blockchain sont ouverts à tous. N'importe quel utilisateur dans le monde peut rejoindre le réseau, lire et écrire des transactions dans la blockchain. Il garantit également l'anonymat de l'utilisateur. Ce type de déploiement est plus adapté aux contextes Business to Customer (B2C) ou Customer to Customer (C2C).

Blockchain de consortium Une liste de nœuds présélectionnés contrôle l'accès en lecture et en écriture dans la blockchain. Il est utile dans un contexte d'organisations multiples ayant des besoins de partage sécurisé de données, comme dans le contexte de la chaîne logistique Business to Business (B2B).

Blockchain privée Une seule organisation contrôle tous les droits dans la blockchain.

Le choix d'implémentation de la blockchain dépend donc des exigences des parties prenantes.

4.1.2 Exemples d'algorithmes de consensus

Preuve de travail (POW) Un exemple d'algorithme de consensus qui est utilisé afin de sélectionner un mineur pour l'ajout du prochain bloc dans la blockchain. Le concept consiste à mettre les noeuds en compétition afin de résoudre un problème mathématique complexe appelé puzzle. La réponse à ce problème se nomme le nonce et a été mentionné précédemment dans la définition du bloc. Dans cette compétition, les noeuds portent le nom de mineurs. Une caractéristique essentielle du concept de preuve de travail est l'asymétrie relative au coût de mise en oeuvre. En effet, le travail doit être difficilement réalisable pour l'auteur de la requête alors qu'il est facilement vérifiable par le serveur, appelé le vérificateur. Le premier noeud parvenant à résoudre le puzzle soumet sa solution au réseau qui va procéder à la vérification. Si elle est valide, le mineur gagnant reçoit une récompense (par exemple, de l'argent dans le cas de la blockchain du bitcoin). Cette récompense a pour but d'encourager le respect des règles étant donné qu'il n'y a aucun intérêt à les enfreindre. La preuve de travail participe également au renforcement de la sécurité puisqu'elle rend difficile toute altération de données dans la blockchain. Altérer les données d'un bloc nécessiterait de miner de nouveau tous les blocs générés a postériori et impliquerait de consommer une grande quantité d'énergie (équivalente à ce que l'ensemble du réseau a dû consommer pour miner les blocs en question) [99].

Proof of Stake (POS) Pour résoudre le problème d'inefficacité énergétique de la Preuve de Travail, la preuve d'enjeu a été introduite en premier lieu par Peercoin [100]. Elle propose de lier la puissance minière au pourcentage de jetons détenus par le mineur. Cela réduit considérablement la consommation d'énergie et améliore la sécurité du mécanisme de consensus.

Practical Byzantine Fault Tolerance (PBFT) Dans ce mécanisme de consensus [101], un nœud diffuse un message et lorsqu'il reçoit des autres noeuds $2F + 1$ réponses identiques où F est le nombre maximum de nœuds byzantins tolérés, soit un tiers des noeuds présents dans le réseau PBFT, alors la valeur de la réponse est considérée comme valide. Par conséquent, l'accord est atteint plus rapidement que dans le cas de la POW, avec une consommation d'énergie moindre. Le PBFT est largement utilisé par les blockchains permissionnées telles que Hyperledger Fabric [102].

Replicated And Fault Tolerant (RAFT) Il se décompose en deux étapes principales. La première est l'élection du leader et la seconde est la réplication des logs. Dans l'étape d'élection du leader, un nœud est sélectionné pour agir en tant que leader et en cas d'erreur, un nouveau leader est élu. Dans l'étape de réplication des journaux, le leader accepte les commandes des clients, réplique ses journaux vers les autres nœuds du réseau et en cas de conflit, le contenu du nœud est remplacé avec les journaux du leader. Cet algorithme garantit un débit de transaction élevé. Cependant, il n'est pas conçu pour gérer les nœuds défectueux byzantins. Le RAFT est soutenu par certaines des blockchains autorisées telles que Hyperledger Fabric [102].

Proof of Elapsed Time (PoET) Il a été proposé par Intel afin de résoudre le problème byzantin des nœuds défectueux dans le réseau blockchain. Il s'appuie sur l'utilisation d'un Trusted Execution Environment (TEE) afin de sélectionner le mineur qui générera le bloc suivant. Une fonction est exécutée dans le TEE générant des valeurs de minuterie aléatoires pour les nœuds participants, et le nœud avec la valeur de minuterie la plus basse est désigné pour la prochaine génération de bloc. Cela réduit par conséquent la quantité de calcul requise, mais cet algorithme est dépendant du TEE tel que l'implémentation Intel Software Guard Extensions (SGX) qui est restrictive quant à l'exécution de l'algorithme sur un réseau avec différentes architectures matérielles. Cet algorithme est utilisé par Hyperledger Sawtooth [103], une blockchain permissionnée mise en œuvre par Intel.

4.2 Etat de l'art de la blockchain dans l'industrie 4.0

La traçabilité ne consiste pas seulement à enregistrer des données. D'autres aspects sont impliqués dans la traçabilité, tels que la garantie de la confidentialité et d'un stockage efficace, ainsi que la transparence vis-à-vis des autres acteurs de la chaîne d'approvisionnement. Outre les problèmes de sécurité [104], plusieurs articles de la littérature ont étudié l'utilisation de la technologie blockchain pour répondre à ces aspects [105] [106]. La blockchain est définie comme un registre distribué des événements ou transactions numériques exécutés entre différentes parties, et qui peut être vérifié à tout moment dans le futur. Par exemple, [107] présente une liste de solutions blockchain aux problèmes de traçabilité, essentiellement liés à la coordination des activités individuelles, à la validation décentralisée, à la légitimité et à la préservation des transactions. En offrant transparence [108] et immuabilité, le paradigme blockchain contribue à créer un niveau de crédibilité unique pour toutes les parties prenantes et contribue à renforcer la relation avec les clients ainsi qu'à en attirer de nouveaux.

La sécurité et la sûreté sont d'autres défis mentionnés, mais qui ne sont pas réellement abordés dans ce travail. Afin de fournir des informations approfondies sur cette technologie, [95] propose un examen de la mise en œuvre de solutions basées sur la blockchain pour diverses applications dans lesquelles la sécurité reste primordiale. Il décrit la manière dont la technologie blockchain pourrait résoudre les problèmes rencontrés par les systèmes traditionnels en matière de transparence, de centralisation, d'évolutivité, de confiance et de sécurité. La manière dont une blockchain peut être utilisée pour soutenir le "Smart Manufacturing" est également abordée dans [109]. Les auteurs proposent un middleware qui permet d'obtenir des applications sécurisées, dignes de confiance, traçables et fiables.

Le système de fabrication intelligent sécurisé par blockchain présenté par [110] utilise la blockchain pour résoudre certains problèmes courants dans les systèmes de fabrication, tels que la traçabilité des opérations, la confidentialité et la confiance. De plus, cela permet également d'éviter la défaillance des nœuds clés qui pourrait survenir dans les plates-formes centralisées. Ce système est basé sur la norme internationale ISA95 (www.isa.org), largement référencée dans l'industrie 4.0 en matière de technologies de l'information. Le document présente également certaines mesures liées à l'utilisation de la blockchain dans la fabrication industrielle (manufacturing), à savoir la transparence de la provenance des données, la flexibilité du système, la durabilité du système et les économies de coûts [111]. La protection de la vie privée n'est mentionnée que comme une future direction de recherche pour la blockchain.

Afin d'étudier la précision de la technologie blockchain pour la transparence en temps réel et les économies de coûts dans l'industrie manufacturière, [111] propose une comparaison des bénéfices réalisés par deux entreprises manufacturières. Plusieurs aspects sont considérés dans cette étude, notamment les coûts de déploiement de la blockchain [112] qui sont souvent ignorés, et certaines autres limitations liées à cette technologie. Ces problèmes de coût et de profit de mise en œuvre sont également examinés par [113] dans leur jeu de chaîne d'approvisionnement composé de deux entreprises, à savoir un fournisseur et un détaillant, utilisant la blockchain et les smart contracts. Les aspects étudiés incluent les risques commerciaux, les coûts de transaction et les cas stochastiques dans lesquels le déploiement de la blockchain n'en vaut pas la peine. Une autre proposition de [114] était une architecture de chaîne de blocs à trois niveaux pour les systèmes cyber-physiques (CPS) afin de relever les défis associés à la mise en œuvre de la structure 5C-CPS. En termes

de sécurité et de stockage, l'architecture tente de garantir l'intégrité, la tolérance aux pannes, la résilience, la confidentialité et la transparence.

Avec l'avènement de l'industrie 4.0, de nouveaux paradigmes de fabrication sont apparus, comme le cloud manufacturing. Cependant, ils souffrent encore de certains problèmes liés aux réseaux industriels centralisés. Par conséquent, une architecture en réseau peer-to-peer (P2P) basée sur la blockchain a été présentée par [115]. Les principaux objectifs de cette architecture étaient de sécuriser le partage des données et d'améliorer la fiabilité et la flexibilité du cloud manufacturing. Garantir la fiabilité des données est un problème clé pour tirer parti des outils d'analyse liés au big data afin d'améliorer l'efficacité de la chaîne d'approvisionnement. Ainsi, [116] souligne qu'il est essentiel de déterminer si les informations collectées à partir des capteurs sont valides ou non. Afin de résoudre ce problème, une combinaison d'UAV (véhicules aériens sans pilote) et d'un système basé sur la blockchain est présentée pour gérer les applications d'inventaire et de traçabilité dans la gestion de la chaîne d'approvisionnement basée sur les outils big data. Cependant, bien que la technologie blockchain soit prometteuse, la divulgation publique de toute transaction est considérée comme un risque de sécurité par la plupart des organisations [72].

Un système basé sur la blockchain préservant la confidentialité pour les systèmes de traitement des transactions est présenté dans [117] et fournit à ces problèmes quelques solutions. L'idée principale consiste à trouver un compromis entre le bénéfice du partage d'informations (transparence) et le coût associé à l'affaiblissement de la confidentialité. La préservation de la confidentialité dans la blockchain fait référence au fait que seules les parties autorisées peuvent accéder aux données de transaction sensibles ou exclusives. Le système proposé est basé sur le cryptage homomorphe et une innovation récente appelée preuves à connaissance nulle *zero-knowledge proof*. Nous poursuivons les mêmes objectifs, tout en évitant les preuves à connaissance nulle qui introduisent des coûts de traitement plus élevés et peuvent ne pas être pratiques pour un large éventail de types de données impliqués dans les processus de fabrication.

4.3 Présentation de l'approche proposée

Dans cette partie, nous allons présenter notre approche visant à intégrer l'utilisation de la blockchain dans le cadre de la traçabilité centrée produit. Cette approche se nomme Blockchain Product-Centered Approach for Traceability (BPCAT) et sera référencée tout au long de ce chapitre par l'acronyme BPCAT.

4.3.1 Objectifs et garanties visées

Plusieurs travaux sur l'application de la blockchain à la traçabilité ont déjà été mentionnés dans la section 4.2. La technologie blockchain est principalement connue pour son immuabilité et sa transparence, ce qui en fait une solution toute désignée pour relever les défis de la traçabilité dans l'Industrie 4.0 [113]. Cependant, ces fonctionnalités ne suffisent pas à elles seules à répondre à toutes les préoccupations liées à la traçabilité. Au lieu de cela, dans notre travail, l'immuabilité sera utilisée comme une caractéristique centrale autour de laquelle d'autres concepts graviteront pour augmenter leur fiabilité. Notre approche blockchain vise à assurer la confiance entre les parties prenantes participantes et à fournir d'importants avantages pour la traçabilité des produits concernant les aspects suivants :

Sécurité Dans la fabrication, les données de traçabilité sont associées à des problèmes de confidentialité, il est donc important de maintenir cette confidentialité dans notre approche. De plus, une usine doit maîtriser le processus de validation des participants à la blockchain qui accèdent à son système de traçabilité. Contrairement aux solutions de blockchain publiques traditionnelles dans lesquelles les participants peuvent être presque anonymes, une blockchain privée telle que celle utilisée dans ce travail ne permet qu'à un nombre restreint d'utilisateurs d'accéder et d'utiliser la blockchain.

Transparence Dans une blockchain, chaque transaction validée génère un hash de transaction qui peut être consulté par n'importe quel participant. Cela permet de vérifier qu'une action spécifique a été effectuée dans la blockchain à un instant donné.

Intégrité L'intégrité implique le maintien de la cohérence, de l'exactitude et de la fiabilité des données tout au long de leur cycle de vie. Pour garantir l'intégrité, les données de traçabilité sont hachées et le hachage est stocké dans la blockchain. Étant donné que la blockchain est inviolable, le hash peut être considéré comme inviolable par extension. La vérification d'intégrité consiste à calculer le hash des données, et à le comparer avec le hash d'origine stocké dans la blockchain.

Confidentialité La transparence ne signifie pas que toutes les informations doivent être rendues publiques. En effet, les données de traçabilité peuvent contenir certaines informations privées liées au processus de fabrication. Par conséquent, nous considérons que les données confidentielles doivent être gérées de manière appropriée ; par exemple en étant cryptées avant leur insertion dans la blockchain ou en n'insérant dans celles-ci certaines données dérivées des données confidentielles au lieu des données confidentielles elles-mêmes. Cependant, le hachage des données inséré dans la blockchain concernant les données confidentielles doit être calculé à partir des données d'origine avant le cryptage ou toute autre opération. En effet, l'authenticité et l'intégrité devront être vérifiées par rapport aux données réelles. De cette manière, il sera possible d'assurer à la fois la confidentialité (grâce au cryptage des données) et la transparence (grâce aux données dérivées et au hachage des données d'origine). Lorsque les données dérivées sont préférées au chiffrement, le protocole produisant ces données dérivées doit être exposé en toute transparence et doit garantir que les données d'origine ne peuvent pas être récupérées à partir des données dérivées. Les protocoles proposés dans les approches à connaissance zéro offrent de telles garanties. Cependant, ces approches sont complexes et peuvent ne pas être pratiques dans le contexte de la traçabilité des usines de fabrication en raison de la quantité de données et du fait que ces dernières sont produites en temps réel lors de la fabrication du produit.

Non-répudiation Elle garantit qu'un participant ne peut nier aucune de ses actions, qu'elles soient liées aux données ou aux transactions dans la blockchain. Étant donné que les participants sont authentifiés dans une blockchain privée, des mécanismes spécifiques doivent être introduits afin de vérifier et de garder une trace de la validation des données et des transactions. Un tel mécanisme pourrait être l'utilisation de signatures numériques. En effet, ce dernier offre la possibilité de signer les données avec une clé privée et la signature peut être vérifiée par toute personne membre de la blockchain en utilisant la clé publique correspondante.

La Figure 4.2 décrit une application de notre approche de la traçabilité centrée sur le produit. Deux étapes de traçabilité sont représentées.

Au cours de la phase d'assemblage, deux composants sont ajoutés à un produit ayant déjà subi plusieurs transformations ; où le composant A a été fabriqué par un fournisseur tandis que le composant B a été fabriqué à l'intérieur de l'usine. Pendant la phase de test, des tests de conformité sont exécutés par l'usine et produisent des données de test confidentielles. Au milieu de la figure, on retrouve le produit finalisé qui agrège toutes les données de traçabilité avec différents niveaux de confidentialité (confidentiels ou non) et différents intervenants (fournisseur ou usine). Toutes ces données seront hachées afin de garantir leur intégrité ultérieurement dans la blockchain. Cependant, seules les données confidentielles seront cryptées (ou dérivées).

4.3.2 Vue d'ensemble de l'architecture

BPCAT est une architecture couvrant deux aspects : la traçabilité du produit et son intégration dans la blockchain. Cette architecture globale est représentée dans la Figure 4.3, qui sera décrite de haut en bas.

Trois nœuds de la blockchain représentent les types d'acteurs ou de participants, à savoir le validateur (l'entreprise propriétaire des données de traçabilité, autrement dit l'usine), un client (détaillant, utilisateur

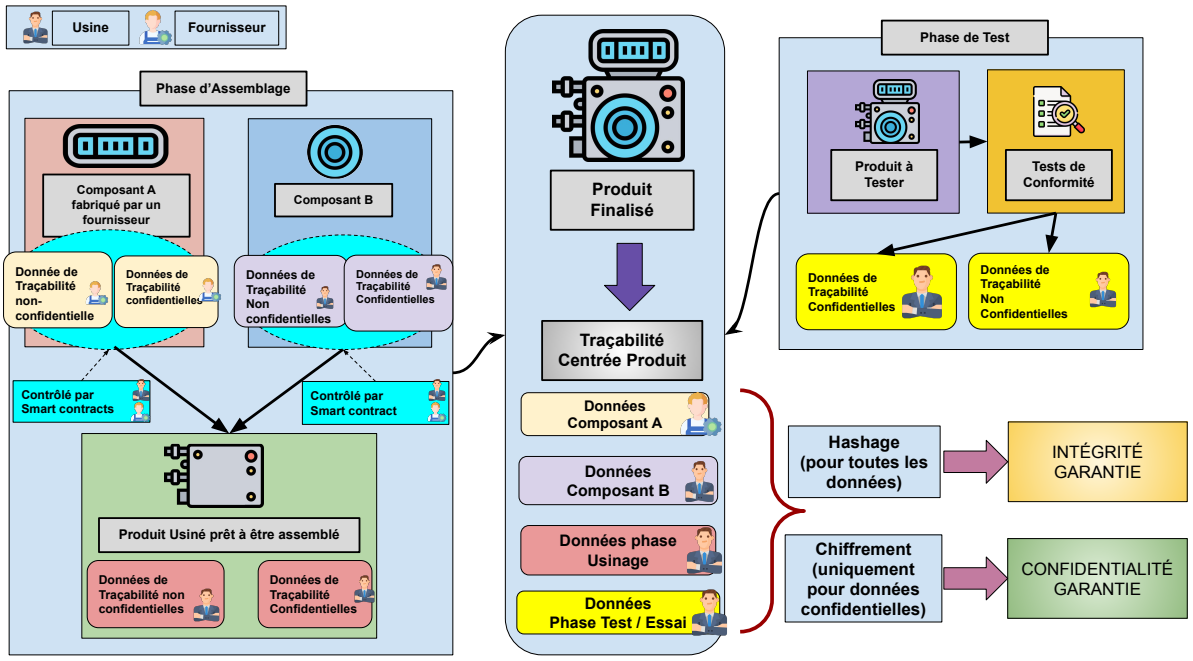


Figure 4.2 – Vue conceptuelle de l'approche traçabilité centrée produit

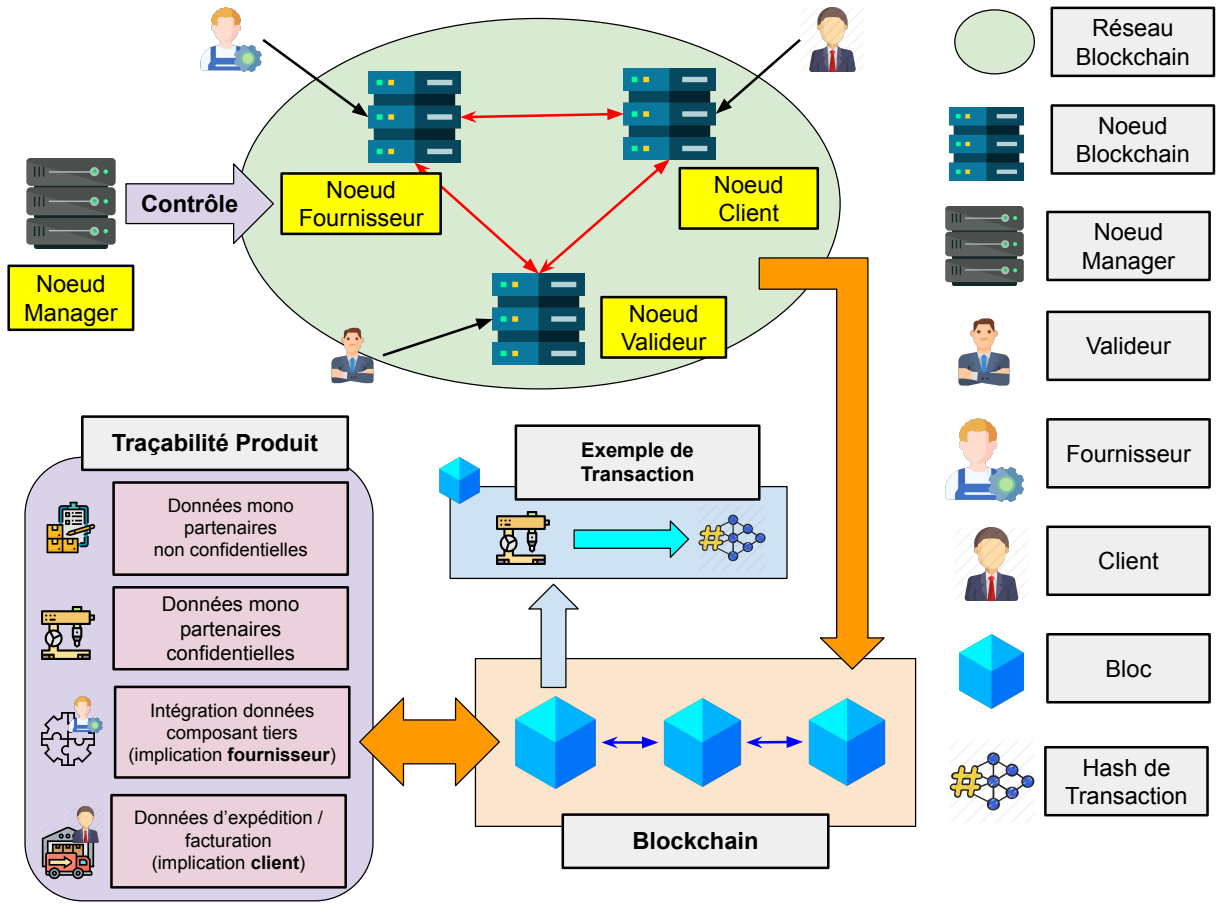


Figure 4.3 – Architecture générale de BPCAT

final, etc.) et un prestataire (fournisseur de composants, fournisseur, etc.) impliqués dans l'élaboration du produit. Ces nœuds forment ensemble le réseau blockchain qui peut être contrôlé à distance par un nœud

appelé nœud gestionnaire. Ce dernier vise à donner plus de contrôle au propriétaire de l'usine (ou à un opérateur désigné) dans la blockchain.

Cette implémentation blockchain s'apparente donc à une blockchain de consortium qui s'avère plus adaptée au contexte de la chaîne d'approvisionnement dans laquelle toutes les identités des parties prenantes sont connues et où la blockchain est contrôlée non pas par une seule entité mais par un consortium de parties prenantes qui peuvent participer de différentes façons à la blockchain grâce au système de permissions. Les nœuds sont également déployés pour chacune des parties prenantes (usine, fournisseur, client). Cela garantit plus de transparence, plus de confiance entre les parties prenantes et la disponibilité du système de chaque côté. A l'inverse, une blockchain publique aurait constitué un mode de travail entièrement transparent mais aurait posé des problèmes juridiques et éthiques tandis qu'une blockchain privée implique un contrôle unilatéral par un seul des acteurs ce qui aurait pu poser un problème en termes de transparence et de disponibilité du système.

4.3.3 Gestion des données

Données brutes

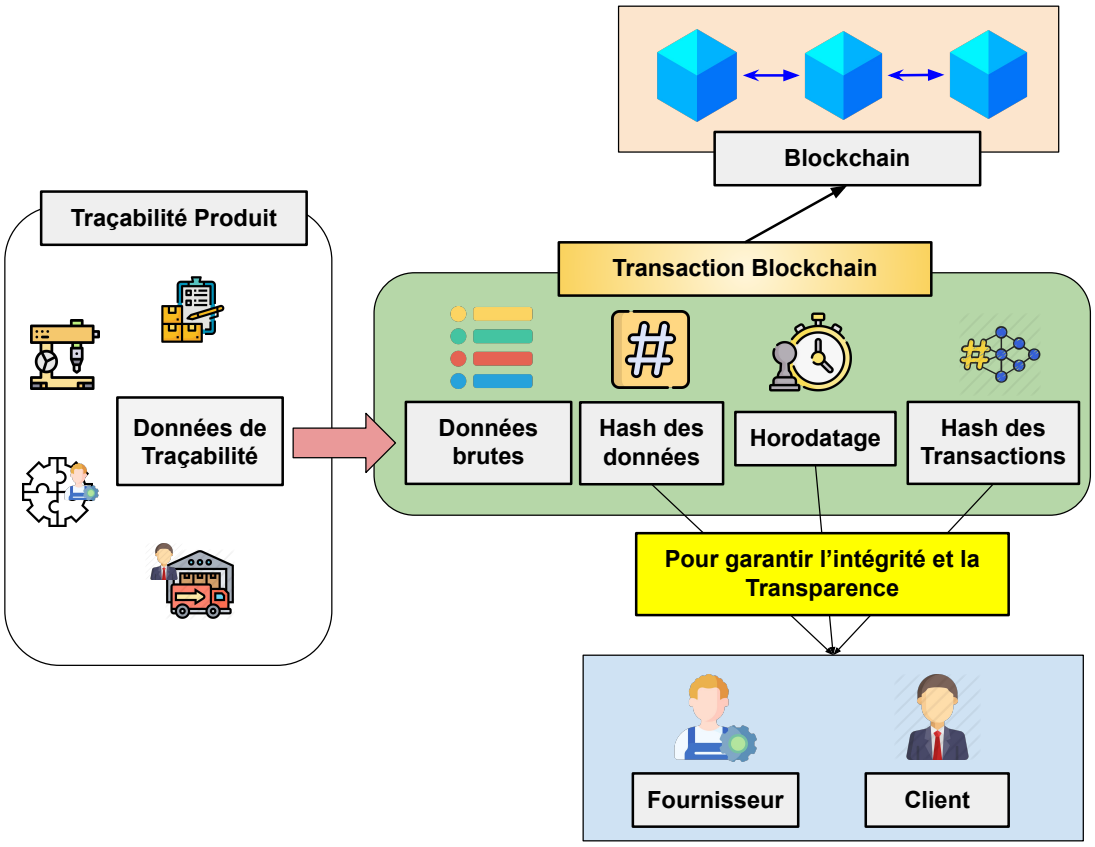


Figure 4.4 – Gestion des données brutes par BPCAT

Dans la blockchain, chaque fois qu'une nouvelle donnée de traçabilité est ajoutée à un bloc, une transaction est créée. BPCAT a certaines conditions préalables sur la façon dont les données doivent être structurées comme le montre la Figure 4.4. Une transaction blockchain habituelle comprend toujours un hachage de transaction et un horodatage afin de garantir qu'une action a été effectuée à un moment précis dans la blockchain. Cependant, le hash de la transaction ne garantit pas l'intégrité des données. Pour cette raison, BPCAT ajoute également le hash des données dans la transaction. Avec le hachage des données, l'horodatage et

le hachage des transactions, la transparence des transactions et l'intégrité des données peuvent être garanties au fournisseur et au client.

Fichiers

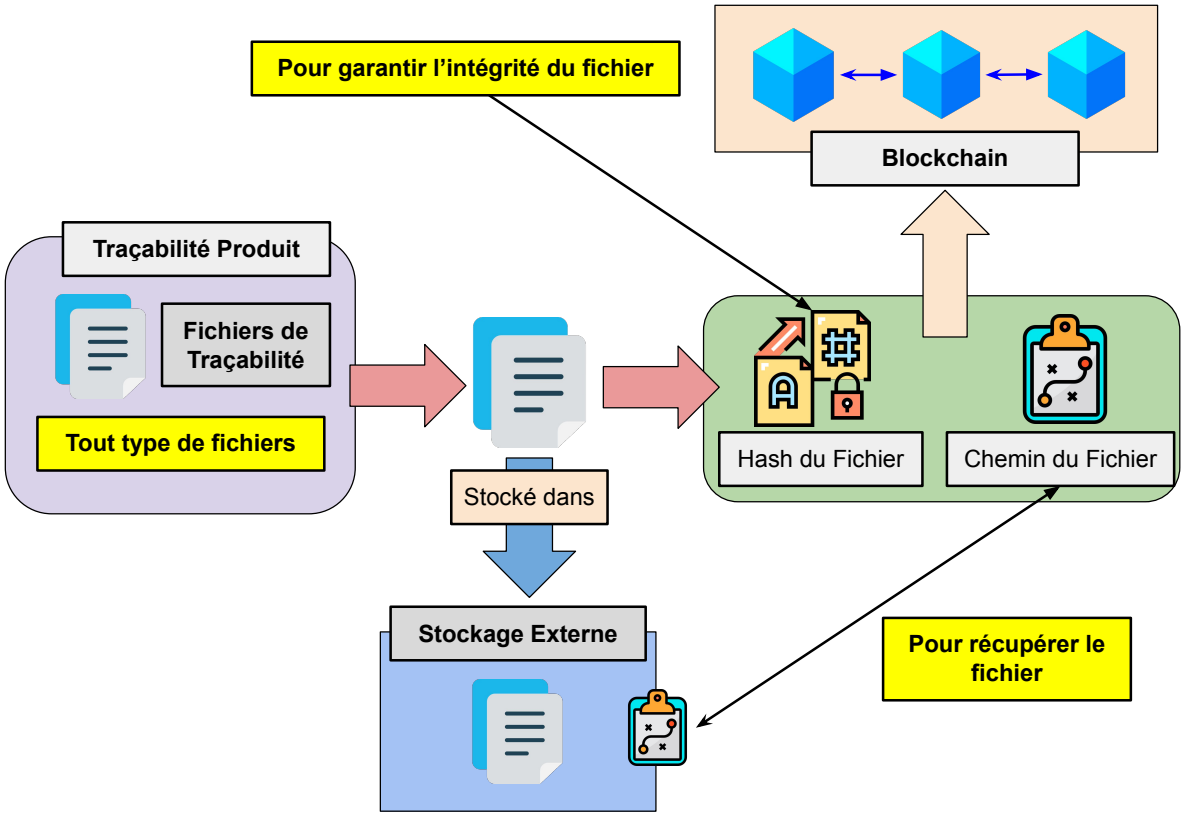


Figure 4.5 – Gestion des fichiers par BPCAT

Afin de faire face au problème volumétrique des données de traçabilité qui pourrait nécessiter le recours à une solution de stockage cloud, la politique de gestion des fichiers BPCAT n'impose pas le stockage des fichiers à l'intérieur de la blockchain. Chaque participant peut utiliser une solution de stockage externe comme on peut le voir sur la Figure 4.5. Cependant, le hash du fichier et le chemin du fichier sont stockés dans la blockchain ; le premier afin de garantir l'intégrité et l'authenticité du fichier, et l'autre pour récupérer facilement le fichier. De cette façon, n'importe lequel des membres peut recourir à un stockage cloud si nécessaire.

4.3.4 Gestion de la confidentialité

Les données de traçabilité confidentielles font référence aux données que le propriétaire de l'usine ne souhaite pas révéler à l'ensemble des participants de la blockchain, mais qui doivent de toute façon être incluses dans les données de traçabilité car elles pourraient être nécessaires comme preuves lors d'enquêtes rétrospectives. En effet, afin d'éviter un conflit avec la transparence habituellement attendue d'une solution blockchain, BPCAT gère les données confidentielles de la manière suivante :

— les données confidentielles sont cryptées à l'aide d'une méthode de cryptage choisie par le propriétaire des données afin de garantir la confidentialité ;
— Afin de garantir la transparence, BPCAT suggère que toute donnée cryptée dans une transaction soit accompagnée d'une information liée qui pourrait évaluer son authenticité et son intégrité en cas de besoin. Le hash des données d'origine avant chiffrement, ou le hash d'une donnée dérivée des données

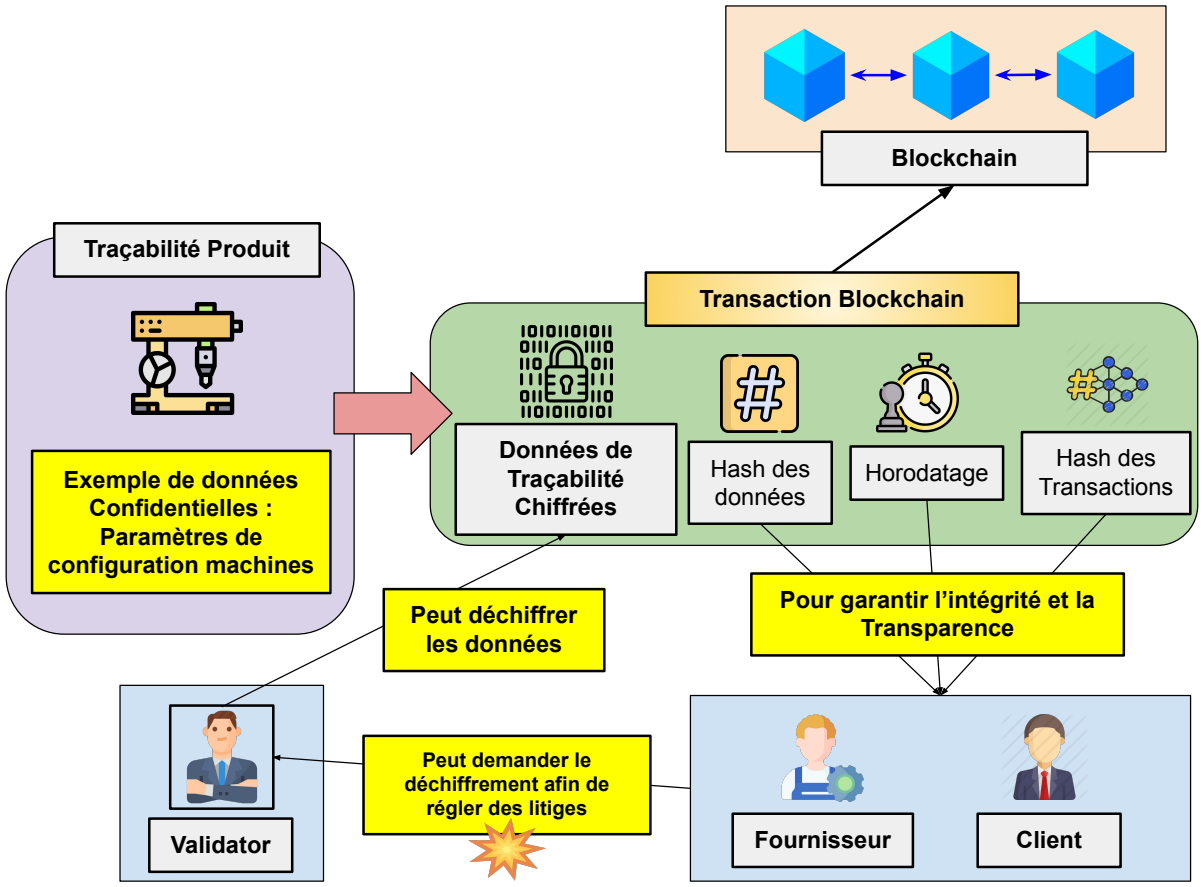


Figure 4.6 – Gestion de la confidentialité avec BPCAT

d'origine de manière prédéfinie, peut être utilisé à cette fin. Toutes ces données sont insérées dans la transaction, et l'horodatage et le hash de la transaction garantissent l'authenticité et l'intégrité de l'ensemble de la transaction.

— en cas de litige, un participant peut demander le déchiffrement des données afin qu'elles soient comparées au hash stocké avec elles dans la blockchain, et établir ainsi leur authenticité.

La Figure 4.6 montre que les données chiffrées ne peuvent être déchiffrées que par le propriétaire (ici le valideur) afin de maintenir la confidentialité. Cependant, le fournisseur et le client ont accès au hash des données d'origine, à l'horodatage et au hash de la transaction, ce qui garantit l'intégrité, l'authenticité et la transparence.

4.3.5 Gestion de la non-répudiation

Concernant la traçabilité, elle peut être utile pour résoudre des litiges où la responsabilité d'un acteur est mise en cause. À cette fin, les signatures numériques peuvent être utilisées dans la blockchain comme illustré dans la Figure 4.7. Pour chaque transaction blockchain impliquant des données de traçabilité, chaque acteur (validateur, client, fournisseur) impliqué dans l'étape de fabrication du produit associé doit signer la transaction. A l'aide de sa clé privée personnelle, chaque intervenant signe un message contenant le hash de la transaction et un horodatage afin d'authentifier la signature. Ensuite, une fois la signature numérique calculée, elle est enregistrée dans la blockchain. La non-répudiation est garantie grâce à toutes les signatures, qui peuvent être vérifiées par toute personne utilisant la clé publique signataire.

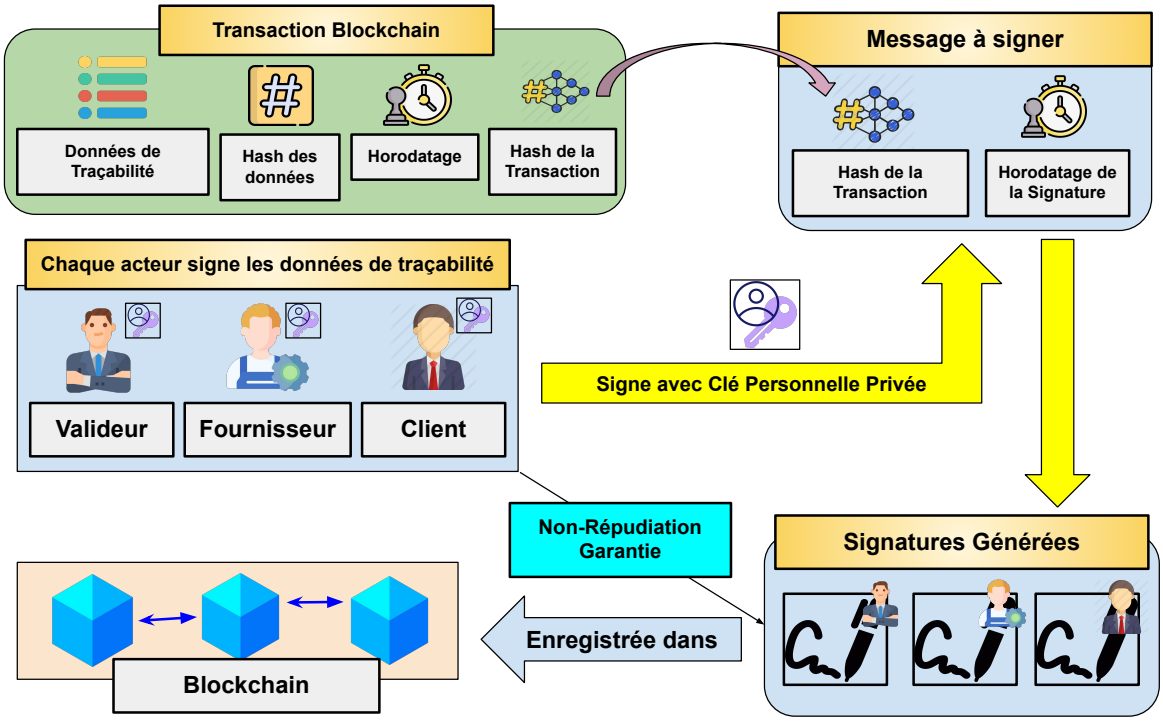


Figure 4.7 – Gestion de la non-répudiation avec BPCAT

4.4 Implémentation de l'approche avec Multichain

Afin de mettre en pratique notre approche, un prototype a été développé à l'aide de l'outil Multichain. Il s'agit d'une plate-forme open source conçue pour créer des blockchain permissionnées. Les principaux avantages de cette plateforme sont le contrôle total sur tous les aspects de la blockchain (gestion des autorisations, algorithmes de consensus, stockage des données...), la possibilité de créer plusieurs blockchains et sa facilité d'utilisation pour les développeurs. Nous nous concentrerons ici sur les fonctionnalités du prototype ayant un rôle direct avec l'approche, cependant une description plus technique est disponible en annexe Chapitre B.

4.4.1 Architecture réseau et représentation des acteurs

Dans Multichain, les participants à la blockchain sont appelés "nœuds", et le réseau de la blockchain est constitué de connexions entre ces nœuds. Dans BPCAT, les nœuds sont comme les avatars des joueurs dans le système de traçabilité. L'implémentation proposée comprend un total de cinq noeuds :

- Valideur : Avatar of the factory owning the traceability data ;
- Client : Avatar du client ;
- Fournisseur : Avateur du fournisseur ;
- Gestionnaire : Noeud gérant l'état des autres noeuds et générant les adresses / clés privées ;
- Archivage : Noeud stockant les anciennes blockchains et permmettant de les consulter.

La Figure 4.8 est une représentation de l'architecture du réseau incluant les interactions entre les participants de la blockchain. Trois réseaux sont considérés : le réseau blockchain formé par les nœuds de la blockchain, le réseau d'entreprise dans lequel les données de traçabilité sont générées et soumises à la blockchain, et le réseau externe qui est une abstraction de toutes les interactions avec les entités extérieures à l'usine.

Les noeuds chauds (valideur, client, fournisseur) constituent le réseau blockchain et sont connectés les uns aux autres. En tant que nœud principal, le Valideur représentera l'usine de fabrication et publiera les données de traçabilité. Les nœuds Client et Fournisseur seront les points d'entrée de la transparence. En effet, ils permettront au client et au fournisseur d'accéder aux données de traçabilité.

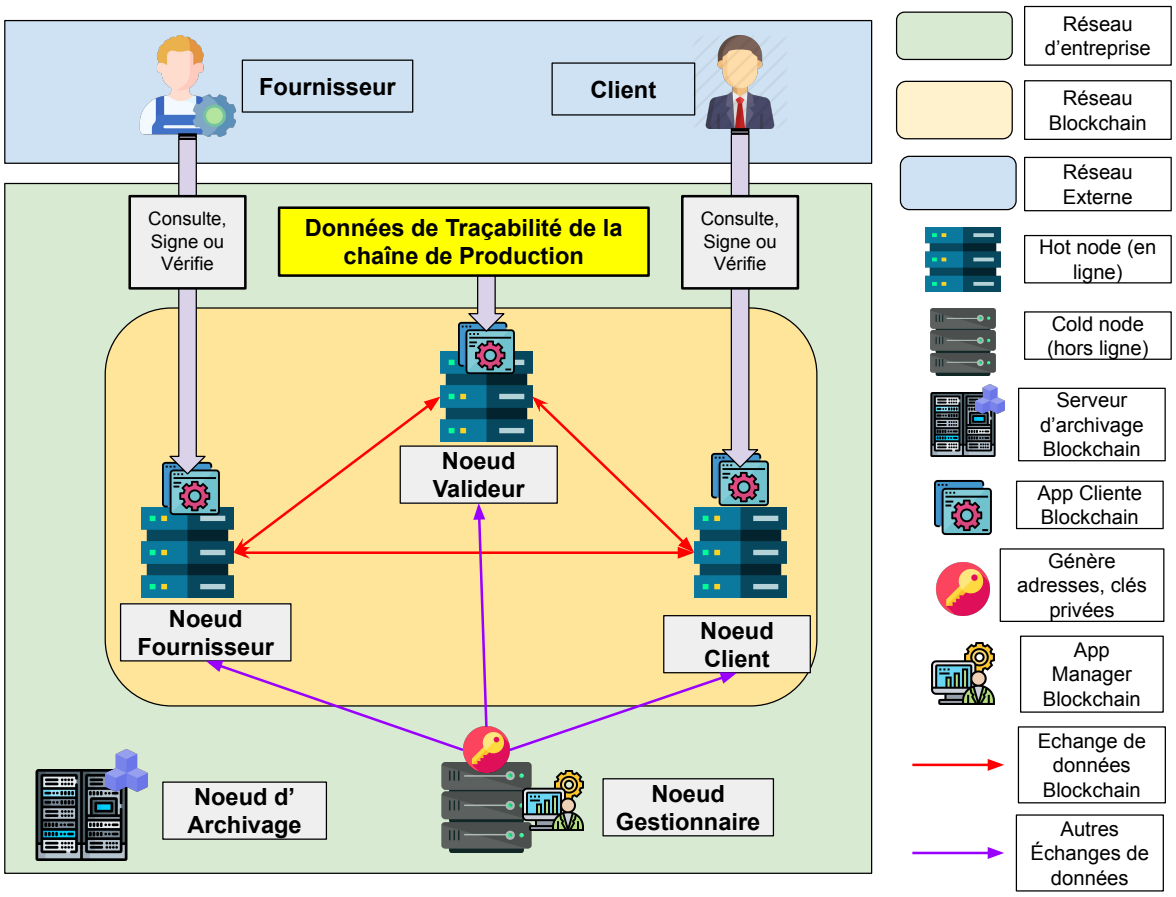


Figure 4.8 – BPCAT Vue d'ensemble de l'implémentation réseau et des noeuds

Contrairement aux noeuds précédents, le noeud Gestionnaire fait référence à un noeud "froid" Multichain : c'est un nœud isolé qui n'est pas connecté à la blockchain. Du point de vue de la blockchain, ce nœud "n'existe" pas réellement. Son rôle consiste à surveiller l'état de la blockchain et des autres noeuds. La raison de son existence est de répondre à l'un des principaux problèmes souvent mentionnés par les entreprises à propos de la blockchain, essentiellement : "la blockchain n'est pas facilement gérable en raison de sa caractéristique décentralisée". Le contrôle à distance consiste à démarrer et à arrêter d'autres nœuds, ou plus globalement à instancier une nouvelle blockchain. De plus, il est également responsable de la génération des adresses des autres nœuds, des clés privées/publiques pour le cryptage, le décryptage et la signature des données. Une infrastructure à clé publique (PKI) externe choisie par les partenaires pourrait assurer certaines de ces fonctionnalités.

Le noeud Archivage est dédié au stockage des anciennes blockchains et donne la possibilité de les explorer ultérieurement. La Figure 4.9 montre les trois noeuds de la blockchain et comment ils sont surveillés depuis le noeud gestionnaire dans Multichain.

Enfin, il faut remarquer que du fait de la composition spécifique des nœuds de la blockchain qui regroupe l'usine et ses partenaires, le consenus peut être simplifié afin d'éviter une consommation d'énergie inutile.

4.4.2 Gestion du stockage et de la blockchain

Comme mentionné précédemment, Multichain fournit un moyen simple de créer et de démarrer de nouvelles blockchains. Dans BPCAT, cette fonctionnalité est utilisée comme un moyen d'éviter un autre inconvénient de la technologie blockchain pour les usines de l'industrie 4.0, à savoir sa caractéristique qui consiste à être "en croissance constante". Nous suggérons un critère de temps définissant la durée de vie d'une blockchain avant

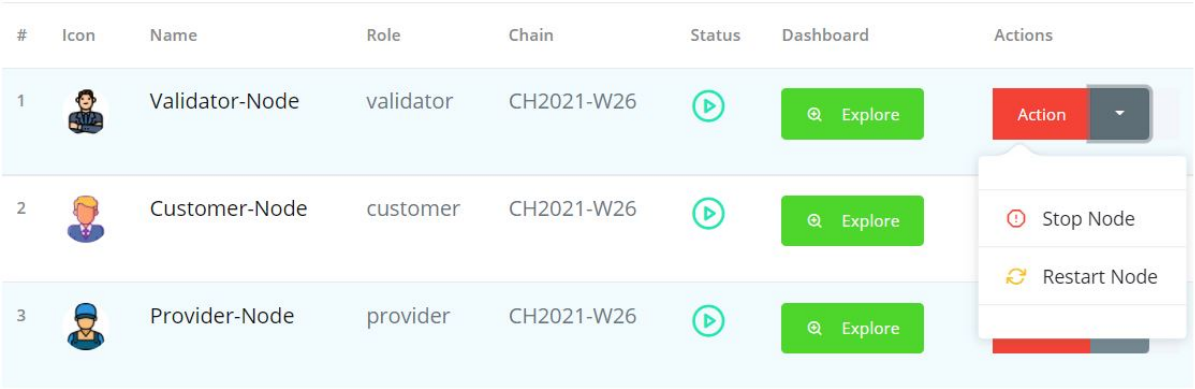


Figure 4.9 – BPCAT Visualisation du statut des noeuds et de la blockchain depuis le noeud manager

de l'arrêter et d'en démarrer une nouvelle. Ce faisant : la taille des blockchains est stable, et la volumétrie des données ainsi que le coût de stockage sera faciles à estimer et à anticiper. Ensuite, la conservation de l'historique des anciennes chaînes est gérée par le nœud Archivage, et la conservation des données consiste à supprimer les blockchains dont l'âge dépasse une valeur définie selon la politique de traçabilité de l'usine.

4.4.3 Gestion des données de traçabilité brutes

Afin de gérer et de stocker les données, Multichain propose une abstraction des concepts blockchain dédiés au stockage et à la gestion des données appelés "streams". Ce sont des conteneurs de données fonctionnant comme des stockages clé-valeur. Chaque donnée est appelée un item, et un item est un couple clé-valeur dans lequel chaque clé est utilisée pour indexer et récupérer un élément spécifique. Un élément peut avoir plusieurs clés et la même clé peut être utilisée plusieurs fois. La Figure 4.10 offre un aperçu de la façon dont les données de traçabilité sont organisées dans l'implémentation ainsi que des différents types de données rencontrés.

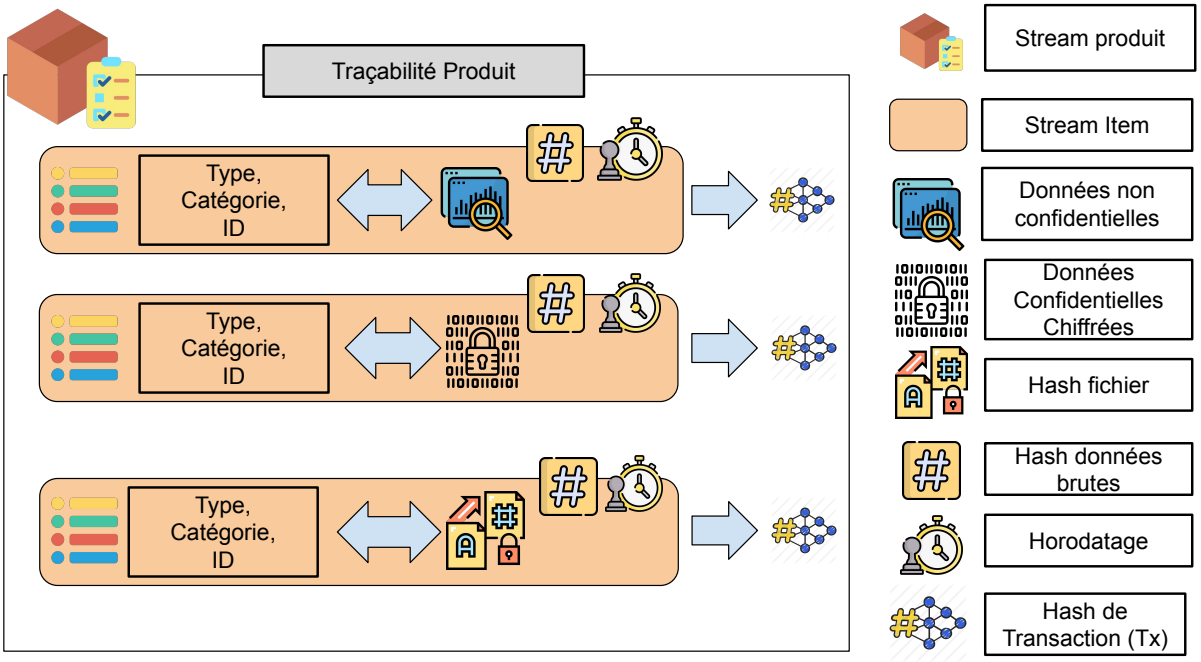


Figure 4.10 – BPCAT Implémentation de la gestion des données brutes

Dans la mise en oeuvre de l'approche, nous utilisons un stream par produit dans lequel le nom du stream peut être le numéro de série du produit ou tout autre attribut unique de ce dernier. Chaque donnée de traçabilité

du produit est conditionnée sous la forme d'un item dont les clés sont son type ou sa catégorie. Pour chaque item, le hachage de données est stocké afin que l'intégrité et l'authenticité des données puisse être vérifiée.

Dans l'extrait de Code 4.1, on peut observer la structure d'un item tel qu'il est soumis à Multichain par le prototype en JSON. Cet item se décompose en deux parties : un en-tête appelé *header* contenant le type, l'horodatage, le niveau de chiffrement ainsi que le hash des données nommé *data_hash*. Le corps, quant à lui, correspond aux données de traçabilité.

Code 4.1 – Exemple de données de traçabilité brutes en JSON dans un stream Multichain

```
{
    "header": {
        "data_type": "TRACEA_STEP",
        "encrypted": false,
        "timestamp": 1660060004,
        "data_hash": "eaf19d9b4ad0cdcf125c395e1c292df9e3a7cc5c5fc3b52997270ba64bb7e515"
    },
    "body": {
        "location": "Line9-u-3",
        "location_entry": 1660063000.73701,
        "location_exit": 1660064354.73701,
        "process_desc": "Reduced discrete Internet solution"
    }
}
```

4.4.4 Garantie de la confidentialité

Dans notre approche, nous considérons que les données peuvent être divisées en deux catégories du point de vue de la sécurité : les données confidentielles et non confidentielles. Pour gérer cette différence, les données confidentielles doivent être chiffrées avant leur insertion dans la blockchain. La Figure 4.11 décrit la façon dont la confidentialité des données est préservée dans l'implémentation avec Multichain. Un cryptage hybride utilisant un cryptage symétrique (AES 128) et un cryptage asymétrique (RSA 2048) est utilisé. Tout d'abord, les données sont cryptées à l'aide d'une clé de cryptage aléatoire unique générée par le cryptage symétrique. Ensuite, cette clé de cryptage est cryptée avec le cryptage asymétrique pour produire ce que l'on appelle une clé de cryptage cryptée (clé CC). La clé CC ne peut être déchiffrée que par ceux qui possèdent la clé privée associée. La clé CC est stockée dans un stream dédié en tant que stockage "Tx Hash - Clé CC", où "Tx Hash" est le hachage de transaction produit lorsque les données chiffrées ont été stockées dans la blockchain. Ainsi, déchiffrer les données d'une transaction consiste à récupérer la clé CC liée au hash de la transaction, à la déchiffrer avec la clé privée asymétrique, et à déchiffrer les données avec la clé déchiffrée finale.

L'extrait de Code 4.2 présente la fonction *encrypt_bytes* qui illustre cette façon de procéder. Cette dernière permet de chiffrer un tableau de données sous forme de bytes et retourne en résultat la donnée chiffrée *encr_data* ainsi que la clé CC représentée par la variable *encr_aes* et renvoyée sous la forme d'un objet *StreamItemSecret* pour pouvoir être inséré dans un stream Multichain.

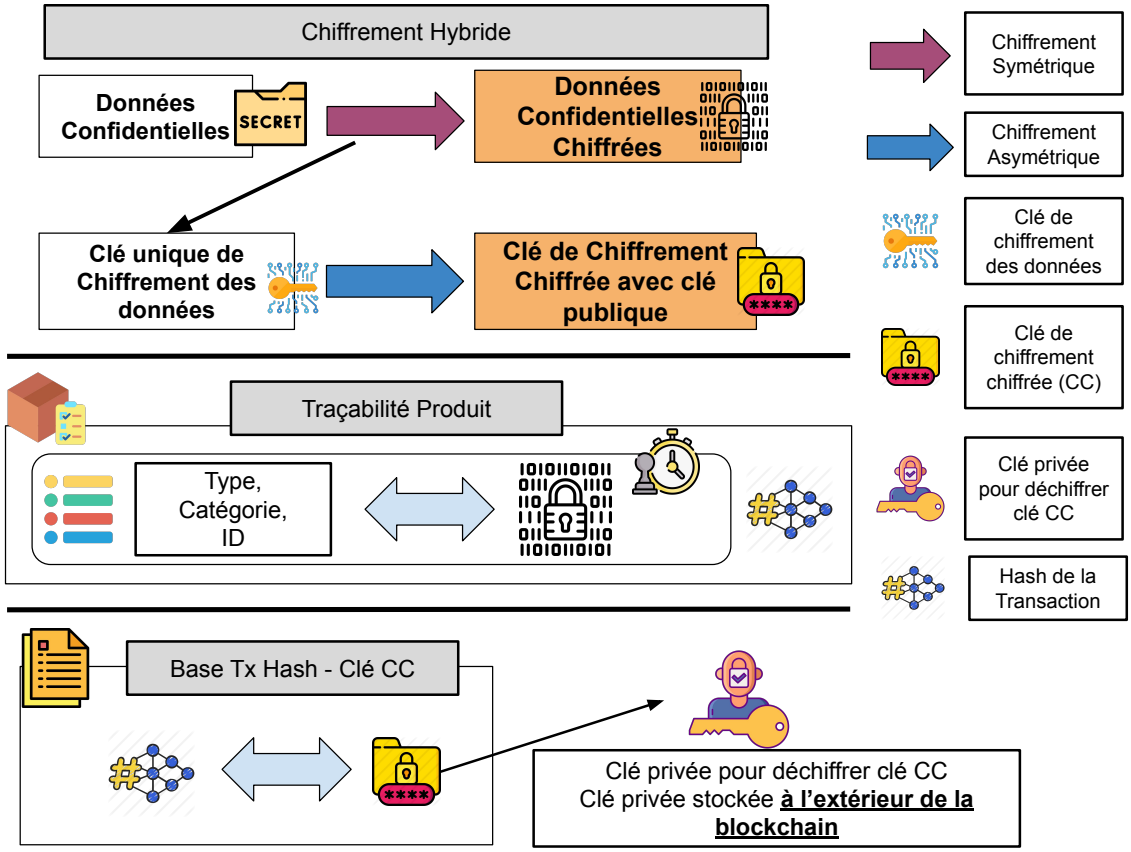


Figure 4.11 – BPCAT Implémentation de la confidentialité

```python
def encrypt_bytes(self, databytes):
    aes_key = generate_aes_key(AES_KEY_SIZE)
    rsa_pub = read_rsa_key("pub")
    encr_aes = encrypt_rsa_msg(rsa_pub, aes_key)
    encr_data, tag, nonce = encrypt_aes_msg(aes_key, databytes)
    item_secret = StreamItemSecret(
        secret=bytes_to_str(encr_aes, encode=True),
        tag=bytes_to_str(tag, encode=True),
        nonce=bytes_to_str(nonce, encode=True),
    )
    return encr_data, item_secret
```

Code 4.2 – Fonction de chiffrement des données dans l'implémentation de BPCAT

4.4.5 Gestion des fichiers

Les données de traçabilité des produits peuvent prendre plusieurs formes, notamment celle d'un fichier. Dans notre approche, les fichiers ne sont pas directement stockés dans la blockchain. En effet, l'entreprise est libre de choisir le mode de stockage qui répond le mieux à ses besoins. La Figure 4.12 décrit la manière dont les fichiers de traçabilité sont gérés et comment ils sont interconnectés aux données de traçabilité décrites dans la section précédente. Dans l'implémentation avec Multichain, les informations de fichier sont stockées dans un stream dédié. Les items prennent la forme de couple "Hash Fichier - Chemin du fichier" et pour chaque item, un hash de transaction est produit. Enfin, pour référencer le fichier dans la traçabilité du produit, on utilise le

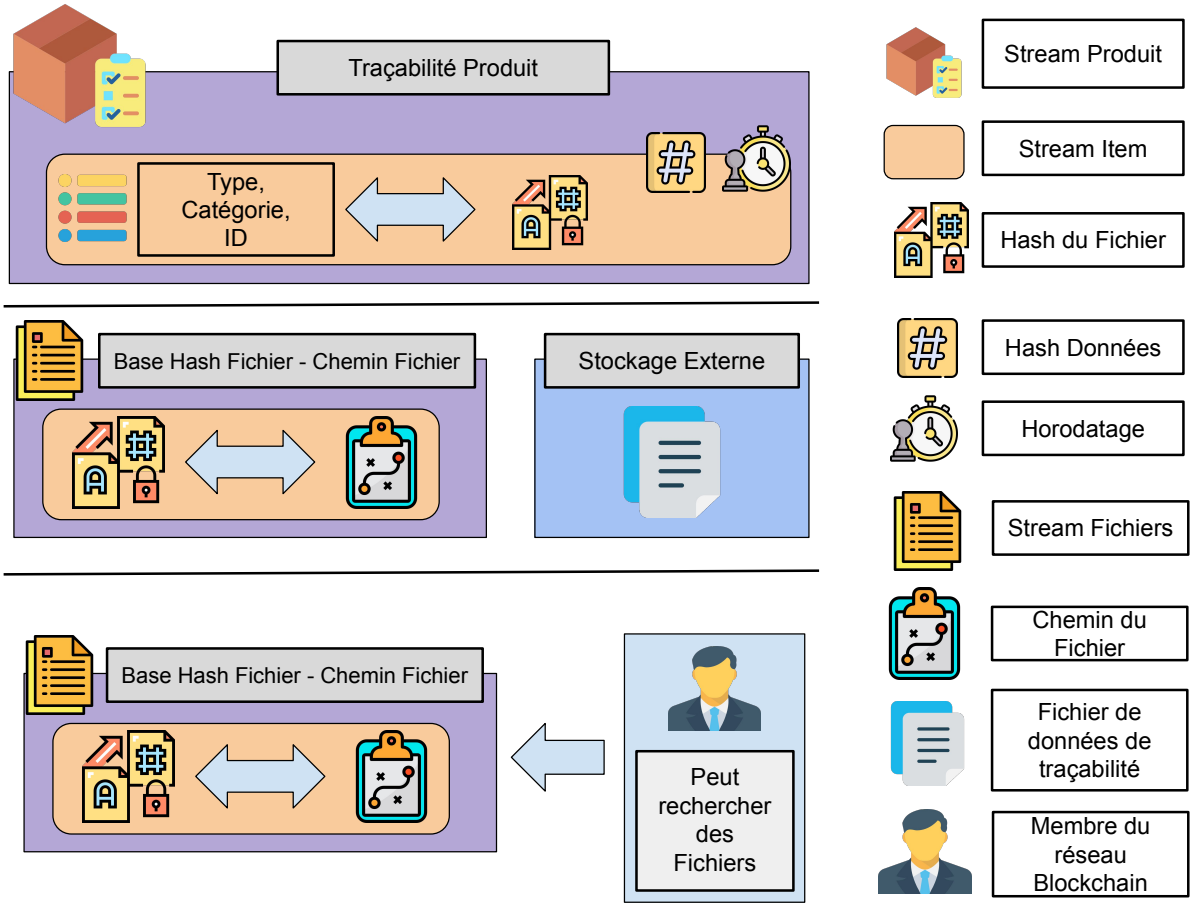


Figure 4.12 – BPCAT Implémentation de la gestion des fichiers

hash de la transaction précédemment mentionné.

L'extrait de Code 4.3 montre la fonction *save_file_bytes* qui permet d'illustrer cette gestion des fichiers. Cette dernière calcule le hash du fichier, écrit le fichier dans un répertoire donné qui peut représenter le stockage externe. Enfin, les informations du fichier telles que son type, sa taille, son chemin sont enregistrés dans la blockchain sous la forme d'une transaction.

```
def save_file_bytes(self, filepath, filename, save_dir,
        filetype):
        filehash = hash_file(filebytes)
        filepath = os.path.join(save_dir, filename)
        write_bytes(filepath, filebytes)
        return self.save_file_infos(
            StreamFile(
                filetype=filetype,
                filename=filename,
                filehash=filehash,
                filesize=len(filebytes),
                offchain=False,
            )
        )
```

Code 4.3 – Fonction permettant de sauvegarder un fichier avec Multichain

4.4.6 Signature des données

Multichain propose la possibilité pour un participant de la blockchain de signer des données ainsi que la possibilité pour lui de vérifier la validité de la signature. Dans notre approche, la signature représente l'approbation d'une donnée par un acteur blockchain différent de celui qui l'a publiée. La Figure 4.13 montre comment les signatures sont stockées dans Multichain et comment elles peuvent être sécurisées et fiabilisées.

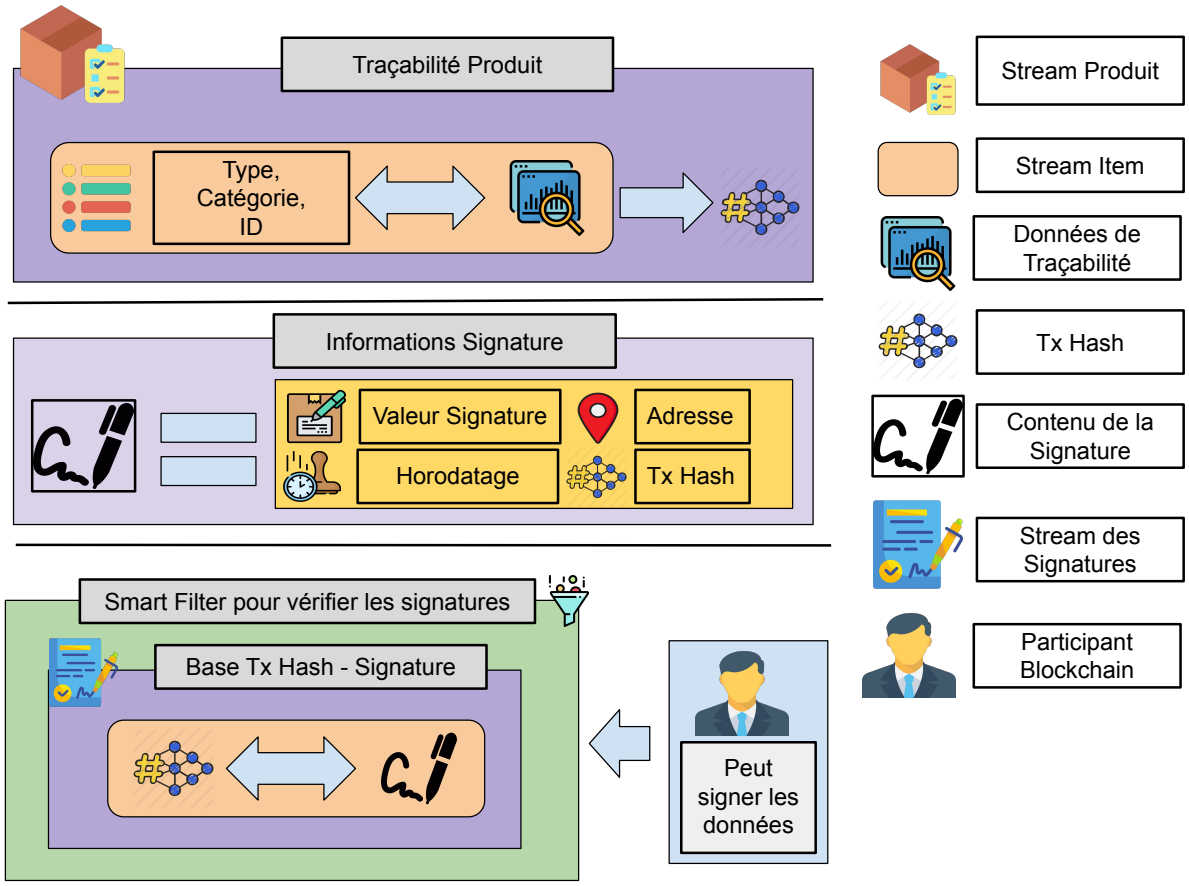


Figure 4.13 – BPCAT Implémentation de la signature

Les signatures sont stockées dans un stream dédié où chaque élément est un couple "Hash de la transaction - Signature" où "Hash de la transaction" correspond au hash de la transaction que l'on souhaite signer. Afin de sécuriser le processus et de garantir l'unicité de toutes les signatures, la génération de la signature est basée sur de multiples paramètres tels que l'horodatage, l'adresse du noeud effectuant la signature ainsi que le hash de la transaction. Ces 3 informations permettent donc de répondre aux questions "Qui ? Quand ? Quoi ?". La Figure 4.14 illustre la façon dont les données sont signées dans l'implémentation BPCAT avec Multichain.

De plus, pour s'assurer que seules les signatures valides soient ajoutées à la blockchain, un filtre de stream permet de vérifier la signature avant de l'ajouter, ce dernier est consultable dans le Code 4.4. Cela permet de maintenir la fiabilité à la fois de la blockchain principale et de l'application cliente de la blockchain.

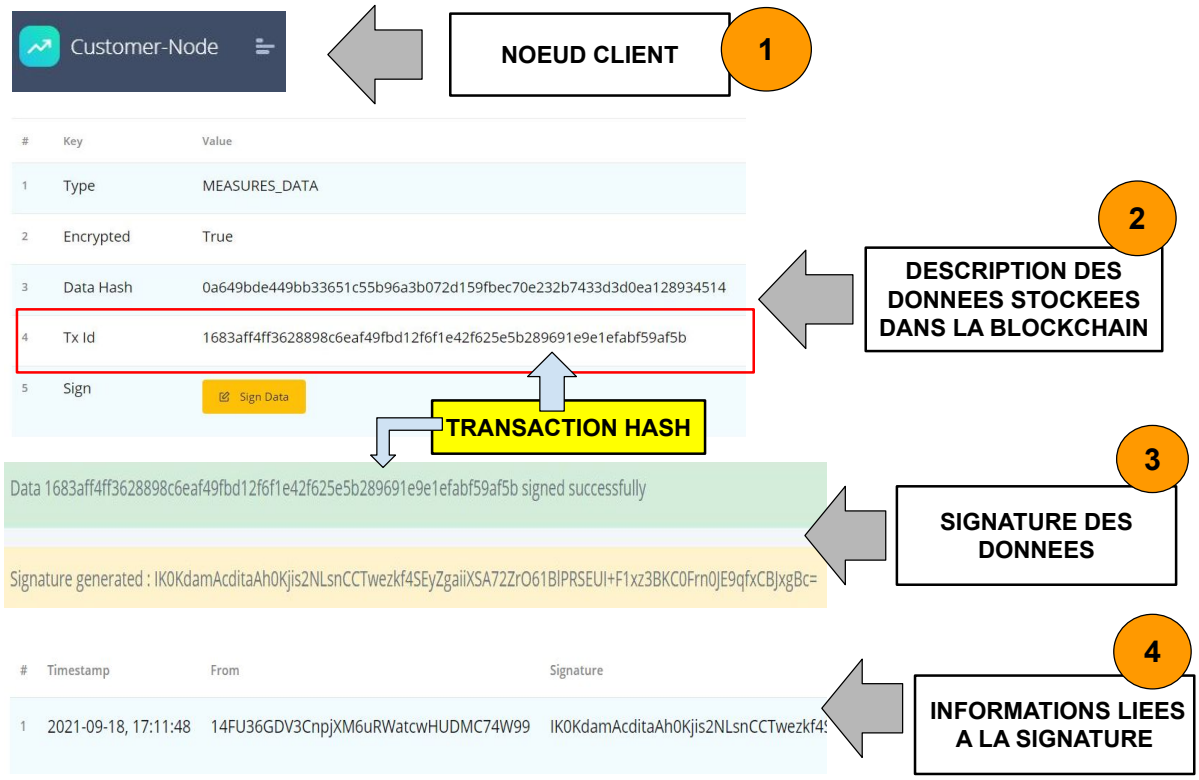


Figure 4.14 – BPCAT Exemple de signature des données

Code 4.4 – Smartfilter permettant de vérifier l'authenticité d'une signature

```
const SIGNATURES_STREAM = "SIGNATURES";

function filterstreamitem() {
  stream = getfilterstream();
  streamitem = getfilterstreamitem();

  if (stream.name == SIGNATURES_STREAM) {
    address = streamitem.data.json.body.from_addr;
    txid = streamitem.data.json.body.for_tx;
    signature = streamitem.data.json.body.signature;

    if (!verifymessage(address, signature, txid)) {
      return (
        "Signature " + signature +
        " is invalid from address " + address +
        " for tx " + txid
      );
    }
  }
}
```

4.4.7 Vérification de l'intégrité des données

Pour toutes les données de traçabilité, le hachage des données brutes est également stocké. Par conséquent, il est possible de garantir l'intégrité des données à partir du moment où elles ont été ajoutées à la blockchain et à chaque fois qu'elles sont surveillées. Dans notre approche, nous donnons à tout participant la possibilité de vérifier l'authenticité des données. Par exemple, concernant les données textuelles non confidentielles, il est possible de copier-coller les données et de recalculer le hash directement dans l'interface client du prototype. Si le hachage calculé correspond à celui stocké, l'intégrité des données est vérifiée. Concernant les données confidentielles, deux situations peuvent se présenter :

— Si l'auteur de la requête peut déchiffrer les données, rien ne diffère de l'exemple précédent ;
— Cependant, si l'auteur de la demande ne peut pas déchiffrer les données, il doit envoyer une "demande de déchiffrement" au propriétaire des données, très probablement la société propriétaire des données de traçabilité. Ensuite, si la demande est approuvée, il lui suffit de copier-coller les données déchiffrées et de comparer le hachage calculé avec celui stocké.

Le processus de vérification de l'intégrité des fichiers (voir la Figure 4.15) est assez similaire. Il nécessite le téléchargement du fichier à vérifier afin de calculer son hachage. Ensuite, ce hachage est comparé au hachage stocké. Si les deux hachages correspondent, l'intégrité du fichier est garantie.

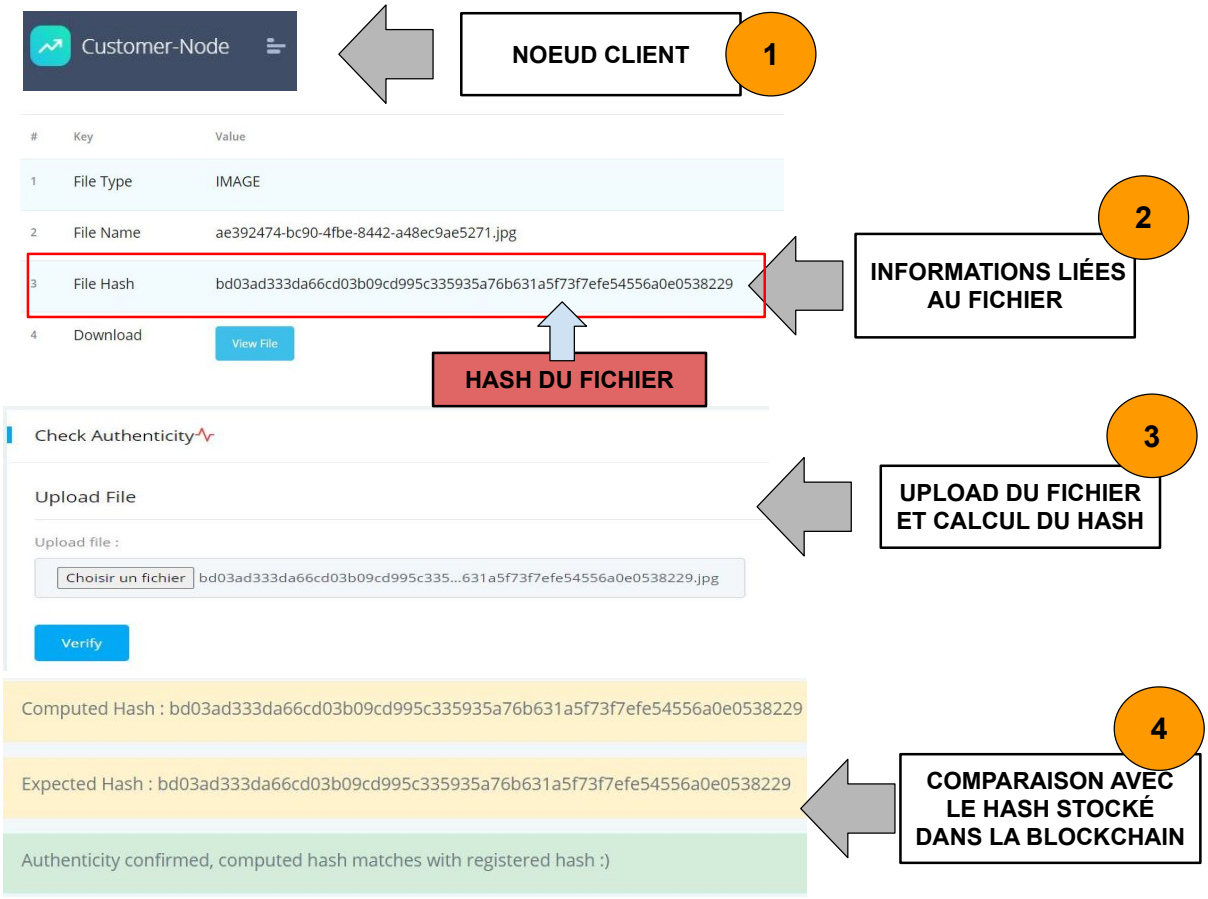


Figure 4.15 – BPCAT Exemple de vérification de l'intégrité des données

Le Code 4.5 montre un extrait de la fonction permettant de vérifier si deux fichiers sont similaires et donc de confirmer l'intégrité.

```python
def check_file_integrity(filetxid):
    stream_file = services.file.get_file_by_txid(filetxid)
    form = CheckFileHashForm()
    file = form.file.data
    expected_hash = stream_file.data.body.filehash
    computed_hash = hash_file(file.read())
    if computed_hash == expected_hash:
        flash_success("Files are similar")
    else:
        flash_error("Files are not similar")
```

Code 4.5 – Extrait de la fonction permettant de vérifier l'intégrité d'un fichier

4.5 Défis techniques à relever et solutions proposées

En complément de ses avantages, la technologie blockchain a cependant plusieurs défis techniques, les principaux étant la consommation énergétique liée au minage des blocs, la réduction du volume de stockage au niveau de la chaîne, mais également au niveau de chaque noeud. Dans cette section, nous allons proposer plusieurs solutions visant à les relever en nous appuyant sur différents scénarios testés grâce au prototype développé. Un système de « benchmark » a notamment été intégré afin de pouvoir effectuer des mesures et est décrit plus en détail dans l'annexe B.2.

4.5.1 Minage et consommation énergétique

La réduction de la consommation d'énergie est l'un des défis de la technologie blockchain en matière de big data. Le principal élément lié à la consommation d'énergie est l'algorithme de consensus utilisé dans le processus de minage. Dans le cas du bitcoin, par exemple, l'algorithme de consensus est la Preuve de travail (POW). Multichain est un « fork » du noyau bitcoin et a donc la capacité d'utiliser la preuve de travail (POW) ainsi que le schéma de validation à tour de rôle (Round-Robin). Le round-robin est le schéma par défaut dans Multichain, et il est basé sur la diversité minière où les nœuds possédant l'autorisation de miner ajouteront des blocs soit de manière aléatoire ou bien l'un après l'autre en fonction du paramètre de diversité. Il n'y a donc pas de concurrence dans le cas de cet algorithme de consensus. Dans le cas de la preuve de travail, les noeuds sont mis en concurrence afin de résoudre le plus rapidement possible un puzzle mathématique ce qui explique la forte consommation d'énergie. Afin d'illustrer l'impact de l'algorithme de consensus sur la consommation d'énergie, une comparaison du taux de CPU utilisé par Multichain lors de l'utilisation de la preuve de travail et d'un schéma à tour de rôle est proposée Figure 4.16.

Pour chaque noeud, deux courbes sont disponibles : POW et Non-Pow (Round Robin). On peut observer à côté des courbes "Non-Pow" le bénéfice par rapport à la courbe "POW" associée à chaque noeud. Par exemple, utiliser un schéma à tour de rôle fait une différence de -59% pour le validateur, -83% pour le fournisseur et -82% pour le client. Une discussion peut donc être engagée concernant la consommation d'énergie. En effet, si une usine considère qu'une blockchain est nécessaire à sa politique de traçabilité et qu'elle justifie la consommation d'énergie supplémentaire associée, certains de ses fournisseurs et clients peuvent ne pas partager son avis. Actuellement, BPCAT utilise le schéma de tour de rôle afin d'économiser de l'énergie pour chaque acteur participant, mais il pourrait également être décidé que seule l'usine (noeud valideur) effectue les opérations de minage afin d'épargner aux fournisseurs et aux clients tout coût énergétique supplémentaire dû au minage.

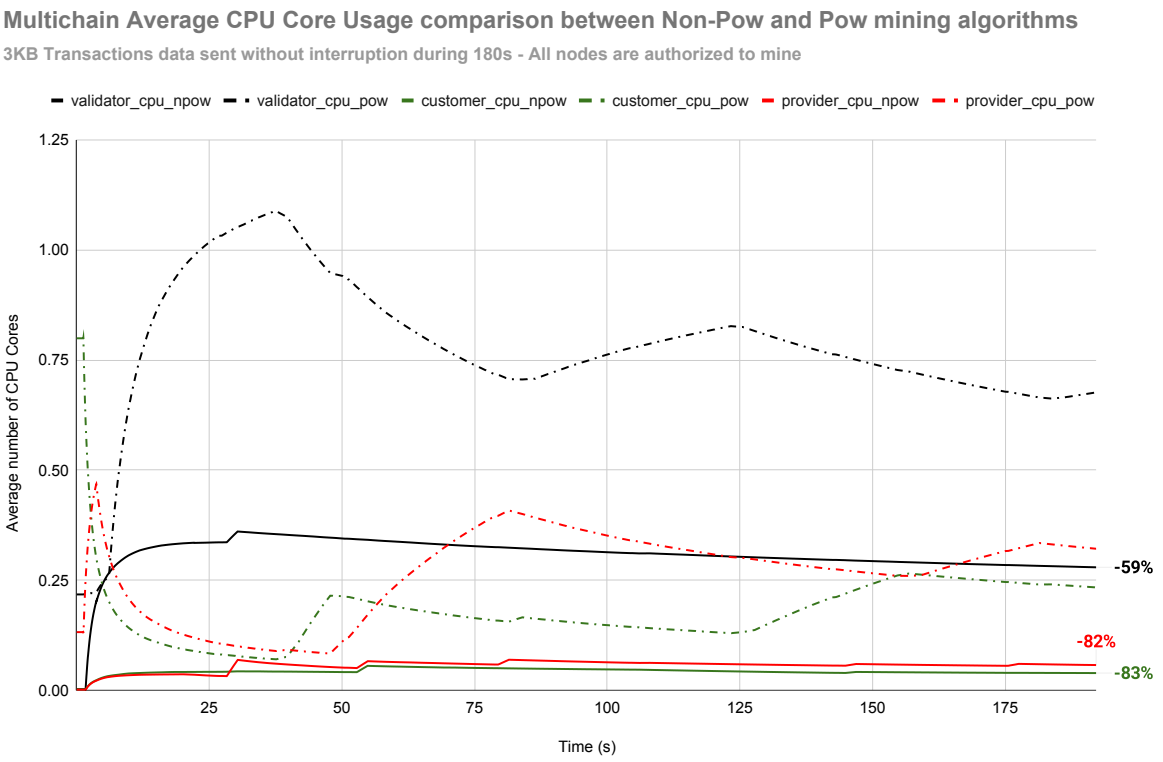


Figure 4.16 – Comparison de la consommation CPU entre 2 algorithmes de minage : la preuve de travail (POW) et le round robin (Non-POW)

4.5.2 Optimisation du volume de stockage

Un autre défi lié à la technologie blockchain est le volume de stockage. La blockchain étant un système distribué, cela implique que tous les nœuds participants doivent stocker une copie de la blockchain, rendant ainsi la solution moins efficace qu'un système centralisé dans lequel les données ne seraient stockées qu'une seule fois (dans l'infrastructure de l'usine par exemple). Les sous-sections suivantes décrivent une manière de résoudre ce problème à travers deux paramètres : la réduction de la taille des données elles-mêmes et la réduction du volume stocké sur chaque noeud.

Optimisation à l'échelle des données

Multichain offre la possibilité de stocker des données en tant qu'éléments offchain (hors-chaîne) ce qui signifie que les données sont stockées en dehors des blocs et référencées par un hash contrairement au stockage onchain où les données sont directement stockées dans les blocs. Ce choix n'est pas neutre comme on peut l'observer sur la Figure 4.18.

Le stockage de plusieurs éléments par transaction permet de réduire le volume de stockage nécessaire, tandis que le stockage onchain (en chaîne) nécessite presque le double du volume par rapport au stockage hors chaîne. Lorsque les données sont soumises hors chaîne, elles sont divisées en interne en morceaux de taille fixe et pour chaque morceau, un hash est calculé. Ce processus est similaire au fonctionnement d'IPFS [76] qui est un système de stockage de fichiers distribué souvent associés avec les sytèmes blockchain pour sa capacité à gérer les fichiers. Cependant, contrairement à IPFS, Multichain inclut nativement cette fonctionnalité.

La caractéristique la plus intéressante de ce processus est que : lorsqu'un morceau avec le même hash existe déjà, il n'est pas stocké une seconde fois et à la place, une référence au morceau existant est créée à la place. Par conséquent, l'espace de stockage peut être très fortement réduit concernant les données avec un taux de

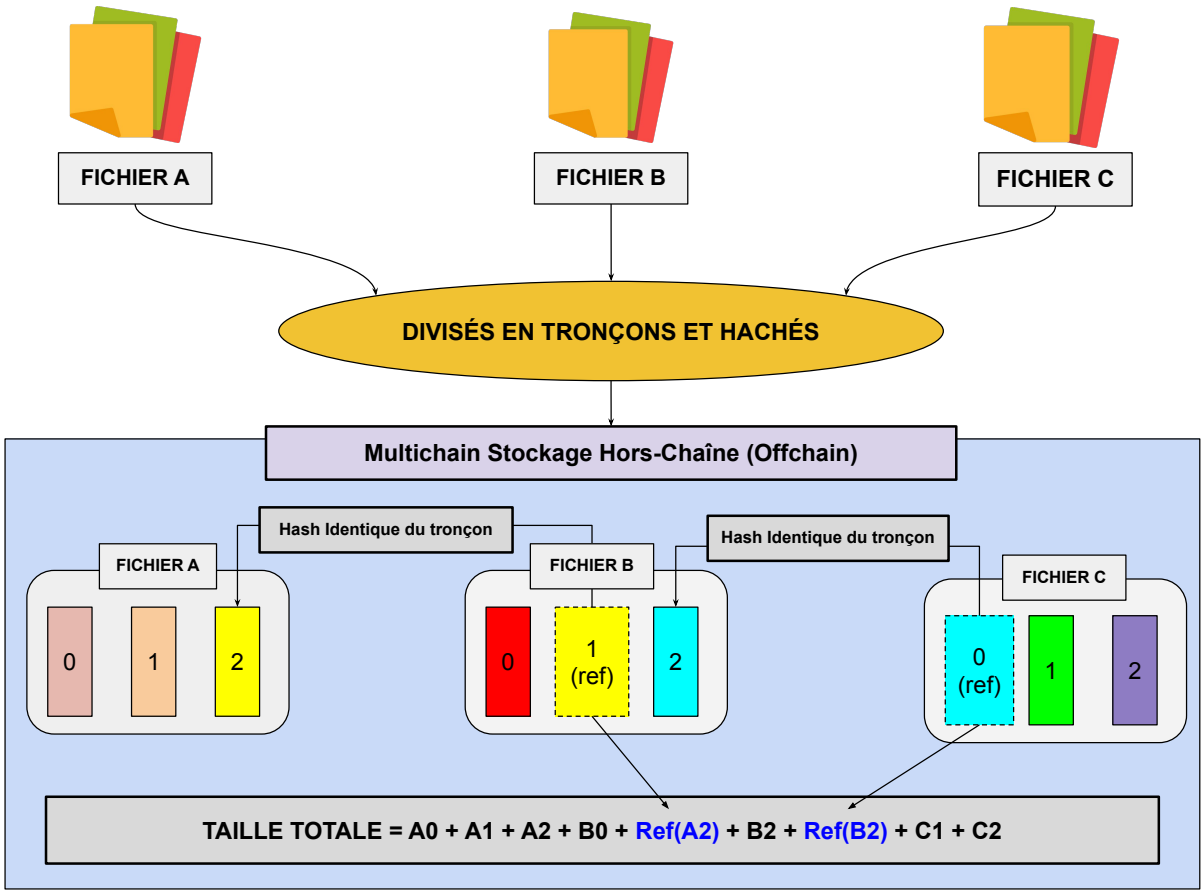


Figure 4.17 – Description du fonctionnement du mode de stockage offchain de Multichain

similitude élevé puisque les données dupliquées ne sont pas stockées plusieurs fois. Afin d'évaluer la mesure dans laquelle cette fonctionnalité peut permettre de faire des économies de stockage, nous avons réalisé une expérience sur un volume important de fichiers présentant différents taux de similitude puis nous les avons soumis à Multichain en mode hors-chaîne.

Le contexte est le suivant : un millier d'images ayant pour taille un mégaoctet (1Mo) sont générées avec divers taux de différence allant de 5% à 100%. Le Code 4.6 montre la fonction permettant de générer à partir d'un tableau de pixels original un autre tableau différent de N% où N est le taux de variance dans cette fonction.

```
1 def get_alternate_pixels(orig_pixels, variant_rate=5):
2     alternate = orig_pixels.copy()
3     nb_update = round(len(alternate) * variant_rate / 100)
4     for idx in range(nb_update):
5         rand_pixel = services.random.rand_int(min_value=0,
6             max_value=255)
7         alternate[idx] = rand_pixel
8     return alternate
```

Code 4.6 – Fonction de création d'une image différente de N% d'une image originale

Le résultat de cette expérience est illustré par la Figure 4.19 où l'optimisation du volume de stockage en fonction du taux de variance entre les fichiers est présentée.

La courbe intitulée "Volume de référence total" indique le volume de stockage total nécessaire pour simplement stocker les fichiers sur un disque dur. Les autres courbes représentent sept situations où la variance entre

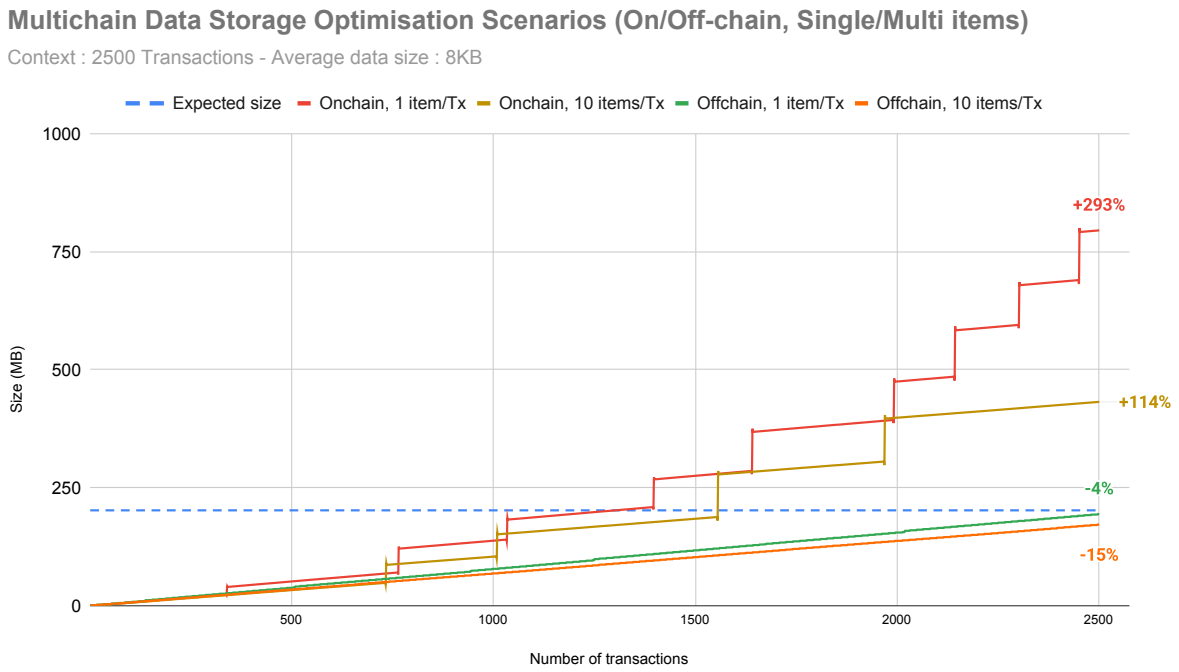


Figure 4.18 – Optimisation du volume de stockage dans Multichain en fonction du mode de stockage (onchain / offchain) et du nombre d'items par transaction

les fichiers varie de 5% à 100%. Une variance de 5% signifie que les 1000 fichiers ne sont différents que de 5% les uns des autres, et une variance de 100% signifie que les fichiers sont complètement différents les uns des autres. Les économies de volume de stockage dans chaque situation selon la référence de volume total sont reportées en pourcentage de réduction sur la courbe correspondante. Cette économie est comprise entre -90% lorsque les fichiers n'ont qu'un écart de 5% et +1% lorsque les fichiers sont complètement différents (un écart de 100%). Dans cette situation, le volume est supérieur à la référence de volume total en raison des informations supplémentaires introduites par Multichain dans la gestion du stockage des fichiers.

Le contexte d'une usine de fabrication implique que les mêmes processus se répètent indéfiniment pour les mêmes machines et les mêmes produits. Par conséquent, la probabilité de trouver des similitudes entre les données générées par les processus de traçabilité centrés sur les produits est très élevée. Par conséquent, la fonctionnalité qui consiste à découper les données en morceaux et à ne pas sauvegarder deux fois le même morceau pourrait permettre de réduire le volume total nécessaire au stockage des données de traçabilité.

Optimisation à l'échelle des noeuds

L'autre problème est le volume de données stockées dans chaque noeud. L'idée que la blockchain est un registre distribué pourrait suggérer que chaque nœud doit stocker exactement le même volume de données dans sa copie de la blockchain, ce qui n'est pas tout à fait vrai. Comme il a été mentionné précédemment, Multichain peut enregistrer les données dans des flux appelés "streams", permettant ainsi à la blockchain d'être utilisée comme une base de données à usage général fournissant uniquement l'horodatage, la notarisation et l'immuabilité. Par défaut, chaque nœud doit stocker les blocs et les hash de transactions. Cependant, les données stockées dans les flux impliquent une gestion spécifique du stockage du fait de la fonctionnalité d'abonnement. S'abonner à un flux implique l'indexation et le téléchargement de l'ensemble du contenu du flux sur le noeud, y compris les données hors chaîne. Cela signifie que chaque acteur dispose d'une certaine flexibilité lors du choix des données à stocker dans son stockage local, ce qui peut permettre d'économiser un certain volume non négligeable.

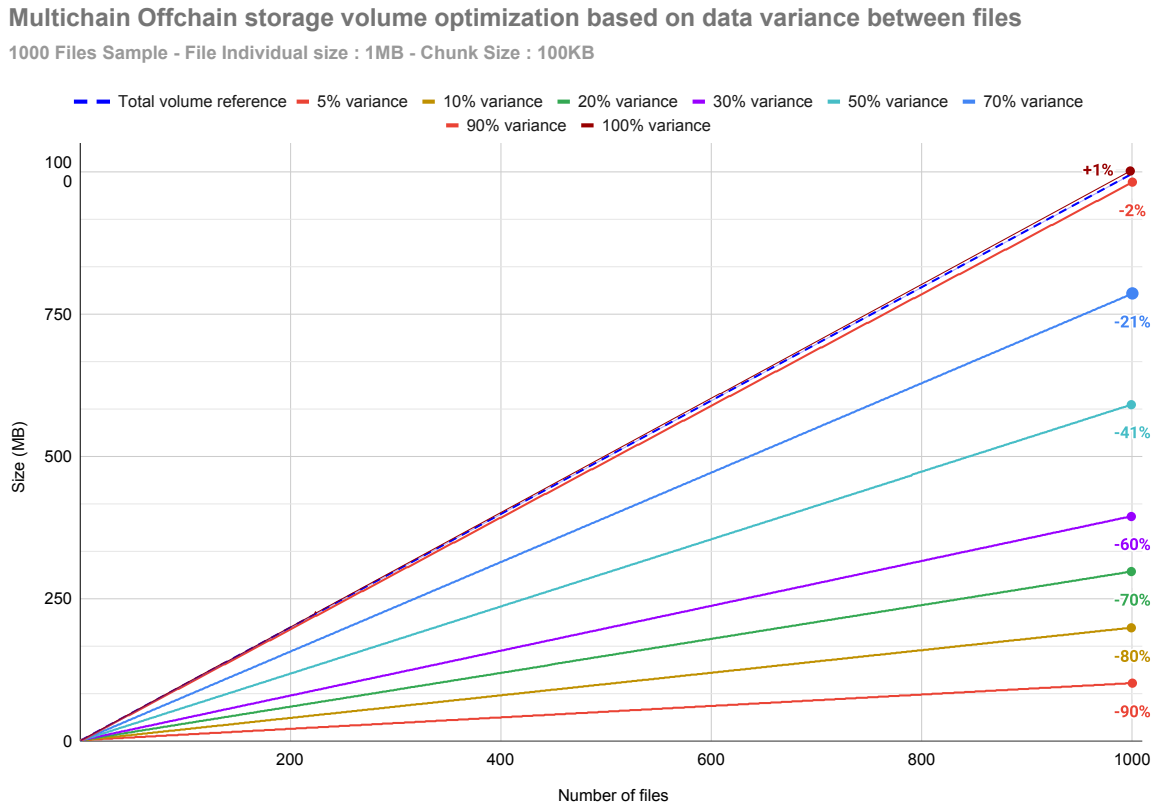


Figure 4.19 – Optimisation du volume de stockage dans Multichain en fonction du taux de variance entre les fichiers (mode offchain)

Pour illustrer l'étendue de cette flexibilité, nous avons donc réalisé une expérience visant à montrer la possibilité d'optimiser le volume de stockage dans l'implémentation BPCAT avec Multichain à partir de l'abonnement aux flux.

Le contexte peut être décrit comme suit : trois flux spécifiques ont été créés pour chaque joueur, et chacun s'est abonné à son propre flux (Valideur, Client et Fournisseur). Un quatrième noeud appelé "Observateur" (Watcher) a été créé, sans abonnement à aucun flux afin de montrer le volume de stockage minimal nécessaire à tout noeud participant au réseau blockchain.

Dix mille transactions de fichiers de 100 Ko ont été envoyées au hasard à ces flux comme le montre le Code 4.7.

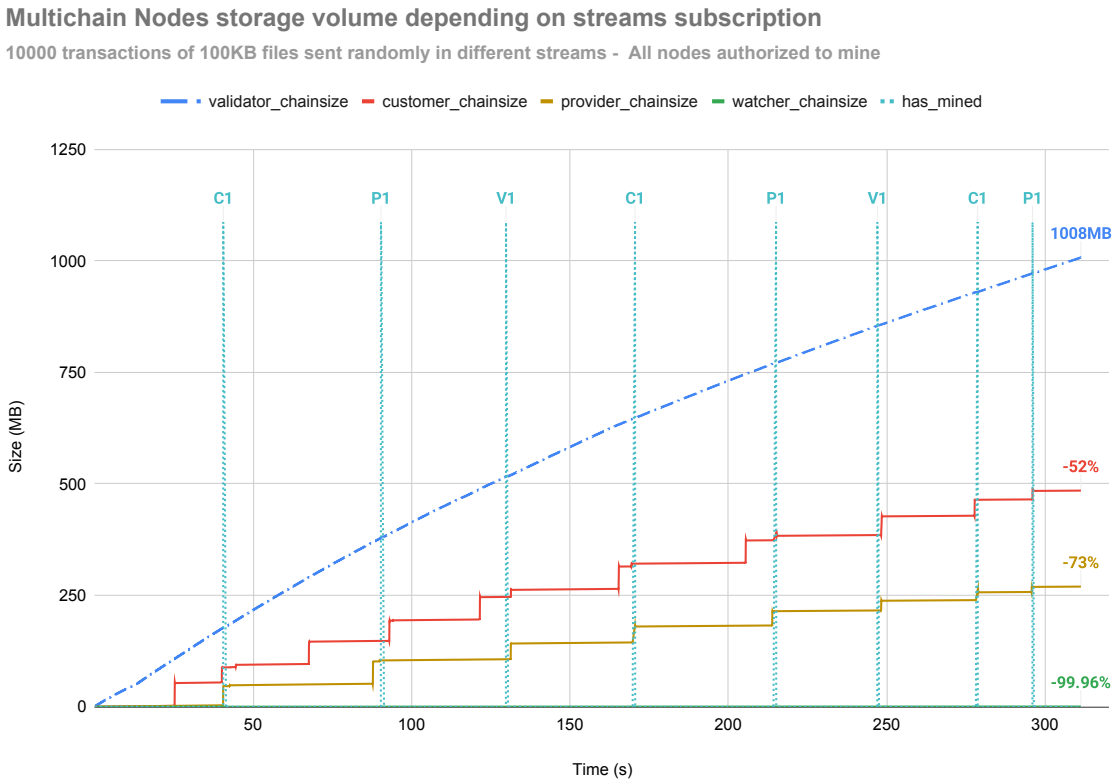


Figure 4.20 – Optimisation du volume de stockage au niveau des noeuds en fonction de l'abonnement aux streams Multichain

```python
def start_file_simu_for_stream():
    # Liste des streams possibles
    for_streams = ["VALIDATOR", "CUSTOMER", "PROVIDER"]
    folderpath = os.path.join(DATAFILES_DIR, filetype)
    filelist = os.listdir(folderpath)
    for i in range(nbfiles):
        filepath = os.path.join(folderpath, filelist[i])
        # Sélection aléatoire d'un stream
        for_stream = for_streams[
            round(random.randint(0, 100) * (len(for_streams) - 1) / 100)
        ]
        bench.measure_perf(
            func=services.publish.prepare_file_request,
            filepath=filepath,
            encrypted=encrypted,
            publish_now=True,
            for_stream=for_stream,
        )
```

Code 4.7 – Extrait de la fonction permettant de soumettre une transaction à un stream aléatoire

Sur la Figure 4.20 synthétisant les résultats, la première courbe montre que le volume total stocké par le noeud Valideur est le plus élevé, ce qui s'explique par le fait que c'est lui qui a soumis toutes les transactions (les éditeurs d'une transaction n'ont d'autre choix que de stocker leurs propres données). Cependant, le Client et le Fournisseur observent une différence de volume par rapport au Valideur, respectivement de -52% et

-73%. Le noeud Observateur a le volume le plus faible (-99,96% en comparaison avec le volume de stockage du valideur).

Il apparaît donc qu'en définissant clairement les relations entre les données et les acteurs, il est possible de les organiser efficacement dans les flux afin d'économiser un important volume de stockage sur les noeuds.

4.6 Synthèse des points forts et points faibles de la contribution

Dans ce chapitre, nous avons vu de quelle manière la technologie blockchain pouvait être intégrée dans la traçabilité d'une usine 4.0. Tout d'abord, une présentation du paradigme de la blockchain a permis de mettre en avant certaines fonctionnalités clés telles que : l'élimination du besoin d'un tiers de confiance pour effectuer les transactions, la transparence mais surtout l'immuabilité grâce au registre partagé qui fait que les transactions ne peuvent être modifiées ou supprimées. De plus, la blockchain ajoute un aspect qualitatif aux données grâce au fait qu'elles sont complètes, datées et leur disponibilité est garantie en raison de l'aspect distribué de l'architecture. Le réseau distribué permet également de grandement réduire les risques de perte de données ou encore leur indisponibité en raison d'une défaillance.

Dans le domaine de la traçabilité, la blockchain peut aider au partage sécurisé et transparent des données entre tous les acteurs de la chaîne d'approvisionnement. Notre approche BPCAT (Figure 4.21) ainsi que le prototype réalisé avec Multichain prouve qu'il est possible de garantir différents aspects critiques relatifs aux données de traçabilité à savoir :

— la sécurité grâce à l'implémentation en blockchain de consortium (ou permissionnée)
— l'intégrité par l'intermédiaire des hash des données stockés et la possibilité de les vérifier
— la confidentialité par le chiffrement des données confidentielles et le stockage du hash de la donnée originelle
— la transparence grâce à la visibilité des transactions par tous les acteurs (mais pas forcément des données)
— la non-répudiation grâce aux signatures des transactions par les acteurs concernés

Tous ces éléments apportent aux parties prenantes la confiance dans le système d'information car ils sont intégrés dans un registre de transaction partagé et inviolable qu'est la blockchain. De plus, les aspects garantis sont vérifiables a posteriori tels que les hash des données pouvant être recalculés pour confirmer l'intégrité ou encore les signatures des transactions permettant ainsi de valider leur authenticité. Ce système peut donc assurer la traçabilité, réduire les risques, faciliter la gestion, aider à la résolution de litiges entre les acteurs et améliorer la pérennité et la flexibilité du système de traçabilité de l'usine.

Cependant, la blockchain possède également quelques inconvénients qui sont fréquemment cités. Le premier d'entre eux est la consommation énergétique que nous avons également abordé et qui est dû en majeure partie à l'algorithme de consensus utilisé, la POW étant un des plus gourmands en énergie. Nous avons à cette occasion montré que le recours à d'autres types de consensus peut contribuer à réduire cet impact énergétique.

Le temps nécessaire à la validation des transactions est également cité [118] notamment dans les réseaux blockchain publics de taille très importante tels que le Bitcoin cependant ce n'est pas le cas dans notre approche car nous utilisons une blockchain permissionnée où le nombre de participants est limité.

Aussi, la redondance des données sur chacun des noeuds peut être source de consommation importante d'espace de stockage c'est pourquoi nous avons montré dans l'implémentation avec Multichain qu'il était possible de limiter cette consommation en répartissant les données dans différents streams et que chaque acteur pouvait choisir de s'abonner aux streams qu'ils désiraient afin de ne stocker en local que les données qui l'intéressent.

Bien que la technologie blockchain n'en soit qu'à ses balbutiements, bon nombre de ses inconvénients identifiés sont liés à des erreurs dans les choix d'implémentations (types de blockchain, algorithme de

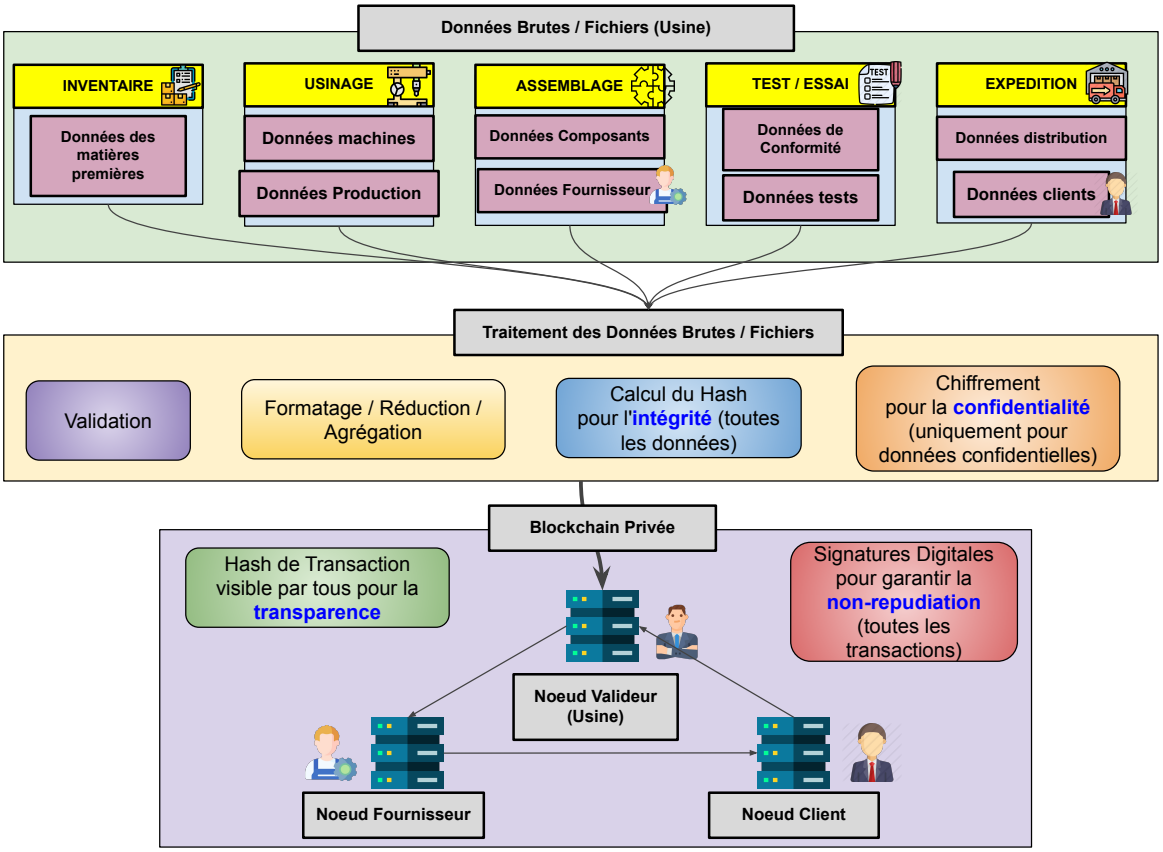
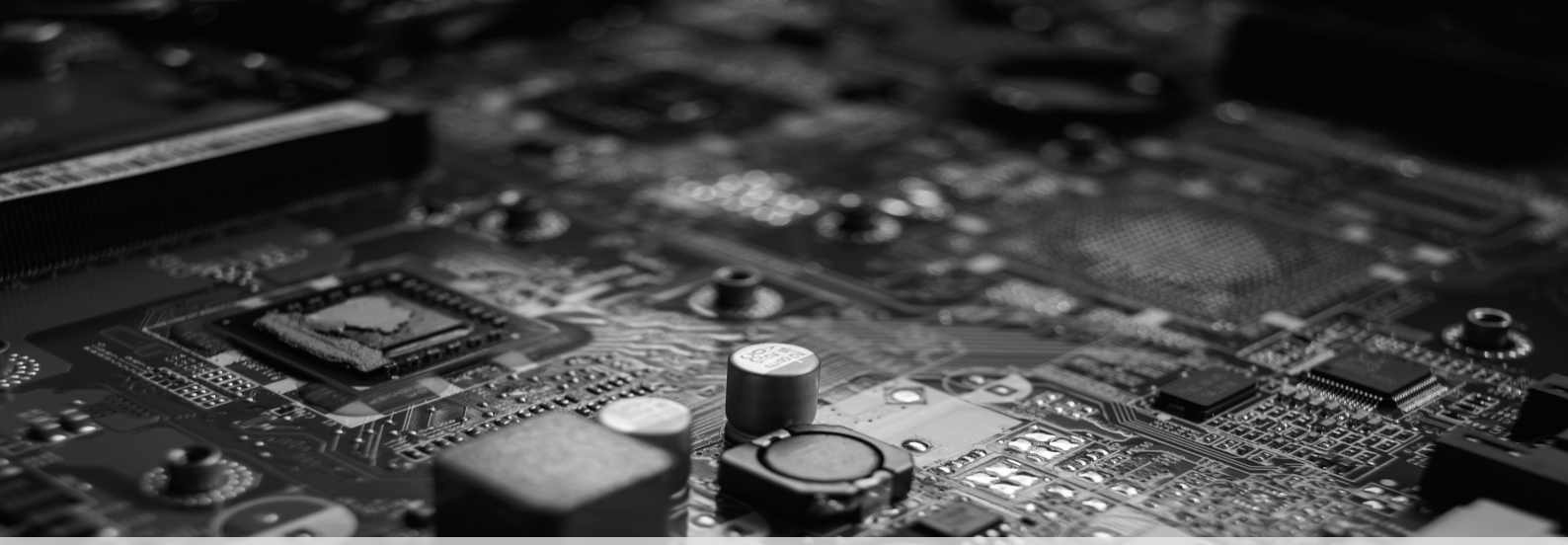


Figure 4.21 – Synthèse de l'approche BPCAT

consensus, paramétrage de la solution choisie). Le développement des blockchains autorisées ouvre de nouvelles opportunités pour l'adoption de cette technologie dans les contextes B2B comme la chaîne d'approvisionnement de la traçabilité. L'adoption de cette technologie s'accélère dans les domaines où la traçabilité et la transparence sont des exigences légales et réclamées par les utilisateurs finaux comme par exemple dans les chaînes d'approvisionnement pharmaceutiques [119] [120] [121] et alimentaires [122] [123] [124].

Enfin, la blockchain constitue un changement de paradigme complet pour l'industrie qui se voit passer d'un système de traçabilité centralisé interne à l'usine à un réseau décentralisé où peuvent être intégrés les acteurs concernés tels que les clients et les fournisseurs. Cela peut entraîner des problèmes d'adoption et d'intégration de cette technologie dans les écosystèmes existants. Compte tenu de ces éléments, il apparaît donc nécessaire de disposer d'un moyen d'évaluation des impacts de la blockchain lors de son intégration dans la traçabilité produit de l'usine 4.0.

Modélisation et Simulation



5 Modélisation et évaluation par simulation pour l'usine 4.0

La traçabilité est transversale aux différents piliers de l'industrie 4.0, car elle fournit des informations essentielles à la gestion de l'exploitation de l'usine et facilite les audits post-production. Récemment, la technologie blockchain a été largement étudiée comme une solution pour améliorer l'immuabilité et la transparence des systèmes de traçabilité industrielle, dans le but d'accroître la confiance entre les différentes parties prenantes. Cependant, les évaluations proposées par la plupart des travaux portent sur les performances de la blockchain elle-même plutôt que sur son impact sur le système industriel. Dans ce chapitre, nous proposons une modélisation à événements discrets ainsi qu'un framework collaboratif qui peut être adapté à n'importe quelle usine de production industrielle.

À travers une étude de cas, nous montrons comment il pourrait être utilisé pour analyser l'impact de notre proposition de traçabilité basée sur la blockchain BPCAT, en termes de différentes métriques personnalisées, telles que les besoins de stockage, la consommation d'énergie ou encore l'impact environnemental lié à l'adoption d'une telle solution.

5.1 Fondements théoriques de la simulation

Cette première partie sera dédiée à la présentation des concepts de base liés à la modélisation et à la simulation.

5.1.1 Définitions

Système Un système est caractérisé par le fait que l'on sait distinguer ce qui lui appartient de ce qui ne lui appartient pas. Il est supposé contrôlable et/ou observable. Des variables, générées par l'environnement, agissent sur le comportement du système qui, à son tour, réagit sur cet environnement.

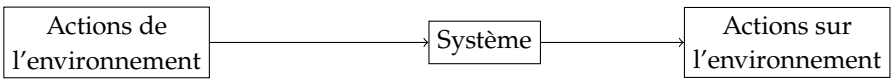


Schéma d'expérimentation Une expérimentation est un processus par lequel on récolte des données sur un système en agissant sur ses entrées. Le schéma d'expérimentation définit un ensemble limité de circonstances sous lesquelles l'expérimentation est conduite :

— Variables observées
— Séquencement des entrées
— Conditions initiales
— Conditions d'arrêt
— Collecte et codage des données

Modèle Un modèle M d'un système S pour une expérimentation E correspond à toute chose à laquelle on peut appliquer E pour répondre à des questions concernant S.

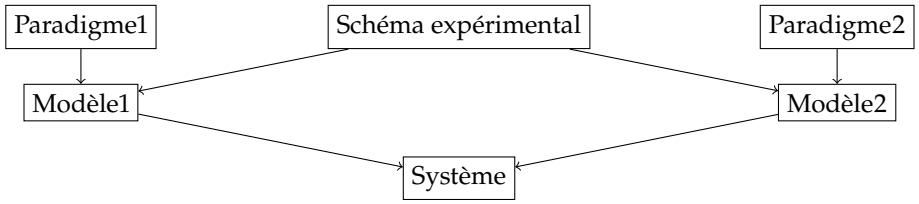
Il s'agit donc d'une représentation simplifiée et observable du comportement de la structure d'un système afin de résoudre un problème d'analyse ou de conception.

On distingue deux types de modèle :

— Modèle prédictif où on cherche à prédire une situation ou un état du sytème
— Modèle descriptif où on capitalise la connaissance au sein d'un modèle.

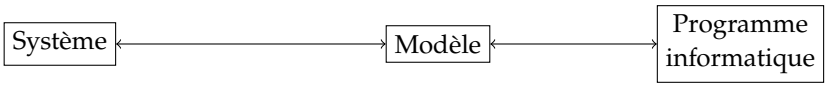
Modélisation Processus par lequel on organise les connaissances portant sur un système donné [125].

Paradigme Un paradigme est un ensemble de lois et de concepts permettant de définir une collection de modèles.



Simulation La simulation est la reproduction du comportement d'un système réel en s'appuyant sur un modèle. Elle a pour objet d'observer le comportement en fonction du temps d'un modèle d'un système.

En informatique, il s'agit de la traduction en programme informatique du comportement dynamique des modèles. Son exécution se fait en temps fini.



5.1.2 Classes de modèles

Les classes de modèles se différencient principalement en fonction de deux paramètres : le temps et le type des variables d'états.

Temps continu Dans un modèle à temps continu, le temps est spécifié comme évoluant de manière continue, il s'agit d'un nombre réel.

Temps discret Dans un modèle à temps discret, le temps avance par saut d'une valeur entière à une autre, il s'agit d'un nombre entier.

État discret Dans un modèle à états discrets, les variables prennent leurs valeurs dans un ensemble discret.

État continu Dans un modèle continu, les variables sont des nombres réels.

5.1.3 Hiérarchie de spécification

En modélisation, une approche hiérarchique consiste à catégoriser les informations connues dans différents niveaux. Le Tableau 5.1 représente une synthèse de ces différents niveaux ainsi que des informations connues à chacun d'entre eux.

Tableau 5.1 – Hiérarchie de spécification dans la théorie de la modélisation et de la simulation

Niveau	*Nom*	*Éléments connus à ce niveau*
0	Cadre d'observation	Variables à mesurer et comment les observer sur une base de temps
1	Relation d'entrée - sortie	Données indexées sur le temps ainsi que les ports d'entrée - sortie
2	Fonction d'entrée - sortie	Connaissance de l'état initial
3	Transitions d'état	Manière dont les entrées affectent les états et commment les états affectent les sorties
4	Composants couplés	Façon dont les composants de niveau inférieur sont couplés

5.1.4 Formalismes de modélisation

Dans le processus de modélisation, le choix d'un formalisme représente une hypothèse forte. Il est nécessaire de tenir compte des propriétés du système dans ce choix. Le Tableau 5.2 présente des exemples de formalisme pouvant être utilisés en fonction des caractéristiques d'un modèle.

Tableau 5.2 – Aperçu des types de formalisme en fonction des caractéristiques du modèle

Caractéristiques du modèle			
Changement d'état	*Temps*	*Espace*	*Formalisme*
Continu	Continu	Absent	Equations différentielles ordinaires
Continu	Continu	Continu, Discret	Equations différentielles ordinaires
Continu	Discret	Absent	Équations aux différences
Continu	Discret	Continu, Discret	Équations aux différences et spatialisées
Discret	Continu	Absent, Continu et Discret	Modèles à évènements discrets
Discret	Discret	Absent	Automates à états finis
Discret	Discret	Continu	Modèles à temps discret
Discret	Discret	Discret	Automates cellulaires

Dans le cadre de l'évaluation pour la thèse, nous allons nous intéresser au formalisme des modèles à évènements discrets.

DEVS, Discrete Event System Specification

DEVS est un formalisme de modélisation initié par B.P Zeigler en 1976 [125]. Il s'agit d'un formalisme systémique :

— évènements discrets
— ensembles, états et fonctions de transition d'état
— possédant une approche modulaire et hiérarchique

5.2 Etat de l'art sur l'évaluation des systèmes usines et de la blockchain

La traçabilité ne consiste pas seulement à enregistrer des données. D'autres aspects sont impliqués dans la traçabilité, tels que la confidentialité et un stockage efficace, ainsi que la transparence vis-à-vis des autres acteurs de la chaîne d'approvisionnement. Outre les problèmes de sécurité[104], plusieurs articles de la littérature ont étudié l'utilisation de la technologie blockchain pour répondre à ces aspects [126, 127]. Dans ce manuscrit, nous nous concentrons sur la principale méthode d'évaluation utilisée dans certains d'entre eux. En conséquence, une enquête sur les solutions blockchain pour la fabrication durable et l'autonomisation de la gestion du cycle de vie des produits est proposée dans [78]. Diverses mesures liées à la confiance dans l'utilisation des solutions de blockchain dans les usines de fabrication sont introduites, concernant la transparence, la décision décentralisée, la réputation et les relations avec la clientèle. Du point de vue de l'usine, la blockchain pourrait être utilisée comme un moyen de piloter des systèmes déjà existants pour les ateliers, tels que la ERP et le MES.

Du point de vue de la gestion des produits, la blockchain pourrait offrir une base de données unifiée pour partager les informations sur les produits. Les propositions présentées dans divers secteurs manufacturiers [79], dont le textile [80], soulignent l'importance des solutions pratiques de blockchain pour la traçabilité dans l'industrie 4.0. Afin de fournir des informations approfondies sur la technologie blockchain, [95] propose un examen de la mise en œuvre de solutions basées sur la blockchain pour diverses applications dans lesquelles la sécurité reste primordiale. Il décrit la manière dont la technologie blockchain pourrait résoudre les problèmes rencontrés par les systèmes traditionnels en matière de transparence, de centralisation, d'évolutivité, de confiance et de sécurité. Le système de fabrication intelligent sécurisé par blockchain présenté par [110] utilise la blockchain pour résoudre certains problèmes courants dans les systèmes de fabrication, tels que la traçabilité des opérations, la confidentialité et la confiance. De plus, cela permet d'éviter que la défaillance des nœuds clés ne se produise éventuellement dans les plates-formes centralisées. Le document présente plusieurs paramètres liés à la blockchain dans la fabrication, à savoir la transparence de la provenance des données, la flexibilité du système, la durabilité du système et les économies de coûts. Cependant, ces enquêtes ne fournissent aucune évaluation quantitative.

Afin de tracer les pièces de rechange agrégées dans un produit manufacturé, [76] présente une approche basée sur la blockchain qui répond à certains problèmes traditionnels rencontrés par la gestion de la chaîne d'approvisionnement, tels que la transparence et la surveillance active des opérations qui affectent la conformité et la sécurité des composants des pièces de rechange. La proposition comprend un stockage de fichiers décentralisé basé sur le système de fichiers interplanétaire (IPFS) pour les données des pièces de rechange, les algorithmes des smart contracts, la sécurité et l'analyse des coûts. La méthode d'évaluation consiste principalement à lister les avantages qualitatifs de la blockchain afin de démontrer la fiabilité de la solution de suivi de propriété des pièces détachées. Une analyse des coûts liés aux algorithmes de smart contracts est fournie sur la base des coûts de transaction et d'exécution en éther/gaz/dollars. Cependant, cette évaluation est spécifique à l'architecture proposée et ne peut être généralisée.

Un autre exemple de traçabilité de produits basée sur la blockchain est proposé par [77] pour la fabrication additive. L'objectif était d'assurer une traçabilité sécurisée et fiable, ainsi que l'accessibilité et l'immuabilité. L'architecture proposée utilise les contrats intelligents Ethereum et IPFS comme solution de stockage de fichiers distribués. Les auteurs proposent une évaluation axée sur les exigences de sécurité telles que l'intégrité, la responsabilité, la non-répudiation et l'autorisation. Cependant, le coût dû à l'utilisation d'IPFS comme solution de stockage n'est pas analysé.

Afin d'étudier la précision de la technologie blockchain pour la transparence en temps réel et les économies de coûts dans l'industrie manufacturière, [111] propose une comparaison des bénéfices réalisés par deux entreprises manufacturières. Plusieurs aspects sont pris en compte dans cette étude, notamment les coûts de démarrage de la blockchain qui sont souvent ignorés, et certaines autres limitations liées à cette technologie. L'évaluation proposée repose sur un modèle mathématique qui établit globalement qu'une usine de fabrication a des incitations à mettre en œuvre la technologie blockchain, sans aucune étude de cas pour illustrer cette conclusion.

Les questions de coût et de profit sont également examinées par [113] dans leur jeu de chaîne d'approvisionnement composé de deux entreprises, à savoir un fournisseur et un détaillant, utilisant la blockchain et les smart contracts. Les aspects étudiés incluent les risques commerciaux, les coûts de transaction et les cas stochastiques dans lesquels le déploiement de la blockchain n'en vaut pas la peine. Cependant, la méthode d'évaluation consistant en des équations de la théorie des jeux est difficile à suivre pour les professionnels de l'industrie.

Une autre proposition de [114] était une architecture blockchain à trois niveaux pour les systèmes cyber-physiques (CPS) afin de relever les défis associés à la mise en œuvre de la structure 5C-CPS. L'architecture tente d'atteindre l'intégrité, la tolérance aux pannes, la résilience, la confidentialité et la transparence, en termes de sécurité et de stockage. Les auteurs ne fournissent aucune évaluation pour étayer leur contribution.

Avec l'avènement de l'industrie 4.0, de nouveaux paradigmes de fabrication sont apparus, comme le cloud-manufacturing. Cependant, ils souffrent encore de certains problèmes liés aux réseaux industriels centralisés. Par conséquent, une architecture de réseau peer-to-peer (P2P) basée sur la blockchain a été présentée par [115]. Les principaux objectifs de cette architecture étaient de sécuriser le partage des données et d'améliorer la fiabilité et la flexibilité de la fabrication dans le cloud. Garantir la fiabilité des données est un problème clé pour tirer parti de l'analyse des mégadonnées afin d'assurer l'efficacité de la chaîne d'approvisionnement. L'évaluation ne considère que l'évolutivité du réseau blockchain concernant le nombre de création de portefeuille en fonction du nombre de participants. Ces résultats sont très limités, puisque seulement 15 participants sont impliqués, et aucune autre métrique n'est prise en compte.

Un système basé sur la blockchain préservant la confidentialité est présenté dans [117] pour les systèmes de traitement des transactions. L'idée principale consiste à trouver un compromis entre le bénéfice du partage d'informations (transparence) et le coût associé à l'affaiblissement de la confidentialité. La préservation de la confidentialité dans la blockchain fait référence au fait que seules les parties autorisées peuvent accéder aux données de transaction sensibles ou exclusives. Le système proposé est basé sur le cryptage homomorphe et une innovation récente appelée preuves à connaissance nulle (ou « zero-knowledge proofs »). L'évaluation ne rapporte que l'historique des transactions dans l'outil Multichain. Malheureusement, aucune évaluation des performances n'a été proposée afin de discuter des coûts de traitement, ce qui, à notre avis, pourrait être peu pratique pour un large éventail de types de données impliquées dans les usines de production.

5.3 Cahier des charges pour l'évaluation proposée dans la thèse

Une présentation complète et pertinente de notre approche pour évaluer l'impact d'une solution de traçabilité basée sur la blockchain dans une usine, nécessite une étude de cas. Nous avons choisi une proposition récente pour BPCAT [128], illustrée dans la Figure 4.21. Le réseau blockchain est composé de trois types de nœuds blockchain qui représentent respectivement le valideur (l'entreprise propriétaire des données de traçabilité), les clients et les prestataires impliqués dans la traçabilité du produit. Bien sûr, autant de membres de chaque type que nécessaire peuvent être ajoutés. Dans ce réseau blockchain, une blockchain fonctionne avec une copie stockée sur chaque nœud. Les données de traçabilité de produits enregistrées dans la blockchain peuvent être classées en plusieurs catégories : les données mono-partenaires qui ne concernent que l'usine de fabrication comme les données d'usinage des produits (Figure 4.21) ; les données d'intégration de composants tiers qui impliquent les fournisseurs ; et les données d'expédition impliquant les clients. Certaines de ces données peuvent être confidentielles.

BPCAT impose certaines conditions préalables sur la façon dont les données doivent être structurées. Chaque fois qu'une nouvelle donnée de traçabilité est ajoutée à un bloc, une transaction est créée. Une transaction blockchain ordinaire comprend toujours un hash de transaction et un horodatage, afin de garantir qu'une action a été effectuée à un moment précis dans la blockchain. Cependant, étant donné que le hash de la transaction ne garantit pas l'intégrité des données, BPCAT ajoute également le hash des données dans la transaction. Avec le hash des données, l'horodatage et le hash des transactions, la transparence des transactions et l'intégrité des données peuvent être garanties aux partenaires impliqués.

Les données confidentielles sont cryptées à l'aide d'une méthode de cryptage choisie par le propriétaire des données afin de garantir la confidentialité. Cependant, afin de garantir la transparence, BPCAT suggère que toute donnée cryptée dans une transaction soit accompagnée d'informations liées qui pourraient évaluer son authenticité et son intégrité en cas de besoin. Le hash des données d'origine avant chiffrement, ou le hash d'une donnée dérivée des données d'origine de manière prédéfinie, peut être utilisé à cette fin. Ces données sont insérées dans la transaction, et l'horodatage et le hash de la transaction garantissent l'authenticité et l'intégrité de l'ensemble de la transaction. En cas de litige, tout partenaire peut demander un décryptage des données afin qu'elles puissent être comparées au hash stocké dans la blockchain, et établir leur authenticité. Les données chiffrées ne peuvent être déchiffrées que par le lecteur qui les a chiffrées. De plus, les signatures numériques sont utilisées pour résoudre les litiges lorsque la responsabilité d'un joueur est mise en cause. Pour chaque transaction blockchain impliquant des données de traçabilité, chaque acteur (validateur, client, fournisseur) impliqué dans l'étape de fabrication du produit concerné doit signer la transaction, à l'aide de sa clé privée personnelle. Le message signé contient le hash de la transaction et un horodatage permettant d'authentifier la signature, et de garantir la non-répudiation.

Afin de répondre au problème de la volumétrie des données de traçabilité qui pourrait nécessiter de recourir à une solution de stockage cloud, BPCAT n'impose pas le stockage des fichiers à l'intérieur de la blockchain. Chaque participant à la blockchain peut utiliser un stockage externe. Le hash du fichier est stocké dans la blockchain afin de garantir l'intégrité et l'authenticité du fichier, et le chemin du fichier afin de récupérer facilement ce dernier. De cette façon, n'importe lequel des acteurs peut recourir à une solution de stockage en nuage si nécessaire.

Cette proposition a été évaluée à travers un prototype utilisant la plate-forme multichaîne, en termes de volume de données et de consommation d'énergie [128]. Cependant, si l'évaluation donne une idée de l'impact de la blockchain sur la base d'une sélection de données, le prototype ne permet pas d'évaluer son impact dans des scénarios de production réalistes.

5.4 Présentation du modèle de simulation

Le modèle de simulation est décomposé en deux parties : la vue globale usine visible sur la Figure 5.1 et la vue au niveau des lignes de production présentée sur la Figure 5.2

5.4.1 A l'échelle de l'usine

En premier lieu, on trouve un modèle couplé principal "Usine" servant à représenter l'ensemble de l'usine. Un modèle couplé est composé de plusieurs sous-modèles atomiques. Dans le cas de l'Usine, on trouve :

Générateur d' Ordre de Fabrication (O.F) Il s'agit du point d'entrée du simulateur. Son rôle est de générer les commandes et d'instancier les produits avec leur gamme de fabrication pour pouvoir entamer leur fabrication dans l'usine. Il ne dispose donc que d'un seul port de sortie *OUT* pour pouvoir transmettre les commandes au modèle suivant.

Ligne Représentation générique d'une ligne de production. L'usine peut contenir entre 1 et N lignes, il n'y a pas de limite définie. Elle peut être de type **Traitement**, **Assemblage** et **Essai**. Tout comme l'usine, il s'agit d'un modèle couplé qui sera donc composé de plusieurs sous-modèles atomiques. Ces derniers seront décrits dans la section ci-après. La ligne possède en entrée un port pour accueillir les produits *IN P*, un port pour les composants *IN C* puis en sortie un port pour les produits *OUT P*, un port pour les demandes de composant *OUT D* ainsi qu'un port pour les informations de traçabilité *OUT T*.

Routeur de ligne Etant donné que chaque modèle atomique est indépendant et que les produits ont besoin de transiter entre plusieurs lignes de production lors de leur fabrication, il est nécessaire d'avoir un modèle capable de transférer un produit d'une ligne à une autre : le modèle responsable de cette tâche est le routeur de ligne. Lorsqu'un produit a fini d'être traité sur une ligne de production, il est renvoyé vers le routeur par l'intermédiaire du port de sortie *OUT P* de la ligne pour être acheminé vers sa prochaine destination.

Stock global Représentation du stock de l'usine. On y retrouve l'ensemble des composants provenants des fournisseurs. Ces derniers pourront être acheminés vers les lignes de production via le port de demande *IN D*. Le composant sortira ensuite par le port *OUT C* correspondant à la ligne. Un port de sortie *OUT T* est également présent afin de transmettre des informations de traçabilité.

Expédition Destination finale du produit après être passé par toutes les destinations contenues dans sa gamme de fabrication. Le produit fini entre donc par le port *IN P* et un port de sortie *OUT T* permet de transmettre les informations de traçabilité propres à cette étape.

Blockchain Le modèle atomique représentant la blockchain et plus généralement le système de traçabilité de l'usine. En effet, tous les modèles pouvant générer des données de traçabilité possèdent un port *OUT T* connecté au port *IN T* de la blockchain.

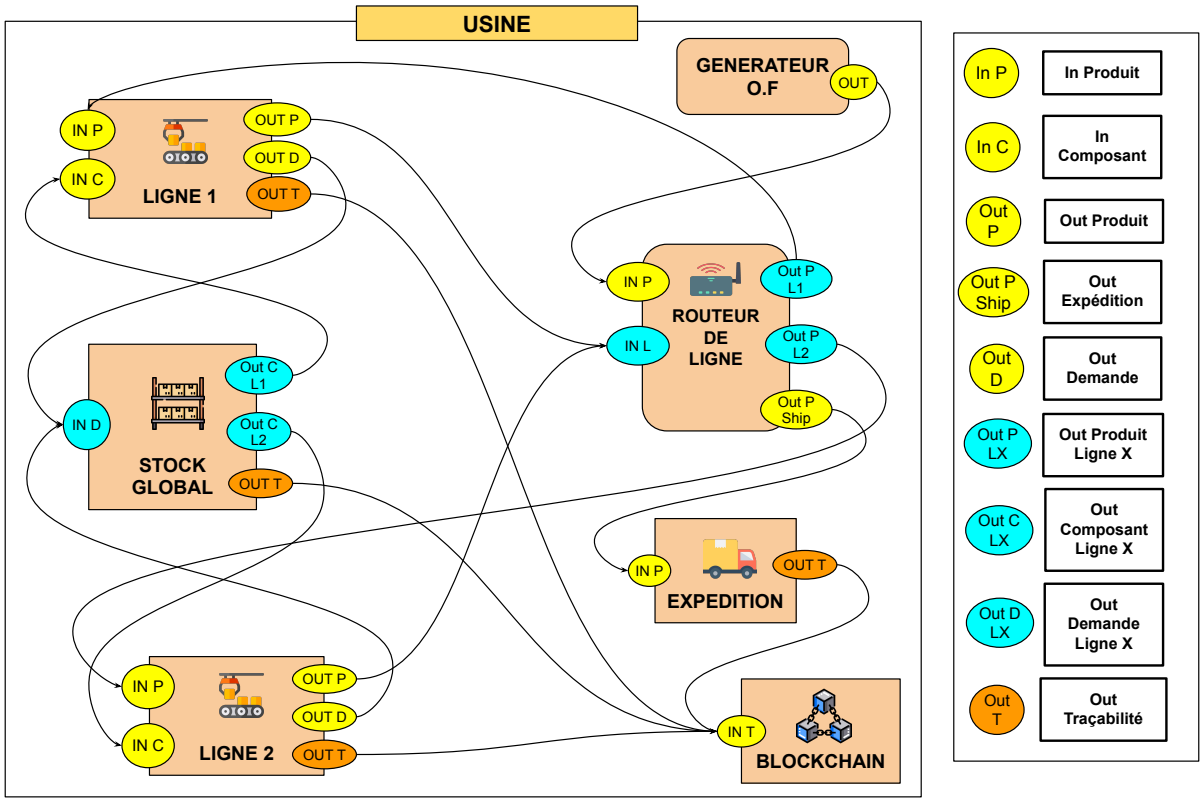


Figure 5.1 – Modèle de simulation à l'échelle de l'usine

5.4.2 A l'échelle des lignes de production

Comme expliqué précédemment, le modèle Ligne observé au niveau de l'usine est un modèle couplé. Il est composé des sous-modèles suivants :

Machine Représentation générique d'une machine. Une ligne de production donnée peut contenir entre 1 et N machines, il n'y a pas de limite définie. Une machine est amenée à recevoir des produits via son port "IN P" mais également des composants du stock global via le port *IN C*. L'apport de composants se fait via une demande envoyée par la ligne au stock global par le port *OUT D*. A l'intérieur d'une ligne, les machines représentent l'élément principal amené à générer des données de traçabilité étant donné qu'elles sont en contact direct avec le produit et effectue des opérations sur ce dernier. Les données de traçabilité sont envoyées via le port *OUT T* vers la sortie de la ligne.

Stock Amont Chaque machine est doté d'un modèle "stock amont" représentant les produits en attente d'être traité par la machine. En effet, les machines ont une cadence de fonctionnement définie appelée "temps de cycle" qui limite le nombre de produits pouvant être traités dans un intervalle de temps. Il ne possède donc qu'un port d'entrée *IN S* pour récupérer les produits en attente et un port de sortie *OUT S* pour l'envoyer vers la machine associée.

Stock Aval Tout comme pour le stock amont, chaque machine comporte un stock en aval qui représente les produits traités par la machine et en attente d'être envoyés vers leur prochaine destination.

Routeur de Machine A l'image du routeur de ligne présent au niveau de l'usine, le routeur de machine permet d'acheminer les produits vers la prochaine machine de la ligne car chaque machine est indépendante. Ce dernier possède donc autant de ports de sortie que de machines présentes sur la ligne (ports *OUT MX*)

Approvisionneur Une fois qu'une demande de composants a été émise par la ligne au stock global, le stock global renvoie le composant demandé à la ligne. Cependant, ce composant est à destination d'une machine au sein de la ligne, il est donc nécessaire d'avoir un modèle capable d'acheminer les composants reçus par la ligne vers la machine adéquate et c'est justement là le rôle de l'approvisionneur. Il possède donc autant de sortie *OUT MX* que de machines sur la ligne et chacune de ces sorties est connectée aux ports d'entrée des composants *IN C* pour les machines.

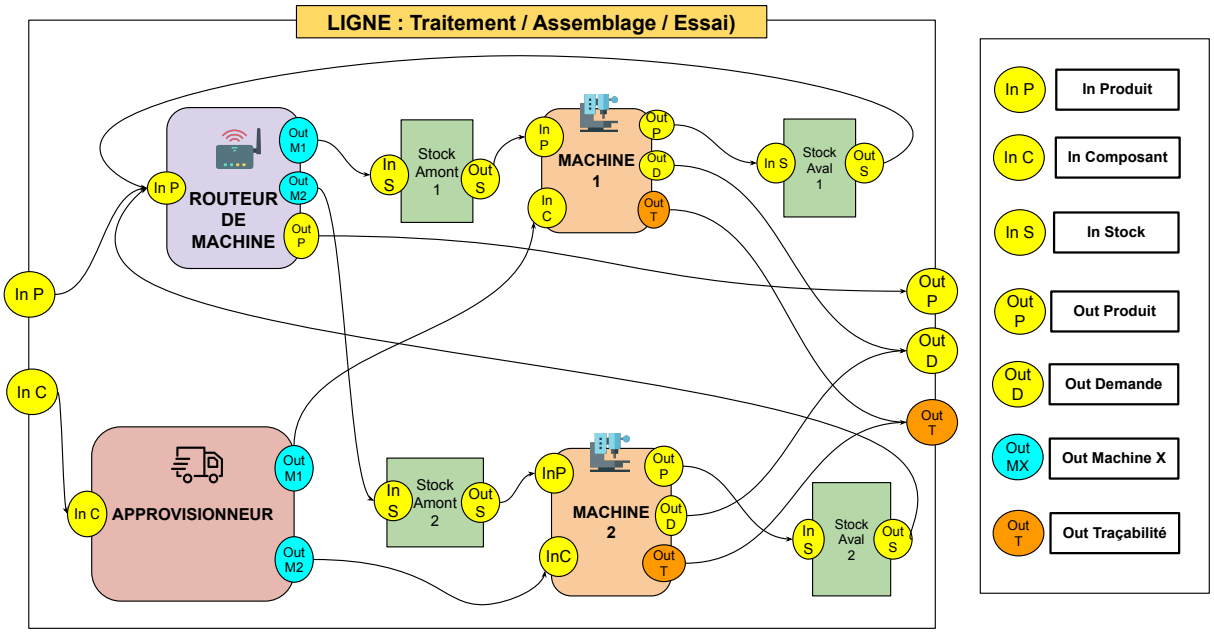


Figure 5.2 – Modèle de simulation à l'échelle de la ligne de production

5.5 Présentation des scénarios

5.5.1 Evaluation de l'espace de stockage utilisé

Dans cette partie, nous allons nous intéresser à la simulation de l'espace de stockage consommé par la blockchain et aux méthodes pour le réduire. Le prototype développé avec Multichain avait permis de montrer que cette consommation pouvait varier selon plusieurs paramètres à savoir le taux de différence entre les fichiers lors du stockage hors-chaîne qui peut réduire la taille globale de la blockchain, mais également l'abonnement ou non aux streams qui permet d'optimiser l'espace de stockage individuellement au niveau de chaque noeud.

Caractérisation des données de traçabilité

La consommation d'espace de stockage est une notion intrinsèquement liée aux données, c'est pourquoi la première étape va consister ici à caractériser la représentation des données de traçabilité au sein du simulateur.

Dans le simulateur, on ne travaille pas directement avec les données mais avec les propriétés liées à ces dernières. Le Tableau 5.3 dresse la liste exhaustive de ces caractéristiques.

Tableau 5.3 – Caractéristiques des données de traçabilité dans le simulateur

Propriété	*Description*	*Exemple*
data_id	Identifiant permettant de caractériser la donnée — LxMy : Ligne x - Machine y — Cx : Composant X — SHIP : Expédition	L2M4TD6
diff_rate	Taux de différence moyen entre deux données ayant le même *data_id*. Il s'agit d'une valeur comprise entre 0 et 1 : 0 indique que les données sont complétement identiques et 1 qu'elles sont complètement différentes	$0 \leq x \leq 1$
is_confidential	Représente si la donnée est confidentielle ou non	True | False
qty	Quantité soumise par produit	1
type	Type de la donnée d'après l'approche centrée produit (fichier ou données brutes)	*RAW* ou *FILE*
unit_size	Taille unitaire de la donnée en octets	1024

Avec ces nouvelles informations, il est désormais possible de représenter les données de traçabilité en JSON dans le scénario soumis au simulateur comme on peut le voir dans le Code 5.1.

```
{
    "data_id": "L2M4TD6",
    "diff_rate": 0.18126772124148832,
    "is_confidential": true,
    "qty": 1,
    "type": "FILE",
    "unit_size": 52571
}
```

Code 5.1 – Caractérisation des données de traçabilité dans le scénario d'une simulation

Calcul de la consommation d'espace en fonction du mode de stockage

Grâce aux caractéristiques préalablement définies, nous pouvons désormais établir la formule permettant de calculer la taille occupée par une donnée de traçabilité dans différents modes de stockage : le mode *En-Chaîne* (onchain) où tout est stocké sans optimisation et le mode *Hors-Chaîne* (offchain) où il est possible d'optimiser.

En mode *En-Chaîne*, le calcul se base essentiellement sur la taille et la quantité de données comme on peut le voir :

$$E_{co} = T_{ind} * Q_{te}$$

— Eco : L'espace total consommé
— Tind : La taille individuelle de la donnée
— Qte : La quantité soumise par produit

Dans le mode *Hors-Chaîne*, on ne stocke pas les données en doublon ; par conséquent, seules les différences sont retenues. Il est donc nécessaire d'intégrer le taux de différence dans la formule qui se présente sous la forme suivante :

$$E_{co} = T_{ind} * Q_{te} * T_{diff}$$

— Eco : L'espace total consommé
— Tind : La taille individuelle de la donnée
— Qte : La quantité soumise par produit
— Tdiff : Le taux de différence associée à cette donnée

Le Code 5.2 présente la fonction *get_consumed_space* qui n'est autre que l'implémentation en C++ des deux formules ci-dessus utilisée lors de la simulation.

```cpp
std::size_t get_consumed_space(TraceaData *tdata) const {
    PolicyStorage::values ps = get_policy_storage(tdata);
    switch (ps) {
    case PolicyStorage::ONCHAIN:
      return tdata->unit_size * tdata->qty;
    case PolicyStorage::OFFCHAIN:
        // Si la donnée a déjà été insérée au moins une fois,
        // on économise de l'espace
      if (_present_data_ids.count(tdata->data_id) > 0) {
        return tdata->unit_size * tdata->qty * tdata->diff_rate;
      } else {
        return tdata->unit_size * tdata->qty;
      }
      break;
    case PolicyStorage::EXTERNAL:
      return tdata->unit_size * tdata->qty;
    default:
      return tdata->unit_size * tdata->qty;
    }
  }
```

Code 5.2 – Fonction permettant de calculer la quantité d'espace occupée par une donnée de traçabilité en fonction du mode de stockage

Surcoût de la consommation d'espace (hash, signature...)

La taille des données n'est pas le seul élément à prendre en compte afin d'évaluer l'espace de stockage utilisé. En effet, dans l'utilisation de la blockchain pour la traçabilité centrée produit, nous intégrons d'autres informations telles que le hash des données pour garantir l'intégrité, la clé de cryptage pour la confidentialité et enfin la signature pour la non-répudiation.

Ces trois éléments sont pris en compte et également ajoutés dans la configuration du scénario (Code 5.3).

```
{
    "data_hash_size": 32,
    "encryption_key_size": 128,
    "signature_size": 65,
    "block_overhead_size": 2048,
    "tx_overhead_size": 200,
    "raw_policy_storage": "ONCHAIN",
    "file_policy_storage": "OFFCHAIN"
}
```

Code 5.3 – Configuration des éléments annexes consommateurs d'espace de stockage dans le scénario en JSON

La taille du hash est évaluée à 32 bits car l'algorithme utilisé est le *SHA-256* qui encode le hash sur 256 bits soit 32 octets. La clé de cryptage quant à elle mesure 128 octets car l'algorithme utilisé est RSA 1024 donc la clé est encodée sur 1024 bits soit 128 octets. La taille de la signature est spécifique à Multichain (64 octets + 1 octet).

Enfin, il est important de noter qu'une solution blockchain peut être amenée à générer des données supplémentaires permettant de garantir son bon fonctionnement. Cet aspect est pris en compte à travers les valeurs *block_overhead_size* qui est le surcôut d'espace consommé à chaque nouveau bloc et *tx_overhead_size* qui est le surcoût par transaction.

Caractérisation de la notion d'abonnement

Le deuxième axe permettant de diminuer l'espace de stockage utilisé est la notion d'abonnement qui, dans Multichain, s'applique à un stream et permet à un noeud de définir les données qu'il souhaite stocker. Les streams ne sont pas représentés dans le simulateur dans le sens où ils permettent de catégoriser les données, or nous possédons déjà une information similaire dans la caractérisation des données à savoir le champ *data_id*.

C'est pourquoi l'abonnement se traduira dans le scénario par le fait d'associer à chaque type de données une liste de noeuds qui leur seront abonnés comme on peut le voir dans le Code 5.4. Les choix des abonnements dans les scénarios ont été faits en accord avec la vision de la traçabilité centrée produit à savoir que le Valideur est abonné à l'ensemble des types de données car il est celui qui les soumet à la blockchain. Le Fournisseur, quant à lui est abonné à la donnée de type *C0TD0* qui est une traçabilité provenant d'un composant qu'il a lui même fabriqué. Le Client est abonné à la donnée de type *SHIPTD1* qui provient de la zone d'expédition.

```
{
    "data_subscribing": {
        "C0TD0": ["VALIDATOR", "PROVIDER"],
        "L6M1TD3": ["VALIDATOR"],
        "L1M2TD4": ["VALIDATOR", "CUSTOMER", "PROVIDER"],
        "SHIPTD1": ["VALIDATOR", "CUSTOMER"]
    }
}
```

Code 5.4 – Caractérisation de la notion d'abonnement dans le scénario d'une simulation

Dans la simulation elle-même, l'implémentation de l'abonnement consiste, pour une donnée de traçabilité, à parcourir la liste des noeuds abonnés à la catégorie de ladite donnée puis à ajouter la donnée uniquement aux noeuds qui lui sont abonnés, ce qui augmentera donc l'espace de stockage utilisé par ces derniers.

```
void update_node_stat(TraceaData *tdata) {
  // Pour chaque noeud "abonné" à cette catégorie de données
  for (auto cnode : _data_subscribing[tdata->data_id]) {
      // On ajoute les données et on augment l'espace de
      // stockage utilisé pour ce noeud
      add_data(&_node_tracea_stat[cnode], tdata);
      add_hash(&_node_tracea_stat[cnode]);
      if (tdata->is_confidential) {
          add_encryption_key(&_node_tracea_stat[cnode]);
      }
      add_signature(&_node_tracea_stat[cnode]);
  }
}
```

Code 5.5 – Implémentation de la notion d'abonnement en C++ dans le simulateur

Résultats de la simulation

A partir de cette implémentation, nous avons réalisé plusieurs simulations visant à montrer l'influence du taux de différence entre les fichiers de traçabilité sur l'espace de stockage utilisé comme on le voit sur la Figure 5.3. Sur ce graphique sont donc présentés cinq scénarios différents :

- 100% : Tous les fichiers sont différents les uns des autres. Il s'agit du scénario type où le mode hors-chaîne ne peut apporter aucune amélioration
- 70-90% : Taux de différence moyen entre 70% et 90%
- 50-70% : Taux de différence moyen entre 50% et 70%
- 30-50% : Taux de différence moyen entre 30% et 50%
- 10-30% : Taux de différence moyen entre 10% et 30%

Pour chacun des scénarios, quatre statistiques sont affichées : la quantité totale de données insérées qui ne change pas entre chaque scénario puis l'espace de stockage utilisé sur les noeuds *Valideur*, *Client* et *Fournisseur*.

Pour commencer, on voit que plus le taux de différence entre les fichiers diminue, plus l'espace consommé par la blockchain diminue également allant de **74.36Gb** pour le scénario 100% à **15.60Gb** ce qui montre bien qu'il y a économie.

De plus, en prenant individuellement chaque scénario, on peut constater les bénéfices de l'abonnement aux données, car l'espace occupé sur les noeuds Valideur, Client et Fournisseur n'est pas le même. Le noeud Valideur stocke l'intégralité des données d'où sa consommation d'espace bien supérieure tandis que le Fournisseur et le Client utilisent moins d'espace. Cela est d'autant plus marquant, qu'ici les noeuds Fournisseur et Client répresentent respectivement tous les fournisseurs et tous les clients. La quantité effective stockée par chaque fournisseur et chaque client individuellement ne serait alors qu'une fraction correspondant aux composants provenant de ce fournisseur ou aux produits livrés à ce client.

5.5.2 Evaluation de la consommation énergétique

Dans le cadre de la simulation, la consommation énergétique que nous cherchons à évaluer est celle de la blockchain et plus particulièrement les différences en fonction de l'algorithme de minage utilisé. Evaluer cette consommation consiste à évaluer celle des noeuds et par conséquent celle de leurs composants informatiques.

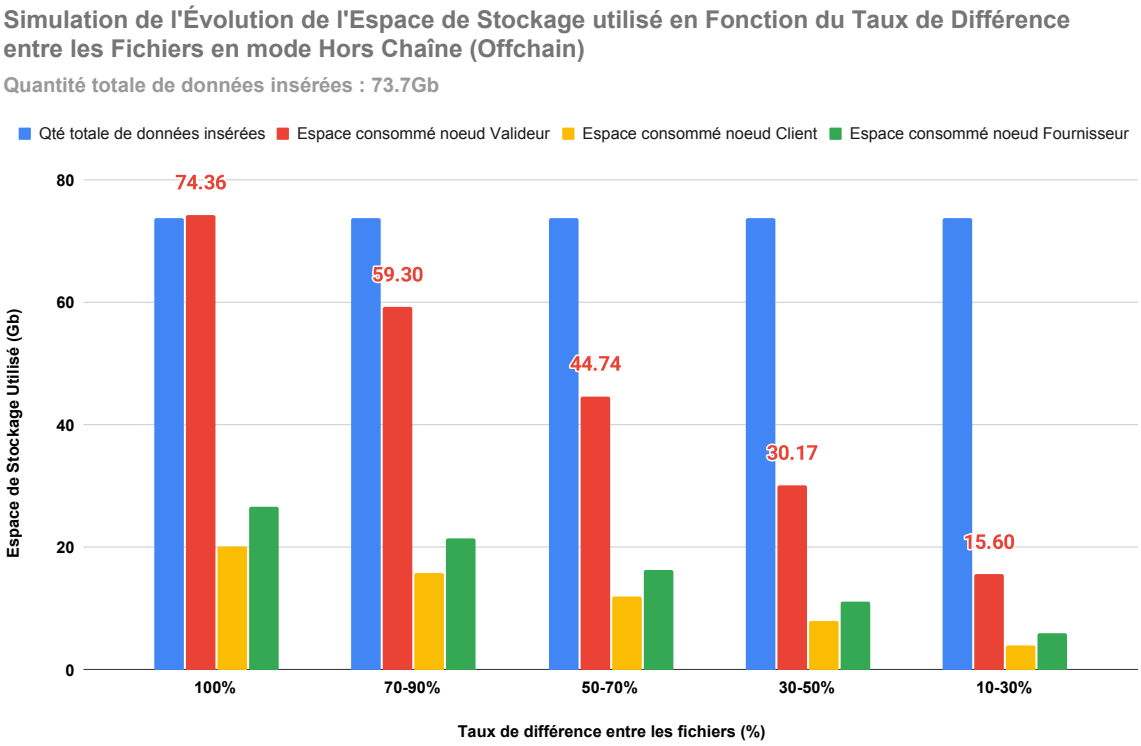


Figure 5.3 – Simulation de l'évolution de l'espace de stockage utilisé en fonction du taux de différence entre les fichiers en mode hors-chaîne

Détermination de la formule de calcul

Dans un premier temps, on s'appuie sur la formule de calcul suivante proposée par Microsoft dans le cadre de la mesure de la consommation électrique de son infrastructure Cloud [129] :

$$P[kWh] = \frac{c * Pc + Pr + g * Pg}{1000}$$

- c : Le nombre de CPU
- Pc : Consommation du CPU (en Watt)
- Pr : Consommation de la mémoire (W)
- g : Le nombre de GPU
- Pg : Consommation du GPU (W)

Etant donné qu'il n'y a aucun calcul graphique effectué au niveau des noeuds, on considère que la consommation du GPU est de 0. De même, la consommation de la mémoire étant très faible, elle est considérée comme négligeable. La consommation énergétique d'un noeud se résumera donc à la consommation Pc du CPU que l'on peut représenter avec la formule suivante :

$$Pc = tdpc * rc * th$$

- $tdpc$: TDP du CPU (en Watt)
- rc : Taux d'utilisation du CPU (en %)
- th : Temps écoulé (en heures)

Definition 5.5.1 *Formule du calcul de la consommation énergétique d'un noeud P dans le cadre de la simulation*

$$P[kWh] = \frac{c * (tdpc * rc * th)}{1000}$$

c : Le nombre de CPU
- *tdpc : TDP du CPU (en Watt)*
- *rc : Taux d'utilisation du CPU (en %)*
- *th : Temps écoulé (en heures)*

Parmi les différents paramètres de cette formule, le nombre de CPU *c* et son TDP *tdpc* font partie des caractéristiques techniques de la machine sur laquelle s'exécute la blockchain. *tdpc* aura donc pour valeur **95W**. Le nombre de CPUs alloués à chaque noeud sera identique et sera défini au niveau de la configuration du conteneur Docker avec une valeur égale à **2.0** comme l'indique le Code 5.6.

```
validator1-node-chain:
    deploy:
      resources:
        limits:
          cpus: '2.0'
          memory: 2gb
```

Code 5.6 – Configuration des ressources CPU du noeud avec Docker

c et *tdpc* sont désormais des valeurs fixes connues, elles peuvent donc être intégrées directement dans la configuration du scénario en JSON comme on peut le voir dans le Code 5.7.

```
{
"VALIDATOR": {
      "can_mine": true,
      "cpu_tdp": 95,
      "gpu_tdp": 0,
      "memory_consumption": 0.001,
      "nb_cpu": 2,
      "nb_gpu": 0
    }
}
```

Code 5.7 – Configuration du noeud valideur dans le scénario en JSON pour le calcul de la consommation énergétique

En revanche, le taux d'utilisation *rc* et le temps écoulé *th* sont des paramètres dynamiques qui varient en fonction de la charge de travail imposée au noeud. La prochaine étape va donc consister à déterminer les formules permettant de prédire ces deux valeurs.

Prédiction du taux d'utilisation CPU

Cette étape va donc consister à trouver une corrélation entre le taux de CPU utilisé et les indicateurs pouvant être mesurés dans un système blockchain réel, comme par exemple la taille des données et des transactions. Grâce au prototype développé dans le cadre de l'implémentation avec Multichain, des mesures ont pu être effectuées au niveau des noeuds en fonction de différents paramètres :

- Le mode de sauvegarde : Onchain ou Offchain
- L'algorithme de minage utilisé : Preuve de travail ou Round Robin

— L'autorisation ou non de miner

Afin de déterminer si une tendance se dégageait entre le taux de CPU et lesdits paramètres, une régression a été appliquée afin d'en déduire une équation. L'adéquation entre la droite de régression et les points de données a été évaluée grâce à la valeur du R^2.

On retrouve par exemple sur la Figure 5.4 la relation entre le taux de CPU et la taille des données dans le cadre d'un Round-Robin où tous les noeuds ont le droit de miner.

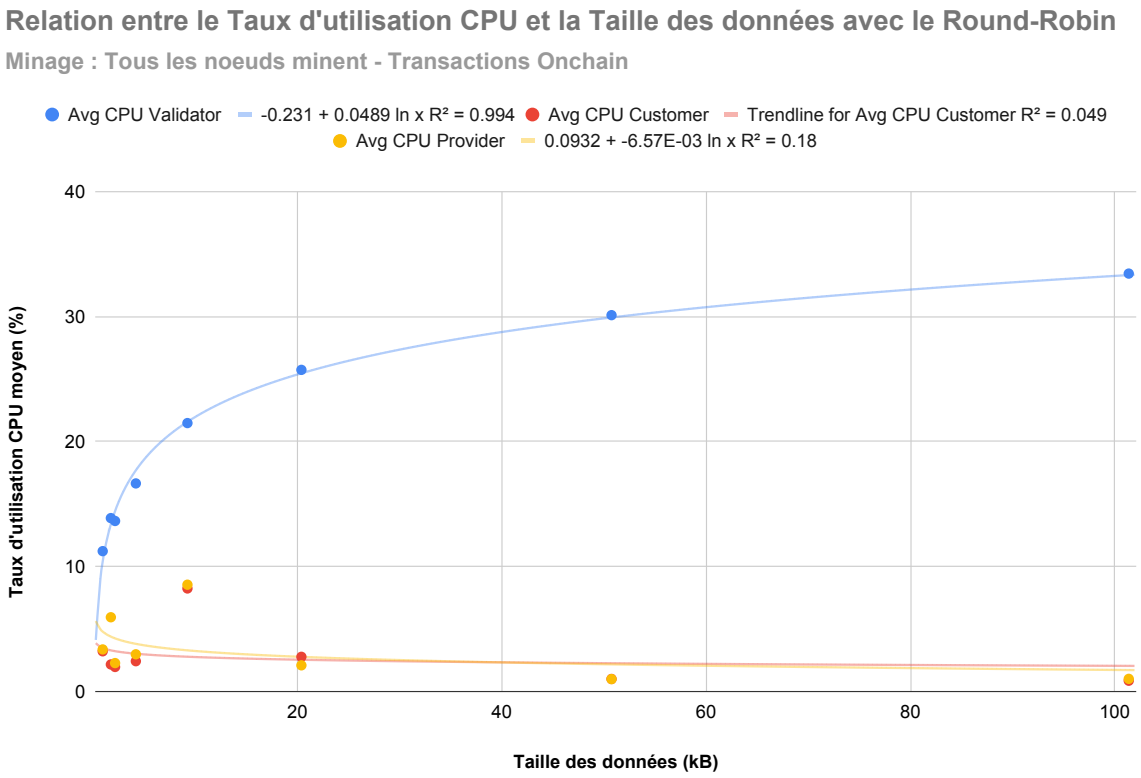


Figure 5.4 – Relation entre le taux de CPU et la taille des données pour le Round-Robin (tous les noeuds minent)

Une tendance se dessine notamment pour le taux de CPU du valideur où la courbe de régression est de type logarithmique avec un R^2 égal à **0.994** ce qui confirme bel et bien la corrélation.

En revanche, pour les noeuds Client et Fournisseur, aucune tendance claire ne se dessine et le taux de CPU reste quasiment constant ce qui paraît logique car leurs actions se résument à récupérer localement les données des transactions transmises par le valideur et à signer les blocs dans le cadre du minage. Or, ces actions ne sont pas très coûteuses en CPU comme le montrent leurs courbes. La majeure partie du travail, à savoir l'enregistrement et le traitement des transactions, est effectuée par le valideur qui a la plus forte activité CPU.

Sur la Figure 5.5, on retrouve une étude similaire mais cette fois-ci avec la preuve de travail.

Au premier abord, les mesures avaient été effectuées de la même manière que pour le graphique précédent, c'est à dire par rapport à la taille des données. Or, il s'est avéré qu'il n'y avait pas de lien fort entre la taille des données et le taux de CPU avec la preuve de travail, car la valeur du R^2 était inférieure à **0.1**. Un autre paramètre a dû être trouvé et la difficulté de minage s'est avérée être le candidat tout désigné.

La difficulté de minage est une mesure arbitraire que la blockchain fait évoluer afin de pouvoir garantir une durée de minage stable des blocs (un bloc généré toutes les N secondes). C'est une valeur qui influe directement sur la difficulté du calcul et donc sur la consommation en terme de ressources CPU, d'où son choix pour cette mise en corrélation. Etant donné que la preuve de travail est un algorithme consommant

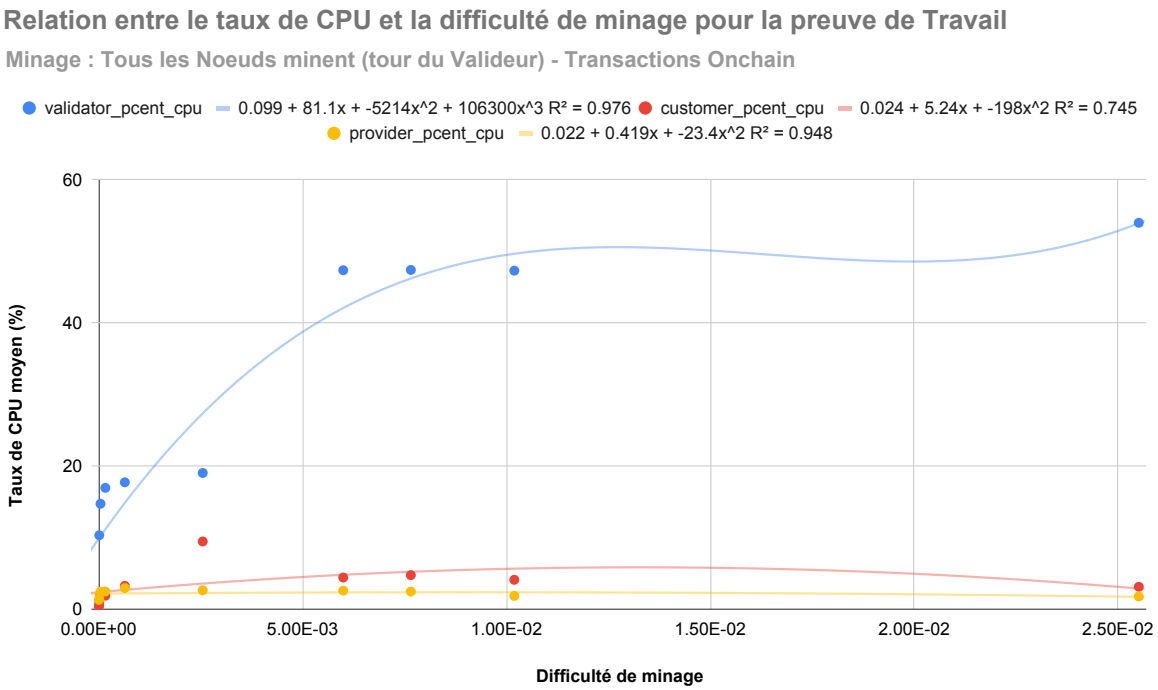


Figure 5.5 – Relation entre le taux de CPU et la taille des données pour la preuve de travail (tous les noeuds minent)

beaucoup de ressources, il est important de différencier pour chaque noeud les mesures où le noeud est le créateur du bloc et celles où il ne l'est pas. La Figure 5.5 montre par exemple les mesures lorsque le Valideur a créé les blocs, ce qui explique pourquoi sa consommation de CPU est plus importante que celles des neuds Client et Fournisseur. On constate une corrélation forte avec la difficulté de minage, car les valeurs des R^2 sont supérieures à **0.95** pour deux des noeuds.

Cela valide donc l'hypothèse d'avoir choisi la difficulté de minage comme paramètre à mettre en relation avec le taux de CPU utilisé.

Les graphiques liés à la détermination du taux de CPU pour les noeuds Client et Fournisseur sont disponibles dans l'annexe B.3.

Une fois que les équations sont déterminées, il est nécessaire de les inclure dans le scénario du simulateur. Le Code 5.8 présente l'intégration des équations des graphiques précédents dans le scénario en JSON utilisé par le simulateur. On retrouve différents types d'équation à savoir des équations logarithmiques ou bien des polynômes de degré N.

```
{
    "cpu_usage_nonpow_onchain_mine": {
      "type": "LOGARITHMIC",
      "coeffs": [0.0489, -0.231]
    },
    "cpu_usage_nonpow_onchain_nonmine": {
      "type": "CONSTANT",
      "coeffs": [0.02]
    },
    "cpu_usage_pow_onchain_mine_multi_validator_turn": {
      "type": "POLYNOMIAL",
      "coeffs": [0.099, 81.1, -5214, 106300]
    }
}
```

Code 5.8 – Exemple d'intégration des équations liés au calcul du taux de CPU dans le scénario en JSON

Prédiction du temps de traitement des transactions

La seconde inconnue dans la formule du calcul de la consommation énergétique est le temps de traitement de la transaction *tc*. De la même manière que pour le taux de CPU, des mesures liées au temps de traitement ont été effectuées en fonction de différents scénarios. Mesurer le temps de traitement de la transaction consiste à calculer le temps d'exécution T de la fonction liée à la soumission des données : on peut représenter le calcul par l'équation suivante :

$$T_{ec} = T_{fin} - T_{deb}$$

- *Tec* : le temps écoulé
- *Tfin* : Temps à la fin de l'exécution de la fonction
- *Tdeb* : Temps au début de l'exécution de la fonction

L'implémentation de cette formule en python dans le prototype est visible dans le Code 5.9. La fonction *process_time_ns* n'inclut pas le temps écoulé durant les phases d'inactivité du CPU.

```
def time_func(func, **kwargs):
    start = time.process_time_ns()
    result = func(**kwargs)
    elapsed = time.process_time_ns() - start
    return elapsed, result
```

Code 5.9 – Fonction permettant de calculer le temps d'exécution d'une fonction

Etant donné que le traitement de la transaction est lié à son enregistrement, il paraîtrait logique que le temps soit lié à la taille des données. La Figure 5.6 présente la corrélation entre le temps de traitement et la taille des données dans le cadre de transactions onchain. On distingue une tendance nette permettant de modéliser l'évolution du temps selon une équation linéaire avec un R^2 supérieur à **0.99**.

Après avoir déterminé l'équation, cette dernière peut désormais être intégrée dans la configuration en json du scénario (Code 5.10).

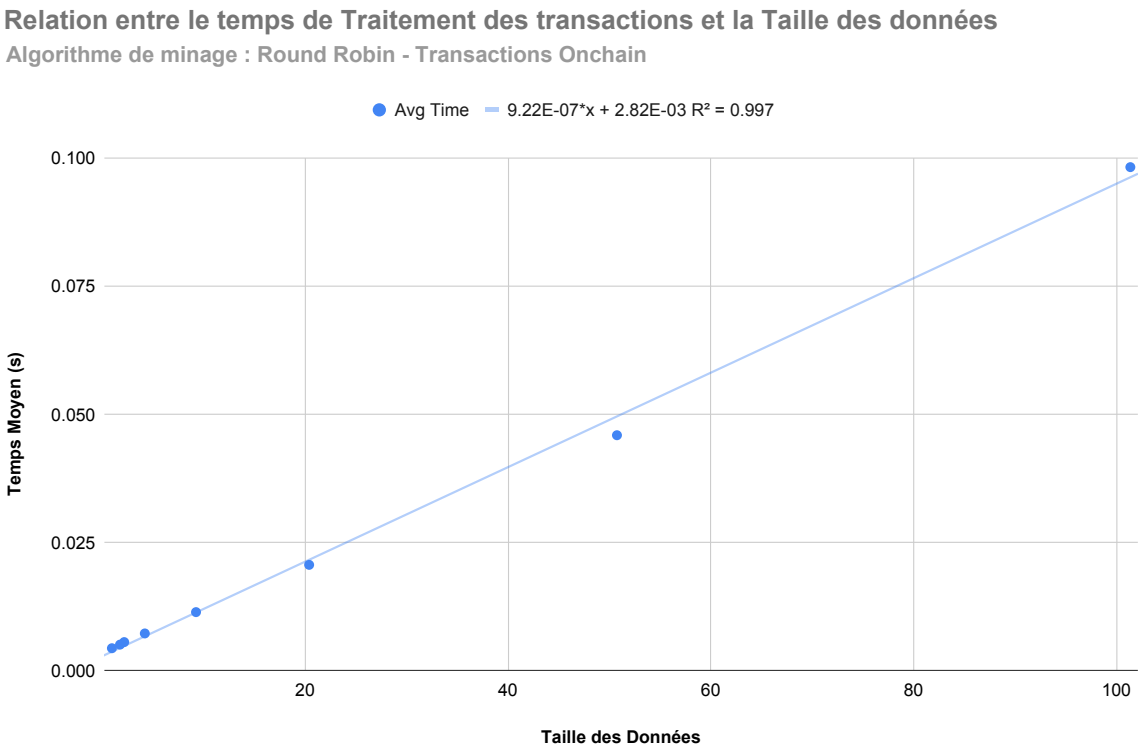


Figure 5.6 – Relation entre le temps de Traitement des transactions et la Taille des données (Transactions Onchain)

```
{
    "processing_time_onchain": {
      "type": "LINEAR",
      "coeffs": [9.22e-7, 2.82e-3]
    }
}
```

Code 5.10 – Intégration de l'équation du calcul du temps de traitement dans le scénario en json

Intégration de la formule de calcul de la consommation électrique dans la simulation

Dans la simulation, le calcul de la consommation électrique est implémenté de la façon suivante (Code 5.11). On récupère donc le taux de CPU utilisé *cpu_usage* ainsi que le temps de traitement de la transaction *tx_duration* puis on applique la Definition 5.5.1.

```
for (auto node : _blockchain_nodes) {
      double cpu_usage = get_cpu_usage(tdata, &node);
      double tx_duration = get_processing_time(tdata);
      // Formule de calcul de la consommation électrique
      double energy_consumption = (node.profile.nb_cpu * node.profile.cpu_tdp *
                                   (tx_duration * cpu_usage)) / 1000.0;
}
```

Code 5.11 – Intégration de la formule de calcul de la consommation électrique d'un noeud en C++

Comparaison des résultats selon différents scénarios

La Figure 5.7 présente la synthèse de plusieurs simulations visant à évaluer la consommation électrique de la blockchain sur une durée de 1 jour en se basant sur les scénarios suivants :

— nPOW - VCP (Round Robin - Tous les noeuds minent)
— POW - VCP (Preuve de travail - Tous les noeuds minent)
— POW - VP (Preuve de travail - Seuls le Valideur et le Fournisseur minent)
— POW - V (Preuve de travail - Seul le Valideur mine)

Pour chacun de ces scénarios, on a la consommation par noeud (Valideur, Client, Fournisseur) ainsi que la consommation globale qui correspond au total des noeuds.

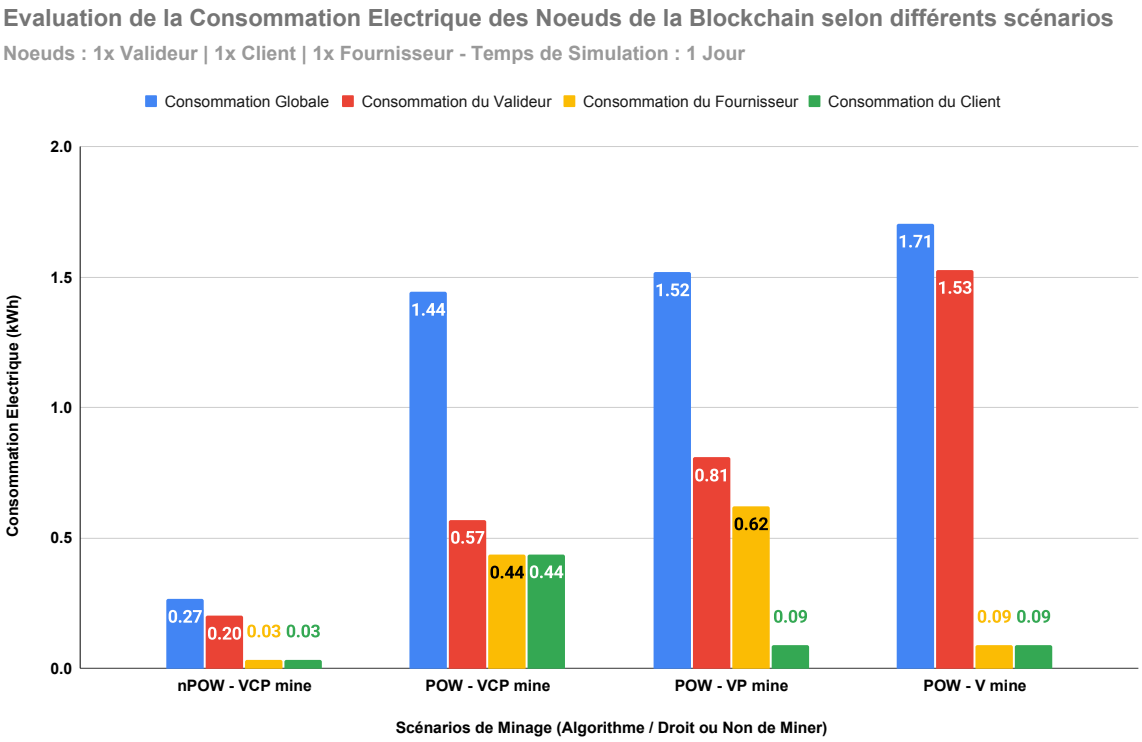


Figure 5.7 – Evaluation de la Consommation Electrique des Noeuds de la Blockchain selon différents scénarios de Minage

Tout d'abord, on constate un écart très important de consommation entre les scénarios utilisant la preuve de travail (**1.44kWh** pour POW - VCP) et le scénario utilisant l'algorithme de Round-Robin (**0.27kWh** pour nPOW - VCP). La supériorité de la preuve de travail en terme de consommation électrique est donc bien confirmée par la simulation.

Entre les scénarios impliquant la preuve de travail se dessine un phénomène de répartition de la consommation électrique au fur et à mesure que le nombre de noeuds autorisés à miner augmente : par exemple, dans le scénario POW - V, le Valideur est responsable de la quasi entièreté de la consommation électrique pour un total de **1.71kWh** alors que dans POW - VP , la consommation commence à s'équilibrer entre le Valideur et le Fournisseur avec **0.81kWh** et **0.62kWh** ce qui est logique car ils sont les seuls à pouvoir miner. Enfin, dans le scénario POW - VCP où tout le monde mine, la répartition est quasi homogène entre les 3 noeuds avec **0.57kWh** pour le Valideur et **0.44kWh** pour les deux autres.

Cet aspect est intéressant, car il ouvre la voie à une discussion sur la répartition des coûts entre les acteurs en terme de consommation électrique liée au minage dans la blockchain. Notamment, dans la mesure où la transparence et l'intégrité offertes par la blockchain pourraient être entrevues comme des services à l'usine, à

ses fournisseurs et à ses clients, il laisse entrevoir ici la possibilité de faire supporter le coût énergétique lié à son fonctionnement à l'ensemble des acteurs par exemple sous la forme d'une souscription.

5.5.3 Evaluation des émissions de carbone (CO2)

L'aspect écologique constitue un des défis de la technologie blockchain comme cela a déjà été rappelé dans les chapitres précédents. Afin de pouvoir évaluer cet aspect, la mesure des émissions carbone est nécessaire. Pour rappel, à ce stade, nous sommes capables d'estimer la consommation d'énergie électrique de la blockchain via le simulateur.

Détermination de la formule de calcul

Avec l'API open-source electricitymap.org, nous pouvons récupérer une multitude de paramètres liés à la consommation électrique et plus particulièrement l'intensité carbone par pays. L'intensité carbone correspond au rapport entre les émissions de CO2 et la consommation électrique et s'exprime en gCO2eq/kWh. Les données de cette API proviennet généralement des gestionnaires de réseau, des entités d'équilibrage ou des opérateurs de marché liés au marché de l'énergie et sont utilisés par les opérateurs Cloud afin de réduire leur empreinte carbone. A partir de l'intensité carbone, on peut donc déterminer la quantité de CO2 émise E en fonction de la consommation P par la formule suivante :

Definition 5.5.2 *Formule du calcul de la quantité de CO2 émise E par un noeud dans le cadre de la simulation*

$$E = P * Ic$$

- E : *La quantité de CO2 émise (en g)*
- P : *La consommation électrique du noeud (en kWh)*
- Ic : *Intensité carbone du pays où le noeud est hébergé (en gCO2 eq/kWh)*

L'intensité carbone Ic est une donnée connue dont la valeur est fixe donc elle peut être intégré dans le scénario en json comme le montre le Code 5.12.

```
{
   "hosting_areas":[{
        "area_name":"FR",
        "carbon_intensity":73,
        "renewable_part":0.3,
        "kwh_cost":0.177
      }]
}
```

Code 5.12 – Configuration des paramètres permettant le calcul des émissions CO2 dans le scénario en json

Au niveau du simulateur, la formule de calcul est intégrée de la manière suivante en C++ comme le montre le Code 5.13.

```cpp
for (auto node : _blockchain_nodes) {
      double cpu_usage = get_cpu_usage(tdata, &node);
      double tx_duration = get_processing_time(tdata);
      double energy_consumption = (node.profile.nb_cpu * node.profile.cpu_tdp *
                                   (tx_duration * cpu_usage)) / 1000.0;
    for (auto &area : _hosting_areas) {
        // Formule de calcul de la quantité de CO2 émise
        double co2_emission = energy_consumption * area.carbon_intensity;
    }
}
```

Code 5.13 – Intégration de la formule de calcul des émissions de CO2 d'un noeud en C++

Comparaison des résultats selon différents scénarios

Les résultats liés aux simulations des émissions de CO2 peuvent être retrouvées sur deux graphiques : Figure 5.8 et Figure 5.9. Sur les deux figures, la durée de simulation est de 1 jour et les scénarios présentés sont les suivants :

- nPOW - VCP (Round Robin - 1 Valideur, 1 Client et 1 Fournisseur)
- nPOW - VC20P5 (Round Robin - 1 Valideur, 20 Clients et 5 Fournisseurs)
- POW - VCP (Preuve de travail - 1 Valideur, 1 Client et 1 Fournisseur)
- POW - VC20P5 (Preuve de travail - 1 Valideur, 20 Clients et 5 Fournisseurs)
- nPOW - VC100P20 (Round Robin - 1 Valideur, 100 Clients et 20 Fournisseurs)
- POW - VC100P20 (Preuve de travail - 1 Valideur, 100 Clients et 20 Fournisseurs)

Dans l'ensemble des scénarios, tous les noeuds minent.

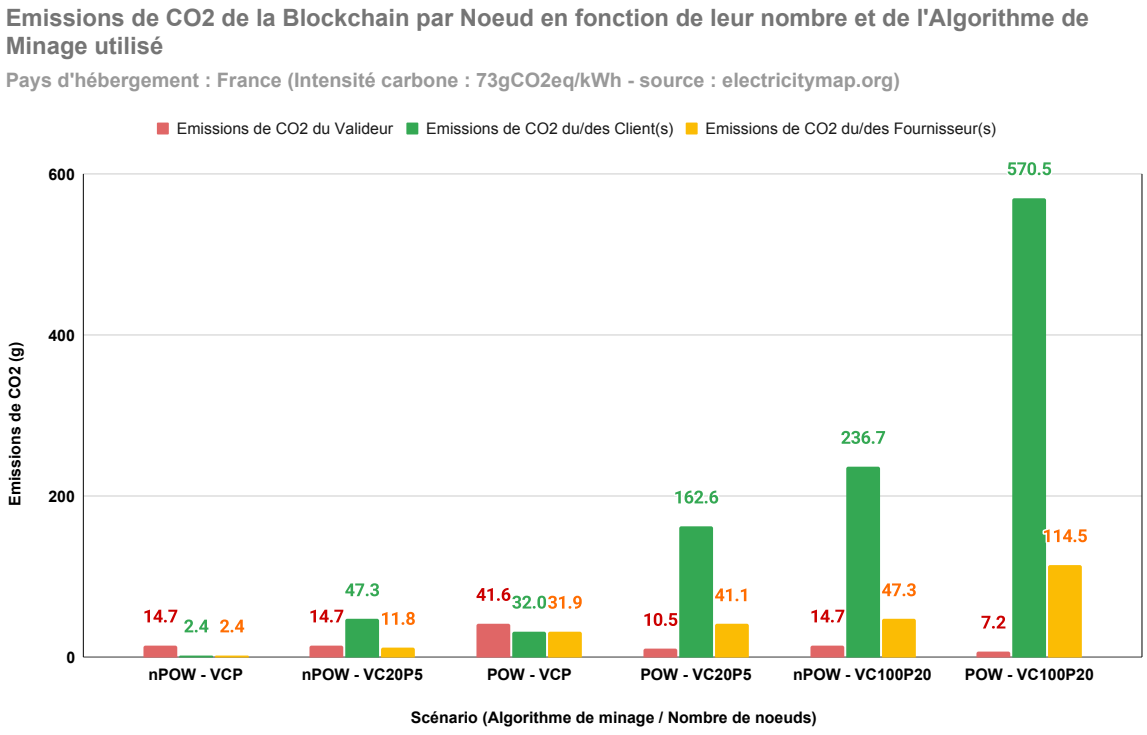


Figure 5.8 – Emissions de CO2 de la Blockchain par Noeud en fonction de leur nombre et de l'Algorithme de Minage utilisé

La Figure 5.8 affiche le détail des émissions de CO2 par rôle de noeud, les rôles possibles étant Valideur, Client et Fournisseur. Plusieurs noeuds peuvent avoir le même rôle : par exemple dans le scénario "POW - VC20P5", on a 20 clients et 5 fournisseurs. Cela signifie donc que les émissions du rôle Client correspondent à celle des 20 noeuds clients cumulés et la consommation du rôle Fournisseur à celle des 5 noeuds fournisseurs cumulés.

Dans un premier temps, on constate que le scénario émettant le moins de CO2 est "nPOW - VCP" ce qui est un résultat attendu car il utilise l'algorithme le moins consommateur de ressource à savoir le Round-Robin tout en ayant le nombre de noeuds le plus faible (1 noeud pour chaque rôle).

Pour l'ensemble des scénarios "nPOW", les émissions du Valideur restent stables à **14.7g** car le nombre de noeud Valideur ne change pas. En revanche, les émissions des rôles Client et Fournisseur ne cessent d'augmenter car leur nombre augmente en permanence la valeur la plus élevée étant celle des clients dans le scénario "nPOW - VC100P20" avec **236.7g**. On note également une répartiton proportionnelle des émissions entre les clients et les fournisseurs : par exemple dans nPOW - VC100P20, les émissions des 100 client sont de 236.7g et celles des 20 fournisseurs sont de 47.3g ce qui correspond à 5 fois moins que les clients.

Avec les scénarios "POW", on observe un phénomène de diminution des émissions de CO2 au niveau du Valideur : **41.6g** dans "POW - VCP" et **7.2g** dans "POW - VC100P20". Cela provient du fait que les émissions de CO2 proviennent essentiellement de l'activité liée au minage et donc à la preuve de travail. Or, plus on augmente le nombre de noeuds clients et fournisseur, plus la probabilité de miner un bloc diminuera pour le valideur : par exemple, dans le scénario POW VCP, le valideur a 1 chance sur 3 de miner tandis que dans POW - VC100P20, il a une chance sur 121. Au niveau des clients et fournisseurs, les émissions sont croissantes au fur et à mesure que le nombre de noeuds augmente et la valeur la plus élevée provient des clients dans le scénario "POW - VC100P20" avec **570.5g**.

Enfin, en comparant les scnarios utilisant la preuve de travail "POW" et les scénarios à base de Round-Robin "nPOW", on peut voir qu'à nombre de noeud équivalent, la preuve de travail émet toujours plus de CO2 que le Round-Robin.

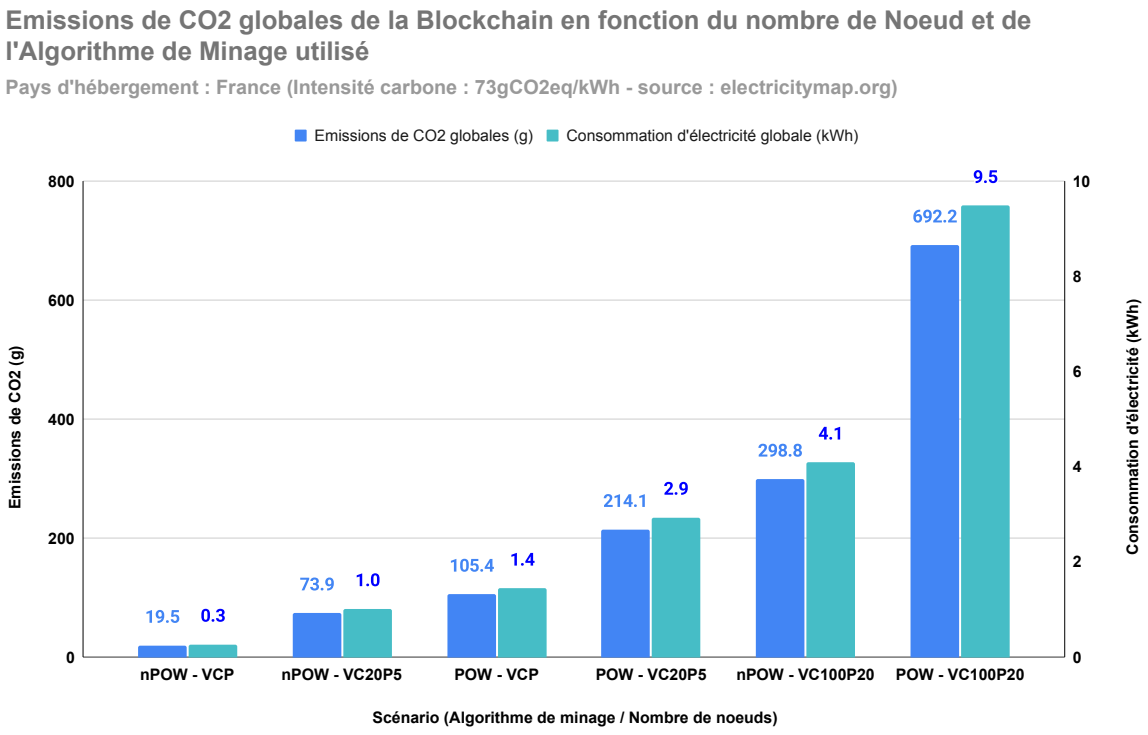


Figure 5.9 – Emissions de CO2 globales de la Blockchain en fonction du nombre de Noeud et de l'Algorithme de Minage utilisé

La Figure 5.9 affiche à la fois la quantité globalde de CO2 émise par la blockchain c'est à dire émise par l'ensemble des noeuds ainsi que la consommation électrique globale afin de bien montrer que la corrélation entre les deux paramètres est bien respectée dans la simulation. Les scénarios sont affichés dans l'ordre croissant en terme de consommation éelctrique et d'émissions CO2.

Sur les six scénarios, on voit que les deux plus économes sont ceux utilisant le Round-Robin à savoir "nPOW - VCP" avec **19.5g** et "nPOW - VC20P5" avec **73.9**. Cependant, un des scénarios utilisant le Round-Robin parvient à se hisser à la seconde place : il s'agit de "nPOW - VC100P20" avec 100 clients, 20 fournisseurs et une consommation totale de **298.8g**. Le nombre de noeuds n'est donc pas à négliger quant à son influence dans l'augmentation de la consommation.

Entre "nPOW - VCP" et "POW - VCP", la consommation totale passe de **19.5g** à **73.9g** en changeant l'algorithme de minage tandis qu'entre "nPOW - VCP" et "nPOW - VC20P5", la consommation passe de **19.5g** à **105.4g** en changeant le nombre de noeuds (de 3 noeuds à 26). On en déduit que l'influence de l'algorithme de minage est supérieure à celle du nombre de noeuds.

5.6 Synthèse des points forts et points faibles de la contribution

Dans ce chapitre, nous avons proposé une approche d'évaluation basée sur la spécification de système à événements discrets pour l'Industrie 4.0. Notre proposition principale est un framework permettant une modélisation collaborative d'une usine de fabrication, dans lequel chaque composant peut être modélisé et configuré par un expert aussi finement que nécessaire en fonction des objectifs d'évaluation. Le framework offre également une vue d'ensemble de l'usine où les responsables peuvent configurer (planification de la production, gestion du stockage,..) et observer les modèles (volume de stockage, consommation d'énergie...).

En appliquant cette approche à notre proposition récente de système de traçabilité basé sur la blockchain dans une usine de fabrication de l'industrie 4.0 (BPCAT), nous avons montré comment les modèles peuvent être configurés et comment la simulation peut être effectuée en fonction d'objectifs spécifiques. Les résultats de cette étude de cas suggèrent que le volume de stockage nécessaire au stockage de données blockchain peut être contrôlé et même réduit lors de l'utilisation du stockage hors chaîne (offchain), en fonction du taux de différence entre les données. Il convient de noter que dans les processus de fabrication, une machine peut souvent produire des données de traçabilité similaires, ce qui suggère un faible taux d'écart et une réelle possibilité de réduire le volume de stockage des données. De plus, nous avons montré que la consommation d'énergie, qui est un problème que les industriels soulèvent souvent contre les solutions blockchain, peut également être contrôlée par une répartition judicieuse entre les participants à la blockchain. En effet, chaque participant bénéficie de l'immuabilité et de la transparence apportées par la blockchain. Par conséquent, au lieu de facturer à l'usine de fabrication l'intégralité des coûts énergétiques ainsi que l'impact environnemental, il est possible de les répartir entre les participants de la blockchain, par exemple en tant que service proposé aux fournisseurs et aux clients qui s'abonnent.

Le simulateur généré à partir de la modélisation peut être utilisé pour évaluer l'usine sur des périodes allant de quelques jours à des années de production, y compris en utilisant des données de production réelles. Dans nos travaux futurs, nous prévoyons d'étendre le framework afin qu'il puisse être utilisé comme un outil de gestion opérationnelle ou un jumeau numérique.

Conclusion et perspectives générales 6

Le travail présenté tout au long de ce mémoire traite de l'intégration d'une solution blockchain dans le système de traçabilité d'une usine 4.0 avec pour objectif sous-jacent l'amélioration de la confiance entre les partenaires sans compromettre la confidentialité des données de traçabilité.

Nous avons tout d'abord procédé à un état des lieux de l'usine 4.0 sous l'angle de la cybersécurité afin d'avoir un meilleur aperçu des technologies et systèmes qui y étaient présentes. Après avoir introduit le concept d'Industrie 4.0, nous avons montré la complexité induite par les mécanismes de cybersécurité pour assurer la sécurité des systèmes, en raison de nombreuses nouvelles technologies impliquées. Nous avons ensuite proposé une caractérisation de la cybersécurité sous ses aspects à la fois techniques et managériaux pour montrer que chaque niveau d'une organisation a un rôle à jouer. Par ailleurs, nous avons souligné que la cybersécurité pouvait aussi être un levier de création de valeur. Etant donné que la cybersécurité doit être considérée en fonction des zones physiques, nous avons proposé un découpage d'une usine en plusieurs périmètres, que nous avons caractérisés en fonction de leurs interactions, équipements et réseaux. Nous avons ensuite rapporté les vulnérabilités, menaces et risques de cybersécurité rencontrés dans les usines de l'Industrie 4.0 pour chaque périmètre, tout en prenant en compte les impacts métiers au niveau organisationnel. Il convient de noter qu'en plus du paradigme du système expert habituellement mis en œuvre dans les solutions de cybersécurité, de nouvelles technologies innovantes, telles que le machine learning, les honeypots et les jumeaux numériques, sont utilisées dans les solutions récentes. Nous avons complété cette revue de solutions scientifiques par une synthèse des bonnes pratiques promues par les organismes officiels et qui peuvent servir de point de départ pour établir une stratégie de cybersécurité.

Notre analyse sur l'ensemble de ces travaux référencés ainsi que sur les normes et préconisations des principaux organismes est que la cybersécurité est encore aujourd'hui menée comme une activité destinée à détecter les vulnérabilités des systèmes existants et à les combler. Cela donne l'impression d'une grande course derrière les attaques qui dans la plupart des cas semblent inévitables, et dont on recherche principalement à retarder la survenue ou à réparer les conséquences. Plus que jamais, une perspective intéressante consiste à repenser la conception et la mise en oeuvre des systèmes en intégrant la sécurité, et donc la cybersécurité, comme une caractéristique indispensable dans son fonctionnement normal et de sélectionner les techniques et les outils qu'on intègre au système avec la contrainte de préserver cette caractéristique.

Par la suite, nous avons présenté une solution de traçabilité basée sur la blockchain pour les usines de fabrication, qui offre un moyen efficace d'assurer la transparence à tous les partenaires impliqués tout en préservant la confidentialité de leurs données critiques respectives. L'idée principale consiste à inclure des données confidentielles cryptées, ainsi que leur hash calculé avant cryptage dans les transactions blockchain. Ainsi, en cas de litige, les enquêteurs peuvent demander au propriétaire des données d'accéder aux informations confidentielles et de les comparer avec les données connexes validées par tous les participants de la blockchain afin d'établir leur authenticité lors de la procédure d'analyse des défauts du produit. Il convient de noter que la cryptographie homomorphique porte la promesse d'apporter une solution au blocage que pourrait représenter le refus par l'auteur des données à les déchiffrer. En effet, les mécanismes annoncés sous-entendent la possibilité de vérifier directement les données cryptées sans avoir besoin de les déchiffrer. Quoiqu'il en soit, cela n'enlevera pas la nécessité de les avoir chiffrées avant leur insertion dans la blockchain afin de préserver leur confidentialité. Afin de réduire justement le délai lié au chiffrement et éviter une consommation de stockage prohibitive pour les fichiers de traçabilité, seul le hash du fichier doit obligatoirement être inclus dans la transaction blockchain. La mise en œuvre des concepts et fonctionnalités proposés a été illustrée à l'aide de l'outil blockchain Multichain. Nous montrons que le stockage des fichiers de traçabilité peut être optimisé afin de réduire le volume de données et le volume de stockage total nécessaire à chaque nœud en utilisant respectivement les fonctionnalités de stockage offchain et de streams qui sont natives dans Multichain.

Une perspective intéressante à nos travaux consisterait à explorer la possibilité d'intégrer des approches telles que la cryptographie homomorphique et le zero-knowledge proof dans la conception de système de traçabilité supportant nativement la confidentialité.

Enfin, nous avons proposé une approche d'évaluation basée sur la spécification de système à événements discrets pour l'Industrie 4.0. Notre proposition principale est un framework permettant une modélisation collaborative d'une usine de fabrication, dans lequel chaque composant peut être modélisé et configuré par un expert aussi finement que nécessaire en fonction des objectifs d'évaluation. Le framework offre également une vue d'ensemble de l'usine où les responsables peuvent configurer (planification de la production, gestion du stockage,..) et observer les modèles (volume de stockage, consommation d'énergie...). En appliquant cette approche à notreproposition de système de traçabilité basé sur la blockchain dans une usine de fabrication de l'industrie 4.0 (BPCAT), nous avons montré comment les modèles peuvent être configurés et comment la simulation peut être effectuée en fonction d'objectifs spécifiques. Les résultats de cette étude de cas suggèrent que le volume de stockage nécessaire au stockage de données blockchain peut être contrôlé et même réduit lors de l'utilisation du stockage hors chaîne, en fonction du taux de différence entre les données. Il convient de noter que dans les processus de fabrication, une machine peut souvent produire des données de traçabilité similaires, ce qui suggère un faible taux d'écart et une réelle possibilité de réduire le volume de stockage des données. De plus, nous avons montré que la consommation d'énergie, qui est un problème que les industriels soulèvent souvent contre les solutions blockchain, peut également être contrôlée par une répartition judicieuse entre les participants de la blockchain. En effet, chaque participant bénéficie de l'immuabilité et de la transparence apportées par la blockchain. Par conséquent, au lieu de facturer à l'usine de fabrication l'intégralité des coûts énergétiques ainsi que ceux liés à l'impact environnemental, il est possible de les répartir entre les participants de la blockchain, par exemple en tant que service proposé aux fournisseurs et aux clients qui s'abonnent. Le simulateur généré à partir de la modélisation peut être utilisé pour évaluer l'usine sur des périodes diverses telles que plusieurs jours, semaines ou même des années de production, y compris en utilisant des données de production réelles.

Une perspective intéressante consisterait à étendre le framework proposé afin qu'il puisse être utilisé comme un outil de gestion et de planification opérationnelles d'une usine 4.0, ou encore comme une base de travail pour la réalisation d'un jumeau numérique de l'usine à des fins d'évaluation de solutions avant leur mise en oeuvre dans l'usine réelle.

7 Liste de publications

— V. Mullet, P. Sondi and E. Ramat, (2022) "A Blockchain based Confidentiality-Preserving Approach to Traceability in Industry 4.0," International Journal of Advanced Manufacturing Technology, 2022 **(Accepted) [IF 3.5]**.

— V. Mullet, P. Sondi and E. Ramat, (2022) "Enhancing Trust in Industry 4.0 Traceability Data using Confidentiality-Preserving Digital Ledger," in Proceedings of BRAINS 2022, the 4th Conference on Blockchain Research & Applications for Innovative Networks and Services, IEEE, Paris, 2022 **(Published)**.

— V. Mullet, P. Sondi and E. Ramat, (2021) "A Review of Cybersecurity Guidelines for Manufacturing Factories in Industry 4.0," in IEEE Access, vol. 9, pp. 23235-23263, 2021 **(Published) [IF 3.4]**.

Annexes

A

Présentation de Multichain

Multichain est une plateforme blockchain open-source permettant de construire et de déployer des applications blockchain privées qui peuvent être utilisés au sein d'une même entreprise ou entre plusieurs entreprises. Il s'agit d'un fork du *bitcoin-core* qui est le coeur de la technologie bitcoin cependant multichain s'est différenciée du bitcoin de par de nombreux aspects allant de sa nature de blockchain privée au stockage de données en passant par un algorithme de minage différent de la preuve de travail.

Elle fournit une interface en ligne de commande mais également une API simple qui permet son intégration dans de nombreux langages de programmation. De nombreuses fonctionnalités sont présentes telles que la gestion des permissions, la création de différentes configurations par blockchain ainsi que les streams de données. Cette annexe décrira

A.1 Fonctionnalités

Cette permière partie vise à introduire quelques points-clés concernant les fonctionnalités de Multichain.

Permissionné Comme toute blockchain privée, l'accès au réseau de Multichain nécessite en premier lieu une autorisation de se connecter pour tout nouveau membre. Cet accès est donné par le noeud initiateur de la blockchain.

Stream Le concept du stream permet d'utiliser la blockchain comme une base de donnée privée partagée sur le réseau. Il est utilisé pour le stockage de données en général, l'horodatage, l'archivage mais également la consultation des données stockées. En terme de structure, il s'agit d'une collection d'objets appelés *stream item* où chaque objet contient un horodatage, une ou plusieurs clés permettant de le retrouver, une signature digitale et bien évidemment la donnée stockée. Comme tout élément dans Multichain, un système de permissions existe permettant de désigner qui peut ou non lire et écrire dans le stream.

Evolutivité Multichain peut résoudre les problèmes d'évolutivité en terme d'utilisation des ressources réseau et de stockage grâce à un double mode de stockage qui consiste à enregistrer les données en mode *onchain (en-chaîne)* ou *offchain (hors-chaîne)*. Contrairement aux autres blockchain où les contenus des transactions sont stockés sur l'ensemble des noeuds, multichain ne réplique pas l'ensemble des données sur tous les noeuds. La taille des blocs est réduite car ces derniers n'incluent que les hash des données avec le mode offchain ou avec le mode onchain lorsque la taille de la donnée est trop importante.

Minage Le minage, qui est le processus correspondant à la création de nouveaux blocs, est exécuté par un groupe d'entités défini et autorisé par le ou les administrateurs du réseau blockchain.

A.2 Paramètres principaux

Les outils intégrés par défaut dans Multichain permettent de contrôler de nombreux paramètres de la blockchain tels que la taille maximale des blocs, le fonctionnement de l'algorithme de consensus ainsi que les permissions réseaux (Tableau A.1).

Tableau A.1 – Paramètres principaux de Multichain

Paramètre	*Description*	*Valeur par défaut*
target-block-time	Nombre moyen de secondes entre le minage des blocs. Il s'agit d'une valeur cible dont la blockchain cherche à se rapprocher au maximum.	15s
maximum-block-size	Nombre maximal d'octets dans chaque bloc. Cette limite permet d'éviter de surcharger le réseau avec des blocs ayant une taille trop importante.	8388608 (8mB)
maximum-chunk-size	Taille maximal d'un tronçon lors du stockage de données en mode offchain. Plus la valeur est faible, plus les fichiers seront découpés en un nombre important de tronçons.	1048576 (1mB)
mining-diversity	Proportion minimum de mineurs requise afin de pouvoir miner un bloc. Si la proportion de mineurs connectés est inférieure, alors plus aucun bloc ne pourra être miné et la blockchain sera donc bloqué.	0.5
mine-empty-rounds	Nombre de tours après lequel on arrête de miner des blocs s'il n'y a pas de nouvelles transactions soumises à la blockchain. Cela permet de réduire l'utilisation disque de la blockchain dans les périodes de faible activité.	10
mining-turnover	Une valeur de 0.0 implique un minage pure de type round-robin où les noeuds minent chacun leur tour tandis qu'une valeur de 1.0 implique une génération aléatoire des blocs entre les noeuds ayant l'uaotirsation de miner.	0.5
target-adjust-freq	Fréquence à laquelle la difficulté du minage est recalculée dans le cas de l'utilisation de la Preuve de travail (POW). Si la valeur est inférieure à 0, alors l'algorithme utilisé est le round-robin.	-1
pow-minimum-bits	Valeur initiale liée à la difficulté du minage lors de l'utilisation de la preuve de travail. Il s'agit du nombre de zéro requis en amont dans la valeur du hash du bloc.	8

A.3 Multichain API JSON

Afin de permettre son intégration dans divers langages de programmation, Multichain propose une API Json-RPC permettant d'interagir avec la blockchain. Cette dernière se présente sous la forme d'un ensemble de fonctions dont plusieurs exemples figurent dans le Tableau A.2.

Ces dernières peuvent être appelées via une requête HTTP POST vers le port d'écoute de l'API. L'extrait de Code A.1 montre le corps de la requête envoyée à l'API pour publier un objet dans un stream appelée *stream1*.

Tableau A.2 – Exemples de commandes de l'API Multichain

Fonction	*Paramètres*	*Description*
getblockchainparams	N/A	Permet de retourner la liste des paramètres de la blockchain
createkeypairs	N/A	Permet de créer un couple clé publique / privée pouvant être utilisée par un noeud. Cette fonction pourrait être utilisée par un "cold node" afin d'assigner des clés aux autres noeuds
publish	streamname, keys, data	Permet de publier un objet dans un stream donné avec une ou plusieurs clés qui permettront de le retrouver par la suite. Cette fonction retourne le hash de la transaction créée.
getstreamitem	txid	Permet de récupérer dans un stream l'objet ayant pour hash de la transaction la valeur *txid*. Cette fonction ne peut pas retourner plus d'un résultat car le hash de la transaction est une valeur unique.
liststreamkeyitems	streamname, keys	Permet de lister les objets dans un stream ayant pour clé celle passée en paramètre

Code A.1 – Exemple de requête à l'API Multichain en JSON

```
{
    "jsonrpc": "2.0",
    "method": "publish",
    "params": ["stream1", "TRACEA_STEP", {"json": {
        "location":"Line9-u-3",
        "process_desc":"Reduced discrete Internet solution"}}],
    "id": 1,
    "chain_name": "MyBlockchain"
}
```

A.4 Comparaison avec d'autres solutions blockchain

Dans la plupart des discussions sur les blockchains, la notion de « Smart contracts » (contrats intelligents) ne tarde pas à apparaître. Dans l'imaginaire populaire, les contrats intelligents automatisent l'exécution des interactions entre les parties, sans nécessiter d'intermédiaire de confiance. En exprimant les relations juridiques en code plutôt qu'en mots, ils promettent de permettre aux transactions d'avoir lieu directement et sans erreur, délibérée ou non.

D'un point de vue technique, un contrat intelligent est quelque chose de plus spécifique : un code informatique stocké dans une blockchain et qui définit les règles des transactions de cette chaîne. Cette description semble assez simple, mais derrière elle se cache une grande variation dans la façon dont ces règles sont exprimées, exécutées et validées. Lors du choix d'une plateforme blockchain pour une nouvelle application, la question « Cette plateforme prend-elle en charge les contrats intelligents ? » n'est pas la bonne question à se poser. Au lieu de cela, nous devons nous demander : « Quel type de contrats intelligents cette plate-forme prend-elle en charge ? »

Dans cette section, notre objectif est d'examiner certaines des principales différences entre les approches de contrats intelligents et les compromis qu'elles représentent. Quatre plates-formes de blockchain d'entreprise populaires prenant en charge une forme de code personnalisé en chaîne seront comparées. Tout d'abord, Hyperledger Fabric d'IBM , qui appelle ses contrats « **chaincode** ». Deuxièmement, la plateforme MultiChain, qui introduit les « smart filters » (filtres intelligents). Troisièmement, Ethereum (et ses spin-offs autorisés Quorum et Burrow), qui ont popularisé le nom de « contrat intelligent ». Et enfin, R3 Corda, qui fait référence à des « contrats » dans ses transactions. Malgré toutes les terminologies différentes, en fin de compte, toutes font référence à la même chose : un code spécifique à l'application qui définit les règles d'une chaîne.

A.4.1 Concepts de base

Nous considérons une application partagée par plusieurs organisations, basée sur une base de données sous-jacente. Dans une architecture centralisée traditionnelle, cette base de données est hébergée et administrée par un seul interlocuteur auquel tous les participants font confiance, même s'ils ne se font pas confiance. Les transactions qui modifient la base de données ne sont initiées que par des applications sur les systèmes de cette partie centrale, souvent en réponse à des messages reçus des participants. La base de données fait simplement ce qu'on lui dit parce que l'application est implicitement approuvée pour ne lui envoyer que les transactions qui ont du sens.

Les blockchains offrent un moyen alternatif de gérer une base de données partagée, sans intermédiaire de confiance. Dans une blockchain, chaque participant exécute un « noeud » qui contient une copie de la base de données et traite indépendamment les transactions qui la modifient. Les participants sont identifiés à l'aide de clés publiques ou adresses, chacune ayant une clé privée correspondante connue uniquement du propriétaire de l'identité. Alors que les transactions peuvent être créées par n'importe quel nœud, elles sont « signées numériquement » par la clé privée de leur initiateur afin de prouver leur origine.

Les noeuds se connectent les uns aux autres de manière peer-to-peer, propageant rapidement les transactions et les « blocs » dans lesquels elles sont horodatées et confirmées sur le réseau. La blockchain elle-même est littéralement une chaîne de ces blocs, qui forme un journal ordonné de chaque transaction historique. Un « algorithme de consensus » est utilisé pour s'assurer que tous les noeuds parviennent à un accord sur le contenu de la blockchain, sans nécessiter de contrôle centralisé.

En principe, toute application de base de données partagée peut être architecturée en utilisant une blockchain en son cœur. Mais cela crée un certain nombre de défis techniques qui n'existent pas dans un scénario centralisé :

Règles des transactions Si n'importe quel participant peut modifier directement la base de données, comment s'assurer qu'il respecte les règles de l'application ? Qu'est-ce qui empêche un utilisateur de corrompre le contenu de la base de données de manière intéressée ?

Déterminisme Une fois ces règles définies, elles seront appliquées plusieurs fois par plusieurs nœuds lors du traitement des transactions pour leur propre copie de la base de données. Comment s'assurer que chaque nœud obtient exactement le même résultat ?

Prévention des conflits En l'absence de coordination centrale, comment traiter deux transactions qui suivent chacune les règles de l'application, mais qui sont néanmoins en conflit l'une avec l'autre ? Les conflits peuvent provenir d'une tentative délibérée de déjouer le système, ou être le résultat innocent de la malchance et du timing.

Alors, d'où viennent les contrats intelligents, les filtres intelligents et le code blockchain ? Leur objectif principal est de travailler avec l'infrastructure sous-jacente d'une blockchain afin de résoudre ces défis. Les contrats intelligents sont l'équivalent décentralisé du code d'application : au lieu de s'exécuter dans un endroit central, ils s'exécutent sur plusieurs noeuds de la blockchain, créant ou validant les transactions qui modifient le contenu de cette base de données.

Commençons par les règles de transaction, le premier de ces défis, et voyons comment elles s'expriment respectivement dans Fabric, MultiChain, Ethereum et Corda.

A.4.2 Règles des transactions

Les règles de transaction remplissent une fonction spécifique dans les bases de données alimentées par la blockchain - limitant les transformations qui peuvent être effectuées sur l'état de cette base de données. Cela est nécessaire car les transactions d'une blockchain peuvent être initiées par n'importe lequel de ses participants, et ces participants ne se font pas suffisamment confiance pour leur permettre de modifier la base de données à volonté.

Voyons deux exemples de la raison pour laquelle les règles de transaction sont nécessaires. Imaginez d'abord une blockchain conçue pour agréger et horodater les documents PDF publiés par ses participants. Dans ce cas, personne ne devrait avoir le droit de supprimer ou de modifier des documents, car cela compromettrait l'objectif même du système - la persistance des documents. Deuxièmement, considérons une blockchain représentant un grand livre financier partagé, qui garde une trace des soldes de ses utilisateurs. Nous ne pouvons pas permettre à un participant de gonfler arbitrairement son propre solde ou de retirer l'argent des autres.

Entrées et sorties

Nos plateformes de blockchain reposent sur deux grandes approches pour exprimer les règles de transaction. Le premier, nommé « modèle d'entrée-sortie », est utilisé dans MultiChain et Corda. Ici, les transactions répertorient explicitement les lignes ou « états » de la base de données qu'elles suppriment et créent, formant un ensemble d'« entrées » et de « sorties » respectivement. La modification d'une ligne est exprimée comme l'opération équivalente à la suppression de cette ligne et à la création d'une nouvelle à sa place.

Étant donné que les lignes de la base de données ne sont supprimées que dans les entrées et créées uniquement dans les sorties, chaque entrée doit « passer » la sortie d'une transaction précédente. L'état courant de la base de données est défini comme l'ensemble des « sorties de transaction non dépensées » ou « UTXO », c'est-à-dire les sorties des transactions précédentes qui n'ont pas encore été utilisées. Les transactions peuvent également contenir des informations supplémentaires, appelées « métadonnées », « commandes » ou « pièces jointes », qui ne font pas partie de la base de données mais aident à définir leur signification ou leur objectif.

Compte tenu de ces trois ensembles d'entrées, de sorties et de métadonnées, la validité d'une transaction dans MultiChain ou Corda est définie par un code qui peut effectuer des calculs arbitraires sur ces ensembles. Ce code peut valider la transaction, ou bien retourner une erreur avec une explication correspondante. Vous pouvez considérer le modèle d'entrées-sorties comme un « inspecteur » automatisé tenant une liste de contrôle qui garantit que les transactions respectent chacune des règles. Si la transaction échoue à l'un de ces contrôles, elle sera automatiquement rejetée par tous les nœuds du réseau.

Il convient de noter que, malgré le partage du modèle d'entrée-sortie, MultiChain et Corda l'implémentent de manière très différente. Dans MultiChain, les sorties peuvent contenir des actifs et/ou des données au format JSON, texte ou binaire. Les règles sont définies dans des « filtres de transaction » ou des « filtres de flux », qui peuvent être définis pour vérifier toutes les transactions, ou uniquement celles impliquant des actifs ou des groupes de données particuliers. En revanche, un « état » de sortie Corda est représenté par un objet dans le langage de programmation Java ou Kotlin, avec des champs de données définis. Les règles de Corda sont définies dans des « contrats » qui sont attachés à des états spécifiques, et le contrat d'un état n'est appliqué qu'aux transactions qui contiennent cet état dans ses entrées ou ses sorties. Cela concerne le modèle de visibilité inhabituel de Corda, dans lequel les transactions ne peuvent être vues que par leurs contreparties ou celles dont elles affectent les transactions ultérieures.

Contrats et messages

La deuxième approche, que l'on nommera le « modèle de contrat-message », est utilisée dans Hyperledger Fabric et Ethereum. Ici, plusieurs « contrats intelligents » ou « chaincodes » peuvent être créés sur la blockchain, et chacun a sa propre base de données et son code associé. La base de données d'un contrat ne peut être

modifiée que par son code, plutôt que directement par des transactions blockchain. Ce modèle de conception est similaire à l'encapsulation du code et des données dans la programmation orientée objet.

Avec ce modèle, une transaction blockchain commence par un message envoyé à un contrat, avec quelques paramètres ou données facultatifs. Le code du contrat est exécuté en réaction au message et aux paramètres, et est libre de lire et d'écrire sa propre base de données dans le cadre de cette réaction. Les contrats peuvent également envoyer des messages à d'autres contrats, mais ne peuvent pas accéder directement aux bases de données de l'autre. Dans le langage des bases de données relationnelles, les contrats agissent comme des « procédures stockées » appliquées , où tous les accès à la base de données passent par un code prédéfini.

Fabric et Quorum, une variante d'Ethereum, compliquent cette image en permettant à un réseau de définir plusieurs *canaux* ou *états privés*. L'objectif est d'atténuer le problème de la confidentialité de la blockchain en créant des environnements séparés, dont chacun n'est visible que pour un sous-groupe particulier de participants. Bien que cela semble prometteur en théorie, en réalité, les contrats et les données de chaque canal ou État privé sont isolés de ceux des autres. Par conséquent, en termes de contrats intelligents, ces environnements sont équivalents à des blockchains distincts.

Exemple de règles

Voyons comment mettre en œuvre les règles de transaction pour un grand livre financier à un seul actif avec ces deux modèles. Chaque ligne de la base de données de notre grand livre comporte deux colonnes, contenant l'adresse du propriétaire et la quantité de l'actif possédé. Dans le modèle entrées-sorties, les transactions doivent satisfaire à deux conditions :

— La quantité totale d'actifs dans les sorties d'une transaction doit correspondre au total dans ses entrées. Cela empêche les utilisateurs de créer ou de supprimer de l'argent de manière arbitraire.
— Chaque transaction doit être signée par le propriétaire de chacune de ses entrées. Cela empêche les utilisateurs de dépenser l'argent des autres sans autorisation.

Ensemble, ces deux conditions suffisent à créer un système financier simple mais viable.

Dans le modèle contrat-message, le contrat de l'actif prend en charge un message « envoyer le paiement », qui prend trois paramètres : l'adresse de l'expéditeur, l'adresse du destinataire et la quantité à envoyer. En réponse, le contrat exécute les quatre étapes suivantes :

— Vérifiez que la transaction a été signée par l'expéditeur
— Vérifiez que l'expéditeur dispose de fonds suffisants
— Déduisez la quantité demandée de la ligne de l'expéditeur
— Ajoutez cette quantité à la ligne du destinataire

Si l'une des vérifications des deux premières étapes échoue, le contrat sera annulé et aucun paiement ne sera effectué.

Ainsi, les modèles d'entrée-sortie et de contrat-message sont des moyens efficaces de définir des règles de transaction et de protéger une base de données partagée. En effet, sur le plan théorique, chacun de ces modèles peut être utilisé pour simuler l'autre. En pratique cependant, le modèle le plus approprié dépendra de l'application en cours de construction. Chaque transaction affecte-t-elle peu ou beaucoup d'éléments d'information ? Doit-on être en mesure de garantir l'indépendance des transactions ? Chaque élément de données a-t-il un propriétaire clair ou y a-t-il un état global à partager ?

Ill n'est pas de notre ressort ici d'explorer comment les réponses devraient influencer un choix entre ces deux modèles. Mais en règle générale, lors du développement d'une nouvelle application blockchain, il vaut la peine d'essayer d'exprimer ses règles de transaction sous les deux formes et de voir laquelle correspond le plus naturellement. La différence s'exprimera en termes de : facilité de programmation, exigences de stockage et débit, et rapidité de détection des conflits.

Règles intégrées

En ce qui concerne les règles de transaction, il y a une façon dont MultiChain diffère spécifiquement de Fabric, Ethereum et Corda. Contrairement à ces autres plates-formes, MultiChain possède plusieurs abstractions intégrées qui fournissent des blocs de construction de base pour les applications basées sur la blockchain, sans obliger les développeurs à écrire leur propre code. Ces abstractions couvrent trois domaines couramment nécessaires : les autorisations dynamiques, les actifs transférables et le stockage des données.

Par exemple, MultiChain gère les autorisations de connexion au réseau, d'envoi et de réception de transactions, de création d'actifs ou de flux ou de contrôle des autorisations des autres utilisateurs. Plusieurs actifs fongibles peuvent être émis, transférés, retirés ou échangés en toute sécurité et de manière atomique. N'importe quel nombre de « flux » peut être créé sur une chaîne, pour publier, indexer et récupérer des données en chaîne ou hors chaîne au format JSON, texte ou binaire. Toutes les règles de transaction pour ces abstractions sont disponibles prêtes à l'emploi.

Lors du développement d'une application sur MultiChain, il est possible d'ignorer cette fonctionnalité intégrée et d'exprimer des règles de transaction à l'aide de filtres intelligents uniquement. Cependant, les filtres intelligents sont conçus pour fonctionner avec ses abstractions intégrées, en permettant de restreindre leur comportement par défaut de manière personnalisée. Par exemple, l'autorisation pour certaines activités peut être contrôlée par des administrateurs spécifiques, plutôt que le comportement par défaut où n'importe quel administrateur le ferait. Le transfert de certains actifs peut être limité dans le temps ou bien nécessiter une approbation supplémentaire au-delà d'un certain montant. Les données d'un flux particulier peuvent être validées pour s'assurer qu'elles se composent uniquement de structures JSON avec des champs et des valeurs obligatoires.

Dans tous ces cas, les filtres intelligents créent des exigences supplémentaires pour la validation des transactions, mais ne suppriment pas les règles simples qui y sont intégrées. Cela peut aider à relever l'un des principaux défis des applications blockchain : le fait qu'un bogue dans certains sur le code *on-chain* puisse avoir des conséquences désastreuses. Nous avons vu d'innombrables exemples de ce problème dans la blockchain publique d'Ethereum, notamment dans la disparition de The DAO et les bogues multisignatures de parité . Des enquêtes plus larges ont trouvé un grand nombre de vulnérabilités communes dans les contrats intelligents Ethereum qui permettent aux attaquants de voler ou de geler les fonds d'autres personnes.

Bien sûr, les filtres intelligents MultiChain peuvent également contenir des bugs, mais leurs conséquences sont plus limitées. Par exemple, les règles d'actif intégrées empêchent un utilisateur de dépenser l'argent d'un autre ou de faire disparaître accidentellement son propre argent, quelle que soit l'autre logique contenue dans un filtre intelligent. Si un bogue est trouvé dans un filtre intelligent, il peut être désactivé et remplacé par une version corrigée, tandis que l'intégrité de base du registre est protégée. Philosophiquement, MultiChain est plus proche des architectures de base de données traditionnelles où la plate-forme de base de données fournit un certain nombre d'abstractions intégrées telles que des colonnes, des tables, des index et des contraintes. Des fonctionnalités plus puissantes telles que les déclencheurs et les procédures stockées peuvent éventuellement être codées par les développeurs d'applications dans les cas où elles sont réellement nécessaires.

Tableau A.3 – Comparaison des règles de transaction entre Hyperledger, Multichain, Ethereum et Corda

Règles de transaction	*Fabric*	*Multichain*	*Ethereum*	*Corda*
Modèle	Contrat-message	Entrée-Sortie	Contrat-message	Entrée-Sortie
Intégrés	Aucun	Autorisations, actifs, streams	Aucun	Aucun

A.4.3 Déterminisme

Passons à la prochaine partie de notre confrontation. Quelle que soit l'approche que nous choisissons, les règles de transaction personnalisées d'une application blockchain sont exprimées sous forme de code informatique écrit par les développeurs d'applications. Et contrairement aux applications centralisées, ce code va être exécuté plus d'une fois et à plusieurs endroits pour chaque transaction. En effet, plusieurs nœuds

de blockchain appartenant à différents participants doivent chacun vérifier et/ou exécuter cette transaction pour eux-mêmes.

Cette exécution de code répétée et redondante introduit une nouvelle exigence rarement rencontrée dans les applications centralisées : le déterminisme. Dans le contexte du calcul, le déterminisme signifie qu'un morceau de code donnera toujours la même réponse pour les mêmes paramètres, peu importe où et quand il est exécuté. Ceci est absolument crucial pour le code qui interagit avec une blockchain car, sans déterminisme, le consensus entre les nœuds de cette chaîne peut s'effondrer de manière catastrophique.

Voyons à quoi cela ressemble dans la pratique, d'abord dans le modèle d'entrée-sortie. Si deux nœuds ont une opinion différente sur la validité d'une transaction, l'un acceptera un bloc contenant cette transaction et l'autre non. Étant donné que chaque bloc est explicitement lié à un bloc précédent, cela créera un « fork » permanent dans le réseau, avec un ou plusieurs nœuds n'acceptant pas l'opinion majoritaire sur l'ensemble du contenu de la blockchain à partir de ce moment. Les nœuds minoritaires seront coupés de l'état évolutif de la base de données, et ne pourront plus utiliser efficacement l'application.

Voyons maintenant ce qui se passe si le consensus échoue dans le modèle contrat-message. Si deux nœuds ont une opinion différente sur la manière dont un contrat doit répondre à un message particulier, cela peut entraîner une différence dans le contenu de leurs bases de données. Cela peut à son tour affecter la réponse du contrat aux messages futurs, y compris les messages qu'il envoie à d'autres contrats. Le résultat final est une divergence croissante entre la vue des différents nœuds sur l'état de la base de données. Le champ « state root » dans les blocs Ethereum garantit que toute différence dans les réponses des contrats conduit immédiatement à un fork de blockchain totalement catastrophique, plutôt que de risquer de rester caché pendant un certain temps.

Sources de non-déterminisme

Le non-déterminisme dans le code blockchain est donc clairement un problème. Mais si les éléments de base du calcul, comme l'arithmétique, sont déterministes, de quoi devons-nous nous inquiéter ? Eh bien, il s'avère que plusieurs choses :

— Les générateurs de nombres aléatoires, puisque par définition ceux-ci sont conçus pour produire un résultat différent à chaque fois.
— Vérifier l'heure actuelle, car les nœuds ne traiteront pas les transactions exactement au même moment et, dans tous les cas, leurs horloges peuvent être désynchronisées. (Il est toujours possible d'implémenter des règles dépendant du temps en faisant référence à des horodatages dans la blockchain elle-même.)
— Interroger des ressources externes telles qu'Internet, des fichiers de disque ou d'autres programmes exécutés sur un ordinateur. Il n'est pas garanti que ces ressources donnent toujours la même réponse et elles peuvent devenir indisponibles.
— Exécuter plusieurs morceaux de code dans des « threads » parallèles, car cela conduit à une « condition de concurrence » où l'ordre dans lequel ces processus se terminent ne peut pas être prédit.
— Effectuer des calculs en virgule flottante qui peuvent donner des réponses avec des différences même infimes sur différentes architectures de processeurs informatiques.

Nos quatre plateformes blockchain utilisent plusieurs approches différentes pour éviter ces pièges.

Exécution déterministe

Commençons par Ethereum, puisque son approche est la plus « pure ». Les contrats Ethereum sont exprimés dans un format spécial appelé *bytecode Ethereum*, qui est exécuté par la machine virtuelle Ethereum *EVM*. Ce code est compilé à partir d'un langage de programmation de type JavaScript appelé Solidity. Auparavant, d'autres langages étaient disponibles mais ont depuis été obsolètes. Le déterminisme est garanti par le fait que Solidity et Ethereum ne supporte aucune opération non déterministe.

Les filtres MultiChain et les contrats Corda choisissent une approche différente en adaptant les langages de programmation et les environnements d'exécution existants. MultiChain utilise JavaScript exécuté dans le moteur V8 de Google qui constitue également le coeur du navigateur Chrome et de la plate-forme Node.js mais avec les fonctions non-déterminisme désactivées. De même, Corda utilise Java ou Kotlin , qui sont tous deux compilés en *bytecode JAVA* qui s'exécute dans une machine virtuelle Java *JVM*. Pour l'instant, Corda utilise la JVM standard non déterministe d'Oracle, mais des travaux sont en cours pour intégrer une version déterministe. En attendant, les développeurs de contrats Corda doivent veiller à ne pas autoriser le non-déterminisme dans leur code.

Le principal avantage d'Ethereum est la minimisation des risques - une machine virtuelle conçue à cet effet est moins susceptible de contenir une source involontaire de non-déterminisme. Bien qu'un tel oubli puisse être corrigé par une mise à jour logicielle, il serait perturbateur pour toute chaîne qui aurait eu la malchance de le rencontrer. Le problème d'Ethereum, cependant, est que Solidity et l'EVM constituent un écosystème minuscule et naissant dans le contexte plus large des langages de programmation et des environnements d'exécution. En comparaison, JavaScript et Java sont les deux principaux langages sur Github , exécutés sur des milliards d'appareils numériques et dont les durées d'exécution ont été optimisées au fil des décennies. C'est probablement la raison pour laquelle la blockchain publique Ethereum envisage une transition verseWASM , un fork déterministe de la norme émergente WebAssembly.

Déterminisme par endossement

En matière de déterminisme, Hyperledger Fabric adopte une approche complètement différente. Dans Fabric, lorsqu'un nœud client souhaite envoyer un message à un code blockchain, il envoie d'abord ce message à certains noeuds appelé *endosseurs*. Chacun de ces nœuds exécute le code blockchain indépendamment, se forgeant ainsi une opinion sur l' effet du message sur la base de données de ce code blockchain. Ces avis sont renvoyés au client accompagnés d'une signature numérique qui constitue un avenant formel. Si le client reçoit suffisamment de mentions du résultat escompté, il crée une transaction contenant ces mentions et la diffuse pour inclusion dans la chaîne.

Afin de garantir le déterminisme, chaque morceau de chaincode a une *politique d'approbation* qui définit exactement le niveau d'approbation requis pour rendre ses transactions valides. Par exemple, la politique d'un code blockchain peut indiquer qu'il est nécessaire d'obtenir l'approbation d'au moins la moitié des noeuds constituant la blockchain. Un autre pourrait nécessiter l'approbation d'au mùoins un tiers des parties de confiance. Dans tous les cas, chaque noeud peut vérifier indépendamment si les approbations nécessaires ont été reçues.

Pour clarifier la différence, le déterminisme dans la plupart des plates-formes blockchain est basé sur la question : **Quel est le résultat de l'exécution de ce code sur ces données ?** et nous devons être absolument sûrs que chaque noeud répondra à cette question de manière identique. En revanche, le déterminisme dans Fabric repose sur une question différente : **Est-ce que suffisamment d'endosseurs sont d'accord sur le résultat de l'exécution de ce code sur ces données ?** Répondre à cette interrogation est une question de comptage assez simple, et il n'y a pas de place pour le non-déterminisme.

Le déterminisme par approbation de Fabric a un certain nombre de conséquences intéressantes. Premièrement, le code blockchain peut être écrit dans de nombreux langages de programmation différents car ceux-ci n'ont pas besoin d'être adaptés au déterminisme (Go, Java et JavaScript sont actuellement pris en charge). Deuxièmement, le code blockchain peut être caché à certains des participants d'une blockchain car il ne doit être exécuté que par les clients et les endosseurs. La base de données elle-même est globalement visible. Enfin et surtout, Fabric chaincode peut faire des choses qui sont interdites dans d'autres environnements de blockchain comme vérifier la météo à l'aide d'une API Web en ligne. Dans le pire des cas, où chaque endosseur obtient une réponse différente de cette API, le client ne parviendra pas à obtenir suffisamment d'endossements pour un résultat particulier et aucune transaction n'aura lieu. Il convient de noter que les membres de l'équipe Fabric recommandent d'utiliser une logique déterministe à l'intérieur du code blockchain afin d'éviter les surprises.

Quel prix Fabric paie-t-il pour cette flexibilité ? Si le but d'une blockchain est de supprimer les intermédiaires d'une application basée sur une base de données partagée alors la dépendance de Fabric vis-à-vis des endosseurs s'éloigne de cet objectif. Pour les participants de la blockchain, il ne suffit plus de suivre les règles du code blockchain. Ces derniers ont également besoin que d'autres noeuds attestent qu'ils suivent bel et bien les règles. Un sous-ensemble malveillant d'endosseurs pourrait approuver les modifications de la base de données qui ne suivent pas du tout le code blockchain. Cela donne aux endosseurs beaucoup plus de pouvoir que les validateurs des blockchains classiques qui peuvent censurer les transactions mais ne peuvent pas violer les règles de la blockchain. Les développeurs d'applications blockchain doivent décider si ce compromis a du sens dans leur cas particulier.

Tableau A.4 – Comparaison de l'approche déterministe entre Hyperledger Fabric, Multichain, Ethereum et Corda

Déterminisme	*Fabric*	*Multichain*	*Ethereum*	*Corda*
Modèle	Avenants	Temps d'exécution adapté	Machine virtuelle dédiée	Temps d'exécution adapté
Langages	Go, Java, JavaScript	JavaScript	Solidity	Java, Kotlin
Visibilité du code	Contre-parties, endosseurs	Blockchain	Blockchain	Contre-parties + dépendants
Forcé	Non	Oui	Oui	Non (pour l'instant)

A.4.4 Prévention des conflits

Jusqu'à présent, nous avons expliqué comment différentes plates-formes de blockchain expriment des règles de transaction dans le code et comment elles garantissent de manière déterministe que chaque nœud applique ces règles de manière identique. Il est maintenant temps de parler d'un troisième aspect : **Comment chaque plate-forme gère-t-elle la possibilité que deux transactions, valables en elles-mêmes, entrent en conflit ?** Dans l'exemple le plus simple, imaginez qu'Alice a 10€ dans un grand livre financier et diffuse deux transactions : l'une envoyant 8€ à Bob et l'autre envoyant 7€ à Charlie. De toute évidence, une seule de ces transactions peut être autorisée à réussir.

Deux modèles

Nous pouvons commencer par regrouper l'approche de MultiChain et de Corda face à ce problème. Comme on l'a vu précédemment, ces deux solutions utilisent un modèle d'entrée-sortie pour représenter les transactions et leurs règles dans lequel chaque entrée de transaction utilise une sortie de transaction précédente. Cela conduit à un principe simple pour éviter les conflits : chaque produit ne peut être dépensé qu'une seule fois. Les filtres MultiChain et les contrats Corda peuvent s'appuyer sur leurs plates-formes respectives pour appliquer cette restriction de manière absolue. Étant donné que les 10€ d'Alice sont représentés par une sortie de transaction précédente, cette règle de dépense unique l'empêche automatiquement de les envoyer à Bob et à Charlie.

Malgré cette similitude, il est important de souligner une différence essentielle dans la manière dont MultiChain et Corda préviennent les conflits. Dans MultiChain, chaque nœud voit chaque transaction et peut donc vérifier indépendamment que chaque sortie n'est dépensée qu'une seule fois. Toute transaction qui effectue une double dépense par rapport à une transaction précédemment confirmée sera instantanément et automatiquement rejetée. En revanche, dans Corda, il n'y a pas de blockchain globale, donc des *notaires* sont tenus d'éviter ces doubles dépenses. Chaque état de sortie Corda est attribué à un notaire qui doit signer toute transaction dépensant cette sortie, confirmant ainsi qu'elle n'a pas été dépensée auparavant. Les participants d'une blockchain doivent faire confiance aux notaires pour suivre cette règle honnêtement et les notaires malveillants peuvent causer des dommages à volonté. Comme pour les avenants dans Fabric, cette *dépense unique en tant que service* présente des avantages en termes de confidentialité mais réintroduit des intermédiaires ce qui est à contre-courant du gain de la blockchain. Il est important de préciser que les notaires Corda peuvent être gérés par des groupes de participants à l'aide d'un algorithme de consensus de façon à ce que l'intégrité de la chaîne puisse être protégée contre les mauvais acteurs individuels.

Passons à Ethereum. Pour rappel, Ethereum utilise des contrats et des messages plutôt que des entrées et des sorties. Par conséquent, les conflits de transaction tels que les deux paiements d'Alice ne sont pas immédiatement visibles pour le moteur de blockchain. Au lieu de cela, ils sont détectés et bloqués par le contrat qui traite les transactions, après confirmation de leur commande sur la chaîne. Lors du traitement de chacun des paiements d'Alice, le contrat vérifie si son solde est suffisant. Si la transaction payant 8€ à Bob vient en premier, elle sera traitée comme d'habitude, laissant Alice avec 2€ sur son compte. En conséquence, lorsque le contrat traite la deuxième transaction en payant 7€ à Charlie, il constate qu'Alice n'a pas les fonds nécessaires et la transaction avorte.

Résultats contre contrats

Jusqu'à présent, nous avons vu deux techniques différentes pour éviter les transactions conflictuelles : les sorties à dépense unique dans MultiChain et Corda puis la vérification basée sur des contrats dans Ethereum. Une question se pose à savoir : **Laquelle de ces deux solutions est la meilleure ?**

Afin d'aider à répondre à cette question, considérons un exemple de compte qui détient 100€ au nom de Gavin et Helen, et permet à l'un d'eux de dépenser cet argent de manière indépendante. Gavin ordonne à sa demande de payer 80€ à Donna, et quelques secondes plus tard, Helen veut envoyer 40€ à Edward. Comme il n'y a pas suffisamment de fonds pour les deux paiements, ces transactions seraient inévitablement en conflit. Dans le cas où les deux transactions sont diffusées, le résultat sera déterminé par celui qui entrera en premier dans la chaîne. Contrairement à l'exemple précédent d'Alice, ce conflit est accidentel puisque personne n'essaie d'enfreindre les règles de l'application : il s'agit simplement d'un problème lié au temps.

En considérant la probabilité que ce conflit se produise, la question clé est la suivante : **une fois que Gavin aura envoyé sa transaction, combien de temps faudra-t-il au nœud d'Helen pour savoir que son paiement pourrait échouer ?** Plus cette période de temps sera courte, moins Helen sera susceptible de tenter ce paiement, lui évitant ainsi la surprise de voir sa transaction rejetée.

Avec le modèle d'entrée-sortie, tout conflit entre les transactions est directement visible sur la plate-forme blockchain puisque les deux transactions tenteront explicitement de dépenser la même sortie précédente. Dans MultiChain, cela se produit dès que la transaction de Gavin s'est propagée au noeud d'Helen, généralement en moins d'une seconde. Avec Corda, le notaire de sortie refusera la demande de signature de la transaction d'Helen puisqu'il a déjà signé celle de Gavin donc Helen saura instantanément que son paiement échouera. Si le notaire Corda est lui-même distribué, il se peut qu'il doive attendre quelques secondes pour obtenir une réponse. Dans tous les cas, il n'est pas nécessaire d'attendre qu'une transaction soit confirmée et propagée dans la blockchain.

Dans le cas d'Ethereum, il n'y a aucun moyen immédiat pour la plate-forme blockchain de savoir qu'un conflit va se produire. Bien que le noeud d'Helen puisse voir la transaction de Gavin sur le réseau, il ne peut pas savoir comment cela affectera la propre transaction d'Helen car de son point de vue, ce sont simplement deux messages envoyés au même contrat. Une fois la commande finale des transactions en conflit confirmée sur la blockchain, le noeud d'Helen recalculera le résultat réel au lieu du résultat attendu et son application mettra à jour son affichage en conséquence. En attendant, Gavin et Helen seront laissés dans l'ignorance.

Mais il ne faut pas en conclure que le modèle entrée-sortie fonctionne toujours le mieux. Prenons une variante de notre exemple de scénario, où Gavin et Helen demandent tous deux des paiements inférieurs de 40€ à partir du solde initial de 100€, exactement au même moment. Dans le modèle d'entrée-sortie, ces transactions seraient en conflit, car elles dépensent toutes les deux la même ligne de base de données contenant ces 100€ et un seul des paiements réussirait. Mais dans Ethereum, les deux transactions seraient traitées avec succès, quelle que soit leur commande finale puisque le compte contient suffisamment de fonds pour les deux. Dans ce cas, Ethereum remplit plus fidèlement les intentions de Gavin et Helen.

Ensembles de lecture-écriture

Enfin, dans Fabric, dont l'approche basée sur l'endossement est un hybride de ces deux techniques : lorsqu'un nœud client Fabric souhaite envoyer un message à un contrat, il demande d'abord à certains noeuds l'approbation d'exécuter ce message en son nom. Les noeuds d'approbation le font de la même manière qu'Ethereum c'est à dire en exécutant le contrat sur leur base de données locale mais ce processus est observé plutôt qu'immédiatement appliqué. Chaque endosseur enregistre l'ensemble des lignes qui seraient lues et écrites, notant également la version exacte de ces lignes à ce moment précis. Cet *ensemble de lecture-écriture* de lignes versionnées est explicitement référencé dans l'endossement et inclus dans la transaction diffusée par le client.

Les conflits entre les transactions Fabric sont résolus une fois leur commande finalisée dans la chaîne. Chaque nœud traite chaque transaction indépendamment, en vérifiant les politiques d'approbation et en appliquant les modifications de base de données spécifiées. Cependant, si une transaction lit ou écrit une version de ligne de base de données qui a déjà été modifiée par une transaction précédente, cette deuxième transaction est ignorée. Pour revenir aux paiements contradictoires d'Alice à Bob et Charlie, ces deux transactions liront et modifieront la même version de ligne, contenant les 10€ avec lesquels Alice a commencé. Ainsi, la deuxième transaction sera automatiquement et en toute sécurité abandonnée.

L'approche de Fabric en matière de résolution de conflits fonctionne très bien, mais en termes de performances et de flexibilité, elle combine le pire des deux modèles précédents. Étant donné que les endossements convertissent les transactions en ensembles de lecture-écriture spécifiques, les paiements simultanés mais compatibles de 40€ de Gavin et Helen entraîneraient un conflit qu'Ethereum évite. Cependant, Fabric ne bénéficie pas de l'avantage de vitesse du modèle d'entrée-sortie, car les endosseurs exécutent des contrats par rapport à la version la plus récente de la base de données confirmée par la blockchain, ignorant les transactions non confirmées. Ainsi, si Helen initie son paiement quelques secondes après Gavin, mais avant que celui-ci ne soit confirmé sur la blockchain, Fabric créera des transactions conflictuelles qu'un modèle d'entrée-sortie pur évite.

Tableau A.5 – Gestion de la prévention des conflits entre Hyperledger Fabric, Multichain, Ethereum et Corda

Prévention des conflits	*Fabric*	*Multichain*	*Ethereum*	*Corda*
Modèle	Ensembles de lecture-écriture	Dépense unique	Contrôles contractuels	Dépense unique
Vérification	Indépendant	Indépendant	Indépendant	Notaires de confiance
Performance (rapidité)	~10 secondes (confirmation)	~1s (propagation)	~10s (confirmation)	0~5s (notaire)

A.4.5 Synthèse

Nous avons passé en revue de nombreuses façons différentes dont Multichain, Ethereum, Fabric et Corda relèvent les principaux défis des *smart contracts*. Et chaque plate-forme a des réponses différentes à nos trois questions fondamentales :

— **Comment les règles de transaction sont-elles représentées ?**
— **Comment le code est-il exécuté de manière déterministe ?**
— **Comment prévenir les conflits ?**

Chaque plate-forme représente un compromis multidirectionnel complexe entre flexibilité, simplicité, performance, désintermédiation, sécurité et confidentialité. Ainsi, le choix de la plate-forme pour une application particulière doit commencer par une compréhension détaillée du modèle de confiance de cette application, des types de transactions qu'elle implique et de leurs modèles de conflit probables.

B

Documentation Technique Prototype Multichain

Les différents résultats présentés tout au long de ce manuscrit concernant le prorotype blockchain telles que les mesures liées au minage ont nécessité le développement d'outils connexes dont les fonctionnalités seront décrites tout au long de cette annexe.

B.1 Architecture

La première difficulté dans la représentation de l'architecture blockchain réside dans son aspect distribué. En effet, développer un prototype blockchain implique d'utiliser une technologie permettant de pouvoir déployer de façon simple plusieurs noeuds en sachant que chaque noeud représente une instance multichain. Il est donc nécessaire de pouvoir déployer plusieurs instances rapidement avec des configurations différentes aussi bien au niveau applicatif qu'au niveau réseau. Ces capacités font justement parties de la technologie Docker d'où son choix comme technologie de base pour le développement du prototype.

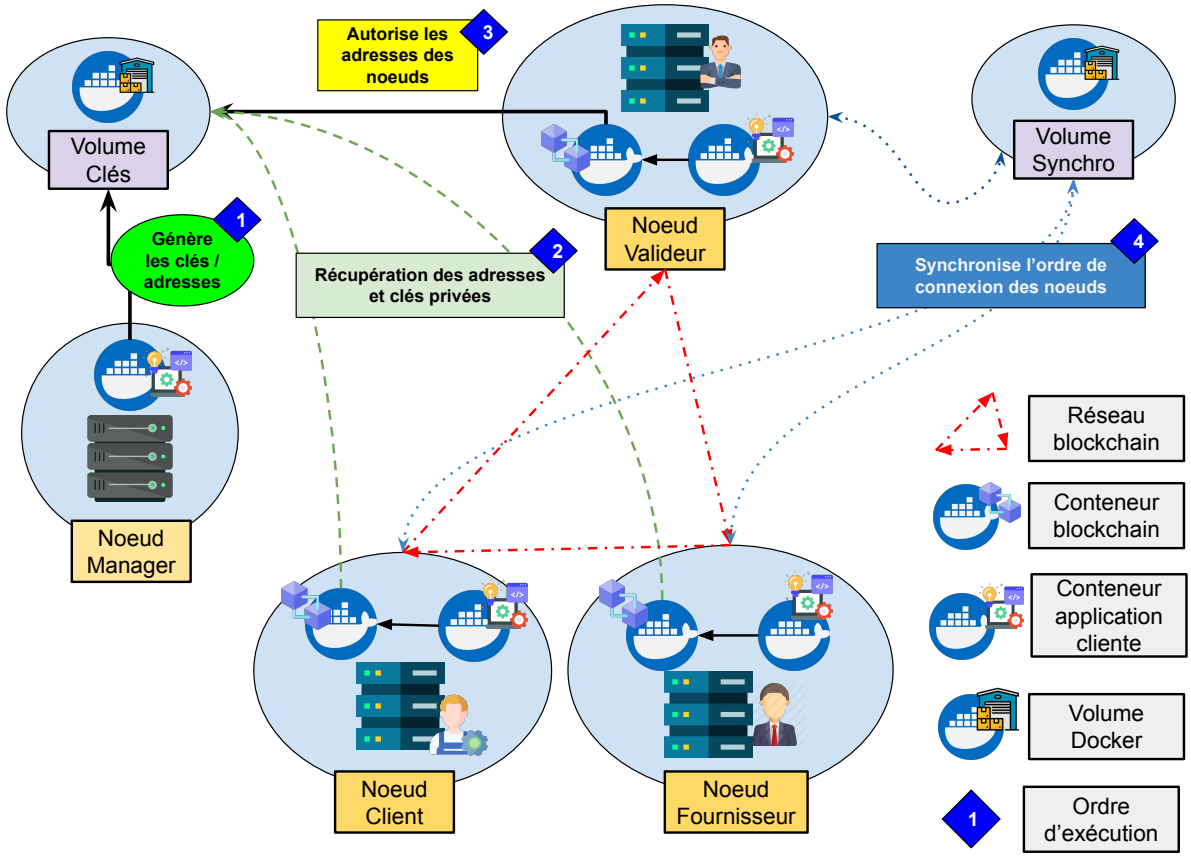


Figure B.1 – Architecture logicielle du prototype

La Figure B.1 présente justement cette architecture où on peut voir que chaque noeud est représenté par deux conteneurs Docker : un conteneur contenant une instance Multichain ainsi qu'un conteneur avec l'application client permettant d'interagir avec l'API Multichain. Cette application client a été développée en *Python*.

Chaque noeud possède un rôle bien défini dans l'architecture et une certaine hiérarchie existe c'est pourquoi il est nécessaire pour eux de pouvoir s'échanger des données notamment lors de la phase précédant l'initialisation de la blockchain. C'est dans ce contexte que les volumes Docker ont été utilisés étant donné que ces derniers peuvent être montés simulatanément sur plusieurs conteneurs permettant ainsi le partage d'informations. Par exemple, le *Volume Clés* permet de stocker les clés privées et publiques générés par le noeud Gestionnaire. Etant donné qu'il s'agit d'un réseau distribué et que la blockchain ne peut être initialisée que par le Valideur, alors il paraît évident qu'une synchronisation et un ordre de démarrage entre les noeuds doit être respectée : ce rôle est assuré par le *Volume Synchro*.

B.1.1 Configuration et déploiement des noeuds

Configuration

Grâce à Docker, il est possible de mettre en place une configuration distincte pour chaque conteneur par l'intermédiaire des variables d'environnement. Le Tableau B.1 illustre la liste des variables d'environnement utilisées dans le conteneur blockchain du Valideur.

Tableau B.1 – Description des variables d'environnement du conteneur blockchain (noeud Valideur)

Nom	*Description*	*Exemple*
NODE_ROLE	Nom du rôle assuré par le noeud. Plusieurs noeuds peuvent avoir le même rôle.	validator
NODE_ID	Identifiant unique du noeud	validator1
MULTICHAIN_DATADIR	Emplacement du répertoire Multichain	/multichain
KEYPAIRS_DIR	Emplacement du *Volume Clés*	/tmp/keypairs
SYNCHRO_DIR	Emplacement du *Volume Synchro*	/tmp/synchro
CHAINNAME	Nom de la blockchain	chain1
NETWORK_PORT	Port réseau utilisé pour les communications entre les noeuds	7000
RPC_USER	Login pour se connecter à l'API Multichain en RPC	rpcuser
RPC_PASSWORD	Mot de passe pour se connecter à l'API Multichain en RPC	secretpassword
RPC_PORT	Port d'écoute utilisée par l'API Multichain	8000
RPC_ALLOW_IP	Liste des adresses IP autorisées à se connecter à l'API (par défaut localhost uniquement)	localhost
CUSTOMER_CAN_MINE	Booléen permettant de définir si on accorde ou non le droit de miner au noeud Client	false | true
PROVIDER_CAN_MINE	Booléen permettant de définir si on accorde ou non le droit de miner au noeud Fournisseur	false | true
WAIT_FOR_NODE	Définit le noeud à attendre avant de démarrer (pour ordonner la séquence de démarrage)	manager-node
PARAM_MINING_DIVERSITY	Permet de définir la valeur du paramètre mining-diversity dans Multichain	mining-diversity | 0.5
PARAM_TARGET_BLOCK_TIME	Permet de définir la valeur du paramètre target-block-time dans Multichain	target-block-time | 30
PARAM_MAXIMUM_CHUNK_SIZE	Permet de définir la valeur du paramètre maximum-chunk-size dans Multichain	maximum-chunk-size | 1024
PARAM_TARGET_ADJUST_FREQ	Permet de définir la valeur du paramètre target-adjust-freq dans Multichain (pour changer l'algorithme de minage)	target-adjust-freq | -1

Les noeuds Client et Fournisseur ne possèdent qu'une variable supplémentaire appelée **MASTER_NODE** qui est l'adresse IP du noeud Valideur étant donné qu'ils ont besoin de ce dernier pour se connecter à la blockchain.

Déploiement

Afin de déployer l'ensemble des configurations liées aux noeuds, on utilise docker-compose qui est un outil associé à Docker permettant de déployer de façon simple des configurations d'applications multi-conteneurs ce qui est le cas de l'architecture blockchain. La configuration de docker-compose consiste à fournir un fichier dans lequel on indique les paramètres de l'ensemble des conteneurs à déployer.

Le Code B.1 montre un extrait de la configuration utilisée par exemple pour déployer le conteneur blockchain du Valideur où on retrouve plusieurs sections :

- **networks** : Réseau auquel appartient le noeud (tous les noeuds doivent appartenir au même réseau)
- **ports** : Liste des ports à rediriger (entre l'hôte et le conteneur par exemple)
- **volumes** : Volume à monter au niveau du conteneur tels que les volumes *Clés*, *Synchro*
- **expose** : Section contenant la liste des ports à ouvrir au niveau du conteneur (port pour la blockchain ainsi que port pour l'API RPC
- **environment** : Section où on insère la liste des variables d'environnement à définir ainsi que leurs valeurs

Code B.1 – Extrait de la configuration docker-compose du noeud valideur

```
validator1-node-chain:
    build: ./node-validator/chain
    container_name: chain-${VALIDATOR_NODE_ID}
    networks:
      - chain_net
    ports:
      - ${VALIDATOR_PORT_RPC_HOST}:${GL_CHAIN_RPC_PORT}
    volumes:
      - ${GL_VOLUME_SYNCHRO}:${GL_DIR_SYNCHRO}
      - ${GL_VOLUME_KEYPAIRS}:${GL_DIR_KEYPAIRS}:ro
      - ${GL_VOLUME_MULTICHAIN}:${GL_DIR_MULTICHAIN}
    expose:
      - ${GL_CHAIN_NETWORK_PORT}
      - ${GL_CHAIN_RPC_PORT}
    environment:
      NODE_ROLE: ${VALIDATOR_NODE_ROLE}
      KEYPAIRS_DIR: ${GL_DIR_KEYPAIRS}
      CHAINNAME: ${GL_CHAIN_NAME}
      MULTICHAIN_DATADIR: ${VALIDATOR_MULTICHAIN_DIR}
      NETWORK_PORT: ${GL_CHAIN_NETWORK_PORT}
      PARAM_TARGET_BLOCK_TIME: ${GL_CHAINP_TARGET_BLOCK_TIME}
```

B.1.2 Initialisation de la blockchain

Génération des adresses, clés privées et publiques

Il s'agit de la phase où le noeud Gestionnaire prépare pour chaque noeud les clés privées pet publiques permettant à Multichain de générer les adresses ainsi qu'à signer les blocs minés par le noeud en question. Pour cela, le noeud gestionnaire possède une application cliente également développée en python permettant d'effectuer cela. Le Code B.2 illustre la fonction de génération des adresses des noeuds par le noeud Gestionnaire. Cette dernière crée 3 fichiers dans le *Volume Clés* à savoir la valeur de l'adresse mais également le couple clé publique / clé privée utilisée par Multichain pour la signature des transactions et des blocs.

```python
def generate_address(node_id, key_role, chainname):
    chaindir = os.path.join(KEYPAIRS_DIR, chainname)
    services.init_services(read_rpc_conf(COLD_CHAINNAME, True))
    keypair = services.wallet.createkeypairs(1)
    write_key_tofile(node_id, "addr", key_role, keypair[0].get("address"), chaindir)
    write_key_tofile(node_id, "priv", key_role, keypair[0].get("privkey"), chaindir)
    write_key_tofile(node_id, "pub", key_role, keypair[0].get("pubkey"), chaindir)
```

Code B.2 – Génération des adresses des noeuds par le noeud Gestionnaire

Récupération des adresses et clés privées

Chaque conteneur récupère ensuite la clé privée qui lui est destinée pour l'intégrer dans sa configuration Multichain. Le conteneur va donc importer le fichier généré depuis le *Volume Clés* comme le montre le Code B.3.

```
# KEYPARDS_DIR => "Volume Clés"
multichain-cli $CHAINNAME -datadir="$MULTICHAIN_DATADIR"
importprivkey $(head -n 1 $KEYPAIRS_DIR/$CHAINNAME/$NODE_ID
.connect.priv)
```

Code B.3 – Récupération de la clé privée générée par un noeud tiers

Autoriser les adresses des noeuds

Le noeud Valideur initialise la blockchain et donne aux différents noeuds le droit de se connecter. Cela se traduit par la récupération des adresses des noeuds générés dans le *Volume Clés* comme on le voit sur le Code B.4.

```
multichain-cli $CHAINNAME -datadir="$MULTICHAIN_DATADIR"
grant $(head -n 1 $KEYPAIRS_DIR/$CHAINNAME/customer.connect
.addr) connect
```

Code B.4 – Récupération de la clé privée générée par un noeud tiers

Synchronisation des noeuds et ordre de démarrage

La séquence de démarrage des noeuds est la suivante : Gestionnaire, Valideur puis Client et Fournisseur. Afin d'assurer ce fonctionnement, chaque noeud signale son statut via la génération d'un fichier dans le *Volume Synchro*. La synchronisation consiste à attendre que le fichier soit généré par le noeud précédent afin d'indiquer qu'il est prêt comme le montre le Code B.5.

```
# En attente que le noeud soit prêt
while [ ! -f "$SYNCHRO_DIR/$WAIT_FOR_NODE.ready" ];
do sleep 1; done

# Indique le statut "Prêt"
touch "$SYNCHRO_DIR/$NODE_ID.ready"
```

Code B.5 – Fonctionnement de la synchronisation entre les noeuds

B.1.3 Pré-Chargement de données

La proposition d'architecture blockchain pour la traçabilité induit la création de certains éléments à l'avance dans la blockchain tels que les smart filters ou encore les streams pour stocker les fichiers, les signatures c'est pourquoi un système de pré-chargement des données a été mis en place afin de créer ces éléments automatiquement. Ce dernier a été intégré dans l'application cliente et fonctionne via un fichier de paramétrage en *JSON* comme on peut voir en exemple sur le Code B.6.

```
{
      "action": "create",
      "category": "stream",
      "data": {
        "name": "SIGNATURES",
        "options": {},
        "details": { "desc": "Stream pour stocker les signatures" }
      }
}
```

Code B.6 – Fichier de configuration pour créer automatiquement le stream Signatures

B.2 Mesures, performances et benchmark

Dans le cadre de l'évaluation des performances du prototype et de l'influence de certains paramètres sur le fonctionnement de la blockchain, un outil dédié à la mesure de ces indicateurs a été mis en place dans l'application cliente. Ce dernier sera référencé sous le terme *benchmark* tout au long de cette section.

B.2.1 Récupération des mesures

Lors de la récupération des statistiques au niveau du minage, la principale mesure utilisée fut le taux de CPU utilisé. Cette dernière a été récupérée grâce à Docker et l'API *docker-stats* fournie avec ce dernier. Cette API n'est cependant disponible qu'au niveau de l'hôte et non depuis le conteneur or le benchmark est lancé depuis le conteneur de l'applicaation cliente. Afin de rendre cette information accessible aux conteneurs, l'idée a été d'exporter les statistiques mesurées par Docker dans des fichiers stockés dans un dossier monté sur le conteneur de l'application cliente.

Le Code B.7 illustre la récupération de la consommation CPU par Docker où on filtre en fonction du noeud que l'on recherche (ici *validator1*) et de la statistique recherchée (CPU). Le résultat est ensuite exporté dans un fichier.

```
while true; do
    DOCKER_STATS=$(docker stats --no-stream)
    echo "$DOCKER_STATS" | grep "validator1"
    | awk '{ gsub("%","",$3); print $3}'
    | tr -d '\n' > tmp/stats/validator1.cpu
done
```

Code B.7 – Récupération du Taux d'utilisation CPU mesuré par Docker

Le fichier précédemment généré est stocké dans un dossier */tmp/stats* monté sur le conteneur de l'application cliente comme on le voit dans le Code B.8.

```
validator1-node-client:
    volumes:
        - /tmp/stats:/tmp/stats
```

Code B.8 – Montage du dossier permettant d'accéder aux statistiques mesurées par Docker

B.2.2 Lecture des mesures dans l'application cliente

Les statistiques mesurées par Docker sont désormais stockées dans des fichiers, il suffit donc de les lire lors de l'exécution du benchmark. Le Code B.9 présente un extrait de la fonction **get_node_stats** qui permet de lire les statistiques (CPU, Mémoire) d'un noeud donné.

```
def get_node_stats(self, node_id):
        cpu = self.get_stat_value(
            node_id=node_id, stat_type="cpu") / 100
        mem = self.get_stat_value(node_id=node_id, stat_type="mem")
        return {
            "{0}_cpu".format(node_id): cpu,
            "{0}_mem".format(node_id): mem
        }
```

Code B.9 – Récupération des stats Docker dans le benchmark

B.2.3 Principe de fonctionnement du benchmark

Dans le contexte de la blockchain, le benchmark va consister à envoyer un certain nombre de transactions appelé *nombre d'itérations* et à évaluer à chaque fin de transaction la valeur certains indicateurs tels que le taux de CPU, la taille de la blockchain, la détection du minage d'un bloc et si tel est le cas l'identité du noeud mineur etc..

Création d'un profil de benchmark

Différentes fonctions de benchmark ont été créées dans le prototype notamment pour faire des mesures en fonction du type de données (brutes ou fichiers) ou encore en fonction du niveau de confidentialité (chiffrement ou non des données). La fonction présentée dans le Code B.10 illustre une fonction de benchmark standard où l'on peut choisir le nombre de transactions, la taille individuelle des données ainsi que l'utilisation ou non du chiffrement des données.

```python
def start_data_simu(simuname, samplesize, nbiterations,
    encrypted):
    bench = Benchmark(name=simuname)
    folderpath = os.path.join(DATAFILES_DIR, "json", samplesize)
    filelist = os.listdir(folderpath)
    for i in range(nbiterations):
        filename = filelist[i]
        filepath = os.path.join(folderpath, filename)
        json_obj = json.loads(read_text(filepath))
        # Envoi de la transaction et mesure des statistiques
        item = bench.measure_perf(
            func=services.data.save_data,
            data_type="MEASURES_DATA",
            keys=["MEASURES", "MEASURES_DATA"],
            obj_data=json_obj,
            encrypted=encrypted,
        )
        bench.show()
    bench.export_csv()
```

Code B.10 – Benchmark pour la soumission de données brutes à la blockchain

Création de la commande

Afin de permettre l'utilisation du benchmark en mode *CLI (Command-Line Interface)*, il est nécessaire de définir une fonction appelant la fonction du benchmark tout en lui passant un certain nombre de paramètres tels que la taille des données, le nombre d'itérations ainsi que le recours ou non au chiffrement. L'extrait de Code B.11 montre la commande associée à la fonction de benchmark présentée dans le paragraphe précédent.

```python
@simu_bp.cli.command("data")
@click.option("-s", "--samplesize", type=str, default="1kb")
@click.option("-q", "--nbiterations", type=int)
@click.option("-e", "--encrypted", is_flag=True)
def cmd_data_simu(samplesize, nbiterations, encrypted):
    simuname = "data-{0}-{1}-encr{2}".format(samplesize, nbiterations, encrypted)
    start_data_simu(
        samplesize=samplesize,
        simuname=simuname,
        nbiterations=nbiterations,
        encrypted=encrypted,
    )
```

Code B.11 – Commande permettant d'exécuter le benchmark en mode CLI

Exécution du benchmark

Il est donc désormais possible de lancer le benchmark en mode *CLI*. Le Code B.12 montre la commande permettant de lancer un benchmark dans le contexte suivant : 2000 itérations basées sur un échantillon de données dont la taille individuelle est de 1kB.

```
flask simu data -s 1kb -q 2000
```

Code B.12 – Benchmark permettant de lancer 2000 itérations basées sur un échantillon de données de 1kb en mode CLI

B.3 Graphiques complémentaires

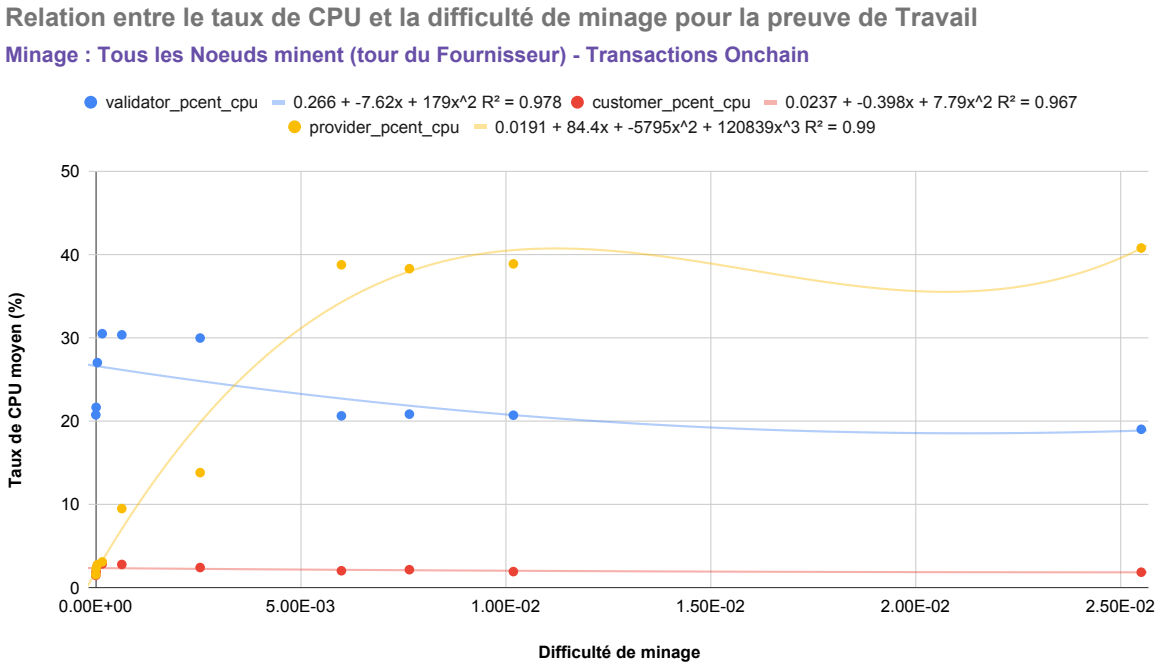


Figure B.2 – Relation entre le taux de CPU et la difficulté de minage pour la preuve de travail (tous les noeuds minent, tour du fournisseur)

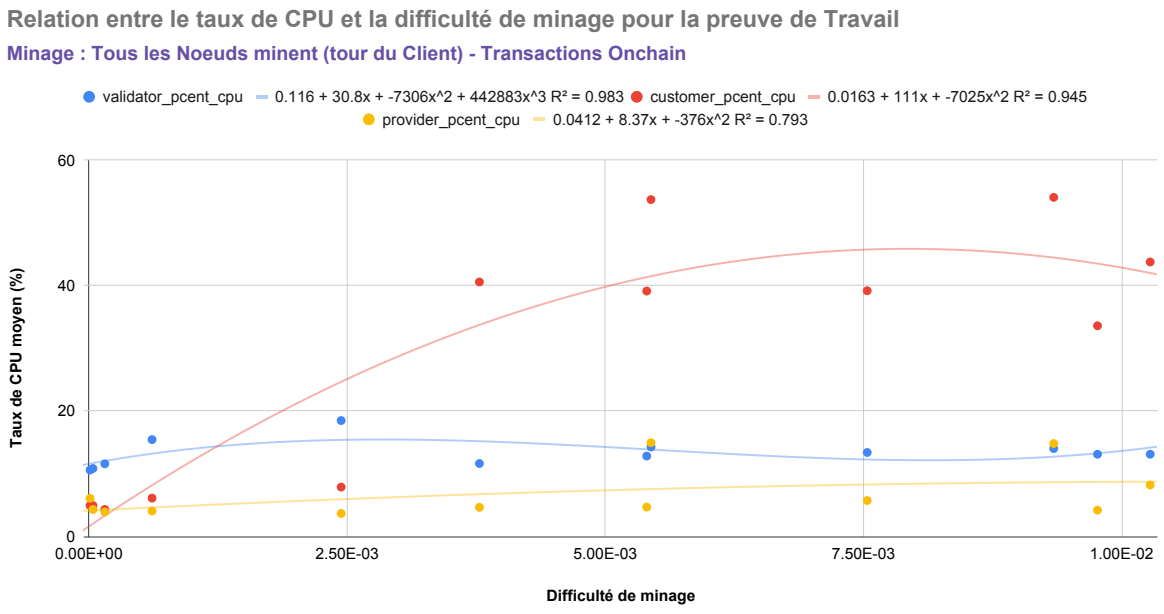


Figure B.3 – Relation entre le taux de CPU et la difficulté de minage pour la preuve de travail (tous les noeuds minent, tour du client)

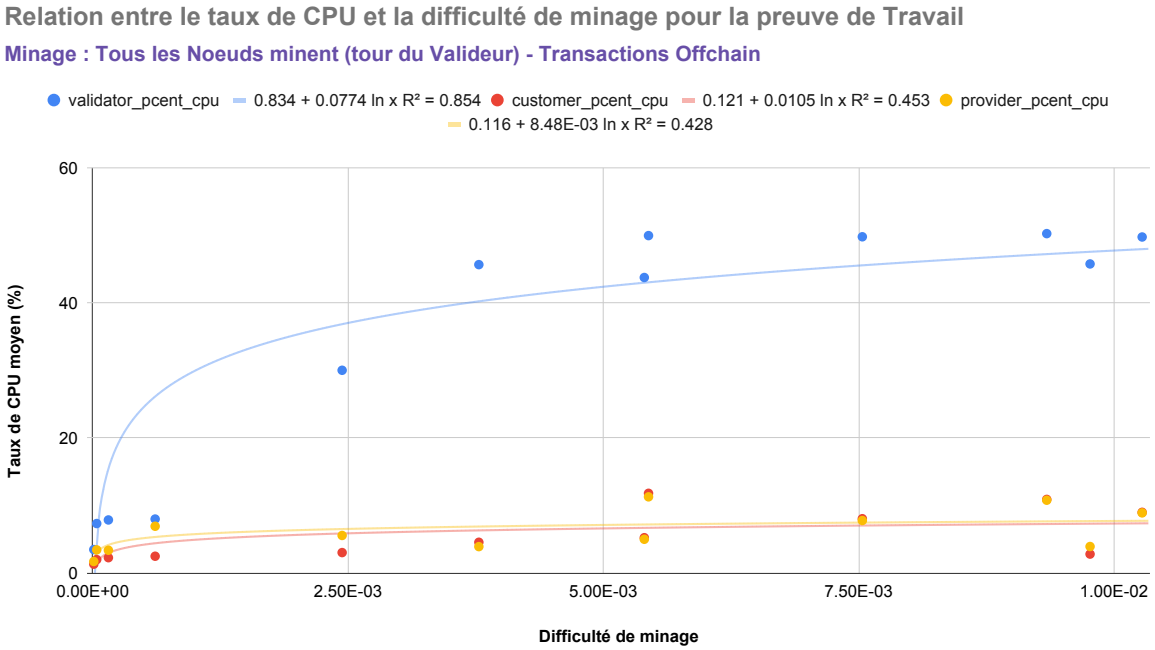


Figure B.4 – Relation entre le taux de CPU et la difficulté de minage pour la preuve de travail en mode offchain (tous les noeuds minent, tour du valideur)

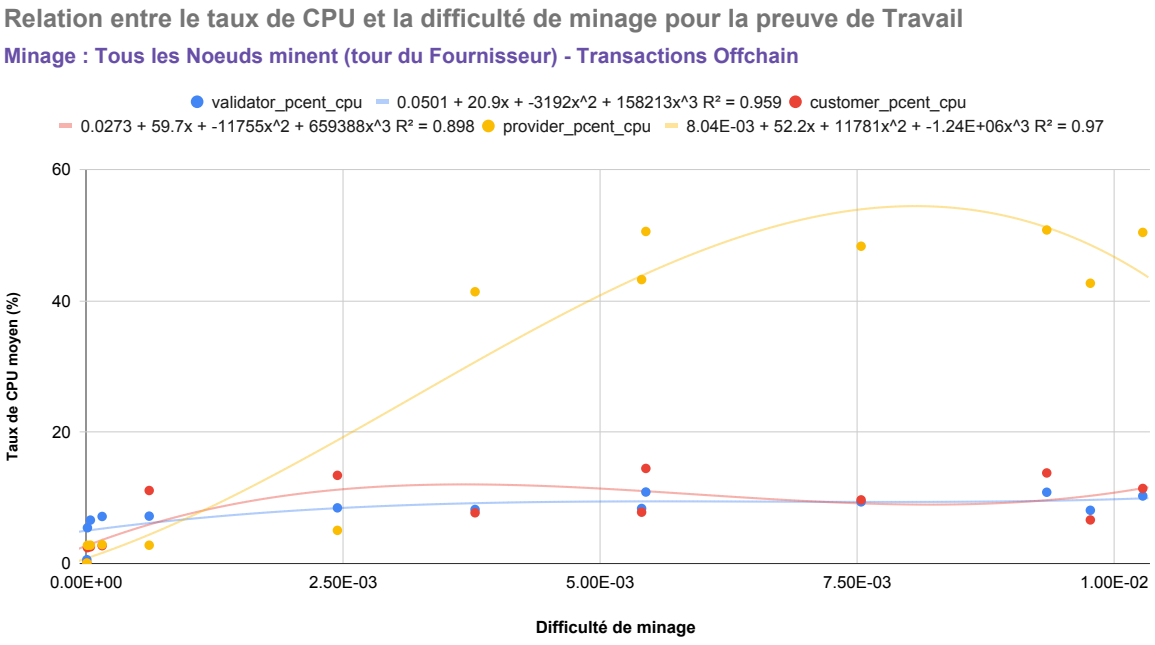


Figure B.5 – Relation entre le taux de CPU et la difficulté de minage pour la preuve de travail en mode offchain (tous les noeuds minent, tour du fournisseur)

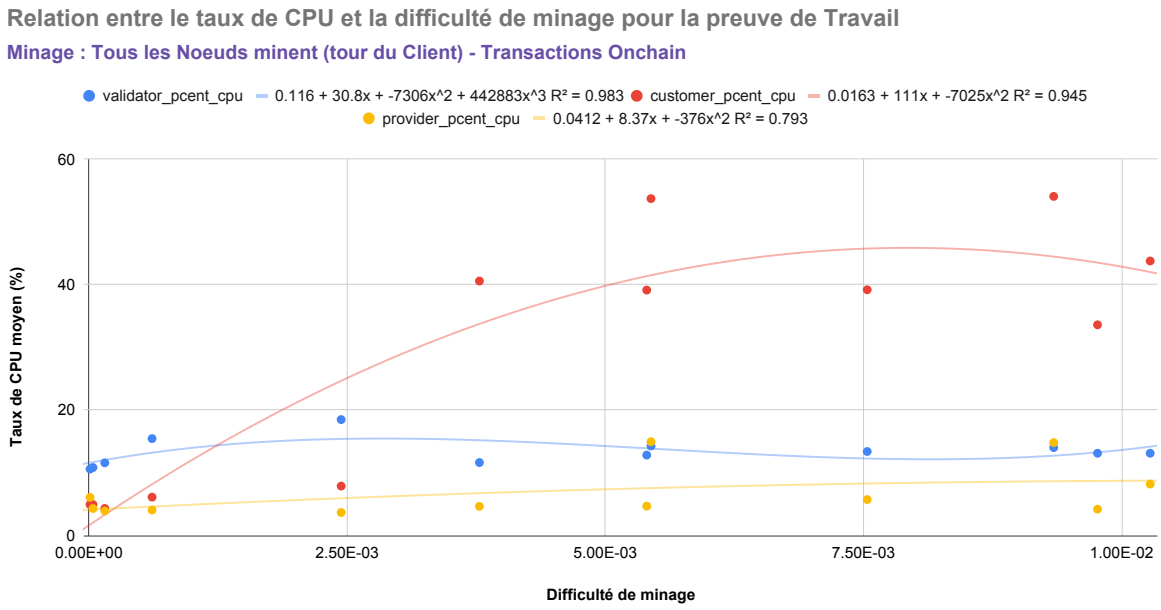


Figure B.6 – Relation entre le taux de CPU et la difficulté de minage pour la preuve de travail en mode offchain (tous les noeuds minent, tour du client)

C

Présentation de ARTIS*, librairie DEVS C++ 11

Le simulateur pour usine 4.0 présenté dans le Chapitre 5 a été développé en se basant sur un framework nommé ARTIS* [130] développé en C++ 11 et dont le rôle est de permettre la multimodélisation et la simulation de systèmes complexes dynamiques.

Il résout le problème de fiabilité en utilisant des développements de la théorie de la modélisation et de la simulation proposés par B. Zeigler avec le formalisme Discrete Event System Specification (DEVS) et ces extensions. Tout y est défini sous forme de classes à base de templates à savoir la dynamique de modèles atomiques, le graphe de connexions, les paramètres, les observations des modèles, la sérialisation des modèles. Cet annexe permettra d'aborder plus en détail certains aspects de son fonctionnement technique.

C.1 Modèle atomique

Un modèle atomique peut s'apparenter d'un point de vue mathématique comme une structure composée de variables et de fonctions.

Definition C.1.1 *Représentation mathématique d'un modèle atomique*

$$DEVS = \langle X, Y, S, S_0, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

X : L'ensemble des ports d'entrée et des valeurs attachées
- *Y : L'ensemble des ports de sortie*
- *S : L'ensemble des états du système*
- *δext : Fonction de transition externe qui représente les évolutions reçues par les ports d'entrée*
- *δint : Fonction de transition interne qui représente les évolution autonomes*
- *δconf : Fonction de conflit qui représente les réponses du système si des évènements externes et internes surviennent en même temps*
- *ta : Temps pendant lequel le modèle reste dans l'état S*
- *λ représente les influences externes*

Dans Artis, un modèle atomique est représenté comme une classe héritant de **artis : :pdevs : :Dynamics**. Cette classe est un template et prend 3 paramètres :

- La notion de temps utilisée
- La classe elle-même
- Le type des paramètres admis par le modèle atomique

Le constructeur permettra d'initialiser le modèle atomique en terme de paramètres. Tout modèle, qu'il soit atomique ou couplé est nommé.

C.1.1 Constructeur et paramètres

Un modèle atomique possède des paramètres qui sont des structures de données. Ces structures sont définies en dehors du code des modèles. Les paramètres sont accessibles dans le constructeur du modèle grâce au contexte.

```
1  struct ExampleParameters {
2    double x;
3    double y;
4  };
5
6  class Example : public artis::pdevs::Dynamics<artis::common::DoubleTime,
7                                                 Example, ExampleParameters> {
8  public:
9      Example(const std::string &name,
10           const artis::pdevs::Context<artis::common::DoubleTime, Machine,
11                 ExampleParameters> &context)
12       : artis::pdevs::Dynamics<artis::common::DoubleTime, Machine,
13             ExampleParameters>(name, context) {
14         // ...
15     }
16 }
```

Code C.1 – Déclaration de paramètres pour un modèle atomique

C.1.2 Caractéristiques et méthodes à implémenter

start

$$DEVS = \langle X, Y, S, \mathbf{S_0}, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Elle est appelée au démarrage de la simulation une fois tous les modèles construits. La durée de l'état initial est défini par la fonction *ta*.

```
1  virtual void artis :: pdevs :: Dynamics :: start(
2  const typename Time :: type& time );
```

Code C.2 – Méthode start du modèle atomique dans Artis*

δint

$$DEVS = \langle X, Y, S, S_0, \mathbf{\delta_{int}}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Elle est appelée à l'échéance d'un *ta* et après λ

```
1  virtual void artis :: pdevs :: Dynamics :: dint(
2  const typename Time :: type& time );
```

Code C.3 – Méthode δint du modèle atomique dans Artis*

δext

$$DEVS = \langle X, Y, S, S_0, \delta_{int}, \mathbf{\delta_{ext}}, \delta_{conf}, \lambda, ta \rangle$$

Elle est appelée lorsque le modèle reçoit au moins un évènement externe. Le modèle peut être modifié mais elle ne déclenche pas l'appel de la fonction λ

```
virtual void dext(const typename Time :: type& t,
const typename Time :: type& e,
const common ::Bag <Time >& bag );
```

Code C.4 – Méthode δext du modèle atomique dans Artis*

δconf

$$DEVS = \langle X, Y, S, S_0, \delta_{int}, \delta_{ext}, \boldsymbol{\delta_{conf}}, \lambda, ta \rangle$$

Elle est appelée lorsque le modèle reçoit au moins un évènement externe. Le modèle peut être modifié mais elle ne déclenche pas l'appel de la fonction λ

```
virtual void dconf(const typename Time :: type& t,
const typename Time :: type& e,
const common ::Bag <Time >& bag );
```

Code C.5 – Méthode δconf du modèle atomique dans Artis*

ta

$$DEVS = \langle X, Y, S, S_0, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, \boldsymbol{ta} \rangle$$

Appelée après toute les fonctions de transition δint, δext ou δconf. Elle retourne la durée pendant laquelle le modèle reste dans l'état fourni par l'appel à δint, δext ou δconf. Par défaut, elle retourne une durée infinie. Il s'agit d'une fonction *const* donc elle ne modifie pas l'état.

```
virtual typename Time :: type ta(
const typename Time :: type& time) const;
```

Code C.6 – Méthode ta du modèle atomique dans Artis*

lambda λ

$$DEVS = \langle X, Y, S, S_0, \delta_{int}, \delta_{ext}, \delta_{conf}, \boldsymbol{\lambda}, ta \rangle$$

Elle est appelée après l'échéance d'un *ta* et avant un δint. Les évènements externes générés sont retournés sous la forme d'un objet *bag*. Il s'agit également d'une fonction *const* donc elle ne modifie pas l'état du modèle.

```
virtual common ::Bag <Time > lambda(
const typename Time :: type& time) const;
```

Code C.7 – Méthode λ du modèle atomique dans Artis*

C.1.3 Déclaration d'attributs

$$DEVS = \langle X, Y, \boldsymbol{S}, S_0, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Les attributs ne peuvent être modifiées que dans les fonctions *start*, *dint* et *dext*. Tous les types d'attributs sont possibles.

```
// ...
private:
    double _x;
    double _y;
    State _state;
```

Code C.8 – Exemple de déclaration d'attributs dans un modèle atomique

C.1.4 Définition des ports d'entrée et de sortie

$$DEVS = \langle \boldsymbol{X}, \boldsymbol{Y}, S, S_0, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

X et Y représentent respectivement les ports d'entrée et de sortie du modèle atomique. Le nommage des ports est réalisé par des énumérations

```
// ...
public:
  struct inputs {
    enum values { IN_1, IN_2 };
  };

  struct outputs {
    enum values { OUT_1};
  };
```

Code C.9 – Nommage des ports dans un modèle atomique

La déclaration des ports d'entrée et de sortie se fait dans le constructeur.

```
// ...dans le constructeur
input_port({inputs::IN_P, "in_1"});
output_port({outputs::OUT_P, "out_1"});
```

Code C.10 – Déclaration des ports dans un modèle atomique

C.1.5 Définition des observables

Un modèle peut être observé à tout moment et doit être capable de calculer certaines variables d'observation. Ces variables d'observation sont définies par le modèle.

```
public:
    struct vars {
        enum values {
            NAME,
            STATE
        };
    };
```

Code C.11 – Nommage d'un observable dans un modèle atomique

```
// ...dans le constructeur
observable({vars::NAME, "name"});
observable({vars::STATE, "state"});
```

Code C.12 – Déclaration d'un observable dans un modèle atomique

Tout modèle atomique implémente une fonction **observe** permettant de faire le lien entre le nom d'un observable et la valeur observée (Code C.13)

```
artis::common::event::Value
Example::observe(const artis::factory::Time &,
                 unsigned int index) const {
    switch(index) {
        case vars::NAME:
            return _name;
        case vars::STATE:
            return _state;
    }
    return artis::common::event::Value;
}
```

Code C.13 – Déclaration d'un observable dans un modèle atomique

C.2 Modèle couplé

Un modèle couplé est un modèle composé de sous-modèles qui peuvent être atomiques ou couplés également. Il possède également des ports d'entrée et de sortie. On le représente comme une classe héritant de **artis : :pdevs : :GraphManager**.

C.2.1 Constructeur et paramètres

Tout comme le modèle atomique, les paramètres du modèle couplé sont définies dans des structures en dehors du modèle. La différence se situe dans le fait qu'il est nécessaire de définir une structure supplémentaire encapsulant les structures des paramètres des sous-modèles (Code C.14). On retrouve ici un modèle couplé ***LineGraphManager*** représentant une ligne de production ainsi qu'une structure *GraphManagerParameters* représentant les paramètres du modèle couplé.

La ligne de production possède deux sous-modèles à savoir l'approvisionneur et le routeur ce qui nécessite la structure *SubModelsParameters* contenant les paramètres des sous-modèles *DispatcherParameters* et *RouterParameters*.

```cpp
struct SubModelsParameters {
    artis::factory::DispatcherParameters dispatcher_parameters;
    artis::factory::MachineRouterParameters router_parameters;
}

struct GraphManagerParameters {
    std::size_t nb_machines;
}

LineGraphManager(
      artis::common::Coordinator<artis::common::DoubleTime> *coordinator,
      const SubModelsParameters &parameters,
      const GraphManagerParameters &graph_parameters)
      : artis::pdevs::GraphManager<artis::common::DoubleTime,
                                   LineSubModelsParameters,
                                   LineGraphManagerParameters>(
            coordinator, parameters, graph_parameters) {
       // ...
}
```

Code C.14 – Exemple de constructeur et de paramètres d'un modèle couplé

C.2.2 Définition des sous-modèles et des connexions

Les sous-modèles sont représentés par des énumérations qui permettent de les identifier (Code C.15).

La première étape consiste donc à déclarer le sous-modèle dans l'énumération.

```cpp
public:
  enum sub_models {
    DISPATCHER,
    ROUTER,
    MACHINE
  };
```

Code C.15 – Nommage des sous-modèles

Il faut ensuite ajouter le sous-modèle au modèle couplé via la méthode *add_child* dans le constructeur du modèle couplé (Code C.16).

```cpp
this->add_child(DISPATCHER, &_dispatcher);
this->add_child(ROUTER, &_router);
this->add_child(MACHINE, &_machine);
```

Code C.16 – Ajout d'un sous-modèle à un modèle couplé (coordinateur)

La dernière étape dans la définition du sous-modèle consiste à faire la liaison avec le modèle couplé et les autres sous-modèles en définissant les connexions entre les ports d'entrées et de sortie.

Dans le Code C.17, on prend l'exemple de la ligne de production (modèle couplé) possédant deux sous-modèles (routeur et machine). L'entrée de la ligne est connectée au routeur, la sortie du routeur est connectée à l'entrée de la machine et enfin la sortie de la machine est connectée à la sortie de la ligne.

```
coordinator -> input_port ({IN , "in"});
coordinator -> output_port ({OUT , "out"});
in({ coordinator , IN}) >> in ({&_router, ROUTER::IN });
out ({&_router, ROUTER::OUT }) >> in ({&_machine, MACHINE::IN });
out ({&_machine, MACHINE:: OUT }) >> out ({ coordinator , OUT });
```

Code C.17 – Déclaration des connexions entre le modèle couplé et les sous-modèles

C.2.3 Définition des vues

Les vues sont définies pour observer le modèle global et sont connectées à des sorties (fichiers .csv dans le cas d'ARTIS*).

Techniquement, une vue est une classe héritant de **artis : :common : :observer : :View** et ayant pour attributs un ensemble de chemins vers les modèles atomiques. Un chemin est composé de la liste des identifiants des modèles couplés, du modèle atomique et de la variable d'observation.

Le Code C.18 présente un exemple de vue pour une machine où le constructeur prend en paramètre l'identifiant de la ligne et de la machine. Les chemins vers les modèles atomiques sont définis via la méthode *selector*.

Pour accéder à l'observable *NAME*, il faut passer par le modèle couplé *LINE*, le modèle atomique *MACHINE* et enfin *NAME* qui est une variable d'observation de *MACHINE*.

```
class MachineView : public artis::common::observer::View<
        artis::common::DoubleTime> {
public:
  MachineView(int line_id, int machine_id) {
    selector("machine:name", {
        artis::factory::FactoryGraphManager::LINE + line_id,
        artis::factory::LineGraphManager::MACHINE + machine_id,
        artis::factory::Machine::vars::NAME});
    selector("machine:state", {
        artis::factory::FactoryGraphManager::LINE + line_id,
        artis::factory::LineGraphManager::MACHINE + machine_id,
        artis::factory::Machine::vars::STATE});
    }
```

Code C.18 – Déclaration d'une vue dans ARTIS*

D

Documentation Technique Simulateur Usine 4.0

Les résultats présentés dans le Chapitre 5 notamment les scénarios sont basés sur le simulateur d'usine 4.0 dont le fonctionnement et certaines caractéristiques techniques seront décrites tout au long de cette annexe.

D.1 Architecture

L'architecture logicielle du simulateur décrite dans la Figure D.1 est constituée des éléments suivants :

- Un générateur de scénario développé en C++ 11 utilisant un fichier de configuration JSON
- Le scénario sous forme de fichier JSON contenant les paramètres du modèle de simulation
- Le simulateur développé en C++ 11 composé du modèle de l'usine

Les résultats de la simulation sont quant à eux exportés sous forme de fichiers .csv

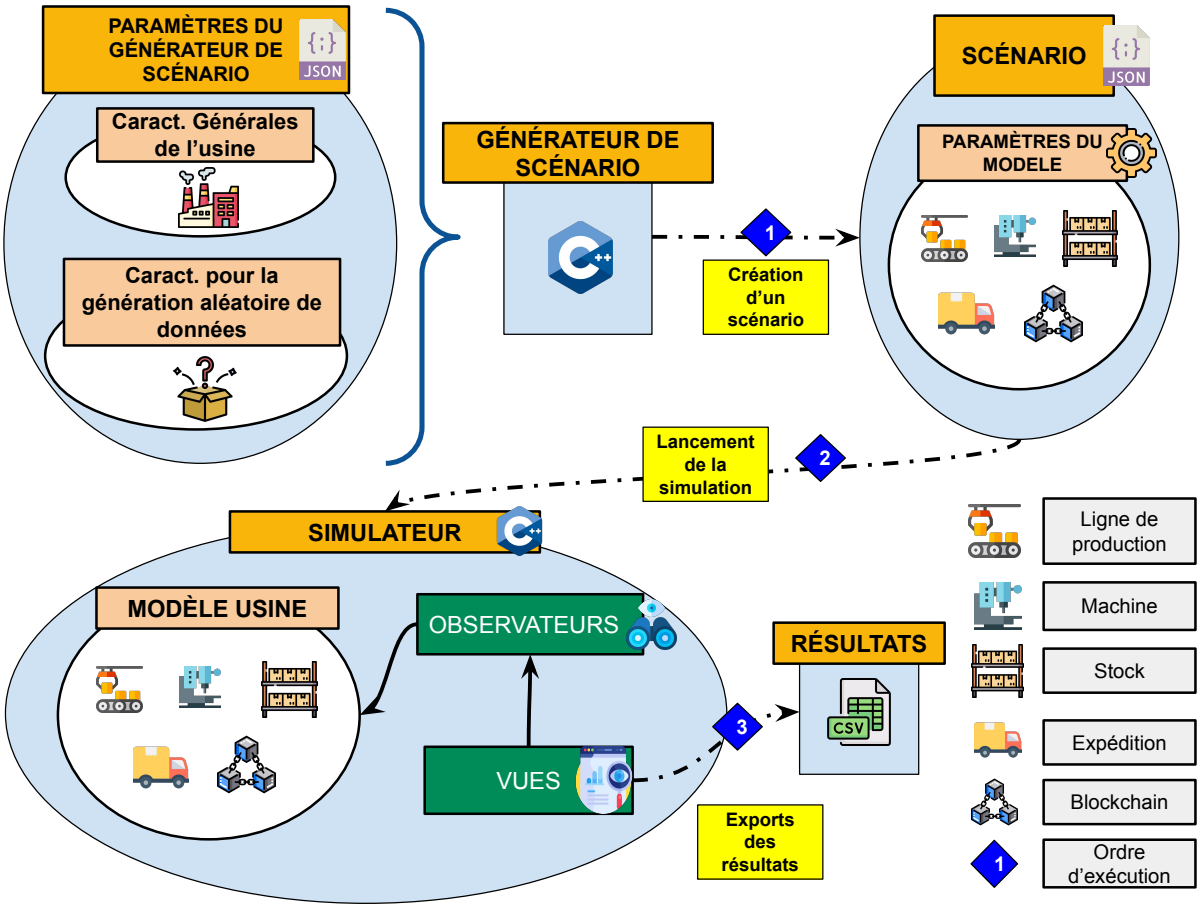


Figure D.1 – Architecture logicielle du simulateur pour l'usine 4.0

D.2 Générateur de scénario

La structure du scénario peut s'avérer complexe compte tenu du nombre important de paramètres nécessaires à la modélisation des différents éléments constituant l'usine. Dans ces conditions, il a paru judicieux de mettre en place un générateur de scénario permettant de créer la configuration complète d'une usine en fonction de diverses caractéristiques d'ordre général. Ce générateur est paramétrable via un fichier de configuration en *JSON*.

D.2.1 Génération de valeurs pseudo-aléatoires

Afin de pouvoir générer des scénarios variés, nous allons recourir à la génération de nombres via des générateurs pseudo aléatoires. Un générateur pseudo-aléatoire est un algorithme déterministe, donc une suite finie et non ambiguë d'opérations ; à partir d'un nombre de départ donné, il générera toujours la même suite.

Graine aléatoire

La génération de scénario se doit d'être déterministe ; pour la même configuration, le générateur doit toujours produire le même scénario. La génération de nombres pseudo-aléatoire est basée sur un nombre de départ appelé **graine aléatoire** permettant de garantir que la suite générée sera toujours la même. C'est la raison pour laquelle la configuration du générateur de scénario possède un paramètre *random_seed* permettant de fixer la valeur de la graine aléatoire et donc d'assurer le côté déterministe.

```
{
    "random_seed": 32456941
}
```

Code D.1 – Configuration d'une graine aléatoire dans le générateur de scénario

En C++, on utilise le générateur pseudo-aléatoire **Mersenne Twister**, plus précisément sa variante *MT19937* dont la particuliarité est d'offrir une période beaucoup plus longue au niveau de la suite de nombres pouvant être générée.

```
uint _random_seed;
std::mt19937 _rng;

_rng = std::mt19937(_random_seed);
```

Code D.2 – Utilisation de la graine aléatoire dans le générateur de scénario en C++ 11

Lois de probabilité

Ces générations de nombres pseudo-aléatoires sont basées sur des lois de probabilité aussi appelées distributions. Dans notre cas, on utilise principalement deux types de loi qui permettent de produire des valeurs variées afin de se rapprocher d'un contexte réel :

- **Loi de Bernoulli** qui permet de produire des valeurs booléennes aléatoires selon une probabilité définie
- **Loi de génération uniforme** d'entiers ou de réels qui permet de produire des valeurs aléatoires distribués uniformément sur un intervalle donné. Le caractère uniforme signifie que la moyenne des valeurs générées tend vers la moyenne de l'intervalle.

```
std::bernoulli_distribution _distrib_data_type;
std::uniform_int_distribution<> _distrib_raw_data_size;
std::uniform_real_distribution<double> _distrib_file_diff_rate;
```

Code D.3 – Exemple de lois de distribution pseudo-aléatoires en C++

D.2.2 Structure de l'usine

Tableau D.1 – Description des paramètres liés à l'usine dans le générateur de scénario du simulateur Usine 4.0

Nom	*Description*	*Exemple*
nb_process_lines	Nombre de lignes de production de type *PROCESS* (Usinage)	40
nb_assembly_lines	Nombre de lignes de production de type *ASSEMBLY* (Assemblage)	20
nb_testing_lines	Nombre de lignes de production de type *TESTING* (Essai)	20
rng_nb_machines_process_lines	Bornes min et max pour la génération aléatoire du nombre de machines sur une ligne de type *PROCESS*	[1,5]
rng_nb_machines_assembly_lines	Bornes min et max pour la génération aléatoire du nombre de machines sur une ligne de type *ASSEMBLY*	[1,5]
rng_nb_machines_testing_lines	Bornes min et max pour la génération aléatoire du nombre de machines sur une ligne de type *TESTING*	[1,5]
nb_product_refs	Nombre de produits différents fabriqués par l'usine (influencera le nombre de gamme de fabrication à générer)	70
nb_stock_components	Nombre de références différentes de composants existentes dans le stock	100
total_day_production	Nombre de produits moyens fabriqués en une journée par l'usine	30000

D.2.3 Caractéristiques des données de traçabilité

Dans le générateur de scénario, les paramètres liés aux données de traçabilité servent à contrôler le fonctionnement des lois de génération aléatoire utilisées. (Tableau D.2).

Tableau D.2 – Description des paramètres liés aux données de traçabilité dans le générateur de scénario du simulateur Usine 4.0

Nom	*Description*	*Exemple*
distrib_data_type	Valeur entre 0 et 1 représentant la répartition des types de données (brutes ou fichiers). Par exemple, une valeur de 0.6 implique qu'environ 60% des données seront des données brutes	0.6
distrib_data_confidentiality	Valeur entre 0 et 1 représentant la répartition des données confidentielles et non-confidentielles. Par exemple, une valeur de 0.4 implique qu'environ 40% des données seront confidentielles	0.4
rng_nb_tracea_data	Bornes min et max du nombre de données de traçabilité générés par un modèle atomique (machine ou autres)	[2, 10]
rng_raw_data_size	Bornes min et max de la taille des données brutes (en octets) générées aléatoirement	[128, 1024]
rng_file_data_size	Bornes min et max de la taille des fichiers (en octets) générés aléatoirement	[10240, 102400]
rng_file_diff_rate	Bornes min et max du taux de différence généré aléatoirement pour les fichiers de traçabilité	[0.1, 0.3]

D.2.4 Quantité de noeuds de la blockchain

Afin de pouvoir faire évoluer facilement l'architecture de la blockchain, il est également possible de définir la quantité de noeuds par rôle.

Le Code D.4 montre la configuration de la quantité de noeuds de la blockchain dans le générateur de scénario. Il s'agit d'une liste de tuples *Rôle-Quantité* où le rôle peut être *Valideur*, *Client* ou *Fournisseur*.

```
{
    "blockchain_nodes_quantities": [
        ["VALIDATOR", 1],
        ["CUSTOMER", 1],
        ["PROVIDER", 1]
  ]
}
```

Code D.4 – Définition de la quantité de noeuds simulés dans la blockchain

D.3 Scénario

Le scénario constitue la pierre angulaire du simulateur dans le sens où il est la transcription en JSON de l'ensemble des paramètres du modèle de simulation.

D.3.1 Paramètres de contexte de la simulation

Les paramètres de contexte correspondent aux valeurs concernant le temps de simulation à savoir le temps de début, de fin et le type de pas d'observation utilisé (Tableau D.3). Par défaut, la simulation s'effectue sur un pas d'observation évènementiel c'est à dire que tout changement d'état dans un modèle induit une mesure des valeurs des observables ce qui permet d'avoir un aperçu précis sur l'évolution du modèle. Si la situation ne s'y prête pas, par exemple dans un cas où l'on souhaite uniquement connaître l'état du modèle à la fin de la simulation, alors il est possible d'utiliser un pas d'observation temporel où le modèle sera observé toutes les X secondes.

Tableau D.3 – Description des paramètres de contexte de la simulation

Nom	*Description*	*Exemple*
simu_start	Temps de début de la simulation en seconde	0
simu_end	Temps de fin de la simulation en seconde	86400 (1 Jour)
timed_observer_step	Pas d'observation de la simulation. Par défaut, l'observation se fait au niveau des évènements (valeur 0) mais on peut également définir un pas temporel (observation toutes les X secondes)	$X \geq 0$

D.3.2 Paramètres du générateur d'O.F (Ordre de Fabrication)

Le générateur d'ordre de fabrication prend en paramètre deux éléments à savoir les gammes de fabrication des produits ainsi que la liste des commandes envoyées à l'usine. Ces deux éléments forment ensemble ce qu'on appelle l'ordre de fabrication qui précisent la quantité à produire, la date prévue.

Gamme de fabrication Une gamme de fabrication est le mode opératoire décrivant les étapes nécessaires à la fabrication d'un produit. Elle est liée à la nomenclature qui est la liste et la quantité des composants à mettre en oeuvre aux différentes étapes de la gamme.

Le Code D.5 présente un extrait de gamme de fabrication telle qu'elle est représentée dans le scénario en JSON soumis simulateur. Ici, une gamme est représenté par une liste de destinations par lesquelles le produit doit passer afin d'être fabriqué. La destination est constitué d'une ligne de production *_line* ainsi que la liste des machines *_machines* de cette ligne.

```
{
    "manu_programs": {
        "PRD0": [
          {
            "_line": { "_line_id": 2 },
            "_machines": [
              { "_machine_id": 1 }, { "_machine_id": 2 },
              { "_machine_id": 3 }
            ]
          },
          {
            "_line": { "_line_id": 8 },
            "_machines": [
              { "_machine_id": 1 }, { "_machine_id": 2 }
            ]
          },
          {
            "_line": { "_line_id": 12 },
            "_machines": [
              { "_machine_id": 1 }, { "_machine_id": 2 },
              { "_machine_id": 3 }, { "_machine_id": 4 }
            ]
          }
        ]
    }
}
```

Code D.5 – Expression de la gamme de fabrication d'un produit dans le scénario en JSON

D.3.3 Paramètres des machines

Les machines sont associées à une ligne et sont représentées par plusieurs paramètres décrits dans le Tableau D.4.

Tableau D.4 – Description des paramètres d'une machine dans le simulateur Usine 4.0

Nom	*Description*	*Exemple*
machine_id	Identifiant de la machine	1
machine_name	Nom de la machine	L1M1
machine_type	Type de la machine ayant pour valeurs possibles : — PROCESS (Traitement) — ASSEMBLY (Assemblage) — TESTING (Essai)	PROCESS
machining_time	Temps de cycle de la machine en secondes	3.6
machine_load_time	Temps de chargement d'une pièce en secondes	1.2
machine_unload_time	Temps de déchargement d'une pièce en secondes	1.1
required_compo	Référence du/des composants dont la machine a besoin pour accomplir ce qu'elle doit faire. Ce champ ne concerne que les machines de type *Assemblage* en accord avec la vision traçabilité centrée produit	C00121
product_tracea_data	Liste des données de traçabilité que la machine génère à chaque fois qu'elle traite une pièce / un produit	

Parmis ces paramètres, on constate que trois d'entre eux sont liés au temps à savoir le temps de cycle *machining_time*, le temps de chargement *machine_load_time* afin de se calibrer avant de commencer à traiter la

pièce et le temps de déchargement *machine_unload_time* qui permet à la machine d'évacuer la pièce afin de pouvoir en traiter une nouvelle.

$$T_{tot} = T_{cyc} + T_{charg} + T_{decharg}$$

— Ttot : Temps total nécessaire au traitement de la pièce par la machine
— Tcyc : Temps de cycle de la machine
— Tcharg : Temps de chargement de la pièce
— Tdecharg : Temps de déchargement de la pièce

D.3.4 Paramètres de la blockchain

Les paramètres de la blockchain sont divisées en deux catégories : les paramètres généraux appliquées à l'ensemble de la chaîne (Tableau D.5) ainsi que les paramètres spécifiques à chaque noeud (Tableau D.6).

Paramètres généraux

Tableau D.5 – Description des paramètres généraux de la blockchain dans le simulateur Usine 4.0

Nom	*Description*	*Exemple*
block_overhead_size	Surplus d'espace consommé en octets par chaque nouveau bloc indépendamment de la quantité de données contenue dans le bloc	2048
data_hash_size	Taille des hash en octets ajoutés aux donnée de traçabilité dans les transactions	32
data_subscribing	Liste représentant les noeuds abonnés aux données	
encryption_key_size	Taille de la clé de cryptage ajoutée aux données confidentielles et insérée dans la blockchain	128
file_policy_storage	Politique de stockage pour les fichiers	*ONCHAIN*, *OFFCHAIN* ou *EXTERNAL*
max_block_size	Taille maximale d'un bloc en octets	8388608 (8MB)
mining_algorithm	Algorithme de minage utilisé	PROOF_OF_WORK | NON_POW (RoundRobin)
raw_policy_storage	Politique de stockage pour les données brutes	*ONCHAIN*, *OFFCHAIN* ou *EXTERNAL*
rng_mining_difficulty		
signature_size	Taille des signatures ajoutées aux données et insérées dans la blockchain	65
target_adjust_freq	Fréquence de mise à jour en secondes de la difficulté de minage (uniquement pour la preuve de travail)	60
target_block_time	Fréquence de minage des blocs en secondes	30
tx_data_capacity		
tx_overhead_size	Surplus d'espace consommé à chaque nouvelle transaction peu importe son contenu	200

Parmi les paramètres généraux, on trouve plusieurs notions permettant de modifier la politique de stockage tels que *file_policy_storage* et *raw_policy_storage* qui permettent de modifier le mode de stockage des fichiers ou des données brutes.

Il est également possible de modifier l'algorithme de minage utilisé via le paramètre *mining_algorithm*

Tableau D.6 – Description des paramètres d'un noeud blockchain dans le simulateur Usine 4.0

Nom	*Description*	*Exemple*
name	Identifiant permettant de différencier les noeuds	v1
can_mine	Booléen indiquant si le noeud a le droit ou non de miner	True | False
cpu_tdp	Taux de dissipation thermique du processeur exprimé en *Watt*	95
gpu_tdp	Taux de dissipation thermique du processeur graphique exprimé en *Watt*	0 (dans le cas présent)
memory_consumption	Consommation électrique en Watt associé à la mémoire	0 (dans le cas présent)
nb_cpu	Nombre de CPU affecté au noeud blockchain	2.0
nb_gpu	Nombre de GPU affecté au noeud blockchain	0
role	Rôle affecté au noeud	— VALIDATOR — CUSTOMER — PROVIDER

Paramètres des noeuds

Les paramètres des noeuds sont essentiellement dédiés à la gestion de leur comportement vis à vis du minage avec par exemple *can_mine* permettant d'autoriser ou non un noeud à miner.

Toujours en lien avec le minage, les autres paramètres servent quant à eux à évaluer la consommation d'énergie tels que *cpu_tdp* ou *nb_cpu*.

D.3.5 Paramètres des données de traçabilité

Détermination du taux de différence entre deux fichiers

```
double compareFiles(ifstream &in1, ifstream &in2, const size_t
        chunk_size) {
  //...
  while (remaining) {
    char *buffer1 = new char[chunk_size];
    char *buffer2 = new char[chunk_size];
    size_t size = std::min(chunk_size, remaining);

    in1.read(buffer1, size);
    in2.read(buffer2, size);

    if (0 != memcmp(buffer1, buffer2, size)) {
      nequal_chunks++;
    } else {
      equal_chunks++;
    }
    remaining -= size;
  }
  return ((double)nequal_chunks / (equal_chunks + nequal_chunks));
}
```

Code D.6 – Fonction permettant de déterminer le taux de différence entre deux fichiers en C++

D.3.6 Paramétrage des vues

Au niveau du générateur de scénario, il est possible de paramétrer les vues qui seront utilisées dans le cadre de la simulation (Code D.7). On retrouve quatre paramètres qui sont : l'identifiant de la vue utilisée dans le générateur de vues, le nom du fichier .csv généré, l'état de la vue (active ou non) et une liste des paramètres propres à cette vue

Code D.7 – Exemple de paramétrage des vues au niveau du générateur de scénario

```
{
    "view_params": [
    ["BLOCKCHAIN_GENERAL_VIEW", "Blockchain_Global", false, []],
    ["BLOCKCHAIN_NODE_STORAGE_VIEW", "Blockchain_Node_Storage", false, []],
    ["SHIPPING_GENERAL_VIEW", "Shipping", false, []]
    ]
}
```

Ces paramètres sont ensuite utilisées pour créer les vues dans la simulation grâce ç un générateur de vues donc la fonction principale *attach_views* permet de lier les vues au simulateur (Code D.8).

Code D.8 – Liaison entre les paramètres des vues et le simulateur

```
void attach_views() {
    for (auto view_param : _scenario->view_params) {
      ViewType view_type = std::get<0>(view_param);
      std::string view_name = std::get<1>(view_param);
      bool is_enabled = std::get<2>(view_param);
      std::vector<std::string> view_args = std::get<3>(view_param);
      if (is_enabled) {
        // ...
        switch (view_type) {
        case POG_GENERAL_VIEW:
          _rc->attachView(view_name, new POGView());
          break;
        case BLOCKCHAIN_GLOBAL_VIEW:
          _rc->attachView(view_name, new BlockchainGlobalView());
          break;
        case BLOCKCHAIN_NODE_STORAGE_VIEW:
          _rc->attachView(view_name, new BlockchainNodeStorageView());
          break;
          // ...
        }
      }
    }
}
```

D.3.7 Description des vues supportées

L'ensemble des vues gérées par le simulateur sont décrites dans le Tableau D.7.

Tableau D.7 – Liste des vues supportées dans le simulateur

Nom	*Observables*	*Description*
POG_GENERAL_VIEW	current_id, po_ref	Vue standard du générateur d'ordre de production (POG)
GLOBAL_STOCK_GENERAL_VIEW	state, po_id, compo_line, compo_machine, compo_ref	Vue standard du Stock
LINE_ROUTER_GENERAL_VIEW	state, po_id, destination, po_ope_index, po_counter	Vue standard du routeur de ligne
SHIPPING_GENERAL_VIEW	po_id, po_counter	Vue standard du modèle d'Exépdition (Shipping)
MACHINE_ROUTER_GENERAL_VIEW	state, po_id, po_ope_index, destination, po_counter	Vue standard du routeur machine
DISPATCHER_GENERAL_VIEW	state, po_id, compo_line, compo_machine, compo_ref	Vue standard de l'approvisionneur
MACHINE_GENERAL_VIEW	name, state, po_id, po_counter	Vue standard du modèle Machine
MACHINE_UPSTOCK_GENERAL_VIEW	state, po_id, is_machine_ready, stock_size	Vue standard du modèle MachineUpStock (stock amont machine)
MACHINE_DOWNSTOCK_GENERAL_VIEW	state, po_id, stock_size	Vue standard du modèle MachineDownStock (stock aval machine)
PRODUCTION_TRACEA_VIEW	lineX_po_counter, lineX_tracea_size	Vue des statistiques en terme de traçabilité au niveau des lignes
BLOCKCHAIN_GLOBAL_VIEW	state, length, total_tx, next_mining_time, total_raw_size, total_file_size, total_nochain_size, total_onchain_size, total_offchain_size, total_external_size, total_tx_overhead_ total_block_overhead, data_hash_size, encryption_key_size, signature_size, blockchain_size	Vue globale sur l'état du modèle Blockchain
BLOCKCHAIN_NODE_STORAGE_VIEW	nochain_size, validator_size, customer_sizee, provider_size, watcher_size	Vue des statistiques de stockage au niveau des noeuds de la blockchain
BLOCKCHAIN_PO_STAT_VIEW	po_id, onchain_raw_confid, onchain_raw_nonconfid, onchain_file_confid, onchain_file_nonconfid (ident. pour offchain)	Vue des statistiques de stockage en terme de traçabilité au niveau de chaque produit fabriqué
BLOCKCHAIN_TX_DETAIL_VIEW	po_id, sender_name, data_id, data_type, storage_policy, is_confidential, data_size, related_item	Vue représentant le détail des transactions soumises à la blockchain
BLOCKCHAIN_BLOCK_STAT_VIEW	index, nb_tx, size, miner, difficulty	Vue représentant les statistiques liées aux blocs de la blockchain
BLOCKCHAIN_HARDWARE_STAT_VIEW	cpu_usage, avg_cpu_usage, energy_consumption, total_energy_consumption	Vue représentant les statistiques matérielles (cpu, consommation d'énergie) pour un seul noeud de la blockchain
BLOCKCHAIN_ALL_NODE_HARDWARE_STAT_VIEW	avg_cpu_usage, total_energy_consumption	Vue représentant les statistiques matérielles globales (utilisation cpu moyenne, consommation d'énergie totale) pour l'ensemble des noeuds de la blockchain
BLOCKCHAIN_ALL_NODE_HOSTING_STAT_VIEW	total_energy_consumption, total_co2_emission, total_kwh_cost	Vue représentant les statistiques liées à la consommation d'énergie et à l'écologie (utilisation cpu moyenne, consommation d'énergie totale) pour l'ensemble des noeuds de la blockchain en fonction du lieu d'hébergement
BLOCKCHAIN_GLOBAL_HOSTING_STAT_VIEW	total_energy_consumption, total_co2_emission, total_energy_renewable_part, total_kwh_cost	Vue représentant les statistiques liées à la consommation d'énergie et à l'écologie (utilisation cpu moyenne, consommation d'énergie totale) pour l'ensemble des noeuds de la blockchain en comparant différents lieux d'hébergement

Bibliographie

[1] Visiativ-solutions.fr. *Industrie 4.0 : définition et mise en œuvre vers l'usine de production connectée*. 2020. URL : https://www.visiativ-solutions.fr/industrie-4-0/. (accessed : 01.09.2022) (cf. p. 1).

[2] Henning Kagermann, Wolfgang Wahlster et Johannes Helbig. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 – Securing the Future of German Manufacturing Industry*. Final Report of the Industrie 4.0 Working Group. München : acatech – National Academy of Science et Engineering, 2013 (cf. p. 1).

[3] Ouafae Cohin et Patrick Sondi. « Internet of things for smart factory ». In : *IEEE COMSOC MMTC E-Letter* 10 (sept. 2015) (cf. p. 1, 61).

[4] David Reinsel, John Gantz et John Rydning. *The Digitization of the World From Edge to Core*. 2018. URL : https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf. (accessed : 01.09.2022) (cf. p. 2, 3).

[5] V. Alcácer et V. Cruz-Machado. « Scanning the Industry 4.0 : A Literature Review on Technologies for Manufacturing Systems ». In : *Engineering Science and Technology, an International Journal* 22.3 (2019), p. 899-919. DOI : https://doi.org/10.1016/j.jestch.2019.01.006 (cf. p. 6).

[6] T. M. Fernández-Caramés et P. Fraga-Lamas. « A Review on the Application of Blockchain to the Next Generation of Cybersecure Industry 4.0 Smart Factories ». In : *IEEE Access* 7 (2019), p. 45201-45218. DOI : 10.1109/ACCESS.2019.2908780 (cf. p. 6).

[7] Maurice Dawson. « Cyber Security in Industry 4.0 : The Pitfalls of Having Hyperconnected Systems ». In : *Journal of Strategic Management Studies* 10.1 (2018), p. 19-28. DOI : 10.24760/iasme.10.1_19 (cf. p. 6).

[8] Marianna Lezzi, Mariangela Lazoi et Angelo Corallo. « Cybersecurity for Industry 4.0 in the current literature : A reference framework ». In : *Computers in Industry* 103 (2018), p. 97-110. DOI : https://doi.org/10.1016/j.compind.2018.09.004 (cf. p. 6, 8, 11, 12, 15, 19, 20).

[9] Jaco Prinsloo, Saurabh Sinha et Basie von Solms. « A Review of Industry 4.0 Manufacturing Process Security Risks ». In : *Applied Sciences* 9 (2019) (cf. p. 7).

[10] Mayank Shukla, S.P. Sarmah et Manoj Tiwari. « A multi-objective framework for the identification and optimisation of factors affecting cybersecurity in the Industry 4.0 supply chain ». In : *International Journal of Production Research* (juill. 2022), p. 1-16. DOI : 10.1080/00207543.2022.2100840 (cf. p. 7).

[11] Deval Bhamare et al. « Cybersecurity for industrial control systems : A survey ». In : *Computers & Security* 89 (2020), p. 101677. DOI : https://doi.org/10.1016/j.cose.2019.101677 (cf. p. 7).

[12] K. Tange et al. « A Systematic Survey of Industrial Internet of Things Security : Requirements and Fog Computing Opportunities ». In : *IEEE Communications Surveys Tutorials* 22.4 (2020), p. 2489-2520. DOI : 10.1109/COMST.2020.3011208 (cf. p. 7).

[13] Franck Sicard, Eric Zamai et Jean-Marie Flaus. « An approach based on behavioral models and critical states distance notion for improving cybersecurity of industrial control systems ». eng. In : *Reliability Engineering and System Safety* 188 (2019), p. 584-603 (cf. p. 8, 29).

[14] Juan Enrique Rubio et al. « Current cyber-defense trends in industrial control systems ». eng. In : *Computers and Security* 87 (2019) (cf. p. 8, 15, 17, 18, 33).

[15] Muhammad Rizwan Asghar, Qinwen Hu et Sherali Zeadally. « Cybersecurity in industrial control systems : Issues, technologies, and challenges ». eng. In : *Computer Networks* 165 (2019) (cf. p. 8, 9, 15, 16, 19).

[16] Tianbo Lu et al. « Cyberphysical Security for Industrial Control Systems Based on Wireless Sensor Networks ». In : *International Journal of Distributed Sensor Networks* 10.6 (2014), p. 438350. DOI : 10.1155/2014/438350 (cf. p. 8).

[17] Amer Abu Jassar et al. « Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks ». In : *Computational Intelligence and Neuroscience* 2022 (avr. 2022), p. 1-13. DOI : 10.1155/2022/5866922 (cf. p. 10).

[18] T. Pereira, L. Barreto et A. Amaral. « Network and information security challenges within Industry 4.0 paradigm ». In : *Procedia Manufacturing* 13 (2017). Manufacturing Engineering Society International Conference 2017, MESIC 2017, 28-30 June 2017, Vigo (Pontevedra), Spain, p. 1253-1260. DOI : https://doi.org/10.1016/j.promfg.2017.09.047 (cf. p. 11, 13).

[19] M.M. Alani et M. Alloghani. « Security Challenges in the Industry 4.0 Era ». In : *Industry 4.0 and Engineering for a Sustainable Future*. Sous la dir. de M. Dastbaz et P. Cochrane. "Springer Nature Switzerland AG 2019", avr. 2019. DOI : https://doi.org/10.1007/978-3-030-12953-8_8 (cf. p. 11).

[20] Miklos Kiss, Gabor Breda et Lajos Muha. « Information security aspects of Industry 4.0 ». In : *Procedia Manufacturing* 32 (2019). 12th International Conference Interdisciplinarity in Engineering, INTER-ENG 2018, 4–5 October 2018, Tirgu Mures, Romania, p. 848-855. DOI : https://doi.org/10.1016/j.promfg.2019.02.293 (cf. p. 13, 21).

[21] Angelo Corallo, Mariangela Lazoi et Marianna Lezzi. « Cybersecurity in the context of industry 4.0 : A structured classification of critical assets and business impacts ». eng. In : *Computers in Industry* 114 (2020) (cf. p. 13).

[22] Tadeusz Sawik. « A linear model for optimal cybersecurity investment in Industry 4.0 supply chains ». In : *International Journal of Production Research* (déc. 2020), p. 1-18. DOI : 10.1080/00207543.2020.1856442 (cf. p. 13).

[23] Abid Haleem et al. « Perspectives of Cybersecurity for Ameliorative Industry 4.0 Era : A Review based Framework ». In : *Industrial Robot* ahead-of-print (jan. 2022). DOI : 10.1108/IR-10-2021-0243 (cf. p. 14).

[24] Cevat Ozarpa, İsa Avcı et Aysun Eldekçi. « Industry 4.0 and Cybersecurity at Automobile Manufacturing in Smart Factories// Akıllı Fabrikalardaki Otomobil İmalatında Endüstri 4.0 ve Siber Güvenlik ». In : *Düzce Üniversitesi Bilim ve Teknoloji Dergisi* (avr. 2022) (cf. p. 14).

[25] Pedro Brandao et Paulo Duarte. « Cybersecurity risk management in the industry 4.0 ». In : *International Journal of Scientific Research And Growth* 9 (mars 2022), p. 661-668. DOI : 10.18535/ijsrm/v10i3.ec01 (cf. p. 14).

[26] Roman Rudenko et al. « A Brief Review on Internet of Things, Industry 4.0 and Cybersecurity ». In : *Electronics* 11 (mai 2022), p. 1742. DOI : 10.3390/electronics11111742 (cf. p. 15).

[27] Ziaur Rahman, Xun Yi et Ibrahim Khalil. « Blockchain based AI-enabled Industry 4.0 CPS Protection against Advanced Persistent Threat ». In : *CoRR* abs/2201.12727 (2022) (cf. p. 15).

[28] Adil Kondiloglu et al. « Information security breaches and precautions on Industry 4.0 ». eng. In : *Tehnologičnij Audit ta Rezervi Virobnictva* 6.4(38) (2017), p. 58-63 (cf. p. 16).

[29] Ahmed Omari et al. « A Closer Look on Challenges and Security Risks of Voice Over Internet Protocol Infrastructures ». In : 22 (fév. 2022), p. 175. DOI : 10.22937/IJCSNS.2022.22.2.23 (cf. p. 17).

[30] Z. Illési, A. Halász et P. J. Varga. « Wireless Networks and Critical Information Infrastructure ». In : *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. 2018, p. 000255-000260. DOI : 10.1109/SACI.2018.8441023 (cf. p. 21).

[31] B. A. A. Nunes et al. « A Survey of Software-Defined Networking : Past, Present, and Future of Programmable Networks ». In : *IEEE Communications Surveys Tutorials* 16.3 (2014), p. 1617-1634. DOI : 10.1109/SURV.2014.012214.00180 (cf. p. 22).

[32] D. Satasiya et Raviya Rupal D. « Analysis of Software Defined Network firewall (SDF) ». In : *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2016, p. 228-231. DOI : 10.1109/WiSPNET.2016.7566125 (cf. p. 22).

[33] Peng Zeng et al. « Time-slotted software-defined Industrial Ethernet for real-time Quality of Service in Industry 4.0 ». In : *Future Generation Computer Systems* 99 (2019), p. 1-10. DOI : https://doi.org/10.1016/j.future.2019.04.009 (cf. p. 22).

[34] Akihiro Tsuchiya et al. « Software defined networking firewall for industry 4.0 manufacturing systems ». eng. In : *Journal of Industrial Engineering and Management* 11.2 (2018), p. 318-333 (cf. p. 22).

[35] Kevin Ashton. « That 'Internet of Things' Thing ». In : *RFiD Journal* 22 (jan. 2009), p. 97-114 (cf. p. 24).

[36] Andrew Whitmore, Anurag Agarwal et Li Xu. « The Internet of Things—A survey of topics and trends ». In : *Information Systems Frontiers* 17 (avr. 2014). doi : 10.1007/s10796-014-9489-2 (cf. p. 24).

[37] Anum Ali et al. « Technologies and challenges in developing Machine-to-Machine applications : A survey ». In : *Journal of Network and Computer Applications* 83 (fév. 2017). doi : 10.1016/j.jnca.2017.02.002 (cf. p. 24).

[38] Luigi Atzori, Antonio Iera et Giacomo Morabito. « Understanding the Internet of Things : definition, potentials, and societal role of a fast evolving paradigm ». In : *Ad Hoc Networks* 56 (déc. 2016). doi : 10.1016/j.adhoc.2016.12.004 (cf. p. 24).

[39] J. Lin et al. « A Survey on Internet of Things : Architecture, Enabling Technologies, Security and Privacy, and Applications ». In : *IEEE Internet of Things Journal* 4.5 (2017), p. 1125-1142. doi : 10.1109/JIOT.2017.2683200 (cf. p. 24).

[40] M. Wolf et D. Serpanos. « Safety and Security in Cyber-Physical Systems and Internet-of-Things Systems ». In : *Proceedings of the IEEE* 106.1 (2018), p. 9-20. doi : 10.1109/JPROC.2017.2781198 (cf. p. 24).

[41] Dr Miraz et al. « A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT) ». In : sept. 2015, p. 219-224. doi : 10.1109/ITechA.2015.7317398 (cf. p. 24).

[42] Vikas Hassija et al. « A Survey on IoT Security : Application Areas, Security Threats, and Solution Architectures ». In : *IEEE Access* PP (juin 2019), p. 1-1. doi : 10.1109/ACCESS.2019.2924045 (cf. p. 24).

[43] Bruno Augusti Mozzaquatro et al. « An Ontology-Based Cybersecurity Framework for the Internet of Things ». eng. In : *Sensors* 18.9 (2018), p. 3053 (cf. p. 24).

[44] B. A. Mozzaquatro, R. Jardim-Goncalves et C. Agostinho. « Towards a reference ontology for security in the Internet of Things ». In : *IEEE International Workshop on Measurements Networking*. 2015, p. 1-6. doi : 10.1109/IWMN.2015.7322984 (cf. p. 25).

[45] Andre Wegner, James Graham et Eli Ribble. « A New Approach to Cyberphysical Security in Industry 4.0 ». In : *Cybersecurity for Industry 4.0 : Analysis for Design and Manufacturing*. Sous la dir. de Lane Thames et Dirk Schaefer. Cham : Springer International Publishing, 2017, p. 59-72. doi : 10.1007/978-3-319-50660-9_3 (cf. p. 25-27).

[46] Yao Pan et al. « Taxonomies for Reasoning About Cyber-physical Attacks in IoT-based Manufacturing Systems ». In : *International Journal of Interactive Multimedia and Artificial Intelligence* 4 (mars 2017), p. 45-54. doi : 10.9781/ijimai.2017.437 (cf. p. 26).

[47] J. Graham, J. Hieb et J. Naber. « Improving cybersecurity for Industrial Control Systems ». In : *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*. 2016, p. 618-623. doi : 10.1109/ISIE.2016.7744960 (cf. p. 27).

[48] Lane Thames et Dirk Schaefer. « Cybersecurity for Industry 4.0 and Advanced Manufacturing Environments with Ensemble Intelligence ». In : *Cybersecurity for Industry 4.0 : Analysis for Design and Manufacturing*. Sous la dir. de Lane Thames et Dirk Schaefer. Cham : Springer International Publishing, 2017, p. 243-265. doi : 10.1007/978-3-319-50660-9_10 (cf. p. 27, 29).

[49] Hongyu Liu et Bo Lang. « Machine Learning and Deep Learning Methods for Intrusion Detection Systems : A Survey ». In : *Applied Sciences* 9 (oct. 2019), p. 4396. doi : 10.3390/app9204396 (cf. p. 28).

[50] G. Zhao, C. Zhang et L. Zheng. « Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network ». In : *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. T. 1. 2017, p. 639-642. doi : 10.1109/CSE-EUC.2017.119 (cf. p. 28).

[51] Franck Sicard, Éric Zamai et Jean-Marie Flaus. « Critical States Distance Filter Based Approach for Detection and Blockage of Cyberattacks in Industrial Control Systems ». In : *Diagnosability, Security and Safety of Hybrid Dynamic and Cyber-Physical Systems*. Sous la dir. de Moamar Sayed-Mouchaweh. Cham : Springer International Publishing, 2018, p. 117-145. doi : 10.1007/978-3-319-74962-4_5 (cf. p. 29).

[52] S. McLaughlin et al. « The Cybersecurity Landscape in Industrial Control Systems ». In : *Proceedings of the IEEE* 104.5 (2016), p. 1039-1057. doi : 10.1109/JPROC.2015.2512235 (cf. p. 29).

[53] R. Mitchell et I. Chen. « Behavior-Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications ». In : *IEEE Transactions on Smart Grid* 4.3 (2013), p. 1254-1263. doi : 10.1109/TSG.2013.2258948 (cf. p. 29).

[54] A. Carcano et al. « A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems ». In : *IEEE Transactions on Industrial Informatics* 7.2 (2011), p. 179-186. doi : 10.1109/TII.2010.2099234 (cf. p. 31).

[55] S. Petrenko. « Developing a Cybersecurity Immune System for Industry 4.0 ». In : *Developing a Cybersecurity Immune System for Industry 4.0*. River Publishers, 2020, p. i-xlvi (cf. p. 32).

[56] Matthias Eckhart et Andreas Ekelhart. « Digital Twins for Cyber-Physical Systems Security : State of the Art and Outlook ». In : "Springer International Publishing", nov. 2019, p. 383-412. doi : 10.1007/978-3-030-25312-7_14 (cf. p. 33, 34).

[57] ENISA. *Good Practices for Security of Internet of Things in the context of Smart Manufacturing*. 2018. url : https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot/at_download/fullReport. (accessed : 01.08.2020) (cf. p. 35).

[58] ANSSI. *Managing cybersecurity for Industrial Control Systems*. 2014. url : https://www.ssi.gouv.fr/uploads/2014/01/Managing_Cybe_for_ICS_EN.pdf. (accessed : 01.08.2020) (cf. p. 35).

[59] Najam Zia et al. « A Managerial Review and Guidelines for Industry 4.0 Factories on Cybersecurity ». In : *European Conference on Cyber Warfare and Security* 21 (juin 2022), p. 336-340. doi : 10.34190/eccws.21.1.499 (cf. p. 39).

[60] Benoit Charlot, Didier Montet et Ramesh Ray. *Food Traceability and Authenticity*. Nov. 2017 (cf. p. 60).

[61] R. Badia-Melis, P. Mishra et L. Ruiz-García. « Food traceability : New trends and recent advances. A review ». In : *Food Control* 57 (2015), p. 393-401. doi : https://doi.org/10.1016/j.foodcont.2015.05.005 (cf. p. 60).

[62] Petter Olsen et Melania Borit. « The components of a food traceability system ». In : *Trends in Food Science and Technology* 77 (2018), p. 143-149. doi : https://doi.org/10.1016/j.tifs.2018.05.004 (cf. p. 60, 61).

[63] Petter Olsen et Melania Borit. « How to define traceability ». In : *Trends in Food Science and Technology* 29.2 (2013), p. 142-150. doi : https://doi.org/10.1016/j.tifs.2012.10.003 (cf. p. 61).

[64] Reuben Schuitemaker et Xun Xu. « Product traceability in manufacturing : A technical review ». In : *Procedia CIRP* 93 (2020). 53rd CIRP Conference on Manufacturing Systems 2020, p. 700-705. doi : https://doi.org/10.1016/j.procir.2020.04.078 (cf. p. 61, 63).

[65] Techsol S.r.l. *INDUSTRY 4.0 : THE IMPORTANCE OF TRACEABILITY*. 2017. url : https://medium.com/@Techsol_srl/industry-4-0-the-importance-of-traceability-c9c05691676f. (accessed : 01.06.2022) (cf. p. 61).

[66] Cameron Shearon. « IPC-1782 standard for traceability of critical items based on risk ». In : fév. 2018, p. 1-3. doi : 10.23919/PanPacific.2018.8318996 (cf. p. 61).

[67] GS1 Global Traceability Standard. *GS1's framework for the design of interoperable traceability systems for supply chains*. Accessed : 5-6-2022 (cf. p. 61).

[68] Abdesselam Bougdira, Akharraz Ismail et Abdelaziz Ahaitouf. « A traceability proposal for industry 4.0 ». In : *Journal of Ambient Intelligence and Humanized Computing* 11 (août 2020). doi : 10.1007/s12652-019-01532-7 (cf. p. 61, 66).

[69] Ray Y. Zhong, Xun Xu et Lihui Wang. « IoT-enabled Smart Factory Visibility and Traceability Using Laser-scanners ». In : *Procedia Manufacturing* 10 (2017). 45th SME North American Manufacturing Research Conference, NAMRC 45, LA, USA, p. 1-14. doi : https://doi.org/10.1016/j.promfg.2017.07.103 (cf. p. 61).

[70] Fabien Bibi et al. « A review : RFID technology having sensing aptitudes for food industry and their contribution to tracking and monitoring of food products ». eng. In : *Trends in Food Science And Technology* 62 (2017), p. 91-103 (cf. p. 61).

[71] Ghadafi M. Razak, Linda Caroline Hendry et Mark Stevenson. « Supply chain traceability : a review of the benefits and its relationship with supply chain resilience ». In : *Production Planning & Control* (2021) (cf. p. 61).

[72] Naeem Firdous Syed et al. « Traceability in supply chains : A Cyber security analysis ». In : *Computers & Security* 112 (2022), p. 102536. doi : https://doi.org/10.1016/j.cose.2021.102536 (cf. p. 61, 75).

[73] Rafael Bettín-Díaz, Alix E. Rojas et Camilo Mejía-Moncayo. « Methodological Approach to the Definition of a Blockchain System for the Food Industry Supply Chain Traceability ». In : *Computational Science and Its Applications – ICCSA 2018*. Sous la dir. d'Osvaldo Gervasi et al. Cham : Springer International Publishing, 2018, p. 19-33 (cf. p. 62).

[74] J Barata et al. « A Systematic Approach to Design Product Traceability in Industry 4.0 : Insights from the Ceramic Industry ». In : *Information Systems Development : Advances in Methods, Tools and Management - Proceedings of the 26th International Conference on Information Systems Development, ISD 2017, Larnaca, Cyprus, University of Central Lancashire Cyprus, September 6-8, 2017*. Sous la dir. de Nearchos Paspallis et al. ISD. Citations : dblp. 2017 (cf. p. 62, 66).

[75] H. Panetto, M. Dassisti et A. Tursi. « ONTO-PDM : Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment ». In : *Advanced Engineering Informatics* 26.2 (2012). Knowledge based engineering to support complex product design, p. 334-348. doi : https://doi.org/10.1016/j.aei.2011.12.002 (cf. p. 62).

[76] Haya R. Hasan et al. « Blockchain-Based Solution for the Traceability of Spare Parts in Manufacturing ». In : *IEEE Access* 8 (2020), p. 100308-100322. doi : 10.1109/ACCESS.2020.2998159 (cf. p. 62, 66, 91, 102).

[77] Wala' Alkhader et al. « Blockchain-Based Traceability and Management for Additive Manufacturing ». In : *IEEE Access* 8 (2020), p. 188363-188377. doi : 10.1109/ACCESS.2020.3031536 (cf. p. 62, 71, 102).

[78] Jiewu Leng et al. « Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0 : A survey ». In : *Renewable and Sustainable Energy Reviews* 132 (2020), p. 110112. doi : https://doi.org/10.1016/j.rser.2020.110112 (cf. p. 62, 72, 102).

[79] Stefan Tönnissen et Frank Teuteberg. « Analysing the impact of blockchain-technology for operations and supply chain management : An explanatory model drawn from multiple case studies ». In : *International Journal of Information Management* 52 (2020), p. 101953 (cf. p. 62, 102).

[80] Manal Hader et al. « Applying integrated Blockchain and Big Data technologies to improve supply chain traceability and information sharing in the textile sector ». In : *Journal of Industrial Information Integration* 28 (2022), p. 100345. doi : https://doi.org/10.1016/j.jii.2022.100345 (cf. p. 62, 102).

[81] Xiaojing Xu et al. « Blockchain applications in the supply chain management in German automotive industry ». In : *Production Planning & Control* 0.0 (2022), p. 1-15. doi : 10.1080/09537287.2022.2044073 (cf. p. 62).

[82] Hongwu Bai et al. « Traceability technologies for farm animals and their products in China ». In : *Food Control* 79 (2017), p. 35-43. doi : https://doi.org/10.1016/j.foodcont.2017.02.040 (cf. p. 63).

[83] *Revisiter la traçabilité avec les technologies industrie 4.0*. eng. Editions T.I., 2020 (cf. p. 64).

[84] Kristín Óskarsdóttir et Guðmundur Valur Oddsson. « Towards a decision support framework for technologies used in cold supply chain traceability ». In : *Journal of Food Engineering* 240 (2019), p. 153-159. doi : https://doi.org/10.1016/j.jfoodeng.2018.07.013 (cf. p. 64).

[85] DynamSoft. *The Comprehensive Guide to 1D and 2D Barcodes*. 2020. url : https://www.dynamsoft.com/blog/insights/the-comprehensive-guide-to-1d-and-2d-barcodes/. (accessed : 01.06.2022) (cf. p. 64).

[86] GS1 Datamatrix Guideline. *Overview and technical introduction to the use of GS1 DataMatrix*. 2018. URL : https://www.gs1.org/docs/barcodes/GS1_DataMatrix_Guideline.pdf. (accessed : 01.06.2022) (cf. p. 64).

[87] Ioakeim Tzoulis et Zaharoula Andreopoulou. « Emerging Traceability Technologies as a Tool for Quality Wood Trade ». In : *Procedia Technology* 8 (2013). 6th International Conference on Information and Communication Technologies in Agriculture, Food and Environment (HAICTA 2013), p. 606-611. DOI : https://doi.org/10.1016/j.protcy.2013.11.087 (cf. p. 64).

[88] Diana M. Segura Velandia et al. « Towards industrial internet of things : Crankshaft monitoring, traceability and tracking using RFID ». In : *Robotics and Computer-Integrated Manufacturing* 41 (2016), p. 66-77. DOI : https://doi.org/10.1016/j.rcim.2016.02.004 (cf. p. 64).

[89] James Macaulay et Markus Kückelhaus. *Internet of things in logistics : Technical report*. 2015. URL : https://discover.dhl.com/content/dam/dhl/downloads/interim/full/dhl-trend-report-internet-of-things.pdf. (accessed : 01.09.2022) (cf. p. 66).

[90] Xinwu Li et al. « Developing a Real-time Monitoring Traceability System for Cold Chain of Tricholoma matsutake ». In : *Electronics* 8 (avr. 2019), p. 423. DOI : 10.3390/electronics8040423 (cf. p. 66).

[91] Jabir Arif, Fouad Jawab et YOUSSEF MOUZOUNA. « Design on Improvement of Traceability Process in the Outsourcing of Logistics' Activities Using the Internet of Things (IoT) Applications ». In : *Maejo international journal of science and technology* 29 (jan. 2020), p. 1093-1108 (cf. p. 66).

[92] Lucía Guzmán. *SIGFOX AND LOUIS VUITTON PARTNER FOR INNOVATIVE LUGGAGE TRACKER*. 2018. URL : https://www.sigfox.com/en/news/sigfox-and-louis-vuitton-partner-innovative-luggage-tracker. (accessed : 01.09.2022) (cf. p. 66).

[93] Sylvere Krima, Thomas Hedberg Jr. et Allison Barnard Feeney. *Securing the Digital Threat for Smart Manufacturing : A Reference model for blockchain-based product data traceability*. en. Fév. 2019. DOI : https://doi.org/10.6028/NIST.AMS.300-6. URL : https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=926019 (cf. p. 69).

[94] Satoshi Nakamoto. « Bitcoin : A Peer-to-Peer Electronic Cash System ». In : *Cryptography Mailing list at https ://metzdowd.com* (mars 2009) (cf. p. 70, 72).

[95] Umesh Bodkhe et al. « Blockchain for Industry 4.0 : A Comprehensive Review ». In : *IEEE Access* 8 (2020), p. 79764-79800. DOI : 10.1109/ACCESS.2020.2988579 (cf. p. 70, 71, 74, 102).

[96] Nick Szabo. « Formalizing and Securing Relationships on Public Networks ». In : *First Monday* 2 (jan. 1997). DOI : 10.5210/fm.v2i9.548 (cf. p. 72).

[97] Vitalik Buterin. *Ethereum Whitepaper*. 2014. URL : https://ethereum.org/en/whitepaper/. (accessed : 01.09.2022) (cf. p. 72).

[98] Vitalik Butterin. *On Public and Private Blockchains*. 2015. URL : https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains. (accessed : 01.09.2022) (cf. p. 72).

[99] Faraz Masood et Arman Rasool Faridi. « Consensus Algorithms In Distributed Ledger Technology For Open Environment ». In : *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. 2018, p. 1-6. DOI : 10.1109/CCAA.2018.8777695 (cf. p. 73).

[100] Sunny King et Scott Nadal. « PPCoin : Peer-to-Peer Crypto-Currency with Proof-of-Stake ». In : (sept. 2022) (cf. p. 73).

[101] Miguel Castro et Barbara Liskov. « Practical Byzantine Fault Tolerance ». In : *OSDI* (mars 1999) (cf. p. 73).

[102] Elli Androulaki et al. « Hyperledger Fabric : A Distributed Operating System for Permissioned Blockchains ». In : (jan. 2018) (cf. p. 73).

[103] Kelly Olson et al. *Sawtooth : An Introduction*. 2018. URL : https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf. (accessed : 01.09.2022) (cf. p. 74).

[104] Valentin Mullet, Patrick Sondi et Eric Ramat. « A Review of Cybersecurity Guidelines for Manufacturing Factories in Industry 4.0 ». In : *IEEE Access* 9 (2021), p. 23235-23263. doi : 10.1109/ACCESS.2021.3056650 (cf. p. 74, 102).

[105] Yong Chen et al. « Applications of Blockchain in Industry 4.0 : a Review ». eng. In : *Information systems frontiers* (2022) (cf. p. 74).

[106] Kaushal Shah et al. « Exploring applications of blockchain technology for Industry 4.0 ». In : *Materials Today : Proceedings* 62 (2022). International Conference on Additive Manufacturing and Advanced Materials (AM2), p. 7238-7242. doi : https://doi.org/10.1016/j.matpr.2022.03.681 (cf. p. 74).

[107] Juan F. Galvez, J.C. Mejuto et J. Simal-Gandara. « Future challenges on the use of blockchain for food traceability analysis ». In : *TrAC Trends in Analytical Chemistry* 107 (2018), p. 222-232. doi : https://doi.org/10.1016/j.trac.2018.08.011 (cf. p. 74).

[108] Harsha Chauhan et al. « Framework for Enhancing the Traceability in Supply Chain Using Blockchain ». In : *Journal of Interconnection Networks* (fév. 2022). doi : 10.1142/S0219265921440084 (cf. p. 74).

[109] Nader Mohamed et Jameela Al-Jaroodi. « Applying Blockchain in Industry 4.0 Applications ». In : *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 2019, p. 0852-0858. doi : 10.1109/CCWC.2019.8666558 (cf. p. 74).

[110] Jiewu Leng et al. « Blockchain-Secured Smart Manufacturing in Industry 4.0 : A Survey ». In : *IEEE Transactions on Systems, Man, and Cybernetics : Systems* 51.1 (2021), p. 237-252. doi : 10.1109/TSMC.2020.3040789 (cf. p. 74, 102).

[111] Taehyun Ko, Jaeram Lee et Doojin Ryu. « Blockchain Technology and Manufacturing Industry : Real-Time Transparency and Cost Savings ». In : *Sustainability* 10.11 (2018). doi : 10.3390/su10114274 (cf. p. 74, 102).

[112] Miguel Rodrigo Pincheira Caro, Massimo Vecchio et Raffaele Giaffreda. « Characterization and Costs of Integrating Blockchain and IoT for Agri-Food Traceability Systems ». In : *Systems* 10 (avr. 2022), p. 57. doi : 10.3390/systems10030057 (cf. p. 74).

[113] Pietro De Giovanni. « Blockchain and smart contracts in supply chain management : A game theoretic model ». In : *International Journal of Production Economics* 228 (2020), p. 107855. doi : https://doi.org/10.1016/j.ijpe.2020.107855 (cf. p. 74, 75, 103).

[114] Jay Lee, Moslem Azamfar et Jaskaran Singh. « A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems ». In : *Manufacturing Letters* 20 (2019), p. 34-39. doi : https://doi.org/10.1016/j.mfglet.2019.05.003 (cf. p. 74, 103).

[115] Zhi Li, Ali Vatankhah Barenji et George Q. Huang. « Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform ». In : *Robotics and Computer-Integrated Manufacturing* 54 (2018), p. 133-144. doi : https://doi.org/10.1016/j.rcim.2018.05.011 (cf. p. 75, 103).

[116] Tiago M. Fernández-Caramés et al. « Towards an Autonomous Industry 4.0 Warehouse : A UAV and Blockchain-Based System for Inventory and Traceability Applications in Big Data-Driven Supply Chain Management ». In : *Sensors* 19.10 (2019). doi : 10.3390/s19102394 (cf. p. 75).

[117] Yunsen Wang et Alexander Kogan. « Designing confidentiality-preserving Blockchain-based transaction processing systems ». In : *International Journal of Accounting Information Systems* 30 (2018). 2017 Research Symposium on Information Integrity and Information Systems Assurance, p. 1-18. doi : https://doi.org/10.1016/j.accinf.2018.06.001 (cf. p. 75, 103).

[118] Miguel Pincheira Caro et al. « Blockchain-based traceability in Agri-Food supply chain management : A practical implementation ». In : *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*. 2018, p. 1-4. doi : 10.1109/IOT-TUSCANY.2018.8373021 (cf. p. 96).

[119] Thomas Bocek et al. « Blockchains everywhere - a use-case of blockchains in the pharma supply-chain ». In : mai 2017, p. 772-777. doi : 10.23919/INM.2017.7987376 (cf. p. 97).

[120] Fátima Leal et al. « Smart Pharmaceutical Manufacturing : Ensuring End-to-End Traceability and Data Integrity in Medicine Production ». In : *Big Data Research* 24 (jan. 2021), p. 100172. doi : 10.1016/j.bdr.2020.100172 (cf. p. 97).

[121] A. Shanley. *Could Blockchain Improve Pharmaceutical Supply Chain Security ?* 2017. URL : https://www.pharmtech.com/view/could-blockchain-improve-pharmaceutical-supply-chain-security. (accessed : 01.09.2022) (cf. p. 97).

[122] Qijun Lin et al. « Food Safety Traceability System based on Blockchain and EPCIS ». In : *IEEE Access* PP (fév. 2019), p. 1-1. DOI : 10.1109/ACCESS.2019.2897792 (cf. p. 97).

[123] Fran Casino et al. « Modeling food supply chain traceability based on blockchain technology ». In : *IFAC-PapersOnLine* 52 (jan. 2019), p. 2728-2733. DOI : 10.1016/j.ifacol.2019.11.620 (cf. p. 97).

[124] Daniel Bumblauskas et al. « A blockchain use case in food distribution : Do you know where your food has been ? » In : *International Journal of Information Management* 52 (oct. 2019). DOI : 10.1016/j.ijinfomgt.2019.09.004 (cf. p. 97).

[125] Bernard Zeigler. *Theory of Modeling and Simulation*. Jan. 1976 (cf. p. 100, 101).

[126] S. Chen et al. « Blockchain applications in PLM towards smart manufacturing ». In : *Int J Adv Manuf Technol* 118 (2022), p. 2669-2683. DOI : https://doi.org/10.1007/s00170-021-07802-z (cf. p. 102).

[127] Ru Huo et al. « A Comprehensive Survey on Blockchain in Industrial Internet of Things : Motivations, Research Progresses, and Future Challenges ». In : *IEEE Communications Surveys & Tutorials* 24.1 (2022), p. 88-122. DOI : 10.1109/COMST.2022.3141490 (cf. p. 102).

[128] Valentin Mullet, Patrick Sondi et Eric Ramat. « Enhancing Trust in Industry 4.0 Traceability Data using Confidentiality-Preserving Digital Ledger ». In : *4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. Paris : IEEE, 2022 (cf. p. 103, 104).

[129] Sara Bergman - Microsoft DevBlogs. *How Can I Calculate CO2eq emissions for my Azure VM ?* 2020. URL : https://devblogs.microsoft.com/sustainable-software/how-can-i-calculate-co2eq-emissions-for-my-azure-vm. (accessed : 01.06.2022) (cf. p. 111).

[130] Eric Ramat - ULCO. *Artis* : DEVS multimodeling and simulation framework*. 2022. URL : https://gitlab.com/artis-star/artis-star. (accessed : 01.09.2022) (cf. p. 148).

Index Alphabétique

Intégration de solution blockchain dans un système global de traçabilité d'une usine opérationnelle
Par ***Valentin Mullet***

Résumé :

L'industrie 4.0 implique des changements majeurs dans la gestion des processus de fabrication. L'Internet des objets et le cloud computing permettent des interactions en ligne entre des tiers, tels que des clients et des fournisseurs, avec le système de traçabilité d'une usine. La blockchain industrielle offre un paradigme qui permet de mettre en place un registre infalsifiable de transactions impliquant plusieurs partenaires, individuellement libres d'y inscrire des actions et de vérifier les actions des autres de manière ad hoc. En ce sens, elle permet de mettre en place une traçabilité fiable, transparente et directement vérifiable par chaque partenaire (fournisseur/usine, usine/sous-traitant, usine-client, etc). A l'ère des usines connectées, elle constitue une solution qui peut être mise en œuvre entre le système d'information de l'usine et les différents partenaires pour automatiser la gestion de la traçabilité. Néanmoins, le système global de traçabilité de l'usine peut comporter des composantes internes avec des implications fortes sur la confidentialité et dont le lien avec la traçabilité n'incombe qu'à l'usine et pas à ses partenaires. Dans ce contexte, la transparence apportée se ferait au détriment de la confidentialité des données. De plus, les évaluations proposées sur la blockchain portent en général sur ses performances en tant que technologie, plutôt que sur son impact global sur le système industriel, ce qui peut être un frein à son adoption.

Ce travail de thèse vise à développer une architecture et proposer des évolutions des systèmes de traçabilité pour permettre l'intégration d'une solution blockchain destinée à garantir la traçabilité entre l'usine et les partenaires sans qu'il n'y ait compromission de la confidentialité au motif de la transparence. Dans un premier temps, un état des lieux de l'usine sera présentée sous l'angle de la cybersécurité avec pour contribution une division de l'usine sous forme de périmètres en vue d'aboutir à une synthèse de bonnes pratiques adaptées à chaque périmètre en matière de sécurité informatique. Ensuite, nous définissons une approche de traçabilité centrée sur le produit soulignant l'implication des partenaires ainsi qu'une catégorisation des données selon leur criticité, ce qui permet de gérer leur accès et leur stockage au niveau des différents partenaires ainsi que dans l'infrastructure de l'usine. La présentation de l'architecture blockchain viendra dans un troisième temps avec une implémentation via la plateforme Multichain ainsi qu'une discussion sur la consommation en termes énergétique mais également en termes de stockage de la blockchain. Enfin, une modélisation à évènement discret pouvant être adaptée à n'importe quelle usine de production industrielle est également proposée afin d'analyser l'impact d'une solution blockchain sur une usine opérationnelle en fonction de différentes métriques telles que les besoins en capacité de stockage, la consommation d'énergie et l'impact environnemental.

Mots-Clés : *traçabilité ; blockchain ; usine 4.0; big data ; sécurité ; simulation*

Abstract :

Industry 4.0 involves major changes in manufacturing process management. Both the Internet of Things and cloud computing allow online interactions between third parties, such as providers, customers and suppliers, with the traceability system of a factory. The industrial blockchain offers a paradigm that makes it possible to set up an unfalsifiable register of transactions involving several partners, individually free to register actions and to verify the actions of others on an ad hoc basis. In this sense, it makes it possible to set up reliable, transparent and directly verifiable traceability by each partner (supplier/factory, factory/subcontractor, factory-customer, etc.). In the era of connected factories, it is a solution that can be implemented between the factory's information system and the various partners to automate traceability management. However, the plant's overall traceability system may include internal components with strong implications on confidentiality, and whose link with traceability lies only with the plant and not with its partners. In this context, the transparency would be obtained to the detriment of data confidentiality. In addition, the evaluations proposed on the blockchain often focus on its own performance rather than on its impact on the industrial system, which is one of the obstacles to its massive adoption.

This thesis work aims to develop an architecture and propose changes to traceability systems in order to allow the integration of a blockchain solution intended to guarantee traceability in full transparency between the factory and the partners without compromising the confidentiality. Firstly, an inventory of the factory will be presented regarding cybersecurity, thus introducing our contribution to a division of the factory in the form of perimeters which will guide our synthesis of good practices regarding the manufacturing system security. Then, we define a product-centered traceability approach underlining the implication of the partners as well as a categorization of the data according to their criticality. The presentation of the blockchain architecture will come in a third step along with its implementation using the Multichain platform as well as a discussion on its resource consumption, such as energy and storage volume related to the blockchain solution. Finally, a discrete event modeling that can be adapted to any manufacturing factory plant will be proposed in order to analyze the impact of a blockchain solution on the overall factory plant, according to different metrics such as storage volume needs, energy consumption, and environmental impact.

Keywords : *traceability ; blockchain ; factory 4.0; big data ; security ; simulation*