THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de « l'INSA Rouen Normandie »

Implémentation industrielle du "Fog Manufacturing" : défis et applications à la logistique interne dans le contexte de l'Industrie du futur.

Présentée et soutenue le 02/07/2021 par
Imen BOUZARKOUNA

Thèse soutenue publiquement le (date de soutenance) devant le jury composé de		
Leila MERGHEM-BOULAHIA	Professeur des Universités, Université de Technologie de Troyes	Rapporteur
Nathalie MITTON	Directrice de Recherche, INRIA Lille Nord-Europe	Rapporteur
Gülgün ALPAN-GAUJAL	Professeur des Universités, Université de Grenoble Alpes	Examinateur,
Aziz BENSRHAIR	Professeur des Universités, INSA Rouen	Président de jury,
Mohammed DAHANE	Maître de Conférences, HDR, Université de Lorraine	Examinateur,
David BAUDRY	Directeur de Recherche CESI, CESI Rouen	Directeur de thèse,
Christian GOUT	Professeur des Universités, INSA Rouen	Directeur de thèse,
M'hammed SAHNOUN	Enseignant-Chercheur, HDR, CESI Rouen	Co-encadrant

Thèse dirigée par D. Baudry (LINEACT, CESI Rouen) et C. Gout (LMI, INSA Rouen)

INSA ROUEN NORMANDIE

ED 590 MIIS Mathématiques, Information, Ingénierie des Systèmes

LMI - EA 3226

LINEACT cesi

Table des matières

Remerciements xv

Résumé xvii

1 Introduction générale 1

2 Contexte & État de l'art 5
2.1 Introduction . . . 5
2.2 L'Industrie 4.0 : État de l'art . . . 7
2.2.1 L'historique des révolutions industrielles . . . 7
2.2.2 La relation entre les données et les révolutions industrielles . . . 7
2.2.3 Les différentes appellations de l'Industrie 4.0 . . . 9
2.2.4 Définition . . . 9
2.2.5 Composants . . . 11
Internet of Things (IoT) . . . 11
Cyber physical system (CPS) . . . 12
Internet of Services (IoS) . . . 14
Big Data . . . 15
Les capteurs (Sensors) . . . 15
Cloud Computing et Cloud Manufacturing . . . 16
2.2.6 Intégration . . . 17
2.2.7 Les exigences de l'Industrie 4.0 . . . 18
2.3 Problématique générale . . . 19
2.3.1 Contexte . . . 19
2.3.2 Le rôle et l'intérêt du Cloud Computing dans les applications industrielles . . . 20
2.3.3 Edge computing - solutions existantes et déclinaisons . . . 21
Cloudlets . . . 21
Mobile Edge Computing (MEC) . . . 22
Fog Computing . . . 23
2.4 Fog Computing . . . 25
2.4.1 État de l'art . . . 25
Définition . . . 26
Les travaux intégrant le Fog Computing . . . 27
Les caractéristiques . . . 28
L'évolution de l'architecture . . . 30
2.4.2 Du Cloud Computing au Fog Computing . . . 31
2.5 Conclusion . . . 34

3 Le Fog Computing dans le contexte de l'Industrie 4.0 37
3.1 Introduction . . . 37
3.2 Étude de l'intégration du Fog Computing dans le contexte de l'Industrie 4.0 . . . 37

3.2.1 Le Fog Computing répond-il aux exigences de l'Industrie 4.0 ? 38
3.2.2 Les domaines d'application existants du Fog Manufacturing . . 39
Maintenance prédictive . . . 39
Traçabilité . . . 40
Monitoring . . . 42
Diagnostic et pronostic . . . 43
Consommation d'énergie . . . 43
Logistique . . . 44
Planification ou ordonnancement . . . 45
3.2.3 Analyse et synthèse . . . 45
3.3 Les défis limitant l'intégration du Fog computing . . . 46
3.3.1 Les défis "Input" . . . 48
Programmabilité (P) . . . 48
Sécurité (S) . . . 50
Hétérogénéité (H) . . . 52
Interopérabilité (I) . . . 54
3.3.2 Les défis "Output" . . . 56
Qualité de Service (QoS) . . . 56
Consommation d'énergie . . . 59
Coût . . . 60
3.3.3 Les liens entre les défis . . . 62
3.4 Les possibles applications du fog computing dans le contexte de l'Industrie 4.0 . . . 66
3.5 Conclusion . . . 69

4 Déploiement optimal du Fog Computing pour la logistique interne dans le contexte de l'Industrie 4.0 71
4.1 Introduction . . . 71
4.2 La logistique dans le contexte de l'Industrie 4.0 . . . 72
4.2.1 Contexte . . . 72
4.2.2 Problématique du déploiement du Fog Manufacturing . . . 72
4.3 L'architecture du Fog Manufacturing . . . 77
La couche des terminaux . . . 77
La couche du Edge computing . . . 78
La couche du Fog computing . . . 79
La couche du passerelle . . . 80
La couche du Cloud computing . . . 81
4.4 L'optimisation du coût de déploiement du Fog Manufacturing . . . 82
4.4.1 Paramètres . . . 83
4.4.2 La fonction objectif . . . 83
4.4.3 Les contraintes de base . . . 84
L'activation des nœuds . . . 84
La capacité de demande des nœuds . . . 85
La latence des transmissions . . . 86
La couverture des AGVs par les nœuds Edge . . . 87
Le nombre maximal des liens supporté par un nœud . . . 87
4.4.4 Nos contributions . . . 87
La linéarisation . . . 88
La probabilité de la capacité de réseau . . . 88
Le type de lien . . . 91
4.5 L'optimisation du modèle . . . 91

4.5.1 La méthode exacte . . . 92
4.5.2 Description des cas d'étude . . . 92
4.5.3 Résolution et résultats . . . 94
Le modèle sans linéarisation . . . 94
Le modèle avec linéarisation & sans nouvelles contraintes . . . 94
Étude de la sensibilité . . . 98
Le modèle avec la contrainte de la probabilité de demande de réseau . . . 101
Le modèle avec la contrainte du choix du lien . . . 104
4.5.4 Perspectives . . . 105
4.6 Conclusion . . . 107

5 Déploiement d'une plateforme Fog Computing pour le traitement en temps réel d'un CPS de production 109
5.1 Introduction . . . 109
5.2 Conteneurisation . . . 109
5.2.1 Définition . . . 109
5.2.2 Conteneurs vs. machines virtuelles . . . 110
5.2.3 Les plateformes de conteneurisation . . . 111
Les plateformes Linux . . . 112
La plate-forme Docker . . . 113
5.2.4 Moteurs d'orchestration des conteneurs . . . 113
5.2.5 Exemple de moteurs d'orchestration . . . 114
5.3 Conception et spécifications des besoins fonctionnels . . . 116
5.3.1 Diagramme de cas d'utilisation . . . 116
Description des acteurs . . . 116
Diagramme de cas d'utilisation globale . . . 117
5.3.2 Diagramme de séquences . . . 117
5.4 Choix Technologiques . . . 119
5.4.1 Les FNs : Raspberry Pi . . . 119
Le choix des FNs . . . 119
Evolution des cartes Raspberry Pi . . . 120
Fiche technique du Raspberry Pi3 . . . 120
5.4.2 Langage de programmation, Python . . . 121
5.4.3 Logiciel libre de conteneurisation, Docker . . . 121
5.4.4 Orchestrateur des conteneurs, Kubernetes . . . 122
5.5 Connectivité . . . 122
5.6 Réalisation . . . 123
5.6.1 Déroulement . . . 123
Première phase : sans intégration des robots . . . 124
Deuxième phase : avec intégration des robots . . . 124
5.6.2 Test et résultats de l'implémentation technique . . . 126
Développement & Test de l'application sur une Machine Virtuelle . . . 127
Création du conteneur . . . 127
Déploiement de l'image dans le cluster . . . 127
5.6.3 Difficultés techniques rencontrées . . . 129
Communication entre les noeuds de l'architecture . . . 129
Traitement temps réel . . . 133
5.6.4 Perspectives . . . 133
5.6.5 Conclusion . . . 135

6 Conclusion générale **137**

Table des figures

1.1 Le Fog et le Cloud computing. 2

2.1 La succession des quatre révolutions industrielles. 8
2.2 Nombre de publications concernant la thématique Industrie 4.0 selon l'appellation. 10
2.3 L'Architecture 5C d'implémentation du CPS (LEE, AZAMFAR et SINGH, 2019). 13
2.4 Les trois types d'intégration de l'Industrie 4.0 (WANG et al., 2016). . . . 18
2.5 L'architecture basée sur les Cloudlets (PAN et MCELHANNON, 2017). . 22
2.6 L'architecture basée sur le MEC (ELGENDY et al., 2019). 23
2.7 L'architecture basée sur le Fog computing. 24
2.8 Les couches de l'architecture basée sur le Fog Computing. 31
2.9 La comparaison entre l'approche Cloud et Fog Computing. 31

3.1 L'architecture du système de traçabilité basé sur le Fog computing (CHEN, 2017). 40
3.2 L'architecture du système d'IAR basé sur le Fog computing (FRAGA-LAMAS et al., 2019). 41
3.3 L'architecture du système basé sur le Fog computing et le protocole MQTT (PERALTA et al., 2017). 44
3.4 Les défis généraux et industriels. 47
3.5 Les liens entre les défis input and output du Fog Manufacturing. . . . 48
3.6 Les PIs d'entrées/sorties de la programmabilité. 50
3.7 Les solutions possibles pour la problématique de sécurité des données (PUTHAL et al., 2019). 51
3.8 Les PIs d'entrées/sorties de la sécurité. 52
3.9 Les PIs d'entrées/sorties de l'hétérogénéité. 53
3.10 Les entrées/sorties de l'interopérabilité. 55
3.11 Les entrées/sorties de la QoS. 57
3.12 Les entrées/sorties de la consommation d'énergie. 60
3.13 Les entrées/sorties du coût. 61
3.14 Les liaisons complètes entre les défis Input et Output. 62
3.15 Les défis liés au Fog Computing. 63

4.1 Le processus de décision opérationnelle basé sur le Fog Manufacturing. 73
4.2 Le problème de déploiement des ENs en terme de mobilité. 75
4.3 Le problème de déploiement des noeuds en terme de consommation d'énergie. 75
4.4 L'architecture basée sur le Fog Manufacturing. 77
4.5 Les trois types d'intégrations de l'Industrie 4.0. 80
4.6 Le principe de la technologie ParStream. 80
4.7 Le principe de la technologie Vortex. 81
4.8 Transmission d'un dispositif mobile (AGV). 89

4.9 Le plan du laboratoire LINEACT CESI. 89
4.10 Les probabilités de chaque cellule en fonction de la mobilité des AGVs. 90
4.11 Le plan du laboratoire LINEACT CESI avec l'emplacement potentiel des différents nœuds. 93
4.12 Le plan et l'emplacement potentiel de différents noeuds de la petite instance du modèle. 95
4.13 Les noeuds activés pour la petite instance du modèle avec la linéarisation. 96
4.14 Le coût et le temps de résolution pour la petite instance du modèle avec la linéarisation. 96
4.15 Le plan et l'emplacement potentiel de différents noeuds pour la grande instance du modèle. 97
4.16 Les nœuds activés pour la grande instance du modèle avec la linéarisation. 97
4.17 Le coût et le temps de résolution pour la grande instance du modèle avec la linéarisation. 98
4.18 les courbes de sortie de l'étude de sensibilité pour la variation de r et Rn. 102
4.19 Les noeuds activés pour la petite instance du modèle avec la contrainte de probabilité de demande de réseau. 103
4.20 Le coût et le temps de résolution pour la petite instance du modèle avec la contrainte de probabilité de demande de réseau. 103
4.21 Les noeuds activés pour la grande instance du modèle avec la contrainte de probabilité de demande de réseau. 104
4.22 Le coût et le temps de résolution pour la grande instance du modèle avec la contrainte de probabilité de demande de réseau. 105
4.23 Les noeuds activés pour la petite instance du modèle avec la contrainte du choix du lien. 106
4.24 Le coût et le temps de résolution pour la petite instance du modèle avec la contrainte du choix du lien. 106

5.1 Containers vs. VMs (KOSKINEN, 2016). 111
5.2 Le moteur d'orchestration Kubernets(*Kubernetes*). 114
5.3 Le moteur d'orchestration Docker Swarm (*Swarm*). 115
5.4 Diagramme de cas d'utilisation globale du système. 117
5.5 Diagramme de séquence de l'environement d'application. 118
5.6 Carte Raspberry Pi3 (RASPBERRYPI., 2020). 120
5.7 Utilisation de Python par rapport aux autres langages. 121
5.8 Les liens entre les composants de l'architecture basée sur le Fog Manufacturing. 123
5.9 Implémentation sans intégration des robots. 124
5.10 Temps de Transfert des données. 125
5.11 Données sous format XML. 126
5.12 Données sous format JSON. 126
5.13 Application sous machine virtuelle. 128
5.14 Docker File. 129
5.15 Building de l'image Docker. 130
5.16 Docker Hub. 131
5.17 Cluster de Raspberry. 131
5.18 Initialisation du Cluster. 132
5.19 Distribution des noeuds. 132

5.20 Déploiement de l'image Docker. 132
5.21 Kubernetes Dashboard. 133
5.22 Fichier de configuration Kubernetes. 134

Liste des tableaux

2.1 Composants de l'Industrie 4.0 proposés par les principaux pays économiques développés. 11
2.2 Comparaison entre les déclinaisons de l'Edge computing (DOLUI et DATTA, 2017, AI, PENG et ZHANG, 2018). 25
2.3 Comparaison du Cloud computing et du Fog computing. 34

3.1 Les cas d'implémentation du Fog computing dans le contexte d'Industrie 4.0. 47
3.2 Les applications possibles du Fog Manufacturing. 68

4.1 Définition des paramètres. 83
4.2 Définition des variables de décisions. 84
4.3 Définition des variables. 84
4.4 Les résultats de la variation du Rn avec r=5. 99
4.5 Les résultats de la variation du Rn avec r=10. 99
4.6 Les résultats de la variation du Rn avec r=15. 99
4.7 Les résultats de la variation du Rn avec r=20. 99
4.8 Les résultats de la variation du Rn avec r=25. 99

5.1 Conteneurs vs. VMs. 112
5.2 Cas d'utilisation global du système. 118

List of Abbreviations

IoT	**I**nternet **O**f **T**hings
IoS	**I**nternet **I**f **S**ervices
CPS	**C**yber **P**hysical **S**system
VM	**V**irtual **M**achine
DMo	**D**ata in **Mo**tion
CM	**C**loud **M**anufacturing
IT	**I**nformation **T**echnology
QoS	**Q**uality **o**f **S**ervice
MEC	**M**obile **E**dge **C**omputing
MQTT	**M**essage **Q**ueuing **T**ransport **T**elemetry
FN	**F**og **N**ode
EN	**E**dge **N**ode
SE	**S**ystème **E**xploitation
AGV	**A**utomatic **G**uided **V**ehicle
PI	**P**erformance **I**ndicator
AP	**A**ccess **P**oint
LAN	**L**ocal **A**rea **N**etwork

Je dédie ce travail à :

Ma chère mère ***Fatma*** *et mon père* ***Mohamed Taher***

Mes chers jumeaux ***Jad*** *et* ***Ayet*** *ainsi qu' à tous mes proches…*

Remerciements

Il n'aurait pas été possible d'écrire cette thèse de doctorat sans l'aide et le soutien des personnes aimables et bienveillantes qui m'entourent, dont il n'est possible de mentionner ici que certaines d'entre elles.

Cette thèse n'aurait pas pu être réalisée sans l'aide, le soutien et la patience de mon encadrant de thèse, **Dr. Sahnoun M'hammed**, grâce à qui j'ai pu décrocher cette thèse. Sans parler de ses conseils et de ses connaissances inégalées de son domaine de recherche, ses bons conseils, son soutien et son amitié ont été inestimables tant sur le plan académique que personnel, ce dont je lui suis extrêmement reconnaissante. Je tiens à remercier **Dr. Baudry David**, qui a su me faire profiter de son expérience dans le domaine, bien orienter et conseiller. La direction de la thèse est partagée avec **Pr. Gout Christian**, ce fut un honneur d'être l'un de ses étudiants en doctorat. J'apprécie toutes ses contributions en temps, et pour rendre mon expérience de doctorat productive et stimulante.

Par ailleurs, j'adresse ma reconnaissance à l'ensemble des membres du jury qui ont bien voulu évaluer ce travail de recherche. Je suis reconnaissante et honorée que vous ayez accepté de faire partie du jury de ma soutenance de thèse. En particulier, je remercie **Pr. Mitton Nathalie** et **Pr. Merghem-Boulahia Leila** d'avoir pris le temps de lire et d'examiner la thèse.

Je tiens à remercier surtout ma famille : mes parents, mon mari, mes frères et ma sœur, mes jumeaux et mes beaux-parents pour leur amour et support inconditionnel. Cette thèse est en particulier dédiée à ma mère : les mots ne peuvent exprimer à quel point je suis reconnaissante pour tous les sacrifices que tu as faits, tes appels et prières étaient ma force, à mon père qui a considéré mon doctorat un rêve personnel.

Je tiens également à exprimer ma profonde et sincère gratitude au **Dr Bettayeb Belgacem**, pour sa perspicacité technique qui a rendu ce projet de recherche à la fois intéressant et réalisable. Il a consacré énormément de temps et d'énergie à cette recherche ; Au **Dr.Louis Anne**, directrice de recherche LINEACT, pour son aide et support et au **Pr. Mazari Belhacène** le directeur national de LINEACT, pour ses précieux conseils et son suivi.

Je tiens également à remercier l'ensemble du laboratoire LINEACT Rouen, pour avoir créé un environnement aussi agréable et idéal. Ce fut un honneur pour moi de faire partie de votre équipe. Je tiens aussi à remercier toute l'équipe du LMI INSA Rouen, spécialement **Mme.Diarra Brigitte** pour son implication et sa bonne gestion des démarches administratives. Aussi, un grand merci à tous les doctorants et à tous les amis que j'ai eu la chance de connaître pendant mes trois années de thèse très agréablement passées.

J'exprime ma gratitude au **Pr. Bensrhair Abdelaziz** et au **Dr. Dahane Mohamed**, membres de mon comité de suivi de thèse pour leur disponibilité et leur vision extérieure.

Résumé

Cette thèse aborde le problème de collecte et de traitement distribué de données massives dans le contexte de l'industrie du futur pour répondre aux exigences de ses applications et surmonter les limitations du "Cloud Computing" en terme de traitement en temps réel et de distribution du calcul. Une approche pour résoudre ce problème consiste à déployer des ressources informatiques de type "Fog/Edge" à proximité des Systèmes Cyber-Physiques (CPS) connectés via l'Internet des Objets (IoT) d'une manière distribuée afin de fournir des réponses/décisions en temps réel. L'objectif général est de surmonter les défis empêchant le déploiement efficace du "Fog Computing" dans les applications de l'industrie du futur. Ces défis étaient définis et par la suite classés en Entrées et Sorties et leur connexité a été étudiée. Cette connexité est nécessaire pour réduire la complexité de résolution de l'ensemble des défis. Afin de contribuer à surmonter l'un de ces défis, nous avons choisi de travailler sur une application de logistique interne dans un atelier de production flexible, vu son adaptabilité à la technologie Fog. Un modèle d'optimisation du coût de déploiement de la solution basée sur le concept de "Fog Manufacturing" a été proposé. Nous avons considéré plusieurs contraintes réalistes, telles que : la capacité de demande, la latence, le nombre maximal de liaison et la couverture souhaitée. Des contraintes liées à la présence et à la communication d'objets mobiles, tels que les AGVs (Automated Guided Vehicles) et aux choix technologiques des connexions filaires, sont également considérés. Ce modèle a été implémenté dans le solveur Fico Xpress et a permis de montrer l'intérêt des nouvelles contraintes par rapport au contexte applicatif, mais aussi par rapport aux résultats existants dans la littérature. La dernière partie de ces travaux de thèse concerne l'implémentation réelle d'une architecture basée sur le "Fog Manufacturing" dans le contexte de la plateforme industrie du futur de CESI. Ce déploiement utilise la technologie "Docker" comme technologie d'orchestration de la couche "Fog". Cette architecture admet comme fonction le filtrage des données avant le stockage permanent dans les serveurs du "Cloud".
Mots clés Industrie du futur, CPS, Fog computing, Cloud computing, Optimisation, Déploiement, Distribution

Abstract

This thesis addresses the problem of collecting and distributing massive data in the context of the industry 4.0 to meet the requirements of its applications and overcome the limitations of Cloud Computing in terms of real-time processing and distribution of computation. An approach is used to solve this problem, it is to deploy the Fog/Edge computing resources near to the set of Cyber-Physical Systems (CPS) connected via the Internet of Things (IoT) in a distributed manner, which provide real-time responses/decisions. The main objective is to overcome the challenges preventing the effective deployment of the Fog Computing in Industry 4.0 applications. These challenges were defined and subsequently classified into Inputs and Outputs and their links were studied. This links are necessary to reduce the complexity of solving the entire challenges. In order to overcome one of these challenges, we chose to work on an internal logistics application in a flexible production workshop, given its adaptability to the Fog technology. A cost optimization model for the deployment of the solution based on the concept of "Fog Manufacturing" was proposed. We considered several realistic constraints, such as : demand capacity, latency, maximum number of links and desired coverage. Constraints related to the presence and communication of mobile objects, such as AGVs (Automated Guided Vehicles), and to the technological choices of the wired connections, are also considered. This model has been implemented in the Fico Xpress solver and has allowed to show the interest of the new constraints with respect to the application context, but also with respect to the existing results in the literature. The last part of this thesis work concerns the real implementation of an architecture based on "Fog Manufacturing" in the context of the CESI Industry of the Future platform. This deployment uses the "Docker" technology as an orchestration technology of the "Fog" layer. This architecture allows the filtering of data before the permanent storage in the Cloud servers.
Keywords Industry 4.0, CPS, Fog computing, Cloud computing, Optimization, Deployment, Distribution

Chapitre 1

Introduction générale

Cette thèse aborde le problème du déploiement du Fog computing dans le contexte de l'Industrie 4.0, par l'exploitation localement du concept de traitement distribué des données des systèmes cyber-physique (CPS). Ce déploiement représente une évolution récente de la manière de traiter des données et est une instanciation qui se rapproche du modèle de calcul centralisé du Cloud computing.

L'industrie est la partie d'une économie qui produit des biens matériels par la transformation de matières premières via des processus (hautement) mécanisés et automatisés. Depuis le début de l'industrialisation, les sauts technologiques ont conduit à des changements radicaux qui sont aujourd'hui appelés des "révolutions industrielles" : dans le domaine de la mécanisation (1ère révolution, XVIIIème siècle), dans ceux de l'utilisation intensive du pétrole, du gaz et de l'énergie électrique (2ème révolution, fin du XIXème siècle), la 3ème révolution industrielle est elle liée à l'émergence de l'informatique et de l'énergie nucléaire (années 1970) et enfin la 4ème révolution industrielle (XXIème siècle), liée, par exemple, à l'interconnexion des appareils mobiles connectés ou à la digitalisation des processus. L'intégration des CPS dans le processus de production est apparu suite à l'introduction de la quatrième révolution industrielle depuis 2010. Sur la base d'une numérisation avancée dans les usines, la combinaison des technologies d'internet et des technologies du futur dans le domaine des objets "intelligents" (machines et produits) semble entraîner un nouveau changement de paradigme fondamental dans la production industrielle. La vision de l'Industrie 4.0 contient des systèmes de fabrication modulaires, communiquant et efficaces nommés CPS. Cela est censé permettre la fabrication de produits individuels en un seul lot tout en maintenant les conditions économiques de la production de masse.

Les technologies de l'information basée sur l'IoT (Internet of Things), avec leur forte pénétration, peuvent fournir des solutions efficaces pour satisfaire les exigences de l'Industrie 4.0 et améliorer la compétitivité de l'industrie. Dans ce contexte, l'industrie 4.0 repose sur de nombreuses technologies (IoT, IoS, Big Data, IA, RA/RV, etc,.) parmi lesquelles on trouve le Cloud Manufacturing qui est un nouveau paradigme commercial orienté fabrication, basé sur le concept et les services du Cloud Computing.

L'IoT est l'un des éléments clés de l'Industrie 4.0. Les dispositifs IoT industriels génèrent une énorme quantité de données par unité de temps. Ces données doivent être traitées en temps réel par diverses applications orientées Industrie 4.0 afin que différents services puissent être réalisés. Par exemple, nous pouvons citer les applications de traitement d'image et de navigation liées aux systèmes d'assistance robotique dans les industries. Si ces applications ne parviennent pas à fournir leurs services en temps voulu, les performances des robots industriels se dégraderont considérablement.

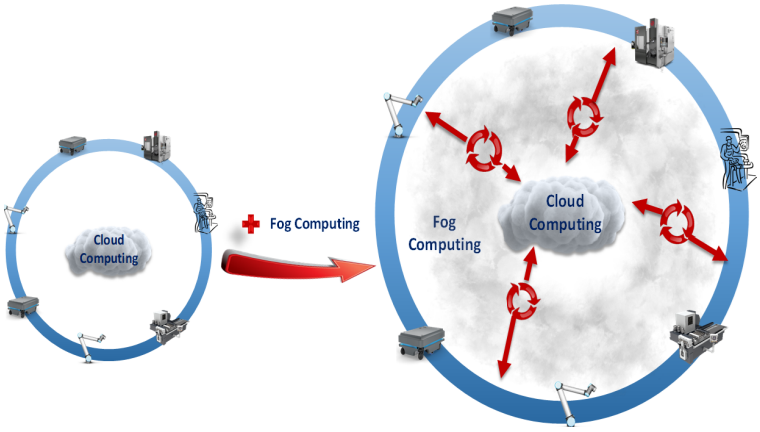


FIGURE 1.1 – Le Fog et le Cloud computing.

Aujourd'hui, un important nombre de services dépend des infrastructures du Cloud computing. Les serveurs de calcul du Cloud computing fournissent des algorithmes de calcul profonds et puissants qui satisfont les différents besoins des clients et ils proposent une capacité de stockage considérable. Étant donné que les centres de données du Cloud se trouvent à plusieurs sauts de distance des dispositifs IoT, le traitement des données industrielles en temps réel dans le Cloud est délicat, voire impossible. Cela augmente le délai de la transmission des données, l'encombrement du réseau et le temps de fourniture des services. Malgré ces puissances qui couvrent la plupart des demandes des utilisateurs, les serveurs du Cloud computing ne sont plus capables de satisfaire la nouvelle génération des contraintes des applications. Les capteurs des systèmes connectés ne possèdent pas les ressources internes pour traiter leurs données et prendre les décisions rapidement. Cette contrainte nécessite l'intégration d'une nouvelle solution afin d'avoir un temps de réponse le plus réduit possible.

Dans cette direction, Cisco a proposé en 2012 le paradigme de Fog computing consistant à déployer un grand nombre de serveurs de calcul de données, distribués sur l'ensemble du site, exploitant les ressources des bords de réseaux et placés au plus près des utilisateurs (BONOMI et al., 2012), comme le montre la figure 1.1.

Dans les industries basées sur le Fog computing, les machines connectées à des dispositifs informatiques tels que les points d'accès, les routeurs et les micro serveurs agissent comme des nœuds de Fog. Ces nœuds offrent une infrastructure comme les centres de données du Cloud pour aider à l'exécution de différentes applications de l'Industrie 4.0. Cependant, les nœuds Fog sont déployés d'une manière distribuée, et ils sont hétérogènes en termes de vitesse de traitement et de normes de réseau. En outre, les caractéristiques des dispositifs IoT, telles que la fréquence de remontée des données et la taille des données, diffèrent d'un dispositif à l'autre. Ces caractéristiques jouent un rôle essentiel dans la définition des caractéristiques des applications. Par exemple, la charge de transmission et l'intensité du calcul ont une relation proportionnelle à la taille des données d'entrée. De même, la charge du réseau change en fonction de la fréquence à laquelle les dispositifs IoT associés envoient des données à cette application.

Le Fog computing a été déployé dans de nombreuses applications et dans divers domaines ainsi que dans des applications liées à l'Industrie 4.0 comme la traçabilité des produits, la maintenance prédictive, le monitoring, la logistique, etc. Il a montré un potentiel prometteur pour répondre aux exigences de l'Industrie 4.0. Dans cette thèse, nous nous intéressons à l'implémentation du Fog Manufacturing dans une

application de logistique, dans le contexte de l'Industrie 4.0. De manière générale, il est souvent souligné que le Fog computing n'est pas suffisamment déployé dans les applications industrielles. Dans la société actuelle, numérisée et centrée sur le client, le secteur logistique est soumis à une pression constante pour rester compétitif, pour fournir un service client de qualité et pour trouver des moyens novateurs d'optimiser ses opérations et processus. Dans ce travail, nous souhaitons intégrer le Fog computing afin de surmonter certains défis comme le coût du déploiement et, ainsi, d'accroître les performances des applications logistiques.

La première contribution scientifique de cette thèse consiste à étudier les contraintes qui empêche l'implémentation réelle du Fog computing dans le domaine industriel. Pour mieux comprendre les contraintes s'appliquant sur une telle implémentation, il faut analyser les différents défis limitant, à savoir : la programmabilité, la sécurité, l'hétérogénéité, l'interopérabilité, la qualité de service (QoS), la consommation d'énergie et le coût. Dans une première étape il est important de représenter les défis en fonction de leurs indicateurs de performances afin, dans une deuxième étape, d'identifier les liens entre les différents défis.

Afin de surmonter les défis, en particulier pour optimiser le coût du déploiement des noeuds constituant l'architecture basée sur le Fog, il faut penser à définir un modèle de minimisation du coût en se basant sur des contraintes liées aux conditions de l'application de logistique. Ce modèle et les contraintes associées représentent une seconde contribution scientifique.

Cette thèse a enfin mené à l'utilisation des résultats d'optimisation du déploiement du Fog manufacturing dans une implémentation réelle sur la plateforme Industrie du Futur du laboratoire LINEACT CESI. Le but de cette implémentation est principalement de prouver l'applicabilité du Fog, de tester l'acheminement et l'orchestration des données et d'appliquer quelques traitements de données localement au niveau de la couche Fog.

Le reste de ce document est organisé en cinq chapitres suivis d'une conclusion générale, qui sont décrits brièvement ici :

- Dans le chapitre 2, nous présentons le contexte de l'Industrie 4.0, tout en détaillant la problématique des limitations du Cloud computing. Ensuite nous rappelons les paradigmes qui composent cette révolution industrielle, ainsi que les exigences à satisfaire et les objectifs à optimiser. Nous présentons également une étude bibliographique concernant les solutions de l'Edge computing ainsi que du Fog computing (définition, caractéristique, architecture et applications).

- Dans le chapitre 3, nous présentons l'intégration du Fog computing dans l'Industrie 4.0 et les défis qui limitent cette adoption. Ces défis sont classés sous deux catégories (Input et Output). Nous présentons les indicateurs de performances (PIs) de chacun afin d'identifier les liens entre les différents défis.

- Dans le chapitre 4, nous nous attaquons à la problématique de l'optimisation du déploiement du Fog Manufacturing et nous présentons une modélisation du déploiement des noeuds d'une architecture de logistique basée sur le Fog computing. Suite à cette modélisation, nous contribuons à sa résolution en considérant certaines contraintes.

- Dans le chapitre 5, notre présentons une application réelle du Fog Manufacturing en utilisant la technologie de conteneurisation Docker et les micro ordinateurs Raspberry Pi. Les données sont acheminées tout le long de l'architecture via différents types de connexion. Le filtrage de données ainsi que le traitement en temps réel des données sera possible par la couche Fog.

Le manuscrit de thèse se termine par une conclusion générale dans laquelle nous synthétisons les apports de nos travaux et nous en dégageons des perspectives potentielles.

Chapitre 2

Contexte & État de l'art

2.1 Introduction

Le contexte économique concurrentiel et le comportement imprévisible du marché mettent les entreprises industrielles dans la quasi obligation de mettre en œuvre des technologies plus innovantes et efficaces dans leurs infrastructures afin d'améliorer leurs systèmes de production. Ce type de transformation est impacté par des défis mondiaux tels que la transition énergétique et la raréfaction des ressources, les changements démographiques,...(KAGERMANN et al., 2013).

Précédée par trois révolutions majeures et basée sur le progrès technologique rapide, une quatrième révolution industrielle est en train de voir le jour contribuant à résoudre le phénomène de complexité croissante des systèmes et notamment la complexité des données des systèmes (ZANDER et MÄHÖNEN, 2013). Les gouvernements et les entreprises manifestent un grand intérêt pour la quatrième révolution industrielle, sous différentes appellations telles que l'industrie du futur, l'Industrie 4.0 ou encore l'Industrial Internet.... Son émergence renforce la position concurrentielle des pays développés, en intégrant le système cyber physique "*Cyber Physical System en anglais"* (CPS) dans les systèmes de production, la logistique et d'autres services liés à la production (LEE, BAGHERI et KAO, 2015).

En conséquence, les systèmes d'information industriels intègrent de nombreuses technologies numériques émergentes, comme le paradigme de l'Internet des objets *"Internet of Things en anglais"* (IoT), considéré comme l'une des technologies influentes (INTELLIGENCE, 2008) et un facteur important pour le progrès des processus industriels (GAZIS et al., 2015).

Cette intégration est un facteur majeur de la complexité croissante en termes de méthodologies adoptées, de données générées et d'opérations exécutées.

En outre, la demande croissante d'automatiser les tâches pénibles faites quotidiennement par l'humain et d'intégrer les technologies connectées augmente l'usage de l'IoT et également le nombre des objets connectés à l'IoT. Cela augmente ainsi le nombre de machines connectées, de capteurs et la quantité de données collectées et échangées par ces différents objets. Par exemple, depuis 2012, 20 communes résidentielles ont généré plus de trafic Internet que le trafic mondial échangé en 2008 (CHIANG et ZHANG, 2016). En 2017, Cisco prévoyait que 50 milliards d'appareils seront connectés à l'Internet d'ici 2020 (MARKAKIS et al., 2017). Ce nombre devrait atteindre 500 milliards d'ici 2025 (EVANS, 2011). Effectivement en 2020, ce nombre a augmenté mais pas comme prévu en 2017 pour atteindre 26 milliards d'objets connectés. Cet écart entre les anticipations de 2017 et la réalité du 2020 est principalement dû à l'arrivée décalée de la 5G. Par conséquent, cet énorme nombre d'objets connectés génère un nouveau phénomène problématique pour les centres de données appelé "Data Tsunami" ou Tsunami des données, pouvant diminuer la qualité de service *"Quality od Service en anglais"*(QoS) et augmenter la crise de stockage des

données (ZANDER et MÄHÖNEN, 2013). Évidemment, cela affectera la performance industrielle si les solutions adéquates ne sont pas conçues et mises en œuvre.

Le Cloud computing est considéré comme un moyen efficace et capable de traiter les données grâce à ses performances en termes de calcul, de traitement et de stockage. Mais face à la croissance de la quantité et à la complexité des données, le Cloud computing a montré ses limites notamment pour les applications industrielles en temps réel ou quasi réel (ISLAM, MORSHED et GOSWAMI, 2013). Cela affecte directement la qualité du service fourni par ces centres de données centralisés. En conséquence, le délai de la transmission et la QoS vont connaître une baisse importante, face à l'énorme quantité de données industrielles hétérogènes.

Afin de surmonter ce problème, l'entreprise Cisco a introduit un nouveau paradigme, nommé le Fog computing, qui s'inspire du principe de l'Edge computing (AI, PENG et ZHANG, 2018). Il est ainsi proposé afin de répondre à la problématique de traitement des données provenant des sources hétérogènes distribuées en temps réel. Cette solution située proche des utilisateurs finaux est présentée aussi comme l'extension du Cloud computing . Ce modèle de calcul distribué se caractérise par ses capacités de calcul et d'analyse et ses réponses en temps réel, qui peuvent satisfaire les exigences de plusieurs domaines notamment, l'Industrie 4.0.

Grâce à ses caractéristiques (distribution, support de la mobilité et de l'hétérogénéité, sécurité et faible latence), le Fog computing pourrait faire disparaître les limitations du Cloud computing dans le contexte de l'Industrie 4.0 ou au moins renforcer son rôle. Le potentiel prometteur du Fog computing est avéré dans divers domaines d'applications. L'utilisation du Fog computing dans le contexte industriel pourrait notamment améliorer les performances de plusieurs applications telles que la maintenance, l'inspection, la logistique (MUKHERJEE et al., 2017a).

Afin d'étudier les différents concepts évoqués auparavant tel que le Fog Computing et l'Industrie 4.0 il est nécessaire de faire un état de l'art afin de comprendre leurs caractéristiques et exigences. Nous avons visé principalement les travaux publiés dans des articles de conférences et de journaux. Les données publiées dans les bases de données de Google-scholar et des organismes de recherche de 2010 à 2020 concernant les deux paradigmes "Industrie 4.0" et "Fog Computing" ont été étudiées dans ce manuscrit. Les principales sources publiant des travaux liés à l'Industrie 4.0 et au Fog computing apparaissaient dans IEEE transaction (63), Science-direct (23), ACM(7), Springer (4)... Cette étude s'est concentrée sur de nombreux concepts clés tels que : L'IoT, le Fog computing (caractéristiques, applications), le CPS, le Smart manufacturing, L'Industrie 4.0 (en prenant en considération plusieurs appellations), le Cloud computing, le temps réel, la distribution, etc.

Cette analyse a confirmé que le Fog computing et l'Industrie 4.0 restent des axes de recherche récents et que de nombreux travaux en cours se concentrent sur la manière d'exploiter les technologies basées sur l'IoT dans la quatrième révolution industrielle.

La structure de ce premier chapitre est la suivante : La section suivante (Section 2.2), présente un aperçu historique des révolutions industrielles, une analyse de leur lien avec la complexité des données et une étude de l'Industrie 4.0 (définitions, composantes, travaux sur l'Industrie 4.0 et exigences). La section 2.3 présente le contexte et les limitations du Cloud computing dans le contexte industriel ainsi que le positionnement technique et scientifique de la thèse.

La section 2.4, donne un aperçu détaillé du Fog computing, contenant ses définitions, ses caractéristiques et son lien avec le Cloud computing. Cette section donne également quelques applications générales du Fog computing et des architectures associées. Le chapitre se termine par une conclusion 2.5 sur les orientations futures

de la recherche pour une mise en œuvre du Fog computing dans le domaine industriel et mentionne les défis les plus importants auxquels cette mise en œuvre est confrontée.

2.2 L'Industrie 4.0 : État de l'art

Dans cette première section, nous allons tout d'abord parler de l'historique des révolutions industrielles ainsi que les liens entre ces évolutions et la place et le rôle des données provenant des objets connectés dans un atelier de production. Par la suite nous définissons la quatrième révolution industrielle et nous présentons ses composants structurants et ses exigences.

2.2.1 L'historique des révolutions industrielles

Les systèmes de productions industrielles ont connu quatre révolutions majeures depuis le début du siècle dernier (BILL, 2016). Chaque nouvelle révolution vient satisfaire les exigences spécifiques du marché en utilisant les technologies existantes. Comme le montre la figure 2.1, la première révolution s'appelle la révolution mécanique (1784) où des machines mécaniques sont introduites pour accélérer la production et proposer de nouveaux types de produits qui ne peuvent être fabriqués manuellement. Ces systèmes de production atteignent leurs limites avec l'augmentation de la demande de produits. Dans ce contexte, la deuxième révolution, appelée révolution électrique (1870), était nécessaire pour satisfaire les exigences en termes de quantité de produits et de rapidité d'exécution.

La croissance continue de la demande et la répétition des opérations pénibles conduisent à une nouvelle révolution appelée la révolution électronique (1969) (WAN, CAI et ZHOU, 2015)(SHAFIQ et al., 2015). Cette génération de systèmes de production utilise des dispositifs électroniques pour exécuter des opérations complexes et des systèmes robotisés pour exécuter des opérations complexes. Les deux dernières décennies ont connu des changements radicaux dans le comportement des consommateurs, où la demande devient imprévisible et la durée de vie des produits devient de plus en plus courte (HERMANN, PENTEK et OTTO, 2016).

Les systèmes de production deviennent incapables de suivre les changements du marché, notamment face à de nombreuses exigences telles que des délais de développement courts, la personnalisation de la demande, la flexibilité, la décentralisation et l'efficacité d'allocation des ressources aux différents départements de production (allocation optimale). Par conséquent, le développement d'une nouvelle génération de systèmes de production capables de répondre à ces nouvelles exigences est une nécessité pour l'industrie. Aujourd'hui, la quatrième révolution industrielle est introduite afin de donner plus d'intelligence aux outils de production, plus de réactivité et une meilleure maîtrise des systèmes de production FERREIRO et al., 2016. Ces révolutions industrielles sont présentées sur la figure 2.1.

2.2.2 La relation entre les données et les révolutions industrielles

Nous vivons à une époque caractérisée par la dynamique des changements technologiques et organisationnels qui influence d'une façon directe le marché. Les clients attendent des produits personnalisés dans des délais de plus en plus courts. Les révolutions conséquentes sont devenues des "révolutions numériques" (IVANOV, TSIPOULANIDIS et SCHÖNBERGER, 2019). Cette digitalisation concerne dans notre contexte ceux des outils et des systèmes de production.

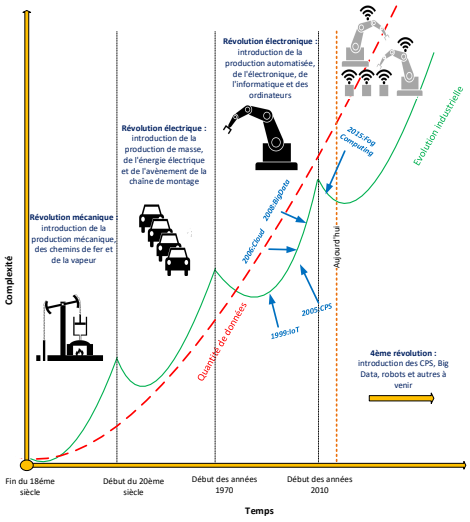


FIGURE 2.1 – La succession des quatre révolutions industrielles.

Les systèmes de production sont plus connectés et plus intelligents grâce aux nouvelles technologies de capteurs, à l'automatisation intelligente et à l'analyse des données, de sorte que des produits personnalisés peuvent également être créés d'une manière flexible. Par exemple, la maintenance des systèmes de production est optimisée grâce à l'analyse prédictive et à la surveillance à distance, à la réalité augmentée pour apporter des informations contextualisées aux opérateurs ou encore l'utilisation de la fabrication additive pour remplacer des pièces endommagées. Dans le contexte de la plus récente révolution industrielle, le pilotage de la production se déplace dans le Cloud et permet un service à la demande par les acteurs de la chaîne de valeur.

Les données circulant vers et depuis les systèmes permettent d'équilibrer l'utilisation des activités et des ressources tout au long de la chaîne de valeur de production, en réduisant les délais et les coûts notamment grâce à une exploitation optimisée des moyens de production. En d'autres termes, grâce à l'introduction de l'Industrie 4.0, un tout nouveau niveau de "Lean manufacturing" peut être atteint (PORTER et HEPPELMANN, 2015), car les données qui sont désormais disponibles permettent d'augmenter la flexibilité, l'intelligence et la disponibilité des ressources et permettent de réduire la complexité de la gestion des systèmes de production.

On observe sur la figure 2.1 que chaque révolution industrielle doit permettre de réduire la complexité de gestion des systèmes de production, mais l'évolution industrielle génère de nouveaux défis qui causent l'augmentation des demandes de personnalisations et effectivement des coûts de la production (SCHUH et al., 2014b) et la quantité de données à gérer. Ainsi, ces systèmes de production arrivent à leurs limites et nécessitent une nouvelle "révolution". Comme illustré sur la figure 2.1, durant chaque période d'une révolution industrielle, le système de production passe par deux phases. Au cours de la première phase, de nouvelles technologies sont introduites et une réduction de la complexité est observée. C'est la phase qui connaît une bonne exploitation des technologies déjà introduites. Les entreprises commencent à avoir une maturité dans l'adoption des technologies existantes et à développer de nouvelle technologies pour rendre ces systèmes encore plus efficaces.

C'est le cas par exemple de l'introduction de la robotique dans la 3ème révolution industrielle qui a connu l'introduction de l'électronique et des automates à ses débuts mais a vu ensuite l'utilisation des robots et de la vision par ordinateur pour assurer plus d'efficacité à ces systèmes.

Ces systèmes de production évoluent et permettent de proposer de nouveaux produits toujours plus personnalisables qui font évoluer avec eux les marchés et les exigences des clients finaux ou intermédiaires. Les technologies introduites lors d'une révolution industrielle arrivent à leur utilisation optimale quelque temps après leurs premières utilisations comme on peut le voir sur la figure 2.1. A partir de ce moment la complexité des système de production qui ne cesse d'augmenter dépasse les capacités des technologies utilisées, jusqu'à l'arrivée d'une nouvelle révolution industrielle qui essaye de répondre de nouveau aux exigences du marché. En effet, la quantité de données générées par les systèmes de production, depuis la première révolution, suit la courbe rouge pointillée qui ne cesse d'augmenter. Ce phénomène montre que l'un des problèmes le plus important à l'avenir sera celui des données. La figure 2.1 illustre aussi ces différentes révolutions en fonction de la complexité et de la quantité des données.

2.2.3 Les différentes appellations de l'Industrie 4.0

La figure 2.2 indique le nombre de publications scientifiques parues depuis l'apparition du terme Industry 4.0 (2011). Nous nous sommes basés sur une requête "GoogleScholar" comprenant les termes : Industrie du futur, Industry 4.0, Industrie 4.0, Smart Factory, Industrial Internet, etc,.. Jusqu'en 2019, le terme allemand **1.Industrie 4.0** est le plus adopté, ce qui peut s'expliquer par le fait que le gouvernement allemand est le premier à avoir lancer l'initiative d'adopter la quatrième révolution industrielle.

Depuis 2017, de nombreux chercheurs d'autres pays commencent à travailler sur ce domaine (voir par exemple CHIARELLO et al., 2018) et attribuent l'appellation : **2.Industry 4.0 ou Industrie 4.0**.

Malgré ces appellations dominantes, la figure 2.2 montre plusieurs noms qui sont utilisés pour décrire cette révolution comme : **3. Industrie du futur** en France, qui existe depuis 1989 mais qui fait référence à d'autres thèmes de recherche. Les autres appellations sont moins utilisées. **4. Industrial Internet** aux États-Unis, **5. Internet+** en Chine et en Corée (RÖPKE et al., 2016). Ce concept est également appelé "Smart Factory" ou "Integrated Industry" (HENG, 2014). Dans la suite de ce manuscrit, le terme "Industrie 4.0" sera utilisé pour désigner la quatrième révolution industrielle.

2.2.4 Définition

La littérature contient plusieurs définitions de l'Industrie 4.0 qui diffèrent d'une institution à une autre et d'un auteur à un autre. Dans ce paragraphe nous allons en présenter quelques-unes, qui nous semblent les plus pertinentes. D'après la NIST (National Institute of Standards and Technology), la définition de l'Industrie 4.0 est la suivante : « les systèmes de fabrication intégrés et collaboratifs qui répondent en temps réel aux exigences et aux conditions changeantes de l'usine, du réseau d'approvisionnement et des besoins des clients ». Cela signifie que ce paradigme peut réagir en temps réel à une situation de fabrication compliquée et diversifiée (NIST., 2020). Ce terme très récent, qui se répète souvent dans les articles parcourus, se réfère à la quatrième révolution industrielle, est souvent considéré comme l'application

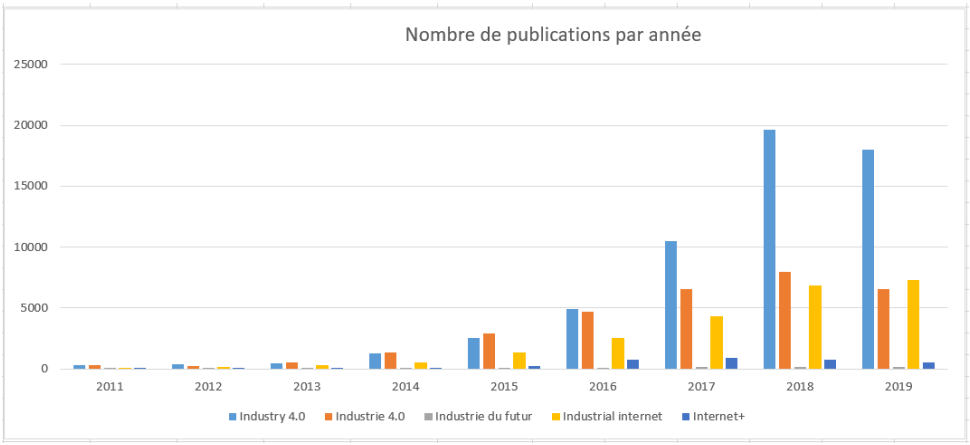


FIGURE 2.2 – Nombre de publications concernant la thématique Industrie 4.0 selon l'appellation.

du concept générique des systèmes cyber-physiques (CPS) aux systèmes de production industrielle (systèmes de production cyber-physique) (DRATH et HORCH, 2014). L'Industrie 4.0 peut aussi décrire un CPS orienté vers la production, qui intègre les installations de production, la logistique et même les processus de création de valeur globale (WANG et al., 2016). Cette quatrième révolution, est également appelée « usine intelligente », où l'on trouve les concepts de CPS et d'IoT, qui convergent vers l'Internet des services qui utilisent largement l'approche basée sur le Cloud computing pour créer, publier et partager les services (PISCHING et al., 2015). L'Industrie 4.0 se concentre sur la création de produits, procédures et processus intelligents, avec comme conséquence que les usines intelligentes deviennent capables de gérer la complexité et de fabriquer des produits plus efficacement (KAGERMANN et al., 2013). Pour d'autre chercheurs (par exemple HERMANN, PENTEK et OTTO, 2016, ou encore BOYES et al., 2018), l'Industrie 4.0 est la combinaison des technologies et des concepts, qui surveillent les processus physiques, créent une copie virtuelle du monde physique, et prennent des décisions décentralisées. En se basant sur l'IoT, les CPS communiquent et coopèrent les uns avec les autres et avec les humains en temps réel. Grâce à l'IoS (Internet of Service), des services internes et inter organisations sont proposés et utilisés par les acteurs de la chaîne de valeur. L'architecture de l'Industrie 4.0 proposée par la vision allemande se repose principalement sur l'IoT, l'IoS, les CPS, le Big data, le Cloud computing et les capteurs (KANG et al., 2016). Les États Unis ont adopté le même concept, en rajoutant le sustainable manufacturing (énergie intelligente et l'intégration des systèmes) et la standardisation aux composantes européennes. Par contre, en Corée du Sud, les politiques étaient convaincus par la nécessité d'intégrer les hologrammes et la réalité augmenté dans la notion d'Industrie 4.0 (ibid.).

Cette quatrième révolution industrielle, est souvent considérée comme l'application du concept générique de CPS aux systèmes de production (SCHUH et al., 2014a). D'après la littérature, nous pouvons constater que toutes les définitions partagent plusieurs aspects. Nous les prenons en considération et proposons la définition suivante : " *L'Industrie 4.0 est un concept émergeant, qui regroupe différentes technologies existantes mais insuffisamment utilisées comme les CPS, l'IoT, l'IoS, le Cloud computing et les RMS (Reconfigurable Manufacturing System) pour rendre les industries de plus en plus intelligentes, efficaces, sécurisées, distribuées et fonctionnant en temps réel. Par conséquent cela va conduire les industries à un autre niveau d'optimisation en terme de coûts, délai et qualité.*".

2.2.5 Composants

D'après la section précédente chaque vision a adopté des composants différents de l'Industrie 4.0 et même si un grand ensemble est commun entre les trois, quelques uns restent spécifiques. Le tableau 2.1 montre qu'il y a des composants communs aux visions des différents pays.

Il y a une dizaine d'années, l'Industrie 4.0 a été inaugurée par l'introduction de l'IoT et l'IoS, les capteurs, le Cloud computing et les CPS dans l'environnement industriel. L'Industrie 4.0 se concentre sur la création de processus intelligents, flexibles et dynamiques, de chaînes de valeur optimisées en temps réel et auto-organisées. (ZHONG et al., 2017).

Les sous sections suivantes définissent ces six composants communs et présentent leur rôle dans le contexte de l'Industrie 4.0.

TABLE 2.1 – Composants de l'Industrie 4.0 proposés par les principaux pays économiques développés.

Composants	Europe	État Unis	Asie
IoT	✓	✓	✓
IoS	✓	✓	✓
CPS	✓	✓	✓
Big Data	✓	✓	✓
Capteurs	✓	✓	✓
Cloud computing	✓	✓	✓
Énergie intelligente	-	✓	✓
Standardisation	-	✓	-
Intégration du système	-	✓	-
Fabrication additive ou impression 3D	-	✓	✓
Hologramme / Réalité augmentée	-	-	✓

Internet of Things (IoT)

Pour les auteurs de (TRAPPEY et al., 2016), (AL-FUQAHA et al., 2015) et (MATTERN et FLOERKEMEIER, 2010), l'IoT est un concept utilisé pour la première fois par Ashton en 1999 (ASHTON, 2011), qui permet aux objets physiques de "voir, entendre, penser et accomplir des tâches en les faisant parler ensemble, partager des informations, accomplir des tâches avec plus de précision et coordonner leurs décisions". L'IoT permet de rendre les "objets" communiquant et faisant partie de l'environnement d'Internet.

Pour CERP-IoT (SUNDMAEKER et al., 2010), l'IoT est défini comme une infrastructure dynamique avec des capacités d'auto-configuration basées sur des protocoles de communication standard et interopérables. Selon (ATZORI, IERA et MORABITO, 2010) (), l'IoT est considéré comme l'infrastructure globale du réseau, reliant les objets physiques et virtuels afin d'exploiter les données collectées, les capacités de communication et l'accessibilité à tout moment, partout et sous toute forme (VIJAYKUMAR, SARAVANAKUMAR et BALAMURUGAN, 2016). De nombreuses définitions ont été proposées par d'autres recherches telles que (GUBBI et al., 2013), (PERERA et al., 2014), qui offrent une vision claire des applications basées sur l'IoT.

Grâce à l'IoT, les objets peuvent avoir des identités virtuelles, des états et des caractéristiques et être intégrés dans le réseau. Cela permet aux objets d'exécuter les tâches suivantes :

— Collecter et échanger des données de capteurs intelligents,
— Détecter et contrôler les objets,
— Connecter le monde physique et les systèmes informatiques,
— Fournir les données aux CPS et au Cloud computing,
— Analyser et représenter les données.

L'intégration de l'IoT dans le domaine manufacturier a joué un rôle majeur pour mener à la quatrième ère de l'industrialisation (UNGUREAN, GAITAN et GAITAN, 2014). Dans ce contexte, toutes les entités impliquées dans les processus de fabrication pourraient être reconnues par un identifiant unique qui sert à distinguer les différentes entités connectées (WU, MENG et GRAY, 2017). Le matériel et les logiciels peuvent fonctionner avec moins d'intervention humaine, pour accomplir toutes les tâches de la chaîne de fabrication, comme la production, l'inspection (SHANOUN et al., 2011) et la surveillance (BETTAYEB, BASSETTO et SAHNOUN, 2014), (PISCHING et al., 2015).

L'adoption de l'IoT assure la connexion entre les machines. Il s'agit de créer un réseau intelligent reliant tous les acteurs de la chaîne de valeur, qui offre un contrôle autonome avec le moins d'interventions possibles (MANAVALAN et JAYAKRISHNA, 2019). Étant la base de l'Industrie 4.0, l'IoT permet les échanges entre les usines intelligentes et les utilisateurs afin de fournir des données sur les besoins des clients et la personnalisation des produits aux usines et fournir des produits et services intelligents avec la réduction des coûts (SHROUF, ORDIERES et MIRAGLIOTTA, 2014). Selon (LOPEZ, 2014), l'IoT dans un contexte industriel intelligent peut former un réseau d'échange entre les différents composants d'un atelier de production, comme :

— Les personnes ayant l'information la plus adéquate concernant les processus de fabrication,
— Les processus automatisés, en rendant le flux d'information plus rapide, ce qui stimule la prise de décision et améliore la réponse aux changements du marché,
— Les données, qui seront intégrées dans la chaîne de valeur manufacturière (nouvelles formes et nouveaux types de données).

Cyber physical system (CPS)

Le terme CPS est parmi les 50 termes à la mode rapportés par Wikipedia (WIKIPEDIA, 2020). L'intégration de ces processus physiques et informatiques assure la fusion des mondes physiques et virtuels. Depuis la dernière décennie (vers 2006), nous avons commencé à parler de ce terme (COLOMBO et al., 2017, BANERJEE et al., 2012). La définition suivante est donnée par (LEE, 2006) : "Les Systèmes Cyber Physiques (CPS) sont l'intégration des calculs avec les processus physiques. Les ordinateurs et les réseaux intégrés surveillent et contrôlent les processus physiques, généralement avec des boucles de rétroaction où les processus physiques effectuent les calculs et vice versa". Ils peuvent aussi être définis comme : Les objets physiques avec des logiciels intégrés et des capacités de calcul (ALMADA-LOBO, 2016).

Lee et al (LEE, AZAMFAR et SINGH, 2019) présentent une architecture CPS nommée 5C et composée des niveaux suivants dans l'ordre croissant comme le montre la figure 2.3 :

— Un niveau de connexion intelligent, le CPS utilise un réseau Plug & Play pour transmettre les données envoyées par un réseau de capteurs ;

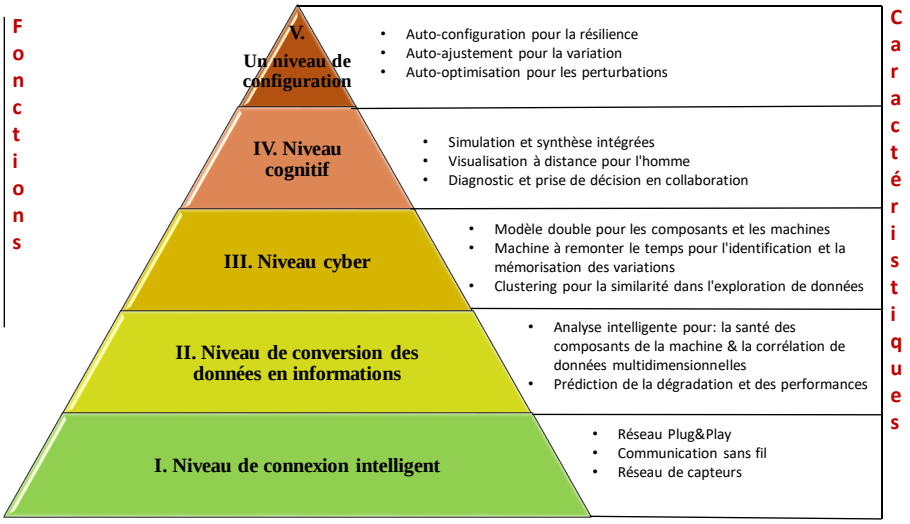


FIGURE 2.3 – L'Architecture 5C d'implémentation du CPS (LEE, AZAMFAR et SINGH, 2019).

— Un niveau de conversion de données en informations, le CPS traite les données et les transforme en informations de plus haut niveau ;
— Un niveau cyber, le CPS a une connaissance des autres CPS appartenant à son environnement et peut interagir avec eux pour enrichir son propre traitement d'information ;
— Un niveau cognitif, le CPS est capable de faire un diagnostic basé sur des simulations de son propre comportement et une analyse différentielle des données parvenant des capteurs ;
— Un niveau de configuration, le CPS peut s'adapter d'une façon autonome en cas de défaillance, se reconfigurer ou ajuster de manière autonome ses paramètres afin de reprendre son comportement initial.

Cette architecture est adoptée pour fournir une ligne directrice pour l'utilisation et la surveillance du CPS pour les applications industrielles (LEE et al., 2015).

Il est évident de noter la corrélation entre l'IoT et le CPS. L'IoT est l'objet de connexion de l'outil. D'autre part, le CPS représente un ensemble d'objets interconnectés à travers l'IoT. Le concept du CPS se concentre sur la façon de contrôler un système physique basé sur son espace numérique ou cyber-espace et l'IoT se concentre sur la façon de les connecter. Ces deux concepts sont donc complémentaires, surtout dans le domaine industriel où l'IoT est adopté pour fournir le lien à l'intérieur du CPS. Cette relation est appelée Industrial IoT (IIoT) (SADEGHI, WACHSMANN et WAIDNER, 2015).

Les domaines d'application des CPS sont nombreux (p. ex. l'aéronautique, l'automobile, l'infrastructure des ouvrages civiles, l'énergie, le transport ou encore la production de produits manufacturés). L'utilisation croissante de capteurs et de machines communicantes génère par conséquent une grande quantité de données. Dans ce cas, le CPS gère la transmission des données et l'interconnectivité via l'IoT des machines et exploite leur puissance de calcul. Cela permettra d'atteindre le niveau requis d'outils de production intelligents, résilients et auto-configurables (YING et SZTIPANOVITS, 2013).

Pour une application industrielle, le CPS a la capacité d'inclure un réseau de capteurs intégrés pour communiquer et contrôler (TAO et al., 2019) à distance le

processus de production. Grâce à ces technologies, le CPS peut produire des comportements et des décisions intelligents en utilisant ses capacités de calcul et de communication RAJKUMAR et al., 2010.

Les travaux de recherche sur les CPS dans le domaine de la production n'en sont qu'à leurs débuts. A notre connaissance, aucune étude n'en est en effet à l'étape de la mise en œuvre complète sur un système industriel, la majorité se concentre sur les étapes de modélisation et de conception ou de preuve de concept (KANG et al., 2016). Le CPS devrait être utilisé dans les ateliers pour intégrer le processus de fabrication, permettre la communication machine-to-machine et virtualiser les processus physiques des systèmes de fabrication (PISCHING et al., 2015).

Les CPS ont des exigences de sécurité croissantes en raison de cette double nature (physique et cyber/virtuel), cet accès possible génèrant des risques de sécurité (SHAFI, 2012) comme il y a un changement des réseaux industriels isolés et privés vers les réseaux ouverts et communs. Plusieurs chercheurs travaillent sur cette problématique et tentent d'améliorer les protocoles de sécurité pour le CPS (LENZINI, MAUW et OUCHANI, 2016, DIBAJI et al., 2019).

L'adoption du CPS dans les industries d'aujourd'hui (Industrie 3.0) peut être un facteur principal de la migration vers une Industrie 4.0 (SCHUH et al., 2014b). L'Industrie 4.0 exige plus de réactivité, de capacités d'autogestion et d'efficacité pour les systèmes de fabrication. Ceci peut être satisfait par de nouvelles plate-formes collaboratives basées sur des systèmes décentralisés, tels que le CPS, pour les usines intelligentes et les produits intelligents (JAZDI, 2014).

Jusqu'à présent, les architectures distribuées étaient des concepts purement théoriques et leur réalisation était presque impossible en raison du manque de technologies répondant à leurs exigences. Grâce au CPS, il est possible d'intégrer l'intelligence dans les entités IoT permettant la mise en œuvre d'architectures distribuées dans les systèmes de fabrication. Ceci introduit le concept de systèmes de production cyber physique (CPPS), qui peut détecter son état et prendre des décisions de manière autonome (WARIS, SANIN et SZCZERBICKI, 2018).

Internet of Services (IoS)

Sachant que les services sont omniprésents, il est très important pour l'Industrie 4.0 d'intégrer les bases et les méthodologies de l'IoS. Les économies modernes ont adopté une stratégie fondée sur les services (REZABAKHSH et al., 2006). Cela signifie une migration des produits traditionnels orientés structure vers le service. Une telle transformation est nécessaire à la survie d'économies développées comme l'Allemagne et les États-Unis (SORIANO et al., 2013). Le concept d'IoS se réfère explicitement au terme "services" qui dépend du contexte. Le service peut avoir des significations très différentes.

Il est défini par VARGO et LUSCH, 2004 comme "l'application des compétences au profit d'autrui". Il représente le concept qui offre des "produits intangibles" et se caractérise par une consommation simultanée.

Le paradigme de l'IoS offre des possibilités intéressantes en termes de dynamisme et d'interopérabilité. Cependant, dans les domaines impliquant des "objets physiques", il existe toujours un fort besoin de représentation des types et des cas qui sont bien connus dans les approches par composants (NAIN et al., 2010). En outre, l'IoS est une multitude de services qui sont connectés par l'IoT, achetés, vendus, utilisés, réutilisés et composés par un réseau mondial de fournisseurs de services et de consommateurs (ZWEREG, 20008).

L'Industrie 4.0 adopte l'IoS pour offrir des services intelligents et connectera l'usine de l'Industrie 4.0 dans un segment de coopération (LOM, PRIBYL et SVITEK, 2016). La quatrième révolution est basée sur les besoins du CPS pour fournir les systèmes de production distribués et améliorer la communication entre les différentes parties de la chaîne d'approvisionnement (HOFMANN et RÜSCH, 2017). En conséquence, les services intelligents sont essentiels pour maintenir l'Industrie 4.0 dynamique, supervisée et distribuée tout en en offrant de nouvelles méthodes de production autonomes et des politiques clients (HERMANN, PENTEK et OTTO, 2016).

Big Data

Le Big Data fait généralement référence à l'énorme quantité d'ensembles de données collectées et stockées (CHEN, MAO et LIU, 2014). Cette quantité de données ne peut être traitée par les méthodes traditionnelles en raison de leur complexité. Les grandes masses de données sont la représentation des données en plusieurs dimensions et natures (XU et DUAN, 2019), telles que le volume (la quantité de données générée par un système), la vitesse (le plus grand défi pour Big Data, c'est le fait de générer et de traiter des données rapidement voire en temps réel), la variété (qui désigne la complexité des données constituées de nombreuses valeurs numériques) et d'autres aspects (SAGIROGLU et SINANC, 2013).

De nombreuses techniques récentes de Big Data, comme l'analyse, la capture, la conservation, la recherche, le partage, le stockage, le transfert, la visualisation et la confidentialité des informations, sont utilisées dans le domaine industriel.

Le Big Data est présenté comme l'une des composantes de l'Industrie 4.0. Le CPS de l'Industrie 4.0 génère en permanence une énorme quantité de données. Un sérieux problème concernant le stockage, le partage, la visualisation et l'analyse de ces données se pose. Dans ce contexte, les technologies du Big Data sont très sollicitées pour résoudre ces problèmes. L'analyse Big Data permet de suivre l'état des processus de production et d'accélérer la prise de décision pour améliorer les performances des systèmes de production (LEE, KAO et YANG, 2014). De nombreux chercheurs travaillent sur l'intégration du Big Data pour résoudre les problèmes de l'Industrie 4.0, tels que (ÇIFLIKLI et KAHYA-ÖZYIRMIDOKUZ, 2010, SHIN, WOO et RACHURI, 2014, GRÖGER, NIEDERMANN et MITSCHANG, 2012, MEIDAN et al., 2011).

Le Big Data peut être un facteur majeur d'amélioration des usines intelligentes pour créer de nouveaux produits personnalisés et être en capacité d'être dynamiques face aux changements du marché. Cependant, sa mise en œuvre nécessite le développement de nouvelles architectures capables de supporter cette grande quantité de données et les ressources informatiques nécessaires.

Les capteurs (Sensors)

Un capteur est généralement utilisé pour mesurer l'état et les sorties des systèmes, il est considéré comme la "passerelle de la machine" pour détecter les caractéristiques de l'environnement physique telles que la pression, la température, l'humidité, etc. (LEE, KAO et YANG, 2014). Les informations recueillies par les capteurs sont utilisées par les systèmes de supervision, de fabrication et de stockage pour apporter plus de flexibilité, d'autonomie et d'intelligence aux systèmes actuels.

L'évolution industrielle est couplée à une évolution des capteurs (capteurs magnétiques, capteurs non électriques, capteurs de compensation, capteurs chimiques,

capteurs intégrés, capteurs intelligents). L'introduction de l'Industrie 4.0 est accompagnée par l'introduction de Sensors 4.0, qui se caractérisent par l'ajout de l'autonomie en énergie et de la capacité de communication (SCHÜTZE, HELWIG et SCHNEIDER, 2018). En conséquence, les capteurs sont importants pour mesurer l'efficacité de l'industrie, en particulier lorsque le nombre de CPS connectés continue à augmenter (IMKAMP et al., 2016).

Cloud Computing et Cloud Manufacturing

Introduit il y a quelques années, c'est pour certains une technologie informatique innovante et pour d'autres un outils de marketing. Le Cloud computing est un modèle d'accès facile et à la demande au contenu partagé de ressources informatiques configurables (réseaux, serveurs, système de stockage, applications et services) qui peuvent être facilement obtenues et fournies avec un effort minimal (PALLIS, 2010, MELL et GRANCE, 2011).

Le Cloud computing se caractérise par son architecture en couches qui offre des services à différents niveaux aux clients. Le Cloud computing est composé de trois modèles de services (BUYYA, CALHEIROS et LI, 2012, KANICKAM, JAYASIMMAN et JEBASEELI, 2017) :

— Software as a Service (SaaS), est un logiciel déployé sur Internet qui facilite l'allocation des services aux clients,
— Platform as a Service (PaaS), qui est une plate-forme informatique pour la création rapide et facile d'applications (comme les applications Web),
— Infrastructure as a Service (IaaS), il s'agit de fournir aux entreprises une infrastructure d'accès payante (serveurs, stockage, réseau et systèmes d'exploitation), physiquement située chez le fournisseur, moyennant un abonnement payant.

Le Cloud computing offre à ses utilisateurs la facilité, la virtualisation et la disponibilité des informations et ressources. En ce qui concerne le contexte industriel, une vision du Cloud orientée production est connue dans la littérature sous le nom de Cloud Manufacturing (CM). Son but est d'assurer une exploitation optimale et étendue des ressources et de définir des normes sémantiques de coordination, d'accès et de partage équitable des ressources (WU et al., 2013).

Cette version industrielle du Cloud computing est un modèle de fabrication centré sur le client, qui peut être défini comme un modèle permettant l'accès universel à un espace partagé de ressources de production configurables (outils, équipements et logiciels de production) qui peuvent être rapidement réservés, attribués et libérés (XU, 2012, TAO et al., 2014). Le CM contrôle l'accès à une collection partagée de ressources de fabrication à la demande. Cela permet d'avoir des systèmes de production temporaires et reconfigurables qui peuvent améliorer l'efficacité, réduire les coûts de production et permettre un chargement optimal des ressources en réponse à des demandes variables.

Le CM est basé sur les systèmes et les technologies de production existants, le Cloud computing, l'IoT, la virtualisation et les technologies orientées services. Grâce au CM, les industries peuvent optimiser l'exploitation de leurs outils de production en partageant leur planification sur un service Cloud computing. Cela permet à d'autres utilisateurs de bénéficier de ces outils de production. Le CM doit satisfaire aux exigences essentielles de l'Industrie 4.0 telles que la qualité de service, la tolérance aux défaillances, la visualisation, l'interopérabilité, les problèmes liés aux services, la gestion des données et l'évolutivité (RIMAL et al., 2011).

2.2.6 Intégration

L'objectif principal de l'Industrie 4.0 est d'améliorer l'efficacité et la productivité des processus industriels par l'environnement d'automatisation intelligente. L'Industrie 4.0 nécessite trois types d'intégration afin d'assurer la mise en réseau entre les machines et les niveaux hiérarchiques traditionnels de la chaîne de production. Afin d'implémenter l'Industrie 4.0, les trois types d'intégration suivants doivent être prise en considération :

- **L'intégration horizontale :** Comment les stratégies commerciales des entreprises, les nouveaux réseaux de valeur et les nouveaux modèles commerciaux peuvent-ils être soutenus et mis en œuvre durablement à l'aide de CPS (KAGERMANN et al., 2013) ? Une société peut être en relation de concurrence ou de coopération avec de nombreuses autres sociétés liées. L'intégration horizontale entre sociétés, permet de former un écosystème efficace. L'information, la finance et le matériel peuvent circuler couramment entre ces sociétés. Par conséquent, de nouveaux réseaux de valeur ainsi que des nouveaux modèles commerciaux peuvent émerger (WANG et al., 2016). Plusieurs travaux de recherche s'intéressent aux différents modèles de collaborations entre les entreprises (BELKADI et al., 2017).
- **L'intégration verticale :** Comment le CPS peut-il être utilisé pour créer des systèmes de fabrication flexibles et reconfigurables (KAGERMANN et al., 2013) ? Une usine possède plusieurs sous-systèmes physiques et informatiques (actionneur, capteur, contrôle, gestion de la production, fabrication, planification). Il est essentiel d'intégrer verticalement le flux d'information des différents niveaux jusqu'au niveau de la planification des ressources de l'entreprise (ERP). Cette architecture facilitera la mise en place des systèmes de production flexibles, reconfigurables et capables de s'adapter aux différents types de produits. L'information massive est collectée et traitée pour rendre le processus de production transparent (WANG et al., 2016). L'Industrie 4.0 s'attend à intégrer verticalement les sous-systèmes hiérarchiques pour pouvoir transférer les systèmes de production actuels vers des systèmes de production reconfigurables et hautement intégrés dans l'usine intelligente.
- **L'intégration numérique du bout en bout (end to end digital integration) :** Comment le CPS peut-il être utilisé pour fournir des processus d'affaires de bout en bout, y compris le flux de travail d'ingénierie (KAGERMANN et al., 2013) ? Dans un processus de création de valeur centré sur le produit, une chaîne d'activités est impliquée (l'expression des besoins du client, la conception et le développement de produits, la planification de la production, l'ingénierie de production, la production, les services, la maintenance et le recyclage). L'effet de la conception du produit sur la production et le service peut être prévu (WANG et al., 2016). La récupération des informations tout au long de la chaîne de valeur du produit en se basant sur les CPS et l'IoT demande la mise en place d'un système d'information adapté et aussi une formation des ingénieurs et utilisateurs.

La figure 2.4 présente schématiquement les trois types d'intégration. Dans l'Industrie 4.0, ces différentes intégrations peuvent connaître une augmentation significative par l'utilisation du CPS et de l'IoT. L'Industrie 4.0 permet aux capteurs omniprésents, aux systèmes de terminaux embarqués, aux systèmes de contrôle intelligents et aux installations de communication de former un réseau intelligent au sein du CPS. L'interconnexion peut se faire d'homme à homme, d'homme à machine,

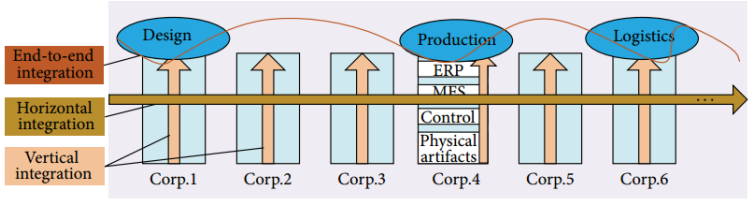


FIGURE 2.4 – Les trois types d'intégration de l'Industrie 4.0 (WANG et al., 2016).

de machine à machine ou de service à service, afin de réaliser une intégration horizontale, verticale et de bout en bout complète. Ceci augmentera aussi l'occurrence d'autres risques liés à l'augmentation de la complexité des systèmes et leur sécurité. Des architectures innovantes doivent être mises en place pour éviter ces risques.

2.2.7 Les exigences de l'Industrie 4.0

Comme présenté auparavant, le Cloud Manufacturing est l'une des solutions les plus importantes proposées pour résoudre le problème du cumul d'une grande quantité de données complexes. Néanmoins, cette solution souffre de plusieurs limitations dans ce nouveau contexte d'Industrie 4.0. Le CM ne permet toujours pas de satisfaire les exigences de l'Industrie 4.0 comme le Cloud computing en termes du temps réel, d'interopérabilité, de la sécurité des communications, etc …. Plusieurs projets ont discuté des exigences de l'Industrie 4.0, comme (GRANGEL-GONZÁLEZ et al., 2016), qui décrit une approche qui représente sémantiquement les informations de l'Industrie 4.0. En conséquence, (LOM, PRIBYL et SVITEK, 2016, KHAN et al., 2017, WAIDNER et KASPER, 2016, SALDIVAR et al., 2015) détaillent les exigences ou les principes de conceptio" sur lesquels se basent la création des applications d'Industrie 4.0 qui pourraient être basées sur le Cloud computing :

— Fiabilité,
— Administration des actifs Shell (normalisation, sémantique, interopérabilité),
— Communication (capacité d'échange en temps réel, protocoles de transfert de données, QoS adaptative dynamique, latence, bande passante),
— Considérations de sécurité (authentification, politiques de confidentialité, contrôle d'accès),
— Optimisation,
— Capacité d'analyse des données,
— Interactions homme-machine.
— Virtualisation
— Décentralisation
— Orientation du service
— Modularité

Les exigences de l'Industrie 4.0 doivent permettre d'assurer le bon fonctionnement du processus de production et ainsi un haut niveau de performance des applications industrielles (maintenance, planification, diagnostic, supervision, etc.)

Comme l'Industrie 4.0 a comme objectifs d'augmenter les profits et la qualité et de minimiser le coût de la fabrication, chaque technologie proposée afin d'intégrer les composants déjà existants de l'Industrie 4.0 doit certainement être en mesure de répondre à ces exigences.

La nouveauté dans la proposition de l'Industrie 4.0 n'est pas dans une nouvelle technologie, mais dans la combinaison des technologies disponibles d'une nouvelle

façon (DRATH et HORCH, 2014). Le quadruplet (IoT, IoS, CPS et Cloud) et l'avenir de la production de biens physiques sont dépendants, ce qui entraîne que ces composants offrent un énorme potentiel d'innovation dans la production.

Ce paradigme prometteur conduit aux nombreux résultats bénéfiques qui peuvent faire face aux défis mondiaux en ce sens que les produits personnalisés peuvent être produits de manière efficace, efficiente, flexible et rentable (WANG et al., 2016).

Conformément au contexte de l'Industrie 4.0, les produits peuvent contrôler leur propre processus de production et le client devrait être plus impliqué dans les processus de création de nouveaux produits (PISCHING et al., 2015). Avec l'Industrie 4.0, il sera possible d'intégrer des fonctionnalités spécifiques aux clients et aux produits dans les phases de conception, de configuration, de commande, de planification, de production, d'exploitation et de recyclage. Cela fournira des produits personnalisés d'une manière rentable. L'Industrie 4.0 permettra aux employés de contrôler, réguler et configurer les ressources et les étapes de production. La réussite d'intégrer les services Web dans l'Industrie 4.0 et la portée du potentiel de cette technologie prédisent une augmentation considérable de son utilisation (KAGERMANN et al., 2013).

2.3 Problématique générale

Dans cette section, nous allons tout d'abord expliquer notre contexte industriel pour délimiter le périmètre général de cette thèse. La question importante qui peut-être posée par les industriels concerne le potentiel de performances du Cloud computing. Les questions de la problématique générale qui se posent sont : si le Cloud computing n'est plus la solution optimale, quels sont les facteurs qui ont causé une telle limitation ? Quelle est la meilleure solution afin de renforcer le rôle du Cloud computing dans le domaine industriel et plus spécifiquement dans le contexte de l'Industrie 4.0 ? Et par la suite pour finir cette section nous présentons le positionnement technique et scientifique de la thèse.

2.3.1 Contexte

Les éléments constitutifs de l'Industrie 4.0 doivent satisfaire ses exigences afin d'assurer un haut niveau de fonctionnement et d'intelligence. Le manque de performances du Cloud computing est problématique devant les besoins des applications industrielles actuelles. L'intégration de nouvelles technologies qui se base sur le traitement des données et sur l'IoT semble la solution la plus adaptée pour répondre à ces limitations du Cloud computing et du Cloud Manufacturing. Le développement de l'Industrie 4.0 est l'un des enjeux prioritaires fixés par l'Europe dans les nouveaux programmes de recherche. Actuellement l'Industrie 4.0 est l'un des thèmes de recherche et de formation dans le cadre de la Chaire CISCO / CESI / Vinci Energies intitulée « Industrie et Services de demain ».

Dans les parties précédentes nous avons présenté le paradigme de l'Industrie 4.0, en évoquant son historique, ses composantes et ses exigences. L'Industrie 4.0 est une usine fortement connectée. Les machines et produits peuvent échanger des données pour élaborer leurs différentes tâches. Des quantités importantes d'informations variées sont générées et remontées chaque seconde par les différents capteurs installés au sein de l'atelier de production que ce soit dans les outils de production, le système logistique ou les services annexes. Ces informations doivent être collectées et traitées en temps réel afin d'assurer un bon fonctionnement du système de production. Les

systèmes centralisés qui collectent et traitent les données (Le Cloud computing) ont montré leurs limites quand il s'agit du traitement d'un très grand nombre de données et de prises de décision en temps réel.

Cependant, les systèmes distribués ont prouvé leur efficacité pour la gestion des systèmes complexes en général et plus particulièrement les systèmes de production complexes (TRENTESAUX, 2009). Pour cela, il faut décentraliser le traitement des données ainsi que la prise de décision tout en assurant une bonne interopérabilité entre les différentes ressources des systèmes.

L'adoption et la mise en œuvre de l'Industrie 4.0 sont soumises à plusieurs défis, tels que :

— Le manque de transmission en temps réel et de mises à jour d'informations concernant les ressources de production en cours du processus de fabrication,
— L'augmentation du temps d'attente (délais) entre la détection d'un problème, la planification ou l'ordonnancement des tâches et la prise d'une décision fiable,
— La garantie d'un niveau respecté de sécurité et de confidentialité des ressources de production des industriels et les informations liés aux clients,
— L'urgence d'être prêt à faire face à tout type de concurrence et aux changements soudains du marché (adoption d'une politique de production flexible et dynamique afin d'augmenter les profits et diminuer le coût).

2.3.2 Le rôle et l'intérêt du Cloud Computing dans les applications industrielles

La cause majeure des problèmes limitant l'adoption de l'Industrie 4.0 est la quantité croissante de données, de plus en plus difficiles à traiter et à échanger (LEE, KAO et YANG, 2014). Comme cette quantité de données va continuer à croître suivant le nombre croissant des objets connectés, un nouveau phénomène gênant pour les centres de données peut arriver en impactant les performances et en créant une crise du stockage (ZANDER et MÄHÖNEN, 2013). Ce phénomène, connu sous le nom de "tsunami des données", peut également affecter les installations industrielles et notamment leurs réseaux informatiques non dimensionnés pour ces nouveaux besoins.

Cette nouvelle génération de systèmes de production commence à générer une énorme quantité de données complexes tout au long du cycle de vie du produit et de la chaîne de valeur. Pour contrôler ce flux d'informations, plusieurs améliorations, réglages et traitements doivent être effectués.

Malheureusement le Cloud, qui est basé sur le traitement de l'information sur demande, n'est pas adéquat pour les applications sensibles à la latence. Par conséquent il ne peut pas couvrir les besoins de ce genre des applications basées sur l'IoT (PISCHING et al., 2015). Donc le manque de liaison entre les différentes parties et l'insuffisance du rendement du Cloud computing, qui est distant, centralisé et avec des risques de latence de traitement, représentent de vrai problèmes qu'il faut résoudre lors de l'implémentation et le fonctionnement des usines de futur.

Ajouté à cela, les risques liées à la cyber-sécurité ou à l'augmentation incontrôlée de la complexité des systèmes (des données dépassant la capacité du système de traitement) existent. L'une des solutions les plus intéressantes qui peuvent résoudre ce genre de problème semble être l'edge computing et ses déclinaisons. Ce nouveau paradigme va être présenté et détaillé dans la section suivante.

2.3.3 Edge computing - solutions existantes et déclinaisons

Les applications utilisées soit par les industries soit dans d'autres domaines applicatifs souffrent de la sensibilité à la latence des réponses ou de la prise de décision quand elles regroupent un grand nombre de nœuds connectés qui ne cessent de croître à une fréquence très rapide. Afin de gérer ces besoins, les applications se basent sur l'IoT et le Cloud computing afin de traiter, acheminer et stocker les données générées. D'après ce qui est conclu dans la section précédente, le Cloud computing ne semble plus efficace devant les exigences croissantes des applications basées sur l'IoT et le Cloud computing.

La solution qui peut résoudre une telle problématique c'est de diminuer le temps de transmission (envoie de demande, traitement et réponse) et de distribuer le calcul sur le plus grand nombre de dispositifs connectés possible. Cette solution doit satisfaire les besoins suivants :

— Être basé sur l'IoT,
— Étendre les fonctionnalités du Cloud computing,
— Être compatible avec le modèle de calcul du Cloud computing,
— Être évolutif en face du nombre grandissant des nœuds qui composent l'architecture de l'application,
— Respecter les faibles latences exigées par les applications,
— Distribuer le calcul tout le long du réseau,
— Être à proximité de la couche la plus basse afin de diminuer les délais.

L'Edge computing se réfère aux technologies qui traitent les données à proximité des utilisateurs finaux. Ce modèle de calcul fournit les ressources du Cloud computing au bord du réseau (PAN et MCELHANNON, 2017, SATYANARAYANAN, 2017). La localisation et les capacités de l'Edge computing semblent être une solution aux limitations des serveurs centralisés du Cloud computing.

Par contre, l'Edge computing est considéré comme la technologie mère, ce qui reste très vaste. Donc elle regroupe de nombreuses particularités, qui dépendent de l'application ou du domaine de l'utilisation. En étudiant la littérature, différentes déclinaisons de l'edge computing ont été identifiées :

— Les Cloudlets,
— Le Mobile Edge Comuting,
— Le Fog Computing.

Dans ce qui suit une brève description de chaque déclinaison de l'Edge computing, afin de décider laquelle est la plus convenable pour surmonter les limitations du Cloud computing dans le contexte d'Industrie 4.0.

Cloudlets

Le Cloudlet est un projet qui vise à rapprocher les services du Cloud computing aux utilisateurs mobiles, tout en créant des "mini" centres de données virtuels et intermédiaires (PAN et MCELHANNON, 2017). Un Cloudlet est présenté comme la miniature d'un serveur Cloud computing. Cette version plus légère est implémentée généralement à proximité des utilisateurs finaux. Les Cloudlets sont installés sur une ou plusieurs machines virtuelles (VMs) afin qu'elles soient localisées et repérables. Les Cloudlets sont caractérisées par leur structure dynamique. Lorsqu'une demande de la part d'un utilisateur mobile est générée, la latence causée par la transmission au serveur du Cloud computing distant peut nuire à la QoS et interrompre la communication (basée sur le WLAN). La figure 2.5 décrit l'architecture hiérarchique basée sur les Cloudlets.

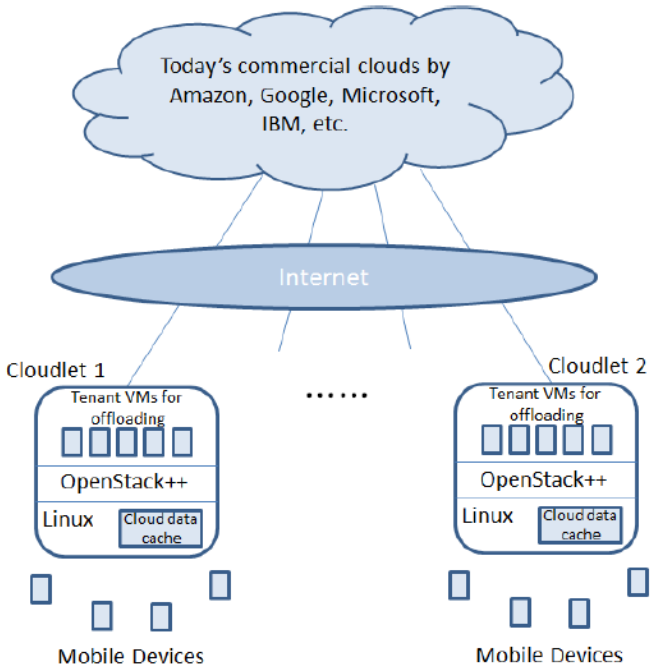


FIGURE 2.5 – L'architecture basée sur les Cloudlets (PAN et MCELHANNON, 2017).

L'avantage des Cloudlets est pour les utilisateurs mobiles, ils ont la possibilité d'obtenir les services personnalisés en forme d'instances des logiciels requis sur les VMs d'une manière fluide et légère (BAHTOVSKI et GUSEV, 2014). L'autre avantage est l'auto-configuration et l'autonomie des Cloudlets (SHAUKAT et al., 2016). Donc les Cloudlets sont utilisés principalement pour les utilisateurs mobiles qui auront besoin des ressources du Cloud computing sans alourdir le réseau ou dépasser les capacités limitées de leurs matériels. Donc la latence et la bande passante sont notamment réduites grâce à cette solution.

Malgré la faible latence des transmissions entre les Cloudlets et les utilisateurs mobiles, le calcul et le stockage des données et l'amélioration de la QoS, les Cloudlets ne peuvent pas assurer l'amélioration des performances du Cloud computing dans le contexte industriel. L'absence de communication continue avec le Cloud computing, le manque d'aspect serveur de calcul local et la personnalisation spécifiques pour les utilisateurs mobiles entraînent une inadéquation avec les besoins en environnement industriel.

Mobile Edge Computing (MEC)

Le MEC est une déclinaison de l'Edge computing qui a un fonctionnement similaire à celui des Cloudlets. Le MEC est introduit afin de rapprocher les services du Cloud computing à proximité des utilisateurs finaux, qui sont généralement les opérateurs et clients mobiles du réseau d'accès radio (AHMED et REHMANI, 2017).

Le MEC est composé des serveurs situés à la périphérique du réseau pour faire des calculs et stockage en temps réel (BUYUKKOC, 2016). Les serveurs du MEC fonctionnent dans les stations de bases. Conséquemment ces utilisateurs mobiles vont profiter des communications avec de faibles latences à l'aide du support de mobilité du MEC.

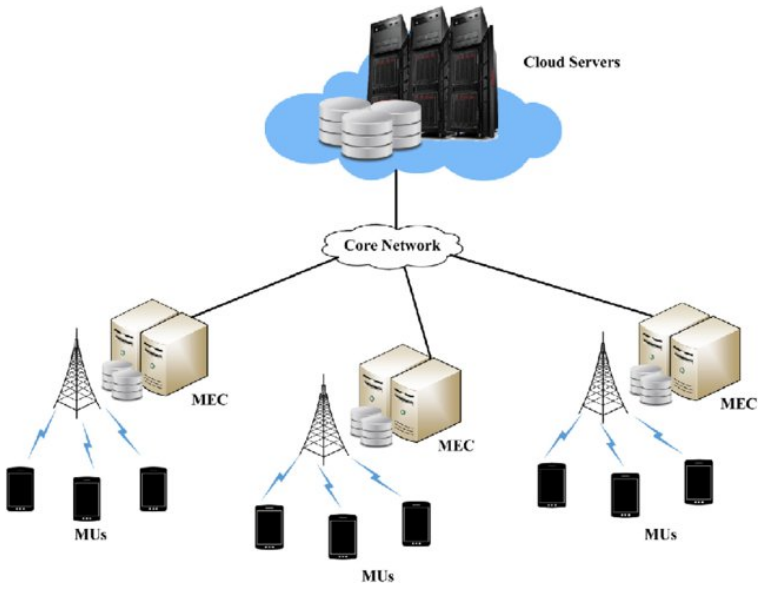


FIGURE 2.6 – L'architecture basée sur le MEC (ELGENDY et al., 2019).

Cette version, dédiée aux opérateurs des réseaux mobiles, est fortement exploitée pour satisfaire les exigences de la cinquième génération (5G) des standards pour la téléphonie mobile. Le MEC est adopté pour de nombreuses applications mobiles (la gestion du trafic, la distribution du contenu, la mise en cache du contenu en bord de réseau,etc.) La figure 2.6 présente l'architecture basée sur le MEC. La couche la plus basse est composée des nœuds mobiles et le réseau de transmission est le réseau radio.

Le MEC est spécifique pour les applications mobiles et les nœuds connectés au réseau radio restreint. Comme l'Industrie 4.0 regroupe des objets connectés à l'IoT qui est un réseau internet ouvert et général, cette version de l'Edge computing est incompatible avec le contexte des données de l'Industrie 4.0.

Fog Computing

Le Fog computing, est une solution de distribution de calcul proposée par Cisco. Il a plusieurs caractéristiques qui conviennent aux exigences de l'Industrie 4.0. Le Fog computing et ses avantages n'étaient pas bien connus à l'époque de l'apparition de l'Industrie 4.0, ce qui explique son absence comme l'une de ses composantes. Plus tard, plusieurs comparaisons ont été établies (O'DONOVAN et al., 2019) et ont prouvé le rôle important du Fog dans le contexte de l'Industrie 4.0. La plupart des travaux concernant ce genre d'étude sont apparus dans les années 2019 et 2020 (pendant les travaux de cette présente thèse).

En effet, le Fog computing est capable de faciliter la mise en œuvre du concept d'Industrie 4.0 où, il peut être adopté pour permettre l'inter-connectivité des CPS dans le cadre industriel et améliorer les fonctionnalités et les performances du Cloud computing. Ce paradigme est capable de répondre aux exigences de l'Industrie 4.0 en termes de flexibilité, de communication, d'autonomie, de sécurité, d'analyse des données en temps réel, de fiabilité (QoS), et d'interopérabilité (BAR-MAGEN NUMHAUSER et MESA, 2013). (BACCARELLI et al., 2017) ont mentionné le rôle important que le Fog computing peut offrir pour améliorer l'Industrie 4.0 en fournissant un traitement évolutif en termes de calcul, de stockage et de routage. L'emplacement du Fog est capable de supporter le flux massif de données provenant des appareils

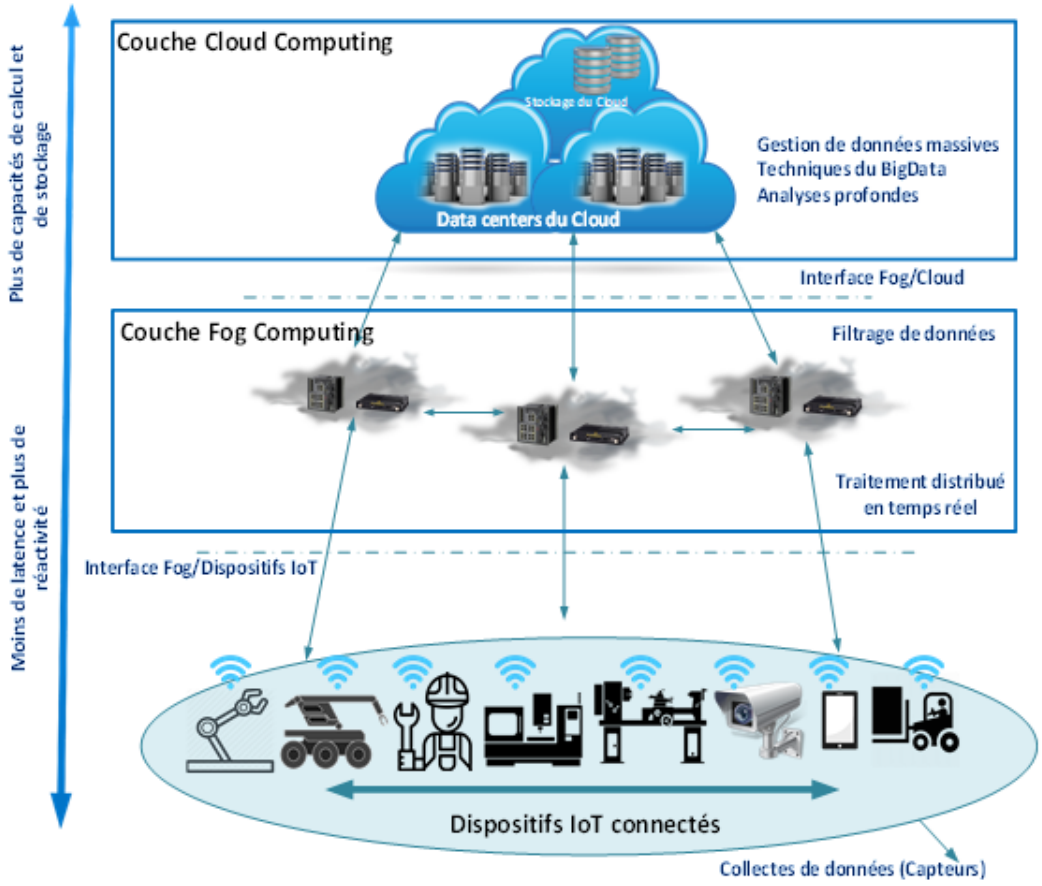


FIGURE 2.7 – L'architecture basée sur le Fog computing.

et dispositifs connectés et de filtrer les données acheminées au Cloud computing qui est distant. Aussi, PIZOŃ et LIPSKI, 2016 ont proposé le Fog computing comme un nouveau composant qui va résoudre le problème de traitement des données pour les systèmes industriels.

(GEORGAKOPOULOS et al., 2016) ont choisi la même direction en confirmant que l'ajout de l'Edge Computing est crucial pour l'Industrie 4.0 pour les mêmes raisons. Son architecture qui se base sur l'IoT est une forte raison pour laquelle le Fog computing semble la technologie la plus adéquate pour être un composant de l'Industrie 4.0 à côté du Cloud computing. La figure 2.7 présente l'architecture générale basée sur le Fog computing.

Le tableau 2.2 résume les caractéristiques de chaque déclinaison (Cloudlets, MEC et Fog Computing) de l'Edge computing et montre les différences entre elles. D'après ce tableau 2.2 le choix de la solution du Fog computing dans le contexte de l'Industrie 4.0 semble la plus susceptible d'être utilisée comme moyen de distribution des calculs, moyen de communication entre différents organes de l'atelier et support pour le cloud computing. Dans le suite de ce chapitre nous allons étudier le fog computing comme solution technologique pour la répartition du traitement des données dans les ateliers de production.

TABLE 2.2 – Comparaison entre les déclinaisons de l'Edge computing (DOLUI et DATTA, 2017, AI, PENG et ZHANG, 2018).

Caractéristiques	Fog computing	MEC	Cloudlet
Localisation	Au bord du réseau	Au bord du réseau	Au bord du réseau
Extension	Des services Cloud	Des services Cloud en utilisant le WLAN	Des services Cloud en boite locale
Utilisation	Applications distribuées et sensibles à la latence	Applications des dispositifs téléphoniques mobiles	Applications de gestions des ressources
Types des noeuds	Routeur, switch, point d'accès ou passerelle	Serveurs tournant sur une station de base	Centre de données en boîte
Sensibilisation au contexte	Moyen	Haut	Bas
Architecture logicielle	Couche d'abstraction du Fog	Basé sur un orchestrateur mobile	Basé sur un agent de cloudlet
Proximité	Mono ou multi-sauts	Mono-saut	Mono-saut
Mécanisme d'accès	Bluetooth, Wi-Fi, Réseau mobile	Réseau mobile	Wi-Fi
Communication inter-modale	Supporté	Partiel	Partiel
Contexte d'application	IoT	Les exigences de la 5G dans le secteur des télécommunications	Les applications du Mobile computing

2.4 Fog Computing

Nous allons présenter un état de l'art sur le paradigme du Fog computing contenant ses caractéristiques et l'évolution de son architecture. Nous allons aussi montré son lien avec le cloud computing en termes de complémentarités, similarités et différences.

2.4.1 État de l'art

Lorsque les techniques et les dispositifs d'IoT s'impliquent encore plus dans la vie des gens, le paradigme actuel du Cloud computing peut difficilement satisfaire leurs besoins en matière de mobilité, de localisation, de faible latence et de gestion d'urgences (STOJMENOVIC et WEN, 2014).

Une vague émergente de déploiements de l'internet, notamment l'IoT, nécessite un support de mobilité et une répartition géographique en plus de la prise en compte de l'emplacement (BONOMI et al., 2012).

Malgré l'efficacité des réseaux et l'augmentation de leur capacité de transmission la distance entre l'origine de l'information et le centre de son traitement est de plus en plus grande. L'introduction de l'IoT et des CPS complexifie la tâche de traitement en augmentant la quantité de données transmise et leur hétérogénéité. (STOJMENOVIC et WEN, 2014) atteste que le fog computing représente l'une des architectures les plus adaptées et plus efficaces pour palier aux problématiques de traitement de données massives dans les ateliers de productions.

Différents dispositifs génèrent divers types de données avec des fréquences différentes. Ces services sont sensibles à la latence et nécessitent une réponse en temps réel. Il est nécessaire de décider quel type de données doit être téléchargé sur le Cloud, sans encombrer le réseau.

L'IoT lié à un Cloud computing génère de l'information qui consomme beaucoup d'énergie et de puissance de calcul lors de son traitement et stockage. Il sera très important de les gérer efficacement et rapidement dans le Cloud computing. Mais à cause de la sensibilité à la latence de certaines applications, il n'est pas toujours possible de communiquer les informations aux centre de données du Cloud lointains. Comme l'une des solutions, le Fog computing est livré aux utilisateurs et fournisseurs pour résoudre ces problèmes (AAZAM et HUH, 2015). De même, le stockage temporaire, le pré-traitement, la sécurité des données et la vie privée, peuvent se faire facilement et plus efficacement en présence d'une amélioration du Cloud computing (AAZAM et HUH, 2014). Afin d'assurer le traitement local et rapide des données il est nécessaire de rapprocher les centres de traitement et de sélection de données des dispositifs IoT en créant une couche entre le Cloud computing et les CPS.

Cette nouvelle couche est appelée Fog Computing. Cette nouvelle solution proposée par Cisco représente donc un Cloud près des terminaux offrant des services à l'utilisateur final (ROY, BOSE et SARDDAR, 2015). Elle fournit une nouvelle génération d'applications de gestion et d'analyse des données.

Définition

Dans la littérature, le Fog computing a été défini de plusieurs façons. Du point de vue de Cisco, il est considéré comme une extension du noyau du Cloud computing à la périphérie du réseau (COMPUTING, 2016). Ce concept est appelé aussi Fog Networking, il permet aux applications sur des milliards d'appareils IoT connectés de fonctionner directement à la périphérie du réseau (STOJMENOVIC et WEN, 2014).

Aazam et al. (AAZAM et HUH, 2014), définissent le Fog computing comme le réseau entre le réseau périphérique et le Cloud computing. Il s'agit de rapprocher les ressources réseau des réseaux sous-jacents, c'est aussi une amélioration du Cloud computing, qui offre des services plus raffinés.

Cette nouvelle plate-forme est simplement un "nuage proche du sol" qui fournit une nouvelle génération d'applications et de services informatiques, de stockage et de mise en réseau entre les terminaux et les centres de données Cloud computing traditionnels (BONOMI et al., 2012). Un système Fog peut également être défini comme un modèle de programmation de haut niveau pour les applications Internet futures (p. ex. géo-spatialement réparties, à grande échelle et sensibles à la latence) (HONG et al., 2013b).

D'autres chercheurs tels que (AAZAM et HUH, 2015) et (MUKHERJEE et al., 2017b) confondent les appellations du "Fog computing" et du "Edge" Computing, et le définissent comme une plate-forme chargée de fournir différents services entre les terminaux de l'IoT et les Clouds traditionnels. (SINGH et al., 2019) considèrent le Fog

comme un paradigme de calcul distribué qui étend fondamentalement les services fournis par le Cloud computing au bord du réseau.

D'après (KIANI, ANSARI et KHREISHAH, 2019), il s'agit d'un intermédiaire entre l'exécution du Cloud computing et les demandes des périphériques, ce qui facilite la gestion des services informatiques et de stockage entre les centres de données et les périphériques. Selon le site web OpenFog consortium.org (*Fogconsortium*), le Fog Computing est "une architecture horizontale au niveau du système qui distribue les fonctions informatiques, de stockage, de contrôle et de mise en réseau au plus près des utilisateurs dans un continuum Cloud-to-thing".

(YI, LI et LI, 2015) donnent une définition intéressante et complète du fog computing comme suit : *"Fog computing comme la plate-forme hautement virtualisée qui traite l'information directement à la périphérie du réseau, pour accomplir le traitement, le calcul, le stockage et les services de routage entre les terminaux et serveurs Cloud computing traditionnels. Ce scénario regroupe un grand nombre de dispositifs hétérogènes, décentralisés et parfois autonomes. Ces dispositifs connectés communiquent et coopèrent pour accomplir des tâches et offrir de nouveaux services d'IoT."* Dans la suite de cette thèse nous allons retenir cette définition qui semble être la plus précise et la plus détaillée d'après l'état de l'art.

D'autres travaux, tels que (YOUSEFPOUR et al., 2019, ROY, BOSE et SARDDAR, 2015, HONG et al., 2013a, NI et al., 2017, STOLFO, SALEM et KEROMYTIS, 2012, KOPRAS, IDZIKOWSKI et KRYSZKIEWICZ, 2019), ont proposé plusieurs définitions du Fog mais elles convergent dans la même direction que celles déjà citées.

Les travaux intégrant le Fog Computing

De nombreux projets de recherche présentent l'utilisation du paradigme Fog Computing dans divers domaines tels que les soins de santé intelligents, la gestion des ressources et la prise de décision, la ville intelligente, la sécurité, l'automatisation industrielle et plusieurs autres. Le paragraphe suivant présente quelques applications générales du Fog. De nombreuses applications nécessitent à la fois la localisation du Fog et la globalisation du Cloud computing, notamment pour l'analyse des données et le Big Data (BONOMI et al., 2012).

Dans le domaine de la santé, on peut citer une application de prévention du virus Zika utilisant le Fog computing présentée dans (SAREEN, GUPTA et SOOD, 2017) et un système de surveillance proposé par (RAHMANI et al., 2018) pour gérer l'utilisation massive des données des systèmes de santé. Dans (DUBEY et al., 2017), les auteurs proposent un système basé sur le Fog computing pour gérer les données médicales et d'autres applications dans le domaine de la santé existent telles que celles dans (KRAEMER et al., 2017).

Le travail de (GUPTA et al., 2016) introduit un simulateur, appelé iFogSim, pour modéliser et simuler les environnements de Fog Computing sans domaine d'application spécifique. Il permet de comparer les techniques de gestion des ressources sur la base de critères de qualité de service. Le Fog est utilisé pour accélérer la transmission de données entre les appareils IoT et le Cloud computing.

Le Fog a été adopté comme solution pour élaborer un nouveau système d'aide à la décision (DSS) (ROY, BOSE et SARDDAR, 2015). Ce DSS vise à empêcher les conducteurs d'utiliser des appareils portatifs pendant la conduite à l'aide des serveurs Fog. (HONG et al., 2013a) présente "Mobile Fog", qui est un modèle de programmation pour la gestion à la demande du traitement des données pour les applications IoT.

Une autre méthodologie est développée par (AAZAM et HUH, 2015) pour aider les clients du Cloud computing à se connecter à l'aide de la couche Fog, afin de négocier et acquérir les services requis au meilleur prix. Dans ce cas, les centres de données sont considérés comme des serveurs Fog qui assurent le traitement et l'échange de données avec la couche Cloud computing.

Dans le contexte d'une ville intelligente, un système intelligent de gestion du trafic est proposé par (DASTJERDI et al., 2016) pour gérer le flux des requêtes exécutant les données générées par les capteurs déployés. Une autre application dans le contexte de la ville intelligent est proposée par (PERERA et al., 2017).

(TAO, OTA et DONG, 2017) propose un modèle hybride de Fog Computing pour les réseaux véhiculaires à grille en 5G nommé "Foud". Dans le domaine de la sécurité, les recherche de (STOLFO, SALEM et KEROMYTIS, 2012) ont montré que les mécanismes de protection des données existants, tels que le cryptage, ne sont pas efficaces à empêcher les attaques aux données, en particulier celles du Cloud computing. Ils ont proposé une approche de sécurité des données du Cloud computing et du Fog computing en utilisant la technologie Offensive Decoy, qui surveille l'accès aux données.

Dans (ZHANG et al., 2016), les auteurs présentent le premier système de cryptage basé sur les attributs de politique textuelle de chiffrement (CP-ABE) basé sur le Fog computing, qui prouve son efficacité.

Les auteurs de (HU et al., 2017a) proposent un système d'identification et de résolution des visages basé sur l'informatique Fog pour résoudre les problèmes de calcul, de communication et de stockage.

De nombreuses autres applications du Fog computing dans plusieurs domaines se trouvent dans la littérature (WANG et al., 2019c, NING, HUANG et WANG, 2019, KOPRAS, IDZIKOWSKI et KRYSZKIEWICZ, 2019), comme (DANTU, KO et ZIAREK, 2017) qui intègre le Fog computing dans le domaine de la téléphonie mobile.

Récemment le Fog computing, qui a montré des performances remarquables dans divers domaines, a commencé à être déployé dans le contexte industriel. Le Fog est utilisé dans des applications industrielles comme la logistique (LIN et YANG, 2018), la maintenance prédicative (GAZIS et al., 2015), la traçabilité des produits (CHEN, 2017), la consommation d'énergie (BACCARELLI et al., 2017), la planification (VERBA et al., 2019), le monitoring (MUKHERJEE et al., 2017a), le diagnostic (WU et al., 2017), etc. Ces applications vont être détaillées au niveau du deuxième chapitre.

Les caractéristiques

Le Fog Computing repose sur plusieurs principes comme le traitement distribué des données avec les niveaux de résilience, d'échelle, de vitesse et de mobilité nécessaires pour délivrer efficacement et effectivement les informations à valeurs ajoutées que les données générées peuvent créer lorsqu'elles sont correctement traitées sur le réseau. De même, le stockage temporaire, le pré-traitement, la sécurité des données et la vie privée, ainsi que d'autres tâches de ce type peuvent se faire plus facilement et plus efficacement en présence du Fog computing.

Ce réseau de bord assure la livraison des données en temps réel. De plus, cette nouvelle infrastructure prend en charge l'hétérogénéité, car les dispositifs Fog incluent les périphériques des utilisateurs finaux, des points d'accès, des routeurs et des commutateurs du bord. Par contre, l'interopérabilité pose toujours un problème avec les sources de données hétérogènes.

Le paradigme du Fog Computing semble bien positionné pour l'analyse de masses de données en temps réel. Il prend en charge les points de collecte des données densément répartis et offre des avantages dans le divertissement, la publicité, l'informatique personnelle, l'industrie et d'autres applications. Basé sur l'analyse de plus de 20 définitions du Fog computing, plusieurs mots communs ont été identifiés pour caractériser le Fog Computing. Ces caractéristiques peuvent être définies comme suit :

— **Distribution :** D'après (BONOMI et al., 2012, STOJMENOVIC et WEN, 2014, AAZAM et HUH, 2015, MUKHERJEE et al., 2017b, ou encore OSANAIYE et al., 2017), le Fog Computing se caractérise par une architecture géographique dense et distribuée. Cette distribution du Fog permet de transmettre rapidement le flux des données et de les traiter sur l'ensemble du réseau. L'accès sans fil prédominant aux nœuds du Fog facilite la mise en œuvre d'architectures distribuées.

— **Mobilité :** La mobilité du Fog est définie comme la capacité d'échanger des informations avec des applications mobiles omniprésentes. Il est basé sur le fait d'éviter de renvoyer les mêmes informations plusieurs fois à plusieurs nœuds du Fog. En général, il a besoin d'un service de découverte pour détecter les nœuds du Fog à proximité (SOO, CHANG et SRIRAMA, 2016). Ceci améliore la QoS et offre une connectivité continue des dispositifs mobiles.

— **Aspect temps réel, faible latence :** Le paradigme du Fog Computing est bien positionné pour l'analyse de grandes quantités de données près de la source et en temps réel. Toutes les interactions des applications sensibles au délai sont traitées en temps réel au lieu d'un traitement lent traditionnel, afin de les transmettre aux serveurs d'analyse du Cloud computing (BONOMI et al., 2012, MUKHERJEE et al., 2017b, OSANAIYE et al., 2017). En conséquence, l'aspect temps réel du Fog améliore directement les performances en termes de latence et de qualité de service (BONOMI et al., 2012, STOJMENOVIC et WEN, 2014). Ceci assure la transmission de données avec des flux de haute qualité et une faible latence vers les nœuds AAZAM et HUH, 2015. Il est particulièrement adapté aux applications à faible latence (streaming vidéo, jeux, réalité augmentée,...).

— **Hétérogénéité :** Le Fog computing se caractérise par la prise en charge de l'hétérogénéité des nœuds et des environnements. Les nœuds (ayant des formes différentes et déployés dans une variété d'environnements) incluent des dispositifs d'utilisateurs finaux, des points d'accès, des routeurs et des commutateurs de périphérie et permettent le traitement de données hétérogènes dans différents serveurs du Fog (BONOMI et al., 2012, AAZAM et HUH, 2014, MUKHERJEE et al., 2017b, OSANAIYE et al., 2017).

— **Interopérabilité :** Le nombre croissant des objets IoT nécessite un certain niveau d'interopérabilité afin d'assurer la communication horizontale entre les dispositifs. Le Fog offre une architecture intéressante pour assurer une communication fiable et rapide à travers le réseau. Selon (BONOMI et al., 2012, **osanaiye2017 Cloud**, CHANG, SRIRAMA et BUYYA, 2017), en raison de l'interaction avec l'IoT et le Cloud computing, qui sont interopérables, il est important que le Fog computing supporte également l'interopérabilité.

— **Sécurité :** En outre, ce nouveau paradigme peut permettre de renforcer la sécurité des données. La confidentialité et le contrôle sont également renforcés lors de la mise en œuvre de la solution Fog (AAZAM et HUH, 2015, CHANG, SRIRAMA et BUYYA, 2017). Les politiques de sécurité centralisées du Cloud ne sont pas adaptées à la distribution de la solution du Fog. La présence de

nombreuses ouvertures Fog distribués menacera l'ensemble du réseau (STOJMENOVIC et WEN, 2014), en créant de nouvelles ouvertures non sécurisés. Pour le Fog Computing, il existe de nombreux niveaux de sécurité comme La sécurité de bout en bout, qui est primordiale pour le succès de l'installation de Fog, impliquant que toutes les couches doivent être sécurisées. Cependant, plusieurs recherches ont étudié ce problème de sécurité et ont tenté de trouver des solutions pour améliorer la confidentialité dans une architecture basée sur le Fog (Voir KHAN, PARKINSON et QIN, 2017, GUAN et al., 2018, HU et al., 2017b).

L'évolution de l'architecture

En conclusion de ce que l'on trouve dans la littérature, les couches d'une architecture basée sur le Fog sont communes à toutes les propositions. Comme le montre la figure 2.7, il est formé de trois couches : La couche des objets connectés, la couche des noeuds Fog et la couche des serveurs Cloud. Par contre cette architecture tiers a évolué depuis sa première proposition (AL YAMI et SCHAEFER, 2019). La couche Fog est maintenant supporté par une ou deux couche supplémentaire selon l'application. Ces couches peuvent être regroupées ou séparées afin d'acheminer et faciliter la distribution du calcul (LIN et YANG, 2018), comme le présente la figure 2.8. On décrit chaque couche dans ce qui suit :

— **Terminaux connectés** : La couche inférieure qui déploie différents types de dispositifs de traitement intelligents, tels que des capteurs (locaux ou globaux), les machines connectées, les robots et robots mobiles, les interfaces homme-machine, ... Ils sont capables de communiquer avec la couche intermédiaire, pour assurer un traitement rapide des données.
— **La couche Edge** : Cette deuxième couche est très proche des nœuds terminaux. Il est composé de nœuds Edge, qui forment une passerelle de points d'accès entre les nœuds terminaux et les serveurs Fog pour acheminer les données et couvrir les appareils mobiles (ibid.).
— **La couche Fog** : La troisième couche regroupe différents nœuds de Fog tels que les serveurs, les routeurs, les passerelles et les commutateurs, qui sont capables d'extraire les connaissances des données puis de les transmettre aux serveurs Cloud computing (MUTLAG et al., 2019). La couche Fog diffère d'une utilisation à l'autre, elle peut être une combinaison de plusieurs sous-couches Fog ou inclure différents modes de calcul selon les besoins de l'application. Chaque sous-couche traite localement les données communiquées et si elle ne supporte pas la quantité de données ou le routage, les données seront prises en charge par une autre sous-couche Fog pour les calculer et les router rapidement.
— **Les Passerelles** : Ce sont des dispositifs qui relient les serveurs Fog au serveurs Cloud. Cette couche est souvent confondue ou attribuée à la couche Cloud.
— **La couche Cloud** : C'est la couche supérieure, qui est composée des serveurs d'autorité pouvant être mis à jour avec les dernières informations fournies par les serveurs Fog et permettant l'analyse des données et des prises de décisions (BONOMI et al., 2014).

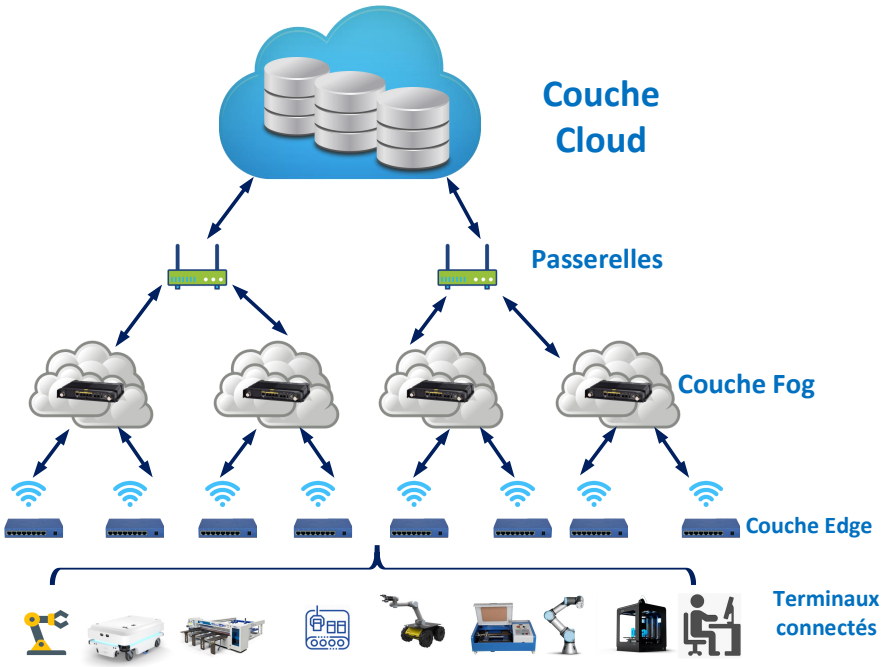
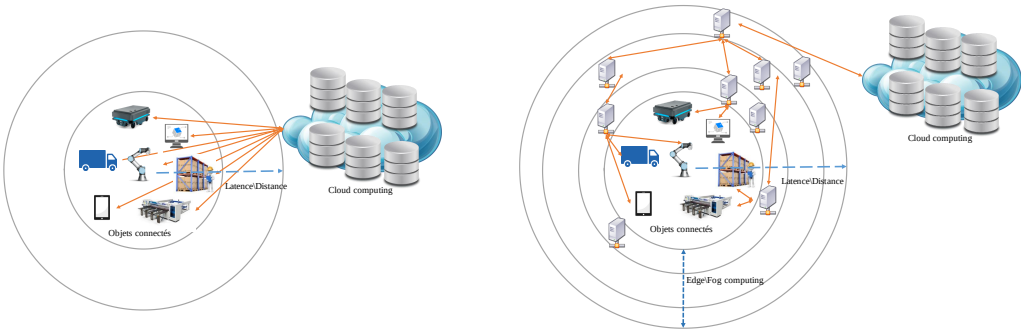


FIGURE 2.8 – Les couches de l'architecture basée sur le Fog Computing.



Approche centralisée du Cloud computing

Approche distribuée du Fpg computing

FIGURE 2.9 – La comparaison entre l'approche Cloud et Fog Computing.

2.4.2 Du Cloud Computing au Fog Computing

L'intégration du Cloud computing dans l'Industrie 4.0 offre plusieurs avantages expliqués auparavant dans la section 2.2.5 (LUONG et al., 2016). Malgré tous ces avantages, le Cloud Computing souffre de nombreuses limitations (voir par exemple (HASHIZUME et al., 2013, ISLAM, MORSHED et GOSWAMI, 2013, YEBOAH-BOATENG et ESSANDOH, 2014)). Cela devient un véritable problème, en particulier pour les applications industrielles sensibles à la latence et à la diffusion en continu et en temps réel de données. De toute évidence, lorsque les services distribués de l'IoT sont de plus en plus intégrés dans l'industrie, le paradigme actuel du Cloud computing ne peut plus satisfaire leurs besoins. La figure 2.9 présente une illustration des deux approches basée sur le Cloud computing seulement à gauche et basée sur le Cloud Computing et le Fog Computing à droite.

Sinon, plusieurs confusions peuvent être faites entre les services du Fog computing et du Cloud computing, en particulier dans les applications industrielles. Afin

de clarifier la position de chaque paradigme, les sections suivantes classent leurs caractéristiques et leur utilité en trois groupes : similarités, différences et complémentarités. Ces trois groupes sont abordés dans le contexte générale de l'implémentation du Fog et du Cloud computing.

— **Les similarités :** Le Cloud computing est connu pour sa capacité puissante et profonde d'analyse, de stockage et de communication. Le Fog fournit également ce type de services et ajoute de nouvelles applications aux utilisateurs finaux, ce qui distribue l'exploitation dynamique des ressources du Cloud computing. Par conséquent, cette nouvelle couche de Fog est une amélioration de la couche Cloud computing, qui offre des services locaux plus raffinés. Il fournit des flux de haute qualité vers les nœuds et il est particulièrement adapté aux applications à faible latence, au lieu de le faire dans un centre Cloud distant. Le Fog transmet la partie profonde de l'analyse au Cloud computing (**nistsite**). Ils orchestrent d'une façon autonome un grand nombre de dispositifs (FIRDHOUS, GHAZALI et HASSAN, 2014, VARGHESE et al., 2017).
De plus, le Cloud computing se caractérise par sa localisation au sein d'Internet, qui coordonne un grand nombre de réseaux, clients, fournisseurs, environnements et ressources hétérogènes. Le Fog partage la même caractéristique de supporter des ressources hétérogènes et des nœuds déployés dans une grande variété d'environnements avec des formes et des utilisations différentes (ZHOU et al., 2019), qui peuvent être déployés d'une manière hiérarchique (ROMAN, LOPEZ et MAMBO, 2018).

— **Les différences :** Contrairement à l'architecture du Cloud computing centralisée, le Fog computing est reconnu pour son architecture géographiquement distribuée, qui accélère la transmission des flux de données (STEINER et POLEDNA, 2016). Par rapport aux serveurs Cloud computing distants, ces nœuds Fog sont implémentés à proximité des utilisateurs finaux à la périphérie du réseau (ZHANG et al., 2019, VARSHNEY et SIMMHAN, 2017).
Les dispositifs du Fog peuvent fonctionner comme des serveurs autonomes couplés à des fonctionnalités de calcul et de stockage embarquées. En outre, les serveurs du Fog computing se distinguent des serveurs du Cloud computing non mobiles par leur prise en charge d'une mobilité accrue, leur disponibilité fiable (services continues même avec une connectivité intermittente) et la connaissance des lieux (BONOMI et al., 2012, LUAN et al., 2015).
Sur un autre plan, les réponses à faible latence sont un besoin pressant des applications de l'IIoT. Une faible latence de bout en bout, de l'ordre de quelques microsecondes est requise (WEINER et al., 2014). De plus cette latence doit être déterministe pour des applications de contrôle/commande de robots industriels par exemple. Le Fog est introduit pour réduire la latence élevée des communication avec le Cloud computing. Sa présence au bord du réseau permet d'offrir une haute qualité de services (en terme de latence) pour les utilisateurs finaux (LA et al., 2019).
Les environnements IoT et IIoT nécessitent ainsi une analyse en temps réel et une prise de décision rapide. Face à la forte limitation du Cloud computing, la faible latence du Fog permet d'interagir directement avec les utilisateurs et équipements pour fournir des ressources et des services, en respectant le délai réseau demandé (SARKAR et MISRA, 2016). Le Fog permet un traitement des données à proximité des utilisateurs finaux et prend en charge les applications sensibles au facteur temps.

— **Les complémentarités :** Partant du pilier de la sécurité, la confiance des clients est très importante pour les fournisseurs de services. L'absence de

sécurité des données limite l'efficacité des performances du Cloud computing (KANICKAM, JAYASIMMAN et JEBASEELI, 2017, LIU, FIELDSEND et MIN, 2017). En raison de sa proximité et de sa localisation au sein de l'usine, le Fog computing comble cette lacune en permettant aux fournisseurs, aux appareils et aux utilisateurs de faire face aux défis de sécurité et de protection de la vie privée et des données de l'entreprise. De nombreuses recherches s'intéressent à cette problématique (ALHAZMI et ALOUFI, 2019).

Lors de la conception d'un réseau, il est primordial de le rendre évolutif, ce qui signifie que le réseau peut supporter une augmentation de la quantité de nœuds sans changer son architecture. Ce type d'évolutivité atteint ses limites avec l'utilisation croissante de l'IoT. L'architecture de Fog computing est capable d'améliorer et de compléter l'évolutivité de l'architecture Cloud computing sans grands changements dans l'architecture réseau (VAQUERO et RODERO-MERINO, 2014, RAMALHO, 2016). Cette capacité est due à la possibilité d'ajouter progressivement des nœuds Fog en fonction des besoins du réseau.

En termes de bande passante du réseau, le Cloud computing a quelques contraintes surtout pour les transmissions sur de longue distance. Le Fog est introduit pour surmonter ces défis de bande passante du réseau (QI et TAO, 2019), car il peut filtrer les données avant de les envoyer vers les serveurs du Cloud computing pour une analyse plus profonde notamment avec les besoins croissants de capacités de calcul CPU et GPU des approches d'IA basées sur du deep learning.

Le Fog computing étend également le nombre de dispositifs déployés dans un environnement hétérogène (noeuds géographiquement répartis et assurant une large couverture), complète le rôle du Cloud computing et orchestre de manière autonome un grand nombre de dispositifs (CHIANG et ZHANG, 2016, FIRDHOUS, GHAZALI et HASSAN, 2014, VARGHESE et al., 2017).

Le Cloud computing permet au noyau d'accéder aux applications qui tolèrent les délais et la gigue. D'autre part, le Fog computing étend ce type d'applications pour fournir aux CPS des applications sensibles au temps (BONOMI et al., 2012). L'accès peut être géré du cœur au bord du réseau, ce qui garantit la disponibilité des ressources (CHIANG et ZHANG, 2016). Les services peuvent être fournis d'une manière autonome et ininterrompue (et même sans connectivité), avec la présence de Fog à côté du Cloud computing.

En conséquence, nous pouvons remarquer que le Fog computing est une solution pour les limitations du Cloud computing devant les exigences des applications basées sur l'IoT. Les différents points discutés sur la comparaison du Cloud computing et du Fog computing sont résumés dans le tableau 2.3.

En conclusion, il devient clair que les caractéristiques du Fog computing sont plus complètes que celles du Cloud computing seul. Néanmoins, le Fog computing ne peut pas supporter toutes les fonctionnalités du Cloud computing. Nous pouvons donc parler ici d'améliorer l'architecture Cloud computing et non de la remplacer totalement. Une sorte d'interaction sera présente entre ces deux modèles. Le Fog Computing complétera le Cloud computing, tout en y ajoutant ses propres avantages au bord du réseau.

Parlant de l'insuffisance du CM dans le contexte d'Industrie 4.0, ce paragraphe spécifiait les contributions du Fog Computing pour améliorer les performances du Cloud computing. La prochaine section traitera de la possibilité d'intégrer le Fog comme étant une nouvelle composante de l'Industrie 4.0 en tenant compte des exigences et des défis envisagés.

TABLE 2.3 – Comparaison du Cloud computing et du Fog computing.

	Caractéristiques	Cloud computing	Fog computing
Similarité	Calcul, stockage et routage	Analyses profondes. Calcul, stockage	Calcul local, Stockage et analyses
	Hétérogénéité	Interagit avec des sources de données et des environnements hétérogènes	Prise en charge de l'hétérogénéité des ressources, des environnements et des nœuds.
Différence	Localisation	Aucun rapport avec l'emplacement de l'utilisateur final	A proximité des utilisateurs finaux.
	Architecture	Centralisée	Distribuée
	Mobilité	N/A	Supporte la mobilité
	Latence	Élevé	Faible latence
	Aspect temps réel	varie de quelques secondes à quelques minutes	$<$ 100ms
Complémentarité	Bande passante	La bande passante n'est pas suffisante à cause de la longue distance et l'augmentation de la quantité de données	Surmonter les défis de bande passante du réseau (filtrer les données avant de les envoyer vers le Cloud computing)
	Disponibilité et connectivité	La connectivité est nécessaire pour assurer la disponibilité des services.	Assurer de façon autonome des services continus même sans connexion Internet.
	Sécurité	Manque important de sécurité et de confidentialité	Comble cette lacune en définissant des politiques sur les nœuds du Fog.
	Nombre de nœuds	Des dizaines/centaines de millions	des dizaines de milliards répartis géographiquement
	Support d'application	Fournit des applications qui tolèrent les délais et la gigue	Fournit des applications basées sur CPS et sensibles au temps
	Accès aux services	À travers le cœur de réseau	Étendu jusqu'au bord

2.5 Conclusion

La majorité des principes et des caractéristiques de l'Industrie 4.0 ont été définis en fonction des technologies disponibles au moment de l'introduction de cette 4me révolution en 2010. Cette révolution repose sur différents composants (IoT, IoS, CPS, Cloud computing, Big Data et capteurs) qui doivent être combinés pour répondre aux exigences de l'Industrie 4.0.

De nombreuses applications axées sur l'adoption de la stratégie de l'Industrie 4.0 commencent à voir le jour, telles que le contrôle de la consommation d'énergie dans les industries, la surveillance des ateliers de production et même l'intégration au secteur public.

Nous avons observé que cette révolution nécessite l'ajout de nouvelles technologies pour réussir son adoption, en particulier les technologies apparues récemment. Dans l'avenir, plusieurs autres composantes, liées au concept de l'Industrie 4.0, pourront être intégrées au fur et à mesure. Le fog computing est l'un des composants importants à ajouter, compte tenu de son rôle important dans l'amélioration des performances du Cloud computing et la couverture de ses limites.

Ce chapitre a exposé le concept du Fog computing en présentant les définitions proposées dans la littérature ainsi que ses caractéristiques et son architecture (initiale et évoluée). Les applications qui implémentent le Fog computing sont cités. Dans ce chapitre nous avons établi une comparaison entre le Cloud computing et le Fog computing en se basant sur leurs caractéristiques.

Dans le chapitre 2, nous étudions les raisons derrière le retard d'intégration du Fog computing parmi les composants de l'Industrie 4.0. Nous présentons ainsi quelques travaux d'adoption du Fog dans le domaine industriel. En se basant sur

ces applications, nous identifions les applications industrielles les plus intéressantes qui peuvent intégrer le Fog computing.

Chapitre 3

Le Fog Computing dans le contexte de l'Industrie 4.0

3.1 Introduction

Récemment, les industries ont convergé vers l'adoption de l'Industrie 4.0 pour continuer à répondre aux demandes variables du marché. La majorité des activités de production (la conception, la planification, la fabrication, le contrôle, la supervision, la maintenance, la gestion, la logistique, etc.) sont concernées par la migration vers l'Industrie 4.0, afin de garder la mise à jour continue de leurs applications liées à l'IoT et au Cloud computing.

Au cours de ces deux dernières décennies, de nombreuses technologies de l'information et de la communication ont été intégrées avec succès dans le contexte industriel. Cependant, une telle intégration peut être insuffisante en termes de traitement des données et de réponse en temps réel, comme par exemple le Cloud computing.

L'adoption d'une architecture basée sur le Fog computing sera la prochaine étape à franchir afin de répondre aux exigences de l'Industrie 4.0 et surmonter les limitations du Cloud. Malgré le succès des applications du Fog computing dans des domaines variés, l'exploitation industrielle du Fog computing demeure modeste. Par contre, de nombreuses applications prometteuses apparaissent en permanence et font de cette direction un axe de recherche riche à explorer.

Les raisons de cette adoption tardive de la mise en œuvre du Fog computing dans le domaine industriel se heurtent à plusieurs défis de natures différentes qui vont être étudiés et identifiés dans ce chapitre.

Le prochaine section (Section 3.2) se concentre sur l'étude de l'adoption du Fog computing dans les applications de l'Industrie 4.0 . Par la suite, les défis d'une architecture industrielle basée sur le Fog vont être relevés et expliqués dans la section (Section 3.3) pour marquer un saut important dans l'adoption d'une version industrielle du Fog computing que nous avons choisi de nommer "le *Fog Manufacturing*".

3.2 Étude de l'intégration du Fog Computing dans le contexte de l'Industrie 4.0

Comme présenté dans l'état de l'art, le paradigme de l'Industrie 4.0 vise à favoriser et à promouvoir les cas d'études axés sur l'innovation dans les industries (LIN et al., 2015). Les applications industrielles sont généralement fondées sur un modèle centralisé de traitement des données. Les données générées par les appareils industriels (qui peuvent être géographiquement répartis sur une vaste zone)

sont acheminées par l'infrastructure du réseau vers une installation de stockage central. Il est facile de comprendre que, dans le cadre du modèle centralisé (basé sur le Cloud Computing), l'infrastructure de communication de l'IoT peut devenir de plus en plus difficile à mettre en place, car le nombre des objets connectés augmente (BRUNER, 2013). Dans cette direction, le Fog computing a été proposé comme une solution à cette centralisation des données et à d'autres problèmes liés à l'Industrie 4.0.

3.2.1 Le Fog Computing répond-il aux exigences de l'Industrie 4.0 ?

D'après ce que nous avons étudié dans la section 2.4, le Fog computing peut être considéré comme une solution officiellement utilisée afin de surmonter les limitations du Cloud computing. Dans de nombreux domaines le Fog computing a été utilisé afin de renforcer les performances et étendre les caractéristiques du Cloud computing.

Les CPS de l'Industrie 4.0 génèrent une grande quantité de données issues de nombreux dispositifs équipés de capteurs. Cette quantité d'information exige un traitement rapide afin de générer des instructions pour les actionneurs et équipements pilotables avec une latence acceptable. Le Fog computing est caractérisé par la diminution de la latence d'échange de données et la rapidité de réponse (aspect temps réel) ce qui répond fortement à la demande de l'Industrie 4.0 en termes de réponse rapide et de diminution des délais de transmission.

Les données qui deviennent de plus en plus complexes consomment et dépassent largement les ressources limitées des systèmes. Le transfert de ce calcul complexe aux serveurs locaux du Fog computing aide à alléger la charge au niveau de la couche des terminaux. Par conséquent, le Fog computing aide à économiser l'énergie consommée lors de calculs réalisés au sein des CPS. Lors de cette tâche de traitement, effectuer un filtrage et une structuration des données par les Noeuds Fog (Fog Nodes - FNs) est possible, ce qui évite de surcharger le réseau par des données inutiles.

L'Industrie 4.0 commence à intégrer de nombreux nouveaux systèmes incompatibles avec l'architecture existante. Le Fog computing peut alors intervenir afin d'interfacer les dispositifs incompatibles (MATT, 2018).

Le Fog computing a un effet positif sur la QoS et la fiabilité des données échangées. De plus, le grand nombre de CPS hétérogènes et connectés (à l'aide de l'IoT) de l'Industrie 4.0 peut être largement couvert par les serveurs distribués du Fog computing. Le Fog Computing, qui constitue une solution de calcul distribué basée sur l'IoT, profite aussi de l'interopérabilité liée à l'IoT pour "comprendre" les différents systèmes d'exploitation présents au sein d'un atelier de l'Industrie 4.0.

Le CPS regroupe divers types de machines et de robots, certains d'entre eux pouvant être mobiles tels que les AGVs (Autonomous Guided Vehicule). Le Fog computing supporte la mobilité des robots ainsi que des opérateurs qui réalisent des tâches au sein des différents départements industriels. De même, en comparant le reste des exigences de l'Industrie 4.0 et les caractéristiques du Fog (sécurité, optimisation, calcul et analyse de données, modularité, etc,..), le Fog computing présente un potentiel prometteur.

De nombreux travaux ont abordé la question de la possibilité d'intégration du Fog Manufacturing dans les applications de l'Industrie 4.0. Par exemple, (PIZOŃ et LIPSKI, 2016) ont étudié la possibilité d'appliquer le Fog Computing dans le domaine industriel et les défis qui limitent une telle implémentation. Leur conclusion était que

le Fog computing satisfait les exigences de l'Industrie 4.0 grâce à ses caractéristiques. Les travaux de (GEORGAKOPOULOS et al., 2016) s'orientent dans la même direction.

3.2.2 Les domaines d'application existants du Fog Manufacturing

Cette section présente les cas d'étude qui étudient la possibilité d'intégrer ou qui intègrent la notion du Fog Manufacturing dans le domaine de l'Industrie 4.0. Nous avons classé les cas d'étude trouvés par domaine d'application.

Maintenance prédictive

La maintenance prédictive est une maintenance conditionnelle qui consiste à prédire la date d'une action de maintenance en se basant sur un seuil prédéfini. Cette prédiction s'effectue selon l'état de dégradation théorique des composants du système.

Le domaine de la maintenance dans l'industrie 4.0 a connu une évolution en terme de méthodes et de principes, nous parlons désormais de e-maintenance et des maintenances préventive, prédictive et proactive. Tous ces types sont basés sur les données collectées et la prise de décisions automatiques ou semi-automatiques en temps réel. Le Fog computing peut jouer un rôle important dans des applications liées à la maintenance. Par exemple, la maintenance prédictive a besoin du Fog principalement pour sa capacité à gérer de grandes quantités de données (filtrage, ingénierie de caractéristiques), pour réduire le temps de traitement de l'exécution des modèles prédictifs (c'est-à-dire recevoir les paramètres mis à jour du nuage et exécuter les états suivants du modèle) et sa capacité à traiter des sources de données hétérogènes grâce à son aspect distribué. Une implémentation du Fog computing dans le domaine industriel est proposée dans (GAZIS et al., 2015). Elle introduit une plateforme d'opérations adaptative (AOP) pour fournir une solution de maintenance prédictive basée sur le Fog. L'AOP assure la maintenance prédictive à l'aide de la combinaison d'un modèle développé (basé sur les probabilité d'échec) et le modèle prédictif extrait des données réelles de productions. Le but de ce cas d'étude est de détecter les anomalies potentielles sur un site de production. Le Fog Computing est utilisé pour fournir la communication, distribuer le calcul, gérer la grande quantité de données, supporter l'hétérogénéité des formats de données. Le scénario décrit est composé de N capteurs qui surveillent l'état des machines et envoient les données à travers des routeurs vers un serveur afin de former une base de données et détecter les anomalies potentielles.

Deux simulations sont faites afin de comparer les performances d'une architecture centralisée (un seul routeur) et l'architecture Fog (routeurs Cisco ISR 819/passerelles) supportant le Data in Motion DMo. Le DMo fait référence à un flux de données se déplaçant à travers tout type de réseau. Elles peuvent être considérées comme l'opposé des données au repos, car elles représentent des données qui sont transférées ou déplacées. Ce cas d'utilisation a montré le potentiel prometteur de l'adoption du Fog dans le contexte de l'Industrie 4.0 mais il est nécessaire de prouver l'efficacité du Fog computing en termes d'autres métriques, très importantes pour l'Industrie 4.0, comme par exemple la consommation d'énergie et le temps de calcul (qui doit être en temps réel).

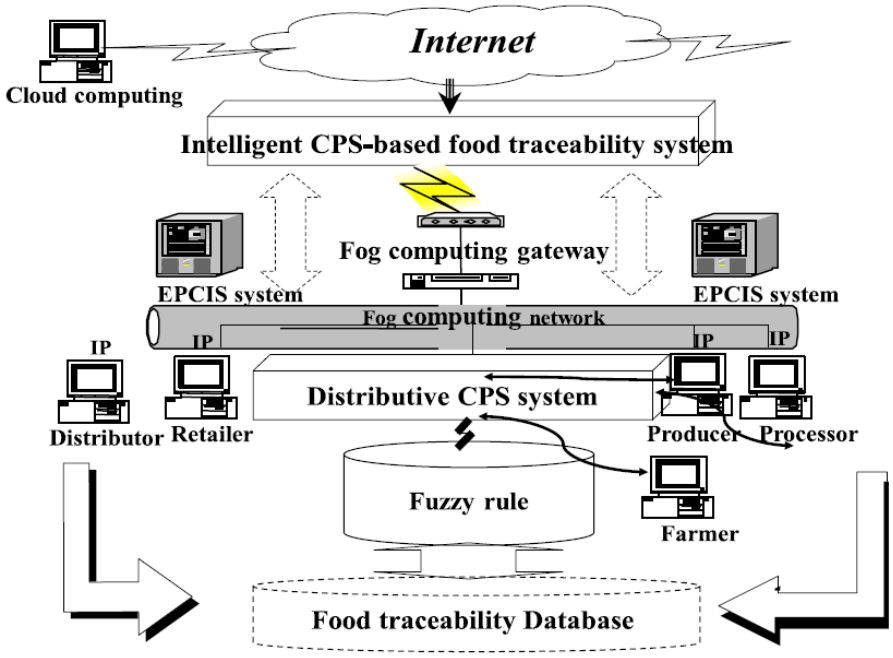


FIGURE 3.1 – L'architecture du système de traçabilité basé sur le Fog computing (CHEN, 2017).

Traçabilité

La traçabilité est appliquée pour gérer la localisation et l'historique des produits ou des systèmes de productions en se basant sur l'identification de ces derniers. La traçabilité doit ainsi permettre de pouvoir connaître la localisation à un instant t d'un produit ou d'un groupe de produits en se basant sur l'identifiant à n'importe quelle phase de production. Il permet également de reconstituer l'historique et de suivre le parcours du produit (BENATIA et al., 2018). Un système de traçabilité alimentaire intelligent est proposé par (CHEN, 2017). C'est le premier travail qui intègre le Fog computing dans une application de traçabilité alimentaire dans le contexte de l'Industrie 4.0.

Il est basé sur une architecture de Fog computing, présentée par la figure 3.1, pour générer des flux de valeur intelligents (VS - Value Stream). Un CPS agro-alimentaire est introduit afin de minimiser les produits de mauvaise qualité dans la chaîne d'approvisionnement alimentaire. Le Cloud Computing est utilisé pour les applications CPS mais il est renforcé par le Fog afin de supporter la mobilité, le placement aware retenu et la faible latence.

L'objectif est d'évaluer les événements les plus traçables pour le suivi, et d'avoir un CPS auto-configurable pour effectuer diverses tâches. Le Fog Computing est utilisé dans cette proposition pour assurer la communication et la distribution du calcul du flux de valeur au bord du réseau. L'article présente un cas d'étude qui intègre le CPS à l'architecture, analyse les machines virtuelles et assure un mécanisme distribué de traçabilité du VS. La dernière étape est réalisée à l'aide du Fog Computing qui identifie/capture/échange en temps réel les données de traçabilité. Une simulation a montré une amélioration de l'efficacité de la communication, de l'intégration des

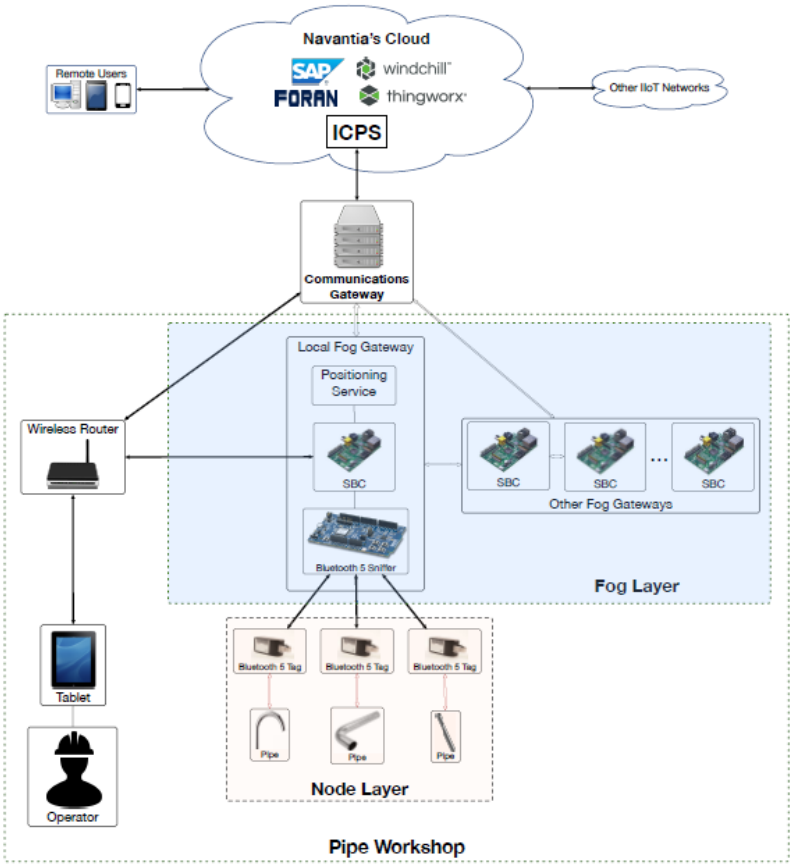


FIGURE 3.2 – L'architecture du système d'IAR basé sur le Fog computing (FRAGA-LAMAS et al., 2019).

ressources et de la gestion des données en temps réel grâce au Fog computing. Cependant, cette application, basée sur le Fog, reste limitée face à son potentiel prometteur dans le domaine industriel, notamment pour filtrer les données et pour fournir des réponses en temps réel.

Dans (FRAGA-LAMAS et al., 2018) et (FRAGA-LAMAS et al., 2019), les auteurs proposent l'intégration du Fog computing dans le contexte de l'Industrie 4.0 pour le processus de mise en œuvre de la réalité augmentée industrielle (Industrial Augmented Reality). Cette intégration a pour but de relever le défi de traçabilité des tuyaux de la construction navale.

Une architecture de communication basée sur le Fog computing d'un système IAR (Industrial Augmented Reality) a été proposée, elle est illustrée par la figure 3.2. Elle permet de prendre en charge des applications d'IAR physiquement distribuées, à faible latence tout en respectant la QoS, réduisant ainsi le trafic réseau et la charge de calcul des systèmes de calculs traditionnels. Cette architecture assure le fonctionnement des applications de traçabilité qui servent à répondre rapidement aux événements classés critiques.

Une simulation de cette architecture a été faite en modélisant le scénario d'un atelier de tuyauterie en 3D en tenant compte du nombre de tuyaux et des principales zones de travail, y compris les machines et la main-d'œuvre quotidienne. Les résultats obtenus ont montré une bonne précision globale, ce qui permet de justifier

l'utilisation du Fog computing pour un tel environnement de l'Industrie 4.0.

Un deuxième cas d'utilisation est proposé dans (BLANCO-NOVOA et al., 2018) afin de mettre en œuvre les technologies de Fog computing et de l'IAR (Intelligent Augmented Reality) dans un chantier naval de l'Industrie 4.0. Le Fog computing sert à réduire les latences des réponses afin de fournir une interaction réaliste en réalité augmentée (Augmented Reality), de distribuer le calcul et les ressources et de rendre le système flexible et extensible.

Les auteurs ont utilisé le Fog computing comme une technologie intermédiaire, dont les performances ont été évaluées à travers les preuves de concept d'IAR. Cependant, même si la validation du cas d'étude prouve le potentiel du Fog computing, il manque la spécification des apports du Fog Manufacturing dans ce cas.

Monitoring

Le monitoring est le suivi des systèmes afin de faciliter la détection, la prédiction et le diagnostic des défauts. Le monitoring est souvent appliqué en industrie afin de faciliter la gestion de la production et limiter les pannes et les retards. Ce domaine d'application gère un grand nombre de données qui concernent les systèmes de production, des produits et des entrepôts de stockage. Le Fog computing peut jouer un rôle important pour faciliter la communication et la prise de décision en temps réel dans ce genre d'applications.

(DE BRITO et al., 2016) proposent un modèle regroupant les dispositifs programmables en déployant des applications conteneurisées qui s'exécutent en périphérie du réseau et communiquent avec les nœuds Fog. Ils ont comme objectif de rendre l'environnement industriel flexible en effectuant l'orchestration des services de monitoring et contrôle du CPS au bord du réseau. Ils doivent aussi permettre au CPS des communications "machine à machine" (M2M) flexibles et autonomes.

Les auteurs ont conclu que l'absence de l'aspect communication M2M autonome entre les noeuds Fog est manquant, ce qui ne répond pas aux exigences de l'Industrie 4.0 que nous avons identifiées : QoS, distribution, faible latence, réponse en millisecondes.

Le cas d'étude tend à assurer une plateforme de gestion et de surveillance distribuée et optimale, ce qui fournit des services de localisation en temps réel ou quasi réel, à l'intérieur de l'atelier. Les indicateurs de performance à mesurer sont : La latence, l'intégration rapide (scalability), la robustesse contre les interférences. Les auteurs n'ont pas obtenu de résultats réalistes. Cependant, la solution proposée est prometteuse, surtout avec la prise en compte de certains aspects comme la sécurité et l'interopérabilité.

(MUKHERJEE et al., 2017a) ont souligné l'urgence de l'intégration du Fog computing dans les applications industrielles afin de réduire le flux de données échangées et de résoudre le problème du calcul et du stockage des données. Ils ont ainsi proposé un futur cas d'étude qui adopte le Fog computing pour faciliter le monitoring au sein des industries. Les auteurs ont mentionné la future mise en place d'un protocole de traitement de la charge inutile. Ils ont défini quelques défis qui limitent une telle proposition : distribution géographique, grand nombre des systèmes connectés, hétérogénéité et latence.

Diagnostic et pronostic

Le pronostic et le diagnostic sont de plus en plus utilisés dans les applications industrielles, en particulier pour la maintenance prédictive. Le diagnostic et le pronostic sont fortement liés : le diagnostic correspond à l'identification et à l'étude de l'ensemble des machines dont on peut observer des défauts ou des dysfonctionnements, le pronostic prévoit l'état futur des machines à partir d'un modèle de dégradation du système concerné. Ces deux applications aident à déterminer l'état des composants des machines et d'anticiper leur durée de vie. Ces applications sont basées sur les modèles d'analyse de données ou des méthodes d'apprentissage basés sur l'intelligence artificielle. Le Fog computing déployé dans ces applications peut soulager les serveurs de calculs et fournir des résultats en temps réel.

Un modèle basé sur le Fog pour la surveillance de l'état de santé des structures de production et des processus pilotés par les données dans les CPPS est introduit par (WU et al., 2017). Ce système basé sur le CPS et le Fog computing, permet la distribution de la détection, la surveillance et le calcul évolutif pour le diagnostic et le pronostic en temps réel.

Selon les auteurs, les systèmes de surveillance existants ne sont ni capables de gérer des grandes quantités de données en temps réel, ni capables de couvrir des zones à grande échelle. Le rôle du Fog dans ce cas est d'assurer la distribution de la collecte et de l'analyse des données en temps réel et d'assurer l'échange entre les différentes parties. Afin de permettre la collecte d'une grande quantité de données en temps réel, une architecture basée sur le Fog computing dans le contexte de l'Industrie 4.0 est conçue. Elle permet également de surveiller l'état des machines et des processus. La collecte des données des machines se fait à travers une passerelle en temps réel, par la suite les données sont transmises aux FNs ensuite les serveurs distants du Cloud. La prochaine étape est de constituer des modèles de diagnostic et pronostic à partir des données collectées et stockées, qui seront par la suite appliqués sur les base de données du Cloud.

Les auteurs ont défini la latence par le délai entre les demandes des opérations et les réponses. Cette latence peut donner une idée sur l'aspect temps réel de la communication et le nombre de message transmis par unité du temps. Afin d'améliorer l'interopérabilité le protocole industriel sémantique MTConnect est implémenté. Une expérimentation est faite à l'aide de plus de 50 capteurs sur les pompes de machines à commande numérique. Des futurs travaux sont envisagés par les auteurs pour valider cette proposition .

Malgré la description détaillée de ce cas d'étude, il manque des résultats réels qui prouvent l'apport et la valeur ajoutée du Fog computing dans ce contexte, en termes de latences d'exécution des modèles de pronostic au niveau des FNs.

Consommation d'énergie

Les industries sont toujours connues pour leur consommation importante en énergie ce qui augmente les coûts de production et de stockage. Le Fog computing étant une couche intermédiaire, il peut forcément diminuer la consommation d'énergie en limitant les échanges directs avec la couche Cloud et en traitant localement les demandes sur les serveurs du Fog.

(PERALTA et al., 2017) proposent une intégration du protocole MQTT (MQ Telemetry Transport) dans la couche Fog Computing dans le contexte de l'Industrie 4.0. Le MQTT est un protocole de messagerie basé sur le protocole TCP/IPSINGH et al., 2015. Cette intégration du MQTT aide à prédire les mesures de données parvenant

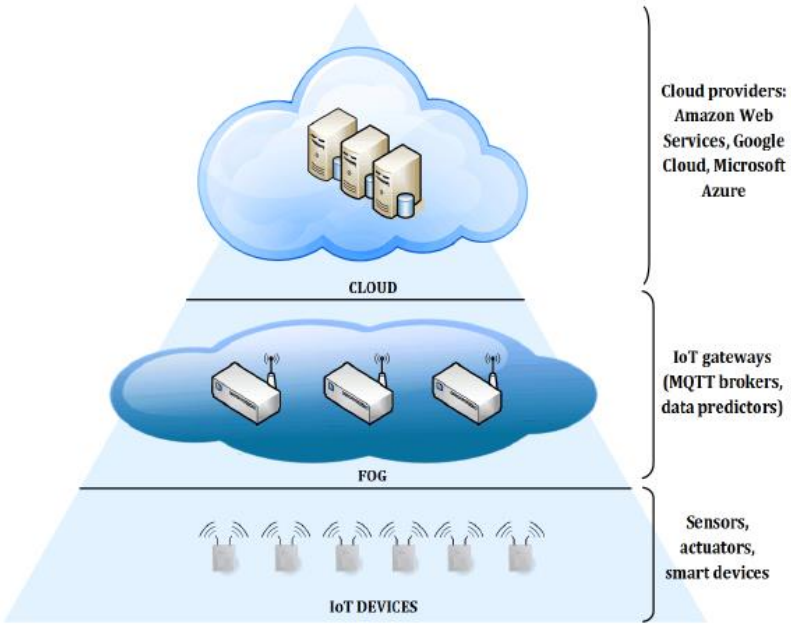


FIGURE 3.3 – L'architecture du système basé sur le Fog computing et le protocole MQTT (PERALTA et al., 2017).

des capteurs, contrôler les couches comme étant une passerelle, fournir la capacité de traitement des données au niveau de la couche Fog et minimiser la latence. La figure 3.3 présente l'architecture déployée pour ces travaux.

Le rôle du Fog Computing dans ce cas est d'assurer la communication rapide et efficace au bord du réseau entre les dispositifs MQTT et les systèmes connectés à l'IoT. Cette approche est validée par une analyse de la consommation d'énergie du réseau des capteurs sans fils et une comparaison avec l'architecture MQTT traditionnelle. Les brokers MQTT sont installés au niveau de la couche Fog Computing et un protocole de communication est ensuite défini.

Les résultats ont montré l'efficacité d'une telle proposition avec la diminution de la consommation d'énergie, des coûts ainsi que l'augmentation des indicateurs de QoS, principalement lors de l'échange de longues requêtes.

Logistique

La logistique est un domaine très important dans l'industrie, elle assure le lien entre les fournisseurs, les fabricants et les clients. La logistique est aussi utilisée pour assurer le lien entre les différents organes dans un atelier de production. Plusieurs caractéristiques du Fog computing peuvent être exploitées pour l'amélioration du fonctionnement de la chaîne logistique interne ou externe telles que la mobilité, la distribution et la réduction de latence.

(LIN et YANG, 2018) ont étudié la possibilité d'intégration du Fog computing dans le contexte de l'Industrie 4.0. Les auteurs veulent adopter une architecture de logistique basée sur le Fog computing. Le rôle du Fog Manufacturing dans ce cas est d'assurer la distribution du calcul et, à travers la caractéristique de mobilité, la couverture de tout l'espace de stockage correspondant aux zones de déplacement des robots AGV.

Après la définition d'une architecture composée par des terminaux, nœuds Edge (ENs), FNs et de passerelles, les auteurs ont modélisé cette architecture afin d'optimiser le coût d'installation. Ce coût dépend principalement des nœuds et des liaisons utilisés. Les contraintes de ce modèle sont définies pour respecter les exigences de l'Industrie 4.0 et du Fog computing telles que la latence, le nombre des liens, la couverture, etc... Une validation de ce modèle à l'aide d'un algorithme génétique a fourni une solution optimale. Ce cas d'étude est bien élaboré, par contre une validation réelle de l'apport du Fog computing reste manquante.

Planification ou ordonnancement

La planification ou l'ordonnancement est le processus d'organisation, de contrôle et d'optimisation du travail et des charges de travail dans un processus de production. La planification est principalement utilisée pour assurer l'allocation des ressources et la planification des processus de production. Elle est considérée un outil important pour minimiser le temps et les coûts de production, en indiquant à une installation de production quand fabriquer, avec quel personnel et sur quel équipement. Le Fog computing peut s'intégrer dans cette application afin de diminuer le temps de traitement et augmenter la précision des décisions.

Suite au manque concernant des travaux sur les applications d'ordonnancement en temps réel, les travaux de (BRIK et al., 2019) proposent un outil de surveillance des systèmes dans le contexte de l'Industrie 4.0, en se basant sur une architecture Fog computing. L'utilisation du Fog computing dans ce cas a pour objectif de diminuer la latence et de fournir des réponses en temps réel tout en traitant les transmissions localement. L'outil du Machine Learning Random Forest se concentre sur les perturbations liées aux ressources et il se base sur un modèle de prédiction de la localisation des ressources. Les résultats ont montré une efficacité en terme de précision et de détection des perturbations du système en temps réel en les comparant aux autres algorithmes du Machine Learning (Regression logistique, Naive Bayes et Arbre de décision).

3.2.3 Analyse et synthèse

Nous avons trouvé d'autres cas d'étude dans la littérature, comme par exemple (YU et al., 2018) qui propose un protocole basé sur le Fog computing, qui décrypte, contrôle l'accès et supprime les données industrielles. Un autre cas d'application du Fog computing pour la gestion des ressources dans un environnement industriel est présenté par (VERBA et al., 2019). Selon les auteurs, les méthodes d'ordonnancement de déploiement des ressources, dans le contexte de l'Industrie 4.0, basées sur le Cloud computing possèdent de grandes inexactitudes entre les simulations et le réel. Les modèles existants entraînent des délais et des traitements supplémentaires. Les auteurs ont donc optimisés ces modèles en utilisant le Fog computing. Un cas d'étude de ce modèle d'optimisation est validé à l'aide des tests dans un environnement Industrie 4.0 réel. Les résultats ont prouvé l'efficacité de Fog computing (90% de précision en termes des délais et de charge de fonctionnement des modèles) à travers une comparaison aux Cloud et Edge computing.

Nous pouvons également citer (CHEKIRED, KHOUKHI et MOUFTAH, 2018), qui ont étudié l'intégration de l'IoT dans l'Industrie 4.0 afin de fournir une gestion des données pour des grandes usines en temps réel. Ils ont proposé un déploiement hiérarchique des serveurs Fog en utilisant des modèles d'analyses probabilistes selon la

priorité de la demande et un algorithme d'affectation de la charge de travail. En utilisant des données industrielles réalistes du groupe Bosch, les avantages de l'architecture proposée par rapport à celle horizontale sont remarquables en terme de performance. De nombreux travaux sur l'intégration du Fog computing dans l'Industrie 4.0 ont été trouvés comme (LEE, SAAD et BENNIS, 2018, SHI et al., 2018, MAHMUD et al., 2019a, WANG et al., 2019b), ou encore (CHEN et al., 2019). A travers l'exploration des différents domaines d'application du Fog Manufacturing nous pouvons attester que le Fog computing trouve sa place dans le domaine industriel grâce à ces différentes caractéristiques telles que la réduction de la latence, la distribution du calcul et la possibilité de traitement de grandes quantités de données en temps réel. Nous avons remarqué que d'autres caractéristiques intéressantes sont moins exploitées telles que la mobilité et l'hétérogénéité. Nous pensons que cela arrivera très rapidement grâce à l'émergence de nouveaux capteurs et de nouveaux besoins tels que la localisation Indoor des opérateurs ou la prise en compte de l'aspect humain dans la prise des décisions (Interactions homme machine). Le Tableau 3.1 résume les applications ou propositions trouvées dans la littérature et montre qu'il y a une évolution chronologique dans les applications et le nombre de travaux cités. Cela peut être expliqué par l'apparition récente du Fog computing (2014) et l'évolution des besoins industriels. La première est apparue en 2015 et leur nombre ne cesse de croître avec deux travaux en 2015 et 2016, cinq en 2017, six en 2018 et six en début de l'année 2019. Nous avons sélectionné les plus pertinents ce qui n'empêche pas que d'autres travaux existent.

Ces applications sont exploitées dans plusieurs domaines des systèmes de production, telles que la supervision, la gestion de l'énergie, la communication, la maintenance, la surveillance et la logistique. Ainsi, nous avons constaté que chaque cas d'usage présenté dans la littérature exploite des caractéristiques spécifiques du Fog. Par exemple la maintenance prédictive utilise la puissance de calcul, le big Data, la distribution et l'hétérogénéité, alors que le monitoring n'utilise ni le Big Data ni l'hétérogénéité, mais il utilise la mobilité et le temps réel. Nous pouvons remarquer également que des applications dans le même domaine peuvent avoir besoin de caractéristiques différentes.

L'observation des colonnes du tableau, nous permet de remarquer que les caractéristiques les plus importantes du Fog computing qui intéressent les industriels sont la puissance de calcul (capacité de traitement) (100% des cas), le traitement de grandes quantités de données (94.7% des cas), l'aspect distribué et le traitement en temps réel est également très utilisé avec 78.9% d'utilisation dans les cas identifiés. En revanche, la mobilité et l'hétérogénéité des systèmes dans une architecture basée sur le Fog, très appréciée dans les applications de télécommunication, est moins explorée dans le contexte industriel (42.10% des cas). Afin de comprendre le manque d'utilisation du Fog computing, la section suivante discute les défis qui limitent l'exploitation du Fog Manufacuring et explique leurs effets mutuels et leurs interactions.

3.3 Les défis limitant l'intégration du Fog computing

En nous basant sur l'état de l'art, nous avons pu identifier plusieurs défis limitant l'implémentation du Fog computing dans des applications industrielles. Certains d'entre eux sont généraux et d'autres sont liés au domaine industriel, comme le montre la figure 3.4.

TABLE 3.1 – Les cas d'implémentation du Fog computing dans le contexte d'Industrie 4.0.

Année	Référence	Domaine	Temps réel	Puissance du calcul	Quantité de donnée	Mobilité	Distribution	Hétérogénéité
2015	GAZIS et al., 2015	Maintenance prédictive	-	✓	✓	-	✓	✓
2016	DE BRITO et al., 2016	Monitoring	✓	✓	✓	-	-	✓
2017	CHEN, 2017	Traçabilité	✓	✓	✓	-	✓	✓
2017	WU et al., 2017	Diagnostic & pronostic	✓	✓	✓	-	✓	-
2017	PERALTA et al., 2017	Consommation d'énergie	✓	✓	✓	-	-	-
2017	BACCARELLI et al., 2017	Consommation d'énergie	-	✓	✓	✓	✓	✓
2017	MUKHERJEE et al., 2017a	Monitoring	✓	✓	-	✓	✓	-
2018	SEITZ, BUCHINGER et BRUEGGE, 2018	Disponibilité	✓	✓	✓	-	✓	✓
2018	BRITO et al., 2018	Planification et ordonnancement	✓	✓	✓	-	✓	✓
2018	YU et al., 2018	Sécurité	✓	✓	✓	-	✓	-
2018	O'DONOVAN et al., 2018	Logistique	✓	✓	✓	✓	✓	✓
2018	SHI et al., 2018	Ordonnancement	✓	✓	✓	✓	✓	-
2018	LEE, SAAD et BENNIS, 2018	Planification et ordonnancement	-	✓	✓	✓	✓	-
2019	BLANCO-NOVOA et al., 2018 & FRAGA-LAMAS et al., 2018& FRAGA-LAMAS et al., 2019	Traçabilité	✓	✓	✓	✓	✓	-
2019	VERBA et al., 2019	Planification et ordonnancement	✓	✓	✓	-	✓	-
2019	BRIK et al., 2019	Planification et ordonnancement	✓	✓	✓	-	-	-
2019	MAHMUD et al., 2019a	Placement des applications en fonction du contexte	✓	✓	✓	✓	✓	✓
2019	WANG et al., 2019b	Ordonnancement	✓	✓	✓	-	✓	-
2019	CHEN et al., 2019	Consommation d'énergie	-	✓	✓	✓	-	-

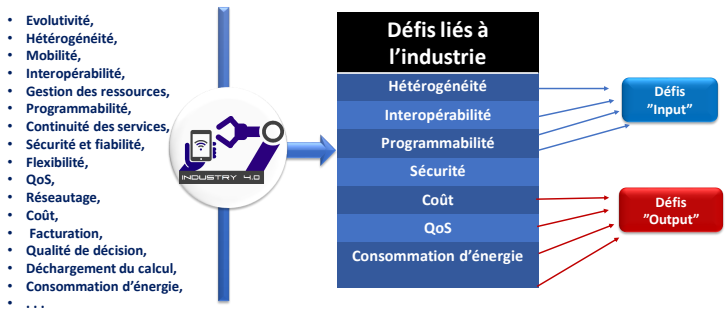


FIGURE 3.4 – Les défis généraux et industriels.

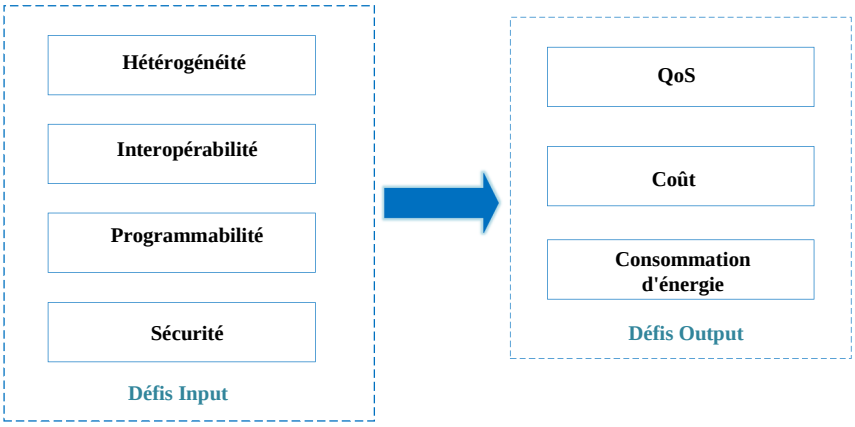


FIGURE 3.5 – Les liens entre les défis input and output du Fog Manufacturing.

Nous proposons de classer ces défis en fonction de leur rôle dans le contexte industriel. Comme le montre la figure 3.5, le premier groupe présente les "Input", qui sont les défis commandables, qu'on ne peut pas faire varier en agissant directement sur le système de production. Par exemple, pour modifier la programmabilité d'un système Fog, nous avons besoin de modifier le système d'exploitation des FNs, ce qui représente une action à part entière sur le système d'information de l'entreprise. Même chose pour les autres défis du groupe "Input" tels que l'hétérogénéité ou la sécurité. Le second groupe dépend des éléments du premier groupe "Input" et d'autres caractéristique de l'entreprise. Ces défis "Output" donnent une idée sur les performances du système d'information de l'entreprise. En d'autres mots, nous pouvons les considérer comme étant des indicateurs de performance du système de production. Cette classification ne limite pas les défis à ceux décrits précédemment, mais elle tente de souligner les défis qui bloquent les applications liées à l'Industrie 4.0 et qui se basent sur le Fog Manufacturing. Il n'est pas évident d'évaluer le niveau de chaque caractéristique du Fog computing vue la forte hétérogénéité et l'interaction de ces paramètres. Nous proposons de faire une échelle d'évaluation à plusieurs niveaux pour chaque défi, où l'échelle varie de 0 (très mauvais ou inexistant) à 10 (excellent). Le lien entre les défis est quantifié par des poids $W_{i,j}$ tel que i est l'indicateur de performance, PI est la caractéristique influente du défi et j est l'indicateur du défi influencé.

3.3.1 Les défis "Input"

Programmabilité (P)

Bénéficier d'outils programmables est indispensable pour que l'environnement industriel puisse suivre l'évolution rapide du marché. Afin de répondre aux exigences de l'Industrie 4.0, les nœuds de la couche Fog computing présents dans l'atelier nécessitent d'être gérés ou modifiés selon le besoin du moment de production. Ces nœuds de calcul, qui sont généralement des routeurs ou commutateurs, ont comme rôle l'exécution des modèles importés du Cloud computing en proximité de la couche des terminaux (machines, robots, interfaces opérateurs, ...).

Plusieurs travaux de recherche (par exemple YI, LI et LI, 2015, VAQUERO et RODERO-MERINO, 2014) indiquent que les environnements contenant une solution

basée sur le Fog doivent pouvoir intégrer ou supprimer de nouvelles ressources pour atteindre le niveau de flexibilité souhaité. (DE BRITO et al., 2016) ont travaillé sur la programmabilité des FNs dans un atelier de production pour offrir plus de flexibilité, fiabilité et efficacité au processus de production.

Quelques modèles et structures de programmation adaptés à l'architecture basée sur le Fog computing sont proposés par (DASTJERDI et al., 2016 et VARSHNEY et SIMMHAN, 2017) afin de permettre aux industries d'exécuter leurs applications sur les nœuds de la couche Fog. La question qui se pose est la suivante : combien et quel type de FNs doit-on installer pour garantir la flexibilité de la couche Fog computing et être compatible avec toutes les applications industrielles ? La programmabilité est un défi sensible aux indicateurs commandables de performance ou les PIs (Performance Indicator) d'entrée suivants :

— Les connaissances que possède chaque module (Cn),
— L'espace de stockage libre et l'intégration continue des nouvelles fonctionnalités (St),
— L'utilisation des règles et des normes/standardisation (N),

Nous pouvons juger les performance de la programmabilité en observant les PIs de sortie suivants :

— La réussite d'exécution des tâches attribuées à chaque module/Fiabilité (Ef),
— Le temps d'exécution des tâches attribuées à chaque module (T),
— La flexibilité (F) : la capacité du module à s'adapter aux différentes situations. En effet un module programmable offre la possibilité de rajouter ou d'adapter des fonctions selon le besoin.
— Le coût des modules programmables et de l'intégration de chaque nouvelle fonctionnalité (C). Ce coût est composé d'une partie statique liée au coût d'installation et de dépenses et d'une partie dynamique qui concerne le coût de traitement et de mise à jour des modules,
— La consommation d'énergie du au fonctionnement des modules et à la des transmissions des données (E).

On peut schématiser la programmabilité en fonction de ses PIs d'entrées (commandables) et de sorties (observables) selon le schéma présenté sur la figure 3.6. Si l'on suppose que le lien entre ces éléments est linéaire on peut modéliser la programmabilité avec la fonction suivante :

$$P = W_{Cn,P} \times Cn + W_{St,P} \times St + W_{N,P} \times N \tag{3.1}$$

sachant que :

$$W_{Cn,P} + W_{St,P} + W_{N,P} = 1. \tag{3.2}$$

Les entrées sont les suivantes : $W_{Cn,P}$ est le poids des connaissances acquises, $W_{St,P}$ est le poids lié à la taille de l'espace de stockage et $W_{N,P}$ est le poids du niveaux d'interopérabilité ou de standardisation. La programmabilité, à son tour, influence plusieurs autres PIs du Fog computing comme le montre la figure 3.6. Nous avons relevé les plus importants en nous basant sur l'état de l'art et notre expérience d'implémentation de Fog dans l'atelier flexible du laboratoire. En effet, la programmabilité influence principalement la flexibilité et la fiabilité du système de production puisqu'elle définit si l'on peut ou non rajouter des nouveaux comportements. Elle influence le coût de gestion du système d'information. Par exemple, elle définit la quantité de modification et d'investissement nécessaire pour assurer la flexibilité et la résilience du système. Son influence sur la consommation énergétique est justifiée par le fait que l'adaptation de plusieurs modules de calcul/programmes consomme d'avantage l'énergie.

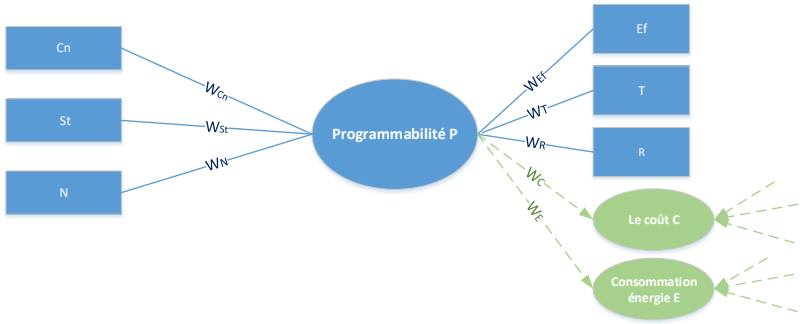


FIGURE 3.6 – Les PIs d'entrées/sorties de la programmabilité.

Les sorties sont les suivantes : **R** est la réussite ou non des tâches attribuées, **T** est le temps de transmission et de traitement des requêtes, **Ef** est la fiabilité des résultats, **C** est le coût d'intégration de nouvelles fonctionnalités et **E** c'est l'énergie consommée pour une telle intégration.

Il faut noter que pour les sorties nous ne détaillons pas leurs poids comme cela n'influence pas directement les défis. Cette démarche va être la même pour les défis qui suivent.

Sécurité (S)

De même, la sécurité et la fiabilité des applications basées sur le Fog computing, en termes de transmission et de stockage de données, constituent un défi fondamental pour le CPS dans le contexte de l'Industrie 4.0 (OSANAIYE et al., 2017, VARSHNEY et SIMMHAN, 2017). Les données industrielles liées aux différents processus de production sont confidentielles et à diffusion limitée. L'architecture centralisée basée sur le Cloud computing a prouvé sa capacité à intégrer des politiques de protection de données.

Étant donné que l'architecture basée sur le Fog computing a une nature distribuée, assurer la sécurité des données transmises entre les différentes couches de l'architecture, présente l'inconvénient le plus important à surmonter (ZAKOLDAEV et al., 2019). La création des accès tout le long du réseau pour insérer les FNs et les passerelles augmente les risques de perte ou violation de données (attaque du spoofing, violation des nœuds, intrusions sur les objets connectés, données malveillantes, attaque par déni de service DoS, panne de FN, attaque de rediffusion, attaques cyber-physiques, etc.) (PUTHAL et al., 2019).

Afin d'augmenter la sécurité au niveau des FNs ou les points d'accès, surtout pour les applications industrielles, il est nécessaire d'intégrer des politiques de sécurité (par exemple l'authentification, le contrôle d'accès, le cryptage, la confidentialité et la détection des parasites, etc., (YI, LI et LI, 2015)) .

Les risques cités auparavant devraient être résolus pour assurer une haute sécurité des données et des systèmes industriels. La fiabilité de l'architecture basée sur le Fog computing peut être améliorée par de nombreuses techniques, nous pouvons citer parmi elles les suivantes :

— La subdivision de l'ensemble d'architecture en zones (la couche des terminaux, la couche des capteurs, la couches des FNs, etc.,) pour faciliter l'intégration des défenses selon les menaces et les besoins existants,

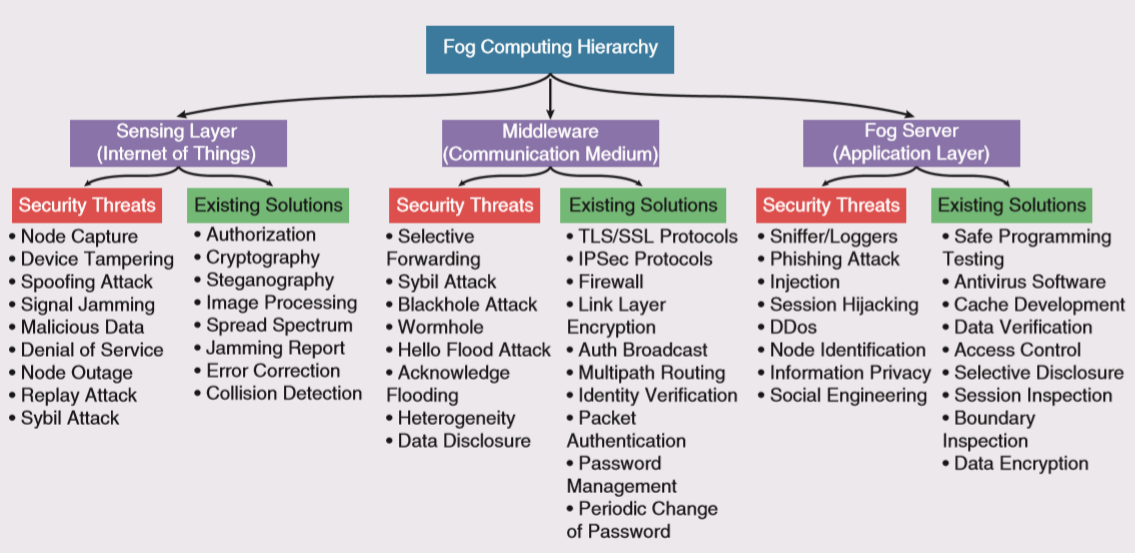


FIGURE 3.7 – Les solutions possibles pour la problématique de sécurité des données (PUTHAL et al., 2019).

— La création et la gestion des politiques de sécurité pour tous les dispositifs d'une architecture Fog. En fait, si un ou plusieurs dispositifs (logiciel ou matériel) sont compromis, le système basé sur le Fog n'est plus fiable,
— L'insertion des politiques pour spécifier qui ou quand on peut accéder à quelle ressource et dans quelles circonstances. Ces stratégies peuvent être intégrées au niveau des FNs ou des logiciels liés au FNs et gérées par un administrateur autorisé. Ce contrôle d'accès augmente la sécurité mais limite la flexibilité et l'extensibilité du CPS,
— La sécurité des communications afin de protéger les données lors de la transmission entre les couches (VAQUERO et RODERO-MERINO, 2014).

La figure 3.7 présente de nombreuses solutions aux problèmes de sécurité selon la localisation (sur l'architecture basée sur le Fog computing).

Concernant les PIs de la sécurité d'un système, nous avons trouvé que c'est plus complexe à définir et à évaluer selon le cas d'application et les politiques de sécurité adoptées (STOLFO, BELLOVIN et EVANS, 2011). Selon MIHALACHE, 2007, la sécurité S à l'instant t est la probabilité d'éviter les événements critiques ou dangereux liés aux données E sur un intervalle [0.1], comme le présente l'équation :

$$S(t) = P[Ev(\ \textit{eviter les evenements critiques ou dangereux sur}\ [0,t])]. \tag{3.3}$$

Lorsque l'on considère que la sécurité est le fait d'éviter des événements dangereux, les PIs d'entrée commandables selon notre contexte sont les suivants :

— Le type de connexions et de modules utilisés (les modules programmables) Cx,
— Le nombre de défenses indépendantes installées D,
— La probabilité d'existence des événements dangereux ou critiques (Parmi une liste des attaques possibles) Ed,
— L'espace de stockage libre qui permet l'intégration continue des nouvelles politiques de sécurité St.

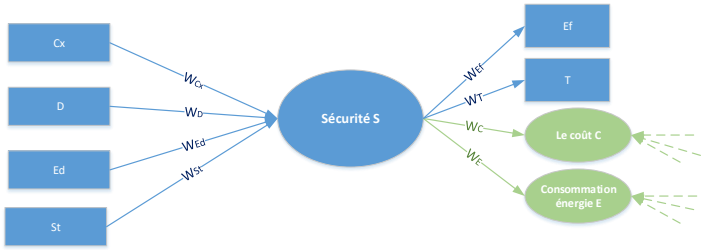


FIGURE 3.8 – Les PIs d'entrées/sorties de la sécurité.

Nous considérons que le défi de la sécurité est une probabilité mesurable en fonction des poids des PIs d'entrées. Cela permet d'évaluer la sécurité en considérant les PIs de sortie possibles :

— La perte ou la fiabilité de données (QoS) Ef,
— Le temps de transmission et de traitement des requêtes T,
— Le coût d'intégration des politiques de sécurité C,
— L'énergie consommée lors des traitements des politiques de sécurité E.

Les PIs cités ci-dessus sont présentés sur la figure 3.8.

L'équation des liens linéaires présente la sécurité en fonction des poids des PIs d'entrées et de sorties présentées sur la figure 3.8 :

$$S = W_{Cx,S} \times Cx + W_{D,S} \times D + W_{Ed,S} \times Ed + W_{St,S} \times St, \tag{3.4}$$

sachant que :

$$W_{Cx,S} + W_{D,S} + W_{Ed,S} + W_{St,S} = 1. \tag{3.5}$$

Les entrées sont les suivantes : $W_{Cx,S}$ est le poids de type des liens utilisés, $W_{D,S}$ est le poids du nombre des défenses installées, $W_{Ed,S}$ est le poids de la probabilité d'existence des événements dangereux et $W_{St,S}$ est l'espace libre de stockage.

Les sorties sont les suivantes : **Ef** est le taux de perte de données, **T** est le temps mis pour appliquer une défense, **C** est le coût d'intégration des politiques de sécurité et **E** est l'énergie consommée lors de la protection des données.

La sécurité, comme le montre la figure 3.8, influence de nombreux aspects liés aux autres défis (Input et Output). Les PIs que l'on a choisis sont sélectionnés en revenant à l'état de l'art, et à notre architecture basée sur le Fog manufacturing conçue pour la chaîne de production du laboratoire. Nous remarquons que la sécurité entraîne des changements sur la QoS principalement. Le taux de perte de données est le principal facteur à observer afin de juger de l'efficacité des politiques de défense. Aussi le délai mis pour la transmission et le traitement des données donne une idée sur l'efficacité des défenses installées. La sécurité influence le coût d'intégration des politiques de sécurité, ainsi que le coût consommé pour sécuriser les données. Par exemple un codage ou un chiffrement à la source suivi par un décodage au niveau de récepteur augmente le coût. Ce défi agit aussi sur la consommation d'énergie, chaque traitement supplémentaire consomme d'avantage d'énergie.

Hétérogénéité (H)

Les industries regroupent un ensemble de systèmes inter-connectés très variables hétérogènes générant des données hétérogènes, qui nécessitent un traitement en temps réel. Ainsi le Fog computing est la solution qui étend les services des serveurs Cloud aux terminaux connectés et aux IoT. En conséquence, il utilise la majorité des

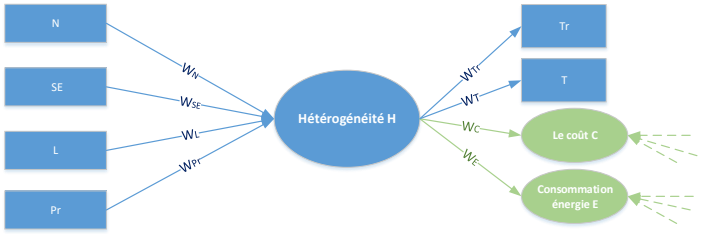


FIGURE 3.9 – Les PIs d'entrées/sorties de l'hétérogénéité.

dispositifs qui résident entre la couche Cloud et la couche IoT. Ces dispositifs hétérogènes sont par la suite considérés comme étant des FNs.

Lors de la prise en considération de cette insertion de la couche Fog computing, les périphériques IoT qui sont déjà présents et les serveurs du Fog disposent d'espaces de stockage et de puissances de calcul hétérogènes (VAQUERO et RODERO-MERINO, 2014). Par conséquent l'architecture basée sur le Fog doit être modélisée d'une manière adéquate pour qu'elle soit facilement déployée par tous les FNs.

Par contre, les types hétérogènes des FNs et leurs capacités de calcul limitées doivent être capables d'apporter des réponses en temps réel avec la QoS requise par les exigences des CPS. Un autre problème d'hétérogénéité peut atténuer la performance du Fog. Les FNs sont distribués et chacun est chargé d'une tâche différente. Pour accomplir cette tâche les FNs doivent consulter des ressources hétérogènes et certains nœuds d'un même niveau peuvent accéder des ressources différentes. La disposition des solutions pour un tel défi est très importante afin de réussir une architecture basée sur le Fog computing. Plusieurs travaux ont pris en considération cette problématique, comme par exemple (PALLEWATTA, KOSTAKOS et BUYYA, 2019) où les auteurs ont proposé une politique de placement d'application basée sur les micro-services pour les environnement hétérogènes du Fog computing.

L'hétérogénéité est le fait d'éviter des événements dangereux, les PIs d'entrée influant ce défi sont les suivants (voir aussi SHARMA, OJHA et BHONDEKAR, 2019 WANG et al., 2019a) :

— L'ensemble des différents types de nœuds utilisés N,
— L'ensemble des différentes catégories des systèmes d'exploitations ou Hardware utilisés dans un atelier SE,
— L'ensemble des différents types de liens utilisés L,
— Le nombre de protocoles sémantiques installés Pr,

Les PIs de sortie identifiés pour l'hétérogénéité sont les suivants :

— Le délai des transmissions entre chaque deux SEs hétérogènes T,
— Le nombre des retransmissions Tr,
— Le coût des communication C,
— L'énergie consommée lors des communications entre deux SEs hétérogènes E.

L'équation des liens linéaires présente l'hétérogénéité en fonction des poids des PIs d'entrées et de sorties présentées sur la figure 3.9 :

$$S = W_{N,H} \times N + W_{SE,H} \times SE + W_{L,H} \times L + W_{Pr,H} \times H, \tag{3.6}$$

sachant que :

$$W_{N,H} + W_{SE,H} + W_{L,H} + W_{Pr,H} = 1. \tag{3.7}$$

En considérant les entrées suivantes : **N** est l'ensemble qui regroupe les différents types présents dans l'atelier (Nœuds), **SE** est le nombre des SEs et hardware existants **L** l'ensemble des différents types de liens et **Pr** est le nombre des protocoles interopérables installés.

Et les sorties suivantes : **T** est le délai mis entre deux nœuds hétérogènes, **Tr** est le nombre de retransmission entre deux nœuds hétérogènes, **C** est le coût de chaque communication entre deux nœuds hétérogènes et **E** est l'énergie consommée par chaque communication.

L'hétérogénéité comme le montre la figure 3.9 influence de nombreux aspects liés aux autres défis, comme par exemple l'interopérabilité. L'utilisation des normes de standardisation comme OPC UA et des protocoles sémantiques aide à homogénéiser l'environnement industriel qui comportes différents SEs. L'hétérogeinété entraîne des changements sur la QoS principalement. Le taux de perte de données est le principal facteur à observer afin de juger de l'efficacité des protocoles sémantiques. Aussi le délai mis pour la transmission et le traitement des données donne une idée sur l'homogénéité des systèmes présents dans l'environnement. L'hétérogénéité influence le coût d'installation des techniques d'homogénéisation d'échange entre deux SEs différents. Ce défi agit également sur la consommation d'énergie, chaque traitement supplémentaire consomme davantage l'énergie.

Interopérabilité (I)

De nos jours, posséder un environnement interopérable en terme de communication devient une nécessité suite à l'émergence des applications basées sur l'IoT, qui intègrent un ensemble de dispositifs inter-connectés hétérogènes utilisant différents systèmes d'exploitation, protocoles et technologies. Malgré le bon lien de communication, dans le cas d'un échange entre deux SEs différents, chacun ne peut interpréter le sens de l'échange avec l'autre, ce qui rend la communication absurde. Cette nouvelle génération d'applications exige un certain niveau de compréhension entre les différentes parties concernées par le service. En particulier, des solutions qui intègrent des protocoles de communication sémantique doivent être conçus.

Les ontologies des protocoles sémantiques sont déjà implémentées sur les serveurs centralisés du Cloud computing (ANDROCEC, VRCEK et SEVA, 2012). Par contre la procédure de demande, d'exécution et de réponse n'est pas forcément en temps réel, ce qui ne convient pas aux exigences de l'Industrie 4.0.

Pour homogénéiser cet environnement, la mise en œuvre du Fog computing sur un site industriel présente un potentiel prometteur (CHANG, SRIRAMA et BUYYA, 2017). L'intégration du Fog computing dans un contexte industriel diminue les délais de transmission, par contre le fait d'intégrer des FNs augmente le nombre d'appareils différents connectés sur la même architecture, ce qui nécessite un échange encore plus interopérable (RAHMAN et HUSSAIN, 2019).

L'intégration des protocoles sémantiques de gestion du site de production constitue un besoin pressant, qui assure une première base d'échange compréhensible pour une architecture industrielle basée sur le Fog computing (VAQUERO et RODERO-MERINO, 2014). Lors de ce renforcement d'interopérabilité, il faut prendre en considération que les approches d'interopérabilité adaptées à l'IoT dépassent les ressources de stockage et d'exécution limitées des dispositifs (FNs et terminaux). Des travaux ont porté sur une telle problématique. Par exemple,(NEGASH, WESTERLUND et TENHUNEN, 2019) ont introduit un serveur dédié aux protocoles d'interopérabilité et qui peut être déployé au niveau de la couche Fog computing.

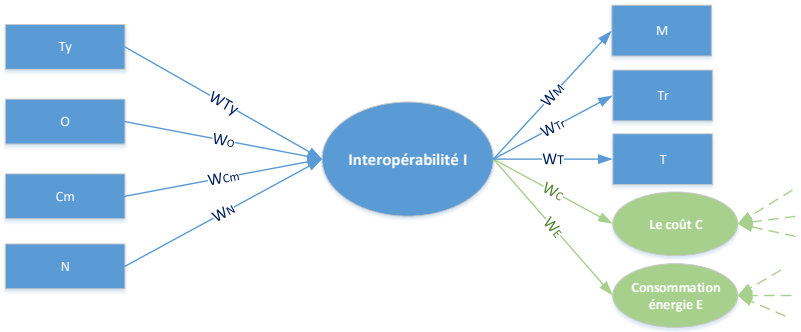


FIGURE 3.10 – Les entrées/sorties de l'interopérabilité.

Afin de mesurer l'interopérabilité, on a besoin des métriques ou des PIs. Ces mesures aident à commander le niveau d'interopérabilité de notre système. D'après l'état de l'art, nous avons identifié trois types de mesure de l'interopérabilité (DUCQ et CHEN, 2008) :

— **Mesure potentielle :** L'objectif est d'évaluer la capacité d'un système à s'adapter de manière dynamique pour surmonter les obstacles éventuels lors de l'interaction avec un autre,
— **Mesure de compatibilité :** L'objectif est d'identifier les obstacles qui empêchent le renforcement de l'interopérabilité,
— **Mesure de performance**.

On a identifié des PIs d'entrée d'interopérabilité liés à notre contexte du Fog Manufacturing :

— Le type de données/activité/processus Ty,
— Les obstacles identifiés : conceptuels, organisationnels, ou technologique O,
— La comptabilité des liens et protocoles d'échange Cm,
— Le nombre de protocoles ou technologies sémantiques installés N.

Les PIs de sortie que l'on peut observer afin de juger le niveau d'intéropérabilité de notre système sont les suivants :

— Le niveau de maturité du système : Isolé (incapable d'être interopérable), Initial (nécessite des effort pour être interopérable), Exécutable (possède potentiel d'interopérabilité), Connecté (peut facilement être interopérable), Interopérable M,
— Le délai des transmissions entre chaque SEs hétérogènes T,
— La probabilité de l'existence des évènements dangereux N,
— Le nombre des retransmissions Tr,
— Le coût en termes de l'intégration des protocoles d'interopérabilité et des communications C,
— L'énergie consommée lors de l'adaptation sémantique des données E.

L'équation des liens linéaires présente l'intéropérabilité en fonction des poids des PIs d'entrées présentées sur la figure 3.10 :

$$I = W_{Ty,I} \times Ty + W_{O,I} \times O + W_{Cm,I} \times Cm + W_{N,I} \times N, \tag{3.8}$$

sachant que :

$$W_{Ty,I} + W_{O,I} + W_{Cm,I} + W_{N,I} = 1. \tag{3.9}$$

Avec les entrées suivantes : **Ty** est le poids de type des liens utilisés, **O** est le poids de nombre des défenses installées et **Cm** est le poids de la probabilité d'existence des

événements dangereux et **N**.

Les sorties suivantes : **M** présente le niveau de maturité du nœud, **Tr** est le nombre de retransmission entre deux nœuds, **T** est le délai de communication entre deux nœuds, **C** est le coût de l'installation des protocoles d'interopérabilité et **E** est l'énergie consommée par chaque communication.

L'interopérabilité est considérée comme un défi très corrélé aux défis Input et Output. Afin de surmonter l'hétérogénéité, les protocoles intéropérables sont primordiaux, car ces protocoles assurent la compréhension des échanges entre les différents systèmes existants dans l'atelier de production. Avoir des normes intéropérables est important afin de surmonter des défis comme la sécurité ou encore la programmabilité. La bonne compréhension mène au bon traitement des demandes provenant des systèmes, ce qui facilite la protection de données ainsi que la réussite des tâches. L'intéropérabilité influence ainsi des défis Output comme le coût (les frais d'installation des protocoles sémantiques) ainsi que la consommation d'énergie supplémentaire en appliquant des traitements de standardisation. La QoS est aussi influencé par un tel défi, tout en observant le temps de réponse et les délais de transmission des demandes et réponses suite à l'intégration de l'intéropérabilité.

3.3.2 Les défis "Output"

Qualité de Service (QoS)

Concernant la performance des applications industrielles, l'indicateur de QoS exprime la qualité de la connectivité, la performance, la gigue, la perte de données et le délai de transmission des données (CHANG, SRIRAMA et BUYYA, 2017). Dans (BROGI et FORTI, 2017), les auteurs se placent dans le contexte d'intégration du Fog Computing, et ils ne considèrent que la latence et la bande passante comme étant des mesures de la QoS car la perte de données et la gigue peuvent être corrigées par la retransmission et la mise en mémoire.

La QoS industrielle vise à permettre une exécution en temps réel/quasi temps réel des décisions générées par la couche Fog Manufacturing.

Les FNs sont caractérisés par la couverture à grande échelle de nombreux dispositifs simultanément. Cette distribution et hiérarchie de calcul dégradent la QoS fournie par les serveurs du Fog Computing aux applications. Cette problématique présente un défi pour l'intégration du Fog Computing dans un contexte d'Industrie 4.0. L'utilisation des FNs avec des capacités de calcul différentes, permettant un certain niveau de QoS est un problème pour les applications industrielles sensibles à la perte des données et à des délais élevés (YI, LI et LI, 2015).

Également, les FNs ont besoin de réserver des ressources pour satisfaire les exigences des applications. Ces réservations se font selon la disponibilité des ressources au moment de la demande. Si les ressources sont limitées ou inadéquates, la QoS va forcément s'atténuer (GUBBI et al., 2013).

Des recherches sur cette problématique ont été récemment introduites comme par exemple les travaux de (CHEN, WANG et GONG, 2019), qui visent à améliorer la QoS pour les applications distribuées basées sur le Fog Computing. Un algorithme est proposé pour améliorer la QoS tout en respectant les ressources, la consommation d'énergie et la capacité des FNs. Pour l'effet de la réservation des ressources, de nombreuses politiques d'estimation des ressources sont développées (MAHMUD et al., 2019b).

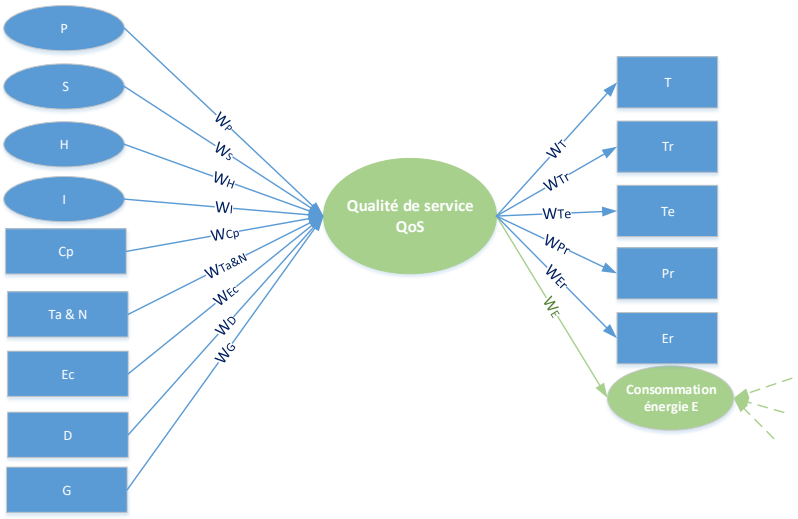


FIGURE 3.11 – Les entrées/sorties de la QoS.

Pour mesurer la QoS, il existe de nombreux standards selon l'application. Concernant notre cas, nous avons considéré les caractéristiques suivantes comme étant des PIs de la QoS (RAZVAN BEURAN, 2004) :

— La capacité de transfert Cp $C = quantite_{donnees}/temps_{transfert}$,
— La taille Ta et le nombre N des paquets,
— L'écart entre les paquets Ec $T_{P_i} - T_{P_{i-1}}$,
— Le débit : D =N/d,
— La gigue : G = $1/(N-1) * \sum_{i=2}^{N} |d_i - d_{i-1}|$, avec i est l'indice de paquet,
— Le délai (Unidirectionnel ou Bidirectionnel) : T= $T_{reception} - T_{transmission}$,
— Le taux de retransmission Tr,
— Le taux d'expédition Te,
— Le taux de perte de paquets Pr (Reçu ou perdu : Fiabilité) : $T_{perte} = N_{paq_perdu}/N_{paq_transmis}$,
— Le taux l'erreur Er (Faux ou vrai : Fiabilité) : $T_{err} = N_{paq_erreur}/(N_{paq_succes} + N_{paq_erreur})$,
— La consommation d'énergie pour une transmission E.

La figure 3.11 présente les PIs identifiés comme étant les PIs d'entrées de la QoS.

L'équation des liens linéaires présente la QoS en fonction des poids des PIs d'entrées présentées sur la figure 3.11 :

$$\begin{aligned} QoS = W_{P,QoS} \times P \quad & +W_{S,QoS} \times S + W_{H,QoS} \times H + W_{I,QoS} \times I \\ & +W_{E,QoS} \times E + W_{C,QoS} \times C + W_{Cp,QoS} \times Cp \\ & +W_{Ta\&N,QoS} \times Ta\&N + W_{Ec,QoS} \times Ec \\ & +W_{D,QoS} \times D + W_{G,QoS} \times G, \end{aligned} \tag{3.10}$$

sachant que :

$$\begin{aligned} & W_{P,QoS} + W_{S,QoS} + W_{H,QoS} + W_{I,QoS} + W_{E,QoS} + W_{C,QoS} \\ & +W_{Cp,QoS} + W_{Ta\&N,QoS} + W_{Ec,QoS} + W_{D,QoS} + W_{G,QoS} \quad = 1. \end{aligned} \tag{3.11}$$

La QoS est classée comme un défi Output qui a comme entrées tous les défis

Input (Programmabilité, sécurité, intéropérabilité et hétérogénéité). Surmonter ces derniers augmente la QoS. La QoS est liée également à la consommation d'énergie et au coût à travers le temps de transmission.

Consommation d'énergie

Les industries considèrent la consommation d'énergie comme étant un défi majeur, et c'est pour cette raison que l'un des objectifs de la quatrième révolution industrielle est de minimiser le coût énergétique du processus de production.

Cependant, la solution basée sur le Fog computing regroupe un grand nombre distribué de FNs qui s'occupent de la collecte et du calcul de données. Ce qui augmente évidemment la consommation d'énergie des FNs par rapport au modèle centralisé traditionnel basé sur le Cloud computing (DASTJERDI et al., 2016, VARSHNEY et SIMMHAN, 2017). Par conséquent, l'optimisation de la consommation d'énergie d'une architecture basée sur le Fog computing est un défi délicat à relever.

Sous l'hypothèse où ce défi serait relevé, le Fog computing peut réduire la consommation d'énergie de la collecte et du calcul des données grâce au traitement local des demandes. L'optimisation de la consommation d'énergie dans ce contexte présente un axe de recherche prometteur. La majorité des travaux (O'DONOVAN et al., 2019, WAN et al., 2018) qui s'intéressent à l'intégration du Fog Manufacturing prennent ce défi en considération. Plusieurs modèles sont élaborés afin de limiter la consommation d'énergie pour les architectures basées sur le Fog computing.

La mesure et l'estimation de la consommation d'énergie est une condition importante pour l'amélioration future de l'efficacité énergétique d'une solution basée sur le Fog Manufactring, c'est également un aspect important par rapport aux questions de développement durable et des enjeux sociétaux qui vont avec. La consommation d'énergie peut se calculer en terme des flux de données, des temps de traitement ou du nombre de couches (JALALI et al., 2016). La consommation d'énergie peut-être mesurée selon le cas ou le besoin, par exemple on peut mesurer :

— La consommation d'énergie par bit Eb,
— La consommation totale d'énergie d'un nœud ETn,
— La consommation totale d'énergie d'un service ETs,
— Le coût de la consommation d'énergie par flux C.

On a identifié de nombreux PIs liés à la consommation d'énergie dans le contexte du Fog Manufacturing comme :

— Les types de systèmes d'exploitation présents dans l'atelier SE,
— Le nombre des nœuds présents dans l'atelier Nd,
— La charge d'un nœud Ch,
— La puissance maximale et moyenne des SEs Pmax,
— La capacité maximale du système Cmax,
— La taille d'un paquet de données transmis (Le nombre de bits échangés) Tp,
— le nombre de flux simultanément transmis ou reçu par nœud Nfx,
— Le seuil de charge Sl.

Ces PIs sont présentés sur la figure 3.12 en fonction de leur rôle (entrées). De nombreuses formules de calcul et modèles existent afin d'optimiser la consommation d'énergie d'une solution basée sur le Fog computing.

L'équation des liens linéaires présente la consommation d'énergie en fonction des poids des PIs d'entrées présentées sur la figure 3.12 :

$$\begin{aligned} E = W_{P,E} \times P \quad & + W_{S,E} \times S + W_{H,E} \times H + W_{I,E} \times I \\ & + W_{QoS,E} \times QoS + W_{C,E} \times C + W_{SE,E} \times SE \\ & + W_{Nd,E} \times Nd + W_{Ch,E} \times Ch + W_{Pmax,E} \times Pmax \\ & + W_{Cmax,E} \times Cmax + W_{Tp,E} \times Tp + W_{Nfx,E} \times Nfx + W_{Sl,E} \times Sl, \end{aligned} \tag{3.12}$$

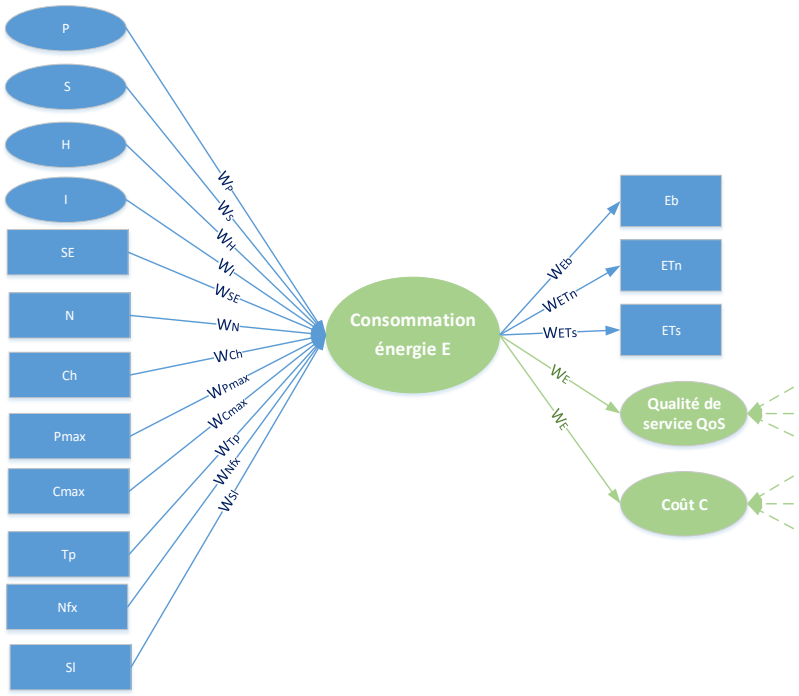


FIGURE 3.12 – Les entrées/sorties de la consommation d'énergie.

sachant que :

$$\begin{aligned} &W_{P,E} + W_{S,E} + W_{H,E} + W_{I,E} + W_{QoS,E} + W_{C,E} + W_{SE,E} \\ &+W_{Nd,E} + W_{Ch,E} + W_{Pmax,E} + W_{Cmax,E} + W_{Tp,E} + W_{Nfx,E} + W_{Sl,E} \quad = 1. \end{aligned} \tag{3.13}$$

De même que les autres défis, la consommation d'énergie est influençable par tous les défis Input. Surmonter ces défis nécessite des traitements supplémentaires qui consomment forcément de l'énergie. L'énergie influence aussi les deux défis Output "coût" et "QoS".

Coût

Le coût de l'installation de chaque nouvelle technologie influence son adoption par les industries. Le Fog computing possède de nombreux avantages pour le contexte de l'Industrie 4.0, mais l'objectif est bien sûr de l'intégrer avec le moindre coût possible.

L'insertion d'une couche additionnelle entre celle du Cloud computing et des terminaux connectés à l'IoT nécessite forcément un coût supplémentaire (FNs, points d'accès, passerelles, liaisons). Le coût du matériel et du logiciel d'installation est considéré comme un véritable défi à surmonter en adoptant un modèle économique adéquat (YI, LI et LI, 2015, VARSHNEY et SIMMHAN, 2017).

La résolution de ce défi exige la définition de l'architecture et l'identification des besoins de l'application en terme de matériel. Le choix des FNs doit être optimal en termes de capacité demandée et de coût unitaire.

L'optimisation du coût de l'installation d'une architecture basée sur le Fog Manufacturing est étudiée dans plusieurs travaux (GU et al., 2015, VERBA et al., 2019). Certains d'entre eux ont diminué le coût et ont prouvé la rentabilité du Fog à long

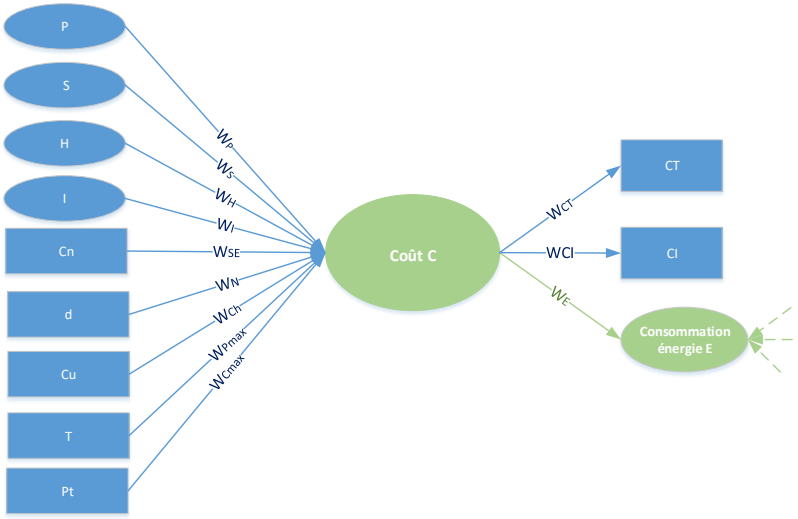


FIGURE 3.13 – Les entrées/sorties du coût.

terme (LIN et YANG, 2018). Par contre, cette étude dans le contexte de l'Industrie 4.0 reste assez modeste. Le coût est une mesure qui concerne principalement deux aspects :

— Le coût de la communication entre deux nœuds : coût de Traitement,
— Le coût de l'intégration de la solution basée sur le Fog computing : coût des équipements,

Comme c'est indiqué, le coût est un défi primordial à surmonter pour que les industries décident d'adopter le Fog computing. Le coût peut se différencier en (YAO et al., 2017, GU et al., 2015, LIN et YANG, 2018) : Coût d'implémentation, Coût de communication, Coût de sécurité, etc. Afin de mesurer et d'optimiser le coût, il faut identifier les PIs et les métriques de mesure du coût, qui sont présentées ci-dessous :

— Le type de module Fog implémenté P,
— Le nombre de politiques de sécurité implémentées S,
— Le nombre de chaque type d'équipement différent (nœud) H,
— Le nombre de protocoles interopérables installés I,
— Le coût de chaque type de nœud Cn,
— La distance entre les différents nœuds d,
— Le coût d'une unité de lien Cu,
— Le délai d'une communication T,
— La taille d'un paquet transmis Pt.

Les PIs du coût ne se limitent pas à ceux cités, nous pouvons intégrer de nouveaux indicateurs selon l'application. La figure 3.13 présente ces PIs selon leur rôle (entrées/sorties).

L'équation des liens linéaires présente du coût en fonction des poids des PIs d'entrées présentées sur la figure 3.14 :

$$\begin{aligned} C = W_{P,C} \times P \quad & + W_{S,C} \times S + W_{H,C} \times H + W_{I,C} \times I \\ & + W_{E,C} \times E + W_{Cn,C} \times Cn + W_{d,C} \times d \\ & + W_{Cu,C} \times Cu + W_{T,C} \times T + W_{G,C} \times G \end{aligned} \tag{3.14}$$

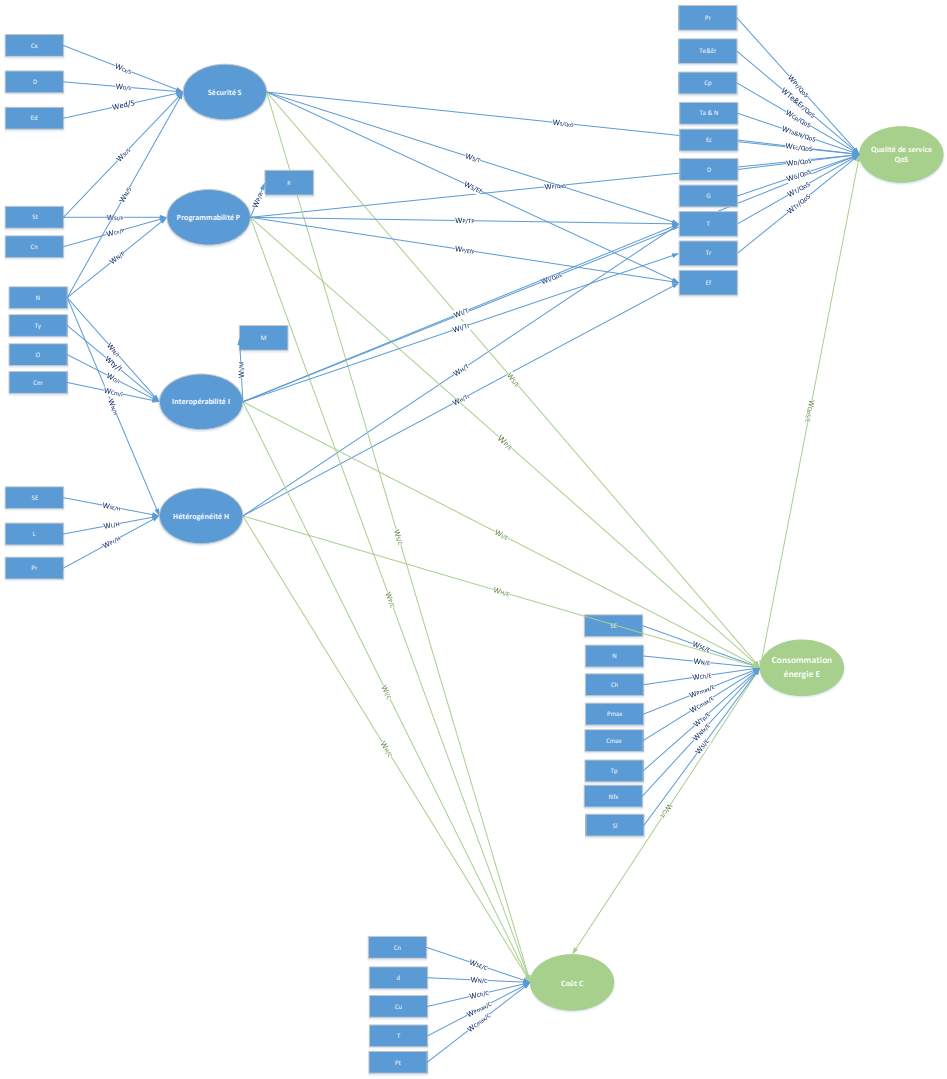


FIGURE 3.14 – Les liaisons complètes entre les défis Input et Output.

sachant que :

$$\begin{aligned} &W_{P,C} + W_{S,C} + W_{H,C} + W_{I,C} + W_{E,C} \\ &+W_{Cn,C} + W_{d,C} + W_{Cu,C} + W_{T,C} + W_{G,QoS} \quad = 1 \end{aligned} \tag{3.15}$$

3.3.3 Les liens entre les défis

Les défis identifiés dans la première section sont bien corrélés avec les exigences industrielles (flexibilité, interopérabilité, QoS, communication en temps réel, sécurité, etc.). Afin de surmonter ces défis identifiés, une analyse de leurs interactions doit être faite, ce qui facilitera la conception des solutions efficaces. L'identification des liens se réfère au numéro de défi représenté par la Figure 3.15, qui sera expliqué ci-dessous.

H VS Coût (1-4) :

L'hétérogénéité des systèmes présents dans l'architecture, qui génèrent une grande quantité de données différentes, a un impact direct sur le coût d'installation et de gestion de ces données. La transmission, le traitement et le stockage de ces données dans les FNs nécessitent une puissance de calcul coûteuse. L'intégration

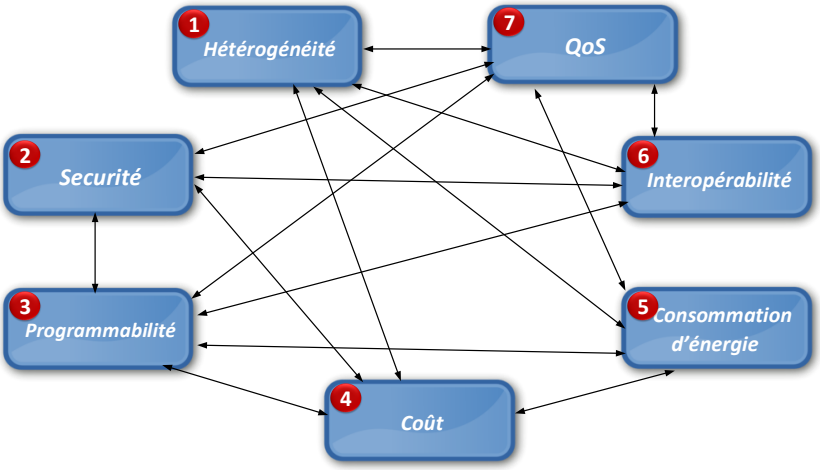


FIGURE 3.15 – Les défis liés au Fog Computing.

des protocoles de communication performants, des adaptateurs, des modules et des FNs adéquats avec le reste des systèmes hétérogènes existants augmente le coût du Fog Manufacturing.

Par la suite l'hétérogénéité a un effet direct sur le coût. Il est donc important de prendre le défi du coût en considération lors de la recherche des solutions afin d'homogénéiser l'environnement du Fog Manufacturing.

L'hétérogénéité touche aussi à la capacité de stockage de chaque dispositif, qui est très limitée pour certains. Homogénéiser les capacité de stockage est primordial afin de supporter la quantité de calcul de données acheminée tout le long de l'architecture. Renforcer ces capacités limitées de stockage rajoute des coûts supplémentaires.

H VS Consommation d'énergie (1-5) :

Le fonctionnement hétérogène des systèmes industriels consomme d'une manière différente les ressources énergétiques. L'utilisation du Fog Computing augmente cette demande énergétique à cause de la puissance électrique nécessaire au fonctionnement des différentes nœuds de Fog. D'autre part, l'architecture distribuée permise par le Fog Computing peut être exploitée pour limiter cette consommation croissante.

Au regard des exigences d'optimisation de la consommation d'énergie, la maîtrise de cette consommation constitue un véritable défi à relever notamment au regard de la forte demande des systèmes hétérogènes pour la fourniture d'énergie. En ce qui concerne les exigences d'optimisation de la consommation d'énergie, la maîtrise de cette consommation composite constitue un véritable défi à relever, notamment en ce qui concerne la forte demande des systèmes hétérogènes pour la fourniture d'énergie.

H VS I (1-6) :

Les sources industrielles hétérogènes génèrent des données de formes et de types divers qui sont recueillies. L'hétérogénéité des données augmente l'interopérabilité du système, mais l'utilisation de la couche de Fog la réduit, car le Fog computing peut faire un pont efficace entre les différents dispositifs du système industriel.

Cependant, l'homogénéisation du flux de données augmentera l'interopérabilité, car l'adaptation des différentes formes de données assurera une communication facile entre des dispositifs hétérogènes. Une classification en fonction de la source homogénéisera le flux de données et rendra la communication interopérable.

L'interopérabilité et l'hétérogénéités sont deux défis qui se dirigent vers la même direction, résoudre l'hétérogénéité des systèmes d'exploitation, des données et des protocoles augmente l'interopérabilité de l'environnement. Cette dernière est la solution pour homogénéiser les échanges dans un atelier industriel.

H VS QoS (1-7) :

La QoS est considérée comme l'indicateur de performance le plus important du Fog Computing. Cependant, l'hétérogénéité des données peut avoir des effets qui dégradent les métriques de la QoS. Lorsqu'un système envoie un tel format de données à un système différent, un refus de réception va causer une perte de requête. Cette perte de données augmente le délai de transmission et entraîne une retransmission. Cet effet diminue forcément la QoS.

La QoS est considérée comme l'indicateur de performance le plus important pour évaluer l'état et l'efficacité du processus de communication. Considéré comme un défi Output, l'homogénéisation des données générées et transmises réduira le délai de transmission et de traitement, ce qui augmentera directement la QoS.

Les FNs peuvent servir d'intermédiaire entre les systèmes connectés qui sont à la fois hétérogènes et distribuent le calcul sur l'ensemble de l'usine.

S VS P (2-3) :

L'intégration des politiques de sécurité dans une architecture basée sur le Fog Manufacturing dépend du choix des FNs. Ces FNs doivent respecter plusieurs caractéristiques (la capacité de stockage, la capacité de traitement, la confidentialité des données, etc.) afin de pouvoir supporter les protocoles de sécurité et de cryptage de données.

L'amélioration de l'infrastructure industrielle avec de nouvelles fonctionnalités ou la suppression de celles qui ne sont pas nécessaires abaissera le niveau de sécurité, comme certaines fonctionnalités intègrent des politiques de sécurité. La sécurité est influençable par la programmabilité des FNs.

S VS Coût (2-4) :

La sécurité est un défi Input influant les défis Output. Un lien entre la sécurité des échanges et le coût de la solution basée sur le Fog Manufacturing existe.

Le fait d'insérer une couche intermédiaire génère des risques de sécurité supplémentaire. Le Fog distribue le calcul des données sur un nombre évolutif des nœuds, ce qui fait que garantir la sécurité des différents niveaux d'une architecture basée sur le Fog représente un défi d'investissement.

L'installation des politiques de sécurité partout dans le réseau augmente les coûts logiciels et de traitements. Posséder une architecture sécurisée au moindre coût est un défi à relever.

S VS I (2-6) :

L'utilisation de systèmes autonomes interopérables qui appliquent l'aspect lecture seule pour partager leurs données, comme le protocole MTConnect (LEE, BAGHERI et KAO, 2015), peut protéger les dispositifs et les données échangées, ce qui augmente la sécurité du CPS d'une application basée sur le Fog Manufacturing.

S VS QoS (2-7) :

Comme la fiabilité et la perte de données font partie des indicateurs de performance de la QoS, la sécurisation des échanges entre les différentes couches augmente ces caractéristiques de QoS.

Cependant, renforcer le canal de transmission avec plus de politiques de sécurité pèsera sur le traitement des données car il faut plus de temps pour crypter et décrypter les données envoyées, ce qui augmente les délais de transmission et de traitement au niveau de chaque nœud. Il faut équilibrer le lien entre ces deux défis.

P VS (Coût & Consommation d'énergie) (3-4) :

La programmabilité permet d'intégrer de nouvelles fonctionnalités aux fonctionnalités existantes, elle devient coûteuse surtout lors de l'ajout de d'extensions de stockage et de calcul. Cependant, obtenir des FN programmables distribuées qui supportent l'évolutivité est considéré comme une solution économique pour ajouter en permanence de nouvelles fonctionnalités. Ainsi la programmabilité évite les modules inutiles qui consomment de l'énergie et optimise les fonctionnalités pour diminuer la consommation électrique.

P VS I (3-6) :

La programmabilité est liée à l'interopérabilité, elle nécessite l'accès aux données et aux fonctionnalités de différents dispositifs pour les manipuler, ce qui est impossible sans protocole de communication interopérable, tel que le protocole MTConnect (ibid.). Même l'ajout de nouveaux modules programmables nécessite une communication interopérable pour intégrer les fonctionnalités.

P VS QoS (3-7) :

La programmabilité des FNs permet d'ajouter d'une manière fluide de nouvelles fonctionnalités et modules, de mettre à jour les fonctionnalités existantes et d'augmenter la flexibilité du système. Cette notion de programmabilité augmente la flexibilité des applications industrielles. Par conséquent, la flexibilité mène à mieux s'adapter aux besoins des systèmes de production et de fournir des décisions précises et en temps réel, ce qui améliore la QoS.

Coût VS Consommation d'énergie (4-5) :

Les industries prêtent attention à l'aspect économique. Le coût de la solution basée sur le Fog Manufacturing a un grand rôle pour encourager les investisseurs du secteur industriel.

Les systèmes de production respectueux de l'environnement constituent un enjeu important pour l'Industrie 4.0. La consommation d'énergie devrait être optimisée afin de réduire les coûts de la production et la pollution.

L'utilisation d'une couche supplémentaire de Fog computing augmente logiquement le coût et la consommation d'énergie, mais une architecture basée sur le Fog peut être conçue pour réduire la consommation d'énergie en raison de l'aspect distribué du Fog Manufacturing.

Par conséquent, l'adoption d'une stratégie économique de consommation d'énergie est un défi à relever. La modélisation de l'architecture permettra d'optimiser le coût d'installation en terme de l'énergie consommé par chaque FNs

Consommation d'énergie VS QoS (5-7) :

Il faut assurer un équilibre et une stabilité entre les exigences des utilisateurs (qualité de service) et celles des fournisseurs (Consommation d'énergie). Pour augmenter la QoS, il faut intégrer des améliorations logicielles et matérielles. Cette couverture des besoins des applications va causer une augmentation de la consommation d'énergie.

Ce lien entre les deux défis Output nécessite un traitement équitable pour les surmonter. L'optimisation de la consommation énergétique du matériel assurant l'architecture basée sur le Fog manufacturing doit être prise en considération.

I VS QoS (6-7) :

L'interopérabilité des systèmes de production permettra de réduire le délai de transmission et de traitement. Le format utilisé sera compréhensible par tous les systèmes, ce qui réduira le délai d'échange entre la FN et les points d'extrémité, ce qui permet de relever le défi de la QoS.

Les défis sont corrélés, ce qui signifie que satisfaire certains autres défis facilitera leur résolution. Les quatre défis en entrée, comme présentés sur la Figure 3.5 influencent directement les défis en sortie : la QoS, le coût et la consommation électrique. Les liens entre les différents défis (Input et Output) sont définis afin de les satisfaire par la mise en œuvre du Fog Manufacturing.

3.4 Les possibles applications du fog computing dans le contexte de l'Industrie 4.0

Malgré les nombreux avantages du Fog computing, son adoption reste limitée dans le contexte d'Industrie 4.0. Les défis identifiés dans la section précédente représentent une barrière : le coût de la solution basée sur le Fog Manufacturing doit être optimisé ainsi que la consommation d'énergie. La couche du Fog doit maintenir un haut niveau de QoS sans oublier de considérer les caractéristiques du Fog computing (Hétérogénéité, Interopérabilité et programmabilité).

Un autre obstacle freinant l'adoption du Fog Manufacturing est estimé très important de notre point de vue : nous pouvons en effet considérer que les industriels n'ont pas une réelle évaluation précise de la valeur ajoutée du Fog Computing. Il manque une étude pour identifier les domaines où il est (clairement) avantageux d'intégrer les caractéristiques du Fog computing.

L'Industrie 4.0 regroupe un ensemble d'applications qui varient sur l'ensemble de la chaîne de production, qui peut concerner différentes étapes du cycle de vie (Beginning of life BOL, Middle of life MOL, End of life EOL). Chacune s'occupe d'un processus spécifique et se caractérise par des besoins matériels et logiciels spécifiques.

Nous introduisons ici un indicateur que l'on a choisi d'appeler le "**degré d'utilité**" pour mesurer l'adaptabilité du Fog computing pour chaque domaine d'application industriel. Cet indicateur est déterminé en fonction du nombre de services fournis ou besoins satisfais par les caractéristiques du Fog computing. Certains domaines d'application présentent plus de besoins que les autres, ce qui augmente les chances d'intégration du Fog Manufacturing.

Les **aspects de mesure** que l'on a identifiés (l'aspect temps réel, le traitement de données, la quantité de données, la mobilité, la distribution, l'hétérogénéité) sont choisis en fonction des caractéristiques du Fog computing adopté en général. Ces aspects de mesure sont utilisés dans des applications identifiées, certains auteurs les considèrent d'ailleurs comme étant des caractéristiques de base pour leurs cas d'études.

L'exploration de ces caractéristiques est très importante pour l'amélioration ou les travaux à venir. Le concept émergeant de l'Industrie 4.0 a un impact sur un grand nombre de services industriels, et il fait évoluer leur mode de fonctionnement et leurs besoins. Par exemple, le passage d'une application industrielle traditionnelle à la vision Industrie 4.0 exige l'addition de l'aspect temps réel ou la mobilité en cas

de présence de robots ou CPS mobiles. Cette amélioration pourrait augmenter les performances de l'application. Par conséquent, les applications choisies sont considérées comme des propositions de mise en œuvre du Fog Manufacturing.

À cette étape, nous avons passé en revue un grand nombre de travaux afin de faire correspondre les besoins des applications industrielles traditionnelles existantes aux caractéristiques du Fog computing identifiées (les aspects de mesure). Tant que le domaine d'application supporte un ou plusieurs aspects de mesure, le degré d'utilité s'incrémente.

Le tableau 3.2 donne quelques exemples de domaine d'applications industrielles où le Fog Manufacturing peut être mis en œuvre. Il présente également le degré d'utilité du Fog computing pour chaque application.

Ce tableau est constitué à partir de l'étude bibliographique afin d'identifier l'ensemble des applications potentielles. Pour chaque domaine d'application on a visé un groupe d'articles qui concerne l'application en domaine industriel et étudié ses besoins en se projetant dans le contexte de l'Industrie 4.0. La sélection des caractéristiques présentées est une conclusion des sections précédentes qui relie les caractéristiques du Fog computing avec les besoins des applications de l'Industrie 4.0.

Compte tenu de la migration vers le paradigme Industrie 4.0, une telle description permet d'adopter le Fog computing tout en améliorant l'ensemble des applications industrielles traditionnelles. Une application basée sur le Fog computing peut non seulement améliorer les performances du Cloud Manufacturing mais aussi ouvrir de nouveaux horizons en termes de distribution, d'interopérabilité, de flexibilité et de rapidité.

A l'aide de cette étude, nous avons sélectionné le domaine d'application de logistique sur lequel on va intégrer la solution du Fog Manufacturing comme il présente le degré d'utilité le plus élevé.

Pour la construction du tableau 3.2, nous nous sommes basés non seulement sur l'état de l'art mais aussi sur la conclusion des autres sections développées tout au long de ce chapitre.

TABLE 3.2 – Les applications possibles du Fog Manufacturing.

Application	Aspect temps réel	Traitement de données	Quantité de données	Mobilité	Distribution	Hétérogénéité	Degré d'utilité
Logistique	✓	✓	✓	✓	✓	✓	6
Consommation d'énergie	✓	✓	✓	✓	✓	✓	6
Traçabilité	✓	✓	✓	✓	✓	✓	6
La disponibilité des système de production distribué	✓	✓	✓	-	✓	✓	5
Localisation en intérieur	✓	✓	✓	✓	✓	-	5
Processus d'ingénierie virtuelle	✓	✓	✓	-	✓	✓	5
Gestion du cycle de vie des produits	✓	✓	✓	-	✓	✓	5
Gestion (Collecte/Filtrage/Suppression) de données et Prise de décision	✓	✓	✓	-	✓	✓	5
Gestion de l'inventaire	✓	✓	✓	✓	✓	-	5
Supervision	✓	✓	✓	-	✓	✓	5
Scheduling	✓	✓	✓	-	✓	✓	5
Diagnostic et pronostic	✓	✓	✓	-	✓	-	4
Interaction homme-machine/robot	-	✓	✓	✓	✓	-	4
Automatisation des tâches manuelles	-	✓	✓	-	✓	✓	4
Monitoring en temps réel	✓	✓	✓	-	-	✓	4
Sécurité	✓	✓	✓	-	✓	-	4
Maintenance prédictive	-	✓	✓	-	✓	✓	4
Inspection	✓	✓	✓	-	✓	-	4
Identification et prédiction des systèmes dynamiques	-	✓	✓	-	✓	-	3
Évaluation des risques	-	✓	✓	-	✓	-	3
Système de contrôle de la qualité	-	✓	✓	-	-	-	2

3.5 Conclusion

Dans ce chapitre, les différentes applications basées sur le Fog Manufacturing ont été présentées à travers des cas d'études de la littérature, de façon à analyser leurs intérêts dans notre contexte applicatif. Par la suite, nous avons constaté que l'adoption du Fog Manufacturing reste modeste à cause de nombreux défis à relever. Les défis ont été identifiés et analysés afin de pouvoir surmonter les limitations d'adoption du Fog Manufacturing. Sur le plan théorique, nous avons identifié les PIs pour chaque défi et nous avons établi les liens entre eux. Cela peut avoir une importance non négligeable si nous souhaitons optimiser les défis simultanément. Comme perspectives, ces liens peuvent être écrits sous forme d'équations séquentielles afin de définir des probabilités bien déterminées, et pouvoir mesurer chaque défi. Nous avons ainsi proposé plusieurs applications pour l'Industrie 4.0 à travers lesquelles il est possible d'intégrer le Fog computing selon un certain degré d'utilité.

Dans la suite de ce mémoire, nous allons nous attacher à détailler notre cas d'étude en proposant un modèle d'optimisation des coûts d'installation de l'architecture basée sur le Fog Manufacturing. Cette solution sera par la suite implémentée afin d'évaluer ses performances vis-à-vis des critères comme la consommation d'énergie et le délai.

Chapitre 4

Déploiement optimal du Fog Computing pour la logistique interne dans le contexte de l'Industrie 4.0

4.1 Introduction

Après avoir examiné dans le chapitre 2 les travaux de la littérature sur les applications de l'Industrie 4.0 intégrant une architecture basée sur le Fog computing, nous proposons un modèle d'optimisation du coût du déploiement d'un réseau à plusieurs niveaux contenant une couche Fog. Ce réseau est utilisé pour la gestion d'un atelier de production flexible (usine 4.0) contenant une flotte de robots mobiles et des machines intelligentes. Ceci représente un support pour l'utilisation des données pour la logistique interne dans un exemple d'usine 4.0 basée sur le Fog Manufacturing.

Dans ce chapitre, nous présentons dans un premier temps les travaux qui ont porté sur l'optimisation du coût de déploiement d'une architecture basée sur le Fog Manufacturing. Dans une deuxième partie, nous allons concevoir l'architecture adoptée lors de la modélisation du déploiement. Finalement, nous allons spécifier le modèle d'optimisation du coût, en détaillant les paramètres, toutes les contraintes adoptées et le résultat d'optimisation selon l'instance.

Ce problème devient encore plus complexe quand il s'agit d'entités intelligentes, communicantes et mobiles. C'est un problème d'actualité concernant l'introduction des IoTs dans le domaine industriel. Il faut noter que le problème occupe une importance majeure pour les industriels qui sont en phase de passage vers l'industrie 4.0 et la communauté de recherche à travers l'apparition de recherches similaires récemment (LIN et YANG, 2018YEH, LAI et TSAI, 2019). La décision de dimensionnement du réseau (nombre de ENs, FNs et passerelles et les types de liaison qu'il faut prendre en compte) influence en grande partie le coût d'investissement sur la nouvelle solution basée sur le Fog. Le type de décision rentre dans le domaine de décision stratégique ou tactique qui ne peut pas être changé tous les jours et qui affecte le fonctionnement de tout l'atelier de production pour une longue période.

4.2 La logistique dans le contexte de l'Industrie 4.0

4.2.1 Contexte

Bien que le Cloud Computing offre des ressources de calcul élevées rendant le traitement des données extrêmement efficace, le mouvement des données qui en résulte pourrait avoir un impact sur les performances de l'application en raison d'éventuelles latences importantes de la transmission. Pour améliorer ce type d'applications, une bonne gestion de l'information devient fondamentale pour livrer l'information au bon moment, au bon endroit, avec une bonne qualité et avec un coût minimal (BILL, 2016).

La logistique, en tant que partie importante du processus industriel, doit s'adapter et profiter de la succession des révolutions technologiques telles que l'IoT le cloud et la vision industrielle. Le processus logistique à l'intérieur de l'atelier implique de nombreux types de systèmes connectés tels que des AGV (véhicules guidés automatisés, qui suivent des routes pour transporter des matériaux et des produits dans l'atelier de production), des machines de production, des postes manuels, des stocks, des tablettes d'opérateurs, etc.

Les manières et technologies ordinaires qui soutiennent la logistique industrielle ne sont plus suffisantes pour les exigences de l'Industrie 4.0. En effet, le système de pilotage est amené à prendre de multiples décisions d'une manière complètement autonome sans pour autant dévier l'objectif global de l'atelier. C'est pourquoi l'intégration des fonctions du Fog Manufacturing est bénéfique (BOUZARKOUNA et al., 2018) vue la capacité de calcul au sein du réseau qu'il peut offrir. Compte tenu de l'importance d'une telle application pour le pilotage du transfert interne, l'Industrie 4.0 exige que les services logistiques deviennent de plus en plus intelligents, efficaces et réagissent en temps réel pour répondre au niveau élevé d'intelligence artificielle.

L'ajout des caractéristiques du Fog Manufacturing aidera à prendre rapidement la décision la plus adéquate pour améliorer les opérations logistiques. En connectant les serveurs du Fog au Cloud computing, on obtiendra davantage d'analyses qui permettront de gérer les futures tâches logistiques.

Pour intégrer le Fog Manufacturing, de nombreux aspects doivent être étudiés, en particulier la modélisation de l'architecture de l'application et l'optimisation du coût de l'installation d'une telle solution, avec la prise en compte des contraintes liées à la latence, la distribution et les limites de capacité de traitement et d'acheminement de l'information. Certains travaux s'intéressent à ce domaine de recherche comme (QU et al., 2016), qui a présenté l'intégration Cloud Manufacturing dans la logistique de la production. Parmi les nombreuses applications de l'Industrie 4.0, présentées dans le chapitre 3 précédant, l'intégration du Fog Manufacturing pour optimiser les performances d'une application de logistique est présente.

4.2.2 Problématique du déploiement du Fog Manufacturing

L'intégration du Fog Manufacturing est certes un investissement pour les industries, mais elle doit être maîtrisée en termes de performance, de coût et de consommation énergétique. La solution Fog est non seulement composée des serveurs Fog mais également des liens avec toutes les couches de l'architecture. Chacune est constituée de nombreux noeuds connectés qui sont déjà installés (terminaux ou Cloud) ou à installer (des noeud edge EN, des noeud Fog FN et des passerelles).

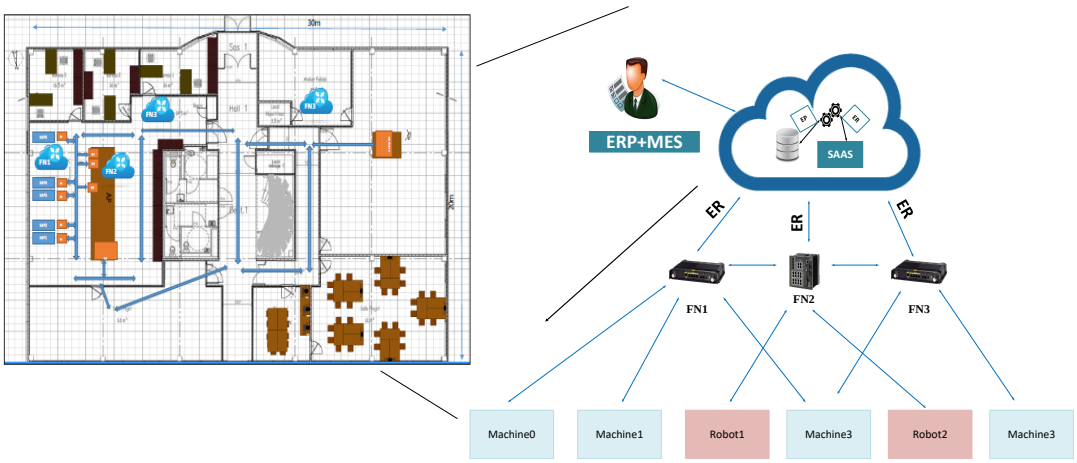


FIGURE 4.1 – Le processus de décision opérationnelle basé sur le Fog Manufacturing.

Le déploiement des noeuds constituant une architecture basée sur le Fog Manufacturing est une étape primordiale afin de préciser le coût d'installation et le coût de fonctionnement d'une telle architecture à long terme.

Une usine 4.0 est composée de nombreux systèmes connectés, pouvant être mobiles, qui assurent un bon déroulement de la production. Les AGVs par exemple sont importants pour la logistique et le transport des pièces entre les différents départements dans une usine, ils sont caractérisés principalement par la mobilité et la communication avec le système de supervision et les systèmes de production connectés. Un AGV doit être présent dans une zone incessamment couverte afin de ne pas interrompre la transmission des requêtes/réponses et également la bande passante doit être suffisamment large pour supporter le grand nombre de transmissions simultanées.

Afin de soutenir un bon niveau de réactivité en temps réel, il est nécessaire d'adopter une stratégie de déploiement des noeuds de l'architecture réseau afin de satisfaire les besoins en termes de couverture, bande passante et du coût d'installation et de fonctionnement. Un autre aspect à considérer lors du déploiement des noeuds, est de ne pas surcharger les réseaux par les requêtes des noeuds de communication et priver les utilisateurs de la rapidité à l'accès aux services principaux liés à la production.

Conséquemment, le déploiement des noeuds d'une architecture basée sur le Fog Manufacturing est une problématique d'importance majeure et complexe à la fois. Le dimensionnement du réseau concerne principalement le nombre des EN, FN et passerelles à installer ainsi que le type de lien adéquat pour les installations des systèmes. La Figure 4.1 présente un exemple de déploiement de l'architecture Fog Manufacturing, et présente un exemple des différentes parties qui peuvent être impliquées. Cette figure présente le processus de la prise des décisions opérationnelles à court terme. On remarque que chaque niveau présente un type de communication spécifique avec le niveau supérieur. Par conséquent, avoir spécifié le type des dispositifs et liens utilisés selon le besoin est nécessaire afin d'optimiser la solution basée sur le Fog. Les composants d'une telle architecture doivent couvrir les besoins de l'atelier de production et assister à la prise des décisions. Les décisions sont classées en trois catégories au niveau des industries, expliquées ci-dessous :

— Stratégiques (Long terme et coût important) : uniques, imprévisibles et irréversibles (changement coûteux) elles concernent l'ensemble de l'usine.

Exemples : fabrication d'un nouveau produit, suppression d'une activité, fusion avec un autre domaine de production.
— Tactiques (Moyen terme et coût moyen) : ces décisions concernent la gestion ou le pilotage des tâches de production peu fréquentes et peu prévisibles. Exemples : acquisition de matériel de production, recrutement.
— Opérationnelles (Court terme) : elles sont fréquentes et répétitives. Exemples : organisation des horaires des tâches, achat des matières premières, planification.

Le Fog manufacturing permet de gérer les décisions opérationnelles en temps réel tout en traitant d'une manière locale les demandes et décide d'une façon distribuée afin de fournir des solutions à court terme aux CPS de l'atelier de production. Les décisions concernant le déploiement des noeuds de l'architecture basée sur le Fog Manufacturing rentrent dans le domaine des décisions stratégiques ou tactiques qui ne peuvent être changées souvent et qui affectent le fonctionnement de tout l'atelier de production pour une longue période.

Si on opte pour la quantité, c'est à dire un grand nombre de noeuds, on va avoir une large couverture ce qui est sensée faciliter le traitement, mais en contre partie, on risque d'augmenter le coût et de surcharger le réseau par des noeuds connectés (Un grand nombre de transmissions et une haute consommation d'énergie) et sous exploités. La qualité est la deuxième option d'installation, ou on trouve beaucoup moins de noeuds qui ont plus de capacité de connexion et de couverture. Dans ce cas, on gagnera certes en terme de coût et d'installation mais on risque d'augmenter les délais de prises de décision et de limiter la flexibilité de l'architecture. C'est pourquoi le placement de différents composants de l'architecture est une décision à la fois stratégique et tactique qui mène à une meilleure prise des décisions opérationnelles. Par exemple, le déploiement peut affecter la couverture réseau des AGVs circulant dans l'atelier : si on sur-déploie des ENs on aura un coût élevé dû à l'achat d'un grand nombre de noeuds EN. Par contre si l'on sous-déploie des ENs, les AGVs vont traverser des zones non couvertes ce qui peut générer des risques de perte d'information et de prise de décision à l'aveugle. La figure 4.2 présente les trois cas possibles de déploiement pour les ENs tout en prenant en compte la mobilité des robots. Les couleurs de cercles de couverture distinguent les différents ENs.

Un autre exemple, qui montre l'importance et la délicatesse d'une telle problématique, est la liaison avec la consommation d'énergie des noeuds installés. Il faut bien étudier les besoins de chaque zone en fonction de l'ensemble de ces systèmes connectés. Dans une usine, il existe des zones de faible productivité ou de faible valeur ajoutée. Par conséquent il ne sera pas optimal d'investir dans cette zone et il est préférable de consacrer du budget dans les zones où le besoin est le plus important. Sur la figure 4.3, on remarque la présence de la même concentration des noeuds sur l'ensemble de l'usine. Par contre chaque zone (I, II et III) possède une activité différente. Sur les zones I et II, la présence massive des noeuds est utile comme il existe une productivité et mobilité importantes. Par contre la zone III (entourée du cercle jaune) ne possède qu'un poste manuel, donc les noeuds présents consomment de l'énergie sans sollicitation. Nous pouvons conclure qu'un mauvais déploiement aura des conséquences non seulement à court terme (Coût d'installation) mais aussi à long terme. Le nombre et la position des noeuds aura aussi un effet sur le coût de la maintenance. La multiplication des noeuds augmente la probabilité d'intervention et ainsi les coûts de maintenance associés.

De nombreux travaux s'intéressent à l'optimisation du coût et du placement des noeuds constituant le réseau du Fog Manufacturing. Les travaux de (Gu et al., 2015) se sont intéressés aux CPS médicaux (MCPS). Pour répondre à la haute QoS des

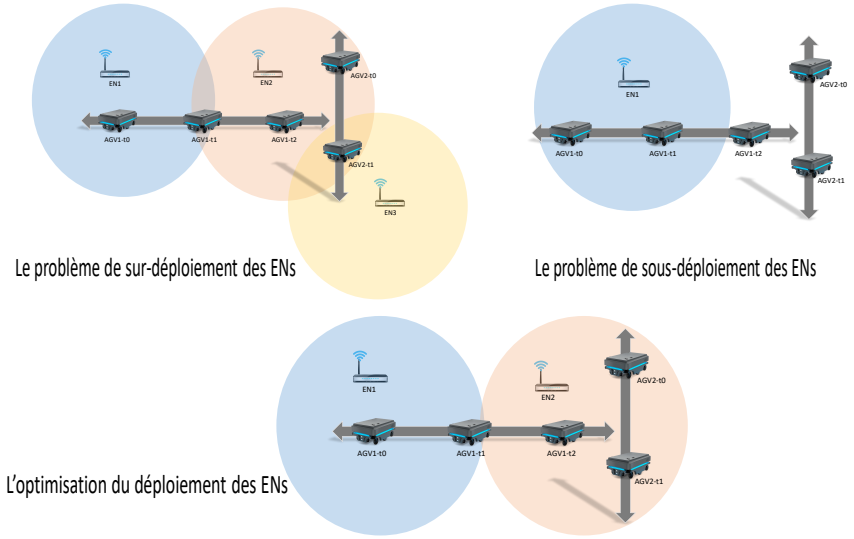


FIGURE 4.2 – Le problème de déploiement des ENs en terme de mobilité.

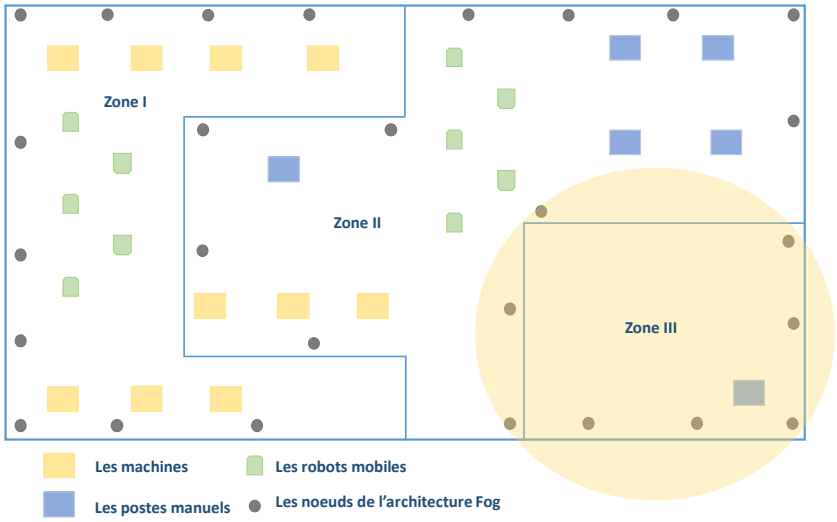


FIGURE 4.3 – Le problème de déploiement des noeuds en terme de consommation d'énergie.

MCPS et pour alléger la couche Cloud, ils proposent d'intégrer le Fog computing (FC-MCPS) et d'optimiser la gestion des ressources et la QoS. Un modèle d'optimisation du coût de la solution FC-MCPS est présenté tout en prenant en compte les contraintes de l'association des noeuds de communication et de calcul (FN) et la distribution des tâches. Afin d'atténuer la complexité du modèle, ils ont linéarisé sous la forme d'un problème de programmation linéaire mixte à nombres entiers (MILP). Les résultats obtenus par l'heuristique proposés par les auteurs ont montré des performances proches de la solution optimale.

Nous avons trouvé d'autres travaux qui ont étudié la même problématique tout en optimisant le coût de l'intégration du Fog computing comme dans (YAO et al., 2017 et WAN et al., 2018). Le déploiement des serveurs était optimisé afin d'améliorer la QoS.

Les travaux de (LIN et YANG, 2018) sont considérés comme les plus proches de notre cas d'étude comme ils s'intéressent au déploiement du Fog computing dans une application de logistique dans un contexte de l'Industrie 4.0. Leur but est d'optimiser le coût d'intégration du Fog computing (FN, EN et passerelles) sous les contraintes de capacité maximale de la demande, de latence maximale, de couverture maximale et de nombre maximal des liens. La résolution est faite à l'aide d'un algorithme méta-heuristique (un algorithme de singe discret génétique). Les résultats de la simulation ont montré que l'algorithme DMGA (Discrete Monkey Genetic Algorithm) proposé donne de bons résultats dans les cas problématiques des centres logistiques de taille moyenne.

Pour la majorité de ces travaux, le coût et l'emplacement des noeuds activés étaient les termes les plus importants à prendre en compte et à optimiser. Par contre, il serait intéressant d'étudier les effets de déploiement sur les flux dynamiques des systèmes et des produits. Pour chaque partie de l'atelier il existe une dynamique (répétitive ou imprévue) des différentes tâches de production. Un déploiement basé sur le Fog aide à optimiser la mobilité, la QoS en terme de latence, de consommation d'énergie, etc,...

La mobilité, par exemple, est un enjeu fondamental dans le contexte de l'Industrie 4.0. De nombreux appareils mobiles (AGVs, Drônes, tablette d'opérateur,...) existent dans l'usine pour assurer la logistique et le déplacement des produits et des outils entre les différents départements et cellules de production. Chaque zone de l'usine possède des allées et corridors bien définis pour stabiliser les temps de déplacement et éviter les collisions. Chaque zone possède une probabilité de présence des AGVs selon le besoin et le volume de chacune.

Ainsi, les différents noeuds constituant l'architecture du Fog Manufacturing sont répartis dans l'ensemble de l'atelier. La prise en considération des emplacements initiaux des noeuds est importante comme cela facilite l'installation de certains noeuds après l'optimisation du déploiement. Par exemple l'emplacement de :

— L'alimentation électrique (prises) ou l'on peut brancher les FN et les EN,
— Les trajectoires fixes des déplacements des AGVs dans l'atelier.

Pour traiter les cas problématiques à plus grande échelle et avec un grand nombre des noeuds, il serait intéressant de proposer de nouvelles contraintes qui renforcent celles utilisées, pour avoir une optimisation plus efficace et réaliste. Les facteurs influençant le déploiement doivent également être examinés, par exemple le type de lien entre les différentes couches, la probabilité de présence des AGVs dans une zone bien définie, la consommation d'énergie, l'équilibre des charges, l'hétérogénéité, l'équité et la qualité de service (QoS).

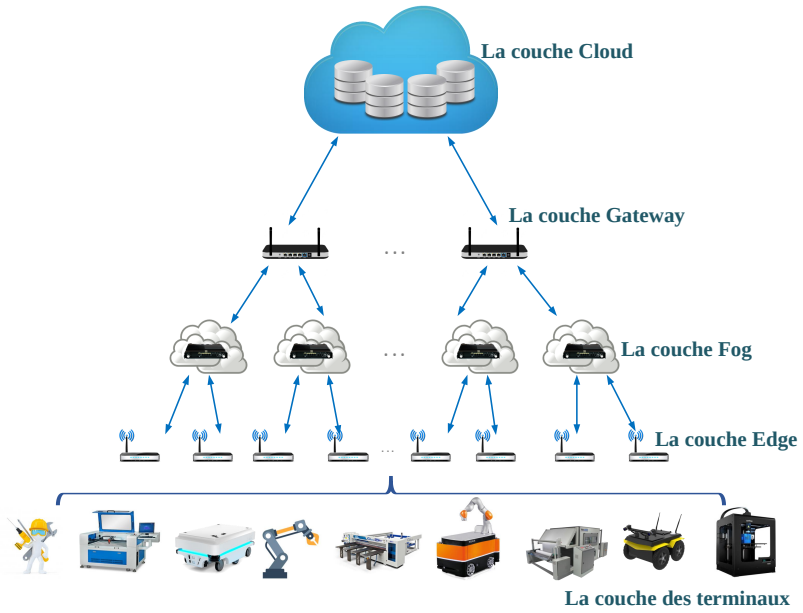


FIGURE 4.4 – L'architecture basée sur le Fog Manufacturing.

4.3 L'architecture du Fog Manufacturing

Dans cette section, nous présentons l'architecture basée sur le Fog Manufacturing que nous allons adopter tout au long de ce chapitre. Cette architecture ainsi que les liens entre ces différentes couches superposées, va être modélisée afin d'optimiser le coût de déploiement de ces noeuds hétérogènes.

Dans notre travail nous avons choisi l'architecture évoluée qui comporte 5 niveaux (les terminaux connectés, les noeuds Edge, les serveurs Fog, les passerelles et la couche Cloud). Les raisons derrière ce choix sont :

— La mobilité de certains CPS présents dans l'atelier de production,
— La distribution à grand échelle des CPS dans l'atelier,
— La nécessité de contrôle de la quantité de données transmise entre deux niveaux.

Cette architecture fournit des fonctions de calcul, de réseau et de stockage plus complexes et plus étendues. Les nœuds de l'architecture Fog intègrent des fonctions de distribution, de coordination, de gestion et de protection des ressources de réseau croisées, ce qui assure les services du Fog Manufacturing.

Les cinq couches communiquent entre elles pour assurer une transmission de données du niveau le plus bas jusqu'au Cloud. La couche Fog est adoptée pour assurer un calcul en temps réel distribué tout le long de l'atelier de production. La figure 4.4 présente l'architecture à cinq niveaux basée sur le Fog dans le contexte de l'Industrie 4.0.

La couche des terminaux

Cette couche est la plus basse, elle regroupe un grand nombre des terminaux hétérogènes et distribués. Dans notre contexte de l'application de logistique de l'Industrie 4.0, les terminaux sont les CPSs ou des dispositifs IoT :

— Les AGVs,
— Les bras robotiques,

— La chaîne de production (les machines),
— Les postes manuels,
— Les machines de découpes laser et les imprimantes 3D,
— Les capteurs de présence, les caméras,
— ...

Les CPS sont équipés des capteurs qui visent à détecter/surveiller le processus de production. Chaque catégorie de CPS présente dans l'usine occupe un rôle spécifique.

Par exemple, les AGV permettent de planifier et gérer les tâches de logistique, le traitement des images et la navigation automatique, intégrant des répartitions des tâches, le transport des pièces ou produits entre les différents département et le calcul de l'inventaire à reconstituer à l'entrepôt. Les AGVs communiquent avec les autres CPS soit directement soit à travers l'architecture basée sur le Fog Manufacturing. Ils transmettent leurs demandes régulièrement afin de recevoir les données nécessaires à la prise de décision en temps réel.

Les bras robotiques sont généralement programmables, avec des fonctions de préhension et de placement de produits. Les segments de ces bras sont reliés par des axes permettant, soit des mouvements de rotation et/ou de translation. Ces mouvements sont nécessaires pour l'exécution des tâches répétitives et exigeant une haute précision. Ces robots peuvent être seuls ou fixés sur des AGVs pour être mobile et exécuter les tâches dans toute l'usine.

L'Industrie 4.0 est caractérisée par la mobilité et la personnalisation. C'est pour cela que des machines de fabrication additive, imprimantes 3D et des machines de découpes laser peuvent être présentes dans les usines. Ces CPS sont connectés et équipés des capteurs. Ils envoient d'une façon continue leur données (état, tâche, panne, disponibilités,etc.) et reçoivent des instructions ou décisions afin d'organiser les tâches selon la priorité et la quantité.

Les postes manuels permettant par exemple des opérations d'assemblage ou de contrôle, ont des emplacements fixes pendant l'activité de production mais peuvent être déplacés afin de réorganiser les flux et les configurations de production au sein de l'atelier.

Les machines sont souvent connues par la contribution directe au processus de production. Elles sont capables de produire, mais aussi de servir à conditionner, transformer et façonner des matières afin d'obtenir un produit fini. Les machines s'appuient sur tous les autres CPS cités ci-dessus afin de répondre aux exigences de l'Industrie 4.0 en termes de qualité, rapidité et efficacité.

Les appareils mobiles utilisés par les opérateurs, représentent aussi des noeud CPS. A travers ces interfaces, les opérateurs envoient leurs rapports, souhaits et commandes. Les personnels sont également dotés de terminaux mobiles pour effectuer la supervision, la gestion, la maintenance et la prise de décisions stratégiques, tactiques ou opérationnels en cas d'urgence. Ces équipements doivent être dans des zones couvertes et connectés au système d'information de l'atelier à n'importe quel moment.

La couche du Edge computing

La couche du Edge est constituée des ENs, ils ne sont pas seulement utilisés pour la transmission de données des CPS aux FNs, mais fournissent également aux AGVs et autres CPS mobiles des réponses en temps réel lorsqu'ils rencontrent d'autres AGV sur la trajectoire du mouvement. Donc les ENs assurent la couverture réseau pour les systèmes mobiles afin de garantir la non rupture des échanges de données.

Les données nécessaires à la prise de décision sont identifiées et collectées par les ENs, et sont transmises aux FNs qui stockent et traitent temporairement les données, et renvoient les décisions aux CPS. Dans le même temps, les données sont également transmises aux serveurs de la couche cloud pour permettre par exemple des analyses basées sur les données d'historique et un stockage à long terme. Il existe plusieurs types de PA :

— Point d'accès sans fil,
— Modem,
— Routeur,
— Commutateur réseau (Switch),
— Concentrateur réseau
— Boitier Courants porteurs en ligne (CPL).

La plupart de ces équipements ont une autre finalité que le simple point d'accès. Par exemple, ils peuvent relier deux réseaux disjoints. Ces ENs sont multi-fonctions et généralement de type Wi-Fi IEEE 802.11n ou IEEE 802.11ac.

La couche du Fog computing

L'introduction de la couche Fog permet aux CPS connectés d'échanger des informations pour coordonner les opérations. Les FNs sont déployés sur le bord du réseau et ils réalisent l'interaction entre les noeuds, la mise en cache des informations et le traitement des tâches, ainsi que certaines autres fonctions en périphérie du réseau. Les FNs s'occupent de la collecte des données générées par les capteurs. Certaines de ces données concernent le contrôle nécessitant un traitement en temps réel (ms). Ce premier niveau du Fog est conçu pour l'interaction machine-machine, la collecte, le traitement des données et l'émission des commandes aux actionneurs. Le Fog filtre également les données à utiliser localement et envoie le reste aux niveaux supérieurs. Toutes les interactions du Fog sont en temps réel (de quelques millisecondes à quelques secondes). En conséquence, le Fog fournit plusieurs types de stockage, de l'éphémère (niveau le plus bas) vers le semi-permanent (niveau le plus élevé, la couverture géographique est plus élevée et l'échelle du temps est plus large). A la fin, la couverture mondiale est fournie par le Cloud, qui est utilisé comme un entrepôt de données permanent (mois, années). La figure 4.5 illustre cette interaction entre les deux modèles de calculs.

Les FNs peuvent être des serveurs de calcul intégrés à des routeurs et destinés au déploiement industriel. Il existe de nombreuses technologies commercialisées qui permettent l'implémentation et l'exploitation de la solution Fog Manufacturing (DASTJERDI et BUYYA, 2016). Nous allons en présenter quelques unes dans ce qui suit :

— **LocalGrid (plateforme Fog)** : Est un logiciel embarqué installé sur des périphériques réseaux (commutateurs, routeurs) et des capteurs. Il standardise et sécurise les communications entre tous les types de périphériques, ce qui minimise les coûts des services. Il réside entre le bord et le Cloud, ce qui permet aux applications de prendre des décisions en temps réel au bord de réseau, tout en éliminant la latence élevée causée par la communication avec le Cloud.
— **ParStream** : Est une plate-forme d'analyse IoT en temps réel. Elle s'intègre aux équipements Cisco pour créer une infrastructure rapide, fiable et hautement évolutive pour l'analyse du Fog, comparée à l'infrastructure traditionnelle. Étant une base de données en mémoire, elle est idéale pour les dispositifs Fog. Elle peut repousser l'exécution de la requête au bord du réseau où

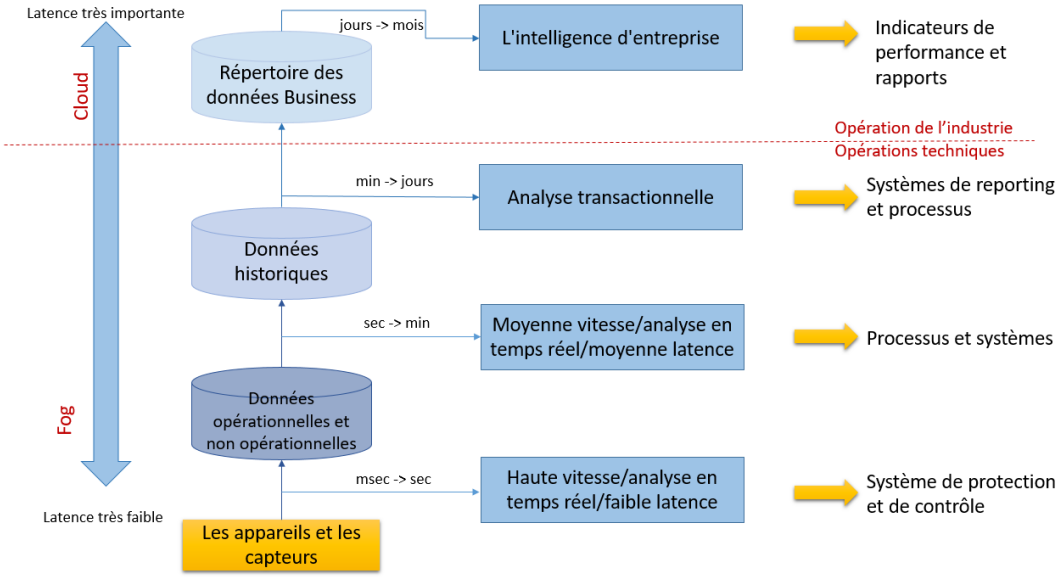


FIGURE 4.5 – Les trois types d'intégrations de l'Industrie 4.0.

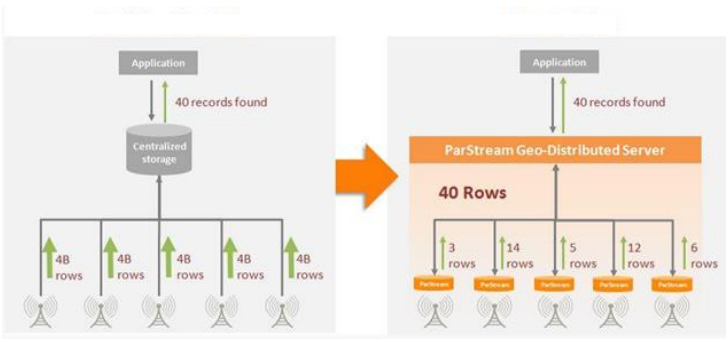


FIGURE 4.6 – Le principe de la technologie ParStream.

les données sont produites et effectuer des analyses de manière hautement distribuées comme l'explique la figure 4.6.

— **Cisco Data In Motion (DMo)** : Est une technologie offrant la gestion et l'analyse des données au bord du réseau. Elle est utilisée pour effectuer des analyses sur les données entrantes (la recherche de données spécifiques d'intérêt, le résumé des données, la génération de nouveaux résultats) et destinée à être déployée sur des périphériques de manière distribuée et à contrôler les flux importants de données provenant des appareils IoT.

— **Vortex** : Est une plate-forme de partage (interopérable, évolutive, efficace, sécurisée et en temps réel) des données pour l'IoT, du bout en bout à travers les périphériques, les bordures, les passerelles et le Cloud. Vortex offre des exigences de sécurité, d'authentification et de confidentialité et ses dispositifs Vortex Edge appartiennent au domaine Fog. La figure 4.7 présente une illustration du principe du Vortex.

La couche du passerelle

Cette couche relie deux réseaux informatiques de types différents, le local (Fog) et le réseau Internet (Cloud). Elle joue principalement le rôle d'une intermédiaire

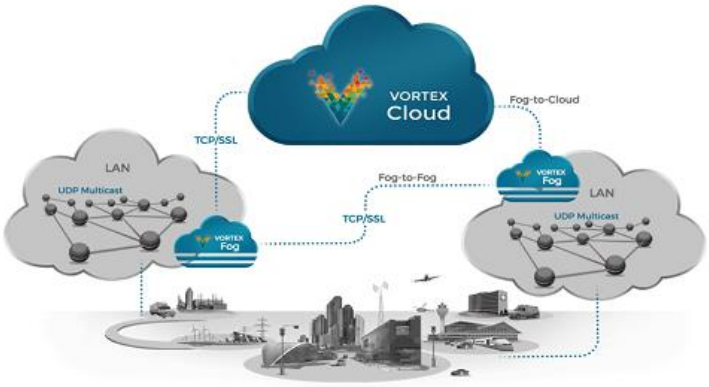


FIGURE 4.7 – Le principe de la technologie Vortex.

entre les FNs et les serveurs du Cloud. Les passerelles concentrent les informations localement reçues de la couche Fog, les regroupent et les transmettent via Internet au Cloud computing.

Lorsqu'un FN contacte une passerelle, ce dernier examine sa requête et, si jamais celle-ci correspond aux règles que l'administrateur réseau a définies, la passerelle crée un lien entre les deux couches. Les données ne sont donc pas directement envoyées, mais interprétées afin d'assurer la continuité des deux réseaux possédant deux protocoles différents.

Cet intermédiaire facilite la transmission malgré l'hétérogénéité et offre une sécurité supplémentaire (pare-feu, proxy, etc.). Cette sécurité est due à la vérification de chaque donnée transmise, ce qui entraîne des délais supplémentaires pouvant causer un ralentissement.

Les passerelles peuvent être logicielles ou matérielles. Dans notre contexte, on considère que les passerelles de l'architecture basée sur le Fog Manufacturing sont matérielles et qu'elles peuvent être de trois niveaux :

— Un répéteur : Niveau 1,
— Un pont : Niveau 2
— Un relais, nommé aussi routeur : Niveau 3.

La couche du Cloud computing

Cette couche est la plus élevée et elle regroupe des serveurs avec des importantes capacités de calcul et de stockage. Ces serveurs se chargent de la synchronisation avec les dernières données transmises par les FNs. Cette couche assure les mises à jour et la prise de la décision. Avec la présence des FNs, le Cloud prend les décisions à long terme et s'occupe des calculs qui nécessitent des méthodes d'apprentissage (notamment d'apprentissage profond) et des techniques de Big Data. Donc le Cloud peut assurer les décisions non urgentes et fédérées. Généralement le Cloud computing est un ensemble des serveurs (publics ou privés) localisés à l'extérieur du site de production. Cependant, pour certaines entreprises ou sites de production, les entreprises préfèrent garder les serveurs en interne pour sécuriser leurs données. Il existe bien évidemment d'autres solutions comme la location des serveurs. Depuis plusieurs années, l'alternative de location des serveurs est proposée aux entreprises, afin de stocker et sauvegarder les données. Il est ainsi possible de louer des capacités de calcul et de stockage chez des hébergeurs donnant ainsi une grande flexibilité en

fonction de l'évolution des besoins de l'entreprise. Cette alternative est choisie parce qu'elle est moins coûteuse et assure le maintien des serveurs par des spécialistes.

4.4 L'optimisation du coût de déploiement du Fog Manufacturing

Dans cette section, nous nous intéressons au déploiement du Fog Manufacturing dans une application de logistique interne. Ce déploiement est basé sur l'architecture illustrée par la figure 4.4 dans une usine 4.0.

Pour modéliser le problème de déploiement du Fog Manufacturing, un ensemble d'hypothèses de travail est considéré pour limiter la complexité du problème étudié. Le problème étudié dans le cadre de cette modélisation s'appuie sur les hypothèses et caractéristiques suivantes :

- Dans cette usine, il existe un Cloud centralisé dont l'emplacement est connu et fixé,
- L'emplacement des systèmes connectés fixes (bras robotisés, machines, postes manuels, imprimantes 3D et découpes laser) est connu,
- Les trajectoires des AGVs mobiles sont fixées et identifiées, les zones de leurs déplacements sont définies lors de leur installation. Chaque AGV ne peut pas être présent dans une zone en-dehors des trajectoires pré-définies.
- Compte tenu de ce qui précède, l'emplacement des passerelles, des FNs, et des ENs sont inconnus, uniquement les emplacements candidats sont connus,
- Les terminaux ou CPS seront connectés aux ENs via WiFi,
- L'architecture de ce réseau, c'est-à-dire comment connecter les noeuds est inconnue. Le lien entre les ENs, FNs, passerelles et Cloud est filaire,
- L'environnement de cette usine est considérée comme homogène et chaque système possède une capacité de génération et de consommation de donnée définie.
- Le nombre de connexions de chaque noeud est supposé défini,
- Chaque système peut être connecté à un seul EN à la fois, et respectivement chaque EN, FN, passerelle à un FN, passerelle Cloud,
- Le problème à optimiser est étudié sous l'angle d'une minimisation du coût d'installation de la solution Fog Manufacturing.

Le modèle de minimisation du coût lié au déploiement est construit en utilisant les FNs. Ce modèle permet non seulement de réduire le coût d'installation et de fonctionnement, mais reflète également l'effet de cette optimisation sur la répartition des noeuds sur les différentes zones de l'atelier et sur la QoS.

La contribution principale de notre modèle pour l'optimisation efficace est présentée dans cette partie et porte sur les points suivants :

- La description de la fonction « objectif »,
- La définition des différentes contraintes,
- La résolution du problème de la non-linéarité des contraintes,
- La spécification des liens en donnant le choix entre de la fibre optique ou le câble RJ45, en fonction de la charge maximale de calcul et le nombre de noeuds connectés à chaque EN, FN ou passerelle,
- L'introduction de la probabilité de la présence des AGVs selon les zones de circulation.

4.4.1 Paramètres

Le problème du déploiement du Fog a été présenté et défini de façon formelle au paragraphe précédent. Les notations utilisées dans le modèle sont indiquées dans le tableau 4.1. Les tableaux 4.2 et 4.3 présentent respectivement les variables de décision et les vecteurs de variables.

TABLE 4.1 – Définition des paramètres.

Paramètre	Définition
Ω_C	L'ensemble des serveurs Cloud
Ω_G	L'ensemble des sites potentiels des passerelles
Ω_F	L'ensemble des sites potentiels des FNs
Ω_E	L'ensemble des sites potentiels des ENs
S	L'ensemble des terminaux
C_f	Le coût d'une unité de fibre
C_{RJ}	Le coût d'une unité de câble RJ45
C_G	Le coût de l'installation d'une passerelle
C_F	Le coût de l'installation d'un FN
C_E	Le coût de l'installation d'un EN
H_c^G	La capacité maximale d'une passerelle c
H_m^F	La capacité maximale d'un FN m
H_n^E	La capacité maximale d'un EN n
N_G	Le nombre maximal de liens d'une passerelle
N_F	Le nombre maximal de liens d'un FN
N_E	Le nombre maximal de liens d'un EN
d_{ij}	La distance entre deux nœuds i et j
L	La longueur des trames de données d'un terminal à un EN (Identique $\forall$ terminales)
L_E	La longueur des trames de données d'un EN à un FN
L_F	La longueur des trames de données d'un FN à une passerelle
R_n	Le diamètre de couverture d'un EN n
r	La charge de travail d'un terminal
μ_{tn}	Le débit de transmission de données d'un terminal t à un EN n
μ_{nm}	Le débit de transmission de données d'un EN n à un FN m
μ_{mc}	Le débit de transmission de données d'un FN m à une passerelle c
γ_t^n	La latence maximale d'une liaison entre un terminal t et un EN n
γ_n^m	La latence maximale d'une liaison entre un EN n et un FN m
γ_m^c	La latence maximale d'une liaison entre un FN m et une passerelle c
$MaxCapR$	La capacité maximale pour l'utilisation du câble RJ45
H_P^E	La capacité de réseau maximale

4.4.2 La fonction objectif

La résolution du modèle a pour objet, dans notre travail, de minimiser un objectif : Le coût de déploiement des noeuds de l'architecture basée sur le Fog Manufacturing. Ainsi, résoudre ce problème consiste à trouver une solution S qui minimise le nombre des ENs, FNs et passerelles utilisés, à préférer la solution avec la plus petite distance totale en fibre tout en favorisant l'utilisation du câble RJ45 lorsque c'est suffisant. Pour surmonter le problème de la non-linéarité des contraintes nous

TABLE 4.2 – Définition des variables de décisions.

Variable de décision	Définition
x_{ij}	Variable binaire, décide si y aura un lien fibre entre un nœud i et j
xR_{ij}	Variable binaire, décide si y aura un lien câble RJ45 entre un nœud i et j
xF_{ij}	Variable binaire, décide si y aura un lien câble RJ45 entre un nœud i et j
G_c	Variable binaire, indiquant la sélection des noeuds c activés des passerelles
F_m	Variable binaire, indiquant la sélection des noeuds m activés des FNs
E_n	Variable binaire, indiquant la sélection des noeud n activées des ENs

TABLE 4.3 – Définition des variables.

Variable	Définition
$\omega(X_i)$	La demande de capacité de calcul pour un site de noeud X_i
$\omega_P(X_{ij})$	La capacité de réseau pour un site de noeud X_i

avons divisé les variables de décisions en sous-ensembles afin de faciliter le calcul. Le détail de cette démarche sera expliqué dans la section 4.4.4.

$$\begin{aligned} Cost = \quad & C_f . \sum_{(i,j)\in\Omega_G \cup \Omega_F \times \Omega_F \cup \Omega_E} xF_{ij}.d_{ij} \\ & + C_{RJ} . \sum_{(i,j)\in\Omega_G \cup \Omega_F \times \Omega_F \cup \Omega_E} xR_{ij}.d_{ij} \\ & + C_G . \sum_{c\in\Omega_G} G_c + C_F . \sum_{m\in\Omega_F} F_m + C_E . \sum_{n\in\Omega_E} E_n \end{aligned} \tag{4.1}$$

Dans cette formulation, l'équation (4.1) modélise la fonction objectif du problème :

— Le premier terme est le coût des liaisons par fibre optique entre Cloud-passerelles, entre passerelles-FNs et entre FNs-ENs ;
— Le deuxième terme est le coût des liaisons par câble RJ45 entre Cloud-passerelles, entre passerelles-FNs et entre FNs-ENs ;
— Les trois autres termes sont les coûts d'installation des passerelles, des FNs et des ENs, respectivement.

4.4.3 Les contraintes de base

De multiples contraintes viennent également limiter l'espace de décision dans lequel la solution d'un problème de déploiement doit être trouvée. La partie suivante détaille les contraintes adoptées pour notre modèle d'optimisation.

L'activation des nœuds

Cet ensemble des contraintes concerne principalement l'établissement de la topologie réseau entre les différentes couches de l'architecture présentée par la section 4.3. Pour faciliter la modélisation, on suppose que le sens des liens entre les différents ensembles de nœuds est ascendant et que chaque nœud ne peut être lié qu'avec un seul noeud appartenant à la couche qui suit.

La contrainte 4.2 impose que pour chaque terminal t, un seul lien x_{nt} avec un EN n peut être établi.

$$\sum_{n \in \Omega_E} x_{nt} = 1, \forall t \in S \tag{4.2}$$

Par la suite, les contraintes 4.3 et 4.4 considèrent que la variable de décision binaire E_n du déploiement d'un EN n est déterminée par les liens avec les terminaux. La possibilité d'activer un EN n dépend du nombre de liaisons avec l'ensemble des terminaux t (il suffit que le nombre de liens soit ≥ 1) :

$$E_n \leq \sum_{t \in S} x_{nt}, \forall n \in \Omega_E \tag{4.3}$$

$$x_{nt} \leq E_n, \forall n \in \Omega_E, \forall t \in S \tag{4.4}$$

La même démarche s'applique pour les autres liens : ENs-FNs et FNs-passerelles. La contrainte (4.5) impose qu'il doit y avoir un seul FN m lié à un EN n.

$$\sum_{m \in \Omega_F} x_{mn} = 1, \forall n \in \Omega_E \tag{4.5}$$

De même, les contraintes (4.6) et (4.7) déterminent l'activité de la variable de décision binaire F_m de déployer un FN m en fonction de ses liens avec les noeuds ENs ;

$$F_m \leq \sum_{k \in \Omega_E} x_{mn}, \forall m \in \Omega_F \tag{4.6}$$

$$x_{mn} \leq F_m, \forall m \in \Omega_F, \forall n \in \Omega_E \tag{4.7}$$

La contrainte (4.8) impose qu'il doit y avoir un seul FN m lié à un EN n.

$$\sum_{c \in \Omega_G} x_{cm} = 1, \forall m \in \Omega_F \tag{4.8}$$

Passons maintenant aux liens des passerelles. Elles seront liées aux FNs et au serveur Cloud. Le même principe va s'appliquer en reliant chaque FN à une seule passerelle. Les contraintes (4.9) et (4.10) déterminent la variable de décision binaire G_c modélisant l'opportunité de déployer une passerelle c en fonction des liens établis avec les FNs.

$$G_c \leq \sum_{m \in \Omega_F} x_{cm}, \forall c \in \Omega_G \tag{4.9}$$

$$x_{cm} \leq G_c, \forall c \in \Omega_G, \forall m \in \Omega_F \tag{4.10}$$

Finalement, il faut activer le lien entre les passerelles c déployées et le serveur du Cloud référé par s. Chaque passerelle c déployée doit être liée avec le Cloud comme le montre l'équation (4.11). La position du serveur Cloud est fixée et occupe un seul emplacement dans l'atelier de production.

$$x_{sc} = G_c, \forall c \in \Omega_G \tag{4.11}$$

La capacité de demande des nœuds

Le deuxième groupe des contraintes concerne la capacité de demande. Les terminaux, ENs et Fns transmettent d'une façon continue une quantité de données

aux noeuds des niveaux supérieurs, ENs, FNs et passerelles, pour les traiter. Par exemple, les données envoyées par tous les terminaux liés à un EN t ne doivent pas dépasser sa capacité maximale pour garantir une haute performance de délivrance des décisions en temps réel.

Cette contrainte est non linéaire comme elle intègre dans son calcul une multiplication de deux vecteurs de variable de décision : la variable binaire x_{ij}, décide si y aura un lien fibre entre un nœud i et un autre nœud j et la demande de capacité de calcul pour un site X_i : $\omega(X_i)$. Dans ce qui suit, une de nos contributions sera la linéarisation de ces équations afin de résoudre ce problème à l'aide d'un solver utilisant la méthode linéaire exacte.

Les contraintes relatives à la capacité de demande de tous les terminaux sont expliquées comme suit. La contrainte (4.12) impose que la quantité de la demande $\omega(E_n)$ desservie par le nœud EN n soit la somme des demandes de tous les terminaux déterminée par tous les liens entre le nœud EN n et chaque terminal t.

$$\omega(E_n) = \sum_{t \epsilon S} r.x_{nt}, \forall n \in \Omega_E \tag{4.12}$$

La contrainte (4.13) impose une limite pour que la somme de la demande provenant des terminaux ne soit pas supérieure à la capacité maximale H_n^E pour un EN n.

$$\omega(E_n) \leq E_n.H_n^E, \forall n \in \Omega_E \tag{4.13}$$

La même condition s'applique pour les données transmises des ENs aux FNs. Les contraintes (4.14) et (4.15) considèrent la capacité maximale H_m^F pour un FN m comme étant une limite de capacité pour la demande provenant des ENs n.

$$\omega(F_m) = \sum_{n \in \Omega E} \omega(E_n).x_{mn}, \forall m \in \Omega_F \tag{4.14}$$

$$\omega(F_m) \leq F_m.H_m^F, \forall m \in \Omega_F \tag{4.15}$$

Pour la capacité de demande entre les FNs et les passerelles, les contraintes (4.16) et (4.17) considèrent la capacité maximale H_c^G pour une passerelle m. La capacité de demande est la somme des demandes des FNs m liées à une passerelle c.

$$\omega(G_c) = \sum_{m \in \Omega_F} \omega(F_m).x_{cm}, \forall c \in \Omega_G \tag{4.16}$$

$$\omega(G_c) \leq G_c.H_c^G, \forall c \in \Omega_G \tag{4.17}$$

La latence des transmissions

La latence pour une application basée sur le Fog Manufacturing est un critère notable afin de juger la performance de cette solution. La latence influence le problème d'optimisation du coût. Pour chaque lien, il ne faut pas dépasser un seuil de latence de transmission.

Les contraintes (4.18), (4.19) et (4.20) calculent les temps de latence entre un EN n et un terminal t, entre un FN m et un EN n, et entre la passerelle c et un FN m. Ces délais ne doivent pas dépasser des latences maximales γ_t^n , γ_n^m et γ_m^c, respectivement.

$$L/(\sum_{t\equiv\in T} x_{nt}.\mu_{tn}).E_n \leq \gamma_t^n \tag{4.18}$$

$$L/(\sum_{n\in\Omega_E} x_{mn}.\mu_{nm}).F_m \leq \gamma_n^m \tag{4.19}$$

$$L/(\sum_{m\epsilon\Omega_F} x_{mc}.\mu_{mc}).G_c \leq \gamma_m^c \tag{4.20}$$

La couverture des AGVs par les nœuds Edge

Parmi les terminaux t appartenant à S, il existe des CPS mobiles qui assurent les tâches de transport dans les différents départements d'un atelier de production. Les AGVs ou les AGVs équipés de bras mobiles sont connectés sans fil aux ENs. Lors de leur déplacement, ils risquent de sortir de la zone de couverture d'un EN 1 par exemple, donc il est nécessaire de garantir le relais de couverture instantané par un autre EN t afin de ne pas interrompre leur connectivité.

La contrainte (4.21) traduit cette condition et considère que la distance entre les ENs et les AGVs t ne doit pas être supérieure à la portée de couverture R_n d'un EN n.

$$x_{nt}.d_{nt} \leq R_n/2, \forall n \in \Omega_E, \forall t \in \Omega_{AGV} \tag{4.21}$$

Le nombre maximal des liens supporté par un nœud

Il est logique d'attribuer un max de dispositifs avec chaque noeud de niveau supérieur mais cette stratégie risque de saturer les noeuds et centraliser le traitement au niveau de quelques nœuds. Par contre le principe du Fog Manufacturing est de distribuer le calcul sur l'ensemble de l'atelier de production. La limitation des liens pour chaque nœud est nécessaire. Le nombre de liens pour chaque EN, FN et passerelle a une capacité maximale N_E, N_F, N_G, respectivement dans les contraintes (4.22)-(4.23) et (4.24).

$$\sum_{t\in T} x_{nt} \leq N_E, \forall n \in \Omega_E \tag{4.22}$$

$$\sum_{n\in\Omega_E} x_{mn} \leq N_F, \forall m \in \Omega_F \tag{4.23}$$

$$\sum_{m\epsilon\Omega_F} x_{cm} \leq N_G, \forall c\epsilon\Omega_G \tag{4.24}$$

4.4.4 Nos contributions

Le modèle adopté comme base dans notre travail (voir. LIN et YANG, 2018) était souvent optimisé à l'aide des méthodes métaheuristiques (les algorithmes génétiques). Ce choix a été adopté par les auteurs parce que le modèle possède des non-linéarité ce qui rend l'optimisation à l'aide de méthodes exactes compliquée lente ou impossible. La deuxième raison est la nature de ce problème qui est considéré comme un problème NP-complet qui consomme beaucoup de temps avant de converger. Malgré ces justifications de l'utilisation des méta-heuristiques, aucune comparaison avec des solutions optimales n'a été mentionnée dans la littérature, ce qui rend difficile d'apprécier les solutions proposées par les méthodes méta-heuristiques. Dans notre travail, nous avons choisi de travailler avec la méthode

exacte pour fournir un seuil de comparaison aux autres méthodes. Nous avons rencontré les même problèmes de non-linéarité et de complexité du problème qui ont empêché les autres équipes de procéder par méthode exacte. Dans ce qui suit, nous allons commencer par traiter le problème de non-linéarité. Cette contribution nous aidera à enrichir le modèle d'optimisation du coût de deux nouvelles contraintes concernant le type de lien et la considération de dispositifs mobiles dans le modèle tels que les AGVs ou les interfaces de commande mobiles. Ces deux contraintes seront expliquées et modélisées dans ce qui suit.

La linéarisation

Le modèle brut utilisé par les autres travaux était habituellement optimisé à l'aide des algorithmes génétiques vu sa grande taille et sa non-linéarisation au niveau des contraintes de la capacité de la demande 4.4.3, qui regroupe à la fois deux variables de décision que sont le lien et la capacité de demande d'un nœud.

La linéarisation des contraintes est utilisée pour respecter les normes de conception et les exigences de la résolution. Néanmoins, il est nécessaire d'écrire des équations non-linéaires afin de définir les conditions à poser au modèle.

Nous avons résolu ce problème en divisant la variable de décision en plusieurs afin de limiter le nombre de variables de décision. Également, nous avons utilisé des variables intermédiaires qui contiennent les variables de décisions des sous ensembles.

La probabilité de la capacité de réseau

Dans un atelier de production, les trajectoires de déplacement des robots mobiles sont généralement pré-définies afin de faciliter le déplacement et empêcher les accidents de travail. Cette dernière contrainte concerne principalement le AGVs qui se déplacent pour accomplir des tâches. Elle impose une condition sur chaque zone de présence d'AGV de telles sorte à ce qu'il existe une probabilité de présence des AGVs. Cette probabilité est influencée par deux paramètres :

— L'impossibilité de présence d'AGV dans deux endroits en même temps, comme le montre la figure 4.8 ,

— La capacité d'une zone à contenir un nombre limité d'AGV. Par exemple un couloir ne peux contenir plus que deux ou trois AGVs selon la largeur du couloir et les distances de sécurité minimales entre les robots.

Cette contrainte sert à répartir la charge sur toute la zone et à éviter la surcharge du réseau par les requêtes dans une petite zone gérée par un faible nombre d'ENs. En conséquence, cette contrainte limite la capacité de réseau à un maximum H_p^E. Pour faciliter la modélisation, on considère le plan, présenté par la figure 4.9, de notre laboratoire sur lequel nous allons appliquer le modèle pour la suite de ce travail. Par exemple, pn prend un plan de surface 30mx20m, dont chaque cellule est un carré de $1m^2$. Chaque cellule ne peut supporter plus qu'un terminal (AGV dans ce cas). Les trajectoires sont marquées par les flèches bleues sur la figure 4.9. Nous avons traduit ce plan en une matrice de probabilités de présence d'un terminal pour chaque cellule. Nous avons également défini un vecteur de variables, nommé ω_P. Ce vecteur est la probabilité de présence des terminaux sur chaque cellule ou ensemble de cellules (zones) multiplié par la demande d'un terminal. L'équation (4.25) donne la probabilité $P_z(X_i)$ de chaque cellule X_i appartenant à une zone z de l'environnement de l'atelier de production comme suit :

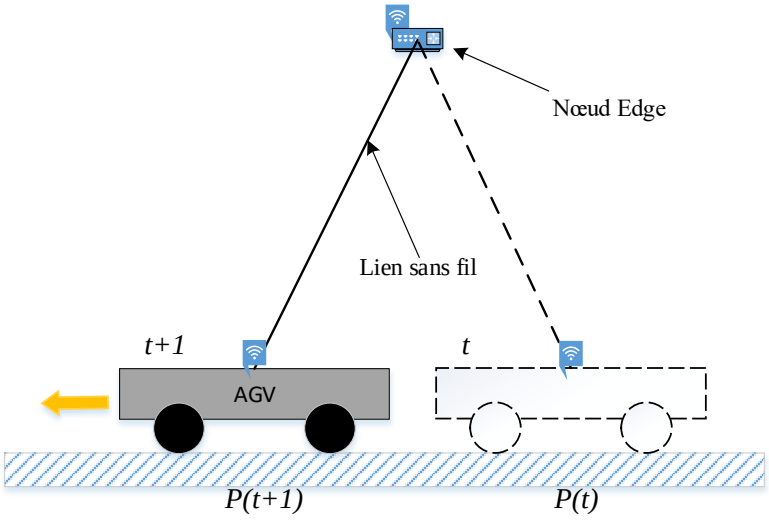


FIGURE 4.8 – Transmission d'un dispositif mobile (AGV).

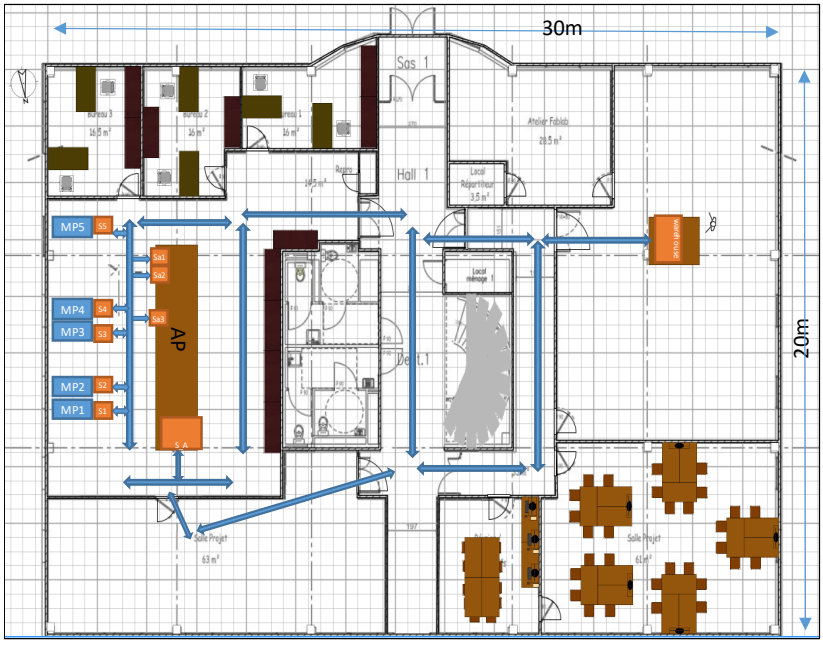


FIGURE 4.9 – Le plan du laboratoire LINEACT CESI.

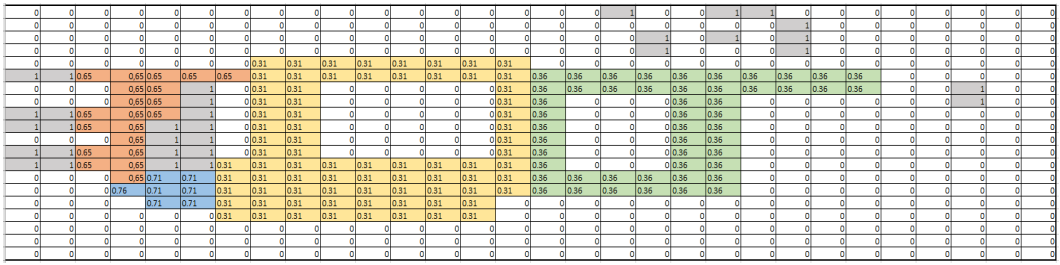


FIGURE 4.10 – Les probabilités de chaque cellule en fonction de la mobilité des AGVs.

$$P_z(X_i) = \frac{NbrTMax_z}{NbrCell_z}, \forall i \in M, \forall z \in Z \tag{4.25}$$

Avec $NbrTMax_z$ le nombre maximal possible des terminaux présents dans une zone z, et $NbrCell_z$ le nombre de cellules constituant la zone z, M est l'ensemble des cellules constituant l'atelier de production, et Z est l'ensemble des zones de l'atelier de production.

La calcul de cette probabilité dans la zone de production de l'atelier du laboratoire CESI est donnée par l'équation (4.26)

$$P_z(X_i) = \frac{NbrTMax_z}{NbrCell_z}, \forall i \in \{1, \ldots, 600\}, \forall z \in \{1, \ldots, 6\} \tag{4.26}$$

Comme le montre la matrice présentée sur la figure 4.10, la probabilité des cellules où sont installés les terminaux fixes est égale à 1. Pour les terminaux mobiles, qui sont les AGVs, leur présence est définie par les trajectoires colorées (Orange, Bleue, Jaune et verte) sur la figure 4.10. Par contre cette présence étant dynamique, nous avons pensé à la rendre statique en introduisant un AGV sur chaque cellule des trajectoires et en gardant une distance de sécurité de 1 mètre. La probabilité varie d'une zone à une autre selon les trajectoires et la présence maximale des AGVs. Nous pouvons distinguer les zones suivantes :

- $Z_1 : P_{z_1}(X_i)$=1, $\forall X_i \in Z_1$ et X_i représenté en gris
- $Z_2 : P_{z_2}(X_i) = 0,65$, $\forall X_i \in Z_2$, Avec $NbrTMax_{z_2} = 7$ et $NbrCell_{z_2} = 20$ et X_i représenté en orange,
- $Z_3 : P_{z_3}(X_i) = 0,71$, $\forall X_i \in Z_3$, $NbrTMax_{z_3} = 5$ et $NbrCell_{z_3} = 7$, et X_i représenté en bleu,
- $Z_4 : P_{z_4}(X_i) = 0,31$ $\forall i \in Z_4$, Avec $NbrTMax_{z_4} = 21$ et $NbrCell_{z_4} = 67$, et X_i représenté en jaune,
- $Z_5 : P_{z_5}(X_i)$= 0,36 $\forall X_i \in Z_5$, Avec $NbrTMax_{z_5} = 18$ et $NbrCell_{z_5} = 50$, et X_i représenté en vert,
- $Z_6 : P_{z_6}(X_i)$= 0 $\forall X_i \in Z_6$, Avec $NbrTMax_{z_6} = 0$ et $NbrCell_{z_6} = 456$, et X_i représenté en blanc,

Finalement la contrainte est donnée par les équations (4.27) & (4.28) :

$$\omega_P(E_n) = \sum_{t \in T} P(t).r.x_{nt}, \forall n \in \Omega_E \tag{4.27}$$

$$\omega_P(E_n) \leq E_n.H_P^E, \forall n \in \Omega_E \tag{4.28}$$

Le type de lien

Nous avons proposé l'optimisation des types de liens filaires entre les ENs-FNs, FNs-Passerelles et Passerelles-Cloud pour minimiser le coût de l'installation de la solution Fog et augmenté la qualité de service. Nous avons créé une contrainte qui détermine le type du lien à choisir entre la liaison avec fibre optique ou avec le câble RJ45 selon la demande en données. Cette contrainte dépend de la contrainte (4.13) de la capacité de la demande expliqué dans la section 4.4.3. Si la demande provenant d'un nœud à un autre ne dépasse pas un seuil on utilise un lien RJ45 sinon on opte pour la fibre optique.

Pour faciliter la modélisation du problème nous avons supposé que le lien entre les noeuds de traitement de données peut être un lien fibre ou un lien câble RJ45. Donc le lien sera activé selon le besoin en terme de la capacité de demande. La contrainte (4.29) traduit cette condition et la contrainte (4.29) force que le lien sera binaire (0 ou 1) :

$$x_{ij} = xF_{ij} + xR_{ij}, \forall i \in \Omega_E \cup \Omega_F \cup \Omega_G, j \in \Omega_F \cup \Omega_G \cup S \tag{4.29}$$

$$xF_{ij} + xR_{ij} \leq 1, \forall i \in \Omega_E \cup \Omega_F \cup \Omega_G, j \in \Omega_F \cup \Omega_G \cup S \tag{4.30}$$

Par la suite nous passons à la définition de la condition d'un tel choix. La première contrainte (4.31) concerne les liens entre les ENs n et les FNs m.

$$x_{mn} = \begin{cases} xR_{mn} & \text{si } \sum_{t \in T} r.x_{nt} \leq MaxCapR \\ xF_{mn} & \text{sinon.} \end{cases} \quad \forall n \in \Omega_E, \ \forall m \in \Omega_F \tag{4.31}$$

La contrainte (4.32) concerne les liens entre les FNs m et les passerelles c.

$$x_{cm} = \begin{cases} xR_{cm} & \text{si } \sum_{n \in \Omega_E} r.x_{mn} \leq MaxCapR \\ xF_{cm} & \text{sinon.} \end{cases} \quad \forall m \in \Omega_F, \ \forall c \in \Omega_G \tag{4.32}$$

Les derniers liens sont ceux entre les passerelles c et le centre Cloud s. Le choix de ces liens dépend principalement de la capacité de demande provenant des FNs m liés aux passerelles c. La contrainte (4.33) traduit cette condition.

$$x_{sc} = \begin{cases} xR_{sc} & \text{si } \sum_{m \in \Omega_F} r.x_{cm} \leq MaxCapR \\ xF_{sc} & \text{sinon.} \end{cases} \quad \forall c \in \Omega_G, \ \forall s \in S \tag{4.33}$$

4.5 L'optimisation du modèle

Les méthodes d'optimisation sont le fruit du croisement de mathématiques appliquées et de l'informatique. Elles représentent une grande partie de la recherche opérationnelle. Elles sont généralement adoptées pour surmonter la grande difficulté des problématiques de résolution et pour faciliter l'optimisation des critères de nombreuses applications. Les nombreux travaux parcourus dans la section 4.2.2 ont utilisé les approches Méta-heuristiques (les algorithmes génétiques) pour obtenir des solutions les plus proches de l'optimal. Dans notre travail nous avons choisi de résoudre le modèle de minimisation du coût à l'aide de la méthode exacte. Le nombre d'instances du modèle étant important, nous avons choisi d'opter pour une démarche en deux temps en résolvant d'abord le modèle pour une petite instance

et dans une seconde étape en nous intéressant à l'instance complète. Ainsi, nous allons présenter les résultats de la résolution avec les cinq premières contraintes (l'activation des liens, la capacité de demande, la latence, la couverture et le nombre maximale, des liens) et par la suite nous allons intégrer le type de lien et ensuite la capacité de réseau.

4.5.1 La méthode exacte

Les méthodes exactes telles que la programmation linéaire, sont des méthodes heuristiques qui garantissent de trouver des solutions optimales exactes et prouvent l'optimalité pour toutes les instances d'un problème d'optimisation combinatoire (POC) de petite et de moyenne taille. En revanche, le temps de résolution du problème augmente selon la taille du problème à optimiser.

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent d'une manière implicite, l'ensemble des solutions de l'espace de recherche. Pour améliorer l'énumération des solutions, une telle méthode dispose de techniques pour détecter le plus tôt possible la solution exacte. Parmi les méthodes exactes, on trouve la plupart des méthodes traditionnelles telles que les techniques de séparation et évaluation progressive (SEP) ou les algorithmes avec retour arrière.

De plus, pour certains problèmes, la consommation de mémoire de ces algorithmes peut être très grande et peut parfois entraîner l'arrêt prématuré de l'application. La programmation linéaire peut être utilisée pour résoudre de façon exacte les POC. En effet, lorsqu'un POC peut être modélisé en programme linéaire, il peut alors être résolu, par exemple à l'aide de l'algorithme du simplexe. Lorsqu'on a la contrainte supplémentaire voulant que les valeurs des variables de décision soient entières, on a un programme linéaire en nombre entier (PLNE).

L'algorithme B&B est utilisé par les méthodes exactes pour résoudre d'une façon exacte des problèmes en nombre entier en utilisant le branchement pour séparer l'ensemble des solutions en sous ensembles et l'évaluation pour mettre des bornes supérieures ou inférieures sur les solutions (KHERBOUCHE et OUBAHRI, 2017).

Pour l'optimisation des différents cas d'étude et exemples nous avons utilisé le solver FICO Xpress IVE. Ce solver utilise comme langage le Mosel, qui est un environnement dédié à la modélisation et à la résolution des problèmes. Il fournit un langage, à la fois de modélisation et de programmation. L'originalité du langage Mosel est qu'il n'y a pas de séparation entre une déclaration de modélisation (par exemple, déclarer une variable de décision ou exprimer une contrainte) et une procédure qui résout réellement le problème (par exemple, appel à une commande d'optimisation). Grâce à cette association, on peut programmer un algorithme de solution complexe en combinant la modélisation et la résolution du problème.

4.5.2 Description des cas d'étude

Cette section met en œuvre le modèle d'optimisation proposé, et effectue des simulations sur des problèmes de petite taille et de taille moyenne. Nous avons optimisé le modèle avec et sans les nouvelles contraintes et nous avons comparé les différentes approches afin de démontrer l'apport de nos contributions en terme du coût. Nous avons considéré le plan du laboratoire LINEACT CESI qui était présenté précédemment par la figure 4.9.

La première étape est l'attribution des nœuds de l'architecture basée sur le Fog Manufacturing qui sont les ENs, FNs et les passerelles. Nous avons choisi de mettre un maximum de noeuds à côté de toutes les prises d'alimentation électriques du



FIGURE 4.11 – Le plan du laboratoire LINEACT CESI avec l'emplacement potentiel des différents nœuds.

laboratoire. La figure 4.11 présente le plan de l'usine (i.e. laboratoire), l'emplacement fixe du centre Cloud, les trajectoires des AGVs, les emplacements des systèmes fixes et tous les emplacements potentiels possibles des ENs, FNs et passerelles. Les deux instances de modèles sont de petite taille et de taille moyenne. Le premier sera une simulation sur un petit plan de$Longueur \times Largeur = 6m \times 6m$ (une surface de 36 m^2) et le deuxième correspond au plan réel du laboratoire $Longueur \times Largeur = 30m \times 20m$ (soit une surface de 600 m^2). Nous avons divisé l'environnement en cellules de 1 m^2. Chaque cellule ne peut contenir qu'un noeud. Afin de pouvoir appliquer le modèle déjà présenté nous allons numéroter les cellules C_i de $Longueur \times Largeur$. chaque cellule C_i a des coordonnées (C_i^x, C_i^y). Les distances utilisées lors de différentes résolutions sont calculées en utilisant Matlab. La première matrice de distance D composée d'éléments d_{ij} est obtenu par les distances de Manhattan représentant les distances en escalier entre les ENs-FNs, FNs-passerelles et Passerelles-Cloud en utilisant la formule de l'équation (4.34) :

$$d_{ij} = \left|C_j^x - C_i^x\right| + \left|C_j^y - C_i^y\right|, \forall i, j \in \{1, \ldots, Longueur \times Largeur\} \tag{4.34}$$

Les distances de lien WIFI peuvent être calculées par la distance euclidienne comme le montre l'équation (4.35) :

$$d_{ij} = \sqrt{(C_j^x - C_i^x)^2 + (C_j^y - C_i^y)^2}, \forall i, j \in \{1, \ldots, Longueur \times Largeur\} \tag{4.35}$$

Ces fichiers de distance sont utilisés comme étant des données d'entrée pour la résolution du modèle. Le paramétrage utilisé dans les simulations sera présenté avant chaque résolution.

4.5.3 Résolution et résultats

Tout au long de cette section les paramètres utilisés pour la résolution du modèle sont fixés à :

- Les coûts cf = 5, cG = 30, cF = 25, et cF = 10, cRJ45=3, MaxCapaR=50 ;
- Les débits de données μ_{tn} = 1000, μ_{mn} = 4000, et μ_{mc} = 4000 ;
- Le diamètre de couverture d'un dispositif de bord (edge) R_n = 15 ;
- Les capacités maximales N_E = 15, N_F = 10, et N_G = 7 ;
- Les longueurs de données L = 5000, L_E = 10000, et L_F = 15000 ;
- Les capacités de demande maximales, c'est-à-dire H_n^E= 150, H_m^F = 1500, et H_c^G= 7000.
- Les temps de latence maximaux ont les valeurs suivantes : γ_t^n = 80, γ_n^m = 20, et γ_c^m = 15 ;
- La demande des terminaux r = 50.

Le modèle sans linéarisation

Nous avons commencé par la résolution du modèle sans la linéarisation et sans les nouvelles contraintes (La probabilité de demande du réseau et le type de lien). Nous avons démarré par l'optimisation d'une petite instance qui a convergé. Par contre cette résolution n'a pas convergé même au bout de nombreuses heures de recherche de solution optimale exacte (Out of Memory) pour l'instance complète. L'instance a généré un énorme nombre de variables qui dépasse largement les capacités du solver malgré l'utilisation des fonctionnalités de résolution non-linéaire. Nous avons donc conclu que la linéarisation était une étape essentielle afin de pouvoir obtenir une solution exacte.

Le modèle avec linéarisation & sans nouvelles contraintes

Dans un premier temps, nous avons résolu l'exemple simplifié afin de visualiser des résultats plus simples à analyser et de tester les contraintes avant de résoudre la grande instance. Le plan et les emplacements potentiels des nœuds de l'architecture sont illustrés par la figure 4.12.

Nous avons supposé que tous les nœuds terminaux sont connectés sans fil aux ENs. Les AGVs assurent la logistique interne dans un atelier de l'Industrie 4.0. Le reste des liaisons entre les ENs, FNs, Passerelles et Cloud est filaire (Liens de type fibre optique). On considère 4 AGVs, 2 machines, 1 imprimante 3D, 12 ENs, 7 FNs, 2 passerelles et 1 serveur Cloud. Il est nécessaire de pré-définir les ensembles de chaque catégorie de nœuds, constituant l'architecture, pour créer les liens et activer les noeuds par la suite. Cette définition est présentée ci-dessous :

- S={21}
- Ω_G={18, 19}
- Ω_F ={8, 11, 13, 23, 24, 25, 32}
- Ω_E ={1, 3, 5, 6, 7, 12, 30, 31, 34, 35, 36}
- T = {9, 10, 15, 16, 26, 27, 28}

La position de chaque nœud est fixe et la distance entre chaque nœud est calculée en fonction du type de liaison (sans fil/fibre). Le modèle est résolu en utilisant l'approche de la méthode exacte et la Programmation Linéaire. Le modèle est optimisé à l'aide du solveur de Fico "Xpress IVE" afin de minimiser le coût de déploiement du Fog Manufacturing avec une architecture de 5 couches.

La figure 4.19 illustre le résultat de l'optimisation du modèle simplifié. Les noeuds activés et les liens établis sont présentés : ENs (7,12,30), FNs (13,23) et la

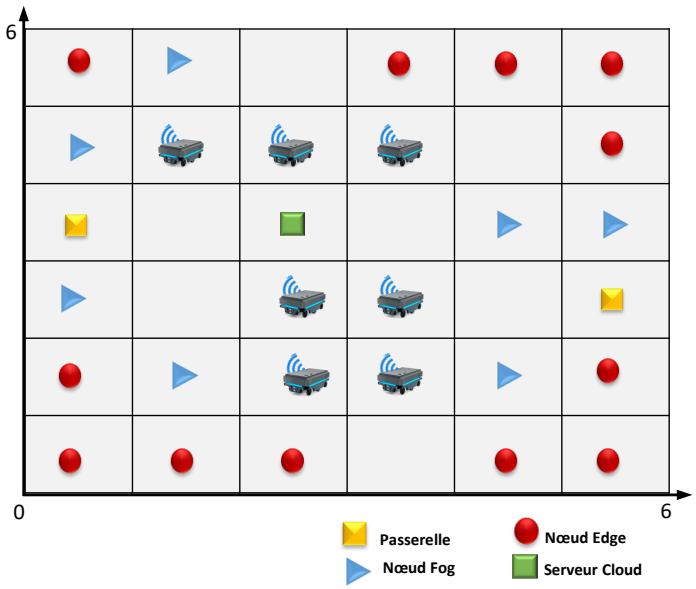


FIGURE 4.12 – Le plan et l'emplacement potentiel de différents noeuds de la petite instance du modèle.

passerelle 19 reliant les FNs au Cloud. Le coût *Cost* du déploiement du Fog Manufacturing dans ce cas est égal à 765 (unit de coût), affiché sur la figure 4.14.

Par la suite nous avons résolu le modèle complet qui modélise le plan de 600 m^2. Nous avons gardé le même paramétrage mais avec de nouveaux ensembles de noeuds adéquats au nouvel intervalle. Ces différents ensembles sont indiqués ci-dessous :

— $S = \{413\}$
— T=$\{18, 22, 51, 53, 79, 81, 83, 109, 103, 128, 129, 130, 131, 132, 133, 134, 135, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 184, 185, 186, 187, 188, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 208, 214, 215, 216, 218, 219, 225, 226, 230, 231, 283, 241, 242, 243, 244, 245, 246, 248, 249, 255, 256, 260, 261, 271, 272, 273, 274, 275, 276, 278, 279, 285, 286, 290, 291, 304, 305, 306, 308, 309, 315, 316, 320, 321, 331, 332, 333, 334, 335, 336, 338, 339, 345, 346, 350, 351, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 380, 381, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 405, 406, 407, 408, 409, 410, 411, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 517, 518, 519, 520, 521, 522, 523, 524\}$
— Ω_G=$\{77, 388, 423\}$
— Ω_F=$\{1, 7, 11, 20, 29, 47, 73, 99, 211, 232, 337, 391, 414, 452, 582, 598\}$
— $Omega_E$= $\{2, 3, 8, 10, 12, 13, 17, 19, 23, 24, 25, 27, 28, 31, 60, 61, 70, 72, 92, 101, 102, 120, 122, 177, 180, 181, 183, 207, 262, 301, 322, 330, 360, 377, 379, 387, 393, 412, 415, 418, 419, 421, 450, 451, 453, 467, 468, 470, 481, 500, 530, 540, 564, 571, 573, 574, 583\}$

Partant de la même démarche faite avec la petite instance du modèle, nous avons réparti des ENs, FNs et passerelles dans toutes les cellules possédant une alimentation électrique avec une distribution maximale sur tout l'atelier. Cette répartition est illustrée par la figure 4.15.

Le modèle est optimisé suite à la linéarisation qui nous a permis d'obtenir une convergence en moins du temps. Le résultat de cette optimisation du modèle est présenté sur la figure 4.16. Le coût est optimisé à Cost=6013 présenté par la figure 4.17.

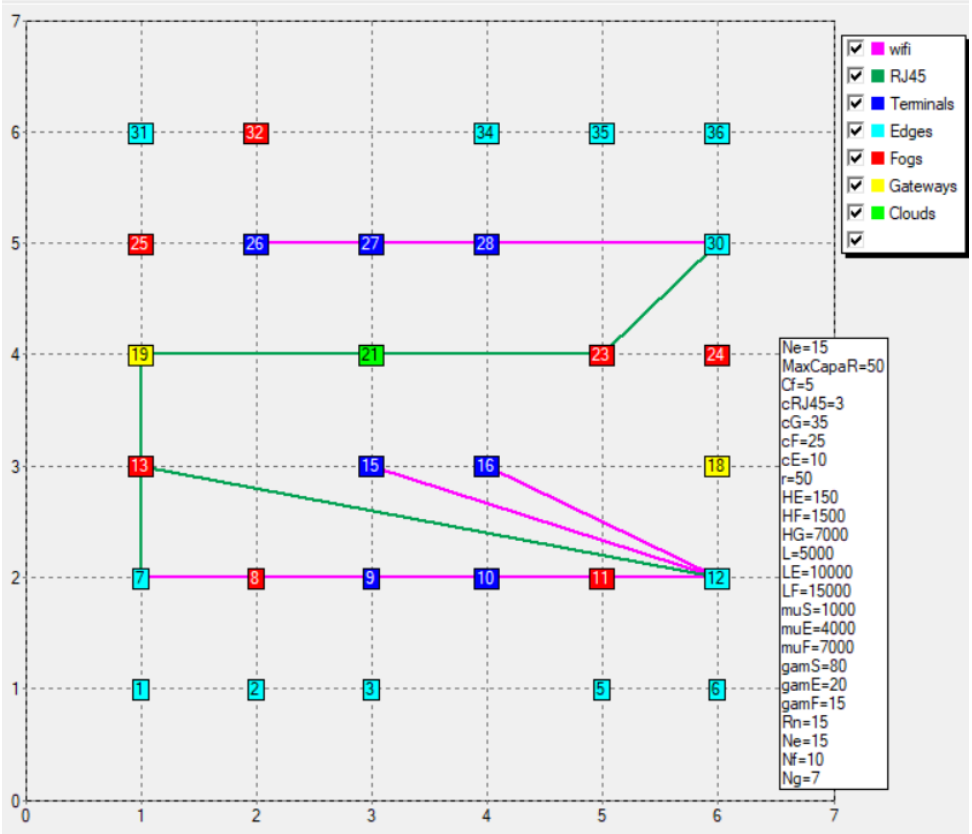


FIGURE 4.13 – Les noeuds activés pour la petite instance du modèle avec la linéarisation.

```
Performing root presolve...

Reduced problem has:     176 rows      71 columns       590 elements
Presolve dropped   :     309 rows     107 columns       477 elements
Presolve tightened :        75 elements
Symmetric problem: generators: 3, support set: 5
 Number of orbits: 2, largest orbit: 3
 Row orbits: 2, row support: 5
Will try to keep branch and bound tree memory usage below 9.2GB

Starting root cutting & heuristics

 Its Type    BestSoln    BestBound   Sols    Add    Del     Gap     GInf   Time
 *** Search completed ***     Time:     0 Nodes:          1
Number of integer feasible solutions found is 4
Best integer solution found is   765.000000
Best bound is   764.999369
Uncrunching matrix
Solution:765
TotalCost:0      AOQ=0
TotalCost:765
Solution:
        t9      -------  e12    -------  f13    -------  g19    -------  c21
        t10     -------  e7     -------  f13    -------  g19    -------  c21
        t15     -------  e12    -------  f13    -------  g19    -------  c21
        t16     -------  e12    -------  f13    -------  g19    -------  c21
        t26     -------  e30    -------  f23    -------  g19    -------  c21
        t27     -------  e30    -------  f23    -------  g19    -------  c21
        t28     -------  e30    -------  f23    -------  g19    -------  c21
```

FIGURE 4.14 – Le coût et le temps de résolution pour la petite instance du modèle avec la linéarisation.

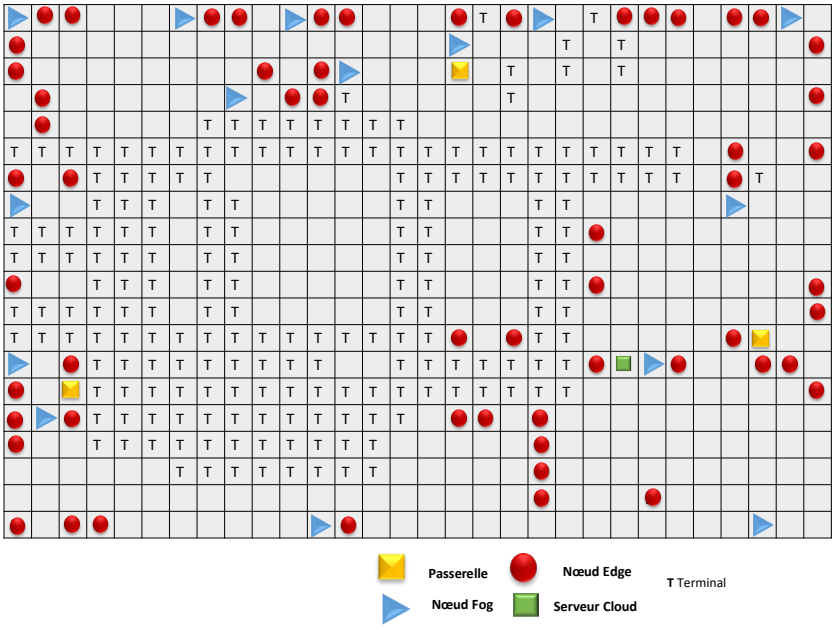


FIGURE 4.15 – Le plan et l'emplacement potentiel de différents noeuds pour la grande instance du modèle.

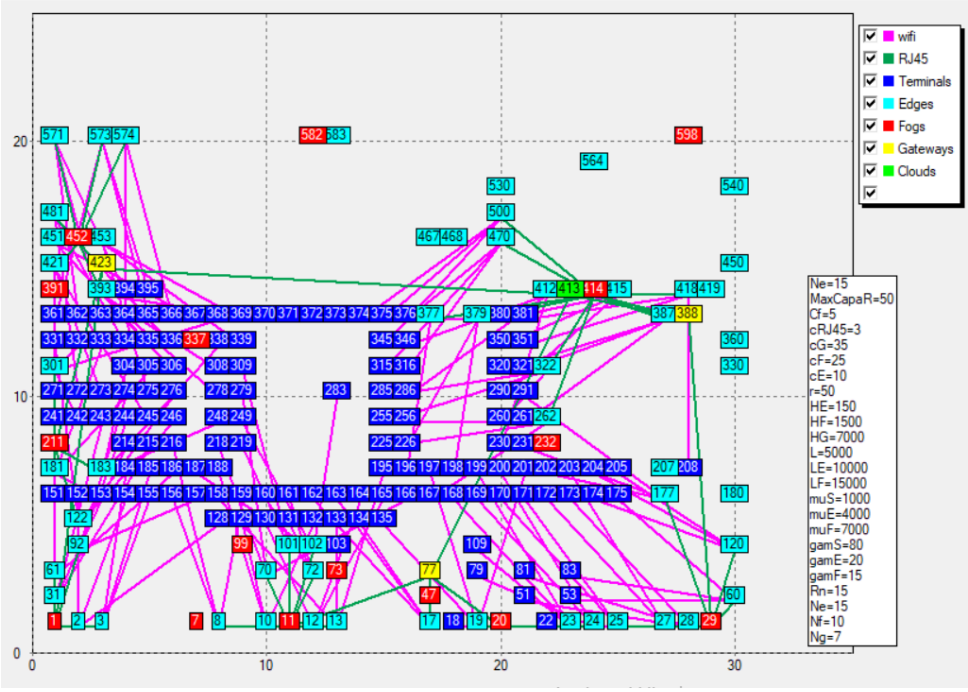
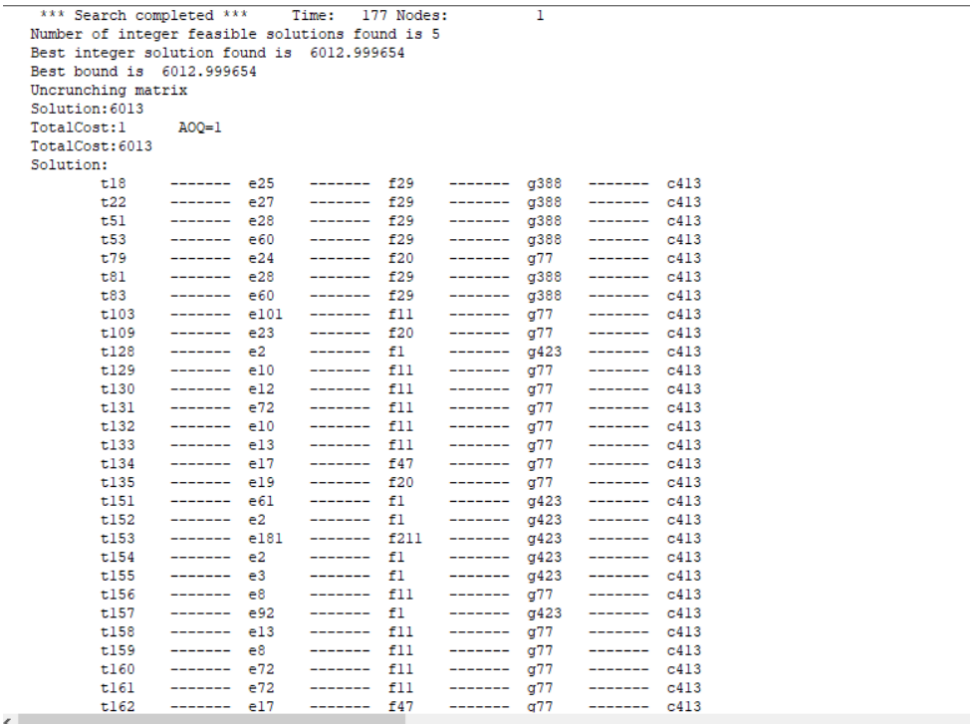


FIGURE 4.16 – Les nœuds activés pour la grande instance du modèle avec la linéarisation.



```
 *** Search completed ***     Time:   177 Nodes:          1
Number of integer feasible solutions found is 5
Best integer solution found is  6012.999654
Best bound is  6012.999654
Uncrunching matrix
Solution:6013
TotalCost:1      AOQ=1
TotalCost:6013
Solution:
        t18     -------  e25    -------  f29    -------  g388   -------  c413
        t22     -------  e27    -------  f29    -------  g388   -------  c413
        t51     -------  e28    -------  f29    -------  g388   -------  c413
        t53     -------  e60    -------  f29    -------  g388   -------  c413
        t79     -------  e24    -------  f20    -------  g77    -------  c413
        t81     -------  e28    -------  f29    -------  g388   -------  c413
        t83     -------  e60    -------  f29    -------  g388   -------  c413
        t103    -------  e101   -------  f11    -------  g77    -------  c413
        t109    -------  e23    -------  f20    -------  g77    -------  c413
        t128    -------  e2     -------  f1     -------  g423   -------  c413
        t129    -------  e10    -------  f11    -------  g77    -------  c413
        t130    -------  e12    -------  f11    -------  g77    -------  c413
        t131    -------  e72    -------  f11    -------  g77    -------  c413
        t132    -------  e10    -------  f11    -------  g77    -------  c413
        t133    -------  e13    -------  f11    -------  g77    -------  c413
        t134    -------  e17    -------  f47    -------  g77    -------  c413
        t135    -------  e19    -------  f20    -------  g77    -------  c413
        t151    -------  e61    -------  f1     -------  g423   -------  c413
        t152    -------  e2     -------  f1     -------  g423   -------  c413
        t153    -------  e181   -------  f211   -------  g423   -------  c413
        t154    -------  e2     -------  f1     -------  g423   -------  c413
        t155    -------  e3     -------  f1     -------  g423   -------  c413
        t156    -------  e8     -------  f11    -------  g77    -------  c413
        t157    -------  e92    -------  f1     -------  g423   -------  c413
        t158    -------  e13    -------  f11    -------  g77    -------  c413
        t159    -------  e8     -------  f11    -------  g77    -------  c413
        t160    -------  e72    -------  f11    -------  g77    -------  c413
        t161    -------  e72    -------  f11    -------  g77    -------  c413
        t162    -------  e17    -------  f47    -------  g77    -------  c413
```

FIGURE 4.17 – Le coût et le temps de résolution pour la grande instance du modèle avec la linéarisation.

Les configurations initiales avant l'optimisation présentées sur les figures 4.12 et 4.15 seront utilisées comme configurations initiales pour la suite des résolutions.

Étude de la sensibilité

L'implémentation du Fog Manufacturing est un investissement pour les industries qui cherchent à diminuer les coûts de la production et du déploiement des nouveaux outils pouvant gérer. Notre modèle est optimisé selon la méthode exacte, donc tous les résultats avec les différents paramétrages sont optimaux. Par contre certains d'entre eux possèdent la meilleure configuration permettant un déploiement rentable (coût et activation des noeuds) non seulement à court terme mais aussi à long terme. Ainsi fournir un conseil pour le choix des caractéristiques du matériel à déployer. On préfère souligner que cette étude de sensibilité touche seulement au coût du déploiement et d'installation de l'architecture basée sur le Fog et surtout pas du coût du fonctionnement des nœuds et des liens. Pour notre exemple, nous avons fait une étude de la sensibilité du modèle suite à la variation des deux paramètres suivants :

- **r :**La charge envoyée par chaque terminal à son EN.
- **Rn :** Le diamètre de couverture d'un EN.

Ces deux paramètres contrôlent les caractéristiques des ENs à choisir. Cette étude est proposée pour permettre un meilleur choix du paramétrage et conseiller les demandeurs du déploiement de l'architecture basée sur le Fog Manufacturing concernant :

- Quel type et portée d'EN choisir ?
- Quelle capacité et fréquence d'envoi de la charge de données par les différents terminaux ?

Les tableaux 4.4, 4.5, 4.6, 4.7, 4.8 présentent les résultats d'optimisation pour la variation du Rn quand r=5,10,15,20,25. Les paramètres à observer sont la faisabilité du modèle, le temps de résolution du modèle et le coût du déploiement.

TABLE 4.4 – Les résultats de la variation du Rn avec r=5.

Rn	5	10	14	15	20	25	30	35	40	45
Possible	Non	Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Temps	0	0	1648	2648	5102	6346	4046	7801	7861	10974
Coût	0	0	2550,87	2313,23	2029,71	1940,68	1750	1750	1750	1750
ENs activés	0	0	15	14	14	14	14	14	14	14
FNs activés	0	0	4	4	4	3	3	3	3	3
Passerelles activées	0	0	1	1	2	1	1	1	1	1

TABLE 4.5 – Les résultats de la variation du Rn avec r=10.

Rn	14	15	20	25	30	35	40	45
Possible	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Temps	2249	2392	7739	18456	13065	10934	7058	21064
Coût	2505,52	2313,23	2029,71	1940,68	1750	1750	1750	1750
ENs activés	14	14	14	14	14	14	14	14
FNs activés	4	4	4	3	3	3	3	3
Passerelles activées	1	1	2	1	1	1	1	1

TABLE 4.6 – Les résultats de la variation du Rn avec r=15.

Rn	14	15	20	25	30	35	40	45
Possible	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Temps	1132	2505	6010	16624	5481	7613	8603	11277
Coût	2505,52	2313,23	2029,71	1940,68	1750	1750	1750	1750
ENs activés	14	14	14	14	14	14	14	14
FNs activés	4	4	4	3	3	3	3	3
Passerelles activées	1	1	2	1	1	1	1	1

TABLE 4.7 – Les résultats de la variation du Rn avec r=20.

Rn	5	10	14	15	20	25	30	35	40	45
Possible	Non	Non	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Temps	0	0	912	702	4438	13709	9824	10563	9657	14359
Coût	0	0	2550,87	2313,23	2029,71	1940,68	1750	1750	1750	1750
ENs activés	0	0	15	14	14	14	14	14	14	14
FNs activés	0	0	4	4	4	3	3	3	3	3
Passerelles activées	0	0	1	1	2	1	1	1	1	1

TABLE 4.8 – Les résultats de la variation du Rn avec r=25.

Rn	14	15	20	25	30	35	40	45
Possible	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Temps	2497	1358	6115	1884	4083	4360	8000	7526
Coût	2678	2441,86	2121,71	2050,93	1887,87	1871,68	1871,68	1871,68
ENs activés	16	16	15	15	15	15	15	15
FNs activés	4	4	4	4	3	3	3	3
Passerelles activées	1	1	1	1	2	1	1	1

D'après les résultats présentés dans les tableaux 4.4, 4.5, 4.6, 4.7, 4.8, nous pouvons remarquer que la valeur de Rn influence le temps de résolution du modèle. Tant que le Rn est faible, le modèle converge rapidement et le champ des ENs est largement distribué. Cela peut s'expliquer avec le fait que la portée limitée de couverture WI-FI des ENs limite le nombre de possibilité de connexion avec les terminaux.

Nous avons commencé la variation de Rn par 14 comme pour les valeurs inférieures à cette valeur, le modèle ne converge pas et la solution est impossible. C'est parce que le champ de chaque EN n'est pas suffisant pour couvrir les terminaux et il existe un terminal éloigné de tous les nœuds EN de plus de 13 m.

Tant que l'on continue à augmenter la valeur du rayon Rn, on couvre évidemment plus de champs et le programme commence à minimiser le nombre d'ENs et par la suite des FNs en les concentrant dans la même zone. Cette concentration des noeuds est basée sur la minimisation du nombre de liens (câblage), qui sont coûteux. La minimisation du coût des liens ne concerne pas les liens WI-FI (entre les terminaux et les ENs) parce que ce n'est pas inclus dans le calcul de la fonction objectif (Ce coût n'est pas visible dans le déploiement, mais on peut avoir des coûts de fonctionnement à cause de la consommation énergétique élevée des grandes portées de WIFI). Une fois un terminal couvert par un EN, la distance n'impacte pas le coût contrairement à la longueur de connexion assurée par un lien filaire.

D'après cette étude, nous pouvons remarquer un autre point important. Notre modèle n'est pas sensible à la quantité de données générées par un terminal avec l'ensemble des valeurs r=5, 10, 15, 20, 25 testées. Cela est dû principalement à la limitation du nombre des terminaux qui peuvent être connectés à un EN en même temps ou bien des ENs à FN. Donc la charge totale ne dépasse pas la capacité maximale, respectivement, d'un EN ou d'un FN. Néanmoins pour la valeur de r=25 le modèle prend un temps tres important pour converger. Nous l'avons arrêté avec un GAP de 0.03%. cela peut s'expliquer par le fait que si $r \geq 25$ le réseau explore des espaces de solution restes impossibles avant ce qui augmente le temps de calcul.

La variation du Rn a un effet sur le nombre des terminaux connectés à un Edge. Si ce diamètre est :

— Trop faible, il y aura une contradiction avec la contrainte de la capacité de demande, la probabilité de la capacité du réseau. Chaque EN sera forcé à se connecter à un nombre très limité de terminaux et par la suite le nombre des ENs déployés va augmenter ainsi que le coût.

— Trop élevé : la majorité des terminaux vont se connecter à un seul EN, ce qui est contradictoire avec la contrainte de la capacité de demande et avec le nombre maximal des liens pour un EN. Une fois un seuil permettant que n'importe quelle cellule est capable de se connecter à n'importe quel edge les solutions données par le solver sont équivalentes même si on augmente encore le Rn.

Quand Rn atteint la valeur 20, nous avons remarqué que le modèle exige l'activation de 2 passerelles (La passerelle 388 pour certains exemples). Nous pouvons expliquer ce déploiement d'une passerelle supplémentaire par le fait que le modèle arrive à une solution ou le coût des liens commence à dépasser le coût d'une passerelle. Donc pour minimiser le coût général du déploiement de l'architecture basée sur le Fog, il est préférable d'activer une deuxième passerelle éloignée et de supprimer des liens sur des longues distances du FN concerné (dans notre exemple 414) vers la passerelle qui connecte la majorité des FNs. La solution optimale ne veut pas dire forcément moins de noeuds activés mais plutôt le moindre coût possible, et comme les liens dépendent des distances, il sera parfois nécessaire de comparer les coûts d'installation par rapport au coût du câblage. Dans les autres cas, où la valeur

de Rn est différente de 20, la différence entre le coût du câblage et le coût d'activation des noeuds est beaucoup plus grande donc le modèle active une seule passerelle.

En ce qui concerne $Rn = 25$, on remarque d'après les tableaux présentés auparavant et les courbes illustrées sur la figure 4.18, que c'est une valeur critique où le modèle met le plus de temps pour converger. Nous pouvons expliquer cette augmentation en terme de temps de résolution par le fait que 25 est une valeur intermédiaire ou médiane et elle n'est pas suffisante pour couvrir n'importe quel terminal par n'importe que EN sur l'ensemble de l'atelier.

Le temps de résolution est l'une des sorties du solver pour laquelle on peut comparer la légèreté du modèle. D'après les courbes qui tracent le temps en fonction de la variation de Rn et de r, on peut constater qu'il existe une allure similaire entre les différentes courbes malgré le fait que les valeurs ne sont pas identiques. Cette différence des valeurs de temps est dûe non seulement à la différence d'instance mais aussi à l'état de l'ordinateur lors de la résolution du modèle (travail sur autre programme, lancement d'anti-virus, etc.,) ce qui joue un rôle important dans la variation du temps de résolution. Nous remarquons également que avec toutes les variations de r, que le coût se stabilise à partir de $Rn = 30$. Après cette valeur de Rn, le coût reste fixe mais par contre le temps de résolution commence à croître à cause de l'exploration de nouvelles possibilités de solutions que des r plus petits ne le permettent pas.

D'après les résultats de l'étude de sensibilité on remarque que l'augmentation de la quantité de données dans un atelier de production demande plus de noeuds fog et ce même si la couverture du réseau WIFI est plus large. On remarque que nous avons eu besoin de plus de noeuds fog (nombre FN = 4) pour un $Rn = 25$ quand la quantité de donnée est augmentée à $r = 25$. Avec une quantité donnée inférieure le nombre de FN est de 3 uniquement.

A travers cette étude on peut donner quelques recommandations pour les industriels qui veulent installer des réseaux avec des noeuds Fog. Il faut choisir un nombre adapté à la moyenne des quantité de données générées par les terminaux de telle sorte que l'on assure un coût minimal permettant plus de flexibilité par rapport à la quantité de données générées. Cela peut être traduit par la modification des outils de production. Dans ce cas il faut que le réseau supporte cette variation sans demander une modification sur le réseau. Par exemple, dans notre cas il est préférable de choisir un $Rn = 25m$ et 4 nœud Fog.

Le modèle avec la contrainte de la probabilité de demande de réseau

Le paramétrage de cette partie est le même que les autres parties afin de garder le même exemple pour juger l'effet de cette nouvelle contrainte sur le coût de l'intégration de la solution basée sur le Fog Manufacturing et sur le nombre des noeuds activés.

La figure 4.19 schématise la nouvelle configuration des nœuds constituant l'architecture basée sur le Fog Manufacturing en intégrant la contrainte de la probabilité du réseau sur le modèle de petite taille. Les nœuds activées et les liens établis sont présentés. Le nombre des noeuds Edge est réduit de 3 ENs (7,12,30) à 2 noeuds activés (12,30). Par conséquence, le coût $Cost$ du déploiement du Fog Manufacturing est réduit de $Cost = 765$ au $Cost = 605$. Les résultats sont présentés par la figure 4.20.
On remarque une amélioration au niveau de la solution optimale grâce à l'intégration de la probabilité de la demande de réseau. Cette contrainte agit principalement pour diminuer le coût dans les zones ou les AGVs ne se déplacent pas (sur les trajectoires de déplacement), ce qui entraine une diminution des ENs dans ces zones.

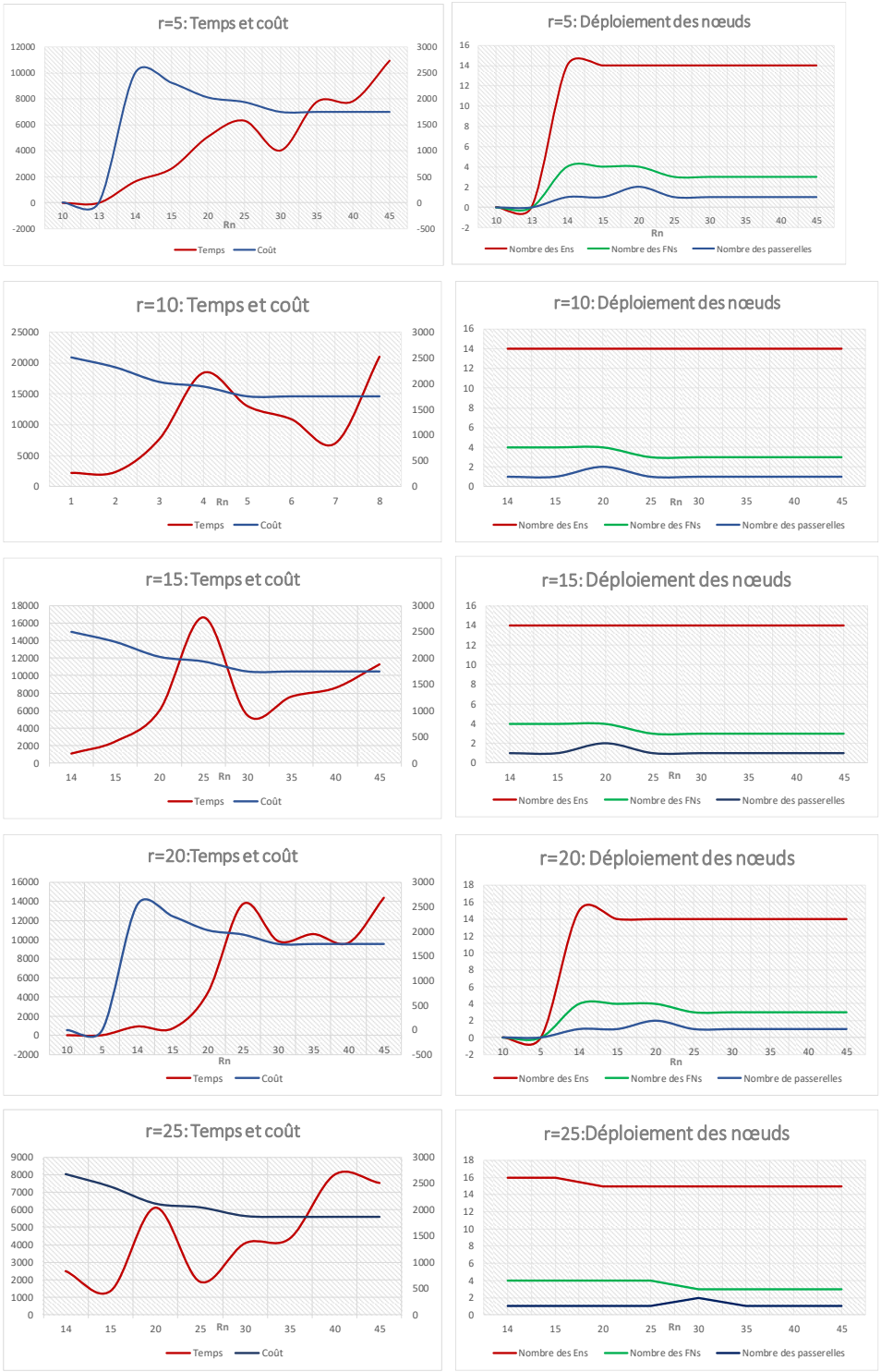


FIGURE 4.18 – les courbes de sortie de l'étude de sensibilité pour la variation de r et Rn.

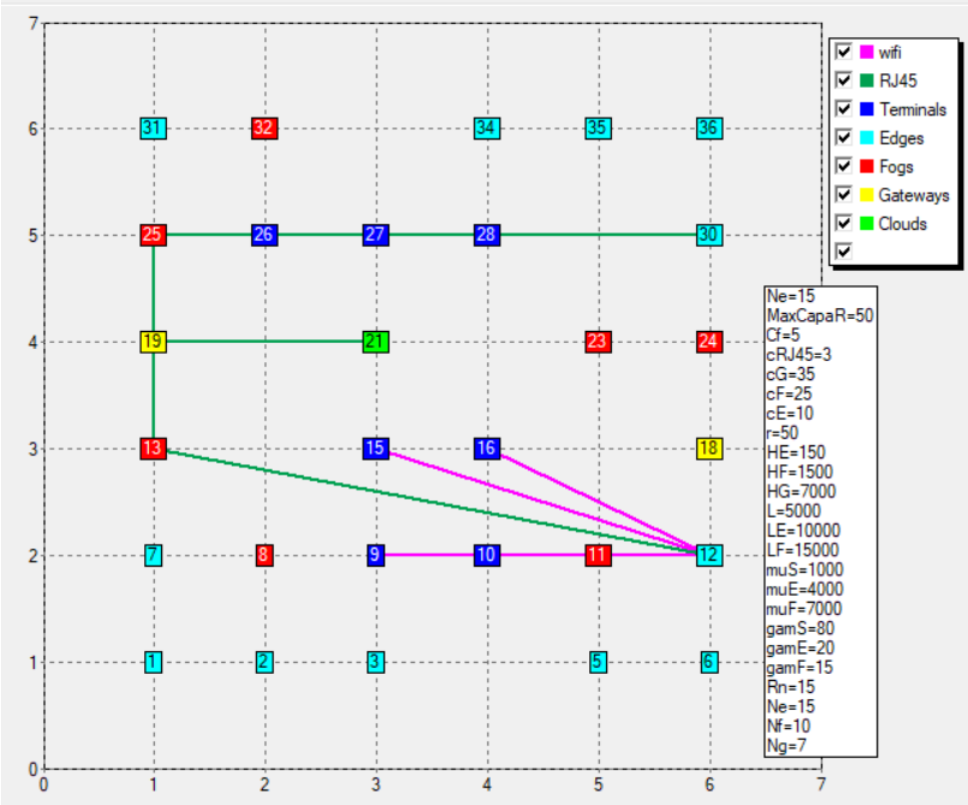


FIGURE 4.19 – Les noeuds activés pour la petite instance du modèle avec la contrainte de probabilité de demande de réseau.

```
Performing root presolve...

Reduced problem has:      30 rows      17 columns        97 elements
Presolve dropped   :    1317 rows     519 columns      1398 elements
Presolve tightened :        49 elements
Symmetric problem: generators: 2, support set: 4
 Number of orbits: 2, largest orbit: 2
 Row orbits: 2, row support: 4
Will try to keep branch and bound tree memory usage below 9.4GB

Starting root cutting & heuristics

 Its Type    BestSoln    BestBound   Sols    Add    Del     Gap     GInf   Time
 *** Search completed ***     Time:     0 Nodes:          1
Number of integer feasible solutions found is 3
Best integer solution found is   605.000000
Best bound is   604.999417
Uncrunching matrix
Solution:605
TotalCost:0      AOQ=0
TotalCost:605
Solution:
        t9      -------  e12    -------  f13    -------  g19    -------  c21
        t10     -------  e12    -------  f13    -------  g19    -------  c21
        t15     -------  e12    -------  f13    -------  g19    -------  c21
        t16     -------  e12    -------  f13    -------  g19    -------  c21
        t26     -------  e30    -------  f25    -------  g19    -------  c21
        t27     -------  e30    -------  f25    -------  g19    -------  c21
        t28     -------  e30    -------  f25    -------  g19    -------  c21
```

FIGURE 4.20 – Le coût et le temps de résolution pour la petite instance du modèle avec la contrainte de probabilité de demande de réseau.

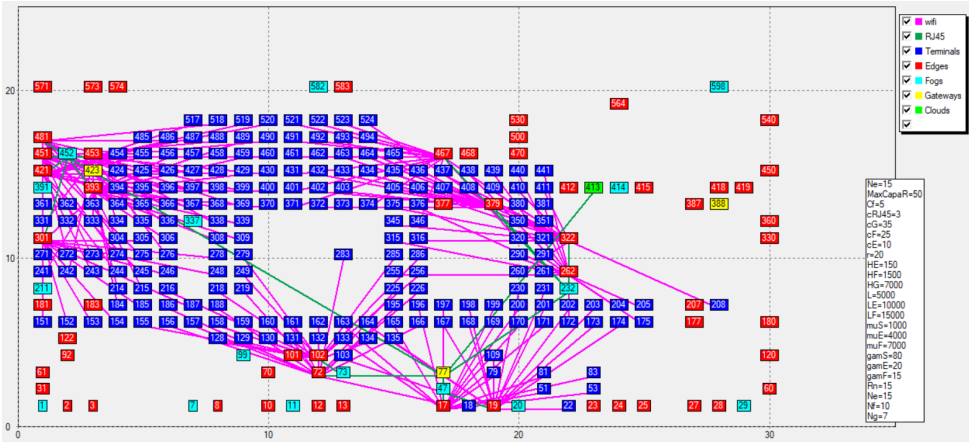


FIGURE 4.21 – Les noeuds activés pour la grande instance du modèle avec la contrainte de probabilité de demande de réseau.

La grande instance est ainsi résolue avec l'intégration de cette dernière contrainte. Le temps de calcul a considérablement diminué si on le compare à la résolution du modèle sans la contrainte de la probabilité du réseau. Les résultats présentés sur la figure 4.22 montrent une minimisation importante du coût optimisé initialement dans la section précédente, il est passé de $Cost = 6013$ à $Cost = 2312$.

Le modèle avec la contrainte du choix du lien

Lorsque l'on ajoute la contrainte du choix du lien à notre modèle nous avons remarqué que la complexité a énormément augmenté et que le nombre de solutions a atteint quelques millions. Par conséquent, nous avons proposé de diminuer la complexité et à la fois linéariser les équations par la division de la variable de décision x_{ij} en sous variable de décision pour soulager le système et alléger le calcul. Donc, on aura :

$$x = xF + xR \tag{4.36}$$

$$x = xF_{te} \wedge (xF_{ef} + xR_{ef}) \wedge (xF_{fg} + xR_{fg}) \wedge (xF_{gc} + xR_{gc}) \tag{4.37}$$

Au lieu de passer par l'ensemble des liens existants à chaque fois, cette démarche aidera à se limiter aux liens de chaque ensemble de deux noeuds consécutifs. Comme le montre l'équation (4.37), nous exigons que tous les liens soient activés pour assurer une connexion entre un terminal et le Cloud en passant par tous les autres noeuds

Les paramètres adoptés sont les mêmes que ceux du premier groupe de résolution présentés dans la section 4.5.3. Nous ajoutons la variable de la capacité maximale pour l'utilisation d'un câble RJ45, nommé MaxCapR=50.

Dans un premier temps, nous avons commencé par le test sur la petite instance. Les résultats de la simulation du modèle avec le choix du lien sont présentés par la figure 4.23, qui schématise la configuration des noeuds. Le temps d'optimisation a augmenté de 0 à 3 à cause de la nouvelle contrainte et la séparation de la variable de décision en sous variables de décision. Par contre, le coût Z est minimisé et passe de $Cost = 605$ avec la linéarisation et la contrainte de la probabilité à $Cost = 138$ avec l'ajout du choix du lien. Ces nouveaux résultats sont présentés sur la figure 4.24. Cette diminution du coût est due au choix du lien et à la réduction du coût des ENs et FNs déployés :

— ENs : De 2 à 4,

```
     403  2323.426194  2279.345249     24     23     11    1.90%     296   2512
     505  2323.426194  2286.743459     24     25     19    1.58%     110   2671
     605  2323.426194  2286.743459     24     37     20    1.58%     246   2786
     705  2323.426194  2287.931993     24     14     21    1.53%     111   2851
a    782  2313.229858  2288.833717     25     14     15    1.05%       0   2889
     805  2313.229858  2288.833717     25      5     14    1.05%     259   2893
 *** Search completed ***     Time:  2911 Nodes:        821
Number of integer feasible solutions found is 25
Best integer solution found is  2313.229858
Best bound is  2313.228385
Uncrunching matrix
Solution:2313.23
TotalCost:0      AOQ=0
TotalCost:2313.23
Solution:
        t18     -------  e72    -------  f73    -------  g77    -------  c413
        t22     -------  e19    -------  f47    -------  g77    -------  c413
        t51     -------  e17    -------  f47    -------  g77    -------  c413
        t53     -------  e17    -------  f47    -------  g77    -------  c413
        t79     -------  e17    -------  f47    -------  g77    -------  c413
        t81     -------  e17    -------  f47    -------  g77    -------  c413
        t83     -------  e17    -------  f47    -------  g77    -------  c413
        t103    -------  e19    -------  f47    -------  g77    -------  c413
        t109    -------  e19    -------  f47    -------  g77    -------  c413
        t128    -------  e102   -------  f73    -------  g77    -------  c413
        t129    -------  e102   -------  f73    -------  g77    -------  c413
        t130    -------  e102   -------  f73    -------  g77    -------  c413
        t131    -------  e72    -------  f73    -------  g77    -------  c413
        t132    -------  e72    -------  f73    -------  g77    -------  c413
        t133    -------  e17    -------  f47    -------  g77    -------  c413
```

FIGURE 4.22 – Le coût et le temps de résolution pour la grande instance du modèle avec la contrainte de probabilité de demande de réseau.

— FNs : De 2 à 1,

Dans un deuxième lieu, nous avons lancé la simulation pour la grande instance. le modèle n'a pas convergé pour l'instance complète mais on a réussi à avoir des résultats pour des instances partielle. Plusieurs pistes ont été testé afin de surmonter ce problème ce qui a conduit à des solution non complètes ou la résolution partielle du problème où des modèles trop complexes. L'utilisation des méta-heuristiques est une solution alternative intéressante qui peut être envisagée dans l'avenir. Cela dit, il faut pouvoir la comparer avec des solutions exactes afin de mesurer la distance par rapport à l'optimal. Jusqu'à l'heure de la rédaction de ce manuscrit nous n'avons pas encore résolu ce problème qui va faire la suite immédiate de ce travail.

4.5.4 Perspectives

Dans ce qui précède nous avons bien détaillé toutes les contraintes du modèle et nous avons proposé de nouvelles contraintes afin de réduire le coût et améliorer la solution. Par contre au niveau de la dernière étape qui est le choix du lien filaire entre la fibre et le câble RJ45, nous avons rencontré des difficultés concernant la taille du problème qui nous a empêché d'obtenir des résultats optimaux pour cette contrainte. Les perspectives au niveau de cette partie sont d'améliorer le code afin de minimiser encore plus le nombre des variables et par la suite essayer d'optimiser le modèle avec les nouvelles contraintes à l'aide des algorithmes génétiques. Ce choix est fait en se basant sur le fait que le problème a énormément augmenté de taille et a atteint les limites de la capacité de la méthode exacte nécessitant ainsi le passage aux méthodes méta-heuristiques. Ce modèle peut trouver des applications ailleurs que dans le réseau informatique telles que le transport multimodal et le placement de hubs de transport aériens ou logistiques.

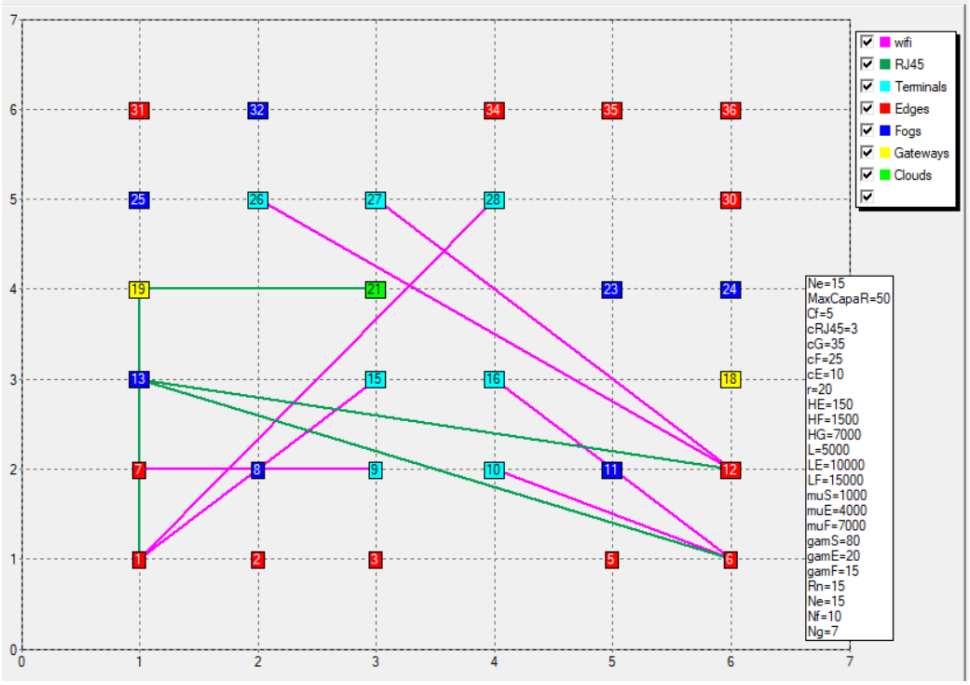


FIGURE 4.23 – Les noeuds activés pour la petite instance du modèle avec la contrainte du choix du lien.

```
    Node     BestSoln    BestBound   Sols Active  Depth     Gap     GInf   Time
     108   138.313708   113.639589      8     27     11   17.84%       7      2
     211   138.313708   116.490700      8     41     11   15.78%      69      3
     313   138.313708   120.332053      8     11     10   13.00%      99      3
 *** Search completed ***     Time:     3 Nodes:        379
Number of integer feasible solutions found is 8
Best integer solution found is   138.313708
Best bound is   138.313624
Uncrunching matrix
Solution:138.314
TotalCost:138.314
Solution:
        t9      -------  e7     -------  f13    -------  g19    -------  c21
        t10     -------  e6     -------  f13    -------  g19    -------  c21
        t15     -------  e1     -------  f13    -------  g19    -------  c21
        t16     -------  e6     -------  f13    -------  g19    -------  c21
        t26     -------  e12    -------  f13    -------  g19    -------  c21
        t27     -------  e12    -------  f13    -------  g19    -------  c21
        t28     -------  e1     -------  f13    -------  g19    -------  c21
```

FIGURE 4.24 – Le coût et le temps de résolution pour la petite instance du modèle avec la contrainte du choix du lien.

4.6 Conclusion

Dans ce chapitre, un modèle d'optimisation de coût de déploiement de l'architecture Fog Manufacturing pour une application de logistique a été présenté. Le modèle a été expliqué à travers différentes contraintes de façon à analyser leurs intérêts. Ensuite nous avons présenté notre étude portant sur la linéarisation et l'introduction de deux nouvelles contraintes concernant la probabilité de demande du réseau et le type de lien choisi (Fibre ou RJ45). Les résultats de la résolution de ce modèle avec les différents scénarios ont été présentés et comparés. L'analyse des résultats a montré que les nouvelles contraintes ont amélioré notablement la solution optimale du modèle en terme de coût et de nombre de noeuds activés. Dans la suite de ce mémoire, nous allons nous attacher à présenter l'implémention des résultats de ce modèle sur un cas réel pour évaluer ses performances globales vis-à-vis des critères comme la latence et l'efficacité de filtrage des données.

Chapitre 5

Déploiement d'une plateforme Fog Computing pour le traitement en temps réel d'un CPS de production

5.1 Introduction

Le chapitre précédent a présenté un modèle d'optimisation du coût du déploiement du Fog Manufacturing pour une application de logistique dans le contexte de l'Industrie 4.0. Outre le coût optimisé nous avons obtenu comme résultat les coordonnées des noeuds à activer. Dans ce chapitre, nous évaluons expérimentalement les résultats du modèle d'optimisation du coût pour étudier la faisabilité du Fog Manufacturing dans un cas réel. Cette implémentation aidera à évaluer les performances du Fog Manufacturing en termes de collecte et de gestion des données distribuées dans un atelier industriel présentant des CPS.

Cette expérimentation a été appliquée sur le démonstrateur d'usine de futur du laboratoire LINEACT CESI. Dans un premier temps, le principe de la conteneurisation est étudié. Ce principe sera adopté pour la mise en place de l'architecture basée sur le Fog Manufacturing. Dans un second temps, une évaluation de l'approche est faite afin de stocker la localisation des CPS, choisir les équipements et identifier les différents types de connectivité. Pour finir, nous présentons des éléments afin de construire un environnement d'évaluation plus réaliste, en interconnectant les terminaux du démonstrateur (principalement des AGVs) aux ENs, par la suite les ENs aux FNs et enfin établir les liens avec le Cloud via les passerelles. L'objectif visé est de tester la faisabilité d'une solution basée sur le Fog Manufacturing et d'évaluer ses performances pour un simple traitement de données (Exemple : Filtrage de données).

5.2 Conteneurisation

Depuis quelques années, les conteneurs ont révolutionné divers domaines en informatique. L'utilisation de la technique de conteneurisation a vu une croissance massive dans le marché des applications informatiques. Ils ont changé la façon de concevoir des applications, tout en permettant aux développeurs de bénéficier de valeur ajoutée en termes de productivité et performance des applications.

5.2.1 Définition

Dans le contexte des applications informatiques, un conteneur est utilisé pour bénéficier d'un espace spécifique d'exécution dédié à une application logicielle. Il peut

se présenter sous la forme d'un type abstrait de données, d'une structure d'informations ou encore d'une classe. Les conteneurs permettent aussi le test d'applications en cours de développement. L'usage d'un conteneur permet en effet de tester des applications en cours de développement, ainsi que des logiciels. L'intérêt est de disposer d'un environnement isolé pour réaliser tous les contrôles nécessaires avant le déploiement d'une architecture (JDN., 2020). Il est donc plus facile de distinguer les dysfonctionnements, les failles de sécurité et d'éventuels problèmes de stabilité.

Actuellement, les technologies du cloud computing transforment la technologie traditionnelle par de nouvelles pratiques plus efficaces (BERNSTEIN, 2014). Les conteneurs en sont un exemple. Ils sont considérés comme l'avenir des machines virtuelles (VM). La virtualisation basée sur les conteneurs est également connue sous le nom de virtualisation du système d'exploitation, qui permet à la couche de virtualisation de s'exécuter comme une application dans le système d'exploitation (SE/OS). Les conteneurs sont apparus comme des micro machines virtuelles, plus légères et plus efficaces car un seul système d'exploitation (OS) gère tous les appels matériels.

Le principe de la conteneurisation est généralement le suivant : les demandes sont emballées sous forme d'images qui contiennent une partie d'un système de fichiers avec les bibliothèques, les exécutables et les fichiers de configuration nécessaires, etc. Les images sont stockées dans des dépôts centralisés où elles sont accessibles à partir de n'importe quel serveur de calcul. Pour déployer un conteneur, il faut commencer par télécharger l'image du dépôt et l'installer localement, à moins que l'image ne soit déjà mise en cache dans le nœud de calcul. Le démarrage d'un conteneur à partir d'une image installée localement est considéré plus rapide que de lancer le processus qui constitue l'application du conteneur. Le temps de déploiement de chaque conteneur est donc déterminé par le temps nécessaire pour télécharger, décompresser, vérifier et localement installer l'image avant de démarrer le conteneur lui-même.

La technologie des conteneurs aide à atténuer le problème de l'exécution des applications logicielles dans divers environnements. Dans la conteneurisation, le SE est partagé par les différents conteneurs plutôt que cloné pour chaque VM. Les conteneurs se composent d'un environnement d'exécution qui comprend l'application logicielle, ses dépendances, les bibliothèques, les binaires et les fichiers de configuration. L'application logicielle s'exécute sur le conteneur et ne dépend pas de l'environnement hôte, à l'exception du système d'exploitation. Un conteneur peut avoir plusieurs applications et chaque application aura son propre environnement. Lorsque le conteneur est déployé dans un environnement différent, le système d'exploitation sera partagé entre les applications.

Actuellement, le modèle de conteneur n'est pas complètement représenté. Apparemment, la technologie des conteneurs semble sûre, comme elle contient toutes les dépendances dans un seul emballage. Cependant, elle n'assure pas sa sécurité. Les plateformes de conteneurs sont également vulnérables, tout comme les autres plateformes de cloud computing. Il existe un certain nombre de menaces qui proviennent de l'intérieur et de l'extérieur (BUI, 2015).

5.2.2 Conteneurs vs. machines virtuelles

Les machines virtuelles et les conteneurs permettent de répartir le charge de travail (ou workload) du matériel, mais les deux approches possèdent d'importantes différences. Le principe de la virtualisation est ainsi résumé : c'est la simulation d'une ou de nombreuses machines ou systèmes physiques via un hyperviseur afin de les exécuter sous forme d'une ou nombreuses machines virtuelles-(VM) sur un

serveur. Chaque machine virtuelle est munie d'un SE sur lequel les applications sont exécutées. Les technologies de virtualisation suivantes sont considérées parmi les plus connues :

— VMWare : il s'agit de la technologie la plus populaire, elle est caractérisée par un fort écosystème, la diversification des outils de configuration et d'administration, la possibilité de changer des VMs d'un serveur à un autre et la haute-disponibilité des services,
— Xen (détenue par Citrix),
— KVM (qui s'appuie sur OpenStack pour la partie Hyperviseur et bénéficie du support RedHat).

Concernant la conteneurisation, le conteneur exécute ses applications en faisant appel directement au SE à partir de sa machine physique. Il partage le noyau de la machine à partir duquel les applications sont implémentées. Dans de nombreux cas, il n'est pas obligatoire d'avoir l'assistance d'un hyperviseur pour effectuer la gestion de toutes les VMs qui s'exécuteront par les conteneurs utilisés. Parmi les nombreuses technologies de conteneurisation, le Docker et par la suite le CoreOs sont considérés respectivement les plus connus et utilisés. Le Docker apporte plus de rapidité de déploiement et une gestion fluide de la création des conteneurs et de l'installation des applications. L'avantage d'une telle approche est le faible poids des conteneurs sur lequel la migration des conteneurs d'une machine physique à l'autre est fluide. Ce qui rend la portabilité d'une application d'un Cloud à l'autre, la duplication d'une application sur deux Clouds différents ou l'équilibre du workload plus simple.

En résumant, la virtualisation est basée sur les conteneurs ce qui rend la couche de virtualisation s'exécutant comme étant une application dans le SE. Lorsqu'on compare des conteneurs avec des VMs, les conteneurs sont beaucoup plus légers. Pourtant, chaque VM comprend son propre SE ainsi que l'application. En ce qui concerne les conteneurs, chacun partage le noyau de l'"SE" hôte avec d'autres conteneurs. Le tableau 5.1 et la figure 5.1 ci-dessous clarifient la différence entre les VMs et les conteneurs :

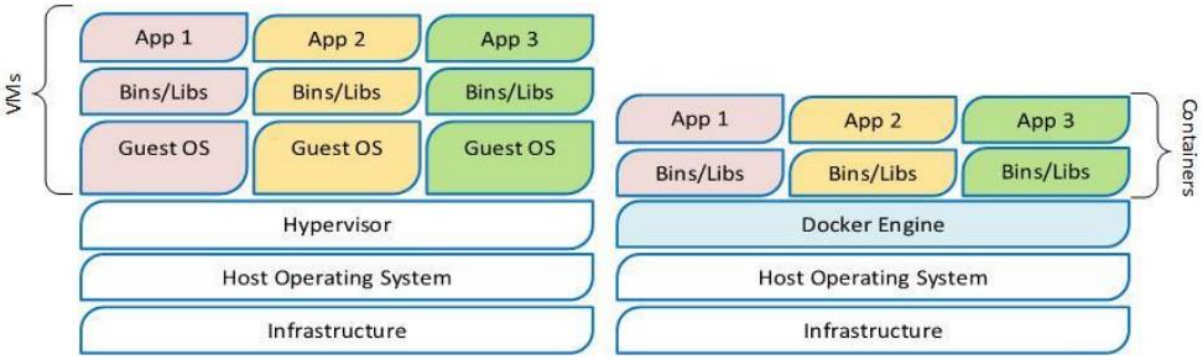


FIGURE 5.1 – Containers vs. VMs (KOSKINEN, 2016).

5.2.3 Les plateformes de conteneurisation

Les plateformes de conteneurisation sont conçues pour faciliter le déploiement des conteneurs. Malgré l'existence d'autres plateformes comme Rocket et OpenVZ,

TABLE 5.1 – Conteneurs vs. VMs.

Paramètres	VM	Conteneurs
Système d'exploitation	Chaque VM s'exécute sur un matériel virtuel et le noyau est chargé dans sa propre région de la mémoire.	Tous les conteneurs partagent le même OS et le même noyau. L'image du noyau est chargée dans la mémoire physique.
Communication	A travers les câbles Ethernet.	A travers les mécanismes IPC standards comme les signaux et les sockets.
Sécurité	Dépend de l'implémentation de l'hyperviseur	Un contrôle d'accès est obligatoire
Performance	Les VMs souffrent d'une surcharge qui s'ajoute lors de la traduction des instructions de l'OS vers L'OS hôte.	Les conteneurs fournissent une performance très proche de celle de l'OS hôte
Isolation	Le partage des fichiers et des librairies entre les VMs invitées ou entre une VM et le système hôte n'est pas possible	Les sous-répertoires peuvent être partagés d'une manière transparente
Temps de démarrage	Les VMs prennent quelques minutes pour démarrer	Les conteneurs démarrent au bout de quelques secondes
Stockage	En Gigaoctets	En Megaoctets

les plateformes Linux et la plateforme Docker sont les plus populaires.

Les plateformes Linux

Les conteneurs Linux constituent une solution afin de virtualiser les logiciels au niveau du SE au sein du noyau Linux et permettre à une machine physique hôte d'exécuter de nombreuses instances Linux isolées « conteneurs Linux », ou LXC (LinuX Containers).

— **LXC :** LXC se base sur les groupes de contrôle, ce qui ressemble bien aux conteneurs Docker ou CoreOS. Cette ressemblance est due au fait que le LXC était la technologie sous-jacente qui a fait le succès du Docker et CoreOS. LXC a été à l'origine de la révolution des conteneurs il y a plusieurs années, et les principes LXC et le code LXC restent au coeur du développement des conteneurs (**containers**),

— **LXD :** LXD est défini comme une extension de LXC. L'un des objectifs fondamentaux du développement de LXD fut de rendre la gestion des conteneurs LXC aussi pratique que celle des machines virtuelles. La manière la plus technique de le définir est de le décrire comme une API REST qui se connecte à libxlc, la bibliothèque logicielle LXC. LXD, qui est écrit dans Go, crée un système daemon auquel les applications peuvent accéder localement à l'aide d'un socket Unix ou via le réseau via HTTPS (**containers**). Il est à noter qu'un système hôte peut exécuter de nombreux conteneurs LXC en utilisant un seul système daemon, ce qui simplifie la gestion et réduit les frais généraux. LXD offre, également, des fonctionnalités avancées non disponibles auprès de LXC, notamment la migration des conteneurs en direct et la possibilité de prendre une capture d'un conteneur en cours d'exécution.

Les conteneurs Linux proposent également une isolation complète de leur espace de noms. Les fonctions telles que les systèmes de fichiers, les identifiants réseau et utilisateur, ainsi que tout autre élément généralement associé aux SEs, peuvent donc être considérés comme «uniques» pour chaque conteneur.

La plate-forme Docker

Parmi les logiciels qui lancent des applications sur les conteneurs, on trouve le Docker qui construit, assemble et gère ces applications (AHMED et PIERRE, 2018). Actuellement, Docker est la plate-forme de conteneur la plus populaire. Bien que l'idée d'isoler des environnements remonte assez loin dans le temps, et qu'il y ait eu d'autres logiciels de conteneurs par le passé, Docker est apparu sur le marché au bon moment et était open source depuis le début, ce qui a probablement conduit à sa dominance actuelle.

Solomon Hykes, qui a lancé Docker en tant que projet interne open-source chez dotCloud et qui a immensément contribué au succès des conteneurs précise : "Vous allez tester en utilisant Python 2.7 et par la suite exécuter sur Python 3, et quelque chose de bizarre se produira. Ou peut-être vous comptez sur le comportement d'une certaine version d'une bibliothèque SSL et une autre sera installée. Vous allez exécuter vos tests sur Debian et la production est sur Red Hat et toutes sortes de choses bizarres arrivent" (BERNSTEIN, 2014). Par conséquent, le Docker a été conçu pour utiliser différentes interfaces pour accéder aux caractéristiques de virtualisation du noyau Linux. Au début, Docker bénéficiait de LXC comme étant son environnement d'exécution. Néanmoins, lors de la sortie de la version 0.9, Docker a lâché LXC et a introduit sa propre bibliothèque. Le langage de programmation Go a été adopté afin de concevoir la bibliothèque Docker libcontainer (YASRAB, 2018).

Docker est actuellement open source, ce qui apporte un moyen automatique de déploiement rapide des applications Linux dans les conteneurs. Les développeurs et les administrateurs système bénéficient, tous les deux, de Docker, qui fait partie de plusieurs chaînes d'outils DevOps (développeurs + opérations). Pour les développeurs, cela signifie qu'ils peuvent se concentrer sur l'écriture du code sans se soucier du système sur lequel il sera implémenté. Cela leur permet également de choisir parmi une multitude de programmes déjà conçus pour fonctionner dans un conteneur Docker. En ce qui concerne les opérations, Docker donne de la flexibilité et réduit potentiellement le nombre de systèmes nécessaires en raison de son faible encombrement et de son moindre coût (BACIS et al., 2015).

Par ailleurs, l'objectif des plate-formes de conteneurisation, comme Docker décrit dans cette partie, est d'appuyer l'intégration rapide des applications dans les conteneurs, ainsi que gérer la mise à jour ces conteneurs et non à leur création (cela reste le rôle du noyau Linux). La plate-forme Docker n'est pas apparue pour remplacer les conteneurs Linux mais ils utilisent plutôt LXC comme une base de conteneurisation et ajoute plus de performances en terme de capacité. Comme exemple de ce passage en capacité supérieure, la plate-forme Docker permet le passage des applications entre les machines qui exécutent Docker. Le LXC seul ne peut pas déplacer l'application sur une autre machine possédant plus de capacités.

5.2.4 Moteurs d'orchestration des conteneurs

Les plate-formes de conteneurisation sont actuellement employées pour créer des ensembles d'applications hautement évolutifs à la demande. Cette création est admissible pour un nombre limité de conteneurs mais imaginons que l'on en possède des centaines ou des milliers, alors la gestion des conteneurs avec leurs ensembles d'applications devient très difficile, surtout lorsque le nombre d'instances s'accroît dynamiquement suivant la demande des applications.

Afin de surmonter un tel problème de gestion du cycle de vie des conteneurs et des services, l'orchestration de conteneurs se présente comme une solution qui

automatise la planification, le déploiement, l'évolutivité, l'équilibre de charge d'exécution, la disponibilité et la préparations des conteneurs à la mise en réseau.

Les moteurs de conteneurisation sont considérés des outils de gestion de l'infrastructure DevOps. Les COE (Container Orchestration Engine) fournissent une couche d'abstraction entre les ressources et les conteneurs d'applications qui s'exécutent sur ces ressources-là. Ils fournissent des services de haut niveau qui permettent la planification des conteneurs, qui consistent à exécuter des fonctions telles que le démarrage et l'arrêt, la distribution des conteneurs sur les ressources mises en commun, la récupération des conteneurs défectueux et la mise à niveau des applications via des conteneurs.

5.2.5 Exemple de moteurs d'orchestration

La liste des moteurs d'orchestration des conteneurs est réellement longue, citons par exemple : Kubernetes, OpenShift, Nomade, Docker Swarm, Docker Compose, MiniKube, Marathon, Cloudify, Rancher, Mesos, etc.,.

Les moteurs d'orchestration les plus utilisés par les entreprises et avec la plateforme Docker sont les suivants :

— **Kubernetes :** Kubernetes (également connu sous le nom "k8s") a été publié en Juin 2014, et est écrit en Go. Traduit du grec ancien, le mot Kubernetes signifie "Helmesman" ou "timonier, pilote" en français. Le projet est basé sur l'expérience de Google dans la gestion de conteneurs à grande échelle. Très largement utilisé, Kubernetes a derrière lui une grande communauté. C'est une plate-forme open source qui a été initialement conçue et utilisée par Google, pour son propre service Container as a Service (CaaS), appelé Google Container Engine (GKE). Actuellement elle est maintenue par la Cloud Native Computing Foundation.
Un certain nombre d'autres plates-formes prennent en charge Kubernetes, notamment Red Hat OpenShift et Microsoft Azure. Or, Docker est le moteur des conteneurs le plus présent, supporté par Kubernetes, comme il prend en charge à la fois la configuration déclarative et l'automatisation. Il peut aider à automatiser le déploiement, l'évolutivité, la gestion de la charge de travail et de nombreux services de conteneurisation. La distinction des Kubernetes de Swarm et Mesos est basée sur le concept de «pods», qui constituent des groupes de conteneurs programmés ensemble pour constituer un «service», dans la terminologie de Kubernetes (VOHRA, 2016). Comme le montre la fi-

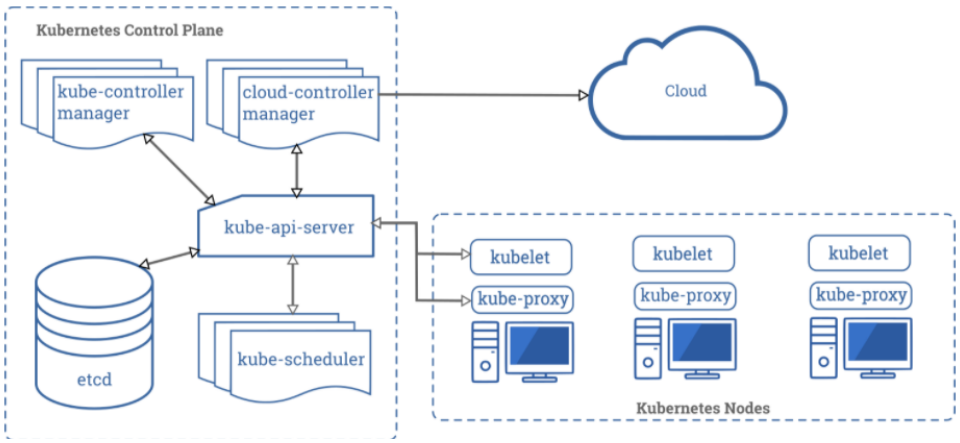


FIGURE 5.2 – Le moteur d'orchestration Kubernets(*Kubernetes*).

gure 5.2 Kubernetes aide à établir la communication entre les différents utilisateurs des applications, les composants du groupe des noeuds kebelet et les composants externes. Le plan de contrôle Kubernetes et les nœuds kebernetes s'exécutent sur un groupe de nœuds qui forment ensemble le cluster. La charge de travail de l'application se forme d'un ou de plusieurs pods qui s'exécutent sur des nœuds de travail. Le plan de contrôle gère les pods et les nœuds de travail.

— **Docker Swarm :** Docker Swarm est un moteur d'orchestration des conteneurs de Docker basé sur un modèle déclaratif. Initialement publié en Novembre 2015, il est également écrit dans Go. Swarmkit est la version recommandée de Docker pour ceux qui souhaitent utiliser Swarm. Swarm est étroitement intégré à l'API Docker, ce qui le rend parfaitement compatible avec Docker. Les mêmes primitives qui s'appliquent à un seul cluster de Docker sont utilisées avec Swarm. Cela peut simplifier la gestion de l'infrastructure des conteneurs, car il n'est pas nécessaire de configurer un moteur d'orchestration distinct ou de ré-apprendre les concepts Docker pour utiliser Swarm (ibid.). Le Docker Swarm comporte deux composants essentiels :
  — **Manager**, qui est un leader élu en se basant sur l'algorithme Raft et qui se charge de la gestion d'attribution des tâches aux nœuds de travail. Le manager assure les décisions de gestion et d'orchestration des tâches.
  — **Le nœud de travail**, qui reçoit les tâches à exécuter de la part du manager.

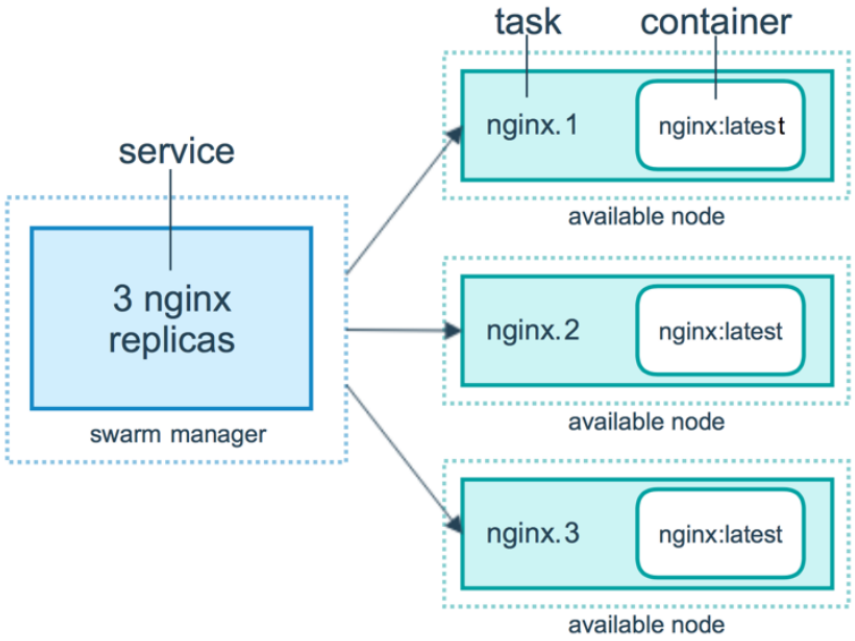


FIGURE 5.3 – Le moteur d'orchestration Docker Swarm (*Swarm*).

— **Mesos :** Apache Mesos version 1.0 a été publié en Juillet 2016, mais ce moteur d'orchestration a des racines remontant à 2009, quand il a été initialement développé par des étudiants en doctorat à UC Berkeley. Contrairement à Swarm et Kubernetes, Mesos est écrit en C++. Parmi les fonctionnalités disponibles dans Mesos, on peut mentionner la prise en charge de plusieurs types de moteurs de conteneurs, y compris Docker, ainsi que l'existence d'une interface utilisateur Web et la possibilité de fonctionner sur plusieurs systèmes d'exploitation, y compris Linux, OS X, et même Windows. En raison de sa

complexité et de sa flexibilité, Mesos est plus contraignant en apprentissage que Docker Swarm. Mais cette flexibilité et cette complexité sont également des points forts qui permettent à des entreprises comme Twitter et Airbnb d'utiliser Mesos pour gérer leurs applications à grande échelle (SAHA et al., 2016).

5.3 Conception et spécifications des besoins fonctionnels

Notre projet propose une solution basée sur une architecture de Fog Computing, appliquée à un environnement d'Industrie 4.0.

Ce chapitre sera consacré à la présentation des diagrammes de conception (UML). Par la suite, on passera à la justification des différents choix techniques adoptés lors du développement de la solution. Enfin, on présente les différentes interconnexions qui vont assurer l'implémentation de l'architecture réelle basée sur le Fog Manufacturing.

Le but de cette partie est de répondre aux besoins potentiels de l'Industrie 4.0 en implémentant une architecture Fog Manufacturing. Il nous faut donc les exprimer clairement avant d'imposer (ou d'orienter vers) une solution finale. Généralement ces besoins sont formulés sous forme d'exigences fonctionnelles, qui sont l'expression de ce que la solution délivrée devrait être. Après étude et identification des besoins fonctionnels, on a procédé à la définition des différents cas d'utilisation et des scénarios d'application.

5.3.1 Diagramme de cas d'utilisation

Le diagramme des cas d'utilisation représente un modèle simplifié, afin de décrire le comportement du fonctionnement d'une application ou d'un système tel qu'un utilisateur externe le voit. Le diagramme de cas d'utilisation exprime les besoins des utilisateurs en les traduisant en fonction des cas d'utilisations qu'utilisent les acteurs du service offert.

Les composants d'un diagramme d'utilisation sont généralement :

— Les acteurs, ils sont les utilisateurs qui communiquent avec une application ou un système. Ils peuvent être une personne, une autre application ou un système externe. Ils sont donc nécessairement des entités extérieures qui utilisent les données,

— Le système, c'est le scénario spécifique des actions et des échanges entre les acteurs et le système,

— Les objectifs, sont les résultats finaux des cas d'utilisation. La réussite d'un diagramme dépend de la description de ses activités.

Description des acteurs

Un acteur représente une abstraction d'un rôle joué par des entités externes au système qui interagissent directement avec lui. Dans cette application, nous identifions deux types d'acteurs, l'administrateur du système d'information et le robot :

L'administrateur du système d'information : Il s'agit d'un acteur principal qui gère les ressources Fog (Les noeuds constituant une architecture basée sur le Fog) et les serveurs Cloud. Ainsi, il assure la collecte et l'acheminement des données vers les bases du Cloud.

Le robot : Il s'agit d'un acteur secondaire qui intervient dans notre environnement en générant de l'information en temps réel. Les informations du robot contiennent sa tâche actuelle, son niveau de batterie, sa position, etc.,.

Diagramme de cas d'utilisation globale

La figure 5.4 montre le diagramme de cas d'utilisation globale de l'application qui implique tous les acteurs de notre système.

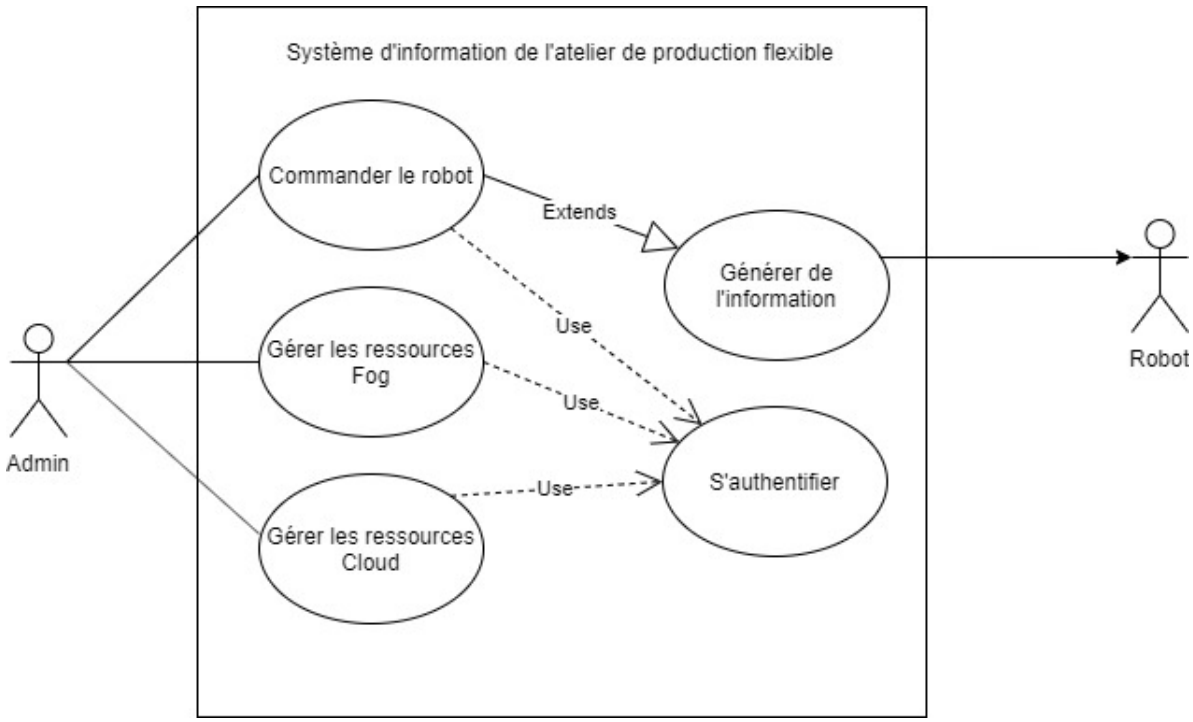


FIGURE 5.4 – Diagramme de cas d'utilisation globale du système.

Le diagramme de cas d'utilisation globale du système est développé davantage dans le tableau 5.2.

5.3.2 Diagramme de séquences

Les diagrammes de cas d'utilisation modélisent à quoi sert le système, en assurant un ordonnancement des interactions possibles. Le diagramme de séquences permet de décrire les différentes interactions ou messages entre les acteurs et le système d'information, appelés dans ce cas les lignes de vie, selon un ordre chronologique ayant une dimension verticale et s'écoule de haut en bas.

Un message traduit une communication bien spécifique entre des lignes de vie. Il existe différent types de messages, les plus utilisés sont les suivants :

— L'envoi d'un signal,
— L'invocation d'une opération,
— La création ou la destruction d'une instance.

La figure 5.5 représente le diagramme de séquences de notre environnement d'application.

Titre	Accéder aux systèmes d'information
Description	Ce cas d'utilisation permet aux utilisateurs ayant le compte administrateur d'accéder aux ressources Fog et Cloud.
Acteurs	Administrateurs, Robot
Pré-conditions	Les administrateurs doivent être authentifiés pour accéder à ces fonctionnalités.
Scénario nominal	1. L'utilisateur s'authentifie. 2. L'administrateur affecte une mission au robot. 3. Le robot génère de l'information. 4. L'administrateur peut accéder aux ressources Fog. 5. L'adminstrateur peut consulter les données stockées au niveau du Cloud.

TABLE 5.2 – Cas d'utilisation global du système.

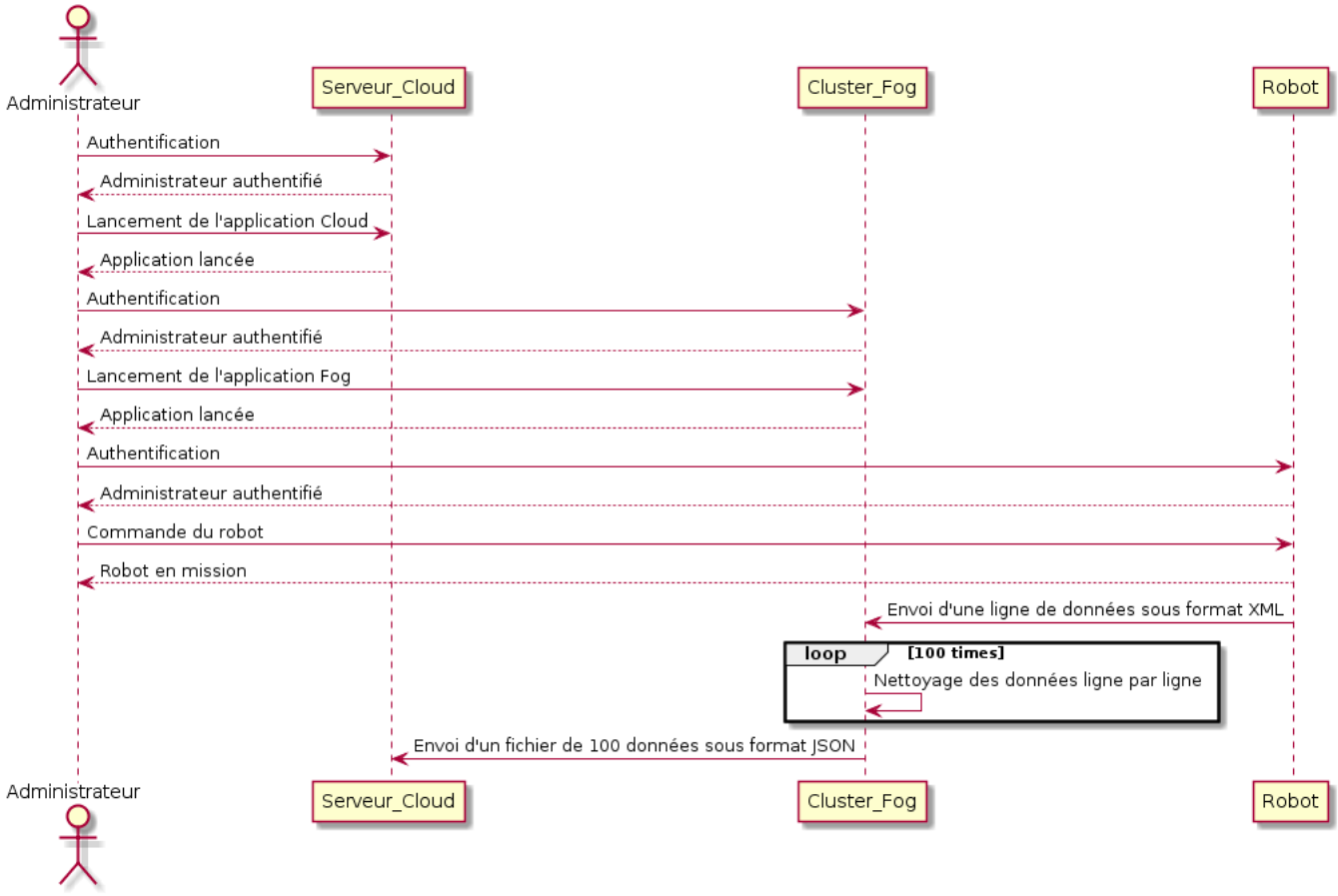


FIGURE 5.5 – Diagramme de séquence de l'environement d'application.

5.4 Choix Technologiques

Il est important de sélectionner les outils technologiques qui aideront à développer la solution. Ce choix est principalement basé sur le fonctionnement optimal du système, l'impact de la future croissance technologique et le coût de la mise en place. Dans cette section, on présente les moyens technologiques utilisés afin de mettre en place notre solution.

5.4.1 Les FNs : Raspberry Pi

Le choix des FNs

Pour les équipements du Fog computing ou précisément les FNs, il était prévu que l'on choisisse les routeurs Cisco dédiés à l'exploitation industrielle de la solution Fog, en considérant leur capacité à gérer et analyser les données au niveau de bord du réseau, leur évolutivité, flexibilité, sécurité, capacité de supporter de nombreux protocoles de communication, ainsi que la capacité à filtrer les données provenant des objets connectés.

Néanmoins, on a rencontré au cours de la thèse des contraintes liées aux moyens d'investissement et aux délais (incluant les contraintes liées à la crise sanitaire). Par conséquent, on a choisi un outil beaucoup plus accessible que les routeurs Cisco, et plus simple à manipuler afin de montrer l'applicabilité du Fog Manufacturing sur un cas réel.

L'architecture basée sur le Fog computing doit nécessairement s'appuyer sur un très grand nombre de FNs répartis géographiquement qui ne sont pas des serveurs puissants, mais des machines beaucoup plus faibles. En fait, l'une des options envisagée consiste à utiliser à cette fin des ordinateurs monocarte tels que les Raspberry Pi. Cette technologie est parmi les plus utilisées dans le contexte de l'IoT et dans les applications basées sur le Fog computing.

Malgré leurs limites matérielles évidentes, les Raspberry Pi offrent d'excellents rapports performances/coûts/énergie et sont bien adaptés aux scénarios où la taille physique et la consommation d'énergie du dispositif sont des facteurs importants pour un déploiement réel. Cependant, une application basée sur des FNs Raspberry Pi reste un défi difficile à relever. En particulier, ces machines ont des performances d'entrée/sortie très limitées.

Dans ce travail, on se concentre donc sur le déploiement des conteneurs Docker dans des cartes Raspberry. On part du principe que les serveurs Fog ont une capacité de stockage limitée et que l'on ne peut pas s'attendre à ce qu'elles gardent en mémoire cache les images des conteneurs de nombreuses applications pouvant être utilisées simultanément dans une architecture Fog. Le déploiement des images des conteneurs peut être très lent, de l'ordre de plusieurs minutes selon la taille de l'image du conteneur et l'état du réseau. Toutefois, de tels retards sont inacceptables dans des scénarios où les applications sont sensibles aux latences, où les objets connectés sont mobiles et où les conteneurs doivent être créés dynamiquement lorsqu'un robot mobile change de zone de couverture. Par contre, les capacités limitées du matériel ont permis de mettre en évidence les inefficacités de ce processus, qui seraient plus difficiles à cerner en utilisant des machines serveurs plus rapides.

Ce choix est propre à l'expérimentation de petite taille qui vise à prouver l'applicabilité et l'efficacité du Fog Manufacturing. De plus les capacités limitées du matériel sont inadéquates au déploiement sur des problèmes de grande taille.

Evolution des cartes Raspberry Pi

Le Raspberry Pi est un nano ordinateur, appelé aussi ordinateur à carte unique ou monocarte, créé par David Braben dans le cadre de son association caritative Raspberry Pi. Le premier modèle du Raspberry Pi était en 2011 avec des caractéristiques plus au moins médiocres (le processeur à 700 MHz et la mémoire RAM de 128Mo), arrivant de nos jours au Pi3 qui porte des spécifications satisfaisantes que nous allons citer par la suite (RASPBERRYPI., 2020).

Fiche technique du Raspberry Pi3

Comme le montre la figure 5.6 (ibid.), la carte Raspberry Pi3 porte :

— Un processeur Cortex-A53 (ARMv8) 64-bit à 1.4GHz - BCM2837B0.
— Une mémoire vive de 1GB LPDDR2 SDRAM
— Un module Wifi 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac
— Un module Bluetooth 4.2
— Un port Ethernet 10/100
— Quatre ports USB 2.0
— Un lecteur Micro SD
— 5V/2.5A DC power input
— Un connecteur HDMI
— Le système d'exploitation convenable pour la carte Raspberry Pi est le Raspbian qui est un OS libre basé sur Debian GNU/Linux optimisé. Pour faciliter diverses tâches, la fondation Raspberry Pi a conçu le système d'exploitation NOOBS.

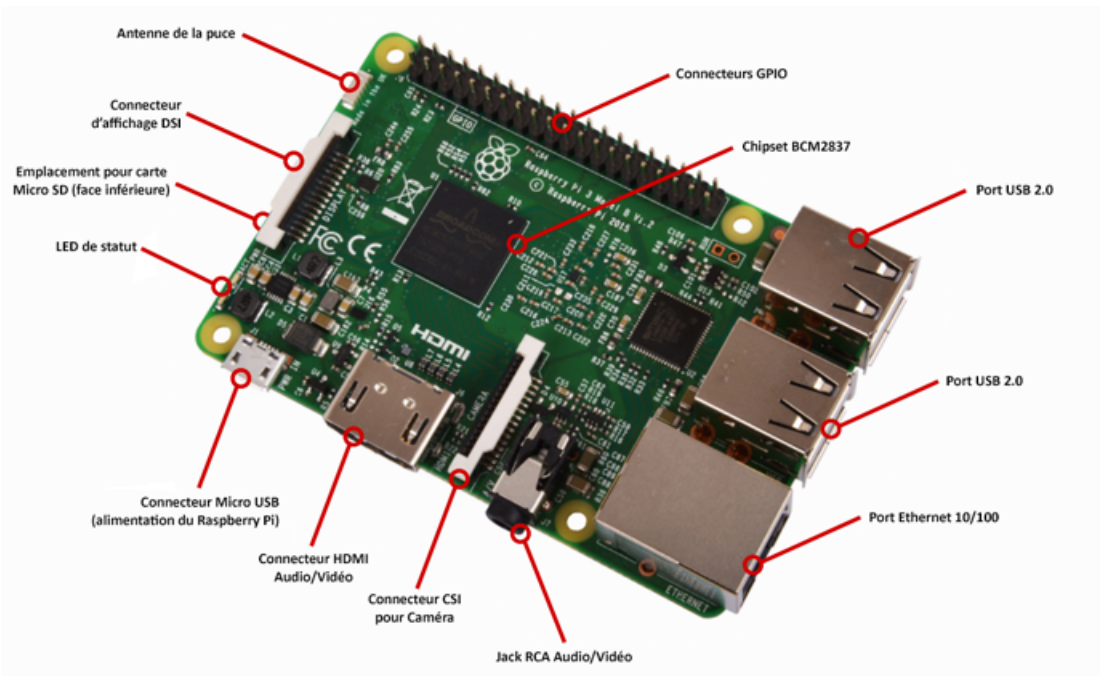


FIGURE 5.6 – Carte Raspberry Pi3 (RASPBERRYPI., 2020).

Comme détaillé auparavant, le Raspberry Pi a la taille d'une carte de crédit donc il est très léger (45g). L'unité de Raspberry posséde un coût très faible (aux alentours de 37 euros) et elle a des capacités puissantes, ce qui amène à un rapport qualité prix très raisonnable. Enfin, cette dernière comporte une diversité de ports tels que USB, HDMI, ethernet, etc. De plus, il est important de souligner que derrière cette

carte, il y a une grande communauté, très active, ce qui permet de trouver différents projets, forums, blogs qui travaillent sur son développement pour de nombreuses applications comprenant les applications basées sur le Fog computing.

5.4.2 Langage de programmation, Python

Python est un langage de programmation facile à apprendre, à comprendre, à lire et à écrire.

Python rend la compréhension de la structure du programme plus facile à apprendre que d'autres langages populaires, comme le montre la figure 5.7 (**python**). La couleur orange présente les langages de programmation les plus utilisés, par contre la grise présente celles les moins utilisés. La largeur des cercle sur la figure dépend du pourcentage d'utilisation du langage de programmation concerné.

Ce langage est polyvalent. Il exécute des sites Web et il est utilisé dans beaucoup d'applications populaires de bureau sur des PC et des Macs. Il peut aussi exister dans des applications mobiles et être incorporé dans beaucoup de dispositifs.

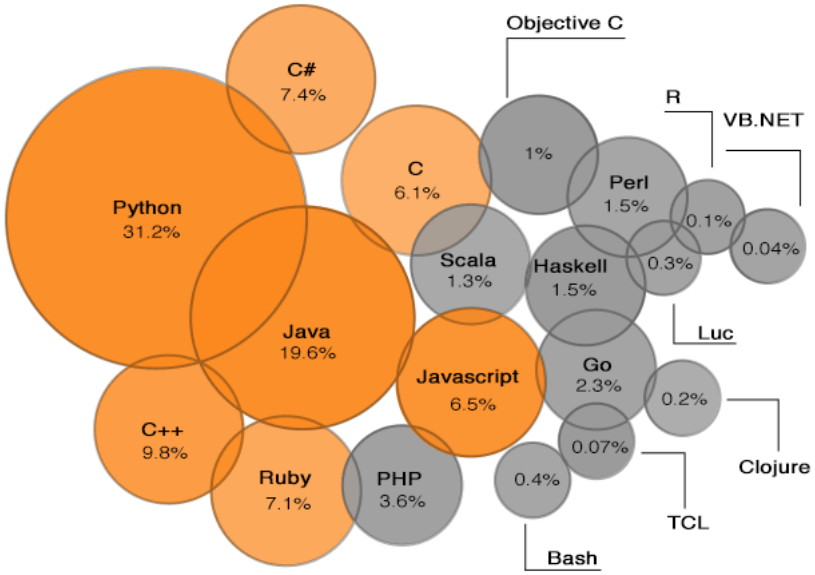


FIGURE 5.7 – Utilisation de Python par rapport aux autres langages.

5.4.3 Logiciel libre de conteneurisation, Docker

Il est impératif de considérer le type de déploiement en question avant de faire le choix de la plateforme de conteneurisation. Mais si on a l'intention de virtualiser un OS entier, ou d'exécuter une application virtuelle persistante pendant une longue période, LXD sera probablement une meilleure solution. Dans le cas où il y aurait un lancement rapide d'un grand nombre de conteneurs sur la base d'images d'applications génériques, il vaut mieux utiliser Docker ou CoreOS. Le deuxième facteur à considérer est l'environnement hôte. LXD ne supporte que Linux et il n'est documenté que pour Ubuntu. Donc, il y aurait certainement un problème lors de l'exécution de LXD dans le cas où une version de Windows ou une autre version de

Linux est utilisée. En revanche, Docker est assez portable sur presque tous les OS basés sur Linux, et maintenant exécutable sous Windows et OSX.

5.4.4 Orchestrateur des conteneurs, Kubernetes

Kubernetes est aujourd'hui un concurrent supérieur et (le plus) fort sur le marché. De plus, Kubernetes est techniquement supérieur aux autres orcherstrateurs des conteneurs. De meilleures dispositions de vérification de l'état de santé au niveau du pod, une meilleure planification et mise à l'échelle des conteneurs, des espaces de noms avec des contrôles d'accès basés sur les rôles et des stratégies réseaux pour contrôler le trafic entrant et sortant entre les services lui confèrent un avantage conséquent. Même les fournisseurs de cloud choisissent de passer aux services gérés de Kubernetes.

5.5 Connectivité

Les deux paradigmes de Cloud Computing et de Fog Computing ne fonctionnent que dans un environnement peuplé, doté d'une connectivité riche et à haut débit. Toutefois, les dispositifs IoT doivent être déployés dans des zones accessibles qui ont une connectivité Internet avec le monde extérieur. Le manque de connectivité empêche effectivement ces dispositifs IoT isolés d'accéder au ENs, FNs ou les passerelles aux Cloud. Cela nécessite un choix de connectivité selon le niveau et la nature de l'environnement : la capacité à connecter, la capacité maximale de connectivité des noeuds de l'architecture, la connectivité des dispositifs mobiles à la demande et la quantité maximale de données à transmettre.

L'architecture basée sur le Fog Manufacturing comporte de nombreuses couches intégrant des noeuds de natures différentes, impliquant forcément que les liens entre deux couches consécutives seront différents afin d'assurer la continuité de la connectivité.

Dans cette section, nous expliquons les différentes parties qui entrent en jeu dans notre architecture. Ces équipements sont : CPS fixes (machines, imprimantes 3D, découpes laser ou postes manuels) ou mobiles (AGVs), ENs : Points d'accès Wi-Fi, FNs : Cartes Raspberry Pi3, Passerelles : Routeurs, Cloud : serveur Big Data.

La figure 5.8 représente l'interconnexion entre les composantes de notre solution.

Les prochaines sous sections décriront les différents types de connectivités que l'on a créées afin de mettre en place notre architecture.

— **Liaison AGV - PAs :** On a supposé que les CPS fixes et mobiles seront liés en Wi-Fi au PA (point d'accès) à proximité, les robots mobiles sont connectés Wi-Fi aux points d'accès déjà installés sur tout l'atelier. Les robots envoient instantanément les données générées par les capteurs vers les points d'accès qui sont bien distribués sur l'atelier dans le but de garantir une large couverture de l'espace de présence des CPS,

— **Liaison PAs - LAN :** La liaison entre les points d'accès et les routeurs du LAN s'est établie filaire en utilisant les câbles Ethernet. Les points d'accès vont acheminer le trafic de données vers les routeurs du LAN. Ce trafic s'oriente par la suite vers le cluster de Raspberry pour le filtrage,

— **Liaison LAN - Cartes Raspberry :** Les cartes Raspberry sont connectées au réseau local à travers les câbles Ethernet. Les cartes Raspberry contrôlent le

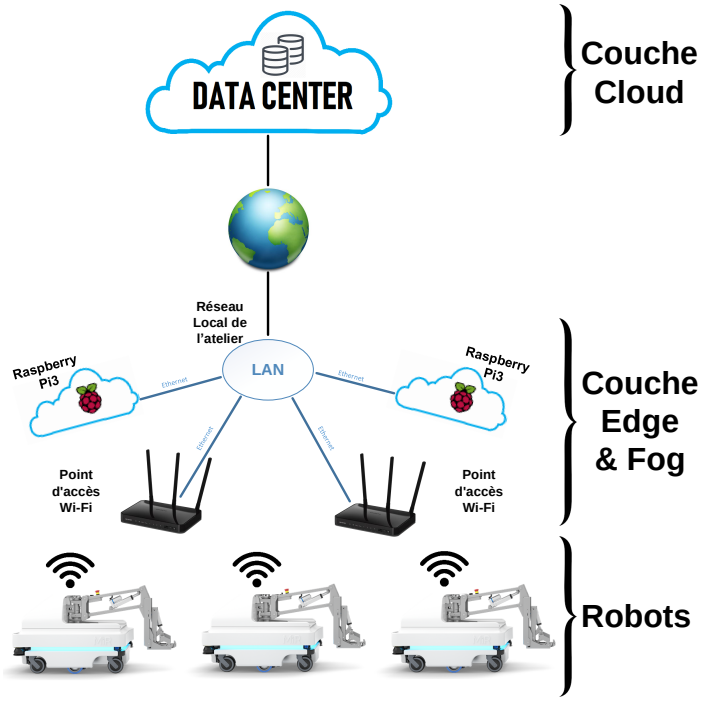


FIGURE 5.8 – Les liens entre les composants de l'architecture basée sur le Fog Manufacturing.

flux des données en entrée et puis transforment le format de données de XML en JSON,

— **Liaison LAN - Data center :** Après filtrage des données au niveau des cartes Raspberry, les données vont être orientées vers le Data Centre pour les stocker. Les routeurs du LAN acheminent le trafic à travers le passage par le réseau Internet,

A travers cette section, on a pu présenter la conception du système d'information au sein de l'atelier de production flexible. Dans cette conception, on a introduit la couche Fog et on a spécifié les différents moyens de communications existants dans l'architecture.

5.6 Réalisation

Aprés avoir identifié l'architecture, les outils technologiques ainsi que les différents types de connectivité, on passe à la réalisation de l'architecture basée sur le Fog Manufacturing. Au cours de cette section, on citera les différentes étapes de l'implémentation, on expliquera la démarche technique suivie et puis on citera les différentes difficultés rencontrées et les perspectives de cette réalisation.

5.6.1 Déroulement

Cette étude s'est élaborée en deux phases :

— La première phase est l'implémentation initiale sans l'intégration des liens avec les AGVs. Durant cette phase, l'objectif est de simuler l'architecture basée sur le Fog en générant des données aléatoires, ce qui permet de détecter les carences avant de rendre l'application plus complexe. Durant cette étape on s'assure que les liens sont correctement établis et que tous les noeuds sont

interconnectés. On teste également la couverture sur l'ensemble de l'atelier de production.
— La deuxième phase est celle de l'implémentation réelle et finale de l'architecture en impliquant l'intégralité des noeuds, y compris les AGVs. Cette étape sert à tester l'effet de la solution Fog Manufacturing sur une fonctionnalité de données. Le test est fait par le stockage des données filtrées par les FNs dans le data center du Cloud.

Première phase : sans intégration des robots

Dans un premier temps, on propose de tester le temps de transfert que mettent les données pour arriver des sources de données jusqu'aux noeuds Fog. C'est pour cela que nous simulons les robots par deux PCs qui génèrent des données aléatoires et les envoient périodiquement aux noeuds Fog. La figure 5.9 représente cette phase.

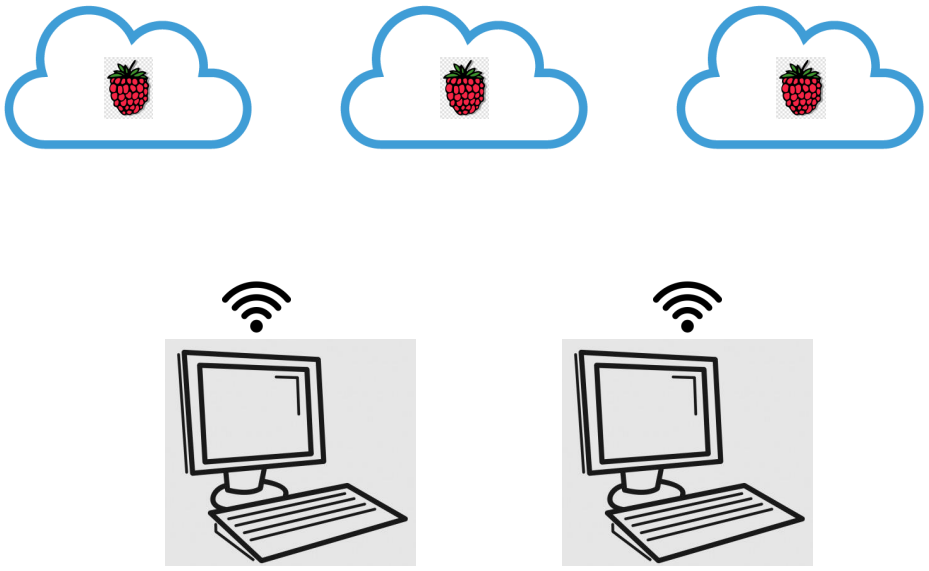


FIGURE 5.9 – Implémentation sans intégration des robots.

A travers cette première phase, on a pu estimer le temps de transfert moyen que mettent 1000 lignes de données générées et envoyées par les PCs connectés en WiFi jusqu'à leurs réceptions au niveau du cluster du couche Fog. Le temps de transfert, représenté sur la figure 5.10, est inférieur à 100 ms. Ce résultat est satisfaisant pour plusieurs applications nécessitant des traitements en temps réel.

Deuxième phase : avec intégration des robots

Concernant la deuxième phase, on intègre les robots et on commence la collecte des données réelles. Les robots envoient instantanément des balises XML, présentées par la figure 5.11. Elles sont décrites dans ce qui suit :
— "Status" : la tâche que le robot exécute.
— "Map" : la région dans laquelle le robot se trouve.
— "Current time" : la date de l'information sous le format universelle.
— "Battery" : le niveau de batterie.
— "Status number" : le label de la tâche exécutée.

```
############################  12  ################################
2019.07.11-09.39.09.492575 : Done sending
2019.07.11-09.39.09.547316 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 54  ms
############################  13  ################################
2019.07.11-09.39.10.694344 : Done sending
2019.07.11-09.39.10.767781 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 73  ms
############################  14  ################################
2019.07.11-09.39.11.886842 : Done sending
2019.07.11-09.39.11.964917 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 78  ms
############################  15  ################################
2019.07.11-09.39.13.087782 : Done sending
2019.07.11-09.39.13.144829 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 57  ms
############################  16  ################################
2019.07.11-09.39.14.282955 : Done sending
2019.07.11-09.39.14.351868 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 68  ms
############################  17  ################################
2019.07.11-09.39.15.493204 : Done sending
2019.07.11-09.39.15.568813 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 75  ms
############################  18  ################################
2019.07.11-09.39.16.715654 : Done sending
2019.07.11-09.39.16.784388 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 68  ms
############################  19  ################################
2019.07.11-09.39.17.928432 : Done sending
2019.07.11-09.39.17.998501 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 70  ms
############################  20  ################################
2019.07.11-09.39.19.161054 : Done sending
2019.07.11-09.39.19.231428 : Server has successfully received the Data
==========================> TRANSFER  TIME = 0  s, 70  ms
```

FIGURE 5.10 – Temps de Transfert des données.

— "Status time" : la durée dans laquelle le robot est dans cette tâche.
— "x,y" : la position (x,y) du robot.
— "Theta" : l'angle d'orientation que fait le robot.

```
############################  997  ################################
.....................Fog Server is processing.......................
<item><status>InTransit</status><map>8</map><current_time>1567522584.87</current_time><battery>
92.7</battery><statusnumber>5</statusnumber><status_time>1567522583.0</status_time><y>17.57</y>
<x>32.59</x><theta>81.59</theta></item>
############################  998  ################################
.....................Fog Server is processing.......................
<item><status>InTransit</status><map>8</map><current_time>1567522585.19</current_time><battery>
92.7</battery><statusnumber>5</statusnumber><status_time>1567522584.0</status_time><y>18.09</y>
<x>32.59</x><theta>98.57</theta></item>
```

FIGURE 5.11 – Données sous format XML.

Par la suite et au niveau des noeuds Fog, nous avons installé un module de contrôle de données pour éviter les lignes qui contiennent des valeurs nulles ou bien des valeurs erronées. Après le contrôle et le filtrage, les données sont transformées sous format JSON (figure 5.12) pour être acheminées vers le serveur du Data Centre.

MirDATA-2019.08.27-18.11.33.179638.jso ×

```
[
    {
        "current_time": 1566921954.45,
        "status": "InTransit",
        "map": 8,
        "x": 32.34,
        "y": 16.11,
        "theta": 107.03,
        "battery": 53.9,
        "statusnumber": 5,
        "status_time": 1566921952.0
    },
    {
        "current_time": 1566921954.77,
        "status": "InTransit",
        "map": 8,
        "x": 32.16,
        "y": 16.58,
        "theta": 113.63,
        "battery": 53.9,
        "statusnumber": 5,
        "status_time": 1566921954.0
    },
```

FIGURE 5.12 – Données sous format JSON.

5.6.2 Test et résultats de l'implémentation technique

Les deux phases identifiées dans la section précédente vont être implémentées tout d'abord sur une machine virtuelle par la suite transférée sous la forme d'une image conteneur vers le cluster.

Développement & Test de l'application sur une Machine Virtuelle

En utilisant la programmation Socket, on a développé une application composée d'une partie serveur et d'une autre client. D'un côté, on genère des données aléatoires et on les envoie. De l'autre côté, on réceptionne ces données et on calcule le temps de transfert. La figure 5.13 montre l'application qui tourne en machine virtuelle.

Création du conteneur

Dans le but de faciliter le déploiement de l'application, on a conteneurisé la partie Fog de l'application afin de la mettre en marche sur le cluster des cartes RPi.

Tout d'abord, on a créé le Dockerfile, représenté sur la figure 5.14, dans lequel on a précisé :

— "FROM python :3.7-alpine" : il s'agit de l'image de base utilisée.
— "Application source code Directory" : création d'un répertoire pour le code source de l'application.
— "Home directory for containers" : Changement du repertoire courant.
— "Source code" : Copier le code source dans le conteneur.
— "Ports" : Exposer le port 5000.
— "Persistent data" : Partager un dossier de log.
— "CMD ["python","server.py"]" : Faire tourner le code source sous python.

Ensuite, on a construit l'image Docker, sur la figure 5.15, image qui est ensuite exécutée pour la tester, puis chargée sur DockerHub.

Le Docker Hub est une plateforme similaire au GitHub mais elle est dédiée aux images Docker seulement. Cette plateforme, représentée en figure 5.16, permet d'uploader, télécharger et mettre à jour les images Docker utilisées dans la preuve de concept de ce chapitre.

Déploiement de l'image dans le cluster

La figure 5.17 présente le cluster, sachant que nous avons utilisé trois cartes Raspberry Pi3, les deux niveaux qui se trouvent en bas sont des "travailleurs" et la carte en haut est le manager du cluster.

On se connecte sur les cartes Raspberry en SSH, on initialise le Kubernetes cluster sur la carte Manager, puis on intègre les autres cartes en tant que Workers.

En initialisant le Kubernetes Cluter, le manager génère un Token, représenté dans la figure 5.18, qui permet aux travailleurs de rejoindre le cluster.

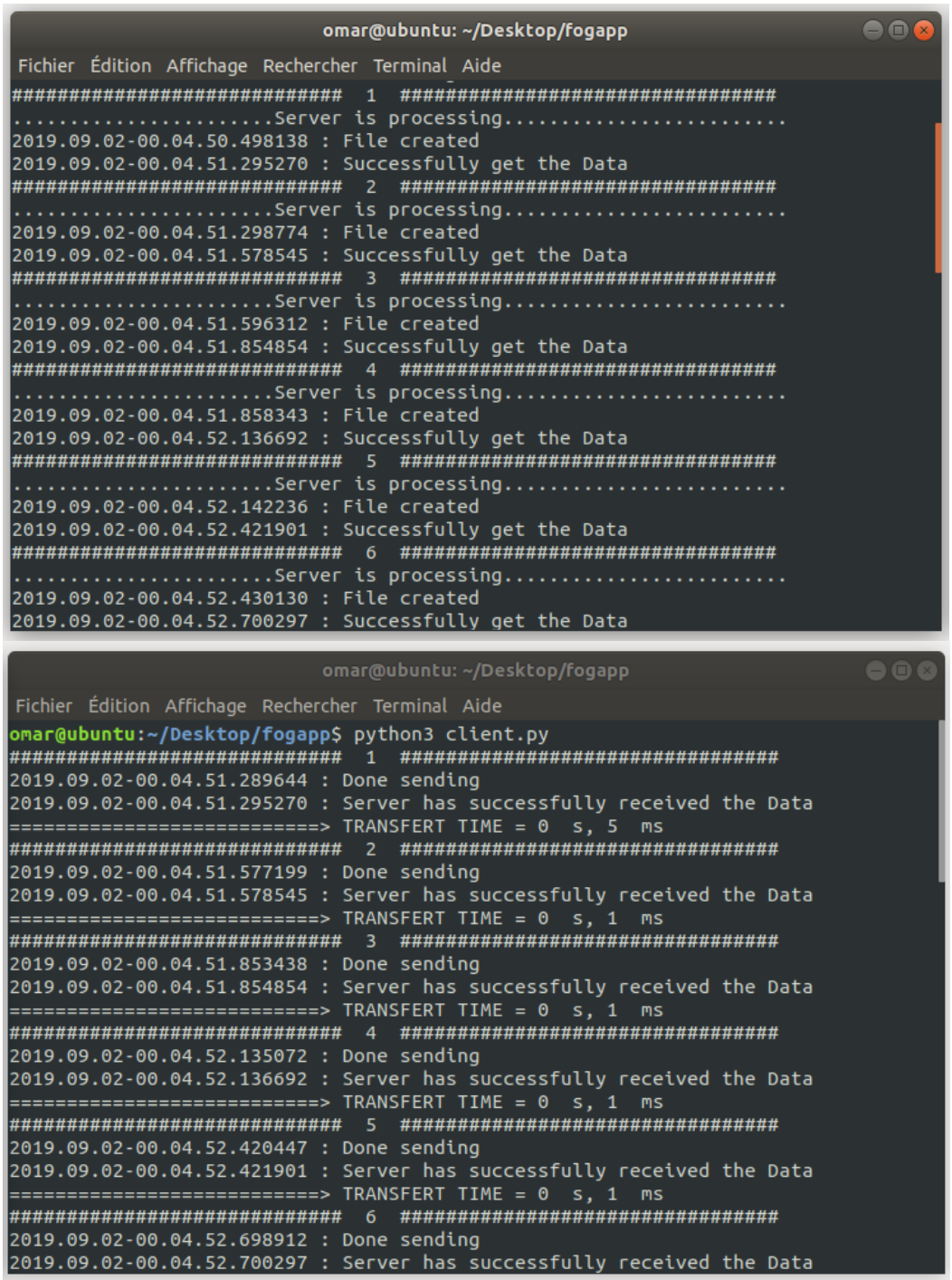


FIGURE 5.13 – Application sous machine virtuelle.

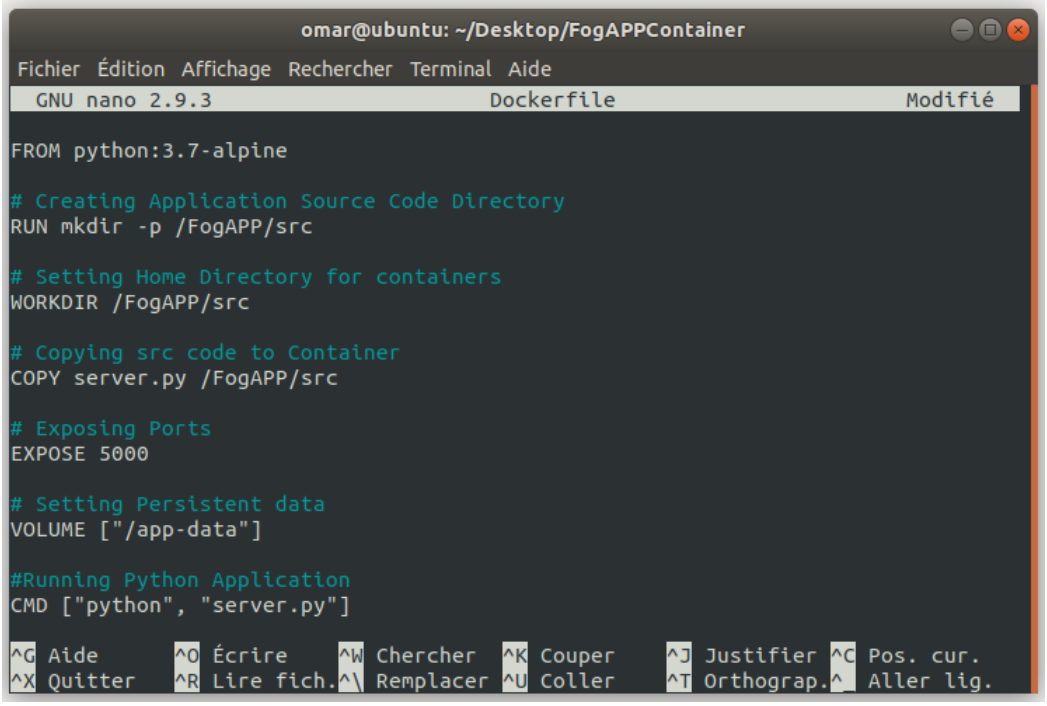


FIGURE 5.14 – Docker File.

La figure 5.19 montre le statut de chacune des instances du cluster. Après l'initialisation du cluster au niveau du Manager et l'intégration des deux autres cartes en tant que Worker, les instances du cluster demeurent maintenant prêtes et disponibles pour l'exécution.

Par conséquent, on a lancé le tableau de bord Kubernetes afin de pouvoir déployer l'image sur le cluster déjà prête. Comme le présente la figure 5.20, on réalise le déploiement en faisant appel à l'image du conteneur, en indiquant le nombre de pods et en choisissant la nature du service et le port désigné.

Une fois déployés (figure 5.21), les noeuds travailleurs commencent à exécuter l'image Docker.

5.6.3 Difficultés techniques rencontrées

On a rencontré de nombreuses difficultés tout au long de cette partie aussi bien du point de vue communication que de point du vue traitement de données.

Communication entre les noeuds de l'architecture

Au début de l'étude, la difficulté majeure rencontrée était de faire transmettre les données (à partir des robots) au cluster Kubernetes. La communication via les Sockets mises en place comporte deux parties ; la première est celle du côté robot dans laquelle on a adressé la communication vers l'IP du noeud Manager et on a utilisé le port de communication 5000 ; la deuxième est celle du Fog qui écoute sur le port 5000. Cependant, la connexion n'était pas établie et une erreur de "Connection Refused" affichée. Cette erreur, selon la documentation, est généralement due au blocage du port ou à son utilisation par un autre programme. En examinant le fichier

```
pi@rpiManager:~/fogcontainer $ sh build.sh
Sending build context to Docker daemon   7.68kB
Step 1/8 : FROM python:3.7-alpine
 ---> 479a172d0d26
Step 2/8 : RUN mkdir -p /FogAPP/src
 ---> Using cache
 ---> f3a4d14ec2fc
Step 3/8 : WORKDIR /FogAPP/src
 ---> Using cache
 ---> a8745ae7961f
Step 4/8 : COPY fogserver.py /FogAPP/src
 ---> Using cache
 ---> 5dc00bf4d49d
Step 5/8 : ENV APP_ENV development
 ---> Using cache
 ---> 5d9a093fead8
Step 6/8 : EXPOSE 5000
 ---> Using cache
 ---> 204df4dcf954
Step 7/8 : VOLUME ["/app-data"]
 ---> Using cache
 ---> c3b8374654a5
Step 8/8 : CMD ["python", "fogserver.py"]
 ---> Using cache
 ---> e902f8152a5f
Successfully built e902f8152a5f
Successfully tagged fogapp:latest
pi@rpiManager:~/fogcontainer $ sh run.sh
.................Hi Omar! I am waiting for tasks...................
```

FIGURE 5.15 – Building de l'image Docker.

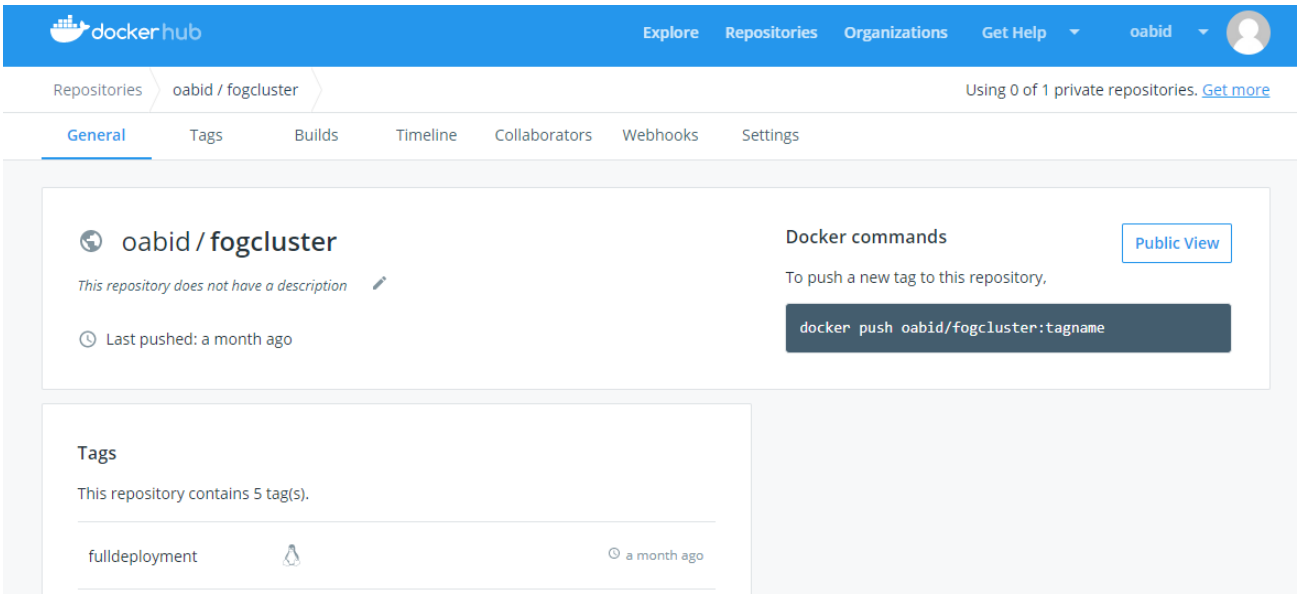


FIGURE 5.16 – Docker Hub.

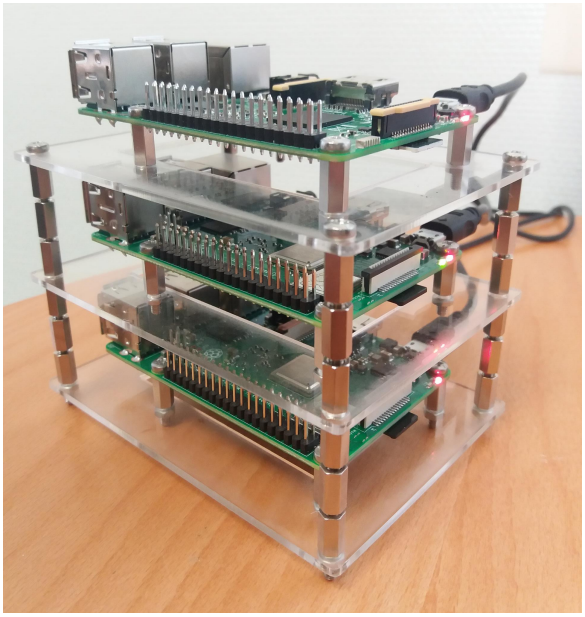


FIGURE 5.17 – Cluster de Raspberry.

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.191.76.73:6443 --token ga0hj6.gewltr13jm02mxtw \
    --discovery-token-ca-cert-hash sha256:0ae9462167ccbe06b488257802c652b3faa298fb91e1de72eb281d80833f9260
pi@rpiManager:~ $
```

FIGURE 5.18 – Initialisation du Cluster.

```
pi@rpiManager:~ $ kubectl get nodes
NAME          STATUS   ROLES    AGE   VERSION
rpimanager    Ready    master   77m   v1.15.0
rpiworker1    Ready    worker   49m   v1.15.0
rpiworker2    Ready    worker   33m   v1.15.0
pi@rpiManager:~ $
```

FIGURE 5.19 – Distribution des noeuds.

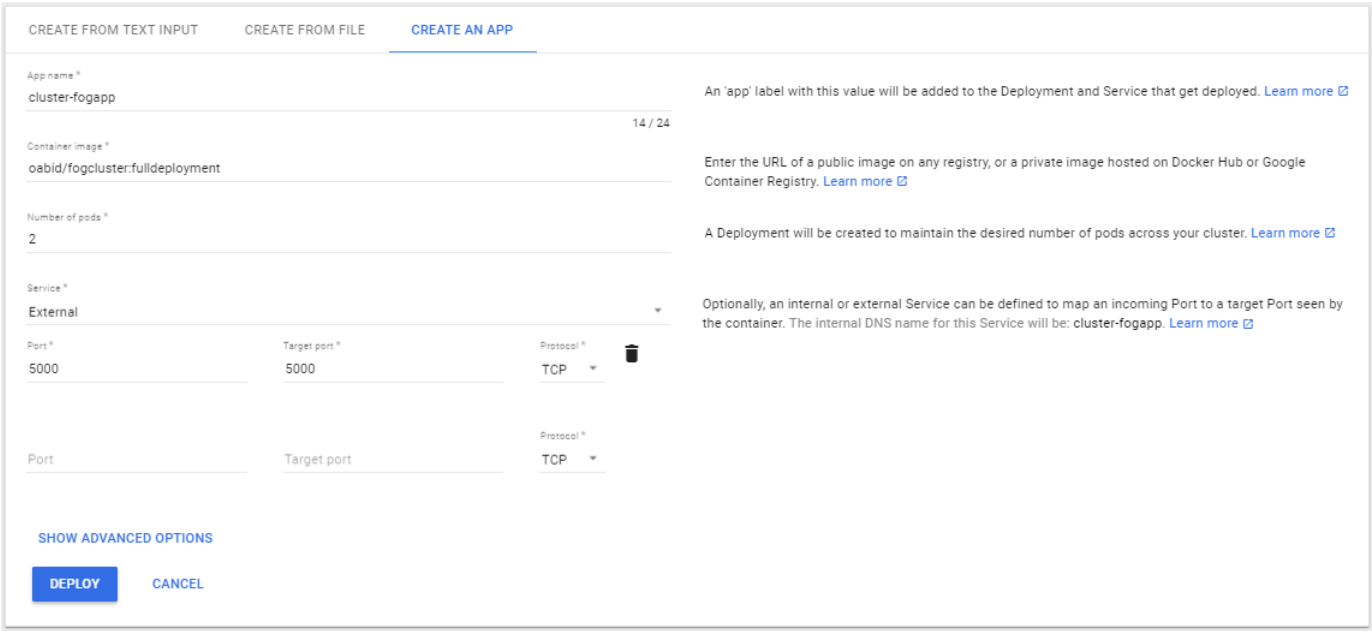


FIGURE 5.20 – Déploiement de l'image Docker.

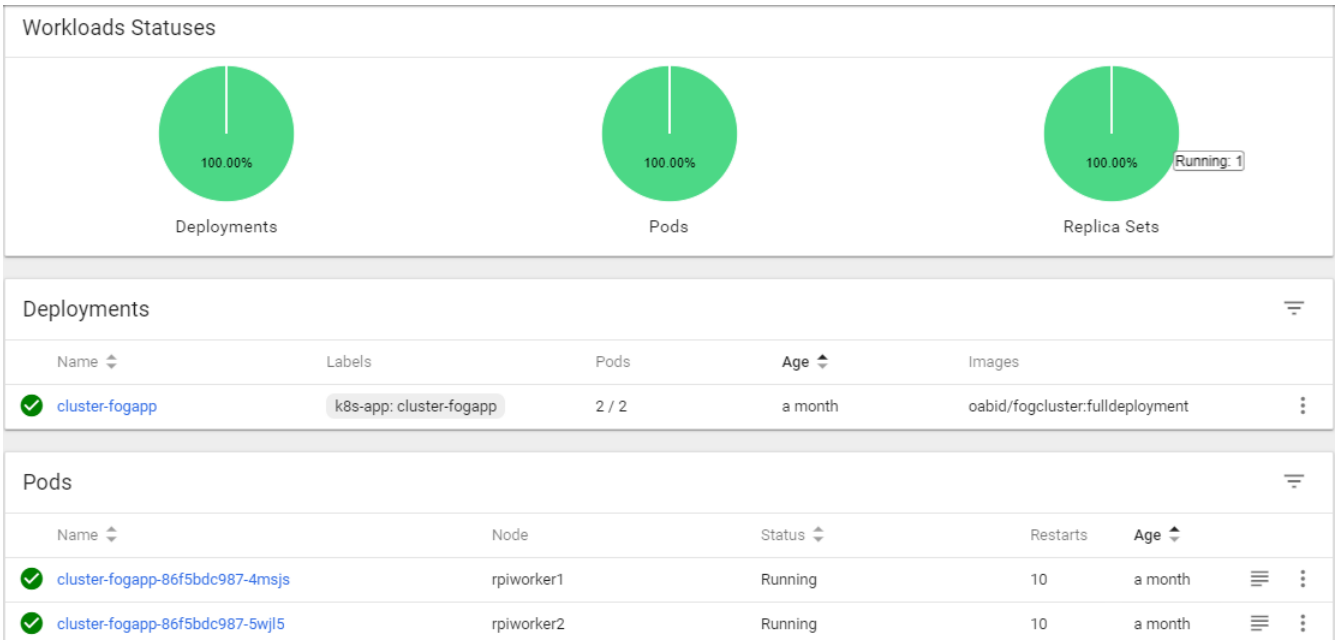


FIGURE 5.21 – Kubernetes Dashboard.

de configuration de Kubernetes, on a trouvé un autre port attribué dynamiquement par l'orchestrateur Kubernetes, représenté sur la figure 5.22, appelé nodePort.

En revenant à la documentation de Kubernetes, on a compris que l'orchestrateur Kubernetes alloue automatiquement un port dans l'intervalle [30000-32767], ce port expose les services à l'intérieur et à l'extérieur du cluster . Nous avons pallié ce problème en utilisant le nodePort alloué par l'orchestrateur Kubernetes au niveau du robot, c'est à dire que le robot envoie les données sur le port désigné par le cluster.

Traitement temps réel

Nous avons rencontré un problème de communication simultanée puisque les deux robots envoient les données au cluster en même temps. Cependant, le cluster ne traite que les données venant du premier robot et il semble indisponible pour le deuxième robot. Ce problème était important parce qu'il ne satisfait pas nos objectifs. On a attendu d'avoir un cluster de haute disponibilité, capable de traiter un flux de données venant de différents robots simultanément.

On a résolu ce problème à l'aide de la programmation parallèle en utilisant les threads, appelée Multithreading, ce qui revient à exécuter plusieurs threads ensemble. En utilisant le Multithreading au niveau des noeuds du cluster, ce dernier demeure capable d'écouter plusieurs robots (qui envoient simultanément leurs données).

5.6.4 Perspectives

Nous sommes arrivés aujourd'hui à avoir une architecture basée sur le Fog Manufacturing qui couvre les CPS d'un atelier d'Industrie 4.0. Les FNs assurent le filtrage des données et leur stockage permanent au niveau de la couche des serveurs Cloud. On a atteint plus de 30000 lignes de données historiques stockées.

```
{
  "kind": "Service",
  "apiVersion": "v1",
  "metadata": {
    "name": "cluster-fogapp",
    "namespace": "default",
    "selfLink": "/api/v1/namespaces/default/services/cluster-fogapp",
    "uid": "1ed19e21-177a-4121-8412-ce63a20bf003",
    "resourceVersion": "845318",
    "creationTimestamp": "2019-08-06T14:20:15Z",
    "labels": {
      "k8s-app": "cluster-fogapp"
    }
  },
  "spec": {
    "ports": [
      {
        "name": "tcp-5000-5000-q22d8",
        "protocol": "TCP",
        "port": 5000,
        "targetPort": 5000,
        "nodePort": 31984
      }
    ],
    "selector": {
      "k8s-app": "cluster-fogapp"
    },
    "clusterIP": "10.98.89.8",
    "type": "NodePort",
    "sessionAffinity": "None",
    "externalTrafficPolicy": "Cluster"
  },
  "status": {
    "loadBalancer": {}
  }
}
```


FIGURE 5.22 – Fichier de configuration Kubernetes.

Comme perspectives, on peut profiter de ce grand volume de données pour mettre en place un modèle d'apprentissage automatique. Nous envisageons d'implémenter un système d'alerte pour les robots en fonction de leurs positions. C'est-à-dire que si le robot est dans une localisation erronée, le système l'avertit et il lui donne la position dans laquelle il devrait être.

Les données stockées dans le Cloud vont être subdivisées en trois parties :

— Training set : ceux sont les données d'entraînement sur lesquelles le modèle apprend.
— Validation set : les données de validation sont utilisées pour évaluer le modèle, mais ceci est pour une évaluation fréquente. Ces données servent à rectifier les hyperparamètres du modèle.
— Test set : Les données du test mesurent l'erreur du modèle final sur des données qu'il n'a jamais vues.

Le modèle s'entraîne afin de prédire la position $(\hat{X},\hat{Y})$ du robot. Après le training, la validation et le test du modèle au niveau du Cloud, on met en place le modèle au niveau de la couche Fog. Ensuite, le modèle reçoit en temps réel les informations du robot et il retourne les valeurs prédites de $(\hat{X},\hat{Y})$. A partir d'un seuil d'erreur entre $(\hat{X},\hat{Y})$ et (X,Y), la couche Fog avertit le robot et elle lui demande d'aller vers la position appropriée.

5.6.5 Conclusion

L'architecture basée sur le Fog Manufacturing a montré un grand potentiel en terme de mise en oeuvre de solutions distribuées pour le contrôle et la supervision d'un atelier de production flexible. En outre, sa mise en oeuvre réelle est actuellement possible, grâce à la disponibilité du matériel, des logiciels et de technologies comme la carte Raspberry, Docker et Kubernetes.

En outre, cette simulation a démontré par une mise en oeuvre réelle que l'utilisation du Fog Computing peut répondre à deux grands défis de l'industrie 4.0 qui sont l'interopérabilité de différents systèmes et la scalabilité du système d'informations. Grâce à l'intégration de la technologie Docker et à la haute performance de l'orchestrateur Kubernetes dans l'architecture Fog Computing, nous avons réalisé un prototype évolutif et interopérable. L'ajout de nouveaux noeuds Fog, des managers ou des robots utilisant différents systèmes d'exploitation devient plus facile à gérer. Par exemple, dans notre cas, nous avons ajouté un noeud Manager supplémentaire dans notre cluster afin d'assurer la redondance et la tolérance aux pannes.

Chapitre 6

Conclusion générale

Cette thèse s'inscrit dans le cadre du traitement distribué en temps réel dans les systèmes cyber-physiques industriels et plus particulièrement le déploiement du Fog computing dans le contexte de l'Industrie 4.0 (Appelé Fog Manufacturing). Ces travaux de thèse proposent deux aboutissements, le premier, théorique, qui consiste en l'identification des défis limitant le déploiement du Fog dans le contexte de l'Industrie 4.0 et le second, plus pratique, propose un modèle d'optimisation du coût et son implémentation sur un cas réel. Le déploiement du Fog Manufacturing vise à distribuer la collecte puis le traitement des données des CPS dans l'industrie du futur pour des fins de logistique, de traçabilité, de surveillance, de maintenance, etc. Pour accomplir ces objectifs, les CPS sont équipés par des capteurs qui génèrent périodiquement les données qui seront envoyées par la suite aux noeuds de l'architecture basée sur le Fog Computing (ENs, FNs et passerelles), qui sont déployés uniformément dans l'usine. Ce déploiement va satisfaire de nombreux défis tout en optimisant le coût d'installation.

Via un état de l'art, le chapitre 2 retrace le concept de l'Industrie 4.0 comme étant la nouvelle révolution industrielle basée sur de nombreux composants (IoT, IoS, CPS, Cloud computing, Big Data, RA, etc.), ainsi que les premiers travaux relatifs à ce concept, à ses intégrations et exigences. Ensuite, une étude sur la solution du Fog computing est réalisée en étudiant ses caractéristiques, son architecture, les travaux intégrant le Fog computing et en réalisant une comparaison avec le Cloud computing. Ce chapitre introduit finalement notre problématique générale dans le contexte de l'Industrie 4.0, qui consiste en l'intégration du Fog computing afin d'améliorer les performances du Cloud computing, tout en traitant localement et d'un façon distribuée les données. Le Fog computing dans ce contexte ne va pas supprimer les services du Cloud computing, mais il vient étendre les services du Cloud au bord du réseau.

Dans la première partie du chapitre 3, nous avons étudié les différentes applications qui intègrent le Fog computing pour améliorer les performances de l'Industrie 4.0. Une telle vision du Fog est appelée le "Fog Manufacturing". Nous avons constaté que son déploiement dans les industries demeure modeste et qu'il n'existe pas une exploitation complète des caractéristiques du Fog computing. Cette emploi modéré est dû principalement à deux raisons majeures : d'une part l'apparition récente des deux paradigmes de l'Industrie 4.0 (2011) et du Fog computing (2015) et d'autre part la présence de nombreux défis qui limitent une telle intégration.

Notre première contribution a consisté à identifier les défis du déploiement du Fog computing dans le contexte de l'Industrie 4.0. Ainsi, dans la deuxième partie du chapitre 3, face à ces différents constats, nous avons donc spécifié les défis Input suivants :

— **Programmabilité :** Comment choisir les FNs programmables afin d'assurer une bonne performance en termes de calcul et de stockage local ?

— **Sécurité :** Comment assurer une communication sécurisée et confidentielle des données entre les FNs et les différents composants de l'architecture ?
— **Hétérogénéité :** Comment faire varier les différents SEs dans l'environnement de production sans affaiblir le fonctionnement du Fog Manufacturing ?
— **Intéropérabilité :** Comment assurer une commnunication entre les différents CPS lors d'un envoi des données des terminaux aux FNs ?

Les défis Outputs sont les suivants :

— **QoS :** Comment transmettre efficacement les données dans l'architecture du Fog Manufacturing ? Le terme efficacité se réfère à des contraintes de fiabilité, de délai, de taux d'erreur, etc.
— **Énergie** Comment assurer une faible consommation d'énergie des différents composants de l'architecture basée sur le Fog Manufacturing ?
— **Coût :** Finalement, comment garder le coût du déploiement du Fog Manufacturing faible tout en surmontant tous les autres défis mentionnés ci-dessus.

Cette identification des différents défis est notre première contribution scientifique. Elle nous a conduit à l'étude des liens entre les différents défis, étude réalisée en se basant sur les indicateurs de performance. Il ressort de cette étude que le fait de surmonter un des défis Input permet de surmonter ses défis connexes et parmi eux les défis Output. Finalement, nous avons cité les applications industrielles qui possèdent des caractéristiques nécessitant l'intégration du Fog Manufacturing pour distribuer le traitement en temps réel des données massives provenant des CPS. Ce croisement des besoins et des caractéristiques nous a permis de justifier notre choix de l'application de logistique.

La problématique scientifique et technique pointue ensuite traitée, a porté sur l'optimisation du coût du déploiement du Fog Manufacturing afin de faciliter son implémentation sur un cas réel. Pour répondre à ces interrogations concernant le coût et la QoS, nous avons proposé dans le chapitre 4 un modèle d'optimisation du coût du déploiement du Fog Manufacturing dans une application de logistique interne à l'usine. La première partie a présenté un état de l'art sur le déploiement du Fog computing dans la logistique, et par la suite nous avons présenté une étude sur la problématique de déploiement du Fog Manufacturing dans ce contexte.

La deuxième contribution associée au chapitre 4 est la proposition d'un modèle d'optimisation du coût du déploiement du Fog manufacturing dans une application de logistique interne. Nous avons contribué à l'amélioration du modèle d'optimisation en assurant la linéarisation des contraintes dans un premier temps. Cette étape a amélioré le modèle en termes de convergence et de temps de résolution. Par la suite, nous avons proposé l'introduction de deux nouvelles contraintes qui sont :

— La probabilité de la présence des AGVs sur une trajectoire définie pour le déplacement. Cela entraîne la modification de la concentration en ENs aux alentours des trajectoires. La simulation du modèle en intégrant cette contrainte a optimisé le coût et le nombre des noeuds activés.
— Le choix du lien filaire entre les différents noeuds de l'architecture (RJ45 ou fibre) selon la capacité de demande des noeuds de la couche inférieure.

Le modèle fut résolu à l'aide de la méthode exacte et du solveur Fico xpress. Malgré la difficulté et la complexité du problème, nous avons obtenu des premiers résultats prometteurs mais cela reste une piste intéressante à améliorer et développer.

Nous avons évalué le résultat du modèle d'optimisation du coût par une simulation dans le chapitre 5. Nous avons déployé une véritable plate-forme de Fog Computing pour pré-traiter les données sur un CPS de production. Notre plate-forme est également basée sur la technologie Docker qui permet de traiter les données d'une

manière conteneurisée. Par cette étude, nous avons prouvé l'efficacité de notre plateforme en termes de temps de communication (latence), de scalabilité et d'interopérabilité. Quant aux FNs, ils ont montré aussi de bonnes performances pour le filtrage des données.

L'objectif initial de cette thèse était de prouver l'applicabilité du Fog Manufacturing malgré les nombreux défis que cela met en jeu, ainsi qu'optimiser le déploiement du Fog Manufacturing dans la logistique, sans oublier le fait d'appliquer réellement le Fog Manufacturing, en se basant sur une architecture distribuée. Ces résultats théoriques ont été publiés dans quatre conférences internationales. Les résultats sur le modèle d'optimisation du coût sont actuellement soumis dans une revue internationale. La suite de nos travaux va donc consister à améliorer la phase d'ingénierie en utilisant des FNs performant afin de tester de nombreux types de traitement de données dans des situations applicatives réelles. Nous pensons que le Fog computing va continuer à grandir tout en répondant aux exigences des technologies émergentes. Parallèlement, le Cloud computing continuera à fournir des services et à répondre aux demandes d'accès au Cloud computing.

Les perspectives sont donc à court terme :

— Compléter le travail sur la contrainte du choix du lien filaire (fibre ou RJ45). Il faut par la suite passer à l'utilisation de méthodes méta-heuristiques, comme les algorithmes génétiques, afin de traiter le problème de multiplier à de nombreuses reprises le nombre d'instances, comme présenté dans le chapitre 4. Une comparaison entre la méthode exacte et les algorithmes génétiques peut se faire. Par la suite nous pensons à alléger le calcul, afin d'accélérer la convergence du modèle même avec une grande instance, en combinant les contraintes.

— Enrichir le modèle en considérant des terminaux hétérogènes présents dans l'atelier de production, et en diversifiant leurs paramètres fixés dans le cadre de ce travail. Cela permettrait de personnaliser le modèle selon les demandes de chaque type de CPS. Cette étape vise à améliorer les résultats d'optimisation.

À plus long terme, plusieurs perspectives s'offrent et constituent des perspectives très intéressantes :

— Faire évoluer la modélisation des défis. Les défis identifiés au niveau du chapitre 3 sont présentés en fonction de leurs PIs, mais il serait très intéressant d'introduire les équations séquentielles afin de modéliser ces défis limitant l'intégration du Fog Manufacturing,

— Examiner d'autre défis à surmonter au delà du coût et de la QoS. Nous pourrions par exemple travailler sur un modèle multi-objectifs qui intègre comme deuxième objectif la consommation d'énergie des noeuds de l'architecture basée sur le Fog manufacturing. Cette optimisation peut améliorer le modèle et faciliter l'intégration du Fog.

— Envisager d'autres applications possibles pour appliquer le modèle. Les contraintes peuvent être adaptées aux composants de l'atelier de production. Par exemple, appliquer le Fog Manufacturing pour la supervision des robots collaboratifs dans un atelier de production.

— Améliorer l'efficacité de l'applicabilité de l'architecture basée sur le Fog Manufacturing en déployant des serveurs Fog (par exemple des routeurs industriel Cisco) et avec un grand nombre de noeuds (à grande échelle),

— Étendre le déploiement du Fog Manufacturing pour de nouvelles fonctionnalités (calcul des modèles légers, traitement selon la priorité, prise de décision

en temps réel). On pourrait par exemple imaginer d'utiliser le Fog Manufacturing pour assurer la prise des décisions urgentes selon les priorités de la source.

— Introduire les concepts de l'Industrie 5.0. Ce terme industrie 5.0, centré sur l'humain, a été introduit récemment. Il est en lien avec l'articulation et les interactions entre humains, CPS, machines intelligentes, robots, tirant avantages de technologies avancées et innovantes comme l'intelligence artificielle, la science des données (incluant acquisitions et traitements efficients et rapides), les interfaces humains -machines avancées, ... et, potentiellement, le Fog Manufacturing qui pourrait induire une valeur ajoutée non négligeable !

Bibliographie

URL : http://www.rfidglobal.eu/userfiles/documents/CASAGRAS26022009.pdf.

AAZAM, Mohammad et Eui-Nam HUH (2014). « Fog computing and smart gateway based communication for cloud of things ». In : *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*. IEEE, p. 464-470.

— (2015). « Dynamic resource provisioning through Fog micro datacenter ». In : *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE, p. 105-110.

AHMED, Arif et Guillaume PIERRE (2018). « Docker container deployment in fog computing infrastructures ». In : *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE, p. 1-8.

AHMED, Ejaz et Mubashir Husain REHMANI (2017). *Mobile edge computing : opportunities, solutions, and challenges*.

AI, Yuan, Mugen PENG et Kecheng ZHANG (2018). « Edge computing technologies for Internet of Things : a primer ». In : *Digital Communications and Networks* 4.2, p. 77-86.

AL-FUQAHA, Ala et al. (2015). « Internet of things : A survey on enabling technologies, protocols, and applications ». In : *IEEE Communications Surveys & Tutorials* 17.4, p. 2347-2376.

AL YAMI, Mohammed et Dirk SCHAEFER (2019). « Fog Computing as a Complementary Approach to Cloud Computing ». In : *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE, p. 1-5.

ALHAZMI, Omar H et Khalid S ALOUFI (2019). « Fog-based internet of things : a security scheme ». In : *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, p. 1-6.

ALMADA-LOBO, Francisco (2016). « The Industry 4.0 revolution and the future of manufacturing execution systems (MES) ». In : *Journal of Innovation Management* 3.4, p. 16-21.

ANDROCEC, Darko, Neven VRCEK et Jurica SEVA (2012). « Cloud computing ontologies : A systematic review ». In : *Proceedings of the third international conference on models and ontology-based design of protocols, architectures and services*, p. 9-14.

ASHTON, Kevin (2011). « That 'internet of things' thing ». In : *RFiD Journal* 22.7.

ATZORI, Luigi, Antonio IERA et Giacomo MORABITO (2010). « The internet of things : A survey ». In : *Computer networks* 54.15, p. 2787-2805.

BACCARELLI, Enzo et al. (2017). « Fog of everything : Energy-efficient networked computing architectures, research challenges, and a case study ». In : *IEEE access* 5, p. 9882-9910.

BACIS, Enrico et al. (2015). « DockerPolicyModules : mandatory access control for docker containers ». In : *2015 IEEE Conference on Communications and Network Security (CNS)*. IEEE, p. 749-750.

BAHTOVSKI, Aleksandar et Marjan GUSEV (2014). « Cloudlet challenges ». In : *Procedia Engineering* 69, p. 704-711.

BANERJEE, Ayan et al. (2012). « Ensuring safety, security, and sustainability of mission-critical cyber–physical systems ». In : *Proceedings of the IEEE* 100.1, p. 283-299.

BAR-MAGEN NUMHAUSER, Jonathan et Jose Antonio Gutierrez de MESA (2013). « XMPP Distributed Topology as a Potential Solution for Fog Computing ». In : *MESH 2013 The Sixth International Conference on Advances in Mesh Networks*.

BELKADI, Farouk et al. (2017). « Collaboration management framework for OEM–suppliers relationships : a trust-based conceptual approach ». In : *Enterprise Information Systems* 11.7, p. 1018-1042.

BENATIA, Mohamed Amin et al. (2018). « QR-code enabled product traceability system : A big data perspective ». In : *Advances in transdisciplinary engineering*, p. 323-328.

BERNSTEIN, David (2014). « Containers and cloud : From lxc to docker to kubernetes ». In : *IEEE Cloud Computing* 1.3, p. 81-84.

BETTAYEB, B, SJ BASSETTO et M SAHNOUN (2014). « Quality control planning to prevent excessive scrap production ». In : *Journal of Manufacturing Systems* 33.3, p. 400-411.

BILL, Lydon (2016). « ORGIndustry 4.0 : Intelligent and flexible productionDigitization ». In :

BLANCO-NOVOA, Óscar et al. (2018). « A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard ». In : *IEEE Access* 6, p. 8201-8218.

BONOMI, Flavio et al. (2012). « Fog computing and its role in the internet of things ». In : *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, p. 13-16.

BONOMI, Flavio et al. (2014). « Fog computing : A platform for internet of things and analytics ». In : *Big Data and Internet of Things : A Roadmap for Smart Environments*. Springer, p. 169-186.

BOUZARKOUNA, Imen et al. (2018). « Challenges facing the industrial implementation of Fog Computing ». In : *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, p. 341-348.

BOYES, Hugh et al. (2018). « The industrial internet of things (IIoT) : An analysis framework ». In : *Computers in Industry* 101, p. 1-12.

BRIK, Bouziane et al. (2019). « Towards Predicting System Disruption in Industry 4.0 : Machine Learning-Based Approach ». In : *Procedia Computer Science* 151, p. 667-674.

BRITO, Mathias Santos de et al. (2018). « Application of the fog computing paradigm to smart factories and cyber-physical systems ». In : *Transactions on Emerging Telecommunications Technologies* 29.4, e3184.

BROGI, Antonio et Stefano FORTI (2017). « QoS-aware deployment of IoT applications through the fog ». In : *IEEE Internet of Things Journal* 4.5, p. 1185-1192.

BRUNER, Jon (2013). *Industrial Internet*. " O'Reilly Media, Inc."

BUI, Thanh (2015). « Analysis of docker security ». In : *arXiv preprint arXiv :1501.02967*.

BUYUKKOC, Cagatay (2016). « Edge definition and how it fits with 5g era networks ». In : *IEEE Software Defined Networks, Newsletter.[Online]. Available : http ://sdn. ieee. org/newsletter/march-2016/edge-definition-and-how-it-fits-with-5g-era-networks*.

BUYYA, Rajkumar, Rodrigo N CALHEIROS et Xiaorong LI (2012). « Autonomic cloud computing : Open challenges and architectural elements ». In : *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*. IEEE, p. 3-10.

CHANG, Chii, Satish Narayana SRIRAMA et Rajkumar BUYYA (2017). « Indie Fog : An Efficient Fog-Computing Infrastructure for the Internet of Things ». In : *Computer* 50.9, p. 92-98.

CHEKIRED, Djabir Abdeldjalil, Lyes KHOUKHI et Hussein T MOUFTAH (2018). « Industrial IoT data scheduling based on hierarchical fog computing : a key for enabling smart factory ». In : *IEEE Transactions on Industrial Informatics* 14.10, p. 4590-4602.

CHEN, Min, Shiwen MAO et Yunhao LIU (2014). « Big data : A survey ». In : *Mobile Networks and Applications* 19.2, p. 171-209.

CHEN, Rui-Yang (2017). « An intelligent value stream-based approach to collaboration of food traceability cyber physical system by fog computing ». In : *Food Control* 71, p. 124-136.

CHEN, Siguang et al. (2019). « Delay guaranteed energy-efficient computation offloading for industrial IoT in fog computing ». In : *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, p. 1-6.

CHEN, Yan, Yu WANG et Daofu GONG (2019). « Fog computing support scheme based on fusion of location service and privacy preservation for QoS enhancement ». In : *Peer-to-Peer Networking and Applications* 12.6, p. 1480-1488.

CHIANG, Mung et Tao ZHANG (2016). « Fog and IoT : An overview of research opportunities ». In : *IEEE Internet of Things Journal* 3.6, p. 854-864.

CHIARELLO, Filippo et al. (2018). « Extracting and mapping industry 4.0 technologies using wikipedia ». In : *Computers in Industry* 100, p. 244-257.

ÇIFLIKLI, Cebrail et Esra KAHYA-ÖZYIRMIDOKUZ (2010). « Implementing a data mining solution for enhancing carpet manufacturing productivity ». In : *Knowledge-Based Systems* 23.8, p. 783-788.

COLOMBO, Armando W et al. (2017). « Industrial Cyberphysical Systems : A Backbone of the Fourth Industrial Revolution ». In : *IEEE Industrial Electronics Magazine* 11.1, p. 6-16.

COMPUTING, Fog (2016). *the Internet of Things : Extend the Cloud to Where the Things Are*.

DANTU, Karthik, Steven Y KO et Lukasz ZIAREK (2017). « RAINA : Reliability and Adaptability in Android for Fog Computing ». In : *IEEE Communications Magazine* 55.4, p. 41-45.

DASTJERDI, Amir Vahid et Rajkumar BUYYA (2016). « Fog computing : Helping the Internet of Things realize its potential ». In : *Computer* 49.8, p. 112-116.

DASTJERDI, Amir Vahid et al. (2016). « Fog computing : Principles, architectures, and applications ». In : *arXiv preprint arXiv :1601.02752*.

DE BRITO, Mathias Santos et al. (2016). « Towards programmable fog nodes in smart factories ». In : *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*. IEEE, p. 236-241.

DIBAJI, Seyed Mehran et al. (2019). « A systems and control perspective of CPS security ». In : *Annual Reviews in Control*.

DOLUI, Koustabh et Soumya Kanti DATTA (2017). « Comparison of edge computing implementations : Fog computing, cloudlet and mobile edge computing ». In : *Global Internet of Things Summit (GIoTS), 2017*. IEEE, p. 1-6.

DRATH, Rainer et Alexander HORCH (2014). « Industrie 4.0 : Hit or hype ?[industry forum] ». In : *IEEE industrial electronics magazine* 8.2, p. 56-58.

DUBEY, Harishchandra et al. (2017). « Fog Computing in Medical Internet-of-Things : Architecture, Implementation, and Applications ». In : *Handbook of Large-Scale Distributed Computing in Smart Healthcare*. Springer, p. 281-321.

DUCQ, Yves et David CHEN (2008). « How to measure interoperability : Concept and Approach ». In : *2008 IEEE International Technology Management Conference (ICE)*. IEEE, p. 1-8.

ELGENDY, Ibrahim A et al. (2019). « Resource allocation and computation offloading with data security for mobile edge computing ». In : *Future Generation Computer Systems* 100, p. 531-541.

EVANS, Dave (2011). « The internet of things : How the next evolution of the internet is changing everything ». In : *CISCO white paper* 1.2011, p. 1-11.

FERREIRO, Susana et al. (2016). « Industry 4.0 : predictive intelligent maintenance for production equipment ». In : *European Conference of the Prognostics and Health Management Society, no*, p. 1-8.

FIRDHOUS, Mohamed, Osman GHAZALI et Suhaidi HASSAN (2014). « Fog computing : Will it be the future of cloud computing ? » In : The Third International Conference on Informatics & Applications (ICIA2014).

FRAGA-LAMAS, Paula et al. (2018). « A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard ». In : *IEEE Access* 6, p. 13358-13375.

FRAGA-LAMAS, Paula et al. (2019). « Design and Validation of a Bluetooth 5 Fog Computing Based Industrial CPS Architecture for Intelligent Industry 4.0 Shipyard Workshops ». In : *arXiv preprint arXiv :1903.00713*.

GAZIS, Vangelis et al. (2015). « Components of fog computing in an industrial internet of things context ». In : *Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015 12th Annual IEEE International Conference on*. IEEE, p. 1-6.

GEORGAKOPOULOS, Dimitrios et al. (2016). « Internet of Things and edge cloud computing roadmap for manufacturing ». In : *IEEE Cloud Computing* 3.4, p. 66-73.

GRANGEL-GONZÁLEZ, Irlán et al. (2016). « Towards a semantic administrative shell for industry 4.0 components ». In : *Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on*. IEEE, p. 230-237.

GRÖGER, Christoph, Florian NIEDERMANN et Bernhard MITSCHANG (2012). « Data mining-driven manufacturing process optimization ». In : *Proceedings of the world congress on engineering*. T. 3, p. 4-6.

GU, Lin et al. (2015). « Cost efficient resource management in fog computing supported medical cyber-physical system ». In : *IEEE Transactions on Emerging Topics in Computing* 5.1, p. 108-119.

GUAN, Yunguo et al. (2018). « Data security and privacy in fog computing ». In : *IEEE Network* 32.5, p. 106-111.

GUBBI, Jayavardhana et al. (2013). « Internet of Things (IoT) : A vision, architectural elements, and future directions ». In : *Future generation computer systems* 29.7, p. 1645-1660.

GUPTA, Harshit et al. (2016). « ifogsim : A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments ». In : *arXiv preprint arXiv :1606.02007*.

HASHIZUME, Keiko et al. (2013). « An analysis of security issues for cloud computing ». In : *Journal of Internet Services and Applications* 4.1, p. 5.

HENG, Stefan (2014). « Industry 4.0 : Upgrading of Germany's Industrial Capabilities on the Horizon ». In :

HERMANN, Mario, Tobias PENTEK et Boris OTTO (2016). « Design principles for industrie 4.0 scenarios ». In : *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. IEEE, p. 3928-3937.

HOFMANN, Erik et Marco RÜSCH (2017). « Industry 4.0 and the current status as well as future prospects on logistics ». In : *Computers in Industry* 89, p. 23-34.

HONG, Kirak et al. (2013a). « Mobile fog : A programming model for large-scale applications on the internet of things ». In : *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, p. 15-20.

— (2013b). « Opportunistic spatio-temporal event processing for mobile situation awareness ». In : *Proceedings of the 7th ACM international conference on Distributed event-based systems*. ACM, p. 195-206.

HU, Pengfei et al. (2017a). « Fog computing-based face identification and resolution scheme in internet of things ». In : *IEEE Transactions on Industrial Informatics*.

HU, Pengfei et al. (2017b). « Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things ». In : *IEEE Internet of Things Journal* 4.5, p. 1143-1155.

IMKAMP, Dietrich et al. (2016). « Challenges and trends in manufacturing measurement technology–the "Industrie 4.0" concept ». In : *Journal of Sensors and Sensor Systems* 5.2, p. 325-335.

INTELLIGENCE, SRI Consulting Business (2008). « Disruptive civil technologies ». In : *Six technologies with potential impacts on US interests out to* 2025.

ISLAM, Mohammad Manzurul, Sarwar MORSHED et Parijat GOSWAMI (2013). « Cloud computing : A survey on its limitations and potential solutions ». In : *International Journal of Computer Science Issues (IJCSI)* 10.4, p. 159.

IVANOV, Dmitry, Alexander TSIPOULANIDIS et Jörn SCHÖNBERGER (2019). « Digital Supply Chain, Smart Operations and Industry 4.0 ». In : *Global Supply Chain and Operations Management*. Springer, p. 481-526.

JALALI, Fatemeh et al. (2016). « Fog computing may help to save energy in cloud computing ». In : *IEEE Journal on Selected Areas in Communications* 34.5, p. 1728-1739.

JAZDI, Nasser (2014). « Cyber physical systems in the context of Industry 4.0 ». In : *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*. IEEE, p. 1-4.

JDN. (2020). *Conteneur en informatique : définition précise et fonctionnement*. url = https ://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445240-conteneur-en-informatique-definition-precise-et-fonctionnement/. Accessed : 2021-04-25.

KAGERMANN, Henning et al. (2013). *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*. Forschungsunion.

KANG, Hyoung Seok et al. (2016). « Smart manufacturing : Past research, present findings, and future directions ». In : *International journal of precision engineering and manufacturing-green technology* 3.1, p. 111-128.

KANICKAM, S Hendry Leo, L JAYASIMMAN et A Nisha JEBASEELI (2017). « A Survey on Layer Wise Issues and Challenges in Cloud Security ». In : *Computing and Communication Technologies (WCCCT), 2017 World Congress on*. IEEE, p. 168-171.

KHAN, Saad, Simon PARKINSON et Yongrui QIN (2017). « Fog computing security : a review of current applications and security solutions ». In : *Journal of Cloud Computing* 6.1, p. 1-22.

KHAN, Waqas Ali et al. (2017). « Analysis of the requirements for offering industrie 4.0 applications as a cloud service ». In : *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*. IEEE, p. 1181-1188.

KHERBOUCHE, Lynda et Zohra OUBAHRI (2017). « Quelques méthodes de résolutions en optimisation combinatoire ». Thèse de doct. UMMTO.

KIANI, Abbas, Nirwan ANSARI et Abdallah KHREISHAH (2019). « Hierarchical capacity provisioning for fog computing ». In : *IEEE/ACM Transactions on Networking*.

KOPRAS, Bartosz, Filip IDZIKOWSKI et Paweł KRYSZKIEWICZ (2019). « Power Consumption and Delay in Wired Parts of Fog Computing Networks ». In : *2019 IEEE Sustainability through ICT Summit (StICT)*. IEEE, p. 1-8.

KOSKINEN, Miina (2016). « Microservices and containers : benefits and best practices ». In :

KRAEMER, Frank Alexander et al. (2017). « Fog Computing in Healthcare–A Review and Discussion ». In : *IEEE Access*.

KUBERNETSS. *Kubernetes*. https ://www.https ://kubernetes.io/.

LA, Quang Duy et al. (2019). « Enabling intelligence in fog computing to achieve energy and latency reduction ». In : *Digital Communications and Networks* 5.1, p. 3-9.

LEE, Edward A (2006). « Cyber-physical systems-are computing foundations adequate ». In : *Position Paper for NSF Workshop On Cyber-Physical Systems : Research Motivation, Techniques and Roadmap*. T. 2.

LEE, Gilsoo, Walid SAAD et Mehdi BENNIS (2018). « Online optimization for UAV-assisted distributed fog computing in smart factories of industry 4.0 ». In : *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, p. 1-6.

LEE, Jay, Moslem AZAMFAR et Jaskaran SINGH (2019). « A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems ». In : *Manufacturing Letters* 20, p. 34-39.

LEE, Jay, Behrad BAGHERI et Hung-An KAO (2015). « A cyber-physical systems architecture for industry 4.0-based manufacturing systems ». In : *Manufacturing Letters* 3, p. 18-23.

LEE, Jay, Hung-An KAO et Shanhu YANG (2014). « Service innovation and smart analytics for industry 4.0 and big data environment ». In : *Procedia Cirp* 16, p. 3-8.

LEE, Jay et al. (2015). « Industrial big data analytics and cyber-physical systems for future maintenance & service innovation ». In : *Procedia CIRP* 38, p. 3-7.

LENZINI, Gabriele, Sjouke MAUW et Samir OUCHANI (2016). « Analysing the Efficacy of Security Policies in Cyber-Physical Socio-Technical Systems ». In : *International Workshop on Security and Trust Management*. Springer, p. 170-178.

LIN, Chun-Cheng et Jhih-Wun YANG (2018). « Cost-efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0 ». In : *IEEE Transactions on Industrial Informatics*.

LIN, Shi-Wan et al. (2015). « Industrial internet reference architecture ». In : *Industrial Internet Consortium (IIC), Tech. Rep*.

LIU, Yang, Jonathan E FIELDSEND et Geyong MIN (2017). « A Framework of Fog Computing : Architecture, Challenges and Optimization ». In : *IEEE Access*.

LOM, Michal, Ondrej PRIBYL et Miroslav SVITEK (2016). « Industry 4.0 as a part of smart cities ». In : *Smart Cities Symposium Prague (SCSP), 2016*. IEEE, p. 1-6.

LOPEZ (2014). « Building smarter Manufacturing With The Internet of Things (IoT) ». In : *The IoT Series*.

LUAN, Tom H et al. (2015). « Fog computing : Focusing on mobile users at the edge ». In : *arXiv preprint arXiv :1502.01815*.

LUONG, Nguyen Cong et al. (2016). « Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models : a survey ». In : *IEEE Communications Surveys & Tutorials* 18.4, p. 2546-2590.

MAHMUD, Redowan et al. (2019a). « Context-aware placement of industry 4.0 applications in fog computing environments ». In : *IEEE Transactions on Industrial Informatics*.

MAHMUD, Redowan et al. (2019b). « Quality of Experience (QoE)-aware placement of applications in Fog computing environments ». In : *Journal of Parallel and Distributed Computing* 132, p. 190-203.

MANAVALAN, E et K JAYAKRISHNA (2019). « A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements ». In : *Computers & Industrial Engineering* 127, p. 925-953.

MARKAKIS, Evangelos K et al. (2017). « EXEGESIS : Extreme Edge Resource Harvesting for a Virtualized Fog Environment ». In : *IEEE Communications Magazine* 55.7, p. 173-179.

MATT, Christian (2018). « Fog computing ». In : *Business & information systems engineering* 60.4, p. 351-355.

MATTERN, Friedemann et Christian FLOERKEMEIER (2010). « From the Internet of Computers to the Internet of Things ». In : *From active data management to event-based systems and more*, p. 242-259.

MEIDAN, Yair et al. (2011). « Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining ». In : *IEEE Transactions on Semiconductor Manufacturing* 24.2, p. 237-248.

MELL, Peter, Tim GRANCE et al. (2011). « The NIST definition of cloud computing ». In :

MIHALACHE, Alin Gabriel (2007). « Modélisation et évaluation de la fiabilité des systèmes mécatroniques : application sur système embarqué ». Thèse de doct.

MUKHERJEE, Mithun et al. (2017a). « A fog computing-based framework to reduce traffic overhead in large-scale industrial applications ». In : *Computer Communications Workshops (INFOCOM WKSHPS), 2017 IEEE Conference on*. IEEE, p. 1008-1009.

MUKHERJEE, Mithun et al. (2017b). « Security and Privacy in Fog Computing : Challenges ». In : *IEEE Access*.

MUTLAG, Ammar Awad et al. (2019). « Enabling technologies for fog computing in healthcare IoT systems ». In : *Future Generation Computer Systems* 90, p. 62-78.

NAIN, Grégory et al. (2010). « Integrating iot and ios with a component-based approach ». In : *Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on*. IEEE, p. 191-198.

NEGASH, Behailu, Tomi WESTERLUND et Hannu TENHUNEN (2019). « Towards an interoperable Internet of Things through a web of virtual things at the Fog layer ». In : *Future Generation Computer Systems* 91, p. 96-107.

NI, Jianbing et al. (2017). « Securing fog computing for internet of things applications : Challenges and solutions ». In : *IEEE Communications Surveys & Tutorials* 20.1, p. 601-628.

NING, Zhaolong, Jun HUANG et Xiaojie WANG (2019). « Vehicular fog computing : Enabling real-time traffic management for smart cities ». In : *IEEE Wireless Communications* 26.1, p. 87-93.

NIST. (2020). *Trustworthy Systems, Components, and Data for Smart Manufacturing Program*. https://www.nist.gov/programs-projects/trustworthy-systems-components-and-data-smart-manufacturing-program. Accessed : 2021-04-20.

O'DONOVAN, Peter et al. (2018). « A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications ». In : *Manufacturing Letters* 15, p. 139-142.

OPENFOG. *Fogconsortium*. https://www.openfogconsortium.org/.

OSANAIYE, Opeyemi et al. (2017). « From cloud to fog computing : A review and a conceptual live VM migration framework ». In : *IEEE Access*.

O'DONOVAN, Peter et al. (2019). « A comparison of fog and cloud computing cyber-physical interfaces for Industry 4.0 real-time embedded machine learning engineering applications ». In : *Computers in Industry* 110, p. 12-35.

PALLEWATTA, Samodha, Vassilis KOSTAKOS et Rajkumar BUYYA (2019). « Microservices-based IoT Application Placement within Heterogeneous and Resource Constrained Fog Computing Environments ». In : *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, p. 71-81.

PALLIS, George (2010). « Cloud computing : the new frontier of internet computing ». In : *IEEE internet computing* 14.5, p. 70-73.

PAN, Jianli et James MCELHANNON (2017). « Future edge cloud and edge computing for internet of things applications ». In : *IEEE Internet of Things Journal* 5.1, p. 439-449.

PERALTA, Goiuri et al. (2017). « Fog computing based efficient IoT scheme for the Industry 4.0 ». In : *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*. IEEE, p. 1-6.

PERERA, Charith et al. (2014). « Context aware computing for the internet of things : A survey ». In : *IEEE Communications Surveys & Tutorials* 16.1, p. 414-454.

PERERA, Charith et al. (2017). « Fog computing for sustainable smart cities : a survey ». In : *arXiv preprint arXiv :1703.07079*.

PISCHING, Marcos A et al. (2015). « Service Composition in the Cloud-Based Manufacturing Focused on the Industry 4.0. » In : *DoCEIS*, p. 65-72.

PIZOŃ, Jakub et Jerzy LIPSKI (2016). « Perspectives for Fog Computing in Manufacturing ». In : *Applied Computer Science* 12.

PORTER, Michael E et James E HEPPELMANN (2015). « How smart, connected products are transforming companies ». In : *Harvard business review* 93.10, p. 96-114.

PUTHAL, Deepak et al. (2019). « Fog Computing Security Challenges and Future Directions [Energy and Security] ». In : *IEEE Consumer Electronics Magazine* 8.3, p. 92-96.

QI, Qinglin et Fei TAO (2019). « A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing ». In : *IEEE Access* 7, p. 86769-86777.

QU, T et al. (2016). « IoT-based real-time production logistics synchronization system under smart cloud manufacturing ». In : *The International Journal of Advanced Manufacturing Technology* 84.1-4, p. 147-164.

RAHMAN, Hafizur et Md Iftekhar HUSSAIN (2019). « Fog-based semantic model for supporting interoperability in IoT ». In : *IET Communications* 13.11, p. 1651-1661.

RAHMANI, Amir M et al. (2018). « Exploiting smart e-health gateways at the edge of healthcare internet-of-things : a fog computing approach ». In : *Future Generation Computer Systems* 78, p. 641-658.

RAJKUMAR, Ragunathan Raj et al. (2010). « Cyber-physical systems : the next computing revolution ». In : *Proceedings of the 47th Design Automation Conference*. ACM, p. 731-736.

RAMALHO, Flávio de Sousa (2016). « SmartEdge : fog computing cloud extensions to support latency-sensitive IoT applications ». Mém. de mast. Brasil.

RASPBERRYPI. (2020). *Raspberry pi 3 mode*. url = https ://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. Accessed : 2021-04-22.

RAZVAN BEURAN, RB (2004). « Mesure de la qualité dans les réseaux informatiques ». Thèse de doct. Jean Monnet U.

REZABAKHSH, Behrang et al. (2006). « Consumer power : a comparison of the old economy and the Internet economy ». In : *Journal of Consumer Policy* 29.1, p. 3-36.

RIMAL, Bhaskar Prasad et al. (2011). « Architectural requirements for cloud computing systems : an enterprise cloud approach ». In : *Journal of Grid Computing* 9.1, p. 3-26.

ROMAN, Rodrigo, Javier LOPEZ et Masahiro MAMBO (2018). « Mobile edge computing, fog et al. : A survey and analysis of security threats and challenges ». In : *Future Generation Computer Systems* 78, p. 680-698.

RÖPKE, Hannes et al. (2016). « Identification of "Industrie 4.0" component hierarchy layers ». In : *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*. IEEE, p. 1-8.

ROY, Sandip, Rajesh BOSE et Debabrata SARDDAR (2015). « A fog-based DSS model for driving rule violation monitoring framework on the Internet of things ». In : *International Journal of Advanced Science and Technology* 82, p. 23-32.

SADEGHI, Ahmad-Reza, Christian WACHSMANN et Michael WAIDNER (2015). « Security and privacy challenges in industrial internet of things ». In : *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, p. 1-6.

SAGIROGLU, Seref et Duygu SINANC (2013). « Big data : A review ». In : *Collaboration Technologies and Systems (CTS), 2013 International Conference on*. IEEE, p. 42-47.

SAHA, Pankaj et al. (2016). « Integrating apache airavata with docker, marathon, and mesos ». In : *Concurrency and Computation : Practice and Experience* 28.7, p. 1952-1959.

SALDIVAR, Alfredo Alan Flores et al. (2015). « Industry 4.0 with cyber-physical integration : A design and manufacture perspective ». In : *Automation and computing (icac), 2015 21st international conference on*. IEEE, p. 1-6.

SAREEN, Sanjay, Sunil Kumar GUPTA et Sandeep K SOOD (2017). « An intelligent and secure system for predicting and preventing Zika virus outbreak using Fog computing ». In : *Enterprise Information Systems*, p. 1-21.

SARKAR, Subhadeep et Sudip MISRA (2016). « Theoretical modelling of fog computing : a green computing paradigm to support IoT applications ». In : *IET Networks* 5.2, p. 23-29.

SATYANARAYANAN, Mahadev (2017). « The emergence of edge computing ». In : *Computer* 50.1, p. 30-39.

SCHUH, Guenther et al. (2014a). « Global Footprint Design based on genetic algorithms–An "Industry 4.0" perspective ». In : *CIRP Annals-Manufacturing Technology* 63.1, p. 433-436.

SCHUH, Günther et al. (2014b). « Collaboration Mechanisms to increase Productivity in the Context of Industrie 4.0 ». In : *Procedia CIRP* 19, p. 51-56.

SCHÜTZE, Andreas, Nikolai HELWIG et Tizian SCHNEIDER (2018). « Sensors 4.0–smart sensors and measurement technology enable Industry 4.0 ». In : *Journal of Sensors and Sensor Systems* 7.1, p. 359-371.

SEITZ, Andreas, Dominik BUCHINGER et Bernd BRUEGGE (2018). « The Conjunction of Fog Computing and the Industrial Internet of Things-An Applied Approach ». In : *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, p. 812-817.

SHAFI, Qaisar (2012). « Cyber physical systems security : A brief survey ». In : *Computational Science and Its Applications (ICCSA), 2012 12th International Conference on*. IEEE, p. 146-150.

SHAFIQ, Syed Imran et al. (2015). « Virtual engineering object/virtual engineering process : a specialized form of cyber physical system for Industrie 4.0 ». In : *Procedia Computer Science* 60, p. 1146-1155.

SHANOUN, M et al. (2011). « Optimisation of the process control in a semiconductor company : model and case study of defectivity sampling ». In : *International Journal of Production Research* 49.13, p. 3873-3890.

SHARMA, Deepak, Amritesh OJHA et Amol P BHONDEKAR (2019). « Heterogeneity consideration in wireless sensor networks routing algorithms : a review ». In : *The Journal of Supercomputing* 75.5, p. 2341-2394.

SHAUKAT, Usman et al. (2016). « Cloudlet deployment in local wireless networks : Motivation, architectures, applications, and open challenges ». In : *Journal of Network and Computer Applications* 62, p. 18-40.

SHI, Chenhua et al. (2018). « Ultra-low latency cloud-fog computing for industrial internet of things ». In : *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, p. 1-6.

SHIN, Seung-Jun, Jungyub WOO et Sudarsan RACHURI (2014). « Predictive analytics model for power consumption in manufacturing ». In : *Procedia CIRP* 15, p. 153-158.

SHROUF, Fadi, Joaquin ORDIERES et Giovanni MIRAGLIOTTA (2014). « Smart factories in Industry 4.0 : A review of the concept and of energy management approached in production based on the Internet of Things paradigm ». In : *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on*. IEEE, p. 697-701.

SINGH, Meena et al. (2015). « Secure mqtt for internet of things (iot) ». In : *2015 fifth international conference on communication systems and network technologies*. IEEE, p. 746-751.

SINGH, Simar Preet et al. (2019). « Fog computing : from architecture to edge computing and big data processing ». In : *The Journal of Supercomputing* 75.4, p. 2070-2105.

SOO, Sander, Chii CHANG et Satish Narayana SRIRAMA (2016). « Proactive Service Discovery in Fog Computing using Mobile Ad Hoc Social Network in Proximity ». In : *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on*. IEEE, p. 561-566.

SORIANO, Javier et al. (2013). « Internet of Services ». In : *Evolution of Telecommunication Services*. Springer, p. 283-325.

STEINER, Wilfried et Stefan POLEDNA (2016). « Fog computing as enabler for the Industrial Internet of Things ». In : *e & i Elektrotechnik und Informationstechnik* 133.7, p. 310-314.

STOJMENOVIC, Ivan et Sheng WEN (2014). « The fog computing paradigm : Scenarios and security issues ». In : *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, p. 1-8.

STOLFO, Sal, Steven M BELLOVIN et David EVANS (2011). « Measuring security ». In : *IEEE Security & Privacy* 9.3, p. 60-65.

STOLFO, Salvatore J, Malek Ben SALEM et Angelos D KEROMYTIS (2012). « Fog computing : Mitigating insider data theft attacks in the cloud ». In : *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*. IEEE, p. 125-128.

SUNDMAEKER, Harald et al. (2010). « Vision and challenges for realising the Internet of Things ». In : *Cluster of European Research Projects on the Internet of Things, European Commision* 3.3, p. 34-36.

SWARM. *Swarm*. https ://docs.docker.com/engine/swarm/.

TAO, Fei et al. (2014). « CCIoT-CMfg : cloud computing and internet of things-based cloud manufacturing service system ». In : *IEEE Transactions on Industrial Informatics* 10.2, p. 1435-1442.

TAO, Fei et al. (2019). « Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0 : correlation and comparison ». In : *Engineering. Forthcoming*.

TAO, Ming, Kaoru OTA et Mianxiong DONG (2017). « Foud : Integrating Fog and Cloud for 5G-Enabled V2G Networks ». In : *IEEE Network* 31.2, p. 8-13.

TRAPPEY, Amy JC et al. (2016). « A review of essential standards and patent landscapes for the Internet of Things : A key enabler for Industry 4.0 ». In : *Advanced Engineering Informatics*.

TRENTESAUX, Damien (2009). « Distributed control of production systems ». In : *Engineering Applications of Artificial Intelligence* 22.7, p. 971-978.

UNGUREAN, Ioan, Nicoleta-Cristina GAITAN et Vasile Gheorghita GAITAN (2014). « An IoT architecture for things from industrial environment ». In : *Communications (COMM), 2014 10th International Conference on*. IEEE, p. 1-4.

VAQUERO, Luis M et Luis RODERO-MERINO (2014). « Finding your way in the fog : Towards a comprehensive definition of fog computing ». In : *ACM SIGCOMM Computer Communication Review* 44.5, p. 27-32.

VARGHESE, Blesson et al. (2017). « Feasibility of Fog Computing ». In : *arXiv preprint arXiv :1701.05451*.

VARGO, Stephen L et Robert F LUSCH (2004). « Evolving to a new dominant logic for marketing ». In : *Journal of marketing* 68.1, p. 1-17.

VARSHNEY, Prateeksha et Yogesh SIMMHAN (2017). « Demystifying Fog Computing : Characterizing Architectures, Applications and Abstractions ». In : *arXiv preprint arXiv :1702.06331*.

VERBA, Nandor et al. (2019). « Modeling industry 4.0 based fog computing environments for application analysis and deployment ». In : *Future Generation Computer Systems* 91, p. 48-60.

VIJAYKUMAR, S, SG SARAVANAKUMAR et M BALAMURUGAN (2016). « Unique sense : Smart computing prototype for industry 4.0 revolution with IOT and bigdata implementation model ». In : *arXiv preprint arXiv :1612.09325*.

VOHRA, Deepak (2016). *Kubernetes microservices with Docker*. Apress.

WAIDNER, Michael et Michael KASPER (2016). « Security in industrie 4.0 : challenges and solutions for the fourth industrial revolution ». In : *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. EDA Consortium, p. 1303-1308.

WAN, Jiafu, Hu CAI et Keliang ZHOU (2015). « Industrie 4.0 : enabling technologies ». In : *Intelligent Computing and Internet of Things (ICIT), 2014 International Conference on*. IEEE, p. 135-140.

WAN, Jiafu et al. (2018). « Fog computing for energy-aware load balancing and scheduling in smart factory ». In : *IEEE Transactions on Industrial Informatics* 14.10, p. 4548-4556.

WANG, Bo et al. (2019a). « Vulnerability assessment method for cyber-physical system considering node heterogeneity ». In : *The Journal of Supercomputing*, p. 1-21.

WANG, Shiyong et al. (2016). « Implementing smart factory of industrie 4.0 : an outlook ». In : *International Journal of Distributed Sensor Networks*.

WANG, Tian et al. (2019b). « Coupling resource management based on fog computing in smart city systems ». In : *Journal of Network and Computer Applications* 135, p. 11-19.

WANG, Tian et al. (2019c). « Solving Coupling Security Problem for Sustainable Sensor-Cloud Systems based on Fog Computing ». In : *IEEE Transactions on Sustainable Computing*.

WARIS, Mohammad Maqbool, Cesar SANIN et Edward SZCZERBICKI (2018). « Community of Practice for Product Innovation Towards the Establishment of Industry

4.0 ». In : *Asian Conference on Intelligent Information and Database Systems*. Springer, p. 651-660.

WEINER, Matthew et al. (2014). « Design of a low-latency, high-reliability wireless communication system for control applications ». In : *Communications (ICC), 2014 IEEE International Conference on*. IEEE, p. 3829-3835.

WIKIPEDIA (2020). *List of Buzz*. https : / / en . wikipedia . org / wiki / ListofbuzzwordsScienceandtechnology. Accessed : 2021-04-22.

WU, Dazhong et al. (2013). « Cloud manufacturing : Strategic vision and state-of-the-art ». In : *Journal of Manufacturing Systems* 32.4, p. 564-579.

WU, Dazhong et al. (2017). « A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing ». In : *Journal of Manufacturing Systems* 43, p. 25-34.

WU, Zhipeng, Zhaozong MENG et John GRAY (2017). « IoT-based Techniques for Online M2M-Interactive Itemised Data Registration and Offline Information Traceability in a Digital Manufacturing System ». In : *IEEE Transactions on Industrial Informatics*.

XU, Li Da et Lian DUAN (2019). « Big data for cyber physical systems in industry 4.0 : A survey ». In : *Enterprise Information Systems* 13.2, p. 148-169.

XU, Xun (2012). « From cloud computing to cloud manufacturing ». In : *Robotics and computer-integrated manufacturing* 28.1, p. 75-86.

YAO, Hong et al. (2017). « Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing ». In : *Concurrency and Computation : Practice and Experience* 29.16, e3975.

YASRAB, Robail (2018). « Mitigating docker security issues ». In : *arXiv preprint arXiv :1804.05039*.

YEBOAH-BOATENG, Ezer Osei et Kofi Asare ESSANDOH (2014). « Factors influencing the adoption of cloud computing by small and medium enterprises in developing economies ». In : *International Journal of Emerging Science and Engineering* 2.4, p. 13-20.

YEH, Wei-Chang, Chyh-Ming LAI et Jr-Yu TSAI (2019). « Simplified swarm optimization for optimal deployment of fog computing system of industry 4.0 smart factory ». In : *Journal of Physics : Conference Series*. T. 1411. 1. IOP Publishing, p. 012005.

YI, Shanhe, Cheng LI et Qun LI (2015). « A survey of fog computing : concepts, applications and issues ». In : *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, p. 37-42.

YING, S et J SZTIPANOVITS (2013). « Foundations for innovation in cyber-physical systems ». In : *Workshop Report, Energetics Incorporated, Columbia, Maryland, US*.

YOUSEFPOUR, Ashkan et al. (2019). « All one needs to know about fog computing and related edge computing paradigms : A complete survey ». In : *Journal of Systems Architecture*.

YU, Yong et al. (2018). « Assured Data Deletion with Fine-grained Access Control for Fog-based Industrial Applications ». In : *IEEE Transactions on Industrial Informatics*.

ZAKOLDAEV, DA et al. (2019). « Centralized, decentralized, hybrid principles of cyber-physical systems control of the Industry 4.0 ». In : *Journal of Physics : Conference Series*. T. 1353. 1. IOP Publishing, p. 012143.

ZANDER, Jens et Petri MÄHÖNEN (2013). « Riding the data tsunami in the cloud : myths and challenges in future wireless access ». In : *IEEE Communications Magazine* 51.3, p. 145-151.

ZHANG, Decheng et al. (2019). « Model and algorithms for the planning of fog computing networks ». In : *IEEE Internet of Things Journal* 6.2, p. 3873-3884.

ZHANG, Peng et al. (2016). « An efficient access control scheme with outsourcing capability and attribute update for fog computing ». In : *Future Generation Computer Systems*.

ZHONG, Ray Y et al. (2017). « Intelligent manufacturing in the context of industry 4.0 : a review ». In : *Engineering* 3.5, p. 616-630.

ZHOU, Yiqing et al. (2019). « Fog computing enabled future mobile communication networks : a convergence of communication and computing ». In : *IEEE Communications Magazine* 57.5, p. 20-27.

ZWEREG (20008). « Internet of Service ». In : *eChallenges*.

Publications Durant la thèse de doctorat

[C1] Bouzarkouna, I., Sahnoun, M. H., Sghaier, N., Baudry, D., & Gout, C. (2018, August). **Challenges facing the industrial implementation of fog computing**. In 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud) (pp. 341-348). IEEE.

[C2] Bouzarkouna, I., Sahnoun, M. H., Baudry, D., & Benatia, M. A. (2020, May). **Fog-based Logistic**

Application Modeling in an Industry 4.0 Framework. In IOP Conference Series: Materials Science and Engineering (Vol. 842, No. 1, p. 012026). IOP Publishing.

[C3] Abid. O., Bouzarkouna. I., Sahnoun., M., Bouziane., B., & David, B. (2019, September). **Deployment of fog computing platform for cyber physical production system based on docker technology**. In 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID) (Vol. 1, pp. 1-6). IEEE.

[C4] Sahnoun, M., Xu, Y., Bettayeb, B., Bouzarkouna, I., Baudry, D., & Louis, A. (2019, September). **Fractal modeling of Cyber physical production system using multi-agent systems.** In 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID) (Vol. 1, pp. 1-6). IEEE.

[C5] Bouzarkouna, I., Sahnoun, M., Bettayeb, B., Baudry, D., & Gout, C. (2021,June). **Optimal Fog-Based Architecture For Internal Logistics in Industry 4.0 Production Floor.** In 2021 1st International Conference On Cyber Management And Engineering (CyMaEn)