

---

## DEDICACE

**À mes chers parents,** Votre amour infini, vos sacrifices et votre soutien indéfectible ont été les fondations solides qui ont guidé chacun de mes pas. Ce rapport est dédié à vous, qui avez toujours cru en moi et m'avez inspiré à poursuivre mes rêves. Merci pour tout.

**À mes merveilleux frères et ma sœur,**

A travers les hauts et les bas, votre présence a été ma source de force. Chacun de vous a apporté une couleur unique à ma vie, et je vous dédie ce travail en reconnaissance de notre lien indissoluble.

**À mes amis exceptionnels,** Votre amitié n'a pas seulement égayé les moments de joie, mais a également été le réconfort nécessaire lorsque les défis semblaient insurmontables.

**À vous tous,**

je dédie ce travail avec une profonde gratitude..

*Noura FITOURI*

# **REMERCIEMENT**

Mes remerciements les plus sincères à tous ceux et celles qui nous ont apporté leur aide, leurs encouragements et leur soutien pour me permettre de mener à bien ce projet.

Je tiens tout d'abord à exprimer ma profonde gratitude envers mon encadrant académique, monsieur **Abderrahmen GUERMAZI**, pour son soutien inestimable tout au long de mon stage de fin d'études. Ses conseils éclairés, sa disponibilité et son expertise ont été des atouts précieux qui ont grandement contribué à l'enrichissement de mon expérience professionnelle.

Mes remerciements s'adressent également à mon encadrant professionnel, monsieur **Kamel BEN GHZAIEL**, sa bienveillance, son partage de connaissances et son accompagnement attentif ont grandement facilité mon intégration dans le milieu professionnel et ont favorisé mon apprentissage pratique.

Je souhaite également exprimer ma profonde gratitude envers les membres du jury qui ont accepté d'évaluer ce travail. Vos remarques importantes que vous fournirez seront des éléments précieux pour mon développement académique et professionnel.

Enfin, je n'oublie pas de remercier les enseignants de l'**Institut Supérieur des Études Technologiques de Sfax** pour la qualité de formation. Les connaissances acquises au cours de mon parcours académique m'ont permis de mettre en pratique les compétences nécessaires tout au long de mon projet de fin d'études.

# ■ TABLE DES MATIÈRES

<b>LISTE DES FIGURES</b>	vii
<b>LISTE DES TABLEAUX</b>	viii
<b>LISTE DES ABRÉVIATIONS</b>	ix
<b>INTRODUCTION GÉNÉRALE</b>	x
<b>1 Chapitre 1 : ETUDE PREALABLE</b>	2
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueil . . . . .	3
1.2.1 Présentation générale de la société « MPBS » . . . . .	3
1.2.2 Organigramme de la société « MPBS » . . . . .	4
1.2.3 Activité principale de la société « MPBS » . . . . .	4
1.3 Présentation du projet . . . . .	5
1.3.1 Contexte . . . . .	5
1.3.2 Problématique . . . . .	5
1.3.3 Etat d'art des Systèmes automatisés . . . . .	6
1.3.4 Présentation des différentes gammes d'automates Siemens . . . . .	9
1.3.5 Etude de l'existant . . . . .	10
1.3.6 Critique de l'existant . . . . .	15
1.3.7 Orientations et Solution proposée . . . . .	15
1.4 Conclusion . . . . .	19
<b>2 Chapitre 2 : ETAT D'ART ET ETUDES COMPARATIVES</b>	20
2.1 Introduction . . . . .	21
2.2 Méthodes d'acquisition de données . . . . .	21
2.2.1 Surveillance d'état . . . . .	21
2.2.2 Algorithme de seuil . . . . .	22
2.2.3 Analyse de tendances . . . . .	23

## TABLE DES MATIÈRES

---

2.3	Etudes des méthodes utilisées pour la lecture des données au système automatisé de la MPBS . . . . .	23
2.4	Méthode choisie pour la lecture des données . . . . .	26
2.5	Le développement Back-End . . . . .	27
2.5.1	Etude comparative des technologies Back-End . . . . .	28
2.5.2	Choix de la solution Python . . . . .	31
2.6	Développement mobile multiplateforme . . . . .	31
2.6.1	Etude comparative des technologies de développement mobile multiplateforme . . . . .	32
2.6.2	Choix de la solution Flutter . . . . .	33
2.7	Connectivité des objets via Internet . . . . .	34
2.7.1	Etude comparative des passerelles IoT . . . . .	35
2.7.2	Choix de la carte Raspberry pi 4 . . . . .	37
2.8	Cloud computing . . . . .	38
2.8.1	Etude comparative de différents Cloud IoT . . . . .	38
2.8.2	Choix de cloud Firebase . . . . .	41
2.9	Conclusion . . . . .	42
<b>3</b>	<b>Chapitre 3 :SPECIFICATION ET CONCEPTION</b>	<b>43</b>
3.1	Introduction . . . . .	44
3.2	Méthodologie de travail adoptée . . . . .	44
3.2.1	Choix de la méthodologie Modèle-Vue-Contrôleur MVC . . . . .	44
3.2.2	Trello . . . . .	46
3.2.3	Unified Modeling Language UML . . . . .	47
3.3	Analyse de besoins . . . . .	47
3.3.1	Identifications des acteurs . . . . .	48
3.3.2	Spécification des besoins fonctionnels . . . . .	48
3.3.3	Diagramme de cas d'utilisation global . . . . .	48
3.3.4	Spécification des besoins non fonctionnels . . . . .	52
3.4	Conception . . . . .	53
3.4.1	Diagrammes de séquences . . . . .	53
3.4.2	Diagramme de classes . . . . .	57
3.5	Conclusion . . . . .	58
<b>4</b>	<b>Chapitre 4 : REALISATION</b>	<b>59</b>
4.1	Introduction . . . . .	60
4.2	Environnement matériel . . . . .	60

## TABLE DES MATIÈRES

---

4.2.1	Architecture générale du nouveau système . . . . .	60
4.2.2	Dispositifs matériels . . . . .	63
4.3	Environnement logiciel . . . . .	65
4.3.1	Outils et Framework de développement . . . . .	65
4.3.2	Installation, configuration et paramétrage . . . . .	69
4.4	Réalisation du Dashboard . . . . .	79
4.5	Conclusion . . . . .	85
<b>CONCLUSION ET PERSPECTIVES</b>		<b>86</b>
<b>BIBLIOGRAPHIE</b>		<b>87</b>
<b>ANNEXES</b>		<b>89</b>

# **LISTE DES FIGURES**

---

1.1	Logo de la société "MPBS" . . . . .	3
1.2	L'organigramme de la société « MPBS » . . . . .	4
1.3	Echange d'information dans un système automatisé [1] . . . . .	7
1.4	Les modules logiques LOGO . . . . .	9
1.5	Siemens S7 200 . . . . .	9
1.6	Siemens S7 400 . . . . .	10
1.7	Siemens S7 1500 . . . . .	10
1.8	Les automates utilisées par la société « MPBS » . . . . .	11
1.9	Architecture existante du système automatisé de la société « MPBS » . . . . .	11
1.10	Diagramme d'état pour notre système . . . . .	17
2.1	Logo du logiciel d'automatisation ouvert (Open Automation Software) . . . . .	25
2.2	Logo WinCC . . . . .	26
2.3	Logo Python . . . . .	27
2.4	Les meilleures technologies Back-End en 2022 . . . . .	28
2.5	La position d'une passerelle IoT dans l'écosystème IoT . . . . .	35
2.6	Carte Beaglebone. . . . .	35
2.7	Carte Raspberry pi 4. . . . .	36
2.8	Carte Rock pi 4. . . . .	36
3.1	Le modèle MVC . . . . .	45
3.2	Tableau de bord de la plateforme Trello. . . . .	47
3.3	Logo UML . . . . .	47
3.4	Diagramme des cas d'utilisation global . . . . .	49
3.5	Diagramme de cas d'utilisation raffinée « suivre ligne de production » . . . . .	49
3.6	Diagramme de cas d'utilisation raffiné « Gérer comptes » . . . . .	51
3.7	Diagramme de séquence « Authentification » . . . . .	54
3.8	Diagramme de séquence « Ajouter compte » . . . . .	55
3.9	Diagramme de séquence « Suivre ligne de production » . . . . .	56

## **LISTE DES FIGURES**

---

3.10 Diagramme de classes . . . . .	57
4.1 Architecture générale à base de Cloud IoT . . . . .	61
4.2 Architecture générale à base d'un serveur Local . . . . .	62
4.3 Automate Siemens S7-400 . . . . .	64
4.4 Rock PI 4 . . . . .	64
4.5 Logo SIMATIC Manager STEP 7 . . . . .	66
4.6 Logo Python . . . . .	66
4.7 Logo Flutter . . . . .	66
4.8 Logo Node-Red . . . . .	67
4.9 Logo Visual Studio Code . . . . .	67
4.10 Logo Firebase . . . . .	67
4.11 Logo Serveur Windows . . . . .	68
4.12 Logo SQL Server Management . . . . .	68
4.13 Installation du bibliothèque Snap7 . . . . .	69
4.14 Etablissement d'une connexion au PLC . . . . .	70
4.15 Interface graphique de SIMATIC Manager STEP 7 . . . . .	71
4.16 Blocs de données siemens . . . . .	71
4.17 Lecture de données de type entier depuis PLC . . . . .	72
4.18 Lecture de données de type booléennes depuis PLC . . . . .	72
4.19 Interface graphique de Node-RED. . . . .	73
4.20 configuration Node-RED. . . . .	74
4.21 Création des noeuds sur Node-RED. . . . .	75
4.22 Configuration des noeuds sur Node-RED. . . . .	75
4.23 Installer le SE du Raspberry Pi à l'aide de Raspberry Pi Imager. . . . .	76
4.24 Création d'un projet dans Firebase. . . . .	77
4.25 Configuration des paramètres de connexion avec Firebase. . . . .	77
4.26 Fenêtre de la structure de la base de données. . . . .	78
4.27 Analyse et stockage des données. . . . .	79
4.28 Interface d'ajout d'un superviseur . . . . .	80
4.29 Interface de connexion . . . . .	81
4.30 Barre de navigation . . . . .	82
4.31 Les indicateurs de performances. . . . .	83
4.32 Identification des cause d'arrêts. . . . .	84
4.33 Sélecteur de plage de dates. . . . .	85

# **LISTE DES TABLEAUX**

1.1	Paramétrage des papiers . . . . .	14
1.2	Exemple de paramétrage de changement des tôles . . . . .	14
2.1	Etude comparative des technologies Back-End . . . . .	30
2.2	Comparaison entre les technologies de développement mobile multiplateforme.	33
2.3	Comparaison entre les différentes cartes. . . . .	37
2.4	Comparaison entre les différents Cloud. . . . .	41
3.1	Description textuelle du cas d'utilisation « Suivre ligne de production » . . . . .	50
3.2	Description textuelle du cas d'utilisation « Gérer comptes » . . . . .	51
3.3	Description textuelle du cas d'utilisation « Authentification » . . . . .	52
3.4	Descriptions des classes . . . . .	58



---

## LISTE DES ACRONYMES

**MDF** Panneaux de Fibres à Densité Moyenne

**MPBS** Manufacture de Panneaux de Bois de Sfax

**HDF** Panneaux de Fibres à haute densité

**PLC** Programmable Logic Controller

**LW** Line Wood

**HMI/SCADA** Human-Machine Interface / Supervisory Control and Data Acquisition

**DAQ** Data acquisition

**OAS** Open Automation Software

**Iot** The Internet of things

**MVC** Model-View-Controller

# INTRODUCTION GÉNÉRALE

L'industrie 4.0, également connue sous le nom de Quatrième Révolution Industrielle, représente une évolution majeure dans le domaine de la fabrication et de la production. Elle englobe l'intégration de technologies numériques avancées au sein des processus industriels traditionnels, visant à améliorer l'efficacité, la productivité et la flexibilité des opérations. Cette révolution est caractérisée par l'interconnexion des machines, la collecte et l'analyse en temps réel des données, ainsi que la mise en œuvre de l'intelligence artificielle et de l'automatisation pour optimiser les performances globales.

Dans ce contexte, les experts du Centre Industrie 4.0 de Sfax ont diagnostiqué le process de fabrication de panneaux stratifiés ainsi que le fonctionnement de la machine de stratification de panneaux de fibres à densité moyenne (MDF) au sein de la Manufacture de Panneaux de Bois de Sfax (MPBS) afin de déterminer des potentiels projets de digitalisation et leurs faisabilités où les objectifs sont : le Suivi instantané de la production par le calcul et l'affichage des indicateurs de performances et la détermination et suivi des causes d'arrêts de la machine. C'est dans ce cadre que s'inscrit mon projet de fin d'études qui consiste à développer un système IoT essentiellement capable de : (1) lire et d'interpréter les données émanant des capteurs des machines puis les stocker sur le cloud. (2) Concevoir et réaliser un Tableau de bord pour le suivi des performances et surtout la justification des arrêts afin d'améliorer la productivité. En fournissant des indicateurs clés et des analyses détaillées sur les performances de la machine de stratification, le tableau de bord permettra aux opérateurs et aux gestionnaires de prendre des décisions éclairées, d'identifier les arrêts potentiels et de maximiser l'efficacité opérationnelle.

## INTRODUCTION GÉNÉRALE

Le rapport est réparti en quatre chapitres, le premier chapitre présente l'étude préalable du projet. Le deuxième chapitre contient l'état d'art, études comparatives et justification de choix. Le troisième chapitre renferme la spécification et conception de la nouvelle solution, alors que le dernier chapitre présente la réalisation de la solution proposée. Le rapport est clôturé par une conclusion générale qui présente un bilan de ce qui a été réalisé tout en proposant des perspectives pour le projet.

---

# **Chapitre 1 : ETUDE PREALABLE**

## **Sommaire**

---

<b>1.1</b>	<b>Introduction</b>	<b>3</b>
<b>1.2</b>	<b>Présentation de l'organisme d'accueil</b>	<b>3</b>
1.2.1	Présentation générale de la société « MPBS »	3
1.2.2	Organigramme de la société « MPBS »	4
1.2.3	Activité principale de la société « MPBS »	4
<b>1.3</b>	<b>Présentation du projet</b>	<b>5</b>
1.3.1	Contexte	5
1.3.2	Problématique	5
1.3.3	Etat d'art des Systèmes automatisés	6
1.3.4	Présentation des différentes gammes d'automates Siemens	9
1.3.5	Etude de l'existant	10
1.3.6	Critique de l'existant	15
1.3.7	Orientations et Solution proposée	15
<b>1.4</b>	<b>Conclusion</b>	<b>19</b>

---

## 1.1 Introduction

Cerner le cadre du projet constitue une étape primordiale à sa réalisation puisqu'elle permet de mettre le travail dans son contexte général. Ainsi, ce chapitre est consacré à la présentation de l'organisme d'accueil puis, on va exposer les objectifs et la problématique du travail. Ensuite, on va passer à l'étude de l'existant à savoir le fonctionnement du système automatisé d'une manière générale, le fonctionnement du système automatisé de la société. Par la suite, on va procéder au critique de l'existant et présenter finalement la solution envisagée. Ce chapitre est clôturé par la description fonctionnelle du projet et la démarche poursuivie.

## 1.2 Présentation de l'organisme d'accueil

Dans le cadre de la préparation du diplôme du mastère DSIR "Développement des Systèmes Informatiques et Réseaux" à l'Institut Supérieur des Etudes Technologiques de Sfax, j'ai eu l'occasion d'effectuer mon mémoire au sein de la société Manufacture de Panneaux Bois du Sud (MPBS). Je suis chargé de développer un projet de digitalisation de la supervision industrielle au sien de la société MPBS.

### 1.2.1 Présentation générale de la société « MPBS »

« MPBS » est une société tunisienne fondée par Mr Mustapha SELLAMI en 1980 à Sfax. C'est une industrie reconnue pour son innovation, sa dynamique de développement et sa dimension humaine, satisfaisant le marché de la transformation et de l'ennoblissement des panneaux de bois. MPBS fournit à ses clients des services d'usinage et de sous-traitance conformes aux normes de qualité internationales les plus exigeantes.



FIGURE 1.1 – Logo de la société "MPBS"

### 1.2.2 Organigramme de la société « MPBS »

En tant que stagiaire à la « MPBS », j'ai été affectée au sein de la direction technique.

La figure 1.2 présente l'organigramme de l'entreprise :

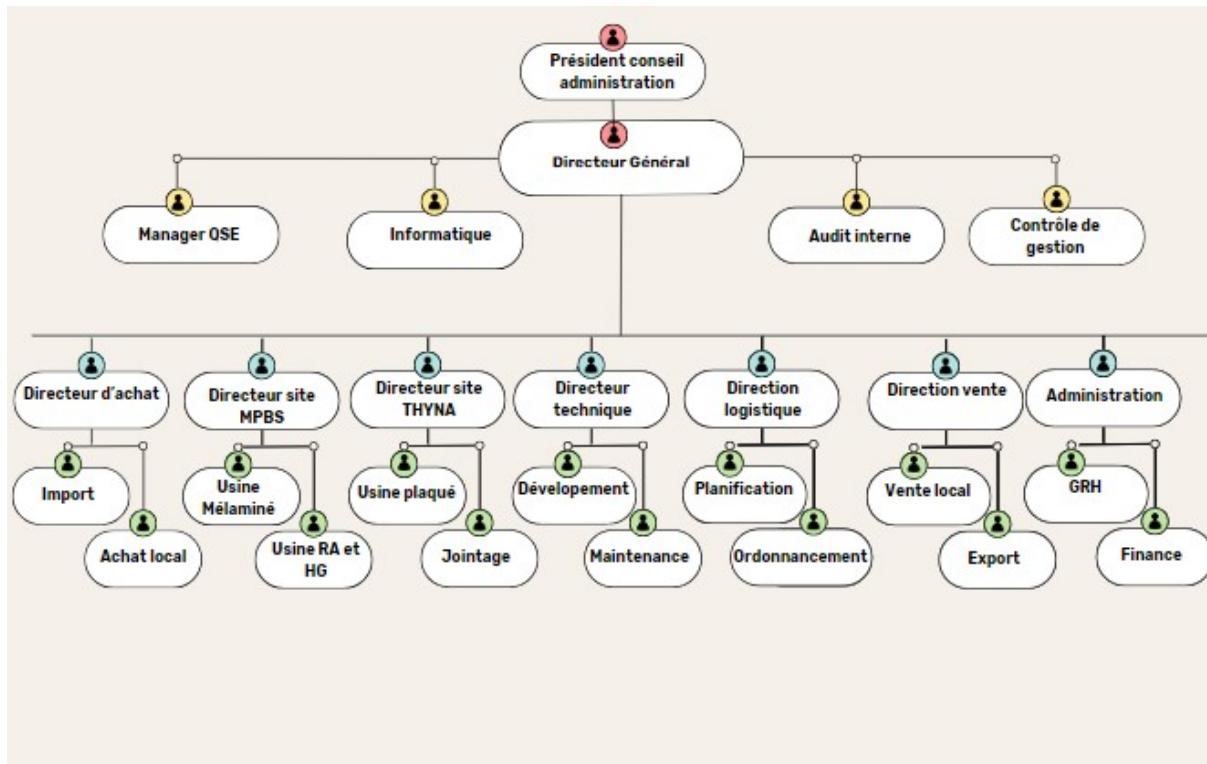


FIGURE 1.2 – L'organigramme de la société « MPBS »

### 1.2.3 Activité principale de la société « MPBS »

MPBS conçoit et anoblit une large gamme de panneaux de bois adaptés à tous les besoins de sa clientèle, qu'ils soient standards ou spécifiques :

- Panneaux contreplaqués en diverses essences de bois
- Panneaux contreplaqués anoblis.
- Panneaux agglomérés mélaminés.
- Panneaux MDF ou HDF mélaminés.
- Panneaux MDF replaqués avec placages tranchés.

Et aussi réaliser des opérations d'usinage sur mesure et des travaux personnalisés.

## 1.3 Présentation du projet

Dans le cadre de ce projet, la société s'engage résolument à explorer des approches novatrices pour surmonter les défis propres à l'industrie, tout en visant à renforcer sa efficacité opérationnelle.

### 1.3.1 Contexte

L'avènement de l'Industrie 4.0 a introduit une révolution majeure dans le domaine de la fabrication et de la production. Les avancées technologiques et la convergence des technologies numériques ont ouvert de nouvelles opportunités pour accroître l'efficacité, la flexibilité et la performance des opérations industrielles. Elle se repose sur la numérisation, l'automatisation et la collecte de données en temps réel pour optimiser les processus de fabrication. Pour ce faire, la société « MPBS » doit se doter des moyens nécessaires et efficaces favorisant le bon déroulement de ses activités par l'emploi d'une technologie de communication moderne, la plus usité aussi bien dans notre vie quotidienne que dans notre activité professionnelle. Ces outils sont en l'occurrence les smartphones et les tablettes qui peuvent servir à la supervision distante des processus industriels et assurer le suivi, principalement, de l'activité de gestion de production.

### 1.3.2 Problématique

Avec l'environnement concurrentiel dans lequel « MPBS » évolue, l'entreprise doit entreprendre le voyage de sa digitalisation et accomplir la transition vers l'ère de l'INDUSTRIE 4.0. Pour ce faire et après avoir discuté avec le directeur de production de la société, il est nécessaire d'améliorer le présent processus d'identification des causes d'arrêts de la machine de stratification. Ce processus repose sur :

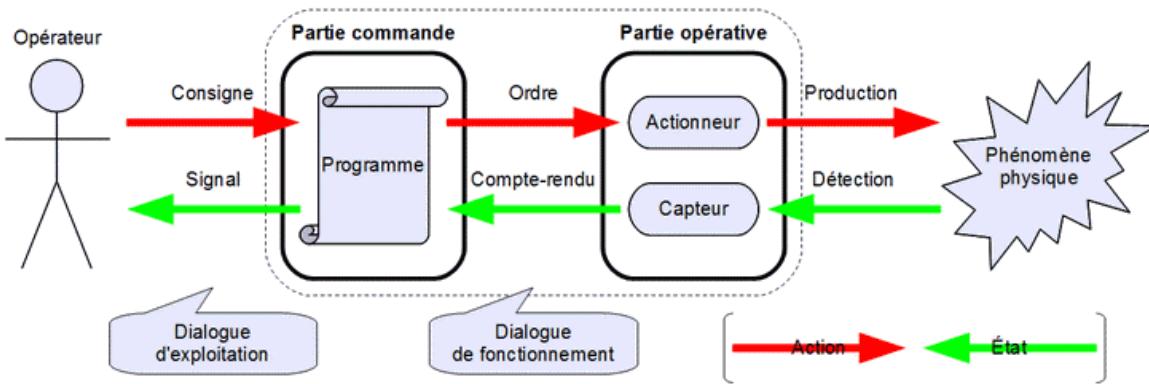
- Une méthode manuelle de recensement des arrêts :

Les opérateurs sont chargés de répertorier à la main les causes d'arrêt survenant pendant leur période de travail. Cette approche engendre des incohérences notables où :

- L'identification des causes d'arrêt dépendant de l'interprétation subjective des opérateurs.
  - La possibilité de divergences entre les opérateurs.
  - Estimation plutôt que mesure précise de la durée des arrêts, introduisant des marges d'erreur considérables.
- Incohérences et inefficacités :
  - Processus manuel de compilation des arrêts entraînant des incohérences.
  - Collecte et agrégation manuelles des informations générant des retards significatifs.
  - Limitation de la réactivité opérationnelle et de la capacité du superviseur à prendre des mesures immédiates.
- Absence de communication en temps réel :
  - La collecte manuelle entrave la capacité du superviseur à obtenir une vue d'ensemble en temps réel.
  - Effet domino en cas de déplacement du superviseur à l'extérieur de l'unité de production.
- Nécessité d'évolution vers l'Industrie 4.0 :
  - Impératif d'une transition vers des méthodes plus avancées.
  - Intégration d'un système automatisé de surveillance et de collecte de données nécessaire.
  - Utilisation d'outils de suivi en temps réel et d'analyses avancées pour optimiser la gestion des arrêts.

### 1.3.3 Etat d'art des Systèmes automatisés

Comme illustré dans la figure suivante, nous exposons le système automatisé à travers un schéma comprenant trois composantes distinctes : la partie opérative, la partie de commande et la partie de dialogue, ainsi que leurs interactions.



**FIGURE 1.3 – Echange d'information dans un système automatisé [1]**

- Le système de commande :

Un système automatisé est un ensemble d'éléments qui effectue des actions sans intervention de l'utilisateur : c'est l'opérateur. Celui-ci se contente de donner des ordres de départ et si besoin d'arrêt. Il est composé de plusieurs parties :

a) La partie opérative : C'est la partie d'un système automatisé qui effectue le travail. Autrement dit, c'est la machine. C'est la partie qui reçoit les ordres de la partie commande et qui les exécute. Elle comporte les capteurs et les actionneurs :

a-1 ) Les actionneurs :

Les actionneurs transforment l'énergie reçue en énergie utile, Ils agissent sur le système ou son environnement, Ils exécutent les ordres reçus tel qu'un déplacement, un dégagement de chaleur, une émission de lumière ou de son.

Exemples d'actionneurs :

Moteur pas à pas, Afficheur 7 segments, Voyants, Electrovanne, Vérin rotatif, Ventilateur, Vérin, Résistance chauffante ...

a-2) Les capteurs :

Les capteurs transforment la variation des grandeurs physiques liées au fonctionnement de l'automatisme en signaux électriques. Ils permettent de recueillir des informations et de les transmettre à la partie commande. Les capteurs sont choisis en fonction des informations qui doivent être recueillies (température, son, lumière, déplacement, position). Les capteurs

communiquent à la partie commande des informations sur la position d'un mobile, une vitesse, la présence d'une pièce, une pression ...

- Les capteurs T.O.R. (tout ou rien) délivrent un signal logique de sortie, c'est à dire 0 ou 1.

Exemple : détecteur de fin de course.

- Les capteurs numériques ou « incrémentaux », associés à un compteur, délivrent des signaux de sortie numérique. Exemple : capteur ou codeur incrémental utilisé pour la mesure des déplacements des chariots de machine à commande numérique.

- Les capteurs analogiques permettent de prendre en compte la valeur réelle d'une grandeur physique. Exemple : sonde de température.

Exemples de capteurs :

Capteur de proximité à ultrasons, Capteur de niveau de liquide, Bouton poussoir, Capteur d'humidité, Cellule photoélectrique, Détecteur de gaz, Détecteur de choc, Capteur à contact, Bouton d'arrêt d'urgence...

### b) La partie commande

Elle donne les ordres et reçoit les informations de l'extérieur ou de la partie opérative. Elle peut se présenter sous 3 manières différentes : un boîtier de commande, un microprocesseur (cerveau électronique), ou un ordinateur.

### c) Partie dialogue

Le pupitre permet à l'opérateur de dialoguer et de commander la partie opérative.

Il comporte :

- Des capteurs de commande (marche, arrêt, arrêt d'urgence ...)
- Des voyants de signalisation (mise sous tension, fonctionnement anormal, buzzer ...).
- Des appareils de mesure de pression (manomètre), de tension (voltmètre), d'intensité (Ampèremètre).
- Interfaces graphiques pour la supervision locale du système industriel.[2]

### 1.3.4 Présentation des différentes gammes d'automates Siemens

La MPBS utilise les automates Siemens, pour cela nous allons voir les différentes gammes d'automates offertes par Siemens. Nous avons listé du plus petit module logique à l'automate haute gamme dédié aux applications les plus robustes.[3]

#### Basse gamme :

- Les modules logiques LOGO : Un module logique programmable compact conçu pour les applications de contrôle simples et de petite envergure. Il est souvent utilisé pour des tâches de surveillance et de contrôle de base.



FIGURE 1.4 – Les modules logiques LOGO

#### Moyenne gamme :

- L'automate SIMATIC S7-200 : Un automate compact et modulaire de Siemens, conçu pour des applications de contrôle simples à moyennement complexes. Il est adapté à un large éventail d'applications industrielles et offre des options d'E/S et de communication pour des tâches de contrôle et d'automatisation. Veuillez noter que le S7-200 est ancien et peut ne plus être répandu.



FIGURE 1.5 – Siemens S7 200

#### Moyenne/haute gamme :

- S7-400 : Un automate modulaire haut de gamme, conçu pour des applications industrielles de grande envergure. Il offre des performances élevées, une redondance intégrée et une grande capacité d'extension.



**FIGURE 1.6 – Siemens S7 400**

- S7-1500 : Un automate modulaire de gamme supérieure, offrant une flexibilité et des performances avancées. Il est adapté aux applications industrielles plus complexes et dispose de fonctionnalités de communication étendues.



**FIGURE 1.7 – Siemens S7 1500**

### 1.3.5 Etude de l'existant

Cette étude comporte une phase de diagnostic dont l'objectif est d'examiner l'état présent du système automatisé de la machine de stratification en tant qu'objet technique exécutant un travail de manière autonome.

#### — Etude du système automatisé de la machine de stratification

Actuellement l'entreprise « MPBS » utilise un système local d'acquisition et de contrôle de données : IHM/SCADA, installée sur un pupitre pour le suivi de la machine de stratification.

Le superviseur peut suivre le fonctionnement du projet avec son PC utilisant le Simatic Manager via une connexion Wi-Fi.

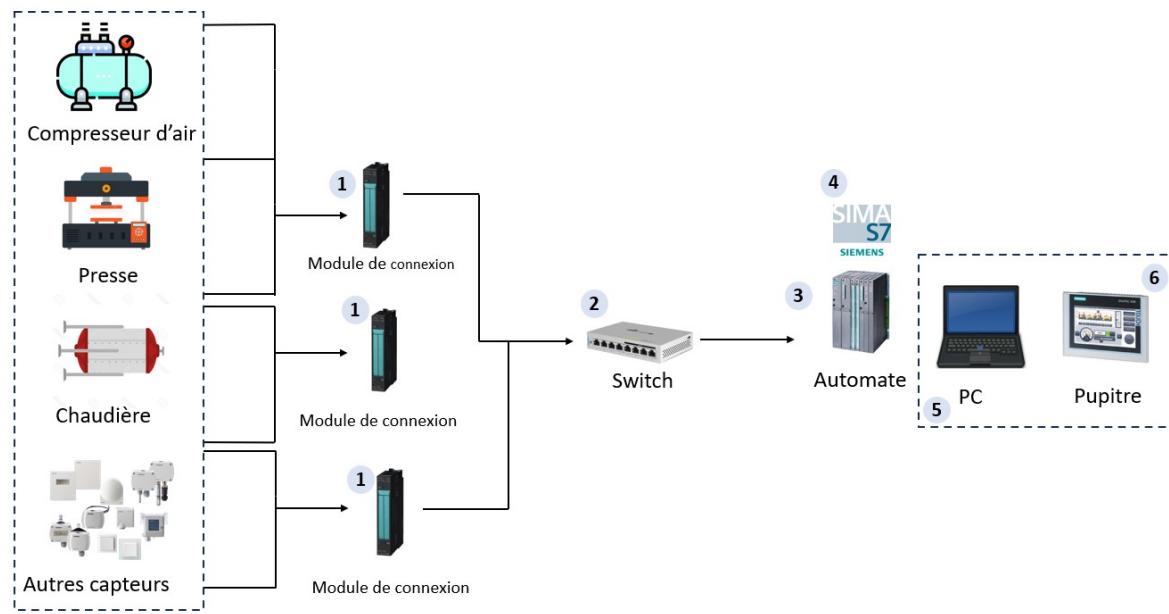


Siemens S7 300

Siemens S7 400

**FIGURE 1.8 – Les automates utilisées par la société « MPBS »**

La Figure 1.8 présente l'architecture existante dans la société « MPBS » :



**FIGURE 1.9 – Architecture existante du système automatisé de la société « MPBS »**

La figure 1.8 présente les principaux éléments constituant le système automatique que nous avons étudié qu'on va aussi les décrire.

**(1) Module de connexion :** sont des modules d'entrées/sorties numériques pour signaux à deux états ou analogiques pour des signaux à évolution continue.

- (2) Switch :** est un commutateur réseau ou un équipement ou un appareil qui permet l’interconnexion de l’automate avec les modules d’entrées/sorties.
- (3) SIMATIC Manager :** est un environnement de développement de systèmes d’automatisation industrielle créé par Siemens. Il s’agit d’un logiciel qui permet aux ingénieurs et aux techniciens de concevoir, configurer, programmer et gérer des systèmes de contrôle et d’automatisation industrielle utilisant des produits Siemens, tels que des automates programmables (PLC) de la famille SIMATIC.
- (4) Automate programmable**
- (5) PC :** sur lequel SIMATIC Manager est installé pour gérer le projet d’automatisation industrielle est généralement configuré de manière spécifique pour répondre aux besoins de développement, de programmation et de gestion de ce projet.
- (6) Pupitre :** c’est l’interface homme machine communique avec l’automate pour obtenir les données.

### — Etude de de la Diversité des Matériaux

Un des défis majeurs auxquels nous avons été confrontés au cours de l’étude de l’existant réside dans la diversité des matériaux utilisés dans notre processus de production. En effet, notre entreprise travaille avec différents types de tôles et de papiers, chacun ayant ses spécificités en termes de paramétrages. Cette variabilité complexe peut rendre la mise en place d’une solution universelle difficile, car les paramètres optimaux pour une tôle ou un papier donné peuvent différer considérablement d’un autre matériau.

Le tableau 1.1 présente quelques exemples des différents paramétrage des papiers :

Référence	Finition	Température	Temps de pressage	Vitesse de Pression
<b>NOYER 53600</b>  NOYER 53600	Line Wood (LW)	165-162 °C	82 secondes	45 kg/ cm <sup>2</sup>
<b>MARRON 2262</b>  MARRON 2262	Matt	200-198°C	17 secondes	28kg /cm <sup>2</sup>

Référence	Finition	Température	Temps de pressage	Vitesse de Pression
<b>CHENE BIEGE</b> 	Nodes	205-202°C	20 secondes	32 kg/cm²

**TABLE 1.1 – Paramétrage des papiers**

Aussi pour le processus de changement des tôles :

Type	Temps de changement	Temps de réglage
<b>La plupart types de tôle</b>	25 minutes	Sans réglage
<b>Nodes</b>	40 minutes	30 minutes

**TABLE 1.2 – Exemple de paramétrage de changement des tôles**

Pour le pressage de papiers pour la tôle nodes :

- Paramétrage : 1/10 de mm pour que la synchronisation de l'empreinte de la tôle avec le papier mélaminé

Le changement de tôle « Nodes » s'avère être une opération plus chronophage en comparaison avec d'autres tôles. Cette particularité est due à l'exigence de surveillance par des caméras spécialement équipées sur notre chaîne de production. Ces caméras sont responsables de la détection précise des croix désignées sur les papiers, ce qui demande un temps supplémentaire pour s'assurer que le processus de changement de tôle est effectué avec une précision maximale.

### 1.3.6 Critique de l'existant

L'actuelle configuration au sein de l'entreprise « MPBS » repose sur l'utilisation d'un système local d'acquisition et de contrôle de données, à savoir une IHM/SCADA installée sur un pupitre. Bien que cette approche ait permis au superviseur de suivre le fonctionnement du projet à l'aide du Simatic Manager via une connexion Wi-Fi, elle présente certaines limites importantes :

- la dépendance à un système local peut entraîner des retards et des contraintes dans le suivi en temps réel des opérations, notamment en cas de problèmes d'arrêts de production sur le site.
- l'utilisation d'un pupitre peut être limitante en termes de mobilité et d'accès aux données depuis des emplacements distants, ce qui peut entraver la réactivité et l'efficacité de la supervision.
- l'usage d'une connexion Wi-Fi peut poser des problèmes de sécurité et de stabilité.

En résumé, bien que le système actuel remplisse sa fonction de manière basique, il est temps de considérer des alternatives plus modernes et flexibles pour améliorer la surveillance et le contrôle des opérations de manufacture et des arrêts.

### 1.3.7 Orientations et Solution proposée

La solution à proposer doit respecter certaines directives à savoir la maîtrise de la complexité du système de digitalisation et son adaptabilité pour les variantes du process de fabrication ainsi que l'identification des causes d'arrêts :

#### 1. Maîtrise de complexité de la solution

La contrainte majeure que nous avons identifiée après avoir examiné la diversité des matériaux réside dans **l'impossibilité de définir une période de temps fixe pour le processus de pressage**, en particulier pour les différentes tôles et papiers. Cette contrainte est étroitement liée à notre processus d'identification des arrêts, car **le temps nécessaire pour chaque processus de pressage peut varier considérablement en fonction des caractéristiques spécifiques du matériau en cours de traitement**.

Par conséquent, l'approche traditionnelle qui repose sur des plages de temps fixes pour détecter les arrêts s'est avérée inadaptée à notre environnement de production diversifié.

Pour résoudre ce défi, notre approche a été axée sur l'adaptabilité de la solution.

### 2. Adaptabilité de la Solution :

L'un des aspects clés de notre solution réside dans la surveillance de l'ouverture de la presse au sein de la ligne de production Wemhöner de la machine de stratification. Dans ce contexte, **l'ouverture de la presse est un indicateur crucial**, car elle peut indiquer un éventuel arrêt de la production. Pour assurer un fonctionnement optimal, nous avons fixé un temps maximal d'ouverture de la presse à 5 minutes.

Ce temps de 5 minutes est défini comme ce que nous appelons une 'période de tolérance'.

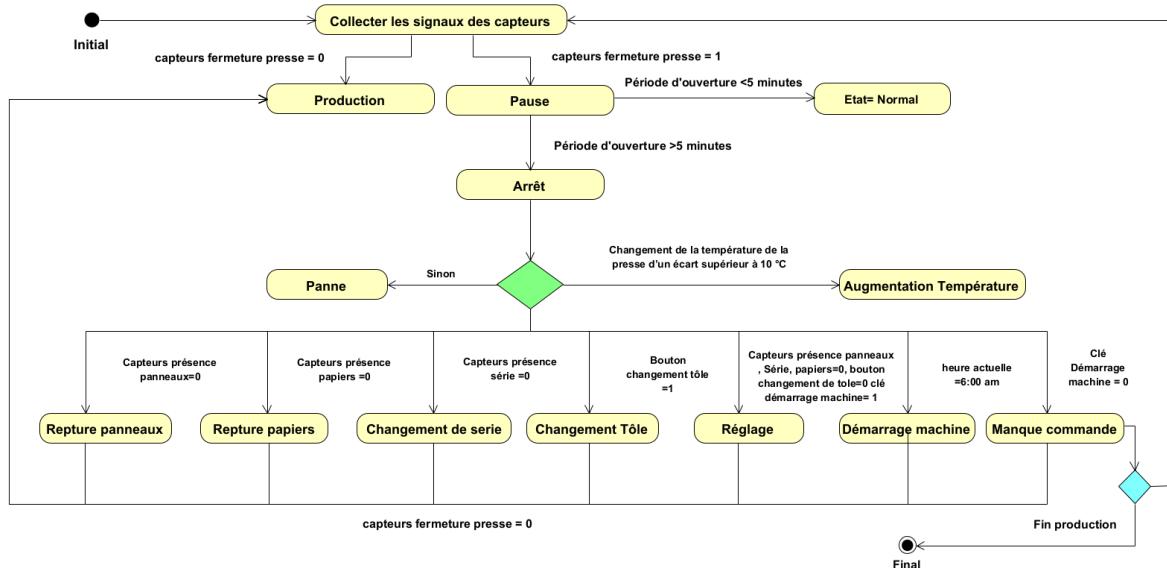
Si le temps d'ouverture de la presse dépasse cette période de tolérance, notre système le détecte automatiquement et signale un état d'arrêt'. Cette période de tolérance a été soigneusement établie en tenant compte de la diversité des matériaux que nous utilisons et des caractéristiques spécifiques à chaque matériau.

Lorsque notre système détecte un état d'arrêt, il passe immédiatement à l'identification des causes d'arrêt. Cela se fait en analysant les états des signaux collectés depuis nos capteurs. Ces signaux fournissent des informations cruciales sur l'état de la production et nous permettent de diagnostiquer rapidement les raisons de l'arrêt. Cette approche proactive garantit que les problèmes potentiels sont identifiés rapidement, minimisant ainsi les temps d'arrêt non planifiés et maintenant un flux de production optimal.

### 3. Identification des causes d'arrêts :

Pour une compréhension plus approfondie du processus d'identification des causes d'arrêt au sein de notre système, nous avons conçu un diagramme d'état. Ce diagramme illustre clairement les différentes étapes que notre système traverse pour diagnostiquer les raisons d'un éventuel arrêt de production.

La figure 1.9 présente les scénarios possibles des causes d'arrêts :



**FIGURE 1.10 – Diagramme d'état pour notre système**

Le diagramme commence par l'état initial, où notre système surveille en permanence les capteurs et vérifie si le temps d'ouverture de la presse est inférieur à la période de tolérance. Si tel est le cas, le système reste dans l'état **Normal**, indiquant que la production se déroule comme prévu.

Cependant, si le temps d'ouverture de la presse dépasse la période de tolérance, le système bascule dans l'état **Arrêt**. À partir de là, le système commence une série de vérifications pour identifier la ou les causes de cet arrêt. Ces vérifications sont basées sur les données des capteurs de la chaîne de la production de la machine.

L'utilisation de ce diagramme d'état nous permet de visualiser de manière claire et systématique le processus d'identification des causes d'arrêt, ce qui est essentiel pour une réaction rapide et précise en cas d'incident de production.

#### 4. Solution proposée

Après avoir examiné en détail la solution qui s'adapte à la diversité des matériaux, nous avons développé la solution finale qui répond à l'ensemble de nos besoins et contraintes spécifiques.

on va chercher à concevoir et réaliser un projet de digitalisation de la supervision industrielle au sein de la société MPBS. Cette solution comporte plusieurs composantes à savoir :

1. Développer un logiciel pont digital reliant d'une manière cohérente les opérations humaines et les machines. Ce dernier novateur établira une communication sans faille avec l'automate. Le cœur de cette démarche réside dans la capacité de ce logiciel à interpréter les données provenant de divers signaux capturés par les équipements de production. Il remplit les fonctionnalités suivantes :
  - Assurer une communication au Contrôleurs logiques programmables « PLC » de l'automate
  - Lire les données des différents signaux des capteurs.
  - Identifier les causes des arrêts en se basant sur les états des capteurs.
  - Calculer les indicateurs de performance (Temps de cycle, les temps des arrêts, Taux de réussite, Taux des arrêts...)
2. Mise en place d'une passerelle IoT permettant de connecter les objets industriels de la MPBS essentiellement des automates.
3. Le développement d'un Tableau de bord qui rend possible à un superviseur :
  - L'accès distant à l'information par le moyen d'un smartphone, tablette ou un écran au moment où on en a besoin, peu importe son emplacement et là où il y a une connexion internet.
  - Le suivi de production en temps réel à un instant t, permettant de savoir exactement l'avancement de la production et agir en conséquence.
  - L'interaction avec les interfaces ergonomiques et conviviales du tableau de bord qui agrège toutes les données et les affiche d'une façon dynamique, simple et compréhensible.

Cette démarche progressive des améliorations contribue à l'optimisation de la productivité, la performance industrielle, la qualité de service et l'exactitude des données.

## 1.4 Conclusion

Dans ce chapitre, on a présenté la société « MPBS » ainsi que ses activités principales. Nous avons exposé la problématique et les objectifs du projet.

Passant à l'étude de l'existant en présentant le fonctionnement du système automatisé d'une manière générale, le fonctionnement réel du système automatisé de la société « MPBS » ainsi son architecture.

Ce travail de diagnostic a été complété par un ensemble de critiques que nous avons achevé par la recherche de la solution adéquate pour finalement proposer une solution qui résoudra notre problématique et répondra à nos objectifs fixés.

Dans le deuxième chapitre, on va aborder l'état d'art, les études comparatives et la justification de choix.

## Chapitre 2 : ETAT D'ART ET ETUDES COMPARATIVES

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>21</b>
<b>2.2</b>	<b>Méthodes d'acquisition de données</b>	<b>21</b>
2.2.1	Surveillance d'état	21
2.2.2	Algorithme de seuil	22
2.2.3	Analyse de tendances	23
<b>2.3</b>	<b>Etudes des méthodes utilisées pour la lecture des données au système automatisé de la MPBS</b>	<b>23</b>
<b>2.4</b>	<b>Méthode choisie pour la lecture des données</b>	<b>26</b>
<b>2.5</b>	<b>Le développement Back-End</b>	<b>27</b>
2.5.1	Etude comparative des technologies Back-End	28
2.5.2	Choix de la solution Python	31
<b>2.6</b>	<b>Développement mobile multiplateforme</b>	<b>31</b>
2.6.1	Etude comparative des technologies de développement mobile multiplateforme	32
2.6.2	Choix de la solution Flutter	33
<b>2.7</b>	<b>Connectivité des objets via Internet</b>	<b>34</b>
2.7.1	Etude comparative des passerelles IoT	35
2.7.2	Choix de la carte Raspberry pi 4	37
<b>2.8</b>	<b>Cloud computing</b>	<b>38</b>
2.8.1	Etude comparative de différents Cloud IoT	38
2.8.2	Choix de cloud Firebase	41
<b>2.9</b>	<b>Conclusion</b>	<b>42</b>

---

## 2.1 Introduction

L'état de l'art est une composante essentielle dans cette étude. Il nous plonge dans le contexte plus large de notre projet en explorant les avancées technologiques et les pratiques actuelles qui sont pertinentes pour notre domaine d'intérêt.

Au sein de ce chapitre, nous aborderons les diverses méthodes d'extraction des données à partir d'un automate Siemens. Par la suite, nous explorerons les applications hybrides, examinerons l'état actuel des solutions de cloud computing, et conduirons une analyse approfondie des passerelles Internet.

## 2.2 Méthodes d'acquisition de données

Les systèmes d'acquisition de données (DAQ) sont des dispositifs utilisés pour acquérir des données à partir de l'environnement qui les entoure. L'acquisition de données permet d'obtenir certaines informations sur l'environnement considéré. En règle générale, les dispositifs d'acquisition de données sont reliés à divers capteurs qui spécifient le phénomène étudié . La plupart des systèmes d'acquisition de données obtiennent des données à partir de différents types de transducteurs qui produisent des signaux analogiques. Dans la plupart des applications, ces signaux doivent être traités.[4]

### 2.2.1 Surveillance d'état

La surveillance d'état des machines a longtemps été réalisée par le biais de méthodes traditionnelles qui impliquent une observation directe et des relevés manuels. L'inspection visuelle par des opérateurs qualifiés est l'une de ces méthodes, où les techniciens inspectent régulièrement les équipements pour détecter visuellement tout signe de détérioration, d'usure ou d'autres anomalies apparentes. Cependant, cette approche est sujette à des limitations liées à la subjectivité humaine, au manque d'objectivité et à la difficulté de repérer des anomalies subtiles.

En outre, les relevés de données manuels constituent une autre approche traditionnelle. Les opérateurs saisissent périodiquement les valeurs mesurées par les capteurs et les enregistrent manuellement dans des journaux. Bien que cette méthode puisse fournir des informations importantes sur les performances de la machine, elle est souvent sujette à des erreurs humaines, à des retards dans la détection des anomalies et à des lacunes dans la couverture temporelle des données.

Ces méthodes traditionnelles de surveillance d'état ont démontré leur utilité, mais elles présentent des inconvénients en termes d'efficacité, de précision et de temps de réaction aux pannes. C'est pourquoi de nouvelles approches exploitant les avancées technologiques et mathématiques ont été développées pour améliorer la détection précoce des défaillances et des causes d'arrêt des machines.

### 2.2.2 Algorithme de seuil

Les algorithmes de seuil ont longtemps été utilisés pour surveiller l'état des machines en se basant sur des valeurs seuils prédéfinies des capteurs. Cette approche implique la définition de seuils supérieurs et inférieurs pour les mesures de capteurs spécifiques. Lorsque les valeurs mesurées dépassent ou chutent en dessous de ces seuils, des alertes sont générées pour signaler des conditions anormales. L'avantage principal de cette méthode est sa simplicité de mise en œuvre et son interprétation intuitive. Cependant, les algorithmes de seuil sont souvent sensibles aux variations normales des données et peuvent générer de fausses alertes en cas de fluctuations temporaires. De plus, ils peuvent ne pas être suffisamment sensibles pour détecter des anomalies subtiles ou des variations progressives qui peuvent indiquer des problèmes imminents. Malgré ces limitations, les algorithmes de seuil ont trouvé leur place dans diverses industries pour la surveillance de machines simples où les variations sont relativement prévisibles. Cependant, avec les progrès de l'apprentissage automatique et de l'analyse de données, de nouvelles approches plus sophistiquées ont émergé pour améliorer la détection de causes d'arrêt plus complexes et subtiles.[5]

### **2.2.3 Analyse de tendances**

L'analyse de tendances historiques des données est une méthode traditionnelle qui vise à identifier des comportements anormaux en observant les schémas et les tendances à long terme dans les données collectées. Cette approche repose sur l'idée que les anomalies peuvent être détectées en comparant les données actuelles avec les données historiques, et en identifiant les déviations significatives par rapport aux tendances établies.

En utilisant cette méthode, les ingénieurs et les opérateurs peuvent détecter des anomalies subtiles ou des changements graduels dans les performances des machines qui pourraient indiquer des problèmes imminents. L'analyse de tendances peut aider à anticiper les défaillances potentielles en identifiant des modèles de dégradation progressive, ce qui permet d'adopter des mesures correctives avant que les problèmes ne deviennent critiques.

Cependant, l'analyse de tendances peut être complexe à mettre en œuvre, en particulier dans des systèmes où les données sont nombreuses et diverses. De plus, elle peut ne pas être aussi sensible à certaines anomalies soudaines ou inattendues. Avec l'avènement de l'apprentissage automatique et de l'analyse de données avancée, de nouvelles approches ont été développées pour combiner les avantages de l'analyse de tendances avec des méthodes plus sophistiquées de détection d'anomalies.

## **2.3 Etudes des méthodes utilisées pour la lecture des données au système automatisé de la MPBS**

Au cours de cette recherche approfondie, diverses méthodes ont été minutieusement évaluées afin de développer une approche optimale pour la lecture des données du système automatisé de la MPBS. Chaque méthode a été soigneusement sélectionnée et testée dans le but d'identifier la solution la plus efficace et adaptée aux besoins spécifiques de notre système :

- **Open Automation software OAS**

Notre première approche de la lecture des données des systèmes d'automatisation de Siemens a consisté à explorer le logiciel Open Automation software c'est un logiciel utilisé pour enregistrer les données de l'industrie 4.0.

IL peut se connecter directement aux contrôleurs Siemens grâce aux interfaces pilotes Siemens intégrées qui prennent en charge les communications par Ethernet avec les S7-200, S7-300, S7-400, S7-1200 et S7-1500.

Après l'installation adéquate du logiciel, vous devez configurer Open Automation Software pour qu'il se connecte au système d'automatisation Siemens.

Cela implique généralement de spécifier le protocole de communication, tel que OPC (OLE for Process Control), et de configurer les paramètres réseau nécessaires, tels que l'adresse IP et le numéro de port de l'appareil Siemens. Open Automation Software s'appuie sur une communication basée sur les balises (Tags) pour lire les données des systèmes d'automatisation Siemens. Configurez les balises (Tags) ou les variables que vous souhaitez récupérer à partir de l'appareil Siemens. Attribuez les types de données appropriés à chaque balise, tels que booléen, entier, virgule flottante ou chaîne de caractères, en fonction des données que vous souhaitez recevoir. Une fois les balises configurées, mettez-les en correspondance avec les points de données correspondants dans le système d'automatisation Siemens. Cette étape établit une connexion entre les balises dans Open Automation Software et les variables ou registres spécifiques dans l'appareil Siemens. Finalement sélectionnez le service OAS dont vous souhaitez extraire la configuration des Tags pour générer un fichier de variables par exemple un fichier CSV. Cependant, malgré les promesses initiales du logiciel Open Automation en tant que solution potentielle, nous avons rencontré un inconvénient important au cours de mon évaluation. Le fichier généré par le logiciel ne contenait pas les valeurs spécifiques des capteurs dont j'avais besoin pour mon projet. Après une enquête plus approfondie, nous avons découvert que le processus de mappage et de récupération des données dans Open Automation Software ne capturait pas avec précision les valeurs de capteur souhaitées du système d'automatisation de Siemens. Cette limitation a rendu la solution inadéquate pour les besoins d'acquisition

de données de notre projet, car elle n'a pas fourni les informations cruciales nécessaires à mon processus d'analyse et de prise de décision.



**FIGURE 2.1 – Logo du logiciel d'automatisation ouvert (Open Automation Software)**

- **WinCC VB script**

Comme alternative à la solution Open Automation Software, j'ai exploré une approche différente en utilisant WinCC, le puissant logiciel de contrôle de surveillance et d'acquisition de données (SCADA) de Siemens. Ma deuxième solution consistait à utiliser un script VB dans WinCC pour générer un rapport Excel. WinCC offre des fonctionnalités étendues pour l'acquisition et la visualisation des données, ce qui en fait une plate-forme adaptée à mon projet. En mettant en œuvre un script VB personnalisé, j'ai cherché à extraire les valeurs de capteur requises du système WinCC et à générer un rapport Excel complet qui réponde aux besoins de mon projet en matière d'analyse de données et de création de rapports. Cependant, alors que la mise en œuvre d'un script VB avec WinCC pour générer un rapport Excel semblait prometteuse, je me suis rapidement rendu compte d'une limitation qui l'empêchait de répondre à notre exigences en matière de prédition des causes d'arrêt en temps réel.

Le rapport Excel n'étant généré que périodiquement, il n'avait pas la fréquence nécessaire pour capturer les mises à jour régulières des valeurs des capteurs requises pour des prédictions précises et opportunes. Cette limitation a mis en évidence la nécessité d'une solution plus dynamique et en temps réel pour atteindre l'objectif de ce projet, à savoir prédire rapidement et efficacement les causes des arrêts.



FIGURE 2.2 – Logo WinCC

## 2.4 Méthode choisie pour la lecture des données

A la recherche d'une solution efficace et précise pour lire les valeurs des capteurs en temps réel, nous sommes orientés vers la programmation **Python** et le package **Snap7** comme approche finale.

Reconnaissant le besoin d'une solution dynamique et fiable capable de gérer l'acquisition de données en temps réel à partir des systèmes d'automatisation Siemens, nous avons développé un script Python utilisant les capacités de la bibliothèque **Snap7**. Ce dernier fournit un ensemble d'outils robustes pour la communication avec les appareils Siemens, ce qui nous a permis d'établir une connexion transparente et d'extraire avec précision les valeurs des capteurs. En tirant parti de la flexibilité et de la polyvalence de Python, nous avons cherché à surmonter les limitations rencontrées dans les solutions précédentes et à réaliser l'acquisition de données en temps réel pour le suivi des causes d'arrêt.

L'utilisation du script Python avec le package Snap7 s'est avérée être la solution adéquate aux objectifs du projet. Cette solution nous permet non seulement de lire avec précision les valeurs des capteurs des systèmes d'automatisation Siemens, mais elle offre également la flexibilité et l'évolutivité nécessaires à l'acquisition de données en temps réel. En enregistrant les valeurs des capteurs dans un fichier CSV et en exécutant le script dans une boucle infinie, nous assurons une collecte continue des données. En outre, en intégrant le script à une passerelle IoT, telle qu'un Raspberry Pi, nous pouvons transmettre les données de manière transparente à un service cloud.

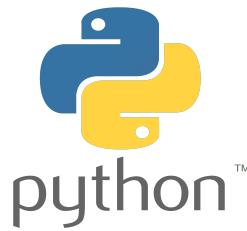


FIGURE 2.3 – Logo Python

## 2.5 Le développement Back-End

Le Back-End est un peu comme la partie immergée d'un iceberg. Le simple utilisateur ne peut la voir pourtant il représente une très grande partie d'un projet web.

Le Back-End se compose de trois éléments :

-Un serveur

-Une application

-Une base de données (sorte de feuille de calcul pour organiser les données). La personne qui administre tout cela est le développeur back end.

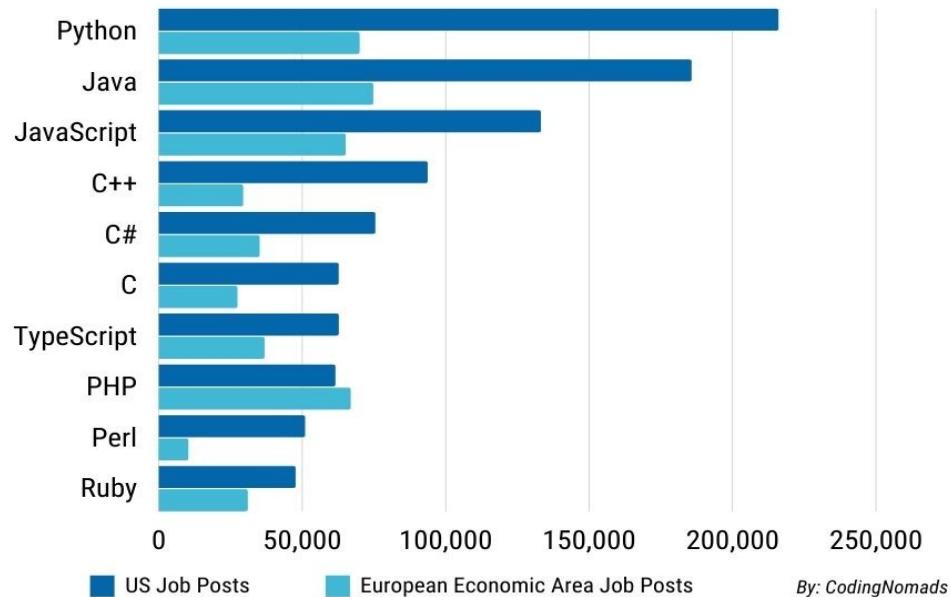
Les technologies Back-End se composent généralement de langages de programmation dite « dynamiques » (PHP, Ruby, Python, etc.), pour les rendre encore plus faciles à utiliser, ils peuvent être améliorés par des Frameworks comme Cake PHP, Ruby on Rails et Django qui rendent le développement plus rapide et plus sécurisé.

Les recherches effectuées sur les langages de programmation les plus demandés en 2022 d'après les offres d'emploi de linkedIn aux États-Unis et en Europe.

Les résultats obtenus sont représentés dans la figure 2.4 :

## Most in-demand programming languages of 2022

*Based on LinkedIn job postings in the USA & Europe*



**FIGURE 2.4 – Les meilleures technologies Back-End en 2022**

### 2.5.1 Etude comparative des technologies Back-End

En tenant compte de ce qui précède, une comparaison entre les technologies Back-End est nécessaire afin d'en dégager les similitudes ou les différences les plus en relief que nous présentons dans le tableau 2.1 suivant :

Technologie Back-end	Description	Avantages	Inconvénients
<b>Ruby</b> 	Haut de niveau	<ul style="list-style-type: none"> <li>- Open-source et fonctionne sur tout SE</li> <li>- La gestion des librairies avec Bundler</li> <li>- Syntaxe flexible</li> </ul>	<ul style="list-style-type: none"> <li>- Peut être très difficile à déboguer à certains moments</li> <li>- Un peu difficile à apprendre</li> <li>- Manque de ressources d'information</li> <li>- Temps de traitement plus lent (temps CPU) par rapport aux autres langages de programmation</li> </ul>

Technologie Back-end	Description	Avantages	Inconvénients
<b>Python</b> 	Dynamique	<ul style="list-style-type: none"> <li>- Open-source et multi paradigme</li> <li>- Facile à apprendre et utiliser (syntaxe propre et simple)</li> <li>- Bibliothèque standard riche et écosystème</li> </ul>	<ul style="list-style-type: none"> <li>-Absence de l'informatique mobile et des navigateurs (le développement d'une application Web avec Python nécessite au moins deux langues différentes, cela peut augmenter le temps et les coûts de maintenance et de débogage)</li> </ul>
<b>Node.js</b> 	Dynamique	<ul style="list-style-type: none"> <li>-Plateforme logicielle libre et fonctionne sur tout SE</li> <li>-Même langage pour le développement Front-End et Back-End</li> <li>-Configuration facile</li> </ul>	<ul style="list-style-type: none"> <li>- Modules pas toujours fiables</li> <li>- API instables</li> <li>- Moins efficace avec les opérations gourmandes en ressources processeur (des tâches lourdes CPU)</li> </ul>

TABLE 2.1 – Etude comparative des technologies Back-End

## 2.5.2 Choix de la solution Python

Suite à l'étude comparative, il s'est avéré qu'une solution d'aide à la décision pour dégager les indices de priorité est nécessaire. Le choix du langage de programmation est une décision cruciale qui influe sur l'efficacité et la réussite de ce projet. Dans le contexte de la solution visant à lire les données en temps réel et à traiter ces données pour détecter les causes d'arrêt et calculer les indicateurs de performance, le choix de Python s'avère pertinent et bénéfique pour plusieurs raisons clés :

- **Large écosystème et bibliothèques :** Python est réputé pour son vaste écosystème de bibliothèques et de modules, qui couvrent une gamme étendue de domaines. Pour la collecte de données en temps réel depuis un automate Siemens, des bibliothèques telles que "snap7", "pyads" peuvent être utilisées pour établir une connexion avec l'automate et récupérer les données sans effort. De plus, les bibliothèques comme "pandas" et "NumPy" permettent un traitement de données efficace et rapide.
- **Flexibilité :** C'est un bon choix pour les applications qui doivent traiter un volume élevé de messages courts nécessitant une faible latence.
- **Traitements et analyse de données :** Python est largement utilisé dans le domaine de l'analyse de données et de l'apprentissage automatique. Les bibliothèques telles que "scikit-learn" et "TensorFlow" offrent des outils puissants pour le développement de modèles prédictifs et d'algorithmes de détection d'anomalies, ce qui est essentiel pour le cœur de votre projet.
- **Intégration facilitée :** Python est connu pour sa capacité à s'intégrer facilement avec d'autres langages et systèmes.

## 2.6 Développement mobile multiplateforme

Le développement multiplateforme consiste à développer des produits ou services logiciels pour plusieurs plates-formes ou environnements logiciels. Les ingénieurs et développeurs utilisent

diverses méthodes pour s'adapter à différents systèmes d'exploitation ou environnements pour une application ou un produit.

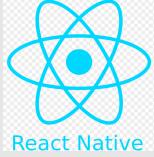
L'idée du développement multiplateforme est qu'une application logicielle ou un produit devrait bien fonctionner dans plus d'un habitat numérique spécifique. Cette capacité est généralement recherchée afin de vendre des logiciels pour plus d'un système d'exploitation propriétaire, afin de permettre une utilisation sur les plates-formes Android et Apple. Avec le développement d'appareils mobiles et d'autres types de plates-formes, ainsi que la prolifération des technologies open source comme Linux, davantage de types de développement multiplateforme sont apparus.

### **2.6.1 Etude comparative des technologies de développement mobile multiplateforme**

L'un des défis les plus courants auxquels sont confrontés les développeurs et les entreprises est de décider quelle technologie multiplateforme utiliser pour atteindre une portée maximale tout en minimisant les coûts et les efforts de développement.

Le tableau 2.3 présente une comparaison entre les technologies de développement mobile multiplateforme :

Technologies	Avantages	Inconvénients	Langages pris en charge	Interface utilisateurs	Tarification
<b>Flutter</b> 	- Performances élevées grâce à son moteur de rendu graphique	- Communauté plus petite par rapport à certaines alternatives	Dart	En croissance	Gratuit

Technologies	Avantages	Inconvénients	Langages pris en charge	Interface utilisateurs	Tarification
<b>React Native</b> 	- Utilise JavaScript, largement utilisé dans le développement web	- Possibilité de besoin de modules natifs pour certaines fonctionnalités	JavaScript (React)	Composants réutilisables	Gratuit
<b>Ionic</b> 	- Utilise HTML, CSS et JavaScript familiers	- Moindres performances comparées aux applications natives	HTML, CSS, JavaScript (Angular)	Composants Web	Plans gratuits et payants

**TABLE 2.2 – Comparaison entre les technologies de développement mobile multiplateforme.**

## 2.6.2 Choix de la solution Flutter

Le tableau comparatif a présenté les points forts et les points faibles de chaque technologie de développement mobiles multiplateforme. Le choix s'est fixé sur Flutter pour les raisons suivantes :

### — Le développement d'un tableau de bord industriel

destiné à afficher les données extraites d'un automate Siemens requiert une approche qui combine performances, fiabilité et esthétique. Dans ce contexte, le choix du framework de développement revêt une importance cruciale pour assurer une expérience utilisateur optimale tout en répondant aux exigences spécifiques de l'industrie. C'est pourquoi notre choix s'est orienté vers Flutter, un framework de développement multiplateforme basé sur le langage Dart, pour la création de notre application.

### — **Intégration du Code Python avec Flutter :**

Une des forces de Flutter réside dans sa capacité à intégrer facilement les composants natifs de différents langages. Dans notre cas, où nous utilisons du code Python pour lire les données depuis l'automate Siemens, Flutter offre des options pour connecter le code Python à l'application, permettant ainsi une synchronisation transparente des données avec l'interface utilisateur du tableau de bord.

### — **Réactivité et Performance :**

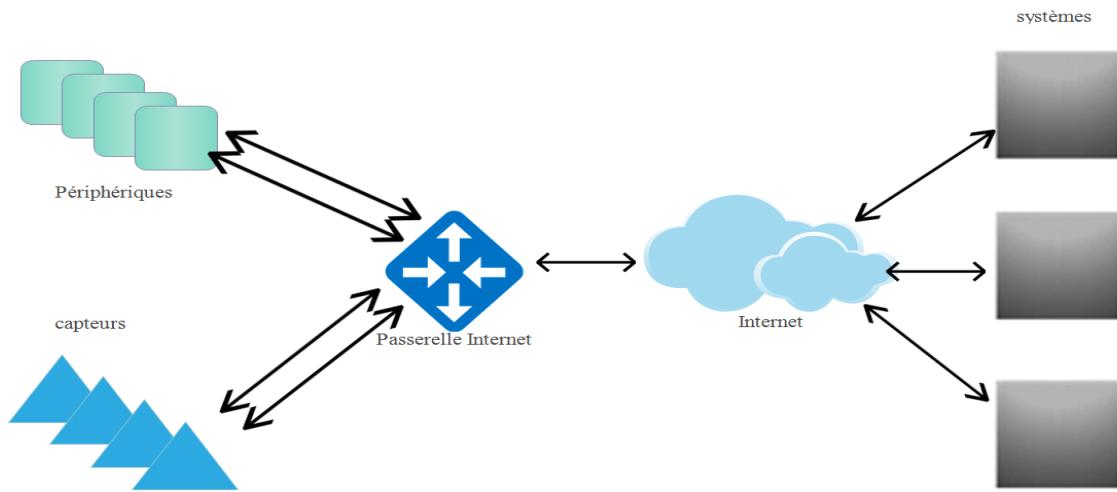
L'industrie exige une réactivité en temps réel et des performances stables pour afficher les données en continu depuis l'automate Siemens. Flutter offre un moteur de rendu graphique optimisé qui assure des performances constantes et élevées, garantissant ainsi une expérience utilisateur fluide et instantanée, même lors de l'affichage de données en temps réel.

### — **Flexibilité pour les Futures Évolutions :**

Dans le secteur industriel, les besoins et les technologies évoluent constamment. Flutter offre une flexibilité pour ajouter de nouvelles fonctionnalités ou s'adapter à de nouveaux scénarios, garantissant ainsi que l'application de tableau de bord pourra suivre l'évolution des exigences industrielles.

## 2.7 Connectivité des objets via Internet

Une passerelle IoT, ou passerelle Internet des objets, est un élément clé dans l'infrastructure des systèmes IoT (Internet des Objets). Elle agit comme un pont entre les dispositifs connectés localement et les services cloud, permettant ainsi la communication bidirectionnelle entre ces deux environnements.



**FIGURE 2.5 – La position d'une passerelle IoT dans l'écosystème IoT.**

### 2.7.1 Etude comparative des passerelles IoT

Durant notre recherche sur la partie Gateway IoT, nous avons constaté qu'au cours des dernières années, de nombreuses cartes de développement modulaire ont apparu. Chaque carte présente des avantages et des inconvénients. Elles diffèrent par leurs fonctionnalités, leurs complexités et leurs prix.

- **La Beaglebone :**

Le BeagleBone Black de beagleboard.org est une carte de développement basée sur le processeur AM335x ARM Cortex-A8. Cette carte BeagleBone est un ordinateur monocarte (SBC) qui peut être utilisé comme un ordinateur autonome ou peut être intégré dans un système.



**FIGURE 2.6 – Carte Beaglebone.**

• **La Raspberry Pi 4 :**

Le Raspberry Pi est un nano-ordinateur mono-carte à processeur ARM conçu à l'université de Cambridge. Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il peut fonctionner avec plusieurs variantes du système d'exploitation libre GNU/Linux (notamment Debian : Raspbian) et des logiciels compatibles.. Le Raspberry Pi 4 de ports USB type C, d'un SoC avec CPU quad-core 1,5 GHz, GPU VideoCore VI prenant en charge OpenGL ES 3.0, décodage HEVC 4K à 60 i/s, deux ports micro HDMI prenant en charge des écrans 4K, et de 1 Go jusqu'à 8 Go de RAM.



**FIGURE 2.7 – Carte Raspberry pi 4.**

• **Rock pi 4 :**

Une série de cartes informatiques monocartes fabriquées par Radxa. Tout comme les Raspberry Pi, les Rock Pi visent à offrir des solutions polyvalentes



**FIGURE 2.8 – Carte Rock pi 4.**

Nom de la carte	Beaglebone	Raspberry Pi 4	Rock Pi 4
Prix (Dinars) [2023]	350 DT	170 DT	306-517 DT

Nom de la carte	Beaglebone	Raspberry Pi 4	Rock Pi 4
<b>Processeur</b>	Sitara335x sur un cortex M8	Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC	2x Cortex-A72 jusqu'à 2.0GHz, 4x Cortex-A53 jusqu'à 1.5GHz
<b>Tension d'entrée</b>	5 V – 3,3 V	5 V	5,5-20V
<b>Consommation (w)</b>	210 à 450 mA (2,5 W max.)	2W-6W	8mA à 20mA
<b>RAM</b>	256 Mo	2 Go, 4 Go ou 8 Go de mémoire LPDDR4-3200 SDRAM	1 ,2, 4 ou 8 Go de RAM LPDDR4
<b>Système d'exploitation</b>	Linux	Linux	Linux

TABLE 2.3 – Comparaison entre les différentes cartes.

## 2.7.2 Choix de la carte Raspberry pi 4

Après l'étude comparative de différents cartes , nous déduisons que le Raspberry représente un outil le plus performant et le mieux adapté à nos besoins pour les raisons suivantes :

- En terme de mémoire vive ou mémoire externe, la capacité de Raspberry est plus importante que les autres cartes ce qui est possible d'installer des logiciels .
- Le Raspberry possède des connectiques (processeur, carte graphique, HDMI, port Ethernet, port USB ...), ce qui signifie que c'est une sorte de micro-ordinateur.
- Suite à une interruption d'alimentation, le Raspberry peut reprendre automatiquement son fonctionnement (auto-Start) au démarrage lorsqu'un courant est rétabli.
- Grâce à sa petite taille, le boîtier Raspberry n'est pas aussi encombrant qu'un ordinateur, par exemple il peut être placé dans une armoire électrique.
- Le fonctionnement de Raspberry est durable grâce à un ventilateur pour refroidir le boîtier.

- Le Raspberry possède des connectiques compatibles à notre système.

## 2.8 Cloud computing

Le paradigme du cloud computing a révolutionné la gestion, le stockage et le traitement des données au sein des entreprises et des projets.

Plutôt que de s'appuyer exclusivement sur des infrastructures locales, le cloud propose une approche souple et évolutive, offrant un accès distant aux ressources informatiques par le biais d'Internet.

Cette approche a ouvert de nouvelles perspectives pour le déploiement d'applications, la gestion des données et la collaboration, transformant fondamentalement les opérations commerciales et la conduite des projets.

### 2.8.1 Etude comparative de différents Cloud IoT

Le tableau 2.5 résume certaines caractéristiques générales, avantages, inconvénients et informations sur la tarification de quelques plateformes :

Cloud	Microsoft Azure IoT	Firebase	AWS IoT Core
Logo			
Tarification	<ul style="list-style-type: none"> <li>- Pour l'instance A0 :                             <ul style="list-style-type: none"> <li>• Cores (noyau) = 1</li> <li>• RAM = 0.75 GB</li> <li>• Stockage temporaire = 20 GB</li> <li>• Prix = £0.015/heure</li> </ul> </li> <li>- Pour l'instance A1 :                             <ul style="list-style-type: none"> <li>• Cores (noyau) = 1</li> <li>• RAM = 1.75 GB</li> <li>• Stockage temporaire = 70 GB</li> <li>• Prix = £0.0597/heure</li> </ul> </li> <li>- Pour l'instance A2 :                             <ul style="list-style-type: none"> <li>• Cores (noyau) = 2</li> <li>• RAM = 3.50 GB</li> <li>• Stockage temporaire = 135 GB</li> <li>• Prix = £0.1193/heure</li> </ul> </li> </ul>	<p>Pour la base de données en temps réel :</p> <ul style="list-style-type: none"> <li>- Commencement gratuit :</li> <li>• GB stocké = 1GB/mois</li> <li>• GB téléchargé = 10 GB/mois</li> <li>- Capacité de stockage &gt; à 1GB : 25/<i>mois</i> :</li> <li>• GB stocké = 2.5 GB</li> <li>• GB téléchargé = 20 GB/mois</li> <li>- capacité de stockage &gt; à 2.5 GB et proportionnellement :</li> <li>• GB stocké = 5/<i>GB</i></li> </ul> <p>L'offre gratuite d'AWS :</p> <ul style="list-style-type: none"> <li>• 2 250 000 minutes de connexion</li> <li>• 500 000 messages</li> <li>-Par exemple,</li> <li>• l'offre gratuite AWS exécute une application de 50 appareils, où chacun d'eux est connecté 24 heures sur 24 :</li> <li>• échange 300 messages par jour</li> <li>• exécute 130 opérations par jour</li> <li>• Déclenche 150 exécutions de règle par jour</li> </ul>	

<b>Cloud</b>	<b>Microsoft Azure IoT</b>	<b>Firebase</b>	<b>AWS IoT Core</b>
Avantages	<ul style="list-style-type: none"> <li>- Flexibilité d'utilisation</li> <li>- Prise en charge d'un large éventail de systèmes d'exploitation, de langages de programmation, de base de données et de périphériques.</li> <li>- Options de Cloud Hybride</li> <li>- Portabilité maximale</li> <li>- Sécurité Cloud maximale</li> </ul>	<ul style="list-style-type: none"> <li>- Base de données en temps réel.</li> <li>- Synchronisation en temps réel.</li> <li>- Rapide et réactif</li> <li>- Fiable, simple et économique</li> <li>- Facilité d'apprentissage</li> <li>- Gratuit initialement</li> <li>- Equilibrage de charges</li> <li>- Performance monitoring</li> <li>- Stockage puissant d'objets</li> </ul>	<ul style="list-style-type: none"> <li>- Flexibilité des protocoles de Communication :</li> <li>- AWS IoT Core prend en charge différents protocoles de communication tels que MQTT, HTTP, et WebSockets, offrant ainsi une flexibilité dans le choix du protocole de communication adapté à vos besoins.</li> </ul>

Cloud	Microsoft Azure IoT	Firebase	AWS IoT Core
Inconvénients	<ul style="list-style-type: none"> <li>- Complexité tarifaire</li> <li>- Service facturé même à l'arrêt de l'application</li> <li>- Pas possible de lancer un batch.exe par l'application pour générer des rapports</li> </ul>	<ul style="list-style-type: none"> <li>- Limite dans la synchronisation des données</li> <li>- Connectivité descendante (serveur très alourdi, temps de réponse affaibli)</li> </ul>	<ul style="list-style-type: none"> <li>- Coûts : Bien qu'AWS IoT Core offre une tarification évolutive, les coûts peuvent devenir significatifs à mesure que le nombre d'appareils et de messages augmente.</li> <li>- Complexité : Pour les débutants, la mise en place d'une solution IoT sur AWS peut sembler complexe en raison de la multitude de services disponibles et de la configuration requise pour les intégrer de manière optimale.</li> </ul>

TABLE 2.4 – Comparaison entre les différents Cloud.

## 2.8.2 Choix de cloud Firebase

L'étude comparative précédente nous a permis de faire la sélection d'un cloud performant qui est la solution pour réaliser notre travail. Les différents critères déterminant notre choix sont les suivants :

### 1. Tarifs

L'un des premiers critères pris en compte lors de notre choix d'un service de stockage en ligne est le prix. En effet, certaines solutions sont gratuites, bien que, de manière générale, les services gratuits proposent un espace de stockage limité et que parfois on est confronté à une augmentation de tarif de la part des opérateurs.

### 2. Capacité de l'espace de stockage

L'espace mis à disposition pour le stockage est évidemment l'un des principaux éléments à prendre en compte lors du choix d'un cloud. Quand il est gratuit, l'espace de stockage offert est un Go. Lorsqu'il est payant, l'offre est- de deux Go ou plus selon les besoins du consommateur.

### 3. Fonctionnalités

Si le principe d'un cloud est de stocker des contenus de différents types, les fonctionnalités additionnelles varient en fonction de chaque prestataire. Selon les habitudes ou les besoins, il est nécessaire de connaître préalablement les différentes fonctionnalités existantes afin de bénéficier d'une utilisation confortable du cloud. En voici quelques exemples :

- La gestion du partage entre plusieurs utilisateurs.
- La protection des données.
- La synchronisation : elle offre la possibilité de modifier un fichier hors ligne puis de le synchroniser lorsqu'une connexion internet est détectée.
- La modification directe de fichiers, sans rapatriement préalable.

## 2.9 Conclusion

En conclusion, ce chapitre consacré à l'état de l'art a été essentiel pour établir une base solide pour notre projet. Il nous a permis d'explorer le paysage technologique actuel, de comprendre les méthodes de collecte de données depuis un automate Siemens, d'analyser les dernières tendances en matière de développement mobile, d'évaluer les solutions de cloud computing et de passerelles IoT.

# Chapitre 3 :SPECIFICATION ET CONCEPTION

## Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>44</b>
<b>3.2</b>	<b>Méthodologie de travail adoptée</b>	<b>44</b>
3.2.1	Choix de la méthodologie Modèle-Vue-Contrôleur MVC	44
3.2.2	Trello	46
3.2.3	Unified Modeling Language UML	47
<b>3.3</b>	<b>Analyse de besoins</b>	<b>47</b>
3.3.1	Identifications des acteurs	48
3.3.2	Spécification des besoins fonctionnels	48
3.3.3	Diagramme de cas d'utilisation global	48
3.3.4	Spécification des besoins non fonctionnels	52
<b>3.4</b>	<b>Conception</b>	<b>53</b>
3.4.1	Diagrammes de séquences	53
3.4.2	Diagramme de classes	57
<b>3.5</b>	<b>Conclusion</b>	<b>58</b>

---

## 3.1 Introduction

Le présent chapitre définit la méthodologie qui sera éventuellement employée pour la réalisation de notre projet, aussi il met l'accent sur la spécification et la conception de l'application mobile. Il s'agit donc de répertorier les intervenants qui interagissent avec le système informatisé et de détailler les exigences regroupées en deux catégories distinctes : les besoins fonctionnels, exprimés à travers un diagramme de cas d'utilisation, et les besoins non fonctionnels.

Après cela, nous nous sommes lancés dans la phase de conception détaillée où nous avons abordé la conception minutieuse de la partie mobile, en mettant particulièrement l'accent sur la modélisation à l'aide de diagrammes de séquences et de diagrammes de classe.

## 3.2 Méthodologie de travail adoptée

Les approches de développement logiciel sont en constante évolution, avec pour s'ajuster aux contraintes environnementales internes et externes qui affectent le projet. Dans cette section, nous allons définir la méthodologie qui sera éventuellement employée pour la réalisation de ce projet.

### 3.2.1 Choix de la méthodologie Modèle-Vue-Contrôleur MVC

Le choix de la méthodologie représente une étape vitale dans le processus de développement de logiciels. En effet, ce choix exerce une influence majeure sur la gestion et l'exécution du projet. La méthodologie adoptée doit être en adéquation avec les besoins spécifiques du projet, en prenant en considération des éléments tels que sa dimension, sa nature et ses buts. Dans notre situation, le choix était d'utiliser la méthodologie MVC, c'est-à-dire une approche de conception architecturale largement utilisée dans le développement de logiciels. Le choix de la méthodologie MVC présente plusieurs avantages clés :

- **Séparation des Responsabilités** : Le modèle MVC permet de séparer clairement les différentes responsabilités au sein de l'application. Le Modèle gère les données et la

logique métier, la Vue gère l'interface utilisateur et l'affichage des données, et le contrôleur gère les interactions entre le Modèle et la Vue. Cette séparation facilite la maintenance, la mise à jour et l'extension de l'application.

- **Réutilisation et Modularité :** Grâce à la structure en couches de MVC, chaque composant peut être développé de manière indépendante, ce qui favorise la réutilisation et la modularité du code. Cela est particulièrement bénéfique lorsque des fonctionnalités similaires, telles que la consultation de chiffres gauge et de statistiques, sont utilisées à différents endroits de l'application.
- **Gestion de l'Interface Utilisateur :** La Vue dans le modèle MVC est spécifiquement dédiée à la gestion de l'interface utilisateur. Dans le cas de notre tableau de bord(Dashboard), la Vue serait responsable de l'affichage des chiffres gauge et des statistiques de manière claire et conviviale pour l'utilisateur.
- **Évolutivité :** La méthodologie MVC facilite l'évolutivité de l'application. Par exemple, si vous décidez d'ajouter de nouvelles fonctionnalités ou de modifier l'affichage des données, vous pouvez le faire en ciblant spécifiquement le Modèle, la Vue ou le Contrôleur concerné sans affecter les autres parties de l'application.

La figure 3.1 illustre les éléments clés du modèle MVC :

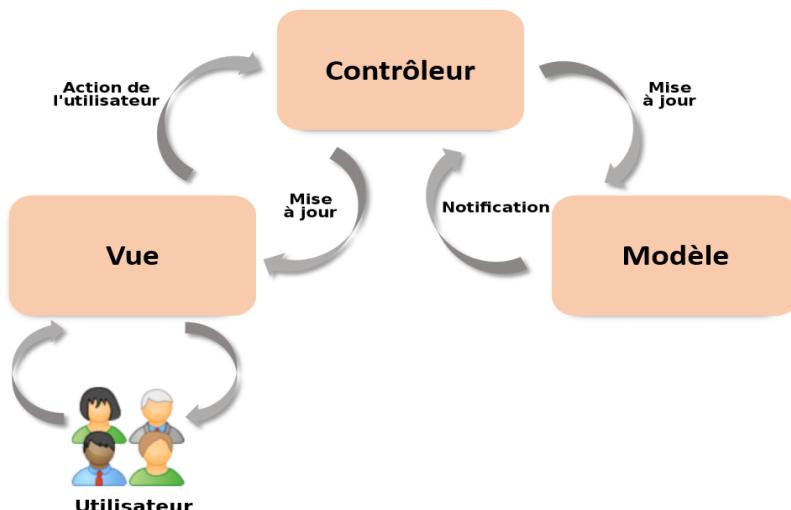


FIGURE 3.1 – Le modèle MVC

### **1. Contrôleur (contient la logique concernant les actions effectuées par l'utilisateur) :**

Il permet la gestion des événements et la synchronisation du modèle et de la vue. Il traite tous les événements de l'utilisateur et lance par la suite les actions nécessaires à effectuer.

### **2. Modèle (contient les données à afficher) :**

Il impose une architecture claire, ce qui simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.

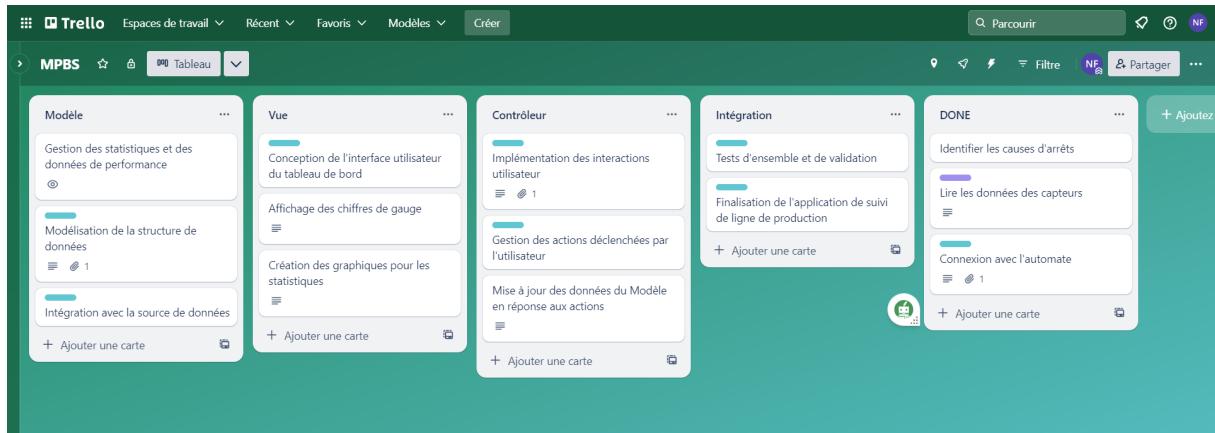
### **3. Vue (contient la présentation de l'interface graphique) :**

La Vue est responsable de l'interface utilisateur et de l'affichage des données au format compréhensible par l'utilisateur. Elle récupère les données pertinentes du Modèle et les présente sous forme graphique ou textuelle. La Vue ne contient pas de logique métier complexe ; elle se concentre uniquement sur la représentation visuelle des données et les interactions avec l'utilisateur. Plusieurs Vues peuvent être associées à un même Modèle pour présenter les données de différentes manières.

### **3.2.2 Trello**

Pour assurer une gestion efficace du projet basé sur la méthodologie MVC, on a opté pour l'utilisation de l'outil bien connu Trello. Trello permet de découper ce projet en tâches distinctes et de les organiser selon les différentes étapes du modèle MVC.

La figure 3.2 démontre le tableau de bord de notre plateforme Trello.



**FIGURE 3.2 – Tableau de bord de la plateforme Trello.**

### 3.2.3 Unified Modeling Language UML

Après avoir défini la méthodologie et choisi l’outil de gestion de projet, nous entrons maintenant dans la phase de conception de la solution logicielle. Pour cela, on a opté pour l’utilisation de l’UML (Unified Modeling Language). Cet outil vise à simplifier la programmation tout en améliorant la clarté des classes, de leurs relations. L’UML, étant un langage de modélisation standard largement reconnu dans l’industrie du développement logiciel, nous a permis d’obtenir une représentation visuelle précise de la structure et du comportement de notre système.



**FIGURE 3.3 – Logo UML**

## 3.3 Analyse de besoins

Le succès d’une application repose nécessairement sur une analyse approfondie des besoins, ainsi que sur une prise de décision éclairée concernant les exigences et les fonctionnalités attendues du système.

Dans cette optique, il est impératif d'identifier les besoins de notre projet de manière adéquate, tout en impliquant les différentes parties prenantes.

### 3.3.1 Identifications des acteurs

Les différents acteurs qui interagissent avec notre système sont :

1. **Administrateur**
2. **Superviseur industriel**

### 3.3.2 Spécification des besoins fonctionnels

Les besoins fonctionnels se rapportent aux fonctions que la solution logicielle doit offrir et mettre en place. En tenant compte de la nature de l'application, les besoins sont différenciés par les deux acteurs :

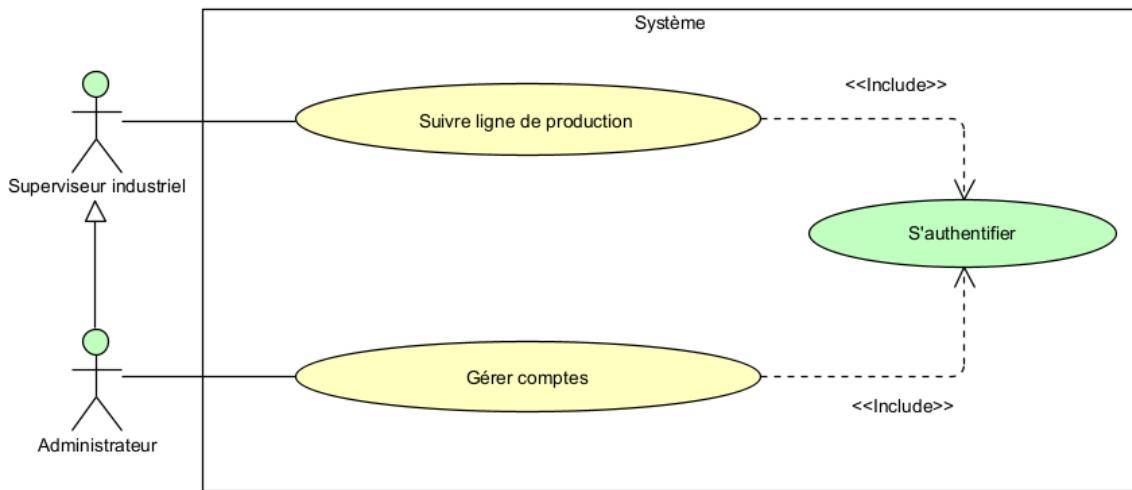
1. L'administrateur est capable de :
  - S'authentifier : saisir un login et un mot de passe.
  - Gérer les comptes utilisateurs : Ajouter, supprimer, modifier.
  - Suivre la ligne de production
2. Le superviseur industriel est capable de :
  - Suivre la ligne de production

Nous avons détaillé ces fonctionnalités par un diagramme global de cas d'utilisation qui va être présenté dans la partie du rapport qui suit.

### 3.3.3 Diagramme de cas d'utilisation global

L'objectif du diagramme de cas d'utilisation est de fournir une vue d'ensemble des interactions futures au sein de l'application. C'est le tout premier diagramme UML qui se compose d'acteurs interagissant avec des scénarios d'utilisation, dépeignant des actions et des réactions qui illustrent le comportement du système du point de vue de l'utilisateur.

Dans la figure nous focaliserons notre attention uniquement sur les acteurs finaux de notre système, à savoir l'administrateur et le superviseur industriel.

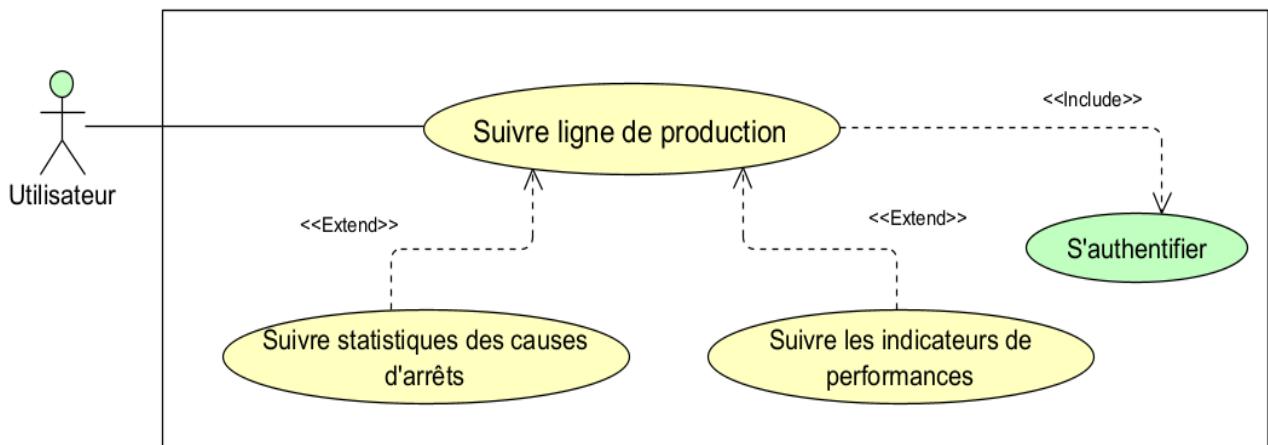


**FIGURE 3.4 – Diagramme des cas d'utilisation global**

L'héritage signifie l'accès avantagé, d'un acteur à un autre. Pour ce cas, un administrateur est une sorte de superviseur donc il peut faire tout ce qui est permis à un superviseur, avec d'autres fonctionnalités.

- **Diagramme de cas d'utilisation raffiné « suivre ligne de production »**

La figure 3.5 illustre le raffinement de diagramme de cas d'utilisation « suivre ligne de production »



**FIGURE 3.5 – Diagramme de cas d'utilisation raffinée « suivre ligne de production »**

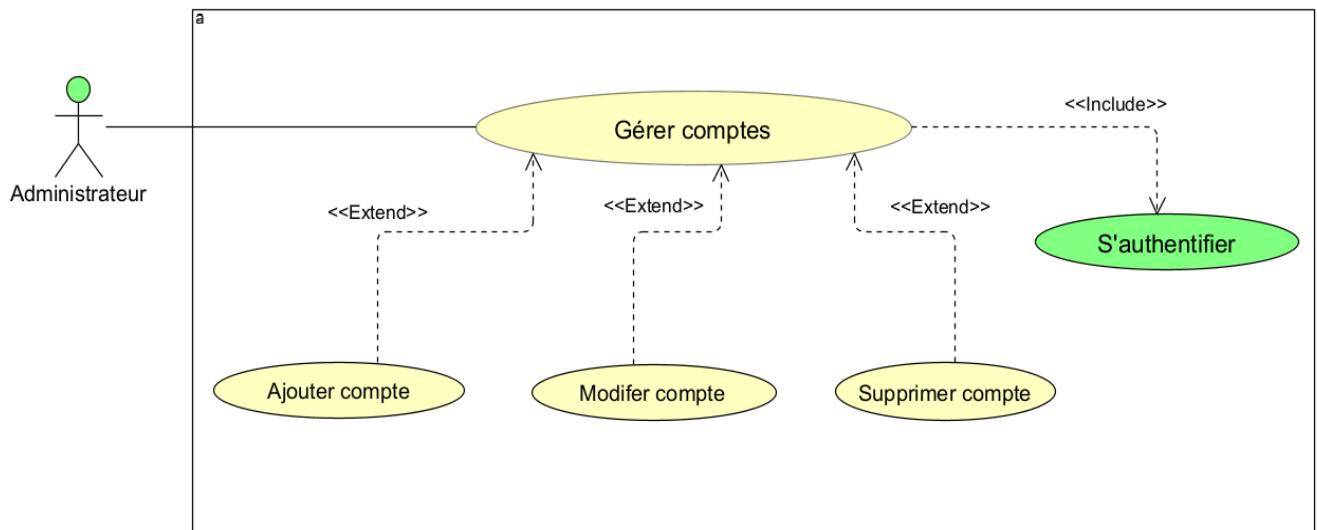
— Description textuelle du cas d'utilisation « Suivre ligne de production »

Sommaire d'identification	
Titre	- Suivre ligne de production
Acteur	- Administrateur, Superviseur industriel
Description des scénarios	
Pré condition	<ul style="list-style-type: none"> <li>- Succès d'authentification.</li> <li>- L'automate est accessible.</li> </ul>
Scénarios nominaux	<p>L'utilisateur suit la ligne de production :</p> <ol style="list-style-type: none"> <li>1. Suivre les causes d'arrêts de la machine.</li> <li>2. Suivre les indicateurs de performances.</li> </ol>
Scénarios alternatifs	<ul style="list-style-type: none"> <li>- Si l'accès à Internet n'est pas disponible, l'application demandera une connexion Internet.</li> <li>- Si l'automate est n'est pas accessible, le système affiche un message d'erreur indiquant la perte de connexion.</li> </ul>
Post condition	- L'utilisateur suit la ligne de production.

TABLE 3.1 – Description textuelle du cas d'utilisation « Suivre ligne de production »

- **Diagramme de cas d'utilisation raffiné « Gérer comptes »**

La figure 3.6 illustre le raffinement de diagramme de cas d'utilisation « Gérer comptes »



**FIGURE 3.6 – Diagramme de cas d'utilisation raffiné « Gérer comptes »**

— **Description textuelle du cas d'utilisation « Gérer comptes »**

Sommaire d'identification	
Titre	- Gérer comptes
Acteur	- Administrateur
Description des scénarios	
Pré condition	<ul style="list-style-type: none"> <li>- Application accessible</li> <li>- Administrateur authentifié.</li> </ul>
Scénarios nominaux	<ol style="list-style-type: none"> <li>1. L'administrateur s'authentifie.</li> <li>2. L'administrateur choisit l'opération qu'il souhaite effectuer :</li> </ol> <p>Ajout, suppression ou modification comptes des superviseurs.</p>
Scénarios alternatifs	<ul style="list-style-type: none"> <li>-2.a L'utilisateur annule l'action : échec du cas d'utilisation.</li> <li>-2.b Données saisies sont invalides : le système affiche un message d'erreur.</li> </ul>
Post condition	L'opération souhaitée par l'administrateur sera effectuée.

**TABLE 3.2 – Description textuelle du cas d'utilisation « Gérer comptes »**

- **Diagramme de cas d'utilisation raffiné « Authentification »**

Sommaire d'identification	
Titre	- Authentification
Acteur	- Administrateur, Superviseur industriel
Description des scénarios	
Pré condition	<ul style="list-style-type: none"> <li>-Application accessible</li> <li>- Existence d'un compte et mot de passe.</li> </ul>
Scénarios nominaux	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page d'authentification.</li> <li>2. Le système affiche le formulaire d'authentification.</li> <li>3. L'utilisateur saisit son login et son mot de passe.</li> <li>4. Le système vérifie l'existence du compte.</li> </ol>
Scénarios alternatifs	<ul style="list-style-type: none"> <li>-Le système affiche un message indiquant :</li> </ul> <p>Utilisateur introuvable ! Veuillez vérifier votre login et mot de passe.</p>
Post condition	<ul style="list-style-type: none"> <li>- L'utilisateur est authentifié et accède aux fonctionnalités qui lui sont dédiées.</li> </ul>

**TABLE 3.3 – Description textuelle du cas d'utilisation « Authentification »**

### 3.3.4 Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement.

Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

Après l'étude de notre projet, nous pouvons citer les besoins non fonctionnels suivants :

- **La disponibilité** : l'application doit être disponible pour être utilisé par n'importe quel utilisateur.
- **La fiabilité** : les données fournies par l'application doivent être fiables.
- **Le déplacement entre les pages** doit être souple et la durée d'exécution des traitements et la synchronisation des variables est en temps réel.

## 3.4 Conception

Le principal objectif de cette phase de travail est de concevoir un tableau de bord qui fournira des informations en temps réel

### 3.4.1 Diagrammes de séquences

Les diagrammes de séquence sont destinés à décrire la manière dont les composants du système interagissent les uns avec les autres.

Le diagramme de séquence présente les interactions entre les objets dans une perspective temporelle et décrit les scénarios types et les exceptions.

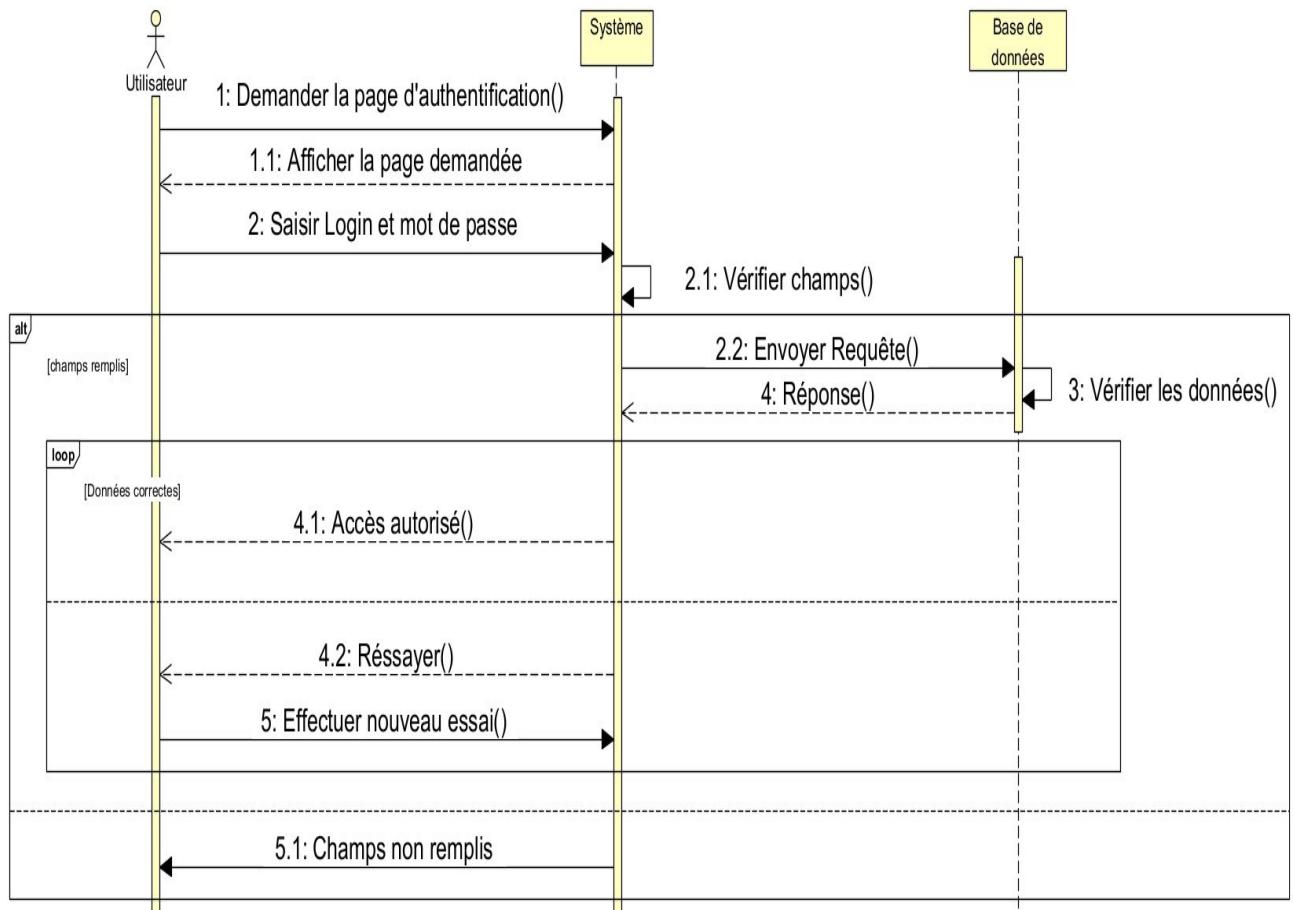
Dans ce qui suit, nous représentons le diagramme de séquence d'un scénario représentatif de chacun des cas d'utilisation décrits précédemment.

- **Diagramme de séquence « Authentification »**

Le scénario initial pour l'utilisateur implique de se connecter au système. Afin d'accéder à l'application, l'utilisateur doit procéder à une authentification en fournissant son adresse e-mail et son mot de passe. L'utilisateur entre ces informations dans les champs de formulaire prévus à cet effet. Le système commence par vérifier si ces champs sont correctement remplis et contiennent des données valides. Si tel n'est pas le cas, le système réagit en affichant un message d'erreur pour informer l'utilisateur.

En cas de succès, le système procède ensuite à une vérification dans la base de données pour confirmer l'existence du compte utilisateur. Cependant, si l'authentification échoue, le système affiche un message d'erreur à l'utilisateur et l'invite à réexaminer et corriger les champs à remplir.

La figure 18 illustre le diagramme de séquence « Authentification » :



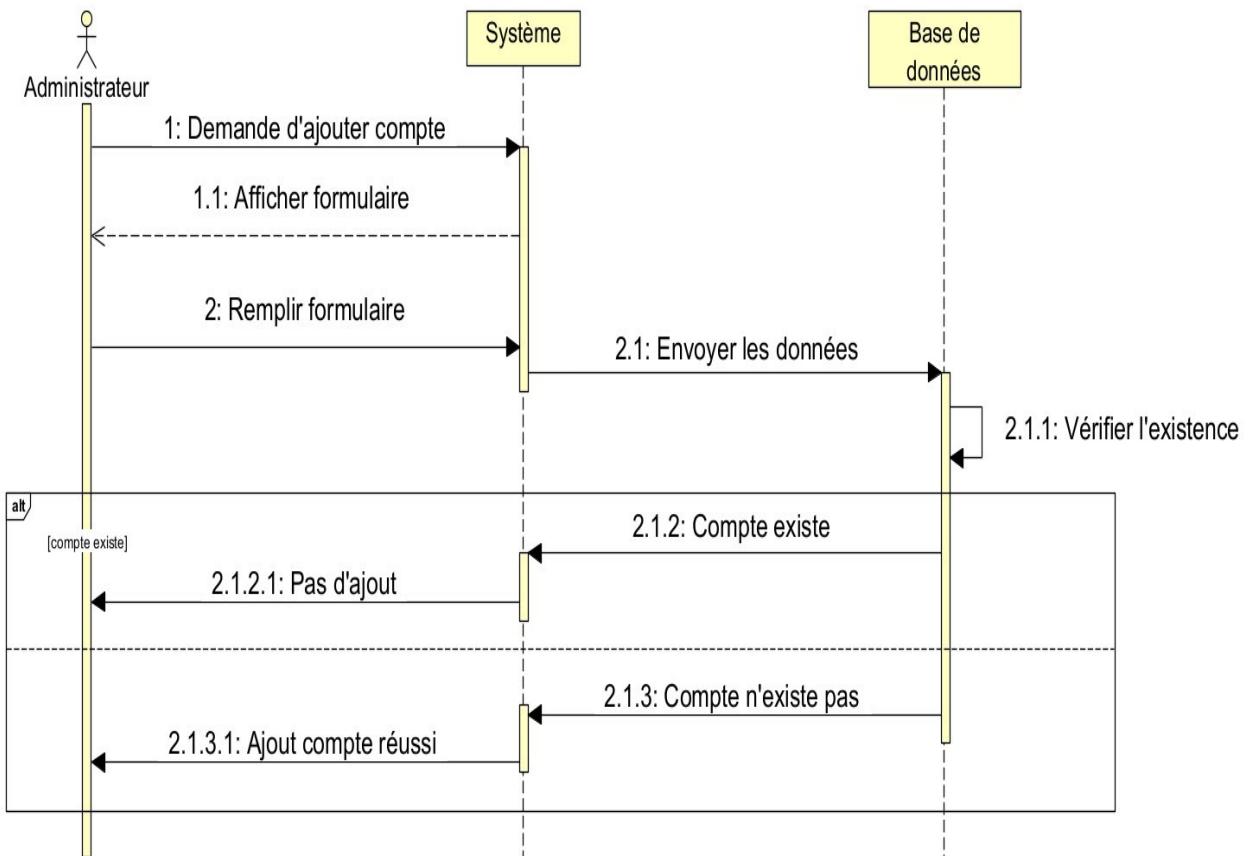
**FIGURE 3.7 – Diagramme de séquence « Authentification »**

- **Diagramme de séquence « Ajouter compte »**

1. Tout d’abord, l’administrateur doit s’authentifier au système en fournissant son identifiant et son mot de passe.
2. Une fois authentifié avec succès, l’administrateur accède à la fonctionnalité pour ajouter un superviseur.
3. L’administrateur remplit le formulaire d’ajout de compte superviseur en saisissant les informations requises, telles que le nom, le prénom, l’adresse e-mail, etc.
4. Après avoir rempli le formulaire, le système envoie ces données à la base de données pour vérifier l’existence d’un compte correspondant.
5. La base de données effectue une vérification en recherchant si un compte avec les mêmes informations existe déjà.
6. Si un compte correspondant est trouvé, la base de données informe le système qu’un compte avec ces données existe déjà.
7. Dans ce cas, le système affiche un message indiquant que l’ajout du compte superviseur n’a pas pu être effectué en raison de la duplication des informations.

8. Cependant, si aucun compte correspondant n'est trouvé dans la base de données, le système poursuit en créant avec succès le compte superviseur avec les informations fournies.

La figure 19 illustre le diagramme de séquence « Ajouter compte » :



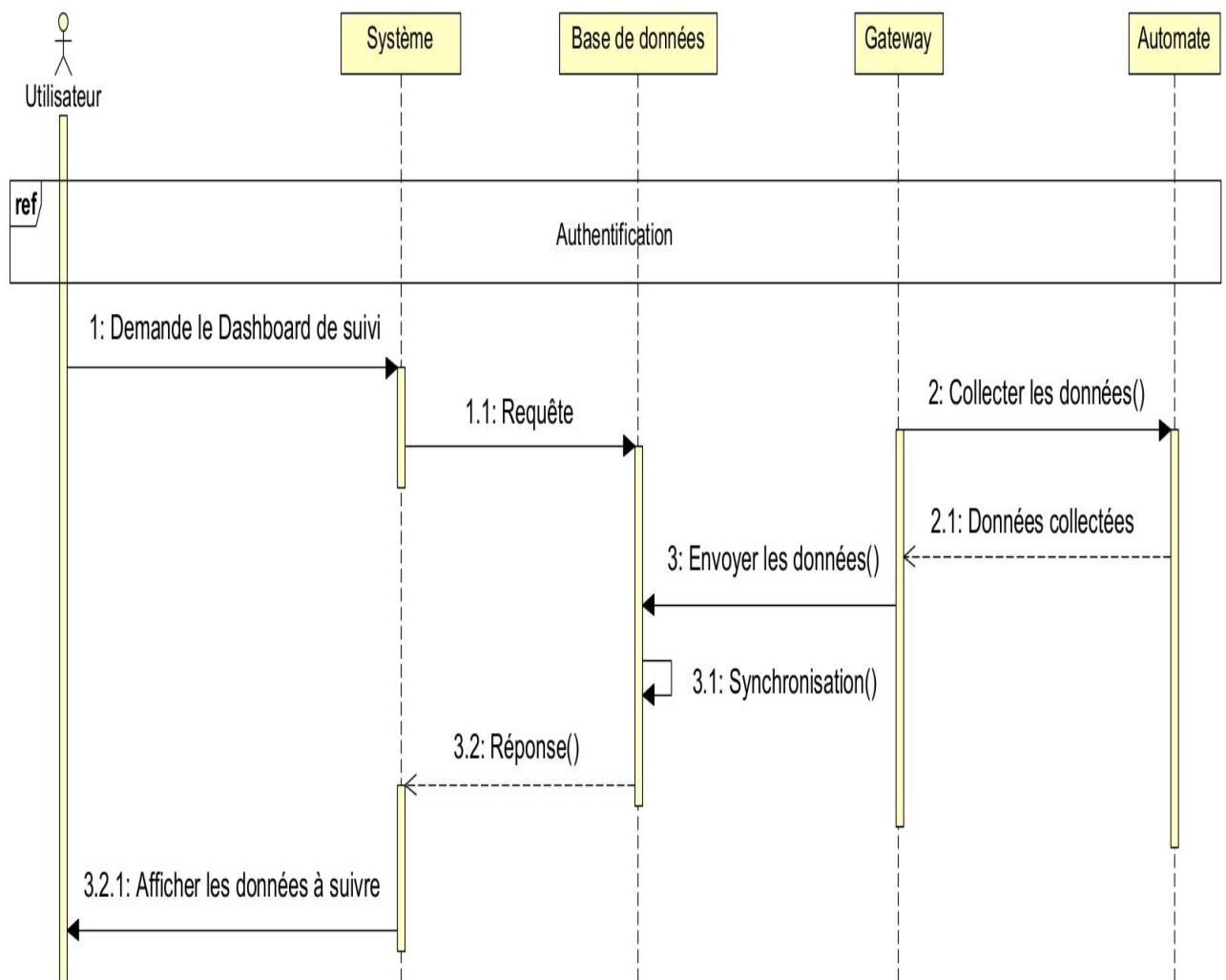
**FIGURE 3.8 – Diagramme de séquence « Ajouter compte »**

- **Diagramme de séquence « Suivre ligne de production »**

1. L'utilisateur, qu'il soit administrateur ou superviseur, doit s'authentifier lorsqu'il accède à l'application.
2. Une fois authentifié avec succès, l'utilisateur demande l'accès au Dashboard pour surveiller les données de production.
3. Lorsque l'utilisateur demande le Dashboard, le système génère une requête vers la base de données pour récupérer les données nécessaires.
4. La base de données est connectée à une passerelle qui collecte les données à partir de l'automate où elles sont stockées.
5. La passerelle effectue l'opération de collecte des données depuis l'automate.

6. Une fois la collecte des données depuis l'automate terminée, la passerelle envoie ces données à la base de données.
7. La base de données synchronise les nouvelles données collectées avec les données existantes.
8. Une fois que la synchronisation est effectuée, la base de données envoie les données mises à jour au système.
9. Le système reçoit les données de la base de données et les affiche à l'utilisateur sur le Dashboard
10. L'utilisateur peut alors visualiser les données demandées, telles que les causes d'arrêts et les indicateurs de performance, directement depuis l'interface du système.

La figure 3.9 illustre le diagramme de séquence « Suivre ligne de production » :



**FIGURE 3.9 – Diagramme de séquence « Suivre ligne de production »**

### 3.4.2 Diagramme de classes

Un diagramme de classes est une représentation visuelle et structurée des interactions entre les classes, les relations et les objets dans un système logiciel ou un modèle conceptuel.

Dans notre cas, le diagramme de classe comprend huit classes,

comme illustré dans la figure 3.10 :

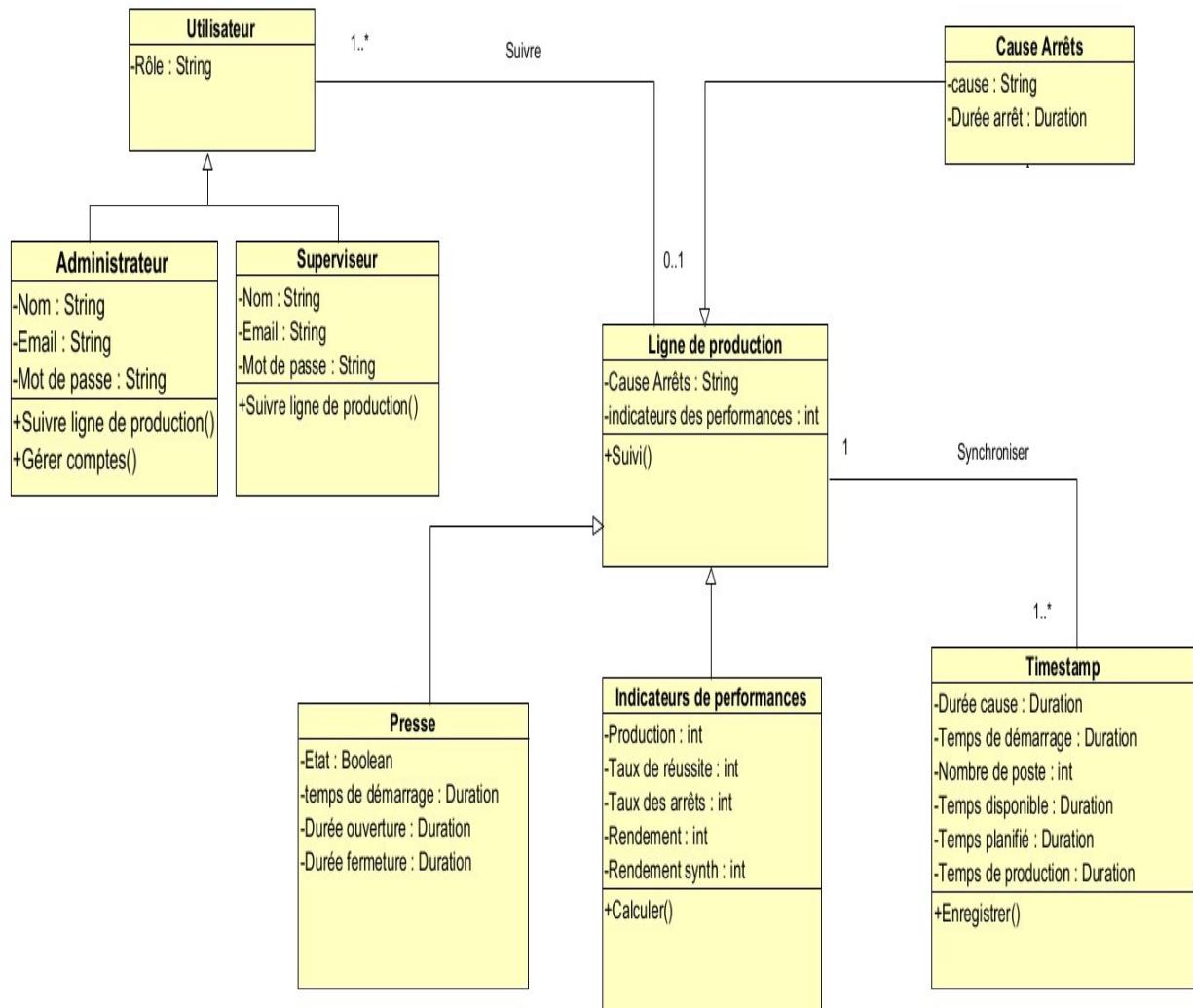


FIGURE 3.10 – Diagramme de classes

Le tableau 3.4 illustre la description des différentes classes de notre application :

Nom de la classe	Description
<b>Utilisateur</b>	Classe regroupant toutes les informations liées aux utilisateurs de notre application.
<b>Administrateur</b>	L'administrateur de l'application, ayant des droits étendus.
<b>Superviseur</b>	Son rôle limiter à suivre la ligne de production.
<b>Ligne de production</b>	Contient les données réelles de production.
<b>Presse</b>	Classe regroupant toutes les informations liées à la presse.
<b>Indicateurs de performance</b>	Les données calculées depuis les performances des machines.
<b>Causes Arrêts</b>	Classe contient les causes identifiées selon les états des capteurs.
<b>Timestamp</b>	Horodatage des évènements.

**TABLE 3.4 – Descriptions des classes**

## 3.5 Conclusion

Ce chapitre a présenté les bases méthodologiques de notre approche et a détaillé les besoins du système, tout en élaborant la conception initiale de l'application mobile. Ces éléments constituent le socle sur lequel nous bâtirons notre solution, garantissant une direction claire pour la suite de notre travail.

Fort de notre compréhension des besoins et de notre conception initiale, nous sommes maintenant prêts à passer à la phase de réalisation de notre projet. Dans le chapitre suivant, nous explorerons en détail la mise en œuvre de notre solution, en utilisant les fondations que nous avons établies dans ce chapitre pour guider nos actions.

---

# Chapitre 4 : REALISATION

## Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>60</b>
<b>4.2</b>	<b>Environnement matériel</b>	<b>60</b>
4.2.1	Architecture générale du nouveau système	60
4.2.2	Dispositifs matériels	63
<b>4.3</b>	<b>Environnement logiciel</b>	<b>65</b>
4.3.1	Outils et Framework de développement	65
4.3.2	Installation, configuration et paramétrage	69
<b>4.4</b>	<b>Réalisation du Dashboard</b>	<b>79</b>
<b>4.5</b>	<b>Conclusion</b>	<b>85</b>

---

## 4.1 Introduction

Dans ce chapitre, nous plongeons dans l'univers concret de l'implémentation, explorant l'environnement logiciel et matériel qui soutient notre système de suivi de production.

Ensuite, nous expliquerons en détail comment nous avons mis en œuvre une méthode de liaison du notre logiciel qui permet la lecture en temps réel des données depuis l'automate Siemens, tout en assurant la collecte et la transmission sécurisée de ces données vers le Dashboard sur le cloud.

## 4.2 Environnement matériel

Nous commencerons par explorer l'environnement matériel que nous avons soigneusement sélectionné pour répondre à nos besoins spécifiques dans l'environnement industriel exigeant de l'usine MPBS.

### 4.2.1 Architecture générale du nouveau système

#### 1. Scénario 1 : Infrastructure à base de cloud IoT

Dans ce premier scénario, nous explorerons l'option initiale que nous avons envisagée pour l'infrastructure de notre système de suivi de production : une passerelle IoT et une infrastructure cloud. Cette vision promettait une série d'avantages uniques qui méritent d'être examinés en détail.

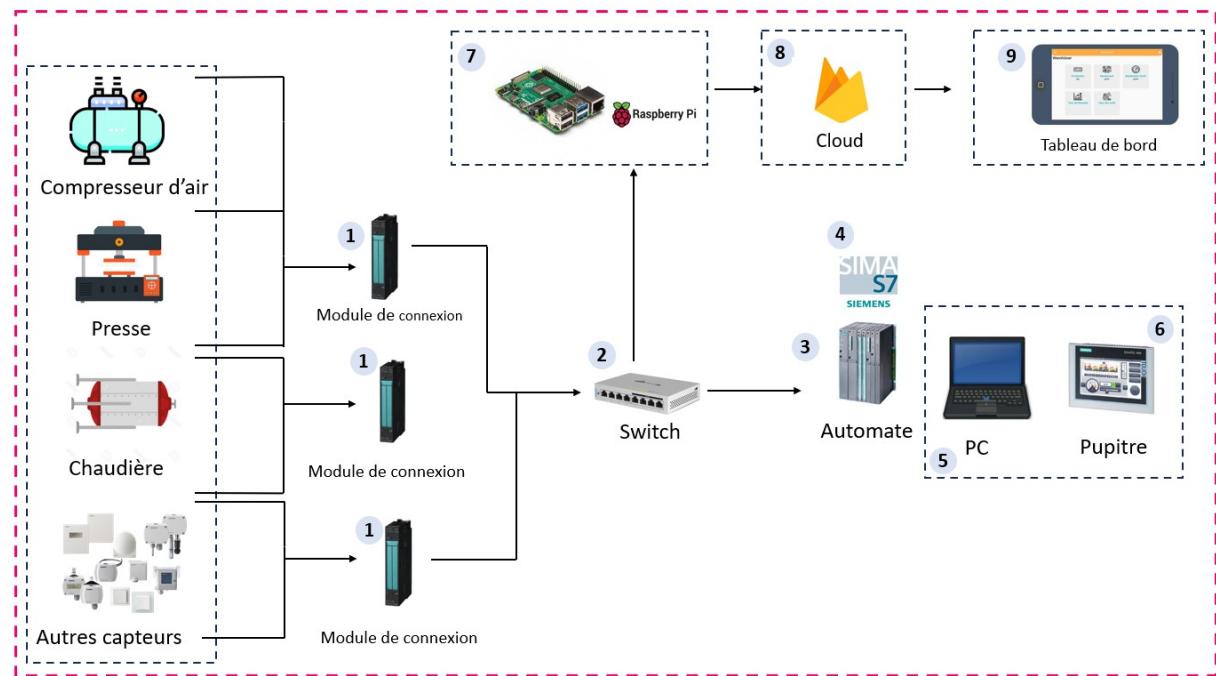


FIGURE 4.1 – Architecture générale à base de Cloud IoT

### • Description

Nous allons décrire le cycle de vie d'un signal dans l'industrie : un signal est émis par un capteur analogique ou digital, INPUT ou OUTPUT, et détecté par des modules de connexion (1), chaque signal est interprété dans l'automate s7 400 (3), pour faire le traitement demandé selon un besoin.

La passerelle Raspberry Pi 4 (7) permet la liaison entre l'automate et le cloud Microsoft Azure moyennant le script python, plus précisément ce dernier :

- Collecte les valeurs des signaux
- Traiter les valeurs collectées pour :
  - Déetecter les causes d'arrêts de la machine
  - Calculer les indicateurs de performances
- Ce script peut également détecter à chaque fois qu'une valeur sur l'automate est changée en temps réel.
- Le cloud envoie automatiquement des messages de modification des variables en aval à chaque changement dans sa base de données au tableau du bord (9).

Les valeurs modifiées sont affichées instantanément sur l'écran du Dashboard. Au début de notre projet, nous avons envisagé une architecture basée sur le cloud, offrant les avantages suivants :

**Évolutivité Inégalée** : L'utilisation du cloud nous aurait permis de faire évoluer notre système à une échelle d'évolutivité illimitée pour répondre aux besoins futurs de l'entreprise.

**Accès Global** : La solution cloud aurait offert un accès mondial aux données, favorisant ainsi la collaboration et la surveillance à distance, de n'importe où

**Gestion Simplifiée** : Avec un hébergement géré, la gestion de l'infrastructure aurait été considérablement simplifiée.

## 2. Scénario 2 : Infrastructure à base d'un serveur local

Dans ce deuxième scénario, nous explorerons l'alternative que la société cliente a finalement choisie : l'infrastructure basée sur un serveur local. Ce choix stratégique découle de considérations spécifiques liées à leur environnement et à leurs besoins.

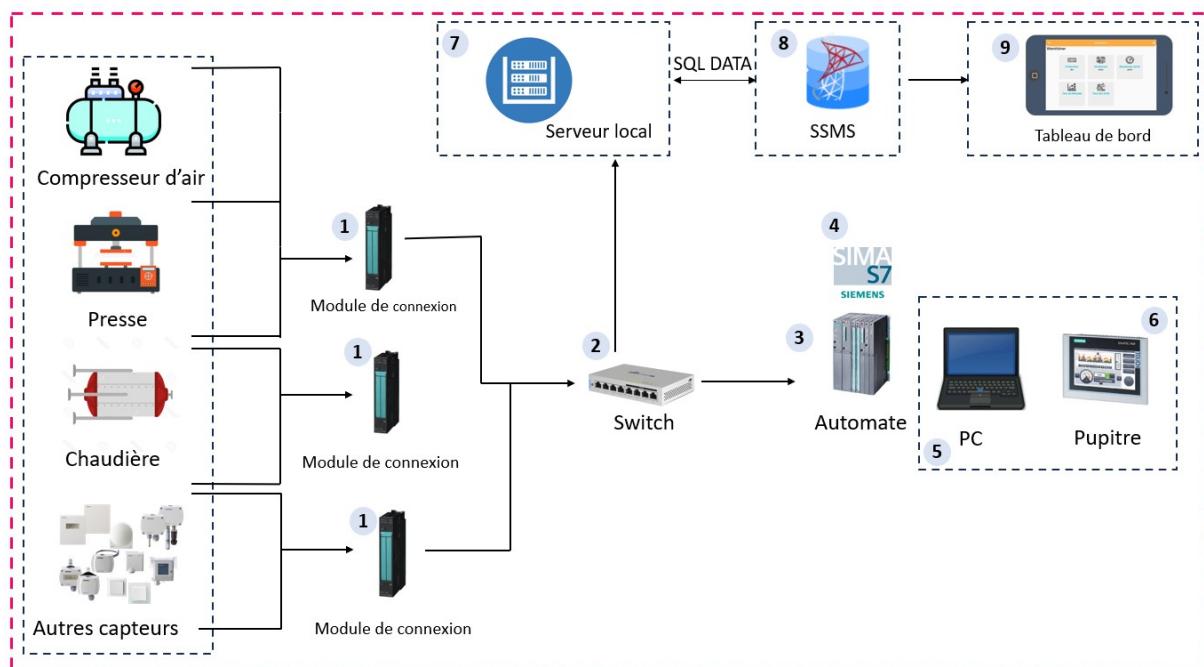


FIGURE 4.2 – Architecture générale à base d'un serveur Local

Suite à des discussions approfondies, notre client a opté pour une infrastructure basée sur un serveur local, motivée par :

**Confidentialité des Données :** La nature sensible de leurs données nécessitait une solution offrant un niveau élevé de confidentialité et de contrôle.

**Latence Minimale :** Des opérations en temps réel nécessitaient une latence minimale, ce qui a été facilité par une infrastructure locale.

**Contrôle Total :** La préférence pour une gestion interne complète de l'infrastructure a conduit à ce choix.

- **Description**

Pour le serveur local (7) : on a configuré sa connexion avec l'automate (3) puis créer notre base de données SQL avec SQL Server Management Studio (8) qui stockera les données collectées depuis l'automate. Concevez la structure de la base de données pour stocker les informations pertinentes, telles que les états des capteurs, les événements de production, et les indicateurs de performance.

### 4.2.2 Dispositifs matériels

#### 1. Scénario 1 : Solution Cloud

Pour l'option initiale que nous avons envisagée de notre système. On dispose d'une carte Raspberry Pi 4, d'un automate Siemens S7 400, d'un ordinateur portable, d'un commutateur et d'un smartphone.

- **Automate S7 400 :**

Nous avons utilisé un automate de type siemens S7 400 pour détecter et interpréter un signal qui est émis par un capteur analogique ou digital, INPUT ou OUTPUT.



**FIGURE 4.3 – Automate Siemens S7-400**

- **Rock Pi 4 :**

Est un nano-ordinateur monocarte basé sur une architecture ARM. Elle est équipée d'un processeur Rockchip RK3399, qui comprend un CPU hexacœur composé de deux coeurs Cortex-A72 haute performance et quatre coeurs Cortex-A53 économies.



**FIGURE 4.4 – Rock PI 4**

- **Commutateur :**

Nous avons utilisé un commutateur Ethernet pour connecter l'automate à la carte Raspberry Pi 4.

- **Ordinateur.**

- **Smartphone :**

Pour simuler et tester notre application.

### 2. Scénario 2 : Solution Serveur local

Durant la réalisation de notre travail, nous avons utilisé des équipements matériels. Au lieu d'utiliser un cloud nous avons choisi un serveur local lié à un commutateur.

- **Serveur local :**

Est utilisé comme composant central de l'infrastructure.

Le serveur local est généralement un ordinateur ou un dispositif spécialement configuré pour répondre aux besoins de l'installation industrielle. Le serveur local permet d'avoir un contrôle direct sur les données et les opérations au sein de notre installation industrielle. Cela signifie que nous pouvons prendre des décisions en temps réel et réagir aux situations d'urgence sans dépendre d'une connexion Internet ou cloud.

- **Commutateur :**

Nous avons utilisé un commutateur Ethernet pour connecter le serveur local à l'automate siemens.

### 4.3 Environnement logiciel

Dans cette partie on va détailler les logiciels, les langages de programmation et les outils que nous avons utilisé dans les deux solutions pour développer, gérer et surveiller notre système intelligent de suivi de production.

#### 4.3.1 Outils et Framework de développement

- **Outils Communs :**

##### **SIMATIC Manager (STEP 7)**

Est un puissant outil logiciel développé par Siemens pour la programmation et la configuration des automates Siemens S7-300 et S7-400. Il s'agit de la principale plate-forme pour la conception, la configuration et la maintenance des projets d'automatisation.

SIMATIC Manager offre un environnement complet pour la création de programmes d'automates, la configuration des composants matériels, la gestion des protocoles de communication réseau tels que PROFIBUS et l'organisation des blocs de programme.

Il reste indispensable pour les anciens modèles d'automates et offre une approche structurée de l'ingénierie de l'automatisation industrielle.[6]



FIGURE 4.5 – Logo SIMATIC Manager STEP 7

**Python :**

Python est un langage de programmation de haut niveau, interprété, orienté objet, et largement utilisé pour le développement d'applications logicielles dans divers domaines.



FIGURE 4.6 – Logo Python

**Flutter :**

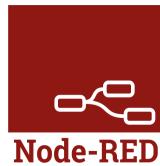
C'est un Framework de développement d'applications mobiles qui permet de créer des applications multiplateformes, ce qui signifie que vous pouvez développer une seule base de code source et la déployer sur plusieurs plateformes différentes, notamment Android, iOS, le web et même le bureau.



FIGURE 4.7 – Logo Flutter

### **Node-RED :**

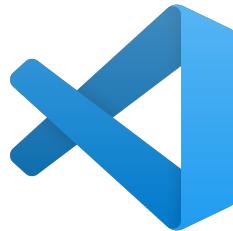
Node-RED est un outil de développement basé sur les flux développés à l'origine par IBM pour le câblage de périphériques matériels, d' API et de services en ligne dans le cadre de l' Internet des objets.[7]



**FIGURE 4.8 – Logo Node-Red**

### **Visual Studio Code :**

C'est un éditeur de code extensible développé par Microsoft pour Windows, Linux et OS X.



**FIGURE 4.9 – Logo Visual Studio Code**

### **• Outils Spécifiques au Scénario 1 : Solution à base de cloud IoT**

#### **Firebase :**

Firebase est une plateforme mobile de Google qui facilite la création de back-end à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications pour mobile et pour le web.[8]



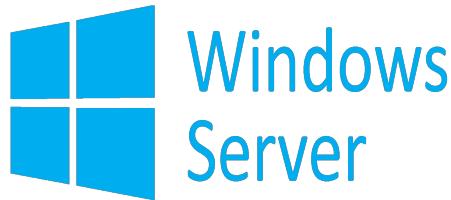
**FIGURE 4.10 – Logo Firebase**

- **Outils Spécifiques au Scénario 2 : Solution à base d'un serveur local**

Comme a été mentionné précédemment au lieu d'utiliser un service cloud le choix s'est porté sur l'utilisation d'un serveur local :

**Serveur Windows :**

C'est un ordinateur ou un système équipé d'une version du système d'exploitation Windows conçue spécifiquement pour les besoins de serveurs. Les versions courantes de serveurs Windows incluent Windows Server 2019, Windows Server 2016, Windows Server 2012, et ainsi de suite.



**FIGURE 4.11 – Logo Serveur Windows**

**SQL Server Management Studio (SSMS) :**

C'est un environnement intégré permettant de gérer toute infrastructure SQL. Utilisez SSMS pour accéder, configurer, gérer, administrer et développer tous les composants de SQL Server, Azure SQL Database, Azure SQL Managed Instance, SQL Server sur Azure VM et Azure Synapse Analytics.



**FIGURE 4.12 – Logo SQL Server Management**

### 4.3.2 Installation, configuration et paramétrage

Dans cette section, nous allons explorer les aspects essentiels de l'installation, du paramétrage et de la configuration de divers systèmes, logiciels et équipements, en mettant l'accent sur les étapes clés nécessaires pour les mettre en service et les adapter à nos besoins spécifiques ,en examinant les deux différentes solutions.

#### 1. Installation de la bibliothèque Snap7 Python

C'est une bibliothèque Python qui permet de communiquer avec des automates programmables industriels (API) de la série Siemens S7-200, S7-300 et S7-400 via le protocole de communication S7 (aussi appelé ISO-on-TCP).

Ce package est principalement utilisé dans l'automatisation industrielle et l'intégration de systèmes. Il offre une interface Python pour interagir avec les automates Siemens S7, ce qui permet de lire et d'écrire des données dans les registres de ces automates, de réaliser des opérations de lecture/écriture dans des bases de données PLC, et de gérer les connexions aux contrôleurs S7.

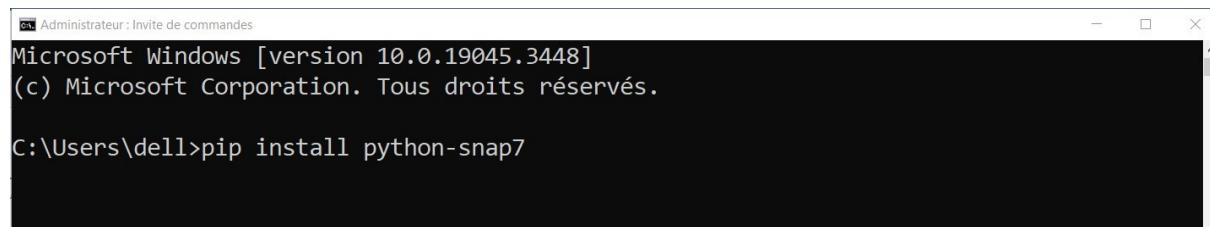


FIGURE 4.13 – Installation du bibliothèque Snap7

Une fois que le package Snap7 est installé, on va l'importer dans notre scripts Python pour l'utiliser dans notre application d'automatisation industrielle.

```
from snap7 import client
```

Pour établir une connexion à l'automate il faut disposer les droits d'accès nécessaires, notamment l'adresse IP de l'automate, le numéro de slot et le numéro de rack.

Voici un exemple simple d'importation de Snap7 et d'utilisation pour établir une connexion à un automate Siemens S7-400 :

```
from snap7 import client

# Créez un client Snap7
plc = client.Client()

# Établissez une connexion à l'automate Siemens S7
plc.connect('192.168.1.1', 0, 2)

# Lire un mot de la base de données DB2, adresse %MW100
data = plc.read_area(0x84, 2, 100, 2)
print(data)

# Fermez la connexion avec l'automate
plc.disconnect()
```

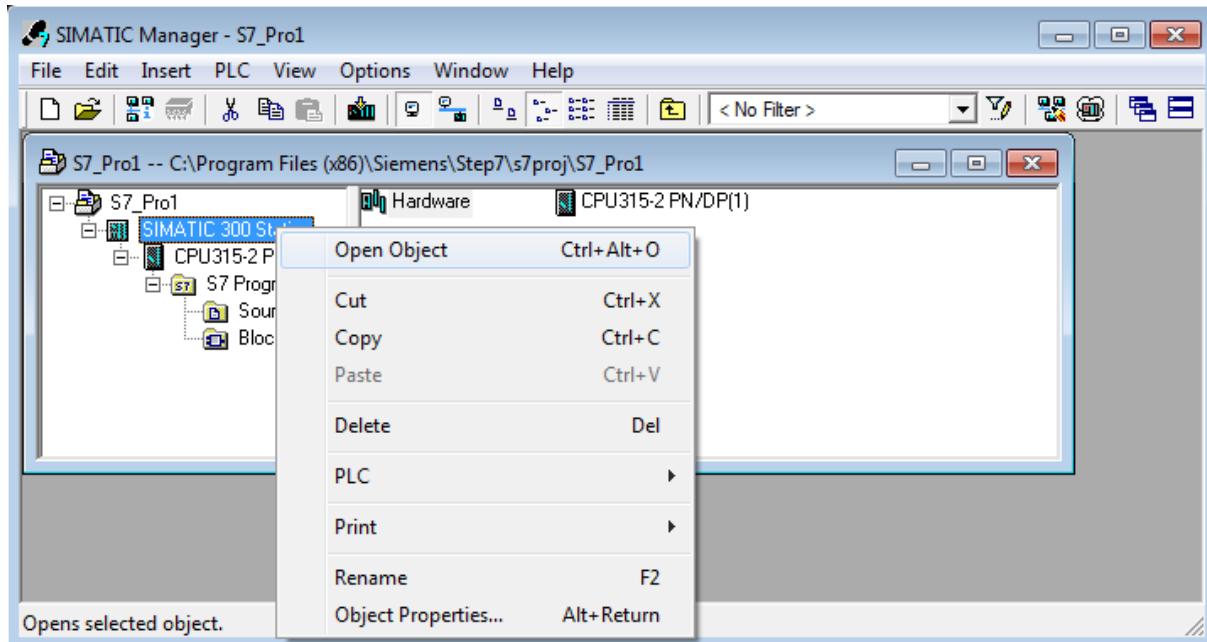
**FIGURE 4.14 – Etablissement d'une connexion au PLC**

### 2. SIMATIC Manager STEP 7

Un logiciel déjà installé qui renferme le programme de l'automate.

L'objectif était d'acquérir une compréhension approfondie des capteurs cibles, en examinant leur fonctionnement, les adresses dans les différents blocs de données , ainsi que leur taille.

À travers l'utilisation de cet outil, j'ai pu naviguer efficacement à travers les paramètres des capteurs, identifiant les adresses pertinentes dans les blocs associés et comprenant comment les informations étaient organisées. De plus, j'ai pris le temps de comprendre comment convertir ces données de manière adéquate pour obtenir une information claire sur les capteurs.



**FIGURE 4.15 – Interface graphique de SIMATIC Manager STEP 7**

Pour obtenir une lecture précise des capteurs, j'ai suivi une démarche méthodique.

- Tout d'abord, j'ai suivi attentivement l'enchaînement de la machine de stratification sur site, identifiant ainsi les capteurs cibles nécessaires à la collecte d'informations pertinentes.
- Ensuite, je me suis plongé dans Simatic Manager, explorant les différentes sections pour localiser les adresses associées à ces capteurs dans les blocs appropriés.

The screenshot shows the SIMATIC Manager editor window for a data block named 'DB1'. The title bar indicates the file type is 'LAD/STL/FBD' and the database is 'DB1'. The main area displays a table of data blocks with columns for Address, Name, Type, Initial value, Actual value, and Comment. The table includes rows for various variables like 'motor\_on\_PB', 'motor\_OFF\_PB', 'MOTOR\_START', etc., and a row for 'string\_store' which is highlighted with a green background. The status bar at the bottom shows 'RUN' and other operational indicators.

Address	Name	Type	Initial value	Actual value	Comment
0.0	motor_on_PB	BOOL	FALSE	FALSE	
0.1	motor_OFF_PB	BOOL	FALSE	FALSE	
0.2	MOTOR_START	BOOL	FALSE	FALSE	
2.0	current	REAL	0.000000e+000	0.0	
6.0	statusofmotor	BOOL	FALSE	FALSE	
8.0	motornormalcurrent	INT	0	0	
10.0	worddatatype	WORD	W#16#0	W#16#0000	
12.0	double_word	DWORD	DW#16#0	DW#16#00000000	
16.0	doubleint	DINT	L#0	L#0	
20.0	string_store	STRING[254]	''	'plc'	
276.0	string_input	STRING[254]	''	'plc'	
532.0	DB_VAR	INT	0	0	Temporary I

**FIGURE 4.16 – Blocs de données siemens**

- Une fois les adresses identifiées, j'ai entrepris de rechercher la syntaxe spécifique dans Simatic Manager qui permettrait de lire les informations souhaitées. Cette étape m'a demandé une compréhension approfondie des différents blocs et de la logique de programmation utilisée dans le système automatisé.

voici un exemple de code qui illustre comment lire la valeur de la température du plateau inférieur de chaudière à partir de PLC :

```
plateau_inferieur = plc.db_read(12, 16, 2)
plateau_inferieur = struct.unpack('>h', plateau_inferieur)[0]

print('plateau inférieur du chaudière',plateau_inferieur)
```

**FIGURE 4.17 – Lecture de données de type entier depuis PLC**

La méthode db\_read est utilisée pour lire des données depuis un bloc de données spécifique du PLC.

Les arguments passés à db\_read sont probablement les suivants :

- 12 : Le numéro du bloc de données.
- 16 : L'offset (décalage) à partir duquel la lecture doit commencer.
- 2 : Le nombre d'octets à lire.

```
plateau_inferieur = struct.unpack('>h', plateau_inferieur)[0]
```

Cette ligne de code utilise le module **struct** pour décompresser (unpack) les données binaires lues à partir du PLC.

Un autre exemple on va extraire des données booléennes d'un octet situé dans la zone d'entrée de processus (PE) du PLC.

```
pe_data6 = plc.read_area(Areas.PE, 0, 53, 1)
inputs["E53.0"] = snap7.util.get_bool(pe_data6, 0, 0)
inputs["E53.1"] = snap7.util.get_bool(pe_data6, 0, 1)
inputs["E53.2"] = snap7.util.get_bool(pe_data6, 0, 2)
inputs["E53.3"] = snap7.util.get_bool(pe_data6, 0, 3)
```

**FIGURE 4.18 – Lecture de données de type booléennes depuis PLC**

```
pe_data6 = plc.read_area(Areas.PE, 0, 53, 1)
```

Cette ligne lit 1 octet de données dans la zone Process Input (PE) de l'automate.

Les données sont stockées dans la variable pe\_data6. Les paramètres spécifient la zone (PE), le décalage de départ (0), le nombre d'octets à lire (1).

```
inputs["E53.0"] = snap7.util.get_bool(pe_data6, 0, 0)
```

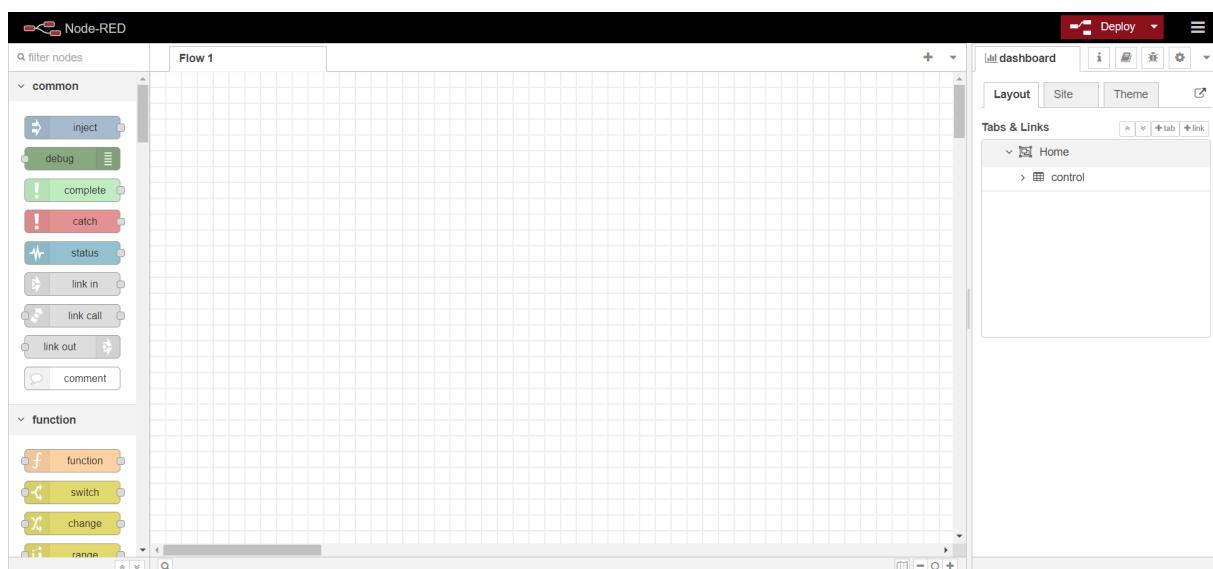
Le code utilise la fonction get\_bool de la bibliothèque Snap7 pour extraire une valeur booléenne du premier bit (bit 0) de l'octet stocké dans pe\_data6.

Cette valeur booléenne est ensuite affectée à l'entrée intitulée "E53.0" dans un dictionnaire nommé inputs.

### 3. Node-RED comme logiciel de visualisation

Node-RED nous a permis de créer un tableau de bord interactif qui affiche les états des capteurs de manière intuitive. Grâce à son interface visuelle conviviale, nous avons pu configurer des nœuds de surveillance des capteurs, des déclencheurs d'événements en cas de changement d'état, et des indicateurs visuels pour une compréhension rapide de l'état de la ligne de production.

Pour ouvrir directement l'interface Node-RED sur le Pi, nous ouvrons le navigateur internet en saisissant : <http://localhost:1880/>

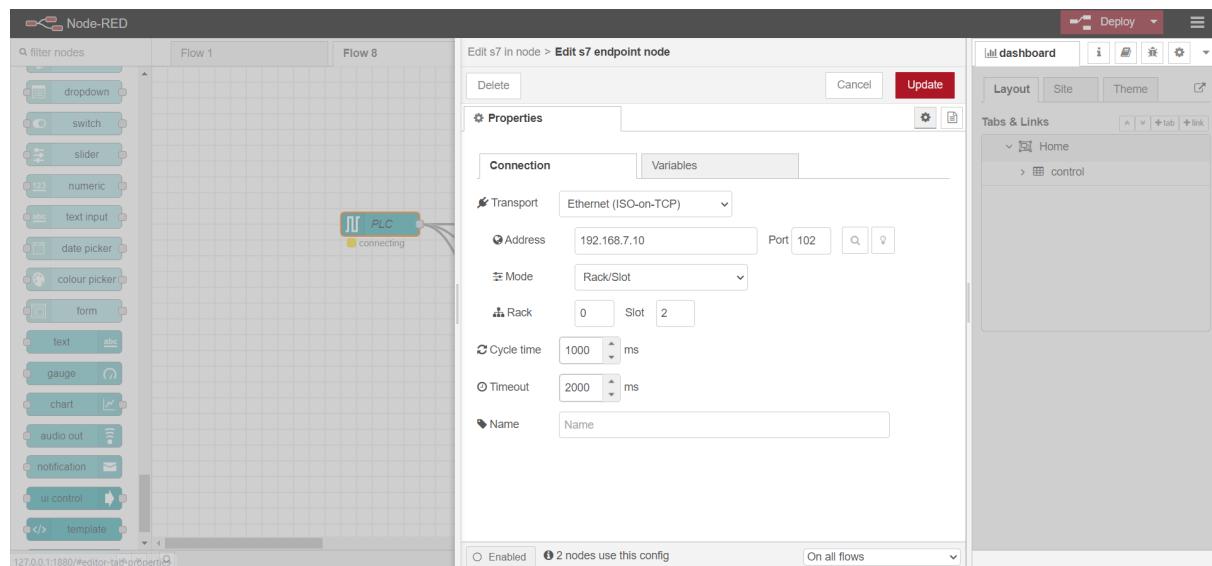


**FIGURE 4.19 – Interface graphique de Node-RED.**

Pour configurer la connexion à un automate Siemens dans Node-RED, suivez ces étapes :

- **Adresse IP de l'automate :** Tout d'abord, identifiez l'adresse IP de votre automate Siemens. Cela peut être obtenu à partir des paramètres réseau de l'automate ou en utilisant des outils tels que STEP 7 de Siemens.
- **Nombre de slots et de racks :** Déterminez le nombre de slots et de racks de votre automate. Ces informations décrivent la configuration matérielle de votre automate Siemens et sont essentielles pour une communication précise.
- **Type de connexion à l'automate :** Choisissez le type de connexion approprié pour votre automate.

Node-RED prend en charge différents protocoles de communication avec les automates Siemens, tels que S7Comm pour les automates Siemens S7. Assurez-vous de sélectionner le bon protocole en fonction de votre automate spécifique.



**FIGURE 4.20 – configuration Node-RED.**

Chaque connexion à un automate est représentée par le nœud de configuration du nœud final S7.

On a alors cette architecture :

## CHAPITRE 4 : REALISATION

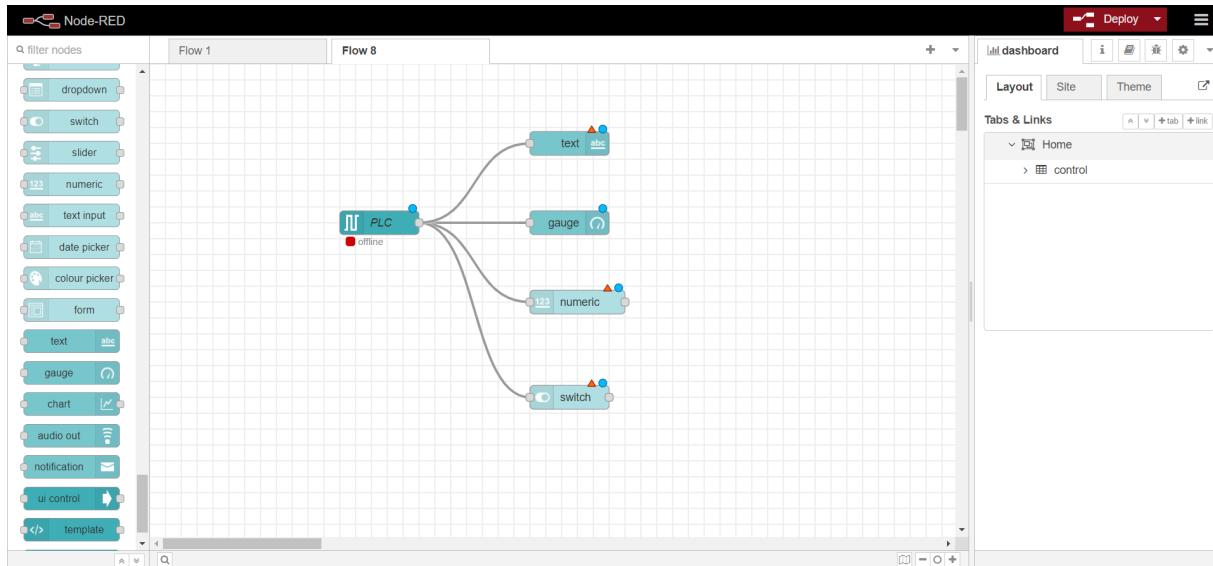


FIGURE 4.21 – Création des noeuds sur Node-RED.

Après avoir ajouté les noeuds à notre flow Node-RED, la prochaine étape consiste à les configurer en fournissant les paramètres nécessaires pour établir la communication avec les capteurs, actionneurs, ou autres périphériques que nous souhaiter intégrer.

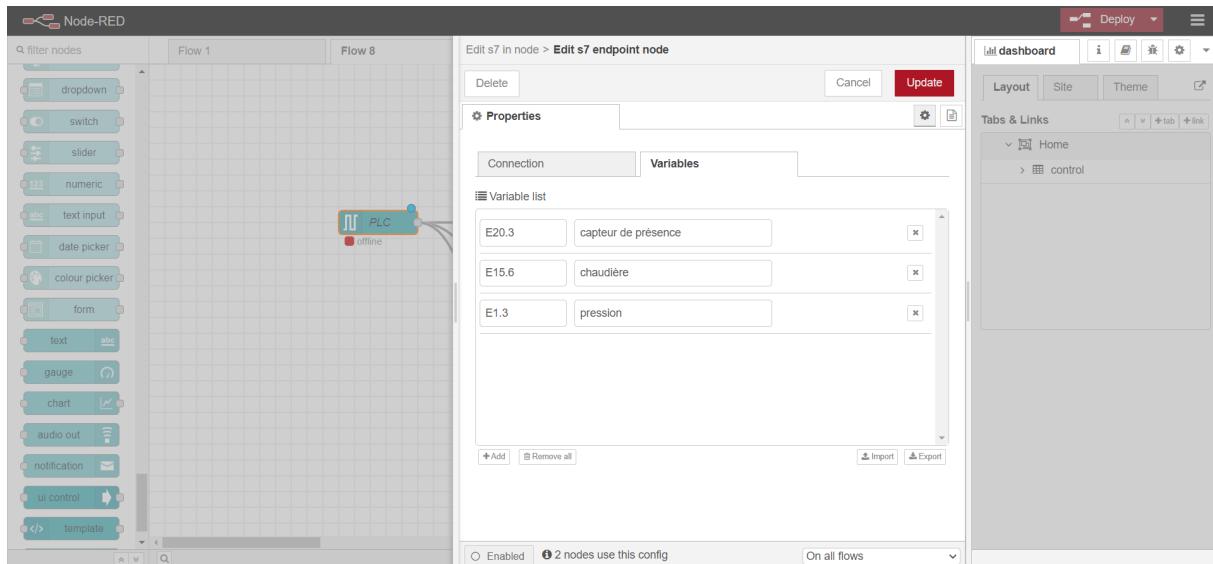


FIGURE 4.22 – Configuration des noeuds sur Node-RED.

### 4. Passerelle IoT Raspberry Pi 4

Le Raspberry Pi est initialement sans système d’exploitation. D’abord, nous avons téléchargé Raspbian à partir du site officiel : [raspberrypi.org/downloads](http://raspberrypi.org/downloads).

Il faut donc installer le système d’exploitation (SE du Raspberry Pi à l’aide de Raspberry Pi Imager.



**FIGURE 4.23 – Installer le SE du Raspberry Pi à l'aide de Raspberry Pi Imager.**

### 5. Installer le logiciel de lecture sur la passerelle IoT Raspberry Pi 4

Une fois que nous avons configuré la carte Raspberry Pi avec son système d’exploitation, activé SSH, et préparé notre code python, on peut le transférer et le faire fonctionner sur la carte.

On va transférer notre fichier Python depuis notre ordinateur vers le Raspberry Pi en utilisant des commandes SCP (Secure Copy) :

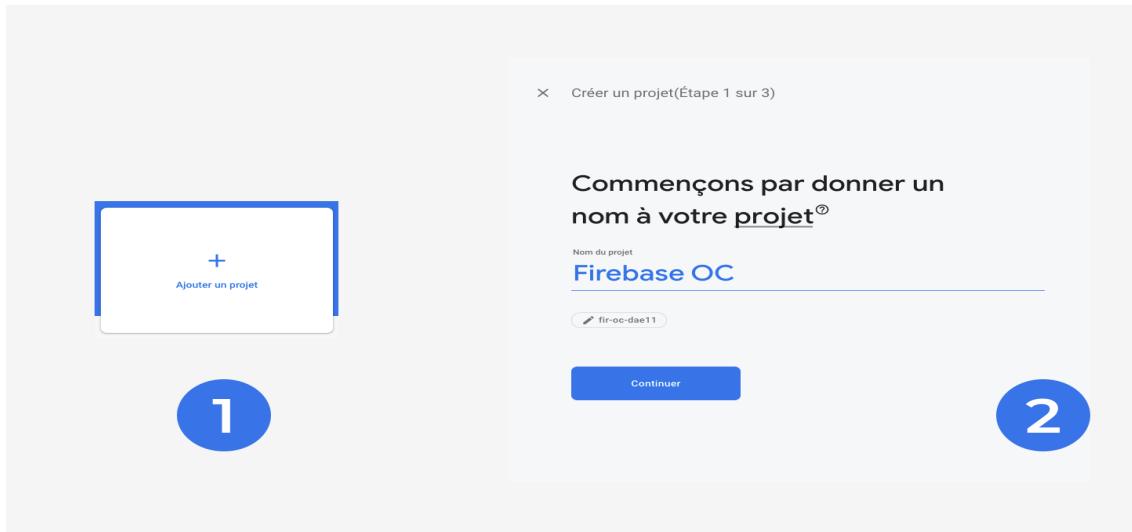
```
sudo systemctl enable ssh
```

### 6. Cloud : Firebase

- **Création d'un projet Firebase**

Pour intégrer Firebase dans notre script Python et stocker les données lues depuis l’automate, la première étape essentielle consiste à créer un nouveau projet Firebase.

Cette opération s’effectue à travers l’interface Firebase Console, où un nouvel espace de travail sera configuré pour héberger notre application.



**FIGURE 4.24 – Crédit d'un projet dans Firebase.**

- **Configuration des paramètres de connexion avec Firebase**

Une fois le projet créé, l'accès aux paramètres du projet dans Firebase Console nous permettra d'obtenir les coordonnées nécessaires, notamment la clé d'identification du projet et l'URL de la base de données en temps réel. Ces informations cruciales serviront à établir la connexion entre notre script Python et Firebase, permettant ainsi le stockage efficace des données provenant de l'automate programmable.

```
# Firebase configuration
config = {
    "apiKey": "AIzaSyBJyO9x5bR-IPX9pJIBBJd_DcFIIImI1r54",
    "authDomain": "https://accounts.google.com/o/oauth2/auth",
    "projectId": "master-2421f",
    "databaseURL": "https://master-2421f-default.firebaseio.com",
    "storageBucket": "https://console.firebaseio.google.com/project/master-2421f/storage",
    "messagingSenderId": "478601383277",
    "appId": "1:478601383277:android:e3e6a231808c44d5ef5e64",
}
```

**FIGURE 4.25 – Configuration des paramètres de connexion avec Firebase.**

- **Structure de la base de données**

Les données provenant de l'automate programmable sont transmises vers Firebase Realtime Database toutes les minutes, créant ainsi une structure organisée pour une gestion efficace. Chaque transmission forme un nouveau nœud dans la base de données en temps réel, contenant les données spécifiques collectées à ce moment-là.

Cette approche permet de créer une chronologie des informations, facilitant l'accès aux données historiques pour une analyse ultérieure.



The screenshot shows a browser window displaying the Firebase Realtime Database at the URL <https://master-2421f-default.firebaseio.com>. A message at the top indicates "Read-only and non-realtime mode activated in the data viewer to improve browser performance. Select a key with fewer records to edit or view in realtime". The main area displays a hierarchical tree structure of data under a node labeled "390". The data is as follows:

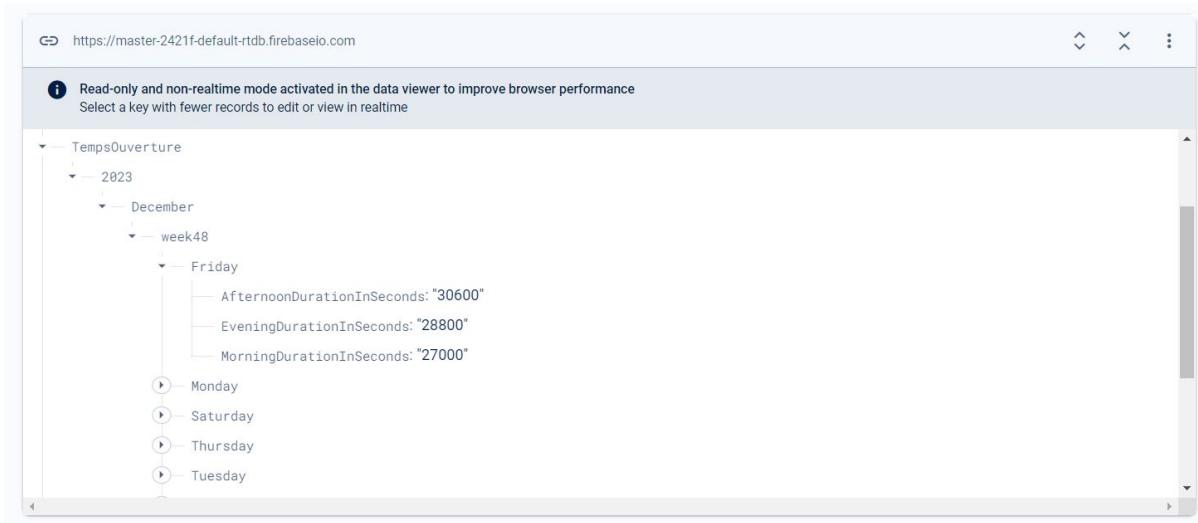
```
lower_plate2: "173"
production_duration: "1.357361078262329"
stoppage_duration: "519.1588535308838"
temperature_chaud: "253"
temperature_inf_a_modifier: "172"
temperature_sup_a_modifier: "175"
timestamp: "2023-08-08 12:22:40.260152"
upper_plate1: "175"
upper_plate2: "176"
```

**FIGURE 4.26 – Fenêtre de la structure de la base de données.**

Chaque nœud est identifié par un horodatage, créant une séquence temporelle des données. À l'intérieur de chaque nœud, la structure reste la même, avec des catégories spécifiques et des valeurs de capteurs. Cette approche de stockage dans Firebase facilite le suivi chronologique des données, offrant ainsi une vision historique des conditions du système automatisé.

- **Traitement de données collectées**

Les données collectées depuis le PLC, sont utilisées pour déterminer des indicateurs de performance clés. Prenons l'exemple du temps d'ouverture de la presse mesuré par un capteur de l'automate. Ces données sont ensuite archivées dans Firebase Realtime Database de manière structurée pour une analyse historique approfondie.



**FIGURE 4.27 – Analyse et stockage des données.**

Les données du temps d’ouverture sont archivées par année, mois, semaine et jour.

Chaque jour contient les temps d’ouverture spécifiques pour chaque poste, et cela peut être ajusté en fonction des changements de poste au fil du temps. Cette organisation permet une analyse détaillée des performances au fil des années, des mois, des semaines et des jours, tout en restant flexible pour accommoder les variations possibles dans les postes de travail.

## 4.4 Réalisation du Dashboard

Le tableau de bord (Dashboard) est un élément central dans notre projet de suivi de production. Il agit comme la fenêtre sur l’état en temps réel de notre ligne de production, fournissant des informations déterminantes sur les indicateurs de performance, les causes d’arrêt et d’autres données essentielles.

L’authentification des utilisateurs et la gestion des comptes sont des éléments fondamentaux de notre système, permettant un accès sécurisé et personnalisé à l’interface de suivi de production. Dans cette section, nous allons explorer comment nous avons mis en place cette fonctionnalité cruciale.

L'administration de notre système dispose de la capacité de créer des comptes pour les superviseurs industriels. Ces comptes permettent aux superviseurs d'accéder à l'interface de suivi de production avec des autorisations spécifiques.

### 1. Interface d'ajout d'un superviseur

L'ajout des superviseurs est une action disponible pour l'Administrateur.

The screenshot shows a mobile-style application interface titled 'Create Account'. At the top left is a back arrow icon, and at the top right is a large blue circular button with a white right-pointing arrow. The main area has a light blue background. It contains three input fields: 'Name' (placeholder text), 'Email' (placeholder text), and 'Password' (placeholder text). Below these fields is a large blue button labeled 'Sign Up' in white. To the right of this button is a smaller blue circular button with a white right-pointing arrow. At the bottom left of the screen is a link labeled 'Sign In'.

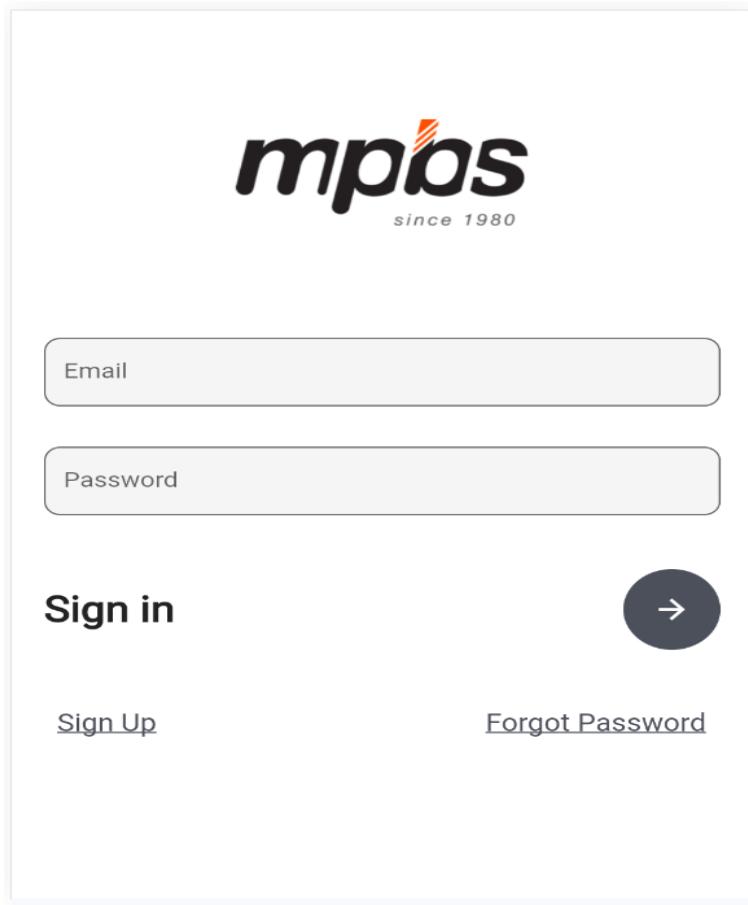
**FIGURE 4.28 – Interface d'ajout d'un superviseur**

Le processus de création de compte consiste à saisir les données suivantes :

- Nom
- Email
- Mot de passe

### 2. Interface d'inscription

Une fois que les comptes des superviseurs sont créés, ces derniers peuvent s'authentifier en utilisant leur adresse e-mail et leur mot de passe. Cette étape de connexion garantit que seuls les utilisateurs autorisés ont accès aux données sensibles de la production.

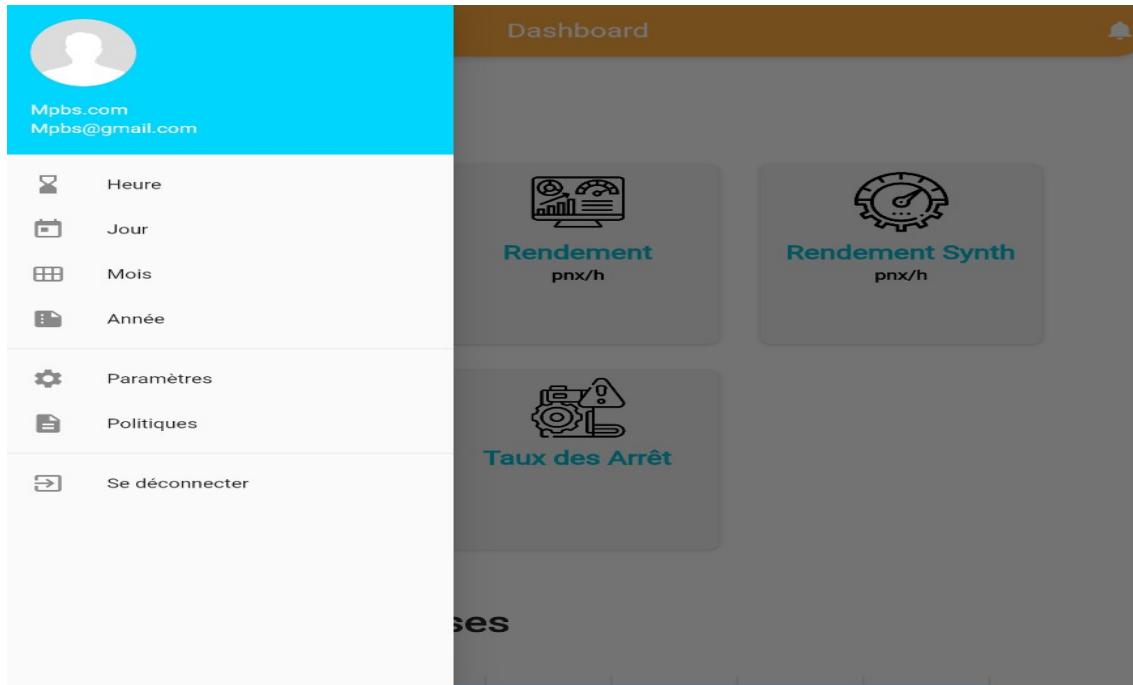


**FIGURE 4.29 – Interface de connexion**

### 3. Barre de navigation

Après une authentification réussie, l'utilisateur est automatiquement redirigé vers le tableau de bord principal de notre système. Ce tableau de bord offre une vue en temps réel de la ligne de production, affichant les indicateurs de performance, les causes d'arrêt, les graphiques et les statistiques essentielles pour une surveillance efficace.

Pour la barre de navigation, les superviseurs ont la possibilité de sélectionner la période de temps sur laquelle ils souhaitent effectuer un suivi des chiffres. Cette fonctionnalité permet de visualiser les données en temps réel ou d'effectuer des analyses rétrospectives sur des périodes spécifiques, y compris par heure, jour, semaine, mois et année.



**FIGURE 4.30 – Barre de navigation**

#### 4. Indicateurs de performances

Les indicateurs de performances cruciaux ont été dérivés avec précision à partir des données acquises par nos capteurs sophistiqués. Ces données, capturées en temps réel, ont subi un traitement minutieux, aligné sur les horaires de travail qui peuvent être ajustés en fonction des conditions changeantes de la manufacture.

Ces indicateurs reflètent non seulement la productivité brute, mais aussi la capacité d'adaptation de notre système aux variations des horaires de travail. La flexibilité intégrée dans notre approche permet une gestion dynamique des performances, garantissant une efficacité maximale même dans des conditions opérationnelles changeantes.

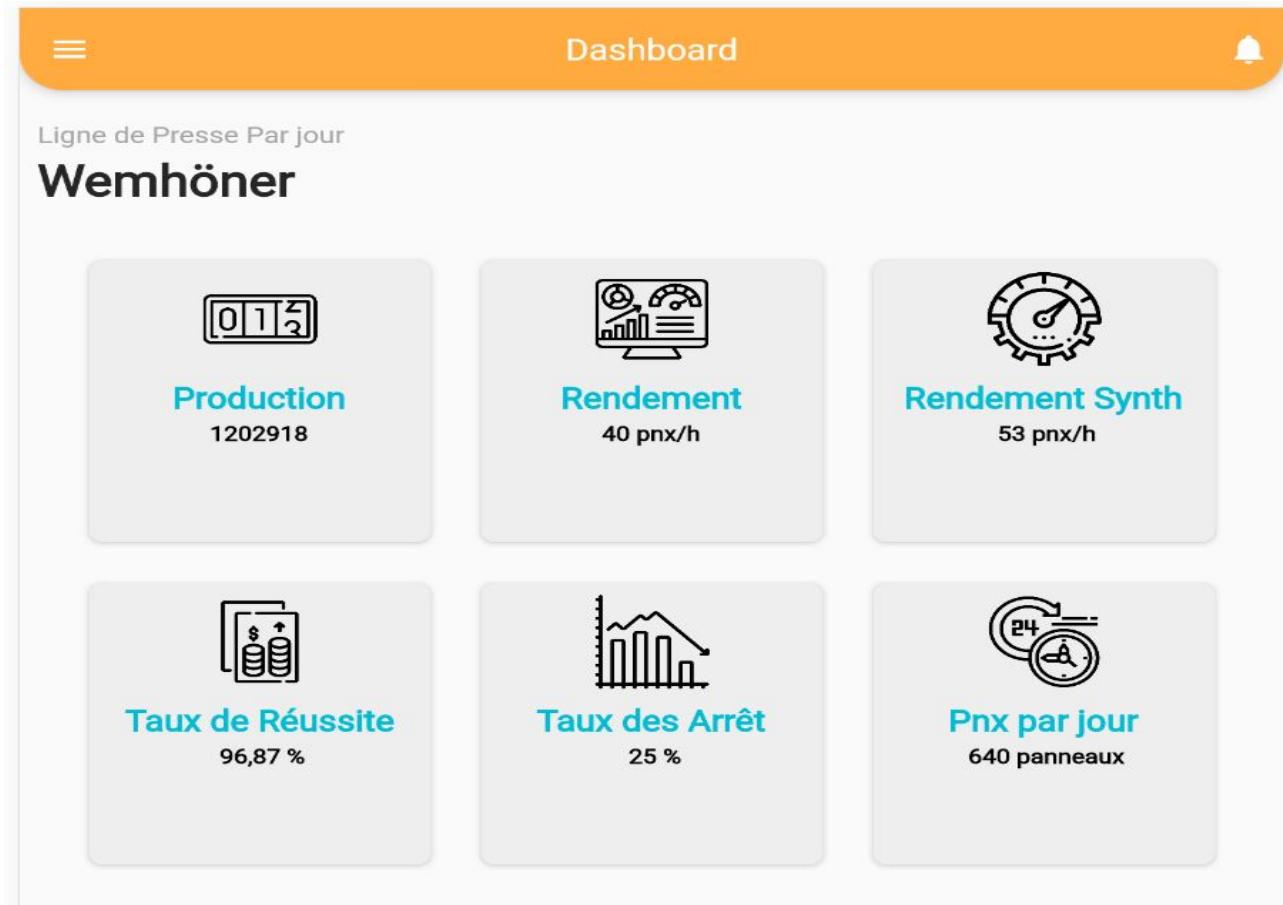


FIGURE 4.31 – Les indicateurs de performances.

### 5. Identification des causes d’arrêts

L’identification des causes d’arrêts revêt une importance capitale, car elle permet une analyse détaillée de chaque arrêt en calculant son pourcentage par rapport aux autres facteurs, le tout contextualisé par rapport au temps d’ouverture global.

Cette approche analytique offre une compréhension approfondie des motifs d’interruption, permettant ainsi de hiérarchiser les causes en fonction de leur impact sur la disponibilité opérationnelle.

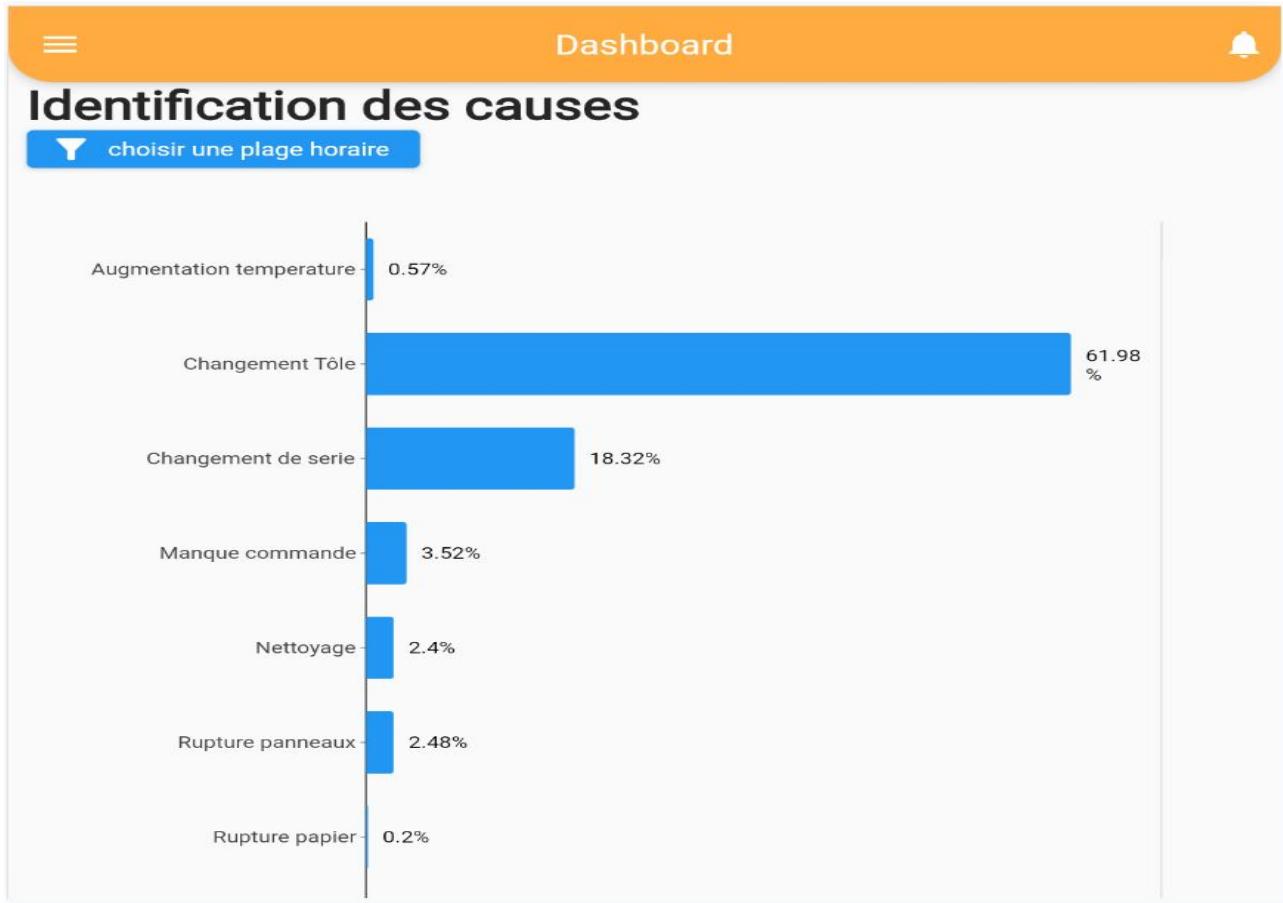
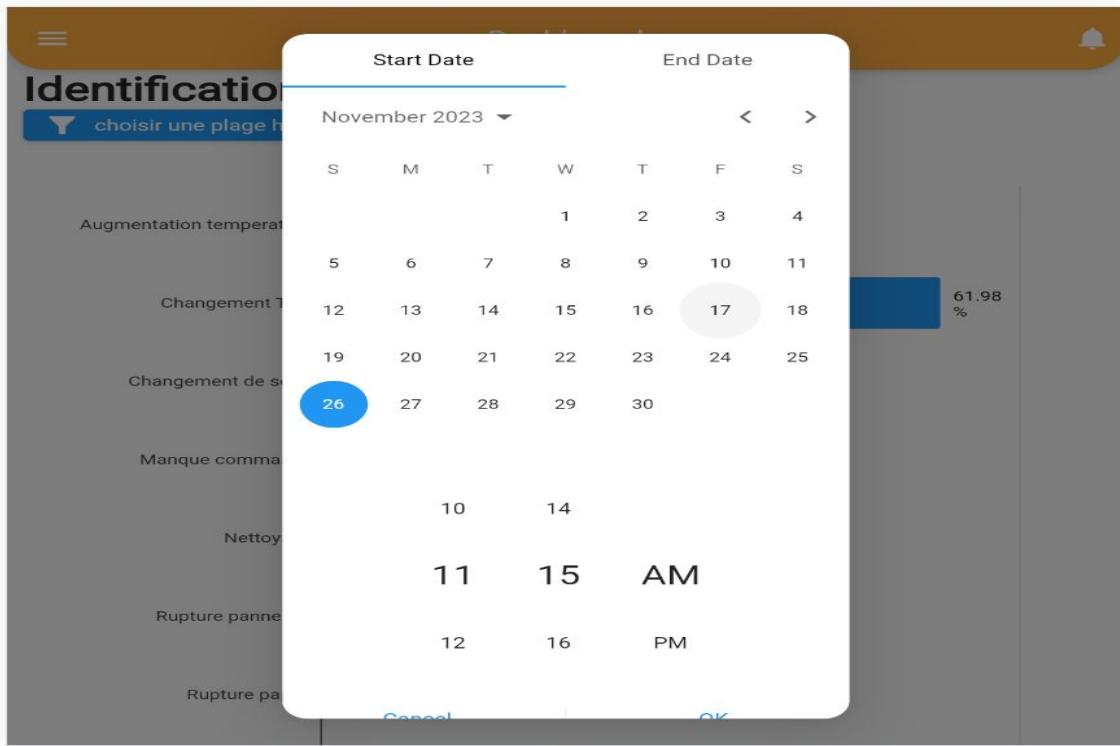


FIGURE 4.32 – Identification des cause d’arrêts.

#### 6. Changement de plage horaire pour les pourcentages d’arrêts

Nous avons ajouté aussi un calendrier visuel permettant de naviguer à travers les jours, les semaines et les mois. Permettez aux superviseurs de sélectionner une date en cliquant sur le temps correspondant.



**FIGURE 4.33 – Sélecteur de plage de dates.**

## 4.5 Conclusion

Ce chapitre a marqué l'aboutissement de nos efforts pour mettre en œuvre une solution complète de surveillance de la ligne de production. À travers ce chapitre, nous avons examiné en détail les éléments essentiels qui ont contribué à la concrétisation de notre vision.

Deux scénarios de déploiement, un basé sur un serveur local et l'autre sur un cloud IoT, ont été envisagés pour répondre aux besoins variés de la MPBS. Chaque scénario a été soigneusement conçu et mis en place, en tenant compte des spécificités de l'environnement matériel et logiciel.

# CONCLUSION ET PERSPECTIVES

Au cœur de la Quatrième Révolution Industrielle, la supervision industrielle émerge comme le pilier essentiel de la transformation numérique des processus manufacturiers. L'étude approfondie du processus de fabrication de panneaux stratifiés et du fonctionnement de la machine de stratification au sein de la Manufacture de Panneaux de Bois de Sfax a jeté les bases de notre projet novateur.

La mise en œuvre réussie d'un système IoT dédié à la lecture et à l'interprétation des données des capteurs des machines, couplée à la création d'un tableau de bord intuitif, a permis d'élever la supervision industrielle à de nouveaux sommets. En fournissant des indicateurs de performances en temps réel et en identifiant les causes potentielles d'arrêts, notre solution a catalysé une gestion proactive des opérations.

Ainsi, la supervision industrielle, désormais alimentée par des données en temps réel et des analyses avancées, devient le moteur de la prise de décision éclairée. Cette symbiose entre la supervision industrielle et la transformation numérique positionne la Manufacture de Panneaux de Bois de Sfax comme un pionnier de l'Industrie 4.0, ouvrant la voie à une gestion plus efficiente et proactive de ses opérations manufacturières.

Les perspectives d'amélioration sont également prometteuses. L'intégration prévue pour l'identification des opérateurs offre une traçabilité accrue, tandis que la modélisation prédictive à l'aide du Machine Learning ouvre la porte à une anticipation proactive des performances.

Cette démarche s'inscrit dans une logique plus vaste d'amélioration continue, où la connectivité des machines, la collecte intelligente de données et l'automatisation convergent pour accroître l'efficacité et la productivité.

## **CONCLUSION ET PERSPECTIVES**

---

La conclusion de notre projet offre une vision élargie des perspectives, envisageant l'intégration future de la couche Enterprise Resource Planning (ERP) pour une gestion holistique des ressources de l'entreprise.

# BIBLIOGRAPHIE

[1] **MOHAMED LAMINE DILMI.** Contribution à la modélisation des systèmes automatisés par un outil graphique.2014.

Disponible sur :

<https://docplayer.fr/86811326-Memoire-de-master-no-ref-mac21-juin-2014-domaine-sciences-et-technologie-filiere-automatique-specialite-commande-des-processus-industriels.html>.consulté le 29 Mars 2023.

[2] **Université Frères Mentouri.** ÉTAT DE L'ART DU GÉNIE ÉLECTRIQUE (GE) [[en ligne](#)]. Mis à jour 2021-01-14. Disponible sur :

[https://www.st.com/content/st\\_com/en/about/st\\_company\\_information/who-we-are.html](https://www.st.com/content/st_com/en/about/st_company_information/who-we-are.html). Format PDF.consulté le 30 Mars 2023.

[3] **SIEMENS.** Gammes siemens automate [[en ligne](#)]. Disponible sur :

<https://www.siemens.com/fr/fr/produits/automatisation-entrainements/systemes-automatisation/industrial/plc.html>.

Consulté le 15 février 2023.

[4] **Mohammed Abdallah,Omar Elkeelany** .A Survey on Data Acquisition Systems DAQ [[en ligne](#)]. Disponible sur :

[https://ieeexplore.ieee.org/abstract/document/5328141?casa\\_token=HMOYyS1fIA0AAAAA:6J32WCOJO7MM78nnj2dpLpQdK3tjYybvK79RykdfkuJ3eqzoD7qOBoubXz77j3fflcVShsn6RQVO.2](https://ieeexplore.ieee.org/abstract/document/5328141?casa_token=HMOYyS1fIA0AAAAA:6J32WCOJO7MM78nnj2dpLpQdK3tjYybvK79RykdfkuJ3eqzoD7qOBoubXz77j3fflcVShsn6RQVO.2) Avril 2009. Consulté le 17 Mai 2023.

[5] **WendBenedo Arnaud** .Application des algorithmes d'apprentissage automatique pour la détection de défauts de roulements sur les machines [[en ligne](#)]. Disponible sur :

## BIBLIOGRAPHIE

---

<https://constellation.uqac.ca/id/eprint/5857/1/Zoungranaauqac0862N10694.pdf>, Mis à jour en 2020. Consulté le 17 Mai 2023.

- [6] **Redouane Ait Ali Yahia.** An Introduction to Siemens SIMATIC Manager (STEP7).

[https://www.solisplc.com/tutorials/an-introduction-to-siemens-simatic-manager-step-7#:~:text=SIMATIC%20Manager%20\(Also%20known%20as,configuring%2C%20and%20maintaining%20automation%20projects..](https://www.solisplc.com/tutorials/an-introduction-to-siemens-simatic-manager-step-7#:~:text=SIMATIC%20Manager%20(Also%20known%20as,configuring%2C%20and%20maintaining%20automation%20projects..) consulté le 20 Mars 2023.

- [7] **Node-Red** .Disponible sur : <https://nodered.org/>

- [8] **Firebase**.Disonible sur : <https://junto.fr/blog/firebase/>

# ANNEXES

## Industrie 4.0

### Définitions de l'industrie 4.0 :

L'industrie 4.0 est un sujet d'actualité impliquant les sociétés contemporaines et les organisations et faisant l'objet des recherches dans le monde universitaire et industriel (Kagermann et al., 2013). Le terme « Industrie 4.0 » a été utilisé, pour la première fois, en 2011 dans le plan d'action stratégique allemande 2020 portant sur les nouvelles technologies (Pilloni, 2018). Bien que les écrits faisant consensus sur les principes de ladite industrie 4.0 sont nombreux, il est difficile de trouver une définition du concept « Industrie 4.0» qui fait consensus. La transdisciplinarité du concept, traduite par le vif intérêt accordé audit concept, conduit à l'émergence d'une diversité terminologique telle que « industrie future », « industrie numérique », « industrie intelligente », « internet industriel» ou « transformation numérique» (Cayrat, 2018) . C'est ainsi qu'en 2013, BITCOM, l'association des télécommunications allemandes a trouvé plus de 100 définitions du concept de l'industrie 4.0 (Bidet-Mayer, 2016). Cependant, afin de mieux cerner le sujet et limiter l'impact du changement auquel les individus, les organisations et les sociétés peuvent faire face, il est essentiel de ne citer que les définitions les plus importantes.

### Historique des révolutions industrielles :

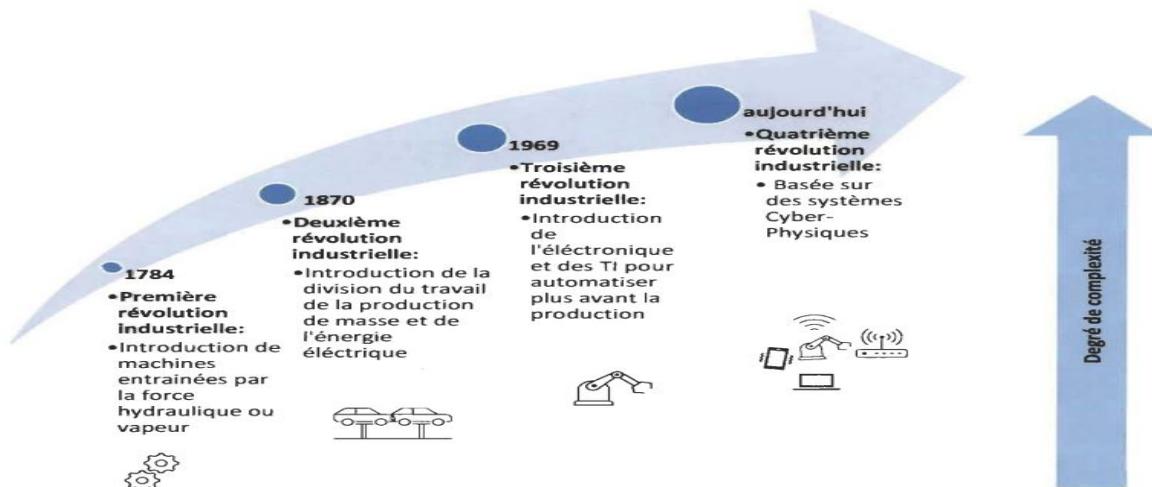


Figure 4: Les quatre révolutions industrielles. Source, (Ministère de l'Économie (MESI)), 2016).

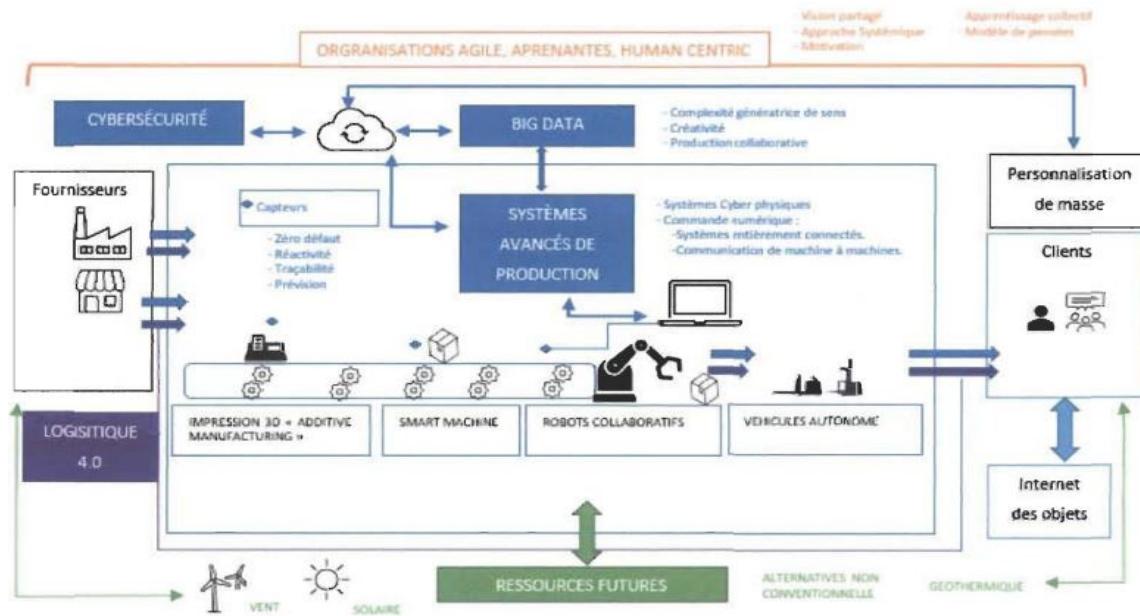


Figure 8 : Écosystème de l'industrie 4.0. Source, (Gimélec, 2014a)

### Ligne de presse Wemhöner

Wemhöner - fabrique des installations pour la finition des matériaux à base de bois et est le leader technologique mondial dans ce domaine. Wemhöner développe et teste en permanence de nouvelles technologies en étroite collaboration avec un vaste réseau d'universités, de fournisseurs et de clients leaders dans sa propre installation pilote.

Une partie importante de la gamme de produits Wemhöner est constituée de presses à cycle court pour le revêtement direct de panneaux de particules ou de panneaux MDF. Ils permettent la technologie de pore synchrone simple et double face ou la technologie EIR (gaufrage en repérage / gaufrage en repérage). La ligne de produits pour le revêtement de pièces tridimensionnelles s'appelle Variopress. Sous le nom de Masterline, Wemhöner propose des systèmes d'impression et de vernissage modulaires pour le vernissage liquide. Enfin, le portefeuille comprend également des presses continues à un ou plusieurs étages, des systèmes de panneaux légers pour la fabrication de portes et de panneaux ainsi que des presses spéciales et des systèmes spéciaux pour les fabricants de diverses industries.

## ANNEXES

En plus de l'usine principale de Herford, Wemhöner fabrique également en Chine depuis 2007. Les clients ont accès à un réseau de partenaires commerciaux pour le conseil, la vente et le service après-vente.



Les presses haute pression à cycle court de Wemhöner jouissent d'une réputation mondiale

# **PROJET DE DIGITALISATION POUR LA SOCIETE MPBS**

---

---

**Noura Fitouri**

---

---

## **Résumé :**

Le présent rapport s'inscrit dans le cadre de l'accomplissement de stage de fin d'études « Projet de Digitalisation pour la Société MPBS » en vue de l'obtention du diplôme du mastère DSIR « Développement des Systèmes Informatiques et Réseaux » à l'Institut Supérieur des Etudes Technologiques de Sfax.

L'objectif principal est la collecte, le traitement et l'analyse des données en provenance de divers capteurs pour instaurer une surveillance en temps réel de la machine de stratification pour identifier les causes d'arrêts et fournir des indicateurs de performance importants.

**Mots clés :** Industrie 4.0, Digitalisation, Internet des Objets (IoT), Machine de Stratification, Capteurs, Tableau de Bord.

## **Abstract :**

This report is part of the completion of the end-of-studies internship « Digitalization project for MPBS » for the DSIR master's degree « Computer and Network Systems Development » at the Higher Institute of Technological Studies, Sfax.

The main objective is to collect, process and analyze data from various sensors to set up real-time monitoring of the stratification machine to identify the causes of stoppages and provide important performance indicators.

**Key-words :** Industry 4.0, Digitalization, Internet of Things (IoT), Laminating machine, Sensors, Dashboard.