**Projet de Fin d’Etudes´**

**Filière : Génie Informatique**

**Présente par :**

**Boudar Mehdi**

**Digitalisation du FabLab : Conception et Réalisation d’une Plateforme Web**

**Encadré par :**

|  |  |  |
| --- | --- | --- |
| Pr. | EMSI Rabat | Encadrant |
| Mme. Eddib Meryem | Orange Digital Center **Soutenu**  **devant :** | Encadrante Externe |
| Pr. Abdelilah | Emsi Rabat | Membre du Jury |
| Pr. [Nom Jury 2] | Emsi Rabat | Membre du Jury |

**Annee Universitaire : 2024/2025**

# Resume

Dans un contexte ou` l’innovation numérique joue un rôle clé dans l’optimisation des processus, ce projet s’inscrit dans une d´démarche de digitalisation du FabLab de l’Orange Digital Center (ODC). L’objectif principal est de concevoir et d´développer une plateforme web moderne et performante, facilitant la gestion des projets, du matériel et des ressources du FabLab.

Cette solution intègre plusieurs fonctionnalités essentielles, notamment la gestion des projets, la réservation du matériel ainsi qu’un syst`eme avance de gestion des stocks et des inventaires. Un module d’impression automatisé a ´été mis en place, utilisant un Raspberry Pi pour assurer la liaison entre la plateforme et les imprimantes 3D/2D du FabLab. Cela permet aux utilisateurs de soumettre leurs fichiers directement depuis l’interface web, tout en offrant un suivi en temps réel de l’´état d’impression.

Par ailleurs, un chabot basé sur l’intelligence artificielle a été développe pour accompagner les utilisateurs en leur fournissant des réponses instantan´ees et adapt´ees `à leurs besoins. Son intégration vise `a am´eliorer l’accessibilité aux informations et `a réduire le besoin d’une assistance humaine constante.

Le d´développement repose sur une architecture robuste et ´évolutive, combinant Spring Boot pour le backend et Angular pour le frontend. Ce choix garantit une performance optimale, une expérience utilisateur fluide et une s´écrite renforcée des donn´ees. La base de données relationnelle assure un stockage structure et sécurisé, facilitant l’accès et la gestion des différentes ressources.

Ce projet a ´également permis d’explorer diverses pratiques de d´développement logiciel, incluant les tests unitaires et d’intégration, l’optimisation des performances ainsi que l’adoption de m´méthodologies agiles comme Scrum, garantissant un suivi efficace et itératif du d´eveloppement.

En conclusion, cette initiative contribue `a la transformation digitale du FabLab en apportant une solution centralisée et interconnect´ee. Grace `a cette plateforme, l’ODC bénéficie d’un environnement plus structuré, favorisant l’innovation, la collaboration et une meilleure gestion des ressources techniques et humaines.

# Table des matières

**Table des matières 2**

**Liste des figures 4**

**Introduction Générale 5**

1. **Contexte General 6**
   1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   2. Présentation de l’Organisme d’Accueil . . . . . . . . . . . . . . . . . . . . 6
      1. Orange Maroc . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      2. Les Clients d’Orange Maroc . . . . . . . . . . . . . . . . . . . . . . 8
      3. Evolution des services´ . . . . . . . . . . . . . . . . . . . . . . . . . 9
      4. Orange Digital Center (ODC) . . . . . . . . . . . . . . . . . . . . . 9
   3. Cadre du Projet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
      1. Domaine du Projet . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
      2. Problématique . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
      3. Objectifs du projet . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
   4. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2. **Analyse et Spécification 13**
   1. Etude Pr´ealable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .´ 13
      1. Etude de l’existant . . . . . . . . . . . . . . . . . . . . . . . . . . .´ 13
      2. Critique de l’existant . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   2. Analyse des besoins . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
      1. Identification des acteurs . . . . . . . . . . . . . . . . . . . . . . . . 15
      2. Besoins du projet . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
   3. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
3. **Etude Technique´ 18**
   1. Architecture Microservices . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
      1. D´efinition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
      2. Avantages des Microservices . . . . . . . . . . . . . . . . . . . . . . 18
   2. Architecture de l’application . . . . . . . . . . . . . . . . . . . . . . . . . . 20

TABLE DES MATIERES` TABLE DES MATIERES`

* 1. Architecture interne d’un microservice . . . . . . . . . . . . . . . . . . . . 20
     1. Architecture 3-tiers . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
     2. Communication entre les microservices . . . . . . . . . . . . . . . . 21
  2. Int´egration avec Raspberry Pi (IoT) . . . . . . . . . . . . . . . . . . . . . . 22
  3. Frameworks, biblioth`eques et API utilis´es . . . . . . . . . . . . . . . . . . . 24
     1. Spring Boot . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
     2. Spring Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
     3. Swagger . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
     4. Rest API . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
     5. Google Gemini via Google API . . . . . . . . . . . . . . . . . . . . 27
     6. Python . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
     7. Node JS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
     8. Lombok . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
     9. MySQL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
     10. Angular . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
  4. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

**Webographie 31**

# Liste des figures

1.1 Orange Maroc . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

1.2 Organigramme Orange Maroc . . . . . . . . . . . . . . . . . . . . . . . . . 7

1.3 Graphique des évolutions des services d’Orange Maroc . . . . . . . . . . . 9

3.1 Raspberry Pi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

# Introduction G´en´erale

L’essor des technologies num´eriques a profond´ement transform´e la mani`ere dont les entreprises et les organisations g`erent leurs ressources et leurs processus. Dans ce contexte de transformation digitale, les FabLabs jouent un roˆle cl´e en offrant aux innovateurs et aux cr´eateurs un espace collaboratif ´equip´e de machines et d’outils de fabrication modernes. Cependant, leur gestion repose encore souvent sur des m´ethodes traditionnelles qui limitent leur efficacit´e et leur accessibilit´e.

Le FabLab de l’Orange Digital Center (ODC) ne fait pas exception a` cette probl´ematique. Actuellement, plusieurs taˆches essentielles comme la gestion des projets, le suivi des stocks de mat´eriel ou encore la r´eservation des ´equipements sont r´ealis´ees manuellement ou via des syst`emes partiellement informatis´es. Ces limites entraˆınent un manque d’organisation, des pertes de temps et une faible tra¸cabilit´e des ressources mises a` disposition.

Pour r´epondre `a ces enjeux, ce projet propose la conception et le d´eveloppement d’une plateforme web d´edi´ee a` la digitalisation du FabLab de l’ODC. Cette solution offrira un espace centralis´e permettant de g´erer efficacement les projets en cours, d’optimiser la gestion des stocks et d’automatiser le processus de r´eservation du mat´eriel. Une attention particuli`ere a ´et´e port´ee a` l’impression 3D/2D, ou` un Raspberry Pi sera utilis´e pour assurer une liaison directe avec les imprimantes du FabLab, permettant ainsi aux utilisateurs de soumettre et de suivre leurs impressions en temps r´eel.

L’un des aspects novateurs du projet r´eside ´egalement dans l’int´egration d’un chatbot intelligent, conc¸u pour assister les utilisateurs en leur fournissant des r´eponses instantan´ees et adapt´ees a` leurs besoins.

Chapitre 1

# Contexte G´en´eral

## 1.1 Introduction

Le pr´esent chapitre vise a` introduire le cadre g´en´eral du projet en pr´esentant l’organisme d’accueil, le contexte dans lequel s’inscrit cette initiative, ainsi que les probl´ematiques rencontr´ees. Enfin, les objectifs poursuivis seront d´etaill´es afin de d´efinir les grandes lignes du d´eveloppement de la solution propos´ee.

## 1.2 Pr´esentation de l’Organisme d’Accueil

### 1.2.1 Orange Maroc

Orange Maroc est une filiale du groupe multinational Orange, un des leaders mondiaux dans le domaine des t´el´ecommunications. Depuis sa fondation en 1999 sous le nom de M´edi T´el´ecom, l’op´erateur a jou´e un roˆle central dans le d´eveloppement des t´el´ecommunications au Maroc, contribuant `a la d´emocratisation des services de t´el´ephonie mobile, Internet haut d´ebit et solutions d’entreprise. Aujourd’hui, Orange Maroc se positionne comme un acteur incontournable sur le march´e des t´el´ecommunications au Maroc, offrant une large gamme de services a` destination des particuliers, des entreprises et des collectivit´es.

En tant qu’op´erateur innovant, Orange Maroc met un accent particulier sur la transformation num´erique du pays, avec des initiatives visant `a acc´el´erer l’acc`es aux technologies modernes pour un plus grand nombre de citoyens. L’entreprise participe activement a` des projets de d´eveloppement durable, de responsabilit´e soci´etale et d’inclusion num´erique, dans le but de r´eduire la fracture num´erique et de promouvoir les nouvelles technologies. Par ailleurs, Orange Maroc soutient de nombreuses initiatives dans les domaines de l’´education, de l’entrepreneuriat et de la sant´e, visant a` favoriser l’innovation et `a am´eliorer la qualit´e de vie au Maroc.

Orange Maroc n’est pas seulement un fournisseur de services de t´el´ecommunication. C’est aussi un acteur cl´e dans l’´ecosyst`eme technologique du pays, collaborant avec des start-ups, des universit´es et des institutions publiques pour promouvoir l’innovation. Cette approche proactive se refl`ete dans ses investissements dans les infrastructures de t´el´ecommunications et dans des projets qui favorisent la transformation num´erique des entreprises et de la soci´et´e en g´en´eral.



Figure 1.1 – Orange Maroc

L’organigramme ci-dessous illustre la structure organisationnelle d’Orange Maroc :

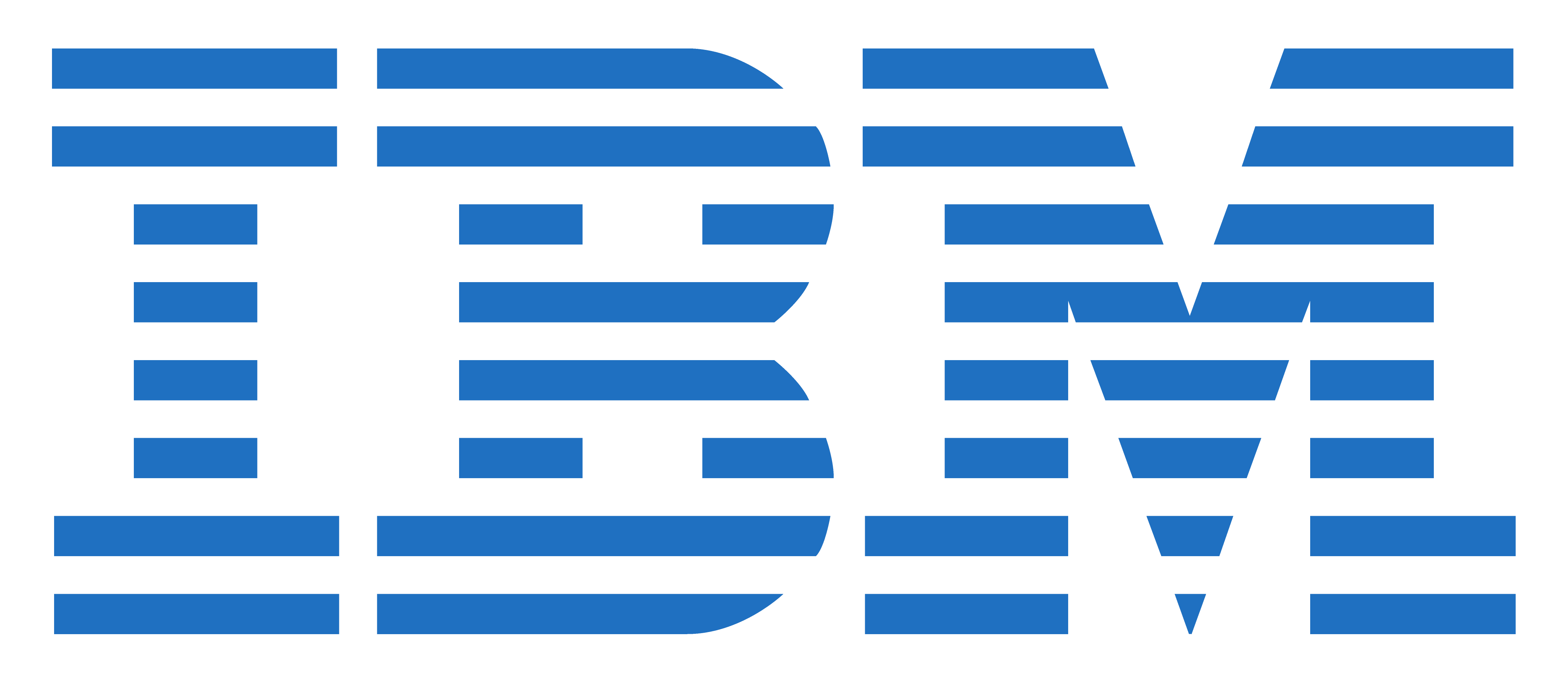
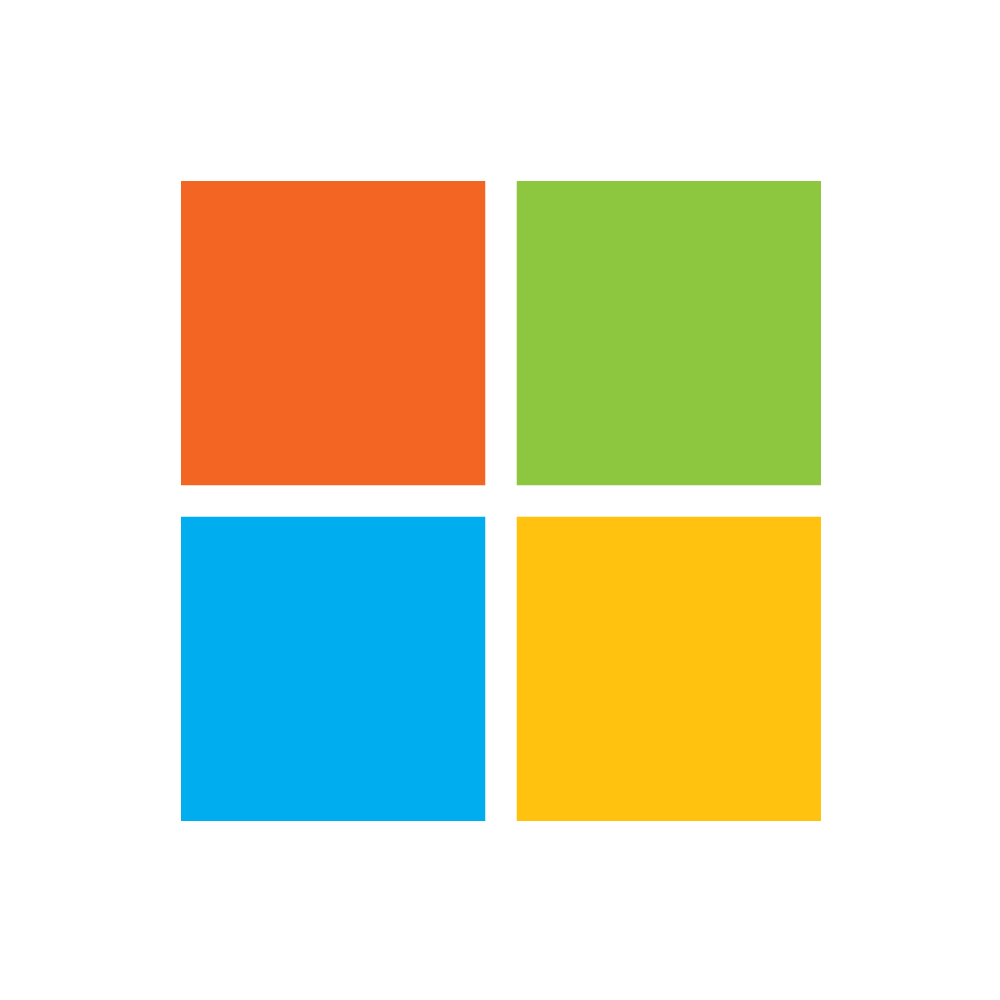


Figure 1.2 – Organigramme Orange Maroc

### 1.2.2 Les Clients d’Orange Maroc

Orange Maroc sert une large base de clients, allant des particuliers aux entreprises de toutes tailles, en passant par les administrations publiques. L’entreprise s’efforce de r´epondre aux besoins vari´es de ses clients en proposant des services adapt´es `a chaque segment. Parmi ses clients, on trouve des ´etudiants, des jeunes professionnels, des grandes entreprises, des PME et des institutions publiques. Orange Maroc met l’accent sur la proximit´e avec ses clients en offrant des solutions personnalis´ees et en assurant un service client de qualit´e.

En outre, Orange Maroc s’engage a` r´eduire la fracture num´erique en facilitant l’acc`es aux technologies de communication modernes. Graˆce a` ses initiatives, comme le programme Orange Digital Center, l’entreprise œuvre pour l’inclusion num´erique en offrant des outils et des formations aux jeunes et aux start-ups.



### 1.2.3 Evolution des services´

Orange Maroc a consid´erablement ´evolu´e depuis sa cr´eation en 1999, passant d’un simple op´erateur de t´el´ephonie mobile `a un fournisseur int´egr´e de services num´eriques. Aujourd’hui, l’entreprise propose des services de t´el´ecommunication classiques tels que la voix, l’Internet mobile, mais aussi des solutions avanc´ees comme la fibre optique, la gestion cloud, et des services IoT (Internet des objets). Ces ´evolutions t´emoignent de l’engagement d’Orange Maroc a` offrir a` ses clients des services innovants et a` la pointe de la technologie.

Avec l’essor de la digitalisation, Orange Maroc a ´elargi son offre en d´eveloppant des solutions adapt´ees `a l’`ere num´erique. Le d´eveloppement des services bancaires mobiles, des solutions pour les entreprises et la transformation digitale des secteurs cl´es sont des exemples de cette ´evolution. Cette transformation continue permet `a Orange Maroc de r´epondre aux attentes croissantes des consommateurs et d’accompagner les entreprises dans leur d´eveloppement num´erique.

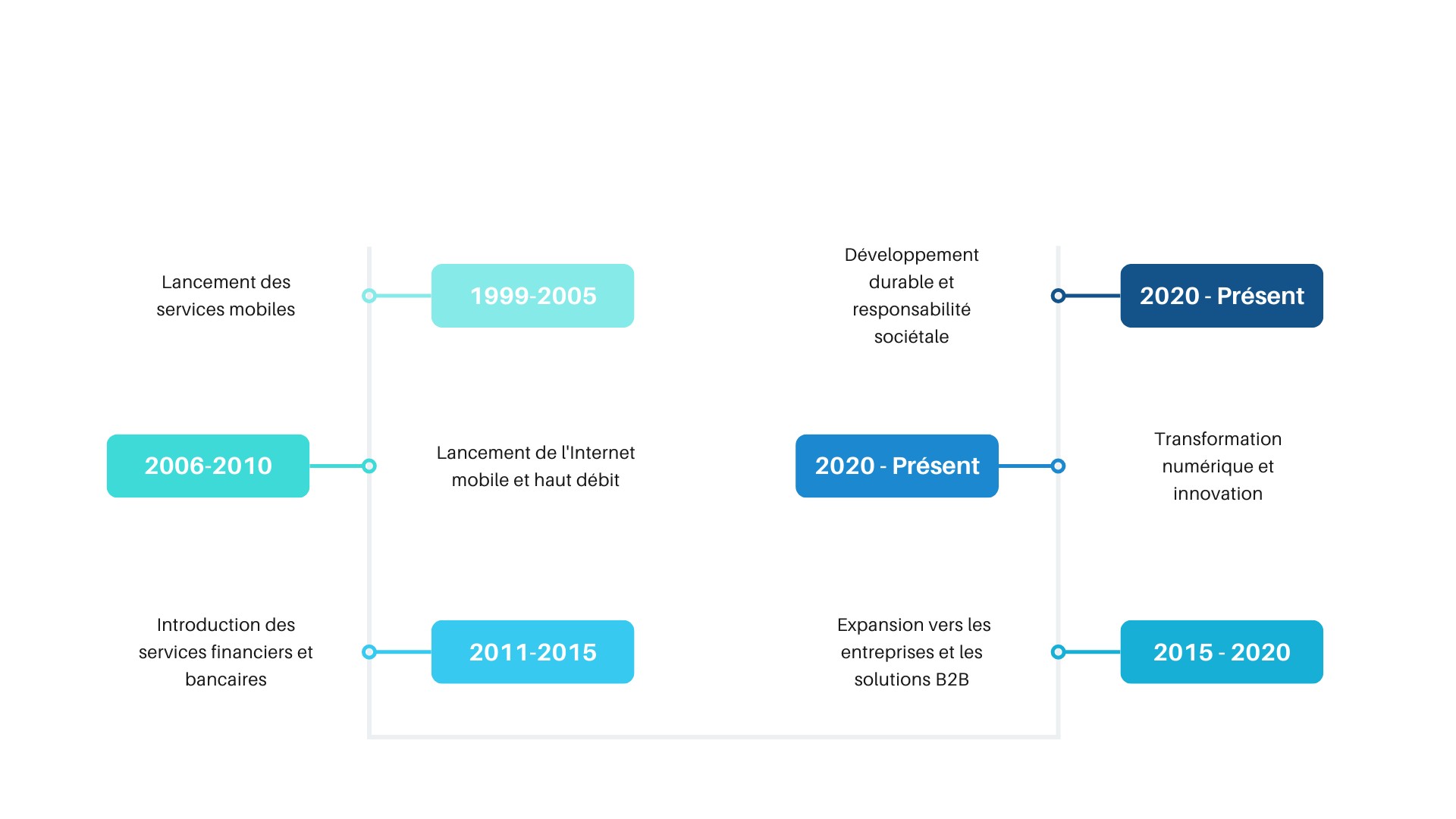


Figure 1.3 – Graphique des ´evolutions des services d’Orange Maroc

### 1.2.4 Orange Digital Center (ODC)

L’Orange Digital Center (ODC) est un programme lanc´e par Orange Maroc dans le but de favoriser l’acc`es `a la formation et `a l’innovation num´erique au Maroc. Ce centre, situ´e `a Rabat, Casablanca et Agadir se veut un v´eritable hub d’innovation, offrant aux jeunes talents, aux ´etudiants et aux entrepreneurs un espace d´edi´e a` l’apprentissage, a` la cr´eation et a` l’innovation technologique. Le programme ODC est d´esormais d´eploy´e dans plusieurs pays et a pour objectif de d´emocratiser l’acc`es aux technologies et de soutenir l’entrepreneuriat num´erique.

Le programme se compose de plusieurs axes strat´egiques :

* L’Ecole du Code : Il s’agit d’un programme de formation intensif dans les domaines de´ la programmation, du d´eveloppement web et des technologies num´eriques. L’objectif est d’outiller les jeunes Marocains avec des comp´etences techniques et pratiques en programmation afin de les pr´eparer aux d´efis du march´e du travail dans le secteur technologique.
* Le FabLab Solidaire : Le FabLab est un espace de fabrication num´erique con¸cu pour offrir aux jeunes et aux startups des outils de prototypage et de cr´eation. Ce laboratoire est ´equip´e de machines de fabrication num´erique telles que des imprimantes 3D, des d´ecoupeuses laser et d’autres ´equipements permettant la fabrication rapide de prototypes et de produits. Ce poˆle est particuli`erement pertinent pour des projets d’innovation dans les secteurs de la robotique, de l’ing´enierie et du design industriel.
* L’Acc´el´erateur de Startups : L’ODC soutient l’entrepreneuriat num´erique en offrant aux startups locales un programme d’acc´el´eration. Cet acc´el´erateur aide les jeunes entreprises `a se structurer, `a d´evelopper leur mod`ele ´economique et `a acc´eder aux financements n´ecessaires pour leur croissance. L’ODC travaille en collaboration avec plusieurs partenaires locaux et internationaux pour offrir un accompagnement complet aux entrepreneurs.

Le roˆle du FabLab dans ce projet est d’une grande importance, car il permet aux utilisateurs d’avoir acc`es `a des ressources mat´erielles de pointe pour leurs projets d’innovation. Cependant, malgr´e sa richesse en ´equipements, la gestion des ressources et des projets du FabLab reste bas´ee sur des m´ethodes traditionnelles, ce qui limite son efficacit´e. Ce projet de digitalisation vise justement `a apporter une solution pour moderniser cette gestion et rendre le FabLab plus accessible et fonctionnel.

En plus de ces axes, l’Orange Digital Center se distingue par son approche collaborative et inclusive. L’objectif est de cr´eer un ´ecosyst`eme d’innovation accessible `a tous, en offrant un environnement propice `a l’apprentissage, `a la cr´eativit´e et `a l’entrepreneuriat. Ainsi, l’ODC joue un rˆole cl´e dans la transformation num´erique du Maroc, en soutenant des initiatives visant a` r´eduire la fracture num´erique et a` offrir des opportunit´es ´egales a` tous les citoyens.

L’Orange Digital Center s’articule autour de plusieurs poˆles, comme le montre l’organigramme suivant :

ERAL´ 1.3. CADRE DU PROJET

## 1.3 Cadre du Projet

### 1.3.1 Domaine du Projet

Le projet s’inscrit dans le domaine de la digitalisation des FabLabs, des espaces collaboratifs ´equip´es de technologies permettant la fabrication de prototypes physiques a` partir de mod`eles num´eriques. Ces espaces offrent des outils tels que des imprimantes 3D, des d´ecoupeuses laser, des fraiseuses, etc., permettant aux utilisateurs de concevoir et de produire des objets divers.

La gestion des FabLabs repose actuellement sur des m´ethodes partiellement informatis´ees ou manuelles, ce qui engendre des difficult´es dans le suivi des projets, la gestion des stocks et la r´eservation du mat´eriel. Le projet vise `a r´epondre `a ces d´efis en d´eveloppant une solution num´erique centralis´ee. Cette plateforme web permettra de faciliter la gestion des ´equipements, des stocks, des projets et des utilisateurs tout en optimisant le processus de r´eservation et de suivi des impressions 3D/2D.

Ainsi, le domaine d’application du projet couvre l’ensemble des activit´es li´ees `a la gestion num´erique des FabLabs, visant a` am´eliorer l’efficacit´e, la trac¸abilit´e et l’accessibilit´e des ressources, tout en offrant une meilleure exp´erience utilisateur.

### 1.3.2 Probl´ematique

Le FabLab de l’Orange Digital Center rencontre plusieurs difficult´es majeures dans sa gestion quotidienne. Actuellement, la gestion des ressources, des projets et du mat´eriel repose en grande partie sur des m´ethodes manuelles et partiellement informatis´ees. Cela entraˆıne une absence de trac¸abilit´e pr´ecise des ´equipements et du mat´eriel, rendant difficile le suivi de leur utilisation et leur r´epartition. De plus, la gestion des stocks n’est pas optimis´ee, ce qui peut mener a` des erreurs d’inventaire, des ruptures de stock ou des surplus inutiles. Le processus de r´eservation du mat´eriel est ´egalement un point de friction, car il est trop complexe et peu intuitif, ce qui entraˆıne une perte de temps pour les utilisateurs et les gestionnaires. Enfin, le manque de suivi automatis´e des impressions 3D et 2D limite l’efficacit´e du FabLab, car les utilisateurs ne peuvent pas suivre en temps r´eel l’´etat de leurs demandes d’impression, ce qui nuit `a la productivit´e et `a l’organisation g´en´erale. Face `a ces enjeux, il devient crucial de d´evelopper une solution num´erique centralis´ee permettant d’optimiser la gestion des ressources, d’am´eliorer la trac¸abilit´e et d’automatiser les processus, afin de rendre le FabLab plus fonctionnel et accessible pour ses utilisateurs.

### 1.3.3 Objectifs du projet

Le principal objectif de ce projet est de concevoir et de d´evelopper une plateforme web d´edi´ee `a la gestion du FabLab de l’Orange Digital Center. Les objectifs sp´ecifiques sont les suivants :

#### ERAL´ 1.4. CONCLUSION

* Gestion des projets : Permettre aux utilisateurs de cr´eer, suivre et g´erer leurs projets au sein du FabLab.
* Suivi des stocks et ´equipements : Assurer une gestion centralis´ee des stocks, avec une tra¸cabilit´e en temps r´eel des ´equipements et mat´eriaux disponibles. • Syst`eme de r´eservation : D´evelopper une interface permettant aux membres de r´eserver facilement du mat´eriel, en optimisant l’attribution des ressources. • Automatisation des impressions 3D/2D : Int´egrer un Raspberry Pi pour une liaison directe avec les imprimantes 3D/2D afin de suivre et g´erer les impressions de mani`ere autonome.
* Assistance par chatbot : Int´egrer un chatbot intelligent qui r´epond aux questions des utilisateurs et les aide dans leurs d´emarches.

Cette plateforme vise `a rendre la gestion du FabLab plus fluide, organis´ee et accessible, tout en am´eliorant l’exp´erience utilisateur et la productivit´e des membres.

## 1.4 Conclusion

Ce chapitre a permis de d´efinir le cadre du projet en exposant le contexte de l’Orange Digital Center et en identifiant les probl´ematiques li´ees `a la gestion des FabLabs. Les principaux d´efis relev´es incluent une gestion manuelle et inefficace des ressources, une faible tra¸cabilit´e des ´equipements et une complexit´e dans la r´eservation du mat´eriel.

En r´eponse `a ces probl`emes, les objectifs du projet ont ´et´e clairement d´efinis : d´evelopper une plateforme web permettant de centraliser la gestion des projets, du mat´eriel et des impressions. Le chapitre suivant se concentrera sur l’Analyse et Sp´ecification de notre projet.

Chapitre 2

# Analyse et Sp´ecification

Avant d’entamer la phase de conception et de d´eveloppement d’une solution logicielle, il est essentiel de proc´eder `a une ´etude fonctionnelle rigoureuse. Cette ´etape constitue le socle sur lequel reposera tout le projet, en permettant d’identifier de mani`ere pr´ecise les besoins des utilisateurs finaux, d’´evaluer les forces et les limites du syst`eme existant, ainsi que de d´efinir les sp´ecifications fonctionnelles et techniques a` respecter.

L’´etude fonctionnelle joue un roˆle cl´e dans la r´eussite d’un projet, car elle garantit que les choix de conception sont align´es avec les attentes des parties prenantes, tout en anticipant les ´eventuelles contraintes li´ees `a l’environnement technologique ou organisationnel. Ce chapitre pr´esente donc une analyse d´etaill´ee du contexte actuel, des acteurs impliqu´es, des besoins `a satisfaire, ainsi que des exigences essentielles `a la mise en œuvre efficace et p´erenne de la solution envisag´ee.

## 2.1 Etude Pr´ealable´

### 2.1.1 Etude de l’existant´

Actuellement, la gestion du FabLab au sein de l’Orange Digital Center repose principalement sur des m´ethodes manuelles et partiellement informatis´ees. Ces m´ethodes incluent l’utilisation de fichiers Excel pour la gestion des stocks, ainsi que des r´eservations de mat´eriel via des carnets physiques. Les utilisateurs doivent r´eserver manuellement des ´equipements, sans syst`eme de suivi automatis´e des r´eservations ou des impressions. Cette situation engendre plusieurs probl´ematiques :

#### 2.1. ETUDE PR´ EALABLE´

* **Manque de trac¸abilit´e** : Il n’existe pas de syst`eme centralis´e pour suivre l’utilisation des ressources, rendant difficile la gestion des stocks et des ´equipements.
* **Risque d’erreurs humaines** : La gestion des stocks et des r´eservations manuelles entraˆıne des erreurs d’inventaire et des conflits de r´eservation.
* **Absence de suivi en temps r´eel** : Les utilisateurs ne peuvent pas suivre l’´etat de leurs impressions ou r´eservations, ce qui affecte l’efficacit´e et l’organisation g´en´erale du FabLab.
* **Complexit´e du processus de r´eservation** : Le syst`eme de r´eservation actuel est peu intuitif et n´ecessite des d´emarches manuelles.

Ces probl`emes montrent l’inefficacit´e du syst`eme existant et la n´ecessit´e de d´evelopper une solution num´erique centralis´ee pour automatiser et optimiser la gestion des ressources du FabLab.

### 2.1.2 Critique de l’existant

La critique du syst`eme constitue une ´etape utile et importante. Elle a pour but de porter un jugement objectif afin de d´eceler les insuffisances ´eventuelles rencontr´ees au cours de l’´etude de l’existant en vue de proposer un syst`eme plus fiable que le syst`eme ancien. Apr`es avoir analys´e l’existant, les principales critiques de l’existant sont :

* **Gestion manuelle des ressources** : La gestion des stocks et des ´equipements est r´ealis´ee manuellement, ce qui prend du temps et expose `a des erreurs.
* **Suivi peu transparent des projets et des r´eservations** : Il est difficile pour les utilisateurs et les gestionnaires de suivre l’´etat des projets et des r´eservations en temps r´eel, cr´eant ainsi de la confusion et des conflits de ressources.
* **Complexit´e des processus** : Les processus actuels de r´eservation, de gestion des stocks et de suivi des impressions sont longs et non automatis´es, ce qui diminue la productivit´e.
* **Absence d’assistance en ligne** : Le manque d’un chatbot ou d’un assistant virtuel empˆeche les utilisateurs d’obtenir une aide imm´ediate pour r´esoudre leurs questions ou probl`emes.

La critique de ces points montre clairement que la solution actuelle ne r´epond pas aux exigences de performance et de simplicit´e n´ecessaires pour un FabLab moderne.

2.2. ANALYSE DES BESOINS

## 2.2 Analyse des besoins

### 2.2.1 Identification des acteurs

La premi`ere des missions est de bien identifier les acteurs qui vont interagir avec le syst`eme. L’analyse des acteurs est une ´etape importante dans un projet de d´eveloppement. Elle permet d’appr´ecier les interrelations et de mieux mesurer l’influence de chacun des groupes d’acteurs sur l’action `a mener. Dans notre cas nous avons des acteurs humains qui sont le superadmin , l’admin et le membre :

* **Administrateur** : L’administrateur est responsable de la gestion quotidienne du FabLab. Il peut g´erer les utilisateurs (cr´eation, modification, suppression), superviser les ´equipements, les projets et les stocks, ainsi que valider les projets, les r´eservations et les tickets.
* **Super Administrateur** : Le super administrateur poss`ede tous les droits du syst`eme. Il g`ere les administrateurs, supervise l’ensemble du fonctionnement de la plateforme et peut intervenir a` tous les niveaux en cas de probl`eme ou de besoin sp´ecifique.
* **Utilisateur** : L’utilisateur peut ˆetre un membre du FabLab ou une personne externe ayant re¸cu une autorisation d’acc`es. Il peut consulter les ´equipements disponibles, faire des r´eservations, proposer des projets, et suivre l’avancement de ses impressions.

Ces acteurs interagiront avec la plateforme de mani`ere diff´erente, mais leur objectif commun est d’optimiser l’utilisation des ressources du FabLab.

### 2.2.2 Besoins du projet

#### Exigences fonctionnelles

Pour r´epondre a` un besoin r´eel, en termes de contraintes fonctionnelles, l’application doit permettre aux utilisateurs plusieurs fonctionnalit´es pr´esent´ees comme suit :

* **Gestion des projets** : Les utilisateurs pourront cr´eer, modifier et suivre leurs projets, avec un tableau de bord pour visualiser l’avancement.
* **Gestion des stocks et des ´equipements** : La plateforme devra permettre une gestion centralis´ee des stocks et une tra¸cabilit´e des ´equipements en temps r´eel.
* **Syst`eme de r´eservation** : Les utilisateurs devront pouvoir r´eserver facilement du mat´eriel, avec un calendrier des disponibilit´es et un syst`eme de validation des r´eservations.
* **Suivi des impressions 3D/2D** : L’int´egration des imprimantes 3D et 2D permettra un suivi en temps r´eel des impressions.
* **Assistance par chatbot** : Un chatbot devra ˆetre int´egr´e pour r´epondre aux questions courantes et guider les utilisateurs dans leur utilisation du FabLab.

2.2. ANALYSE DES BESOINS

#### Exigences techniques

Les exigences techniques pour le d´eveloppement du projet incluent :

* **Frontend** : D´eveloppement de l’interface utilisateur avec Angular pour une exp´erience web dynamique et r´eactive.
* **Backend** : Le backend sera bas´e sur Spring Boot, offrant une architecture robuste pour g´erer les API et l’interaction avec la base de donn´ees.
* **Base de donn´ees** : Utilisation d’une base de donn´ees relationnelle (MySQL ou PostgreSQL) pour stocker les informations sur les utilisateurs, les projets, les r´eservations et les ´equipements.
* **Raspberry Pi** : Int´egration d’un Raspberry Pi pour assurer la communication avec les imprimantes 3D/2D et suivre l’´etat des impressions.

#### Exigences non fonctionnelles

Les exigences non fonctionnelles sont les crit`eres d’´evaluation des performances d’un syst`eme logiciel et un syst`eme logiciel doit avoir certains attributs de qualit´e pour r´epondre aux exigences non fonctionnelles. En ce qui concerne les objectifs en termes de qualit´e, le syst`eme doit r´epondre autant que possible aux exigences suivantes :

* **Performance** : La plateforme doit ˆetre rapide et r´eactive, avec des temps de r´eponse optimis´es pour toutes les op´erations courantes (r´eservations, suivi des impressions).
* **S´ecurit´e** : La solution doit garantir la s´ecurit´e des donn´ees des utilisateurs et la confidentialit´e des informations relatives aux projets et aux r´eservations.
* **Scalabilit´e** : Le syst`eme doit ˆetre conc¸u de mani`ere a` pouvoir ´evoluer et supporter un nombre croissant d’utilisateurs et de projets.
* **Compatibilit´e mobile** : L’application doit ˆetre compatible avec les principaux navigateurs web et responsive pour les appareils mobiles.
* **Ergonomie** : L’interface utilisateur doit ˆetre intuitive, avec une navigation simple et rapide, permettant `a tous les utilisateurs de trouver facilement les informations dont ils ont besoin.

2.3. CONCLUSION

## 2.3 Conclusion

Ce chapitre a permis de r´ealiser une analyse approfondie des besoins du projet, en partant d’une ´etude de l’existant pour en tirer les points de critique, puis en d´efinissant les exigences fonctionnelles, techniques et non fonctionnelles qui orienteront le d´eveloppement de la plateforme. La centralisation de la gestion des ressources, des projets, des r´eservations et des impressions, ainsi que l’int´egration d’un chatbot pour l’assistance, constitueront les bases d’une solution num´erique optimis´ee pour le FabLab de l’Orange Digital Center. Le chapitre suivant se concentrera sur la conception d´etaill´ee de la solution.

Chapitre 3

# Etude Technique´

Ce chapitre est consacr´e `a l’analyse des besoins techniques n´ecessaires `a la mise en œuvre de l’application. Il pr´esente les solutions technologiques `a adopter ainsi que les pr´erequis techniques indispensables pour r´epondre efficacement a` l’ensemble des exigences fonctionnelles pr´ec´edemment d´efinies.

## 3.1 Architecture Microservices

### 3.1.1 D´efinition

Traditionnellement, les applications sont d´evelopp´ees selon une architecture monolithique, ou` tous les composants sont regroup´es dans un seul et mˆeme d´eploiement. Toutefois, `a mesure que l’application gagne en complexit´e, il devient de plus en plus difficile d’y ajouter de nouvelles fonctionnalit´es ou de corriger rapidement les ´eventuels probl`emes. L’architecture microservices apporte une r´eponse `a ces limitations en permettant une organisation de l’application sous forme de services ind´ependants, facilitant ainsi le d´eveloppement, la maintenance et la r´eactivit´e du syst`eme.

### 3.1.2 Avantages des Microservices

Malgr´e certaines hesistements initiales de la part de certains d´eveloppeurs face au changement, les microservices offrent de nombreux avantages qui surpassent largement les inconv´enients potentiels, en particulier pour les applications complexes ou `a grande ´echelle. Cette architecture, fond´ee sur des composants modulaires ind´ependants, facilite la compr´ehension, les tests et la gestion des applications, tout en am´eliorant la productivit´e et l’agilit´e des ´equipes. Voici les principaux avantages des microservices :

#### ETUDE TECHNIQUE 3.1. ARCHITECTURE MICROSERVICES

#### 1. R´eduction de la complexit´e

L’architecture microservices permet de d´ecouper une application monolithique en plusieurs services autonomes, chacun avec des responsabilit´es bien d´efinies, souvent expos´ees via des APIs ou des messages. Cette modularit´e rend les services plus faciles a` d´evelopper, `a comprendre et a` maintenir.

#### 2. Optimisation des ressources et des ´equipes

Chaque service peut ˆetre confi´e `a une ´equipe d´edi´ee, ce qui permet une meilleure r´epartition des ressources humaines et techniques. Cette ind´ependance favorise ´egalement le travail en parall`ele sur diff´erents modules.

#### 3. Libert´e technologique et ´evolutivit´e

Les d´eveloppeurs peuvent choisir les technologies les plus adapt´ees a` chaque service, sans ˆetre contraints par des choix techniques anciens ou obsol`etes. Il est ´egalement plus facile de mettre `a jour ou de r´e´ecrire un service sans impacter l’ensemble du syst`eme.

#### 4. D´eploiement continu facilit´e

Grˆace `a l’ind´ependance des services, les mises `a jour peuvent ˆetre d´eploy´ees individuellement sans perturber les autres parties de l’application. Cela rend possible une livraison continue, plus rapide et plus fiable.

#### 5. Meilleure scalabilit´e

Contrairement aux architectures monolithiques, les microservices permettent un dimensionnement cibl´e. Il est possible de faire ´evoluer uniquement les services les plus sollicit´es, ce qui optimise l’utilisation des ressources mat´erielles et am´eliore la performance globale du syst`eme.

#### 6. Facilit´e de maintenance

La maintenance est simplifi´ee puisque chaque microservice est ind´ependant. En cas de bug ou de mise a` jour, l’intervention peut se faire sur un seul service sans impacter le reste de l’application. Cela permet de corriger plus rapidement les erreurs et de r´eduire les temps d’arrˆet.

##### ETUDE TECHNIQUE 3.2. ARCHITECTURE DE L’APPLICATION

**3.2 Architecture de l’application**

## 3.3 Architecture interne d’un microservice

Chaque microservice de l’application suit une architecture interne bien d´efinie et repose sur un ensemble d’outils et de bonnes pratiques assurant sa robustesse, sa maintenabilit´e et son int´egration dans l’´ecosyst`eme global. Plus pr´ecis´ement, chaque microservice :

* Adopte une architecture en trois couches (pr´esentation, m´etier, donn´ees) pour une s´eparation claire des responsabilit´es.
* Est d´evelopp´e `a l’aide du framework **Spring Boot**, reconnu pour sa l´eg`eret´e et sa facilit´e de configuration.
* Est document´e automatiquement via l’outil **Swagger** (OpenAPI), facilitant l’exposition et la compr´ehension des API.
* Interagit avec une base de donn´ees d´edi´ee ou partag´ee, selon les besoins sp´ecifiques du service.
* Impl´emente un m´ecanisme de journalisation graˆce au framework **SLF4J**, permettant un suivi et une tra¸cabilit´e efficaces des actions.
* Contient des tests unitaires r´ealis´es `a l’aide des frameworks **JUnit** et **Mockito**, garantissant la qualit´e du code et facilitant la d´etection des r´egressions.

### 3.3.1 Architecture 3-tiers

L’architecture trois tiers (ou 3-couches) est un mod`ele d’architecture logicielle structurant l’application en trois niveaux logiques distincts, chacun ayant un rˆole bien d´efini. Cette s´eparation permet une meilleure organisation du code, une ´evolutivit´e facilit´ee et une maintenance plus ais´ee. Les trois couches principales sont les suivantes :

#### ETUDE TECHNI3Q.3U.EARCHITECTURE INTERNE D’UN MICROSERVICE

* **La couche de pr´esentation** : elle est responsable de la gestion des requˆetes entrantes (g´en´eralement HTTP), de la conversion des donn´ees (comme le JSON en objets) et de l’authentification initiale. Elle constitue l’interface entre l’utilisateur et l’application. Dans une application web, elle correspond aux composants front-end ou aux contrˆoleurs dans le back-end.
* **La couche m´etier (ou de traitement)** : elle contient la logique m´etier de l’application. Cette couche prend en charge le traitement des donn´ees, applique les r`egles fonctionnelles, effectue les v´erifications et g`ere les interactions entre les diff´erentes entit´es. Elle utilise ´egalement les services offerts par la couche de persistance.
* **La couche d’acc`es aux donn´ees (ou persistance)** : elle est charg´ee de toutes les op´erations li´ees au stockage des donn´ees. Elle permet de r´ecup´erer, ins´erer, modifier ou supprimer les donn´ees dans la base. Cette couche traduit les objets m´etier en entit´es de base de donn´ees et inversement.

Ce d´ecoupage en couches bien distinctes favorise la r´eutilisabilit´e du code, l’ind´ependance des modules, ainsi qu’une meilleure gestion des responsabilit´es au sein de l’application.

### 3.3.2 Communication entre les microservices

La communication entre les microservices repose sur une s´eparation claire des responsabilit´es, facilit´ee par l’utilisation d’API. Dans notre application, les ´echanges entre services sont assur´es via des API REST. Cette approche permet une interaction standardis´ee entre les composants du syst`eme, en utilisant les m´ethodes HTTP telles que GET pour la r´ecup´eration de donn´ees, POST pour la cr´eation, PUT pour la mise `a jour, et DELETE pour la suppression.

Ce mode de communication assure une interop´erabilit´e entre services, tout en garantissant la simplicit´e, la l´eg`eret´e et la flexibilit´e des ´echanges. Chaque microservice expose ses fonctionnalit´es via des points d’acc`es REST, permettant aux autres services de consommer ces ressources de mani`ere ind´ependante.

*[Voir Annexe – REST API pour plus de d´etails]*

#### ETUDE TECHNIQUE 3.4. INTEGRATION AVEC RASPBERRY PI (IOT)´

## 3.4 Int´egration avec Raspberry Pi (IoT)

Dans le cadre de notre projet, le Raspberry Pi joue un rˆole cl´e en tant qu’interface de communication entre notre syst`eme applicatif et les p´eriph´eriques d’impression 2D/3D. Grˆace `a sa polyvalence, sa connectivit´e r´eseau et sa capacit´e `a ex´ecuter des scripts, le Raspberry Pi permet de piloter les imprimantes `a distance via les microservices de l’application.

L’objectif est de permettre `a notre syst`eme, via des appels API, de lancer des impressions, de suivre leur ´etat ou encore d’interagir avec les imprimantes sans intervention directe sur les machines.

### Fonctionnement de l’int´egration

* **Connexion aux imprimantes** : le Raspberry Pi est connect´e localement aux imprimantes 2D ou 3D via USB ou r´eseau local.
* **R´eception des commandes** : il re¸coit les instructions d’impression depuis les microservices via des requˆetes HTTP REST (par exemple POST d’un fichier a` imprimer ou GET de l’´etat d’une file d’impression).
* **Ex´ecution locale** : un script Python (ou tout autre programme adapt´e) est ex´ecut´e sur le Raspberry Pi pour convertir et envoyer les fichiers aux imprimantes selon le protocole support´e (ex : G-code pour imprimantes 3D).
* **Retour d’´etat** : l’´etat de l’impression (en cours, termin´ee, erreur) est renvoy´e vers le syst`eme central, permettant un suivi temps r´eel via l’interface utilisateur.

### Avantages de cette architecture

* **S´eparation des responsabilit´es** : les imprimantes sont g´er´ees ind´ependamment du syst`eme principal, via une couche IoT.
* **Pilotage `a distance** : les impressions peuvent ˆetre d´eclench´ees et suivies sans pr´esence physique.
* **Souplesse et ´evolutivit´e** : il est possible d’ajouter d’autres imprimantes ou Raspberry Pi sans modifier l’architecture globale.
* **Int´egration fluide avec les microservices** : le Raspberry agit comme un microservice d´edi´e `a l’impression.

Ainsi, le Raspberry Pi assure un pont intelligent entre le syst`eme logiciel et les dispositifs d’impression physique, rendant l’infrastructure plus modulaire, automatis´ee et connect´ee.

ETUDE TECHNIQUE 3.4. INTEGRATION AVEC RASPBERRY PI (IOT)´

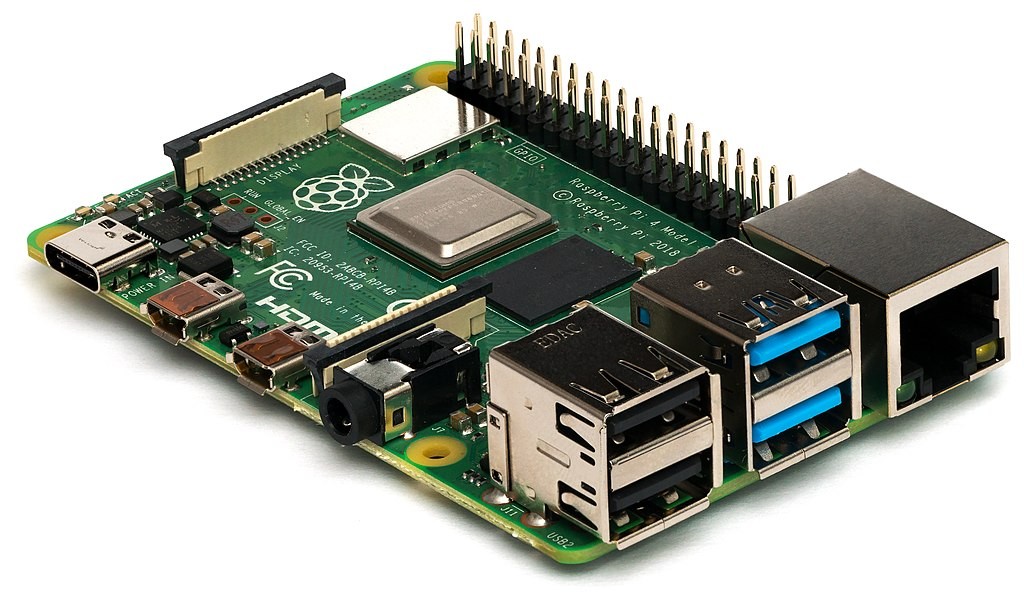


Figure 3.1 – Raspberry Pi

## 3.5 Frameworks, bibliothèques et API utilisés

### 3.5.1 Spring Boot



Spring Boot est un framework open-source d´evelopp´e par la soci´et´e Pivotal en 2012. Il s’inscrit dans l’´ecosyst`eme de la Spring IO Foundation et offre un point d’acc`es unifi´e `a l’ensemble de ses projets (comme Spring Web, Spring Batch, Spring Data, etc.), facilitant ainsi le d´eveloppement d’applications robustes, performantes et modulaires.

L’une des principales forces de Spring Boot r´eside dans sa capacit´e a` r´eduire consid´erablement la configuration n´ecessaire pour d´emarrer une application. Grˆace `a des conventions pr´ed´efinies et des configurations automatiques, les d´eveloppeurs peuvent se concentrer davantage sur la logique m´etier, tout en b´en´eficiant d’un cadre structur´e et coh´erent. Cette approche est souvent qualifi´ee d’≪ opinionated ≫ : le framework propose un ensemble de choix technologiques et de pratiques par d´efaut, tout en permettant de les personnaliser si besoin.

Par ailleurs, Spring Boot ´elimine le besoin d’´ecrire de longs fichiers de configuration

XML, en adoptant une approche plus moderne bas´ee sur les annotations et les fichiers

‘application.properties‘ ou ‘application.yml‘. Cette simplification am´eliore la productivit´e, r´eduit les erreurs de configuration et acc´el`ere le cycle de d´eveloppement.

En r´esum´e, Spring Boot s’impose aujourd’hui comme l’un des frameworks les plus utilis´es pour le d´eveloppement de microservices et d’applications Java modernes, en raison de sa simplicit´e, sa flexibilit´e et sa richesse fonctionnelle.

### 3.5.2 Spring Data



Spring Data est un sous-projet de l’´ecosyst`eme Spring dont l’objectif principal est de faciliter l’interaction avec les syst`emes de gestion de donn´ees, qu’il s’agisse de bases de donn´ees relationnelles, de bases NoSQL, de solutions Big Data ou encore d’API Web.

Graˆce a` une abstraction unifi´ee et une int´egration transparente avec diff´erents moteurs de persistance, Spring Data permet aux d´eveloppeurs de se concentrer sur la logique m´etier sans avoir `a ´ecrire manuellement les requˆetes de persistance. L’un des aspects les plus puissants de Spring Data r´eside dans l’utilisation de conventions de nommage : en d´efinissant simplement des m´ethodes dans une interface, le framework est capable de g´en´erer automatiquement les impl´ementations correspondantes, en fonction de leur nom.

Cette approche r´eduit consid´erablement la quantit´e de code boilerplate, tout en rendant les requˆetes plus lisibles et maintenables. Spring Data s’int`egre parfaitement avec d’autres modules de Spring, notamment Spring Boot, ce qui en fait un outil incontournable pour le d´eveloppement d’applications modernes, performantes et ´evolutives.

### 3.5.3 Swagger



Swagger est une sp´ecification standardis´ee qui permet de documenter efficacement une

API REST, tout en offrant une interface interactive pour tester les diff´erents points d’entr´ee de l’application. Il facilite grandement le travail des d´eveloppeurs en analysant automatiquement le code source pour g´en´erer une documentation accessible via une URL d´edi´ee.

Cette documentation, g´en´eralement disponible au format JSON ou XML, d´ecrit pr´ecis´ement les services expos´es : les m´ethodes HTTP disponibles (GET, POST, PUT, DELETE), les chemins d’acc`es, les param`etres attendus, les formats d’entr´ee et de sortie, ainsi que les codes de r´eponse. Grˆace `a Swagger, les clients ou int´egrateurs peuvent facilement comprendre et consommer les services propos´es, sans avoir besoin de consulter le code source.

### 3.5.4 Rest API



REST (Representational State Transfer) est un style d’architecture logicielle destin´e `a la conception de services web l´egers, flexibles et ´evolutifs. Une API RESTful repose sur un ensemble de contraintes bien d´efinies, telles que l’absence d’´etat, l’utilisation d’une interface uniforme, et la manipulation de ressources identifi´ees par des URI.

Les services web construits selon ce mod`ele, appel´es API REST, favorisent l’interop´erabilit´e entre syst`emes r´epartis sur Internet. Ils permettent aux clients d’interagir avec des ressources (donn´ees) via des repr´esentations textuelles, g´en´eralement en JSON ou XML, en utilisant les m´ethodes HTTP standards telles que GET, POST, PUT et DELETE.

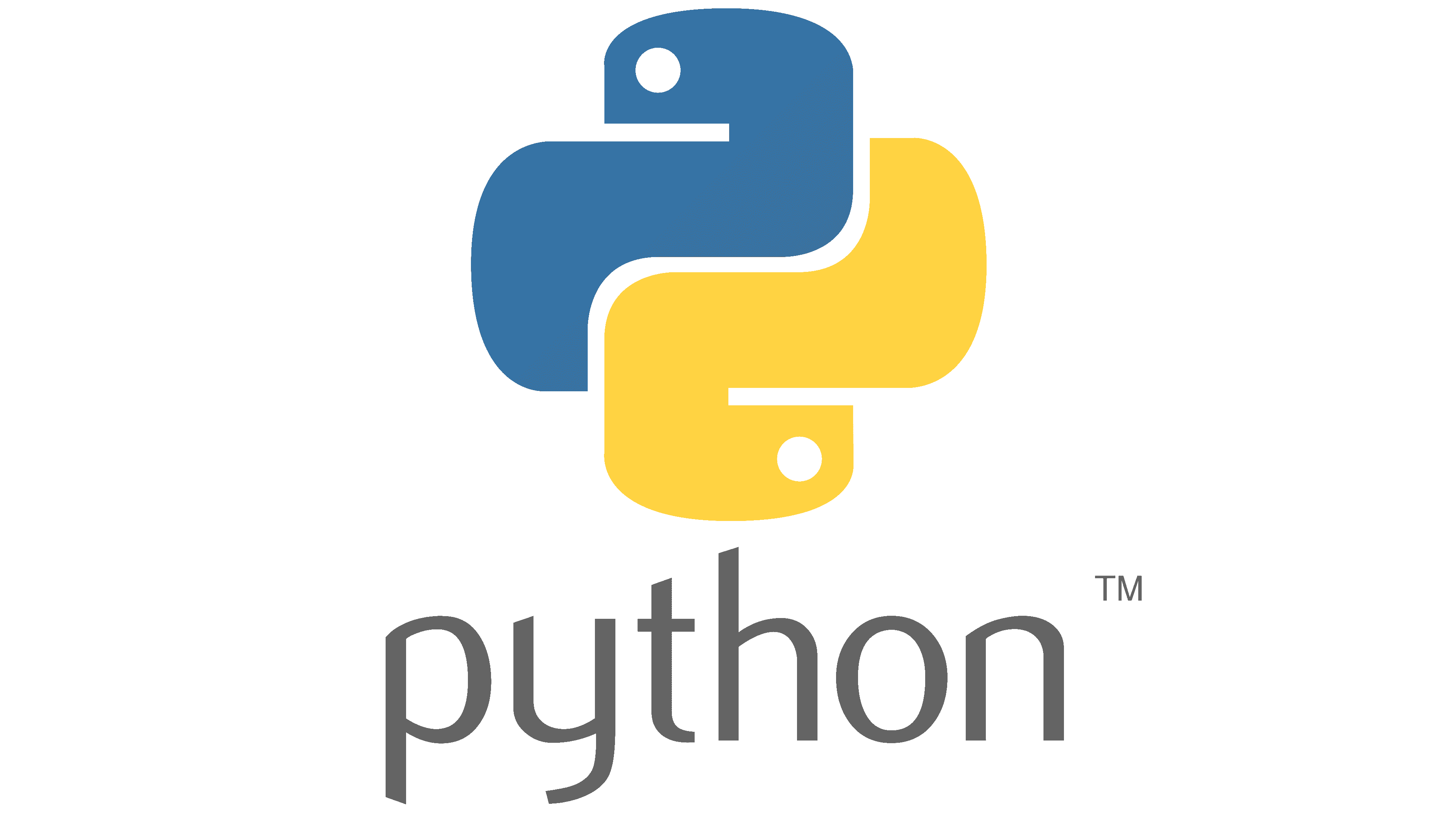
### 3.5.5 Google Gemini via Google API



Le cœur du chatbot repose sur l’intégration du modéle **Gemini**, développé par Google. Ce modéle de traitement du langage naturel, accessible via les **Google APIs**, permet au chatbot de fournir des réponses intelligentes, naturelles et contextuelles. Grâce à sa puissance et sa mise `a jour continue par Google, il assure une interaction fluide avec les utilisateurs.

L’intégration se fait via des appels REST sécurisés a` l’API Gemini, avec une gestion fine des clés d’authentification et des quotas. Ce mod`ele constitue le **modéle principal** utilisé en production pour fournir les réponses en temps réel.

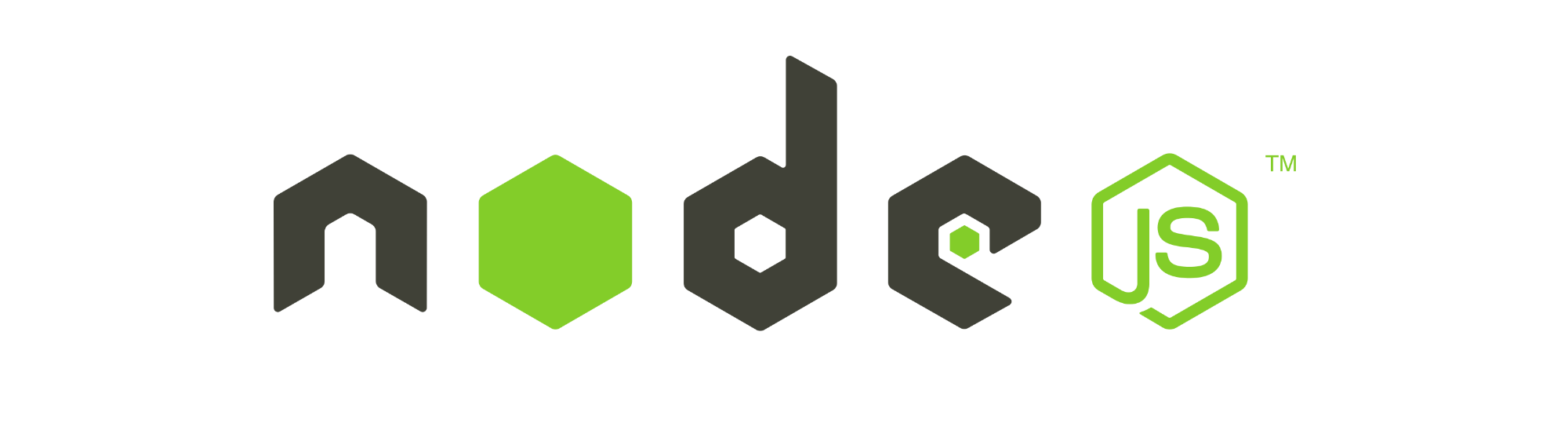
### 3.5.6 Python



Afin de garantir la **disponibilit´e continue du chatbot**, une solution de secours a ´et´e d´evelopp´ee par notre ´equipe en Python. Ce second mod`ele est bas´e sur des techniques d’apprentissage automatique (machine learning) et utilise des biblioth`eques telles que scikit-learn, transformers ou encore Llama.

Il est activ´e automatiquement en cas d’indisponibilit´e de l’API Google Gemini. Cette approche hybride permet de maintenir un haut niveau de performance tout en r´eduisant la d´ependance `a un seul fournisseur externe.

### 3.5.7 Node JS



Toute la logique métier du chatbot ainsi que sa communication avec les utilisateurs est encapsulée dans un microservice développée en **Node.js**. Ce choix technologique permet une gestion efficace des connexions asynchrones et une forte scalabilité.

Le microservice en Node.js :

* G`ere les requˆetes entrantes des utilisateurs (par exemple via une interface web ou mobile),
* Interagit avec les deux modéles (Gemini ou celui développée en Python) selon leur disponibilitée,
* Formate les réponses et les renvoie au client de maniére transparente.

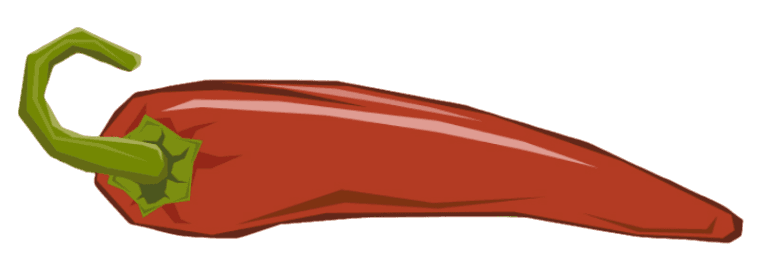
Graˆce a` cette architecture modulaire, le chatbot est capable d’assurer un service fiable, maintenable et évolutif.

### 3.5.8 Apache Kafka



Apache Kafka est une plateforme distribuée de [diffusion de données en continu](https://www.redhat.com/fr/topics/integration/les-donnees-de-diffusion-en-continu-quest-ce-que-cest), capable de publier, stocker, traiter et souscrire à des flux d'enregistrement en temps réel. Elle est conçue pour gérer des flux de données provenant de plusieurs sources et les fournir à plusieurs utilisateurs. En bref, elle ne se contente pas de déplacer un volume colossal de données d'un point A à un point B : elle peut le faire depuis n'importe quels points vers n'importe quels autres points, selon vos besoins et même simultanément.

### 3.5.8 Lombok



**Lombok** est une biblioth`eque Java l´eg`ere qui s’int`egre directement a` l’IDE ou a` l’´editeur de code du d´eveloppeur. Elle permet de simplifier consid´erablement le d´eveloppement en r´eduisant la verbosit´e du code source. Graˆce a` un ensemble d’annotations, Lombok g´en`ere automatiquement du code r´ep´etitif comme les getters, setters, constructeurs, m´ethodes equals, hashCode, et toString, ´evitant ainsi d’avoir a` les ´ecrire manuellement.

Son objectif est d’offrir aux d´eveloppeurs un code Java plus concis, plus lisible et plus maintenable. En effet, quelques annotations Lombok peuvent remplacer des dizaines, voire des centaines de lignes de code standard.

Lombok est compatible avec les outils de construction modernes tels que Maven et Gradle, et il est distribu´e sous la licence open-source MIT, ce qui facilite son adoption dans des projets professionnels ou acad´emiques. Son int´egration est simple et rapide, ce qui en fait un outil appr´eci´e des d´eveloppeurs pragmatiques.

### 3.5.9 MySQL



**MySQL** est un système de gestion de bases de données relationnelles (SGBDR) largement répandu, aussi bien dans le développement d’applications web que dans des environnements professionnels à grande échelle. Distribué sous une double licence, open source (GPL) et commerciale, il offre à la fois flexibilité et performance.

MySQL se distingue par sa simplicité d’utilisation, sa fiabilité et sa compatibilité avec de nombreux langages de programmation. Il constitue une solution robuste pour le stockage, la gestion et la manipulation des données. Grâce à sa popularité, il bénéficie d’une large communauté et d’un écosystème riche en outils. Il fait partie des SGBD les plus utilisés dans le monde, en concurrence directe avec des solutions comme Oracle, PostgreSQL ou encore Microsoft SQL Server.

ETUDE TECHNIQUE 3.6. CONCLUSION

### 3.5.10 Angular



**Angular** est une framework de développement web open source maintenu par Google, principalement destiné à la création d’applications web dynamiques et performantes côté front-end. Basé sur le langage TypeScript, Angular offre une architecture robuste orientée composants, facilitant la réutilisabilité, la modularité et la maintenabilité du code.

Ce framework propose un ensemble complet d’outils intégrés tels que la liaison de données bidirectionnelle (data binding), la gestion de formulaires, le routage, l’injection de dépendances et la communication avec des services REST via HTTP. Dans notre projet, Angular a été utilisé pour concevoir l’interface utilisateur, permettant aux utilisateurs d’interagir de manière fluide avec les fonctionnalités proposées par les microservices. Grâce à Angular, l’expérience utilisateur est intuitive, réactive et cohérente sur l’ensemble de l’application.

3.5.11

## 3.6 Conclusion

Angular est un framework de développement web open source, maintenu par Google, principalement conçu pour la création d’applications web dynamiques et performantes côté front-end. Reposant sur le langage TypeScript, Angular propose une architecture robuste orientée composants, favorisant la réutilisabilité, la modularité et la maintenabilité du code.

Il intègre un ensemble complet d’outils tels que la liaison de données bidirectionnelle (data binding), la gestion des formulaires, le routage, l’injection de dépendances, ainsi que la communication avec des services REST via le protocole HTTP. Dans le cadre de notre projet, Angular a été utilisé pour concevoir l’interface utilisateur, offrant aux utilisateurs une interaction fluide avec les différentes fonctionnalités exposées par les microservices. Grâce à Angular, l’expérience utilisateur se veut intuitive, réactive et homogène sur l’ensemble de l’application.

# Webographie

1. Java Spring Boot, <https://spring.io/projects/spring-boot>
2. JetBrains, <https://www.jetbrains.com/>
3. Baeldung, <https://www.baeldung.com/> : Java, Spring and Web Development tutorials.
4. Angular, <https://angular.dev/>
5. IntelliJ IDEA, <https://fr.vikidia.org/wiki/IntelliJ_IDEA>
6. Stack Overflow, <https://stackoverflow.com/> : QA site for computer science and development.
7. MySQL, <https://www.mysql.com/> : The world’s most popular open source database.
8. Amazon Web Services (AWS), <https://aws.amazon.com/> : Cloud computing services.