



# Spécification des Conditions requises pour l'Architecture

*Projet : Concevoir la nouvelle architecture pour la plateforme Foosus*

*Client : Foosus*

*Préparé par : Yoann VALERO - Architecte logiciel*

*N° de Version du Document : 0.1*

*Titre : Spécification des Conditions requises pour l'Architecture*

*Date de Version du Document :15/09/2022*

*Revu par : Yoann VALERO*

*Date de Révision :15/09/2022*

# Sommaire

<b>1</b>	<b>Objet de ce document.....</b>	<b>2</b>
<b>2</b>	<b>Mesures du succès.....</b>	<b>3</b>
2.1	Indicateur technique :.....	3
2.2	Indicateur business :.....	3
<b>3</b>	<b>Conditions requises pour l'architecture.....</b>	<b>4</b>
<b>4</b>	<b>Contrats de service business.....</b>	<b>4</b>
4.1	Accords de niveau de service.....	4
<b>5</b>	<b>Contrats de service application.....</b>	<b>5</b>
5.1	Indicateurs & Objectifs de niveau de service.....	5
<b>6</b>	<b>Lignes directrices pour l'implémentation.....</b>	<b>5</b>
<b>7</b>	<b>Spécifications pour l'implémentation.....</b>	<b>6</b>
7.1	Item de travail 1: Phase de Lancement / Communication.....	6
7.2	Item de travail 2: Mise en place de la méthodologie de travail.....	6
7.2.1	Mise en place méthodologie agile SCRUM.....	6
7.2.2	Mise en place de la pipeline d'intégration et de déploiement continu 'CI/CD).....	7
7.3	Item de travail 3: Migration de l'architecture existante vers le cloud.....	8
7.4	Item de travail 4 : Conception / développement / test des micro-services.....	9
7.4.1.1	Architecture existante.....	9
7.4.1.2	Architecture cible (orientée micro-service) :.....	10
7.4.1.3	Implémentation du composant de géolocalisation.....	11
7.5	Item de travail 5 : Refonte du Front-End.....	12
7.6	Item de travail 6 : Supervision continue de l'architecture.....	12
<b>8</b>	<b>Standards pour l'implémentation.....</b>	<b>13</b>
<b>9</b>	<b>Conditions requises pour l'interopérabilité.....</b>	<b>13</b>
<b>10</b>	<b>Conditions requises pour le management du service IT.....</b>	<b>14</b>
<b>11</b>	<b>Contraintes.....</b>	<b>14</b>
11.1	Contraintes financières.....	14
11.2	Contraintes organisationnelles.....	14
11.3	Contraintes temporelles.....	14

## 1 Objet de ce document

La Spécification des Conditions requises pour l'Architecture fournit un ensemble de déclarations quantitatives qui dessinent ce que doit faire un projet d'implémentation afin d'être conforme à l'architecture.

Une Spécification des Conditions requises pour l'Architecture constitue généralement un composant majeur du contrat d'implémentation, ou du contrat pour une Définition de l'Architecture plus détaillée.

Comme mentionné ci-dessus, la Spécification des Conditions requises pour l'Architecture accompagne le Document de Définition de l'Architecture, avec un objectif complémentaire : le Document de Définition de l'Architecture fournit une vision qualitative de la solution et tâche de communiquer l'intention de l'architecte.

## 2 Mesures du succès

---

Afin de suivre le bon déroulement du projet, les indicateurs suivants permettront de mesurer le succès du travail d'architecture :

### 2.1 Indicateur technique :

Indicateurs	Valeur cible	Description
Taux d'incidents de production sur 1 mois P1	Pour commencer : réduit de >25/mois à moins de 1/mois	Taux d'incidents de production sur 1 mois P1
Taux de disponibilité (SLA)	99,95 %	Taux de disponibilité (SLA) sur 1 mois
Couverture de code par les tests	>80 %	Couverture de code par les tests

### 2.2 Indicateur business :

Indicateurs	Valeur cible
Nombre d'adhésion d'utilisateurs par jour	Augmentation de 10 %
Adhésion de producteurs alimentaires par mois	Passer de 1,4/mois à 4/mois
Délai moyen de parution par semaine	Réduit de 3,5 semaines à moins d'une semaine
Taux d'incidents de production sur 1mois P1 impactant la satisfaction client	Pour commencer : réduit de >25/mois à moins de 1/mois

## 3 Conditions requises pour l'architecture

---

L'architecture cible de la nouvelle plateforme Foosus devra respecter les conditions suivantes :

- Disposer d'une architecture extensible et robuste ( supporter les pics d'affluence )
- Intégration d'une solution de géolocalisation pour proposer des produits proches de l'emplacement géographique des clients
- Favoriser l'innovation en implémentant une solution d'architecture maintenable et flexible ou il est facile d'itérer dessus.
- Être disponible depuis n'importe quelles régions dans le monde en limitant la perte de bande passante

## 4 Contrats de service business

### 4.1 Accords de niveau de service

---

Les accords de niveau de service business suivants seront suivis par l'équipe de Foosus :

<b>SLO (objectifs)</b>	<b>SLA (contrat)</b>	<b>SLI (Indicateurs clés)</b>
Augmentation du nombre d'utilisateurs	Augmenter de 10 % par jour	Nombre d'adhésion d'utilisateurs par jour
Augmenter les adhésions de producteurs alimentaires	Passer de 1,4/mois à 4 mois	Adhésion de producteurs alimentaires par mois
Réduire le délai moyen de parution de nouvelles offres	Réduit de 3,5 / semaines à moins d'une semaine	Délai moyen de parution de nouvelles offres par semaine

## 5 Contrats de service application

### 5.1 Indicateurs & Objectifs de niveau de service

---

SLO (objectifs)	SLA (contrat)	SLI (Indicateurs clés)
Réduction les incidents en production	Pour commencer : réduit de >25/mois à moins de 1/mois	Taux d'incidents de production sur 1 mois
Disponibilité de l'application	Taux de disponibilité de <b>99.95 %</b> (Crédit de service en cas de non respect des services AWS) <b>Crédit de service en % par mois remboursé :</b> - ≥ 99,0 % = 10 % - ≥ 95,0 % = 30 % - <95,0 % = 100 %	Temps de disponibilité / indisponibilité de la plateforme sur 1 mois

## 6 Lignes directrices pour l'implémentation

---

Les phases d'implémentation et les directives du projets sont les suivants :

- **Item de travail 1: Phase de Lancement / Communication :**

Cette étape est primordiale. Il faudra s'assurer que toutes les parties prenantes ont bien été informées et que toutes les guidelines du projet ont été comprises.

- **Item de travail 2: Mise en place de la méthodologie de travail :**

La méthodologie de travail agile et la méthode devOps doivent être parfaitement comprises par l'équipe de réalisation. Chaque collaborateur doit être formé aux technologies et méthodes de travail pour assurer le succès de la migration de l'architecture de la plateforme Foosus.

- **Item de travail 3: Migration de l'architecture existante vers le cloud :**

L'architecture existante doit être migrée en l'état vers le cloud pour corriger rapidement les incidences en production liées à la non-résistance à la charge en implémentant les solutions de load balancing et d'auto scaling.

Ceci permettra par la suite de commencer la restructuration des composants internes le plus rapidement possible.

---

- **Item de travail 4 : Conception / développement / test des micro-services :**

---

Les développements seront réalisés en méthodologie agile SCRUM. Une priorisation est faite sur le développement de la fonctionnalité de géolocalisation qui devra être implémentée le plus rapidement possible.

---

- **Item de travail 5 : Refonte du Front-End :**

---

Les technologies côté client seront uniformisées avec le framework Angular. L'outil est déjà implémenté dans la majorité des développements front. Il sera généralisé pour les futurs développements côté client.

---

- **Item de travail 6 : Supervision continue de l'architecture :**

---

La supervision du travail de migration débutera dès le début du projet. La partie technique sera supervisée via les outils de monitoring fournis par la pipeline de d'intégration et de déploiement continue. La partie organisationnelle sera assurée par le scrum master et par les membres de la structure de gouvernance du projet.

---

## 7 Spécifications pour l'implémentation

---

### 7.1 Item de travail 1: Phase de Lancement / Communication

---

Dans cette étape, nous allons mobiliser les acteurs et communiquer les guidelines du projet . Une réunion de lancement sera prévue.

### 7.2 Item de travail 2: Mise en place de la méthodologie de travail

#### 7.2.1 Mise en place méthodologie agile SCRUM

Une période sera prévue pour définir les rôles au sein d'une équipe agile utilisant la méthodologie SCRUM. Les éléments suivants seront définis:

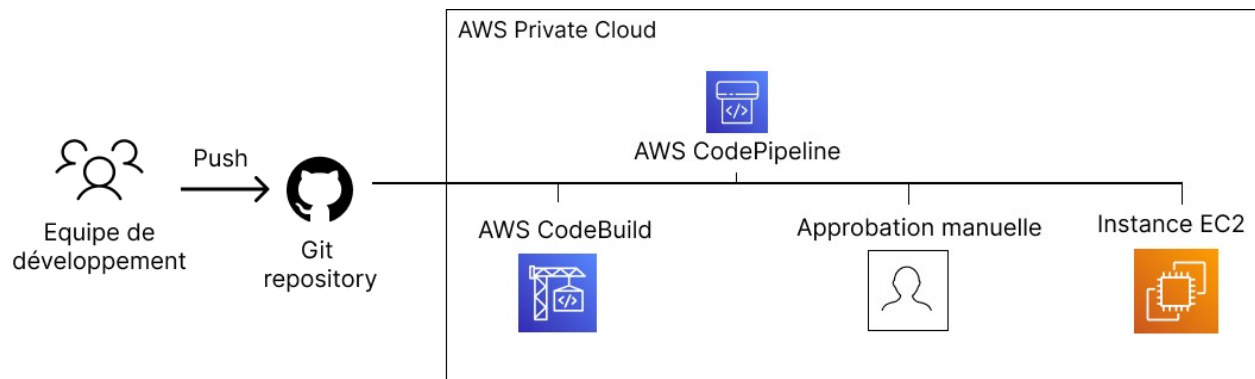
- Scrum master
- Product Owner
- Les responsabilités au sein de l'équipe de développement
- Les durées de sprint
- Durée des revues de sprint

## 7.2.2 Mise en place de la pipeline d'intégration et de déploiement continu 'CI/CD)

La pipeline CI/CD permettra à l'équipe technique de fournir des versions, mises à jour plus fréquemment sans compromettre la qualité de la plateforme Foosus. Cette pipeline présente les avantages suivants :

- Les fonctionnalités pourront être déployées plus rapidement sans interruptions de service
- **Les risques de bug, de pannes seront diminués** car le code déployé est automatiquement exécuté et testé dans un environnement de test et de pré-production avant le déploiement. Le code sera donc de meilleure qualité.
- Le temps de revue des livrables sera diminué car les équipes déploieront des légères incréments tous les jours. L'équipe pourra non seulement s'assurer que tout le monde travaille sur les mêmes bases, mais aussi d'accélérer la révision du code et de faciliter l'intégration des modifications.

Le schéma ci-dessous présente les composants et le processus à mettre en place :  
(Présentation avec services AWS pouvant être substitué avec un autre fournisseur cloud disposant des mêmes principes technologiques)



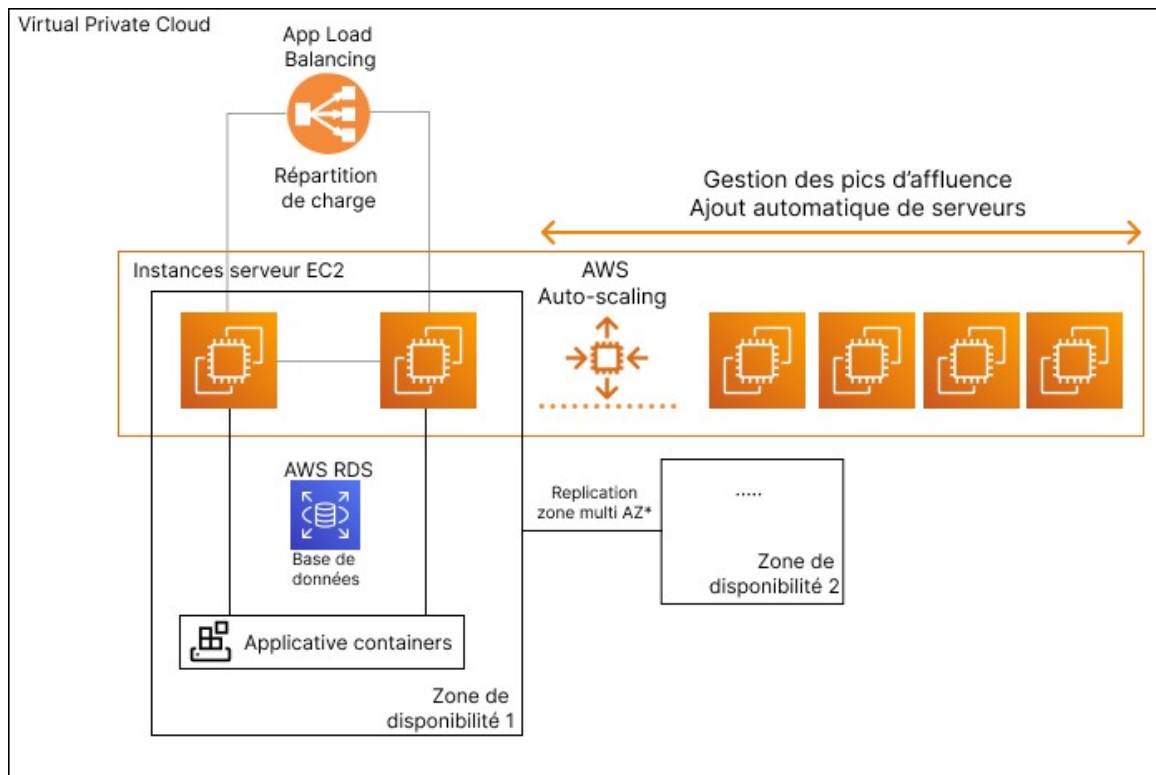
Le processus d'intégration et de déploiement continu est le suivant :

1. L'équipe de développement travaille sur un item de l'architecture.
2. L'équipe technique envoie des modifications (push) au dépôt Git .
3. Les services cloud AWS détectent un changement étant rattachés aux dépôts Github. Le service AWS Codebuild construit un environnement de développement en utilisant la technologie de container.
4. Le service AWS Code pipeline installe toutes les dépendances liées aux applications et lance tous les scripts de test réalisés par l'équipe de développement.
5. Si tout est fonctionnel, une approbation manuelle est requise pour envoyer les modifications directement sur les serveurs de production sans interruption de service.

## 7.3 Item de travail 3: Migration de l'architecture existante vers le cloud

L'architecture actuelle devra être migrée vers une infrastructure extensible. Le schéma ci-dessous représente la couche infrastructure respectant le principe stratégique :

(Présentation avec services AWS pouvant être substitué avec un autre fournisseur cloud disposant des mêmes principes technologiques)



Le schéma ci-dessus présente les composants clés de l'architecture répondant au principe stratégique d'une architecture extensible :

- **Application Load Balancing** : Le Load Balancing d'AWS répartit automatiquement le trafic entrant vers les applications sur plusieurs instances serveur EC2. Un nom d'hôte DNS est attribué et toutes les demandes clientes sont envoyées à celui-ci et déléguées à un pool d'instances Amazon EC2.
- **AWS Auto-scaling** : Ce service AWS permet d'ajouter automatiquement des instances EC2 supplémentaires en réponse à une augmentation du trafic et, une fois que celles-ci deviennent superflues, elles sont automatiquement arrêtées. Cela permettra de gérer complètement les pics d'affluence sans interruptions de service qui posent problème à l'heure actuelle.
- **AWS Relationnal Database Service** : Amazon Relational Database Service est un service web qui facilite l'exécution de bases de données relationnelles dans le cloud. Dans le contexte du développement d'applications tolérantes aux pannes et à haut niveau de disponibilité, Amazon RDS propose plusieurs fonctions visant à renforcer la fiabilité des bases de données critiques. Les sauvegardes automatiques de votre base de données



garantissent une restauration à un instant dans le passé de votre instance de base de données. Amazon RDS sauvegarde votre base de données ainsi que les journaux de transactions, et les conserve durant la période de rétention spécifiée. Cette fonction est activée par défaut.

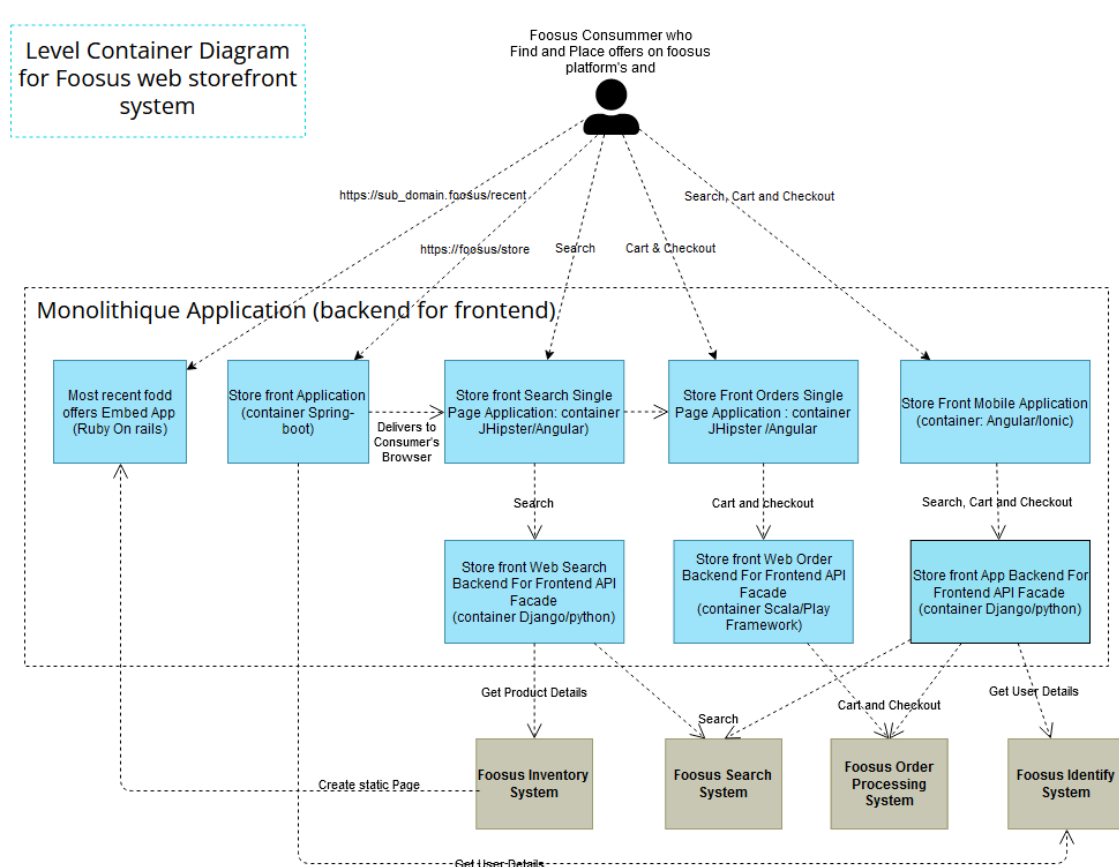
- **Zone multi AZ (Availability Zone)** : Des instances redondantes seront mises en place pour chaque niveau (web,application et base de données) des applications dans des zones de disponibilité distinctes de façon à créer une solution multi site. L'objectif est de disposer d'une copie indépendante de chaque pile d'applications dans au moins deux zones de disponibilité pour garantir un haut niveau de disponibilité.
- **Applicative containers** : Cet élément fait référence à tous les containers applicatifs présents au sein de l'architecture de Foosus. Dans un premier temps, les applications seront migrées vers les instances serveur EC2 en l'état. Les items de travail détailleront les modifications à apporter aux applicatifs.

## 7.4 Item de travail 4 : Conception / développement / test des micro-services

La conception, le développement et test des applications à refondre sous une architecture micro-service seront réalisés dans un environnement test de façon incrémentale. **Une priorisation sera faite sur le composant de géolocalisation** car l'entreprise souhaite intégrer cette fonctionnalité le plus tôt possible. Les applications seront intégrées, testées, déployées de façon continu.

### 7.4.1.1 Architecture existante

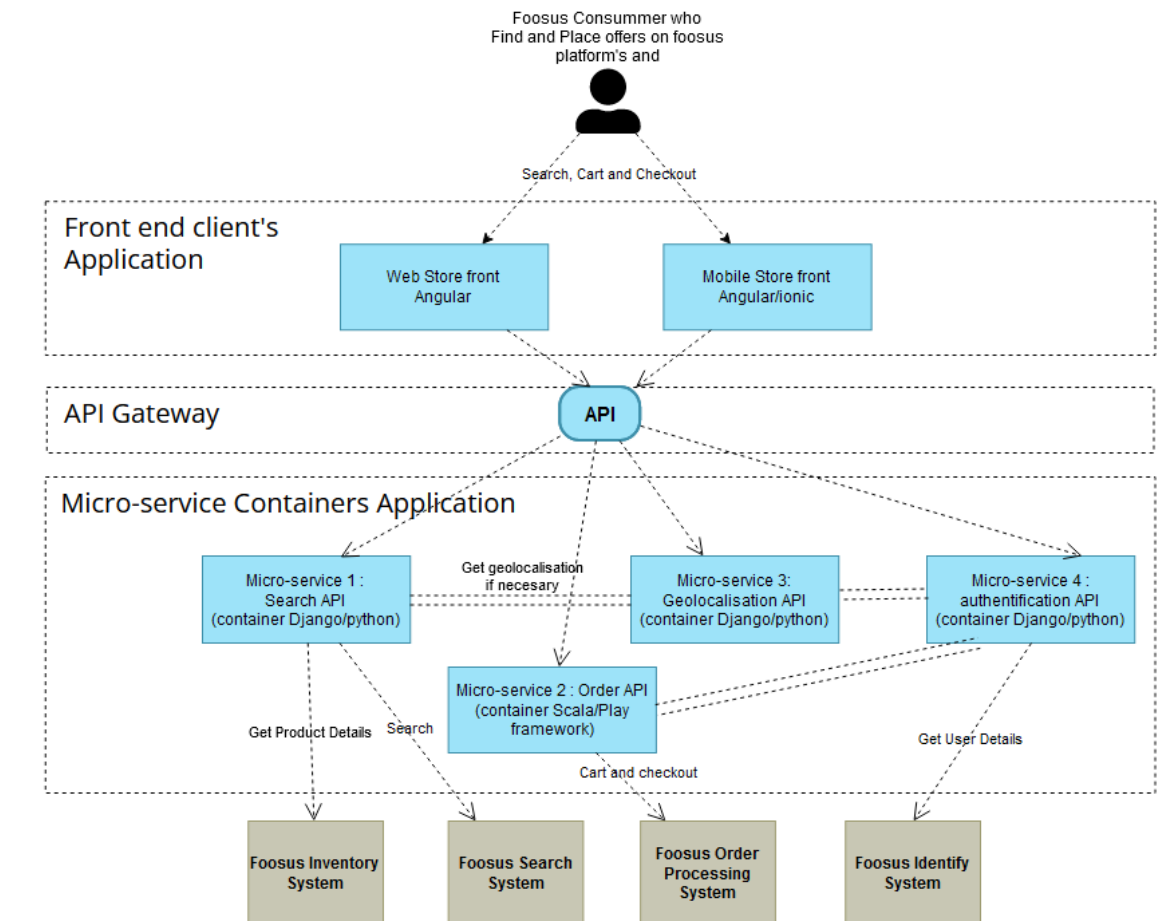
L'architecture actuelle utilise le modèle de design de backend pour le frontend et la propagation du comportement. (voir schéma ci-dessous) :



Chaque conteneur représente des backends de commandes StoreFront. Elles sont en pratique des applications monolithiques devenues complexes à maintenir pour les équipes de Foosus.

#### 7.4.1.2 Architecture cible (orientée micro-service) :

Le but ici étant d'adopter **une architecture modulaire** en créant **des micro-services** qui décomposeront les besoins business en plusieurs applications respectant les principes SOLID.



Il est judicieux de comprendre les avantages et les défis que posent les micro-services. Voici les avantages :

- Les services peuvent évoluer indépendamment en fonction des besoins des utilisateurs.
- Les services peuvent être mis à l'échelle indépendamment pour répondre à la demande de l'utilisateur.
- Au fil du temps, les cycles de développement sont plus rapides, car les fonctionnalités peuvent être mises en production plus rapidement.
- Les services sont isolés et sont plus tolérants aux défaillances.
- Un service unique qui échoue n'entraîne pas la défaillance de l'ensemble de l'application.
- Les tests deviennent plus cohérents grâce au développement piloté par le comportement.

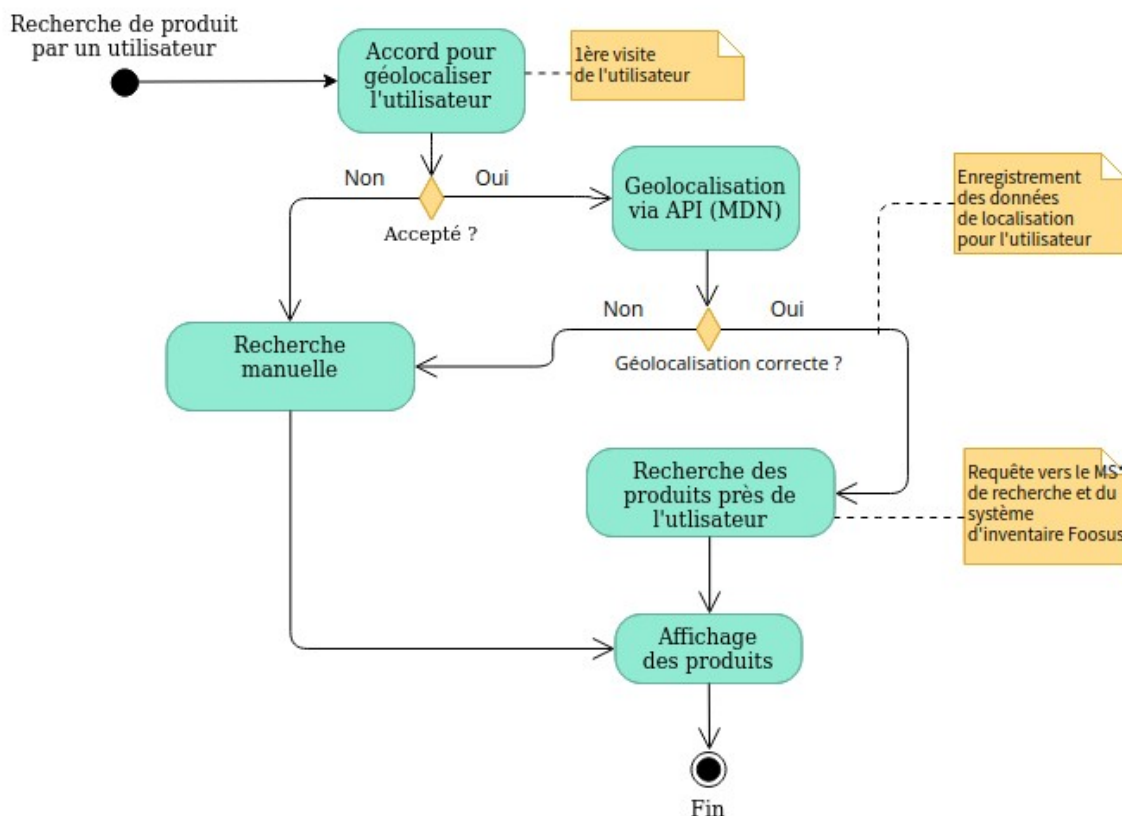
### 7.4.1.3 Implémentation du composant de géolocalisation

L'implémentation de la fonctionnalité de géolocalisation est à réaliser en priorité selon les besoins business de Foosus. Les clients seront mis en relation avec les offres de produits de fournisseurs locaux via leur emplacement géographique.

La technologie préconisée est l'**API Géolocalisation** open-source fournit par la communauté de développeur Mozilla FireFox. Elle sera mise à disposition pour la plateforme web Foosus et l'application mobile. Elle présente les avantages d'être gratuite, compatible multi-navigateurs, maintenue depuis plus de 20 ans par une grande communauté de développeur garantissant le succès du navigateur Firefox.

Elle présente une contrainte car elle utilise **la localisation par adresse IP**. Ce qui signifie que les utilisateurs utilisant un VPN ne pourront pas bénéficier de la géolocalisation sur place. Cependant, la nouvelle fonctionnalité de géolocalisation devra permettre de sauvegarder les localisations lors des recherches manuelles pour chaque utilisateur afin de faciliter les prochaines.

Le diagramme du process de géo-localisation est le suivant :



Le processus de géo-localisation des utilisateurs sera le suivant :

1. Le client se rend sur le site pour chercher des produits
2. Il accepte la géo-localisation ou non (recherche adaptée à sa position ou manuelle en cas de refus)

3. Les informations de géo-localisation seront enregistrées ou non pour chaque client s'il valide la bonne géolocalisation de son emplacement.
4. Le système de recherche devra chercher les produits dans un rayon proche du client
5. Le service de recherche affichera les produits aux clients

Afin d'être conforme aux réglementations des données privées (RGPD,CNIL), un consentement sera demandé au client pour permettre sa géo-localisation.

## **7.5 Item de travail 5 : Refonte du Front-End**

Parallèlement au développement des micro-services par l'équipe de développeur back-end, les développeurs/intégrateurs Front et Designer travailleront en étroite collaboration afin de refondre les interfaces de la plateforme Foosus sous une technologie unique avec le framework Angular. En effet, c'est la technologie la plus utilisée et la plus maîtrisée au sein de l'architecture actuelle. Le principal objectif sera de détacher la logique métier qui a été intégrée dans les applications monolithiques existantes. (container Java-spring et Ruby on rails voir schéma [#7.4.1.1Architecture existante](#)).

## **7.6 Item de travail 6 : Supervision continue de l'architecture**

La supervision du projet se déclenchera dès la phase de migration de l'architecture existante afin de contrôler son bon déroulement. Les métriques et KPI sont listés dans les contrat de conception /développement et d'architecture business.

## 8 Standards pour l'implémentation

---

Les normes et bonnes pratiques suivantes seront respectées lors de l'implémentation de la nouvelle architecture :

- Principes de développement SOLID
- Normes ISO 9001 : Sécurité des systèmes d'informations
- Normes ISO 31000 : Management du risque
- Respect des bonnes pratiques de mise en place de micro-services
  - Monitoring
  - Contrôle des appels API – limite IP
  - Sécurité multi-couche
  - Mise en place de versionning d'api
  - Éviter les communications excessives entre les micro-services (Maximum 3 appels)
- norme ISO 14 006 – 14 062 de l'architecture (référence à l'AGIT – alliance Green IT)

## 9 Conditions requises pour l'interopérabilité

---

Dans un premier temps, la plateforme existante de Foosus devra être migrée sur l'architecture cloud en l'état actuel. Ceci corrigera les problèmes d'interruptions de service dû aux pics d'affluence rencontrés sur l'architecture actuelle. Cela sera géré par les services de **load balancing** et d'**auto-scaling** fournis par le service cloud.

Par la suite, pour migrer vers une architecture micro-service, **une priorisation sera effectuée sur le développement de l'outil de géolocalisation** qui permettra de mettre en lien les produits des fournisseurs avec les clients en fonction de leur position.

# 10 Conditions requises pour le management du service IT

---

L'équipe devra mettre en place un environnement de travail agile et réaliser la migration de l'architecture avec une méthode DevOps. Pour cela, un pipeline de d'intégration et de développement continu devra être préalablement mise en place pour livrer les composants de la nouvelle architecture de façon incrémentale.

Afin d'assurer le bon déroulement du travail d'architecture, les points suivants liés au management du service IT devront être respectés :

- Management en équipe agile
- Attribution des rôles (Product owner, scrum master-équipe de dev, testeurs, )
- Définition des sprints (durée)
- Définition de l'approbateur avant mise en production (CI/CD)
- Des astreintes seront nécessaires lors de la migration de l'architecture existante vers le cloud afin d'assurer une reprise d'activité dans les plus brefs délais en cas d'incident.
- L'équipe de réalisation devra s'assurer via les outils de CI/CD de la non-régression des développements.
- La mise en production devra être réalisée du Lundi au Jeudi et évitée le Vendredi par sécurité au cas ou du code défectueux passeraient à travers les tests lors de déploiement.

# 11 Contraintes

---

## 11.1 Contraintes financières

Le projet de développement de l'architecture est approuvé pour un coût de 50 000 \$. L'architecture doit permettre d'obtenir le meilleur rapport qualité coût.

## 11.2 Contraintes organisationnelles

L'architecture peut inclure de nouveaux composants personnalisés ou des composants du commerce pour favoriser la flexibilité, la stabilité et l'extensibilité.

## 11.3 Contraintes temporelles

Une période de 6 mois est prévue pour définir l'architecture et préparer un projet de suivi afin de développer un prototype.