



# Contrat de Conception et de Développement de l'Architecture

---

- *Projet : XXXX*
- *Client : YYYY*
- *Préparé par : ZZZZ*

## Table des matières

---

1. Objet de ce document
2. Introduction et contexte
3. La nature de l'accord
4. Objectifs et périmètre
5. Description de l'architecture, principes stratégiques et conditions requises
6. Livrables architecturaux
7. Plan de travail commun priorisé
8. Plan de communication
9. Risques et facteurs de réduction
10. Hypothèses
11. Critères d'acceptation et procédures
12. Procédures de changement de périmètre
13. Calendrier
14. Phases de livrables définies
15. Personnes approuvant ce plan

# Objet de ce document

Ce document vise à former un accord avec les fonctions Développement et Design dans le cadre de ce projet d'élaboration d'une nouvelle architecture pour l'entreprise Foosus. Ce document précisera la direction que va prendre l'architecture, les phases du plan de travail ainsi que les livrables qui seront fournis à l'issue de ces phases. Il comprend également des métriques qui permettront de mesurer l'efficacité de la solution une fois déployer.

## Introduction et Contexte

---

L'entreprise Foosus propose un service de mise en relation de producteur alimentaire avec des consommateur via un site web ou une application mobile. Foosus prônant une alimentation durable, il souhaite développer l'alimentation locale en prenant en compte la distance séparant les consommateurs des producteurs afin de priorisé la mise en avant de producteurs à proximité du consommateur.

Lors du développement de leur solution, Foosus, alors constitué de plusieurs équipes de développement coexistantes à choisis de laisser les équipes expérimenter dans leurs technologie de prédilection. Cela se traduit par une implication des équipes de développement au détriment de la stabilité de la solution et de l'efficacité de production de cette dernière.

Bien que l'objectif initial d'implication des équipes est une réussite, le développement des nouvelles fonctionnalités est constamment interrompue par le besoin de corriger des problèmes de déploiement des fonctionnalités précédentes interrompant la disponibilité de service.

Une refonte de l'architecture existante est alors apparu nécessaire afin de fiabilisé les solutions et reprendre un rythme d'innovation constant.

## La Nature de l'accord

---

Cet accord à pour objectif de définir les besoins techniques de l'architecture ainsi que la direction que celle-ci doit tenir pour les remplir.

Les parties prenantes spécifié dans ce document sont, si elles acceptent l'accord, tenu de collaboré dans les phases de l'élaboration de l'architecture afin de contribué au développement de cette dernière. Elles seront tenu informé des avancées réalisées, de la progression parmi les phases du plan de travail, ainsi que de la nécessité de leur implications dans chaque phase qui en aura besoin. Si leur implication lors d'une phase est nécessaire, elle seront tenu de participer au réunion de suivi hebdomadaire s'inscrivant dans la méthodologie Agile qui sera utilisé pour mener à bien ce projet.

Cet accord vaut pour la durée de ce projet d'élaboration d'une nouvelle architecture, ce qui représente une période de six mois.

# Objectifs et périmètre

## Objectifs

---

Les objectifs techniques de ce Travail d'Architecture sont les suivants :

### Fiabilisation de la solution

L'un des principaux maux de l'ancienne solution, si ce n'est le plus critique, était l'apparition de problème bloquant lors du déploiement de nouvelle fonctionnalité. Si Foosus souhaite retrouver son image de société innovante, la solution doit être alimentée de nouvelles fonctionnalité de manière fréquente et régulière. La diminution de ces problèmes bloquant permettra également de tendre vers le deuxième objectif.

### Assurer la continuité de service

Les problèmes bloquant nécessitaient un effort important de la part des équipes de développement afin de les résoudre étant donné qu'ils interrompaient le service. Cette interruption peut engendrer une perte de confiance chez les utilisateurs, c'est pourquoi des mesures seront prise pour éviter de rompre la continuité du service.

### Favoriser le développement de nouvelles fonctionnalité

En modifiant l'environnement de développement et celui de production, nous allons pouvoir mettre en place les outils dont les développeurs auront besoin pour imaginer et développer de nouvelles fonctionnalité de manière fluide et régulière.

## Périmètre

---

Ce travail d'architecture porte sur les services traitant tout le fonctionnel et le visuel du site web, ainsi que de l'application mobile du produit de Foosus. Les services traitant les opérations (Gestion d'inventaire, gestion de la recherche, traitement des commandes, etc) sont en dehors du cadre de ce projet.

En outre, ce projet viendra également modifier les procédures de développement qui étaient en vigueur jusqu'à lors.

## Parties prenantes, préoccupations et visions

Le tableau suivant montre les parties prenantes qui utilisent ce document, leurs préoccupations, et la façon dont le travail d'architecture répondra à ces préoccupations par l'expression de plusieurs visions, ou perspectives.

Partie prenante	Préoccupation	Vision
Natasha Jarson (CIO)	Fiabilisation de la solution, technologie employée, structure de la solution, maintenabilité et évolutivité, outil de déploiement	Micro-service, conteneurisation, répartition de la charge
Pete Parker (Engineering Owner)	Technologie employée, structure de la solution, maintenabilité et évolutivité	Micro-service, conteneurisation, unification des technologies
Jack Harkner (Operations Lead)	Environnement nécessaire au fonctionnement de la solution	Conteneurisation, répartition de la charge
Frontend Squad	Interface utilisateur	Unification des technologies, micro-service
Ruby Development Squad	Changement de technologie	Unification des technologies, formation aux nouvelles technologies, micro-service
Identity Management Squad	Système de rôles	Unification des technologies, micro-service
Backend Applications Squad	Fonctionnel traitant les opérations	Unification des technologies, micro-service
Ops Squad	Environnement nécessaire au fonctionnement de la solution	Conteneurisation, répartition de la charge, redémarrage de service

# Description de l'architecture, principes stratégiques et conditions requises

## Description

---

Voici les points importants de la nouvelle solution de ce projet d'architecture :

### Unification des technologies

L'un des principaux maux de l'ancienne solution est l'importante divergence technologique, chaque équipe de développement se focalisant sur sa technologie de prédilection dans le but de produire des nouvelles fonctionnalités. Cela implique que les recherches d'une équipe ne peuvent bénéficier aux autres. De plus si un problème survient, seul l'équipe ayant développé la fonctionnalité concernée est à même de le corriger.

### Conteneurisation

La disponibilité de la solution étant une cible prioritaire, si ce n'est la cible la plus importante, de la nouvelle architecture, ce principe apportera plusieurs aspects intéressants qui plus est dans une dynamique d'évolutivité.

L'architecture cible sera constituée de services et micro-services, chacun d'eux contenant une fonctionnalité. Nous pouvons tirer parti de ce compartimentage en utilisant des conteneurs (environnement virtuel exécutant le service/micro-service), cela apporte trois avantages majeurs :

- Mise à l'échelle horizontale : Chaque conteneur étant un environnement virtuel indépendant, il sera possible de démultiplier les instances d'un même service en fonction de la charge d'utilisateur à chaque instant.
- Relance automatique : Dans le cas où une instance d'un service échoue dans sa tâche, se retrouvant indisponible, les utilisateurs affectés pourront être redirigés vers une nouvelle instance de ce service, tout en émettant un rapport d'erreur qui sera étudié par les équipes de développement.
- Environnement de test : Un environnement parallèle à l'environnement de production pourra être constitué avec des instances des services déjà présents dans la solution afin de mesurer l'impact d'un nouveau service dans l'architecture cible.

## Formation des équipes

Du fait des précédents points, les équipes auront besoin d'acquérir les compétences leurs permettant de mettre en œuvre cette architecture. Cette montée en compétence est préférable au remplacement des équipes existantes due à l'implication de ces dernières dans le produit que propose Foosus.

Aussi, suivant comment l'étape de définition de la stack technologique se déroulera, il se peut qu'une équipe ait déjà les compétence requise. Si cela arrive nous pourrons mettre en place des équipes favorisant l'apprentissage des employés novices par le biais des membres de l'équipe compétente. Une telle pratique favorisera le partage de connaissance, améliorera la cohésion des équipes et réduira le temps nécessaire à la formation.

## Réduction de la maintenance nécessaire

Pour répondre à ce point, deux axes seront implémenté.

Le premier vise à unifier la solution mobile et la solution web dans le but de diviser par deux les efforts de développement/maintenance. Cela pourra être effectué en affichant le contenu du site dans l'application, le site étant déjà pensé pour être Reactif et Responsif il saura s'adapter à toutes les tailles d'écran et d'appareil.

Le deuxième quant à lui vise à réduire la génération de problème dès la conception des services. En ce sens une procédure de qualité sera développer, elle sera constitué entre autres de test unitaire vérifiant le fonctionnement des éléments internes d'un services, ou bien d'une procédure de déploiement dans un environnement de test pour mesure l'impact positif ou négatif du service vis à vis de la solution. Le but étant de minimisé l'effort alloué à chaque service une fois celui-ci déployé sur l'environnement de production.

## Renforcement de la sécurité

L'ajout de fonctionnalité tel que la géolocalisation implique le traitement de données sensibles, ces données doivent être fortement protégées. De plus au vu de l'ambition de Foosus à évoluer vers diverses régions, si un utilisateur se trouve être un ressortissant européen, ce traitement devra répondre aux RGPD (règlement général de protection des données).

Plusieurs points devrons donc êtres mis en place. Tout d'abord, les communications entre services, ainsi que celles entre la solution et l'utilisateur, devront être chiffré afin d'éviter tout écoute de ces flux de données par un tiers.

Ensuite, la bases de données devra elle aussi être chiffrer afin d'éviter que, si un tiers parvient à extraire une portion de cette dernière, il soit dans l'incapacité de l'exploiter.

Enfin, le traitement de ces données devra être identifier, documenté et porté à la connaissance des utilisateurs. Ces derniers pourront également demandé à ce que lesdites données soit supprimés.

## Principes stratégiques

---

Nous optons pour une conteneurisation des services dans la mesure où l'objectif principal de la nouvelle architecture est de proposer une disponibilité de service maximal.

Pour ce qui est de la gestion du projet, nous allons utiliser la méthodologie Agile tout en s'inspirant des principes du Lean. Nous allons donc découper les étapes de travail en plusieurs sous-tâches de courtes durées avec des points de suivie hebdomadaires.

En accompagnement de cette méthodologie Agile, nous allons mettre en place une CI/CD (Intégration Continue et Livraison Continue). L'utilité d'un tel processus est lié au besoin de l'assurance qualité puisqu'il permet de déclencher des séries de tests à chaque ajout au répertoire de code partagé effectué par un développeur. De plus, il permet d'effectuer de manière automatisé les tests de déploiement sur un environnement miroir de celui de production des nouvelles fonctionnalités.

Les principes du Lean visent entre autres à éliminer le gaspillage et valoriser les individus. Pour les mettre en application nous allons commencer par l'unification des technologie, cela va réduire si ce n'est éliminer les doublons de recherches sur une même fonctionnalité, ainsi que permettre de réutiliser le code que d'autres équipes auront développer. La valorisation des individus quant à elle tient dans le fait que nous privilégions le développement des compétences de nos équipes existantes qui ont conçu et développé le produit de Foosus depuis ses débuts plutôt que le recrutement d'expert externes.

## Référence aux Conditions requises pour l'architecture

---

Avec le déploiement d'une constellation de conteneur nous allons pouvoir mettre en place, en parallèle des tests unitaires, des déploiement dans des environnement reconstitué à l'identique pour en mesurer l'impact. Une telle procédure sera un ajout majeur pour la fiabilité de la solution. Le fait de pouvoir équilibrer la charge ou de pouvoir monter des instances de rechange en cas de défaillance contribuera quant à lui à la disponibilité de service.

# Livrables architecturaux

## Développement de l'architecture

---

Dans le but de développer l'architecture, les services existants devront être reproduits avec la technologie adéquate provenant de la nouvelle stack technologique, s'il ne sont pas déjà définis avec cette dernière.

Ensuite, plusieurs services de traitement devront voir le jour pour répondre au besoin de tous les objectifs. Le service qui devra être une priorité lors de la conception est l'orchestrateur. Il s'agit du service qui sera au cœur de l'architecture puisqu'il analysera les requêtes afin de demander les traitements adéquats aux services concernés, il sera la pierre angulaire de l'architecture.

Par la suite, l'architecture devant être basée sur les rôles, le service de gestion des rôles sera nécessaire dès le début puisqu'il viendra signaler quelles vues devront être retournées au client de l'utilisateur selon son rôle.

Comme l'architecture devra prendre en compte la géolocalisation des utilisateurs, plusieurs services vont devoir être conçus. Le premier sera celui capable de récolter les données de localisation des utilisateurs et de les chiffrer pour les stocker dans la base de données. Ensuite un service permettant de définir la proximité/distance entre les consommateurs et les fournisseurs sera mis au point. Une fois ces deux services développés, un dernier permettant, selon un filtre défini par l'utilisateur, de trier et retourner la liste des fournisseurs proches sera nécessaire pour compléter l'ajout de la fonctionnalité.

En outre, la solution aura besoin d'un proxy inverse devant l'orchestrateur pour équilibrer la charge en demandant la création de nouveaux conteneurs au besoin. La sécurité étant importante au vu des données qui seront traitées, l'envoi de requête au proxy inverse ne sera autorisé qu'une fois la connexion à un VPN établie. Cela nécessitera donc un service permettant d'effectuer cette connexion devant le proxy inverse.

---

## Mesures de l'architecture cible

---

La fiabilité de l'architecture est importante, c'est pourquoi la première mesure à prendre est d'unifier toutes les technologies utilisées dans le développement de services en une seule. Comme précisé dans la description de l'architecture, l'intérêt de cette mesure est de valoriser toutes les recherches effectuées dans le sens où elle pourrait servir à différentes équipes, d'augmenter la force de frappe disponible pour de la maintenance si un problème survient et enfin de favoriser la montée en compétences des équipes puisqu'elles pourront s'entraider pour résoudre les obstacles qu'elles pourraient

---



rencontrer.

Toujours pour améliorer la fiabilité, une procédure d'assurance qualité va être mise au point. Elle sera découpée en plusieurs étapes, la première consiste en une analyse de code statique. Grâce à la mise en place d'une CI/CD, nous pouvons automatiser des processus tel que celui-ci dans le but d'améliorer la qualité du travail produit. Une analyse de code statique permet de vérifier que ce qui est développé respecte des règles préétabli, comme par exemple des conventions de nommages.

Le respect de ces règles rendra le code clair et concis pour tout les employés, favorisant le partage de connaissance, et réduisant les efforts de maintenance.

La deuxième étape de la procédure d'assurance qualité est l'évaluation de test unitaire validant ou non le fonctionnement des fonctionnalités d'un service. Ces tests seront lancés automatiquement à la suite de l'analyse de code statique.

La troisième étape sera une relecture par les pairs, un employé développant une fonctionnalité, devra faire relire son travail par un autre employé dans le but d'éviter un effet tunnel et être sûr de couvrir tous les aspects demandés.

Pour terminer, la quatrième étape avant un déploiement dans l'environnement de production, un déploiement dans un environnement de test, identique à celui de production, sera effectué afin de mesurer l'impact du service dans l'architecture. De cette façon les éventuels problèmes critiques qui n'auraient pas encore été identifiés pourront être résolus.

---

## **Livraison de l'architecture et métriques business**

---

Pour livrer l'architecture cible, un processus de livraison continue sera mis en place. En accord avec la gestion Agile, le travail d'architecture sera découpé en phases à l'issue desquelles seront produits des livrables, que ce soit des documents expliquant les résultats d'une formation ou bien un service complet ajoutant une fonctionnalité. Ces livrables seront fournis graduellement tout au long du projet.

Au fur et à mesure que la nouvelle architecture sera déployée, nous pourrions observer les effets de cette dernière via plusieurs métriques business tel que : la création de nouveaux comptes en fonction de la région et de l'appareil utilisé, l'adhésion des producteurs alimentaires, ou encore le délai moyen de parution.

---

## Phases de livraison définies

---

Les livraisons surviendront lors de la complétion d'une activité de phase que vous retrouverez en détails sur la prochaine page. Voici un résumé des activités comprises dedans :

- Définition des conditions requises pour l'architectures et établissement des contrats avec les équipes cibles
- Définition de la stack technologiques
- Définition de la procédure d'assurance qualité
- Définition de l'environnement détaillé de la nouvelle solution
- Restructuration des équipes de développement
- Formations des équipes
- Création de l'orchestrateur
- Conversion des services existants
- Mise en place de l'ingress
- Renforcement de la sécurité
- Mise en place de la gestion par rôle
- Mise en place de la géolocalisation
- Mise en place de la centralisation des logs

# Plan de travail commun priorisé

## Définition du futur environnement

---

Cette première phase a pour but de poser les bases du projet, les multiples livrables qui en seront issus permettront de définir le nouvel environnement en effectuant les choix nécessaire à cet effet.

## Activités

- **Définition des conditions requises pour l'architectures et établissement des contrats avec les équipes cibles**

---

  - Les conditions spécifiques à cette architecture doivent être décrite dans un document propre afin de garder une trace des contraintes à respecter lors de l'établissement de la solution cible. Par ailleurs, un contrat d'architecture est nécessaire pour les équipes business, ainsi que celles de développement afin de leurs présenter la solution cible et les mesure nécessaire pour atteindre cette dernière.
  - Livrables :
    - Document de Spécification des conditions requises pour l'architecture :

---

      - Ce document décrit tout les conditions nécessaire que l'architecture devra contenir, qu'il s'agisse de conditions business, fonctionnel ou légal.
    - Contrat d'architecture avec les Utilisateurs Business :

---

      - Le contrat d'architecture permettant d'obtenir l'accord des utilisateurs business pour ce projet de travail d'architecture.
    - Contrat d'architecture avec les Fonctions Développement et Design :

---

      - Le contrat d'architecture permettant d'obtenir l'accord des fonctions Développement et Design pour ce projet de travail d'architecture. Il s'agit du présent document.
- **Définition de la stack technologiques**

---

  - Une étude sera menée afin déterminé quelles technologies devrait être privilégiés pour mettre en place la solution.
  - Livrable :

---

    - Document de Définition de la stack technologique :

---

      - Un compte rendu de l'étude détaillant la stack technologique futur de la solution, il couvrira les pistes de recherches qui auront été étudié, leur

pertinence ou non par rapport à la solution et la liste de celles qui ont été retenu en présentant les diverse avantages qui auront été préféré.

---

- **Définition de la procédure d'assurance qualité**

---

- Conceptualisation des solutions permettant d'éprouver d'une part la qualité des fonctionnalité développé, et d'autre part l'impact de l'intégration de ces dernières dans l'environnement de production.
  - Livrable :
    - Document de définition de la procédure d'assurance qualité :
      - Toute la procédure sera décrite et expliqué dans ce document afin de pouvoir la mettre en place de la manière la plus efficace qu'il soit. Une visualisation du processus sera également adjoint au document pour le rendre le plus clair possible.
- 

- **Définition de l'environnement détaillé de la nouvelle solution**

---

- La nouvelle solution nécessitant la mise en place d'un environnement spécifique usant de technologies avancé de développement opérationnel, requiert un plan détaillé contenant les technologies utilisées, les sécurités mises en places ainsi que le plan des flux réseaux du nouvel environnement. Une phase dédié à la conception de ces différents élément est donc nécessaire.
  - Livrable :
    - Document de définition de l'environnement de la nouvelle solution :
      - Ce document regroupe tous les choix qui auront été pris lors de cette phase. Il contient également les détails et explications quant au nouvel environnement requis pour l'architecture cible, ainsi que sa mise en application.
- 

- **Restructuration des équipes de développement**

---

- Les équipes de développement devront être réarrangé en fonction de la nouvelle stack technologique. Si une des équipes est déjà formé à ces technologique alors elles sera dispersé dans les futurs équipes afin que les employés ayant la connaissance de la technologique puissent apporté leur aide à ceux qui devront s'y former. À noté qu'une équipe de maintenance de l'ancien systèmes devra être formé pour prévenir d'une interruption de ce dernier, quand bien même aucune nouvelle fonctionnalité ne sera déployé dessus.
  - Livrable :
    - Plan de restructuration des équipes :
      - Ce document contiendra la nouvelle configuration des équipes en détaillant les choix derrière cette constitution. Les besoins en formations des différents employés seront également détaillé, pour une meilleur planification de la monté en compétences de ces derniers.
-

# Mise en place du nouvel environnement

---

Cette deuxième phase vise à installer le nouvel environnement et ainsi débiter la migration des service vers ce derniers.

## Activités

---

- **Formation des équipes**

---

- La nouvelle architecture va nécessité la mise en place de nouvelle technologie, que ce soit pour le développement opérationnel ou bien logiciel. En ce sens les équipes devront suivre les formations nécessaires à l'acquisition des compétences leurs faisant défauts.
  - Livrable :
    - Relevé des compétence des employés suite aux formations effectuées :
      - Ce document contiendra le détail des compétences des employés ayant suivi la formation. Avec ces compétences en main, les employés seront en mesures de développé et maintenir l'environnement de la nouvelle solution.
- 

- **Création de l'orchestrateur**

---

- Avant de pouvoir convertir les services existants, la structure pouvant accueillir les conteneurs doit être installé. L'orchestrateur devra posséder la capacité de pouvoir équilibrer la charge, et celle de relancer des conteneurs défaillant.
  - Livrable :
    - L'orchestrateur dirigeant l'environnement :
      - Cet environnement sera prêt à accueillir les services converti dans la prochaine activité.
- 

- **Conversion des services existants**

---

- Une fois l'environnement disponible, les services existant devront être traduit dans la bonne technologie si tel n'est pas déjà le cas. À l'heure actuelle ils sont construit comme de grosses application monolithiques différent de l'objectif de service/micro-service que l'on souhaite atteindre, c'est pourquoi nous allons devoir les convertir en séparant les fonctionnalités en de plus petit services.
  - Livrables :
    - Les nouveaux modèles de services :
      - Ces modèles permettront de déployer les services dans la constellation afin de créer les bases du nouvel environnement de la solution.
    - Les anciens services archivés :
-

- Dans le but de pouvoir rétablir rapidement la disponibilité des différents services dans le cas d'un échec critique au déploiement, il nous faut conserver les anciennes version monolithiques (bien qu'obsolètes) comme option de secours tant que le nouvel environnement n'est pas terminé.

---

- **Mise en place de l'ingress**

- Quand les différents services auront été transférés dans la nouvelle architecture, l'ingress permettant de faire le traitement des futures requêtes devra être conçu. Les autres services devront également être modifiés pour fonctionner avec ces derniers.

- Livrable :

- Service Ingress :

- Le service traitant les requêtes et formulant les demandes adéquates aux différents services.

---

- **Renforcement de la sécurité**

- En vue de la mise en place de la fonctionnalité de géolocalisation, les ajouts concernant la sécurité doivent être déployés. Cela concerne le chiffrement de la base de données et des communications entre services ainsi qu'entre la solution et les clients des utilisateurs.

- Livrables :

- Plan de sécurité :

- Ce document contiendra le détail des sécurités mises en place dans la nouvelle solution.

- Protocole SSL :

- Un protocole SSL sera ajouté à la solution dans le but de chiffrer les flux de données.
-

# Mise en place des fonctionnalités cœurs de la nouvelle solution

---

Cette troisième phase consiste à développer les nouvelles fonctionnalités porteuses des principes que Foosus souhaite insuffler à la solution.

## Activités

---

- **Mise en place de la gestion par rôles**

---

- La nouvelle architecture devant gérer les vues accessibles par les utilisateurs en fonction de leurs rôles (consommateur, fournisseur, administrateur), le service ajoutant cette fonctionnalité doit être conçu lors de cette étape.
  - Livrable :
    - Service de gestion des rôles :
      - Ce services sera capable, en fonction de l'utilisateur formulant les requêtes, de retourner la liste des vues pouvant être retourné.
- 

- **Mise en place de la géolocalisation**

---

- Le traitement de la géolocalisation peut, maintenant que les préparatifs sont terminé, être déployé. Pour cela, les différents services déjà décrits dans la section «Développement de l'architecture » devront être conçu lors de cette étape.
  - Une hypothèse levé par la fonctionnalité de la géolocalisation est que la base de données peut ne pas être adaptée au traitement de donnée de géolocalisation. Cela vaut également pour le service de recherche de Foosus qui ne prend pas en compte la géolocalisation pour proposer les fournisseurs les plus proches. Cela implique donc une modification de ce service et une potentielle migration de la base de données.
  - Livrables :
    - Service de collecte :
      - Ce service recueille les données de géolocalisation afin de les stocker après les avoir chiffrées.
    - Service de mise en relation :
      - Ce service établit les relations entre les utilisateurs et les fournisseurs afin de pouvoir déterminer les distances les séparant.
    - Service de filtre :
      - Ce service retourne la liste triée des fournisseurs devant être proposée à un utilisateur donné selon sa préférence quant à la distance le séparant desdits fournisseurs.
-

- Service de recherche modifié :

---

  - Le précédent service de recherche avec la prise en compte de la géolocalisation.
- Plan de migration si cette dernière est nécessaire

---

  - Document détaillé décrivant le processus de migration étape par étape.
- **Mise en place de la centralisation des logs**

---

  - Dans le but d'améliorer et de faciliter le développement de fonctionnalité ainsi que la maintenance, les logs des différents services doivent être rassemblés via l'aide d'un service dédié en un seul endroit. Plusieurs solution existent déjà, aussi une étude permettra de déterminer la solution la plus adapter à ce projet.
  - Livrables :

---

    - Études comparative des solutions de centralisation de logs :

---

      - Cette étude répertoria les différentes technologies existantes avec leurs avantages/inconvénients ainsi qu'une solution développé afin de déterminer la meilleure solution à adopter.
    - Service de centralisation des logs :

---

      - Ce service centralise les logs pour en extraire des statistiques utiles pour le développement ou la mesure de réussite d'une livraison.

# Plan de communication

## Évènements

Lors du déroulement d'une phase, des réunions hebdomadaires s'inscrivant dans la méthodologie Agile auront lieu. Elles auront pour but de permettre aux parties prenantes concerné par la phase en cours de suivre l'avancé des travaux, ainsi que d'émettre leur avis sur les choix effectués ou demander des précisions supplémentaires.

Dès la fin de chaque phases du plan de travail présenter ci-dessus, un événement sera produit en envoyant des emails aux parties prenantes concernés afin que ces dernières valide le travail effectué, les autres parties prenantes seront tenu informé du changement de phase dans le but de garder un suivi du déroulement du projet.

Lorsqu'un livrable d'une étape se trouve être un service, alors le déploiement de ce derniers dans l'architecture constituera un événement devant être communiqué aux partie



prenantes concernés.

## **Canaux**

---

Les documents seront transmit aux parties prenante concernées par mail, puis déposé dans un répertoire de version en ligne afin de garder une traçabilité des changements effectués aux documents. Pour les parties décisionnaires, des présentations pourront avoir lieu afin de détailler les documents dans le but d'une validation et, le cas échéant, déterminer la liste des points à redéfinir.

---

## **Formats**

---

Les documents seront stocker sous formes de fichier ODT sur le répertoire de version en ligne, ainsi qu'en format PDF, ce qui facilitera leurs distribution aux parties prenantes. Pour les services, il s'agira du code source sur le répertoire de version en ligne.

---

## **Contenu**

---

Dans le but de facilité la compréhension des informations transmises par les différent livrables de ce travail d'architecture, lors de la distribution par mail, un avant-propos présentera les grandes lignes des sujets traités dans les documents ci-joints. De même pour les services l'avant-propos permettra de résumé la fonctionnalité contenu dans ce derniers.

---

## **Rythme de communication**

---

Les réunions de suivis avec les parties prenantes concernées seront hebdomadaires, de même que les mail d'informations hormis dans le cas où un service est déployé auquel cas un mail détaillant ce service est transmis.

---

# **Risques et facteurs de réduction**

## **Structure de gouvernance**

---

Pour le développement de ce projet, les décisions finales seront prises par Natasha Jarson (CIO), cela étant dit les avis de Pete Parker (Engineering Owner) et Jack

Harkner (Operations Lead) seront consulté étant donné que Foosus tiens à impliquer ses équipes dans le développement de la future solution. Pour finir les équipes de développement et opérationnel seront tenu informé des choix qui auront été pris dans le cadre de ce projet.

## Analyse des risques

Voici la liste des principaux risque identifiés pour ce projet :

ID	Risque	Facteur de réduction	Propriétaire
R1	Fuite de données sensibles	Sécurisation des communication inter-service, chiffrement de la base de données	Jack Harkner (Operations Lead)
R2	Mauvaise attribution des privilèges (gestion des rôles)	Procédure de test exhaustif via l'assurance qualité	Pete Parker (Engineering Owner)
R3	Insertion d'erreur dans les données	Procédure d'assurance qualité	Pete Parker (Engineering Owner), Jack Harkner (Operations Lead)
R4	Bug nécessitant un effort trop conséquent pour les équipes suite à un déploiement	Procédure d'assurance qualité	Pete Parker (Engineering Owner)
R5	Génération de bug due à l'inexpérience des équipes	Procédure d'assurance qualité, répartition des employés expérimenté dans chaque équipe	Pete Parker (Engineering Owner)

R6	Allongement de la durée de formation d'une équipe de développement	Management du Lean	Pete Parker (Engineering Owner)
R7	Génération d'erreur dans les données suite à la migration	Anticipation des nouveaux besoin et mise en conformité de la base de données	Jack Harckner (Operations Lead)

Voici maintenant la répartition des risques présentés ci-dessus dans une matrice Probabilité/Impact :

		Impact				
Probabilité		Négligeable	Mineur	Modéré	Majeur	Critique
	Très probable				R4	
	Probable		R5	R3		
	Peu probable			R6	R7	R1, R2
	Improbable					

Cette matrice nous permet de mettre en valeur les principaux risques auquel nous devons faire attention pour ce projet. En l'occurrence, le risque R4 est préoccupant. Il rentre en contradiction avec la volonté de Foosus d'offrir une haute disponibilité de service.

Ce risque engendrerait une discontinuité de service le temps que les équipes soit en mesure de régler le problème, aussi la procédure d'assurance qualité doit être mise en place afin que ce genre de bug trop conséquent pour la force de frappe des équipes de développement soit repéré et traité avant le déploiement de la feature associé.

Les autres risques majeurs concernent la sécurité des données des utilisateurs. Les données de géolocalisation sont catégorisées comme données sensible, aussi une fuite de ces dernières serait critique pour l'entreprise. En ce sens la sécurité doit être renforcée via un chiffrement de la base de données ainsi que des flux de communications. De plus, la gestion des rôles doit être minutieusement développée et testée afin d'éviter toute erreur d'attribution de droit, autorisant des individus à accéder à des données qui ne leur seraient pas destinées.

# Hypothèses

---

Le tableau suivant résume les hypothèses pour cette Déclaration de travail d'architecture.

I D	Hypothèse	Impact	Propriétaire
1	Une équipe obsolète formé à de nouvel compétence sera plus efficace qu'une nouvelle équipe due à l'implication dans le produit des équipes actuelles	Durée de formation, implication des équipes	Natasha Jarson (CIO)
2	Unifier les technologies devrait faciliter les efforts de maintenance vue que toutes les équipes seront en mesures de s'ajouter à la force de frappe disponible en cas de besoin	Durée et effort de maintenance	Pete Parker (Engineering Owner), Jack Harckner (Operations Lead)
3	La base de données peut ne pas être adapter au traitement de données de géolocalisation	Besoin d'une migration vers une base de données adaptée	Pete Parker (Engineering Owner), Jack Harckner (Operations Lead)

# Critères d'acceptation et procédures

## Métriques et KPIs de l'État Cible de l'Architecture

De plus, les métriques suivantes seront utilisées pour déterminer le succès de ce travail d'architecture :

Métrique	Technique de mesure	Valeur cible	Justification
Délai moyen de parution	Une mesure du temps de développement de chaque fonctionnalité devra être mis en place dans la gestion du projet	De 3,5 semaines à <1 semaines	Pour rétablir la dynamique d'innovation recherché par l'entreprise, le délai de parution de fonctionnalité doit être réduit au minimum
Taux d'incidents de production P1	Une étude constante des rapports d'incidents permettra de mesurer leur fréquence d'apparition	De >25/mois à <1/mois	Réduire le taux d'incident permet d'augmenter la confiance des utilisateurs envers la solution, que ce soit les producteurs ou les consommateur. Ce facteur est essentiel puisqu'il va influencer sur l'attractivité des utilisateurs, il permettra également d'augmenter la production de fonctionnalité puisque moins de ressources seront alloué à la résolutions des incidents
Nombre d'adhésions utilisateur par jour	Étude de la base de donnée utilisateur afin de détecter les nouveaux arrivants	+10 %	L'un des objectifs de Foosus est de développer la croissance de l'entreprise, pour cela, traquer la quantité d'adhésion permet de se rendre compte de l'attractivité de la solution

Adhésion de producteurs alimentaires	Étude de la base de donnée producteur afin de détecter les nouveaux arrivants	De 1,4 par mois à 4 par mois	Dans le but de croître et d'attirer de nouveaux consommateur, Foosus à besoin de nouveaux producteurs afin de diversifié les offres de proximités de la solutions, aussi l'attractivité des producteurs doit être mesuré
--------------------------------------	---	------------------------------	--

## Métriques de livraison de l'architecture et du business

Métrique	Technique de mesure	Valeur cible	Justification
Disponibilité du nouveau services	Monitoring depuis l'orchestrateur pour traquer le taux de disponibilité	>90 %	Le succès d'une livraison dépend principalement du comportement du service dans l'environnement de production, aussi il doit être le plus disponible possible pour accueillir les nouvelles connexions
Ratio de requêtes réussite / totales	Monitoring en utilisant les logs pour traquer la quantité de rapport d'échec des requêtes	>0,95	Une fois que le service est disponible, il faut s'assurer qu'il est en mesure de répondre aux requêtes qui lui seront adressées

## Procédure d'acceptation

Le projet d'architecture devra être validé par chaque chef de section, en tenant informé les équipes correspondantes, avant d'être validé par le CEO. Pour ce faire un contrat business en plus de ce contrat technique sera fournis afin de présenter le projet de la meilleure manière aux parties intéressés, selon les éléments importants pour chacun. En amont de ces contrat, une déclaration de travail d'architecture à également été

rédigé.

# Procédures de changement de périmètre

---

Dans le but d'effectuer la transition entre l'architecture actuelle et l'architecture cible nous allons en emprunté une intermédiaires où les anciens services coexisteront avec les nouveaux.

Pour commencer, chaque service déjà existant n'étant pas développé via la technologie unifié devra être reproduit. Une fois que ces services de remplacements seront développé et testé, nous pourrons les intégrer comme suit :

1. Intégration du service remplaçant dans l'environnement de production en parallèle de son homologue.
2. Rediriger les nouvelles requêtes progressivement vers le service remplaçant.
3. Une fois les dernières requêtes attribué à l'ancien service étant terminé, mettre le service sous le statut d'archive.

L'état d'archive sera attribué à tout les anciens services jusqu'à la fin de la migration pour que, si jamais un problème se produit lors du déploiement/fonctionnement, nous puissions faire machine arrières et réutiliser les services archivés le temps de résoudre les problèmes survenus.

Ensuite, pour chaque service apportant une fonctionnalité inédite qui aura successivement passé les tests sera déployé comme suit :

1. Intégration du nouveau service dans l'environnement de production.
2. Redirection du flux de requêtes pour prendre en compte le service nouvellement ajouter : Si le service intervient dans un enchaînement de service, le flux est dévié pour transité par ce service avant d'atteindre la destination de la requêtes, exemple : un service de traductions de la locales de base vers celle choisi par l'utilisateur, qui intervient entre le service retournant le contenu de la page du site et l'envoie au client de l'utilisateur. Si en revanche le services ajoute une fonctionnalité en parallèle du reste, une nouvelle route pour le flux est créé mais aucune déjà existantes n'est modifié, exemple : un service de changement des préférences d'un utilisateurs en ce qui concerne la locales qui doit lui êtres affiché.

Vous pouvez retrouver en annexe de ce document une représentation de cette procédure, ainsi que les architecture d'origine et cible de ce projet.

# Conditions requises pour la conformité

Étant donné que la solution va recueillir des données sensibles sur ses utilisateurs, dans le cadre de la fonctionnalité de géolocalisation notamment, la solution devra respecter le RGPD (règlement général de protection des données) si au moins un utilisateur est un ressortissant de l'union européenne.

Dans le cadre de ce règlement, les données récoltées devront être identifiées, documentées et chiffrées. Par ailleurs les utilisateurs seront tenu informés de l'utilisation desdites données.

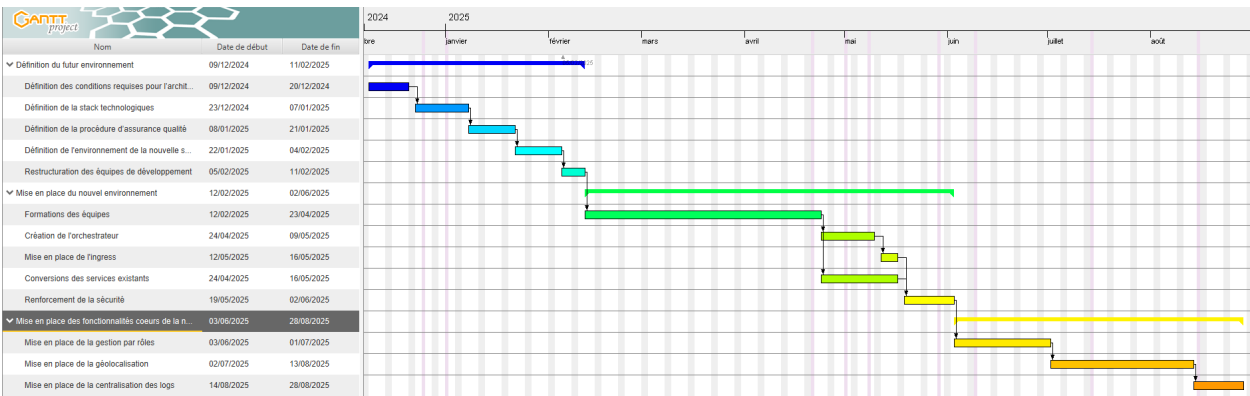
# Développement et propriété de l'architecture

Dans le cadre de ce projet d'architecture, le processus Agile et le management du Lean seront mis en place, il s'agit là d'une demande de Foosus. En raison de ces processus de management, les solutions opensources gratuites seront préférés à leurs homologues propriétaires sous licence. Le même principe s'applique aux équipes de développement existantes qui seront formées aux nouvelles technologies plutôt que de les remplacer par de nouvelles équipes n'ayant pas le niveau d'implication dans la solution des équipes actuelles.

La propriété de l'architecture appartiendra à l'entreprise Foosus, qui sera libre de la modifier et de l'utiliser à sa guise.

# Calendrier

Voici le diagramme de gantt de ce projet d'architecture :



Le projet est découpé en trois grande phases selon le plan de travail. Ici ces phases sont représenté comme jalon englobant les différentes activités.

La première phase, « Définition du futur environnement », est estimé à 45 jours.



La seconde, intitulé « Mise en place du nouvel environnement », est estimé à 75 jours.

Enfin, la troisième et dernière tâche « Mise en place des fonctionnalités cœurs de la nouvelle solution » est estimé à 60 jours.

## Phases de livrables définies

---

Lors de chaque activité composant les phases de projet, activité que vous pouvez retrouver représenté sur le diagramme ci-dessus, une livraison sera effectuée.

La gestion du projet suivant la méthodologie Agile, ces activité constitueront les sprints durant lesquelles des réunions de suivi seront effectué de manières hebdomadaires. C'est pourquoi il est pertinent de faire coïncider les livraisons avec les fins de ces sprints.

## Personnes approuvant ce plan

---

Valideur	Domaine de responsabilité	Date
Natasha Jarson (CIO)	Responsable des équipes de développement et opérationnel	

## Annexes :

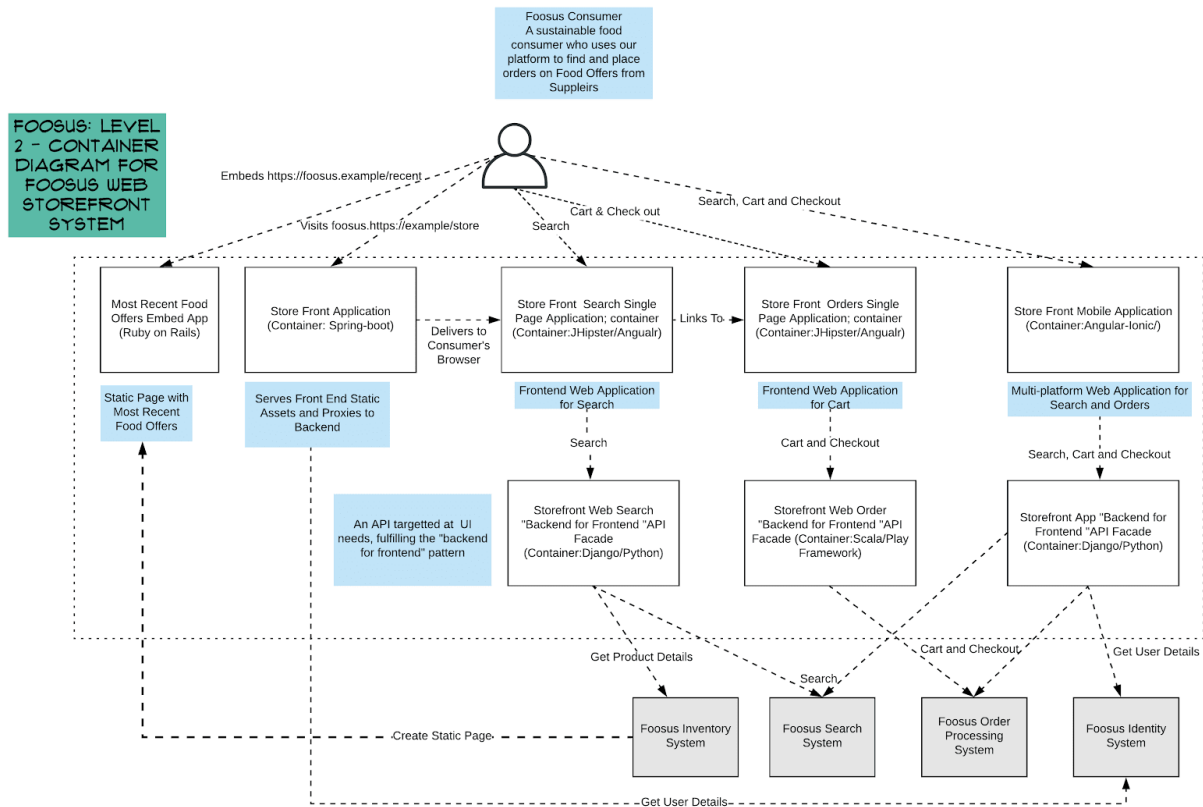


Figure 1: Architecture d'origine de la solution de Foosus

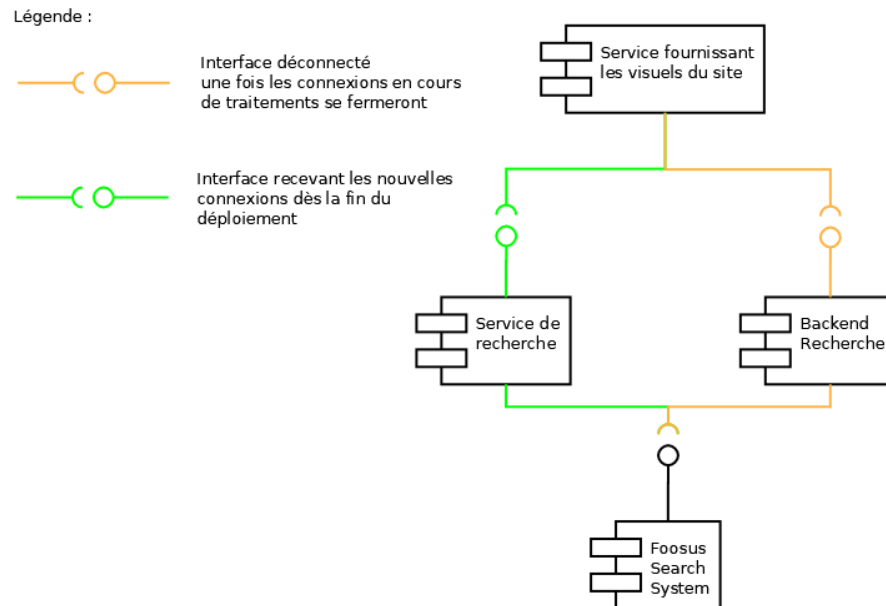


Figure 2: Remplacement d'un service

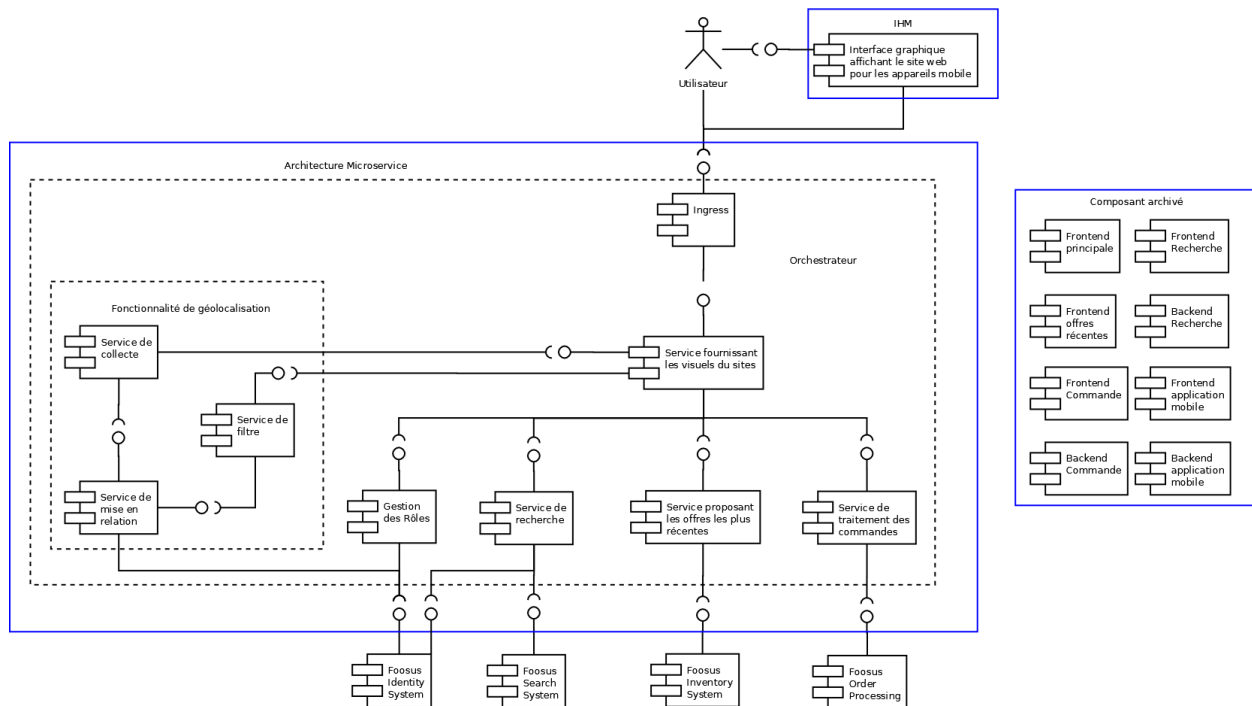


Figure 3: Architecture cible de ce projet