

Table des matières

I.	Cadrage du projet.....	3
A.	Contexte de l'entreprise.....	3
B.	Objectifs du projet.....	3
C.	Contraintes du projet	4
D.	Parties prenantes	5
II.	Rôle et périmètre du projet	6
A.	Organigramme (interne)	6
B.	Matrice RACI.....	7
C.	Périmètre du projet.....	8
D.	Gestion du projet.....	8
1.	Suivi du projet	8
E.	Démarche de la mise en place du projet.....	9
III.	Approche architecturale (par la méthode TOGAF)	10
A.	Les différentes phases de la méthode ADM	11
B.	Tableau descriptif des différentes phases du cycle ADM.....	12
C.	Contenue de l'architecture	12
1.	Architecture vision (Phase A)	12
a)	Les problèmes soulevés sur l'architecture actuelle	12
2.	Les contraintes et difficultés	13
3.	Objectifs et exigences qui doivent être atteints par l'état cible de l'architecture.....	13
D.	Architecture métier (Phase B)	14
1.	Analyse de l'existant.....	14
a)	Les différents composants de l'architecture système actuelle.....	14
b)	Les différentes dépendances des conteneurs	15
2.	Processus et fonctionnalités de la plateforme cible	15
3.	Analyses et diagnostics.....	16
4.	Architecture globale (cible)	17
a)	Le fonctionnement des micro-services	18
b)	Analyse et interconnexions des composants	20
(1)	Les interconnexions.....	20
c)	Les différentes solutions et technologies.....	21
5.	Sécurité des données	22
IV.	Plan de travail et plan de communication.....	22
A.	Activités	22

B.	Les livrables du projet	23
C.	Plan de communication.....	24
V.	Risques et facteurs de réduction.....	24
VI.	Critères d' acceptation et procédures.....	25
A.	Métriques et KPIs	25
VII.	Approbations signées	26

I. Cadrage du projet

A. Contexte de l'entreprise

Foosus est une start-up âgée de 3 ans, dans le secteur de l'alimentation durable. Leur objectif est de soutenir l'alimentation locale et de mettre les consommateurs en contact avec des producteurs et des artisans locaux.

Après plusieurs années de développement, **leur solution technique complexe n'évolue plus au rythme de l'activité** et risque d'entraver leur croissance.

Malheureusement, les choix historiques de Foosus ont engendré un volume important de dette technique et un manque de cohérence, qui ont commencé récemment à impacter de manière significative le développement de fonctionnalités.

Foosus veut construire une solution géo ciblée avec une nouvelle architecture. L'entreprise a besoin de frontières claires pour pouvoir développer une plateforme qui permette de l'innovation rapide et se mette à l'échelle du business.

En effet, une nouvelle plateforme d'e-commerce est nécessaire afin d'améliorer la compétitivité par rapport aux grandes entreprises d'e-commerce internationale.

B. Objectifs du projet

L'objectif du projet est de créer une plateforme de commerce électronique polyvalente pour faire passer l'entreprise à un niveau supérieur.

L'objectif business est de sortir de manière rapide et itérative un nouveau produit qui pourra coexister dans un premier temps avec la plateforme existante, avant de la remplacer.

En effet, l'entreprise a besoin d'une nouvelle architecture qui peut évoluer avec l'entreprise, prendre en charge un nouvel emplacement géographique et fournir la fiabilité nécessaire à nos clients, fournisseurs et consommateurs.

De plus, l'entreprise souhaite la mise en place d'une certaine standardisation pour la maintenance des développements futurs et à venir (ensemble hétérogène de technologies).

Les principaux objectifs exprimés par l'entreprise en matière d'architecture sont les suivants :

- tirer parti de la géolocalisation pour relier des fournisseurs et des consommateurs et pour proposer des produits disponibles à proximité des lieux de résidence de ces derniers. Un calculateur de distance devra être inclus pour permettre aux consommateurs de trouver les fournisseurs les plus proches d'eux.
- l'architecture devra être évolutive pour permettre à nos services de se déployer sur diverses régions à travers des villes et des pays donnés.
- les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.

- notre solution doit être disponible pour nos fournisseurs et nos consommateurs, où qu'ils se trouvent. Cette solution doit être utilisable avec des appareils mobiles et fixes. Elle doit tenir compte des contraintes de bande passante pour les réseaux cellulaires et les connexions Internet haut débit.
- elle doit pouvoir prendre en charge différents types d'utilisateurs (par exemple, fournisseurs, back-office, consommateurs) avec des fonctionnalités et des services spécifiques pour ces catégories.
- les livrables doivent pouvoir être fournis à intervalles réguliers pour que le nouveau système soit rapidement opérationnel et puisse être doté de nouvelles fonctionnalités au fil du temps
- la nouvelle plateforme devra également permettre à nos équipes produits d'innover rapidement en réorientant des solutions existantes, en expérimentant de nouvelles modifications et en facilitant l'intégration avec des partenaires internes et externes

Les premières études sur les meilleures pratiques en matière d'architecture en font apparaître plusieurs qui présentent des risques techniques réduits. Il s'agit notamment de micro-services potentiels, de normes prenant en charge des solutions Web et mobiles, de bases de données standard et d'autres approches similaires.

C. Contraintes du projet

Des contraintes devaient être prises en compte pour le bon déroulement du projet :

- le projet initial est approuvé pour un coût de 50 000 USD (45 190 €) et une période de 6 mois est prévue pour définir l'architecture et préparer un projet de-suivi afin de développer un prototype.
- l'architecture doit permettre d'obtenir le meilleur rapport qualité-coût.
- l'architecture peut inclure de nouveaux composants personnalisés ou des composants du commerce pour favoriser la flexibilité, la stabilité et l'extensibilité.
- les solutions open source sont préférables aux solutions payantes.
- toutes les solutions du commerce ou open source doivent, dans la mesure du possible, faire partie d'une même pile technologique afin de réduire les coûts de maintenance et de support continus.

D. Parties prenantes

Partie prenante	Fonctions	Préoccupation	Vision
Ash Callum	CEO	1-Taux d'inscriptions des utilisateurs	Rentabilité de la solution
Natasha Jarson	CIO	2-Innover dans le périmètre d'une Architecture d'Entreprise 3-Soutenir l'innovation technique rapide et l'expérimentation	Stabilité Sécurité Performance Disponibilité
Daniel Anthony	CPO	2-Innover dans le périmètre d'une Architecture d'Entreprise 3-Soutenir l'innovation technique rapide et l'expérimentation 4-Visibilité de la plateforme	Stabilité Sécurité Performance Disponibilité Evolutivité
Christina Orgega	CMO	2-Soutenir l'innovation technique rapide et l'expérimentation 4-Visibilité de la plateforme	Disponibilité Stabilité Performance
Jo Kumar	CFO	1-Taux d'inscriptions des utilisateurs 2-Innover dans le périmètre d'une Architecture d'Entreprise 3-Soutenir l'innovation technique rapide et l'expérimentation	Rentabilité de la solution Performance Qualité Coût
Jack Harkner	Directeur des opération	4-Visibilité de la plateforme	Sécurité Disponibilité Evolutivité

Légendes :

CEO (Chief Executive Officer): PDG

CIO (Chief Information Officer) : directeur des systèmes d'information

CPO (Chief Product Officer) : directeur produit

CMO (Chief Marketing Officer) : responsable de la direction marketing

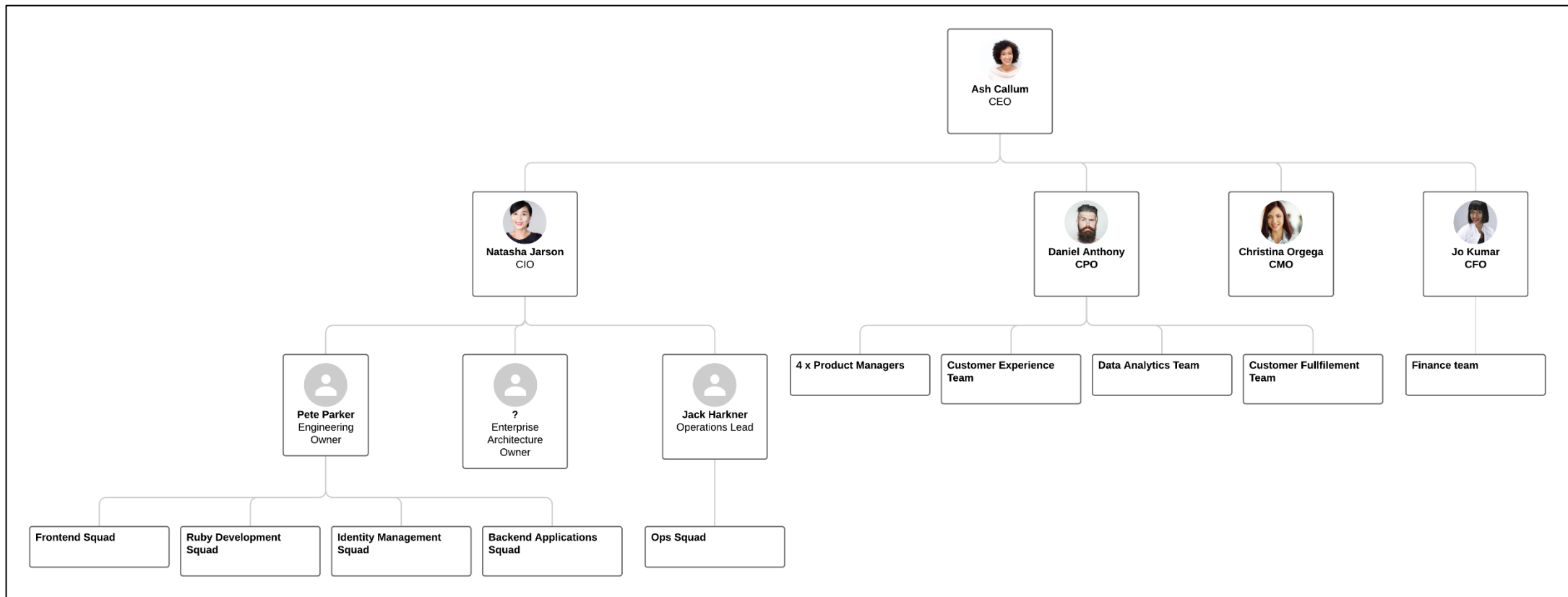
CFO (Chief Financial Officer) : directeur financier

DO : directeur des opérations

AL : Architect logiciel

II. Rôle et périmètre du projet

A. Organigramme (interne)



B. Matrice RACI

Le RACI est un outil de formalisation des rôles et responsabilités pour chaque partie prenante au projet. Cet outil est indispensable pour établir les attendus vis-à-vis de chaque partie prenante et ainsi lever toute ambiguïté dans les processus de décision.

- **R : Responsible** – c’est la personne qui est en charge d’effectuer l’action.
- **A : Accountable** – c’est la personne qui est en charge de superviser l’action et de référer à la hiérarchie.
- **C : Consulted** – c’est la personne qui peut aider, apporter son expertise ou son avis.
- **I : Informed** – c’est celui qui doit être tenu informé de l’avancée du projet.

Description de l'activité	CEO	CIO	CPO	CMO	CFO	DO	AL
Approbation de la mission	A	R	C	I	C	C	I
Gestion du planning	I	A/C	I	I	I	C	R
Gestion des changements	I	A	C	C	I	C	R
Rédaction de la documentation d'architecture	I	A/C	I	I	I	C	R
Définition des objectifs globaux	A	R	C	C	C	C	C
Description des processus du système	I	I	I	I	I	A	R
Identification et évaluation des risques		A				C	R
Définition de l'architecture du système	I	A	I	I	I	I	R
Validation technique	A	R	I	I	I	C	C
Validation des coûts	I	A	I	I	R	C	C
Communication aux collaborateurs	I	A	I	I	I	C	R

C. Périmètre du projet

La mission du projet consiste à déterminer d'une part la faisabilité d'une mise en place d'un nouveau système qui pourra coexister dans un premier temps avec la plateforme existante, avant de la remplacer, et, d'autre part analyser le système existant et faire des recommandations.

- Analyse de l'existant
- Proposition d'une nouvelle architecture
 - Conception
 - Réalisation
 - Migration
 - Mise en production de la nouvelle architecture

D. Gestion du projet

1. Suivi du projet

Dans notre équipe, nous travaillons sur le modèle agile. Cependant, nous faisons des réunions quotidiennes. Nous serons donc amenés à solliciter l'équipe afin de vous faire des démonstrations des applications en cours de développement, afin d'avoir un retour des utilisateurs.

- *Point journalier (Daily)*

Dans notre équipe, nous travaillons sur le modèle du cycle agile. Cependant, nous faisons des réunions quotidiennes entre développeurs afin de faire un point sur l'avancée des tâches de chacun. Cette réunion permettait d'aborder les difficultés rencontrées, les points de blocages et l'avancement concernant chaque tâche. Il permettait de suivre et structurer le projet.

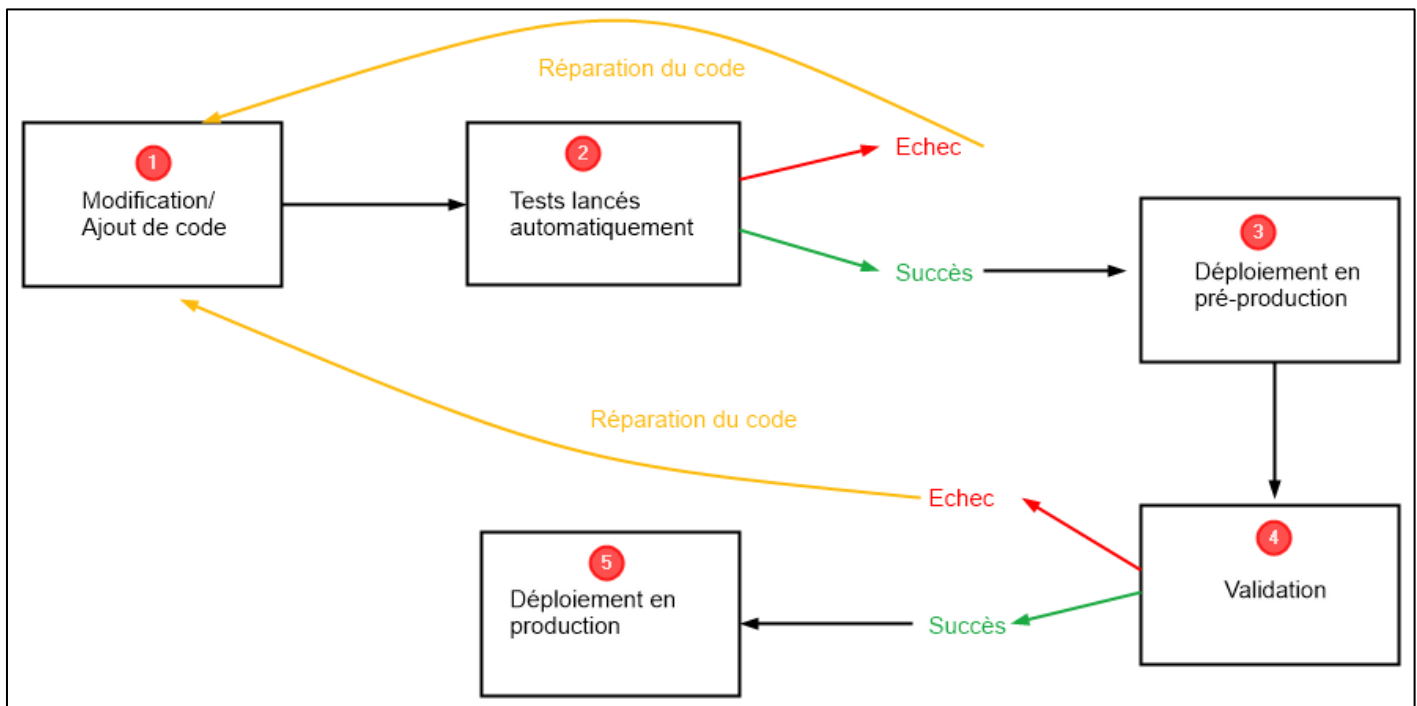
- Point hebdomadaire

Une seconde réunion est réalisée une fois par semaine. Cette réunion rassemble toute l'équipe du projet. Cette réunion nous permet de faire un point sur l'avancée du projet, ainsi de pouvoir respecter les délais du projet.

D'autres réunions sont réalisées à la fin des sprints par exemple pour faire des démonstrations des applications en cours de développement, afin d'avoir un retour concret. Cela permet d'ajuster à temps les fonctionnalités du projet.

- Intégration continue

Nous mettons en place **l'intégration continue**, qui consiste à tester de manière automatisée chaque révision de code avant de le déployer en production.



En effet, dans un premier temps, le développeur code une fonctionnalité, ou modifie une fonctionnalité, il ajoute par la suite à son dépôt de code. Le serveur d'intégration va ensuite faire tourner tous les tests pour vérifier qu'aucune régression n'a été introduite dans le code source suite à cet ajout.

- Si un problème est identifié, le déploiement n'a pas lieu et les développeurs sont notifiés.
- Sinon, le serveur déploie le code en préproduction puis en production.

Cela va garantir un code de qualité, et donc une meilleure satisfaction des utilisateurs ce qui n'est pas le cas actuellement.

E. Démarche de la mise en place du projet

Ce document, la **déclaration de travail d'architecture** définit le périmètre et l'approche qui seront utilisés pour mener à bien un projet d'architecture. En effet, il constitue habituellement le document qui permet de mesurer la réussite de l'exécution du projet d'architecture et peut former la base de l'accord contractuel entre le fournisseur et le consommateur de services d'architecture.

De ce fait, durant tout le long de ce projet, nous allons nous baser sur l'architecture TOGAF. La **méthode TOGAF (The Open Group Architecture Framework)** est devenue ces dernières années un **cadre de référence pour l'architecture d'entreprise**.

En effet, TOGAF ne prescrit pas les modèles à utiliser pour représenter l'architecture, mais guide le processus lors de la création de l'architecture. Il permet de :

- Apporter une vision d'ensemble
- Permettre ainsi d'avoir un fil directeur et les pratiques sont harmonisées
- Permettre de rendre beaucoup plus simple la collaboration entre services métiers et techniques

Nous estimons que cette démarche fournit une base raisonnable. **C'est** un recueil de bonnes pratiques qui apporte un **cadre complet pour l'architecture de l'entreprise**.

III. Approche architecturale (par la méthode TOGAF)

La **méthode TOGAF (The Open Group Architecture Framework)** est devenue ces dernières années un **cadre de référence pour l'architecture d'entreprise**.

Cette méthode couvre les 4 domaines de l'architecture :

- le métier (business)
- les données
- les applications
- l'infrastructure technique

En effet, TOGAF intègre les stratégies, les exigences, les processus métiers, les applications, les infrastructures techniques et des associations efficaces entre ses différents aspects et va même jusqu'à la planification et la gestion du changement.

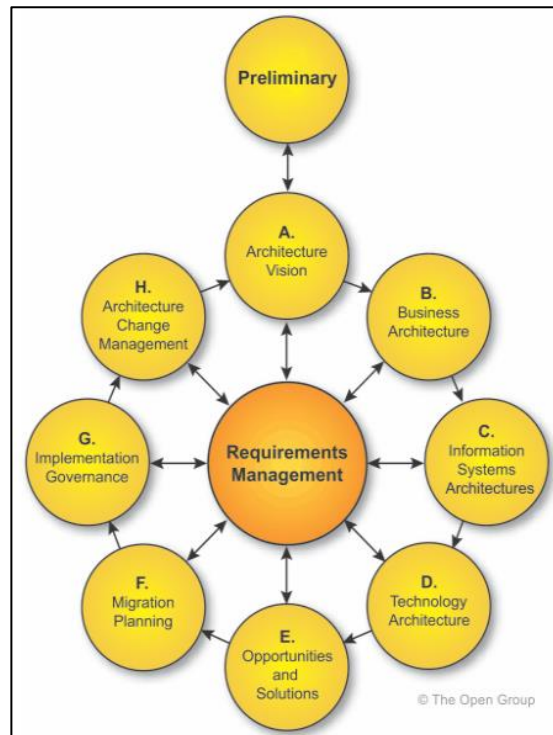
Cette collaboration s'appuie sur un processus organisé : c'est le rôle de la démarche ADM , qui fournit un cadre de progression du projet de transformation de l'architecture.

La démarche consiste à connaître l'existant, fixer la cible, établir la meilleure trajectoire pour l'atteindre et mettre en place les moyens pour réaliser avec succès la transformation.

Le cycle **ADM (Architecture Development Method)** constitue le cœur de cette démarche.

A. Les différentes phases de la méthode ADM

La **méthode de développement architectural** (ADM) est une approche détaillée, divisée en phase, qui explique comment gérer l'ensemble du cycle de vie d'une architecture. L'ADM se présente sous forme d'un découpage en phase, dont une phase préliminaire, nécessaire à l'organisation du projet, suivie de 8 phases (numérotées de A à H). Il permet de construire l'architecture des domaines métiers, applications, données et techniques, de planifier son déploiement, de la mettre en œuvre et finalement, de gérer les changements à opérer :



Source : the open Group

- La phase préliminaire
- Phase A : vision de l'architecture
- Phase B : architecture Métier
- Phase C : architecture des systèmes d'information
- Phase D : architecture technique
- Phase E : solutions et opportunités
- Phase F : planning de migration
- Phase G : gestion de l'implémentation
- Phase H : gestion de la maintenance et des évolutions
- Phase gestion des exigences

B. Tableau descriptif des différentes phases du cycle ADM

Phases	Descriptions
La phase préliminaire	L'organisation du projet. Dans cette phase, l'équipe d'architecture d'entreprise, les principes architecturaux et le cadre à utiliser sont définis.
Phase A : vision de l'architecture	Définition du périmètre du projet d'évolution de l'architecture, aboutissant à une validation des acteurs concernés.
Phase B : architecture Métier	Définition de l'architecture cible des processus métier, analyse des écarts avec l'architecture actuelle.
Phase C : architecture des systèmes d'information	Le but de cette phase est de décrire l'architecture des données et l'architecture des applications.
Phase D : architecture technique	Le but de cette phase est de décrire l'architecture technologique qui constituera la base des travaux de mise en œuvre ultérieurs.
Phase E : solutions et opportunités	Début de la planification. Identification des principaux projets de mise en œuvre, qui se regroupent en architecture de transition.
Phase F : planning de migration	Développement d'un plan détaillé de mise en œuvre et de migration
Phase G : gestion de l'implémentation	Pilotage du projet de mise en œuvre et contrôle de sa conformité aux plans d'architecture.
Phase H : gestion de la maintenance et des évolutions	Gestion permanente des modifications apportées à l'architecture, afin qu'elle reste en phase avec la stratégie de l'entreprise et les besoins métier.

C. Contenu de l'architecture

1. Architecture vision (Phase A)

a) *Les problèmes soulevés sur l'architecture actuelle*

- La **plateforme a atteint un stade critique** où elle n'est plus adaptée à son objectif initial. La solution technique complexe n'évolue plus au rythme de l'activité et risque d'entraver notre croissance
- Les équipes de développement sont pleinement investies dans **l'extinction d'incendies et dans son maintien en état de marche** (ce qui a ralenti notre capacité à livrer de nouvelles fonctionnalités et à rester compétitifs).

- En effet, un **volume important de dette technique** et un **manque de cohérence**, qui ont commencé récemment à impacter de manière significative le développement de fonctionnalités
- **Instabilité de la plateforme** et par une image de marque négative causée par des interruptions de service visibles par le public
- Un **fort déclin des inscriptions utilisateurs** (cette métrique a chuté rapidement au cours des derniers mois et doit être améliorée en priorité)
- Les **contraintes de performances du système existant** empêchaient de supporter le niveau d'engagement et de croissance attendus de nos futurs programmes marketing.
- **La fonctionnalité la plus critiquée de la plateforme existante est notre fonctionnalité de recherche de fournisseurs alimentaires.** Les enseignements actuels montrent que 48 % des parcours commencent par une recherche, mais sont abandonnés avant qu'un client ait même consulté une offre alimentaire de l'un de nos fournisseurs.

2. Les contraintes et difficultés

- Le taux d'inscriptions de nouveaux utilisateurs a chuté rapidement au cours des derniers mois. En effet, les inscriptions constituent une métrique clé aux yeux des investisseurs. On doit permettre de maintenir un taux positif d'inscriptions de nouveaux utilisateurs.
- La plateforme historique de Foosus a naturellement évolué vers la complexité. On doit proposer un périmètre clair.
- Avec le marché actuel et avec la concurrence, on doit concevoir une plateforme permettant de garder à l'idée l'extensibilité et la personnalisation des fonctionnalités.
- La marque Foosus doit être renforcée en réduisant les interruptions de service visibles par les utilisateurs. Cela implique :
 - Des processus pour réduire le risque de sortir des solutions qui échouent ou qui soient de mauvaise qualité
 - La capacité de sortir de nouvelles versions de notre plateforme sans impacter l'utilisateur par des interruptions de service.

3. Objectifs et exigences qui doivent être atteints par l'état cible de l'architecture

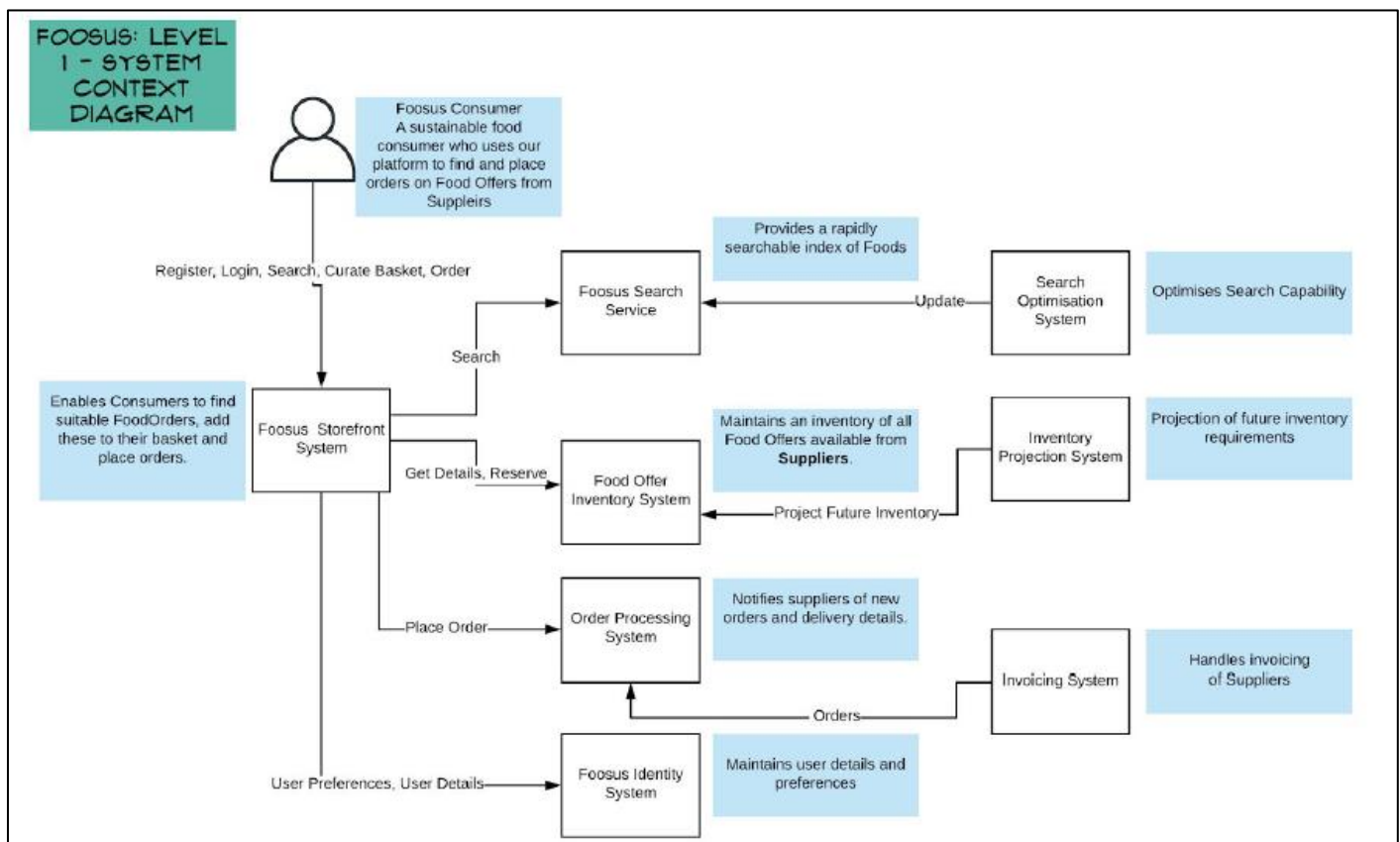
- Des campagnes de marketing Foosus doivent être menées dans de nombreuses grandes villes, avec l'assurance que la plateforme demeurera utilisable, réactive, et délivrera une expérience client de première classe. Pour parvenir à ce but, l'entreprise a besoin de concevoir une solution d'architecture et un processus de gouvernance qui nous aident à atteindre l'ensemble des objectifs business actuels, ainsi que la vision globale.

- Même si le système est surchargé, les utilisateurs connectés doivent pouvoir continuer à accéder à tous les services de façon dégradée. Cela n'est pas le cas aujourd'hui !
- Nous prévoyons de mener des campagnes Foosus dans plusieurs régions géographiques et nous voulons que la plateforme puisse non seulement absorber le trafic, mais soit également capable d'évoluer pour gérer les augmentations de charges.
- Nous ne pouvons plus accepter de désactiver la plateforme à chaque installation d'une nouvelle version ou à chaque modification du schéma de la base de données.

D. Architecture métier (Phase B)

1. Analyse de l'existant

a) Les différents composants de l'architecture système actuelle



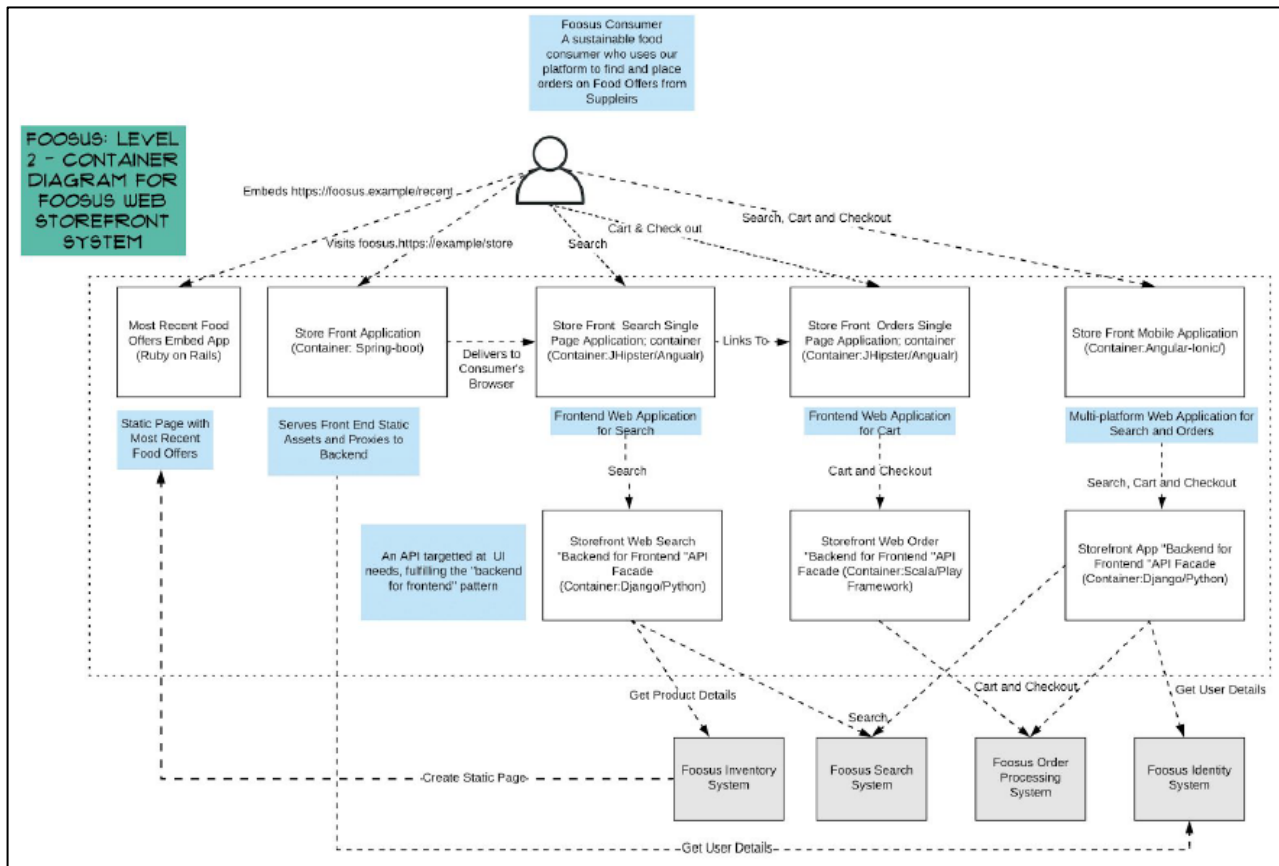
Source document interne de l'entreprise (diagramme de composants)

L'utilisateur se connecte à la plateforme. La plateforme offre la possibilité aux consommateurs de trouver des commandes de produits alimentaires, de les ajouter à leur panier et de passer des commandes.

En effet, on a 4 composants :

- la partie recherche de produit
- la partie inventaire des produits (inventaire de toutes les offres alimentaires disponibles auprès des fournisseurs)
- la partie pour passer la commande (informe les fournisseurs des nouvelles commandes et des détails de livraison)
- la partie espace utilisateur (détails de l'utilisateur, ces préférences)

b) Les différentes dépendances des conteneurs



Source document interne de l'entreprise (diagramme de niveau conteneur)

2. Processus et fonctionnalités de la plateforme cible

Dans la plateforme, nous avons plusieurs composants :

- système d'inventaire
- système de commande
- système de recherche
- système de facturation

Nous avons également les différents utilisateurs qui interagissent avec ces composants :

- les fournisseurs alimentaires qui soumettent à Foosus un inventaire des produits alimentaires disponibles. Les fournisseurs recevront également les commandes passées.

- les clients des produits de consommation rechercheront, trouveront et commanderont des produits alimentaires
- l'équipe finance de Foosus recevra les paiements pour la partie facturation.

Tableau des différents utilisateurs et leurs rôles :

Utilisateurs	Rôles
Client de produit de consommation	Acheteur
Fournisseur alimentaire	Vendeur
Représentant de l'équipe satisfaction client	Face au client
Développeur	Exécutant
Membre de l'équipe finance	Facturation

3. Analyses et diagnostics

Exigences	Solutions
Gestion de montée en charge	<ul style="list-style-type: none"> - Un répartiteur de charge (Load balancer) - Le système doit être capable de s'adapter à une forte augmentation de la demande sans que son niveau de performances n'en soit impacté (Scalabilité)
Gestion de la solution sur différentes devises : <ul style="list-style-type: none"> - Windows - Mobile 	<ul style="list-style-type: none"> - Une architecture dont les services peuvent être consommés par différents clients quelle que soit la plateforme - Une indépendance des modules (autonomie) - Une architecture offrant une hétérogénéité technologique
Gestion des évolutions futures de la solution : <ul style="list-style-type: none"> - Permettre des améliorations et modifications futures du système 	<ul style="list-style-type: none"> - Une architecture avec un faible couplage - Une indépendance des modules (autonomie) - Une possibilité de réutilisations des modules développés - Une architecture facile à maintenir
Gestion de plusieurs utilisateurs à travers le monde	<ul style="list-style-type: none"> - Un système permettant une connexion et un échange de données plus facilité à distance et par internet - Il faut des instances du système <ul style="list-style-type: none"> • Faible latence • Haute disponibilité
Les solutions open source sont préférées aux solutions payantes	<ul style="list-style-type: none"> - Choisir des technologies et architecture offrant des solutions opens source

4. Architecture globale (cible)

A partir de l'ensemble des éléments que nous avons traité tout au long de cette étude, nous proposons l'architecture **micro-service**.

En effet, selon les exigences techniques énoncées par Foosus à savoir que la solution doit permettre des améliorations et modifications futures du système. Cette architecture a un couplage faible qui permettra par la suite de pouvoir faire des évolutions sans impacter toute la solution.

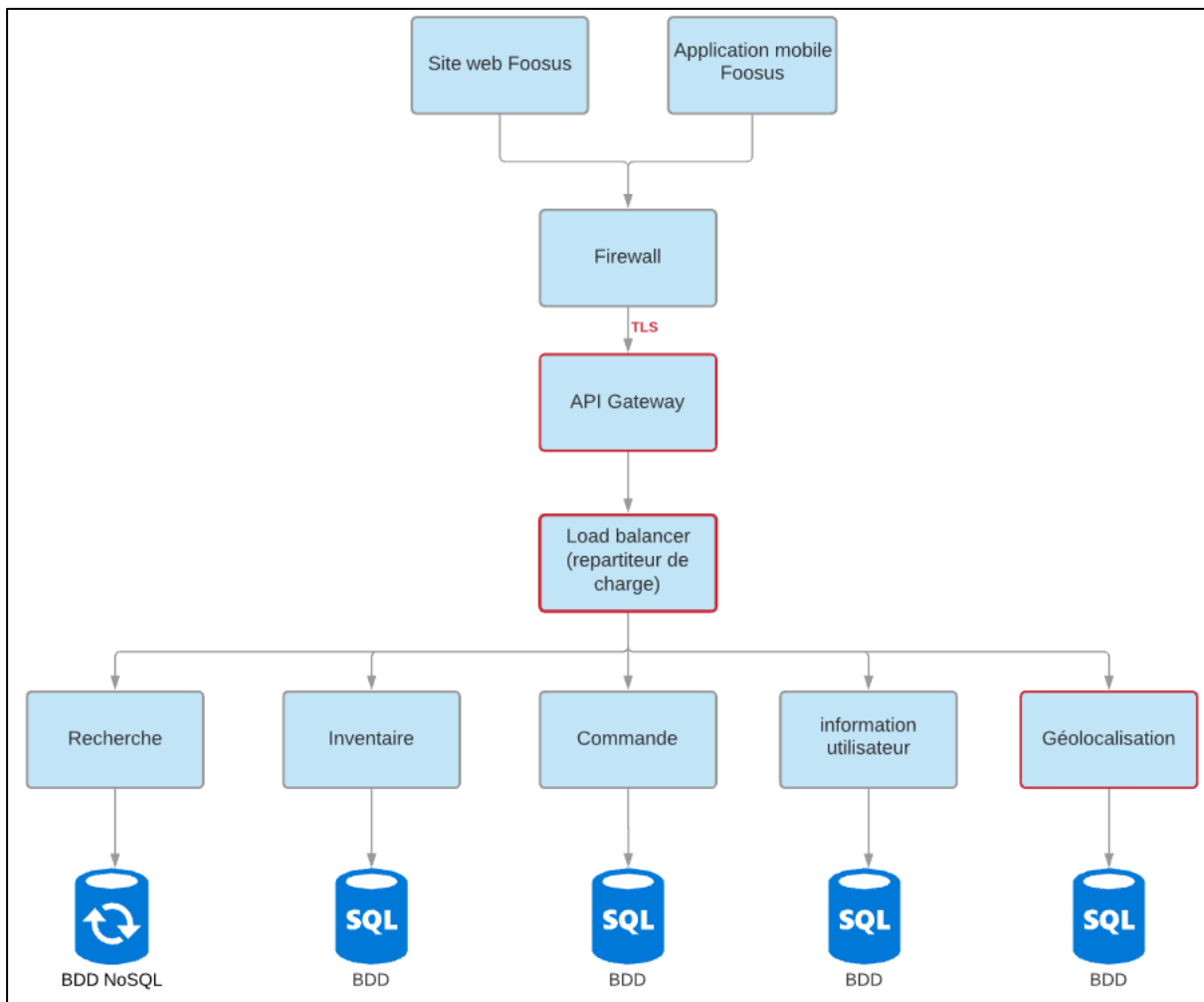
De plus, la solution doit également gérer les ordinateurs fixes et portables. On a vu que ces architectures sont compatibles avec de nombreux autres environnements de développement. L'architecture est donc indépendante des plates-formes.

On peut également répartir la charge dans les micro-services, on peut donc augmenter le nombre d'instance s'il y a trop de charges.

Par conséquent, nous avons donc rajouté dans l'ancien système des micro-services . Ces derniers sont parfaitement autonomes. En effet, chaque micro-service à sa propre base de données et pour faciliter la communication les uns avec les autres, chacun à son propre API REST. Cependant, pour la fonctionnalité de recherche, suite aux différents problèmes engendrés, nous avons mis en place une base de données non relationnelle (NoSQL). En effet, cette base de données pourra lire et traiter de gros volumes de données à une vitesse-record.

Pour répartir la charge, nous avons mis en place un répartiteur de charge (Load balancer).

On peut avoir plusieurs instances de micro-service.



⇒ Les composants en rouge, ce sont ceux qui ont été ajouté à la nouvelle architecture. Les autres composants existent déjà mais des modifications vont être apportées afin de les rendre plus performants.

a) *Le fonctionnement des micro-services*

Nous proposons de découper les différents micro-services. Les micro-services sont les suivants :

- Recherche :

C'est le micro-service qui va permettre de faire des recherches des offres alimentaires, des fournisseurs ou des régions.

- Inventaire

C'est le micro-service qui va permettre de rechercher toutes les offres alimentaires disponibles auprès des fournisseurs.

- Commande

C'est le micro-service qui va permettre d'informer les fournisseurs des nouvelles commandes et des détails de livraison mais aussi permettre à l'utilisateur de passer commande.

- Information utilisateur

C'est le micro-service qui va contenir toutes les informations des utilisateurs (détails de l'utilisateur, ces préférences)

- Géolocalisation

C'est le micro-service qui va permettre de relier les fournisseurs et les consommateurs afin de proposer des produits disponibles à proximité des lieux de résidence de ces derniers.

- API Gateway

Une API Gateway (également appelée passerelle API) est le point d'entrée unique pour les API et micro-services. En effet, elle permet d'agréger différents micro-services. L'agrégation des requêtes sera faite directement au travers de l'API Gateway, cela permet de réduire la charge réseau et la multiplication des appels. L'API Gateway permet de faciliter la mise en place d'une politique unique de sécurité, permet également de simplifier les évolutions d'architecture.

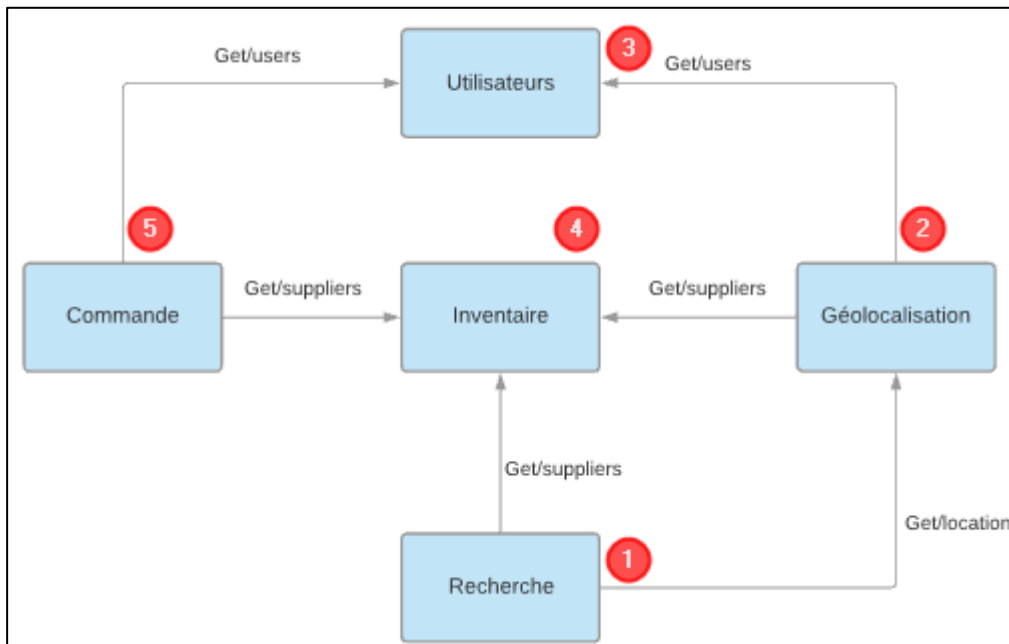
- Load balancer

Un load balancer permet de répartir la charge de travail entre un ou plusieurs serveurs. Il a pour avantage :

- d'augmenter la qualité des services
- Améliorer le temps de réponse des services tout en évitant de surcharger de manière inégale
- Palier la défaillance
- Gérer des hauts niveaux de trafic

b) Analyse et interconnexions des composants

(1) Les interconnexions



Chaque utilisateur pourra se connecter soit par le site web ou l'application mobile. Les informations demandées vont passer par le Firewall puis par l'API Gateway et enfin par le répartiteur de charge.

L'utilisateur se connecte à la plateforme. La plateforme offre la possibilité aux consommateurs de trouver des commandes de produits alimentaires, de les ajouter à leur panier et de passer des commandes.

- Recherche

Pour gérer la recherche, on aura besoin des fournisseurs et leurs offres mais aussi de la géolocalisation.

- Géolocalisations

Pour gérer la géolocalisation, on aura besoin des utilisateurs. En effet, à l'exemple du schéma ci-dessous, on a besoin des informations des utilisateurs et des fournisseurs et de ces offres pour la localisation.

- Utilisateurs

Communique avec les services commande et géolocalisation. C'est dans ce service que sont gérées les informations des utilisateurs.

- Inventaire

Communique avec les différents services sauf le service utilisateurs.

- Commande

Pour gérer les commandes, on aura besoin des fournisseurs et leurs offres mais aussi des utilisateurs.

c) Les différentes solutions et technologies

Composant	Technologie	Avantage	Particularité
Back-end	Sprint boot	<ul style="list-style-type: none"> - Simplicité de prise en main (permet donc de créer une API de services très simplement et est productif dans la construction de son API) - Légèreté (très léger et utilise le strict minimum pour faire tourner le service) - Intégration facilitée (s'intègre particulièrement bien dans une architecture orientée micro-services) 	<p>Open source</p> <p>Utilisé dans l'ancienne architecture</p>
Front-end web	Angular	<ul style="list-style-type: none"> - Performance - C'est un Framework complet proposant tous les outils nécessaires au développement d'applications web, mobile et desktop - Evolution (Il permet de faire évoluer au fur et à mesure du développement des besoins d'une entreprise) - Simplicité (facilitent la prise en main et la compréhension du Framework) 	<p>Open source</p> <p>Utilisé dans l'ancienne architecture</p>
Mobile	Ionic	<ul style="list-style-type: none"> - Ionic propose une architecture beaucoup plus simple à utiliser et une documentation claire et concise. - la possibilité de développer votre application une fois, et la déployer sur plusieurs terminaux mobiles. - Simplicité de prise en main 	<p>Open source</p> <p>Utilisé dans l'ancienne architecture</p>
BDD	MySQL	<ul style="list-style-type: none"> - Rigidité (la structure des tables et le type d'attributs sont définis et fixés en avance) - Puissance (on peut récupérer des données dans plusieurs tables en une seule requête) - Réduit au maximum la redondance des données 	Open source
	NoSQL (Elasticsearch)	<ul style="list-style-type: none"> - Performance (la vitesse à laquelle la base de données est en mesure de réagir aux différentes requêtes) - Scalabilité (capacité à s'étendre sur plusieurs serveurs au lieu d'être limitée à un seul) - Flexibilité 	Open source
API Gateway et Load balancer	HAProxy	<ul style="list-style-type: none"> - L'un des plus populaires sur le marché - Utilise moins de ressources (CPU et RAM) pour de meilleures performances 	Open source
Géolocalisation	OpenStreetMap	<ul style="list-style-type: none"> - logiciel libre - Facile à configurer - Fait partie des plus populaires sur le marché 	Open source
Cloud	Azure	<ul style="list-style-type: none"> - Mets fortement l'accent sur la sécurité - Offre une haute disponibilité et une redondance dans leurs centres de données à une échelle très globale 	

5. Sécurité des données

Les comptes seront protégés par des noms d'utilisateur et des mots de passe. En effet, tous les utilisateurs accédant à la plateforme doivent avoir un compte utilisateur dans le système. En plus de l'identification, d'autres solutions ont été mise en place pour garantir la sécurité des données avec la mise en place de :

- **Firewall** : Il est configuré pour protéger les systèmes Foosus de la circulation réseau inattendue ou non planifiée.
- **TLS** : il permet d'échanger des informations de manière plus sûr. Il garantit que les données seront chiffrées et protégées contre toute interception. Il assure :
 - La confidentialité : protège les transmissions de données
 - L'intégrité : garantit que les données qui sont demandées ou soumises sont bien celles qui sont fournies
 - L'authentification : garantit que le serveur auquel vous êtes connecté est le bon serveur

En effet, le TLS c'est pour sécuriser les données en transit.

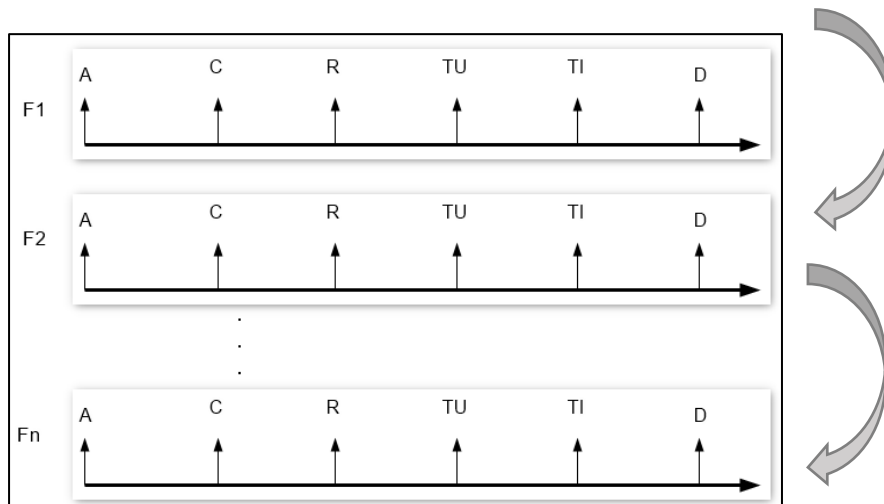
IV. Plan de travail et plan de communication

A. Activités

Pour ce projet, nous avons découpé nos tâches en différentes phases. Nous avons 6 phases, qui correspondent aux 6 mois de délais prévus pour ce projet :

- Phase d'analyse
- Phase de conception
- Phase de réalisation
- Phase de test unitaire
- Phase de test d'intégration
- Phase de déploiement

En effet, comme précisé précédemment nous avons mis en place l'intégration continue. Nous travaillons en mode itérative. En effet, chaque petit changement nous ramène à l'architecture cible.



Légendes :

F1 : fonctionnalité 1

A : analyse

C : conception

R : réalisation

TU : test unitaire

TI : test d'intégration

D : déploiement

B. Les livrables du projet

- Une déclaration de travail d'architecture
- Une spécification des conditions requises pour l'architecture
- Un contrat d'architecture avec les utilisateurs business
- Des contrats d'architecture avec les fonctions business et développement et design.
- Un dépôt architectural navigable contenant vos artefacts

C. Plan de communication

Evènement	Canaux	Format
Mise en place des différentes étapes du projet	Tableau de bord	Excel Réunion
Suivi du planning	Mail Agenda Application de gestion de projet (Redmine)	Excel Réunion
Suivi des avancements des tâches	Application de gestion de projet (Redmine) Mail	Excel Réunion
Suivi des livrables	Git	Excel Réunion
Formation des utilisateurs	Mail	PowerPoint

V. Risques et facteurs de réduction

Critère d'évaluation du risque :

Niveau de probabilité :

- 1 = Risque faible
- 2 = Risque moyen
- 3 = risque élevé
- 4 = risque très élevé

Niveau de Criticité :

- Très critique
- Critique
- Moins critique

Risques	Parties prenantes	Niveau de criticité	Niveau de probabilité	Commentaires / solutions
Indisponibilité du système	CEO	Très critique	Faible	Les données ne sont pas connectées entre elles. Ce qui, au cours d'une panne, on aura une indisponibilité des données faibles. ⇒ Utilisation des micro-services
Forte demande des ressources de l'application	CEO	Très critique	Faible	Le système doit être capable de s'adapter à une forte augmentation de la demande sans que son niveau de performances n'en soit impacté ⇒ Utilisation d'un Load balancer

Coût de maintenance	CIO	Critique	Moyen	Chaque système est indépendant et est développé avec la même technologie
Risque humain : absence d'un développeur, démission, maladie	CIO	Critique	Moyen	<p>Une équipe a été prévue en cas de problème. Nous travaillons avec des prestataires qui sont réactifs en cas de besoin.</p> <p>Nous disposons également des référents techniques qui accompagnent nos développeurs et qui sont là pour apporter de l'aide en cas de besoin pour chaque projet.</p>
Dépassement de budget	CFO	Critique	Fort	On n'a pas beaucoup d'informations pour le planning et pour les équipes à mettre en place.
Dépassement de l'objectif délai	CFO	Critique	Fort	On n'a pas beaucoup d'informations pour le planning et pour les équipes à mettre en place.

VI. Critères d'acceptation et procédures

A. Métriques et KPIs

Composants/Descriptions	Indicateurs	
Inventaire	Adhésion de producteurs alimentaires	Passer de 1,4 /mois à 4/ mois
Recherche	Taux de latence	Taux de délais d'attente pour une recherche doit chuter considérablement
Utilisateur	Le nombre d'adhésions des utilisateurs	Le nombre d'adhésions des utilisateurs doit augmenter (augmentation de 10%)
Géolocalisation	Distance entre fournisseurs et clients	Assurer que tous les fournisseurs sont liés aux clients proches d'eux
Production	Taux d'incidents de production	Réduire et passer de > de 25/mois à moins de 1/mois
Parution	Délai de parution	Réduire de 3.5 semaines à moins d'une semaine

VII. Approbations signées

Nom	Poste	Signature/ initial	Date
Ash Callum	CEO	A.C	20/04/2021