

# Déclaration de Travail d'Architecture

6 juin 2021

Aftis Saadi



# Table des matières

<b>1</b>	<u>Objet de ce document</u>
<b>2</b>	<u>Déclaration de travail d'architecture</u>
2.1	Requête du projet et contexte . . . . .
2.2	Descriptpion du projet et périmètre . . . . .
2.3	Vue d'ensemble . . . . .
2.4	Alignement stratégique . . . . .
<b>3</b>	<u>Objectifs et périmètre</u>
3.1	Objectifs . . . . .
3.2	Périmètre . . . . .
3.3	Approche managériale . . . . .
<b>4</b>	<u>Rôles et responsabilités)</u>
4.1	Structure de gouvernance . . . . .
4.2	Process du projet . . . . .
4.3	Rôles et responsabilité . . . . .
<b>5</b>	<u>Approche architecturale</u>
<b>6</b>	<u>Plan de travail</u>
6.1	Element de travail 1 . . . . .
6.1.1	Activité . . . . .
6.1.2	Livrable . . . . .
6.2	Element de travail 2 . . . . .
6.2.1	Activité . . . . .
6.2.2	Livrable . . . . .
6.3	Element de travail 3 . . . . .
6.3.1	Activité . . . . .
6.3.2	Livrable . . . . .
6.4	Element de travail 4 . . . . .
6.4.1	Activité . . . . .
6.4.2	Livrable . . . . .
<b>7</b>	<u>Plan de communication</u>
<b>8</b>	<u>Risques et facteurs de réduction</u>
<b>9</b>	<u>Critères d'acceptation et procédures</u>
<b>10</b>	<u>Approbations signées</u>

# **1    Objet de ce document**

Ce document est une Déclaration de travail d'architecture pour le projet Foosus.

La Déclaration de travail d'architecture définit le périmètre et l'approche qui seront utilisés pour mener à bien un projet d'architecture. La Déclaration de travail d'architecture 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. En général, toutes les informations de ce document doivent se situer à un haut niveau.

## **2    Déclaration de travail d'architecture**

### **2.1   Requête du projet et contexte**

Après plusieurs années de travail et une expansion notable avec le système actuel, Foosus a remarqué que son architecture actuelle n'était plus apte à suivre les avancées technologiques requises par le domaine, de plus le système actuel utilise tellement technologies que Foosus passe le plus clair de son temps à réparer plutôt que d'innover.

C'est à cause de ces raisons que Foosus a décidé de construire un nouveau système performant et standard.

### **2.2   Description du projet et périmètre**

Foosus souhaite un nouveau système pour son site e-commerce et son application mobile, voici les principales demandes pour le nouveau système :

- Un système utilisant la géolocalisation.
- Un système standard apte à l'évolution.
- Avoir une première version d'ici 6 mois pour un budget de 50.000 euros.
- Auditer le framework architectural actuel et le modifier si nécessaire.
- Prendre en charge tous les types d'utilisateur.

### **2.3   Vue d'ensemble**

Le nouveau système adoptera une architecture Micro-Service, architecture qui convient parfaitement à notre projet, on va standardiser les technologies en choisissant des langages performant et tendance "Spring boot, Angular et Ionic". Le système possédera aussi la fonctionnalité de géolocalisation qui nous permettra de cibler les clients où qu'ils soient et leur proposer le meilleur service.

### **2.4   Alignement stratégique**

Pour commencer comme discuté avec Pete Parker on va opter pour une architecture MicroService elle permet l'évolution, la réutilisation et facilite la maintenance.

Concernant les technologies de développement "Spring-boot" et "Angular" c'est des Frameworks très utilisés et super efficace pour respectivement le Back-end et le Front-end, la limitation des technologies répond à la condition de la standardisation. Pour la fonctionnalité de la géolocalisation on utilisera l'api Google Maps pour avoir une carte et DistanceMatrix Api pour nous donner les magasins les plus

proches. Finalement pour que le système supporte plusieurs type d'utilisateurs la sécurité de Spring-Boot permet l'implémentation d'un système d'accès basé sur les rôles à savoir fournisseurs, back-office, consommateurs...

### 3 Objectifs et périmètre

#### 3.1 Objectifs

<i>Objectif Business</i>	<i>Notes</i>
Augmenter le nombre d'inscrit	Nouvelle interface, plus fluide avec CDN
Avoir une meilleure image	Mieux sécuriser le système, innover souvent et enlevé les temps pour déployer solution
Viser des clients partout dans le monde	utilisation de la géolocalisation

FIGURE 1 – Objectifs

#### 3.2 Périmètre

#### 3.3 Approche managériale

Foosus a remarqué que le déploiement de patch ou de mise a jour volumineuse était contre productif, d'un coté ça mettait le site hors service longtemps et de l'autre on avait une probabilité de bugs plus grande. La meilleure manière de travailler c'est d'utiliser le Framework Scrum, l'équipe aura des daily scrum pour parler de ce qu'ils ont fait la veille et leurs objectifs du jour. Concernant les livrable régulier, la méthode scrum utilise les "sprint" qui sont des période de 15 jours ou on doit a la fin remettre un livrable ce qui en fait des livrable régulier et rarement volumineux.

## 4 Rôles et responsabilités)

### 4.1 Structure de gouvernance

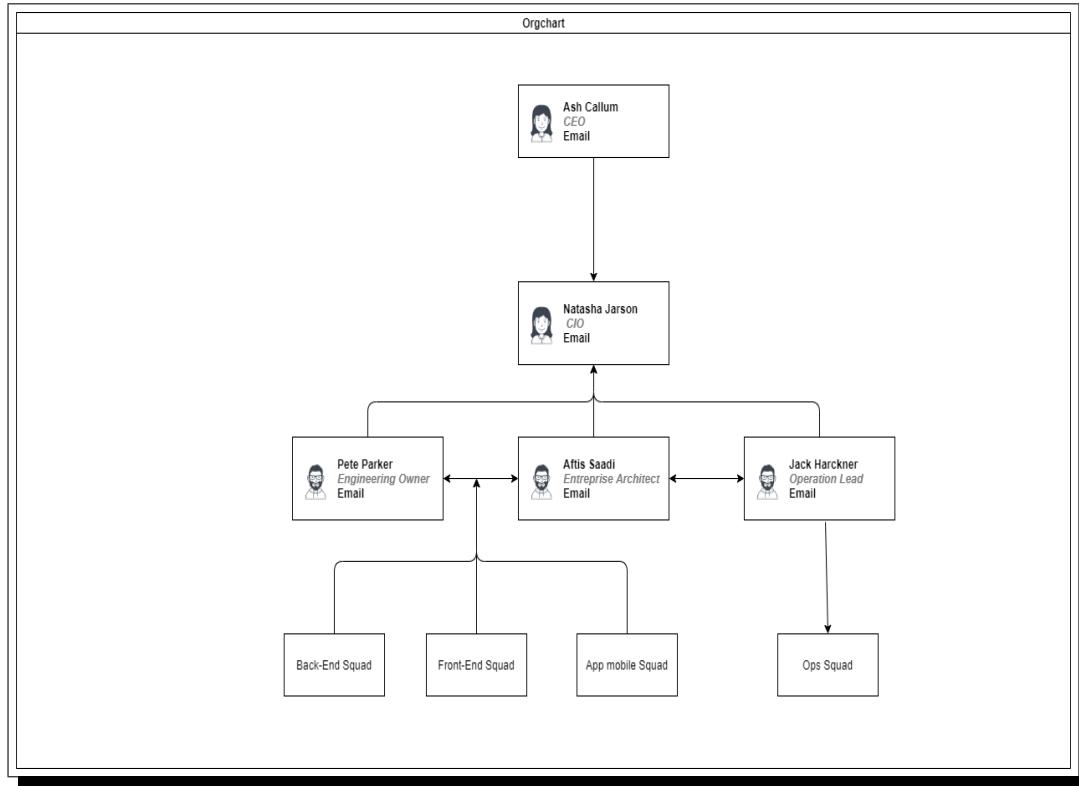


FIGURE 2 – Objectifs

## 4.2 Process du projet

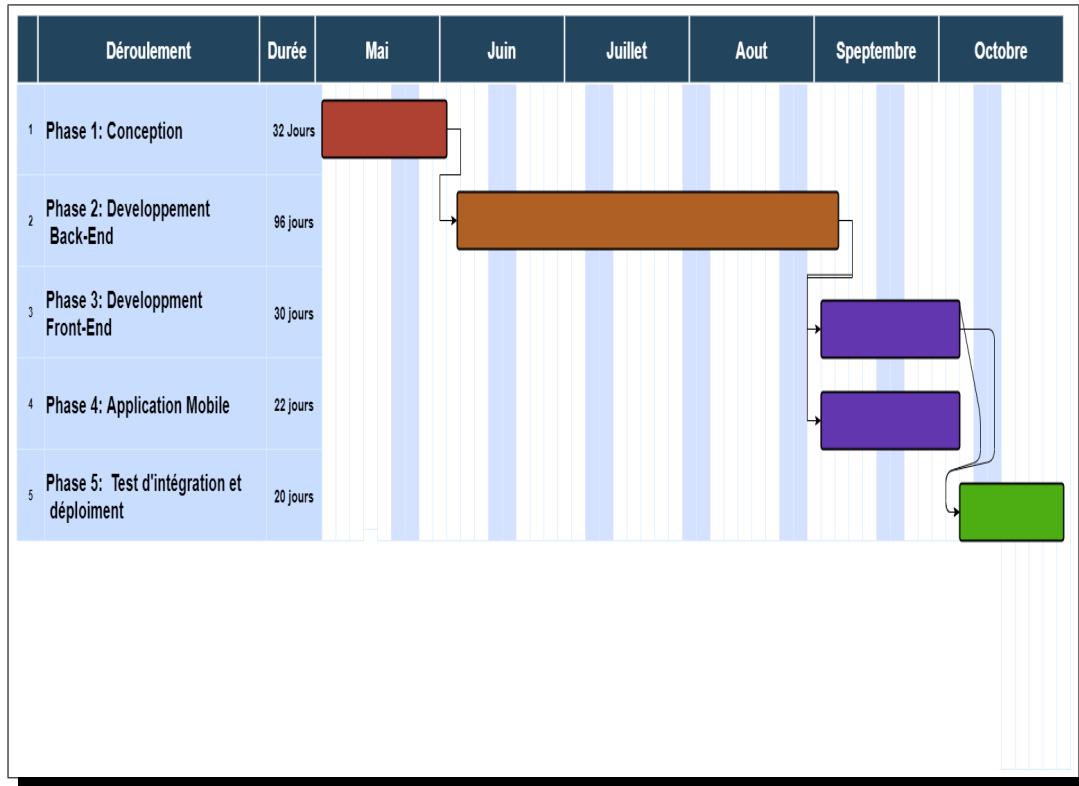


FIGURE 3 – Objectifs

## 4.3 Rôles et responsabilité

Project Tasks	CEO	CIO	Entreprise Architecture Owner	Engineering owner	Operation lead	Back-end team	Front-end Team	Ops Team	CPO	CFO
<b>Phase de conception</b>										
Calcul budget du projet	I	A	R							C
Back-end Conception		A	R	I		I				
Front-end conception		A	R	I			I		I	
<b>Phase de développement</b>										
Back-end développement		I		A,C		R				
Front-end développement		I		A,C			R			
<b>Phase de tests et déploiement</b>										
Tests Finaux		I			A,C			R		
Déploiement	I	I			A,C			R		

FIGURE 4 – Objectifs

## 5    Approche architecturale

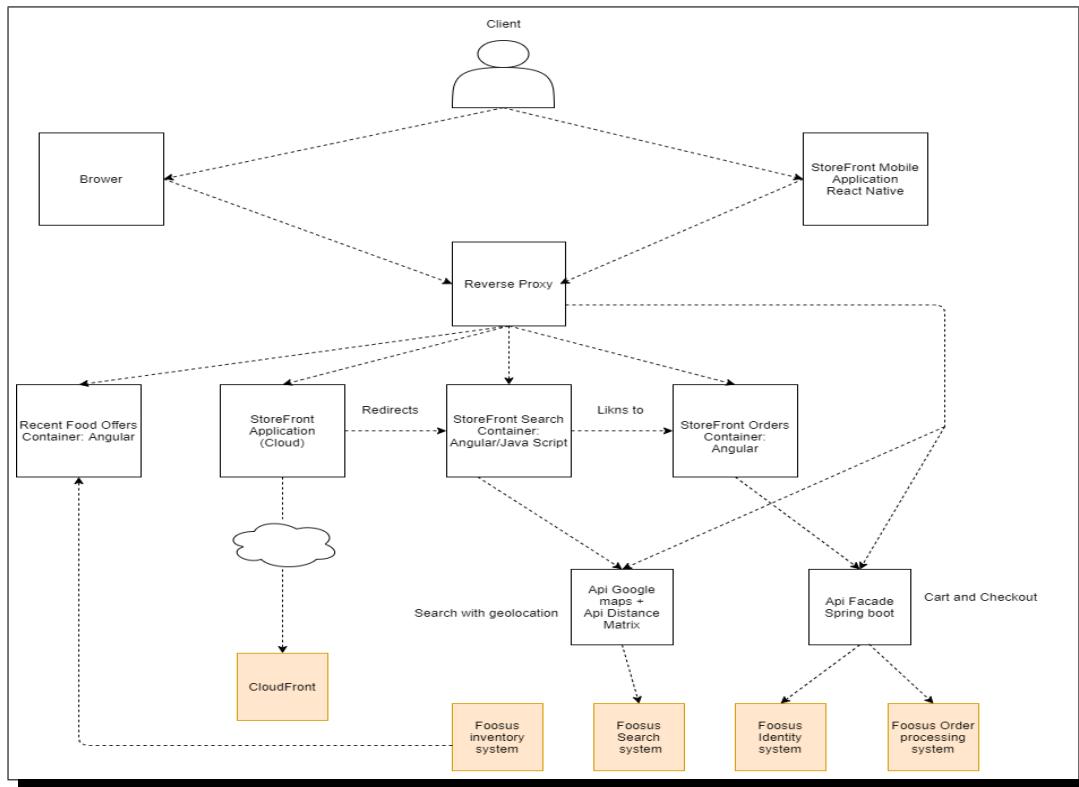


FIGURE 5 – Container Diagram Level 2

### 5.1   Différence entre l'ancien et le nouveau système

- Le nouveau système est standardisé avec seulement 3 technologies (Spring-Boot, Angular, Ionic).
- Ajout de la fonctionnalité de géolocalisation.
- Ajout d'un reverse proxy qui sécurise et rend le nouveau système fluide.
- Utilisation d'un CDN cloud (CloudFront) qui rend la navigation encore plus rapide.
- Adoption du "Canary Deployment" pour ne plus mettre le site hors service lors d'ajout de mise à jours.

## 6    Plan de travail

### 6.1   Element de travail 1

#### 6.1.1   Activité

Présenter la nouvelle architecture à toute l'équipe et s'assurer de la bonne compréhension.

#### 6.1.2   Livrable

### 6.2   Element de travail 2

#### 6.2.1   Activité

Recoder les api qui doivent être recoder et développé l'api Facade pour "Google Maps"

### **6.2.2 Livrable**

Api Facade reliant le front-end et le back-end

## **6.3 Element de travail 3**

### **6.3.1 Activité**

Implémenter le service CDN CloudFront d'amazon pour avoir des temps de chargements relativement court.

### **6.3.2 Livrable**

Un CDN fonctionnel qui route les utilisateurs vers le serveur le plus proche.

## **6.4 Element de travail 4**

### **6.4.1 Activité**

Developper le nouveau Front-End avec les nouvelles interfaces.

### **6.4.2 Livrable**

Des livrables Angular et une nouvelle présentation pour le client.

## **7 Plan de communication**

### **7.1 Évènements**

- Réunions Régulière "Daily Scrum"
- Réunions tous les 15 jours "Sprint"
- Réunions mensuel avec le comité de direction
- Réunions d'urgence.

### **7.2 Canaux**

- Présentiel
- A distance (Teams).

### **7.3 Formats**

- Mail.
- Messagerie Teams.

### **7.4 Durée**

Comme montré dans le diagramme de Gantt la durée du projet est d'environ 5 mois et demi.

### **7.5 Effort**

En suivant le model COCOMO pour le calcul de l'effort on a obtenu les résultats suivants :

- Le Back-End pour une durée de 3 mois demandera 7 programmeur / mois
- Le Front-End pour une durée de 1 mois demandera 6 programmeur.
- L'application mobile pour une durée de 20 jours demandera 4 programmeur.

NB : Le model COCOMO se base sur le nombre de ligne de code pour estimer l'effort ce n'est donc pas applicable pour la (conception et les tests)

## 8 Risques et facteurs de réduction

Description	Gravité	Probabilité	Actions préventives	Actions correctrices
Changement courant dans le cahier des charges	Catastrophique	Peu probable	Se mettre d'accord lors de la réunion de présentation du cahier des charges fonctionnel	Atelier pour figer le périmètre avec la MOA
Manque de compréhension de la nouvelle architecture	Catastrophique	Probable	Expliquer clairement et dans le détail à tous les membres de l'équipe la nouvelle architecture	Travailler en scrum permet de corriger l'erreur assez rapidement
Responsables non satisfait	Grave	Peu probable	Réunion mensuel avec les responsables pour présenter les nouveaux changements	Se remettre d'accord sur la direction du projet et raccourcir la durée entre 2 réunions avec les responsables.
Frictions au sein de l'équipe projet	Mineur	Peu probable	Définir clairement et précisément les rôles et les tâches de chacun dans le projet	Aborder tous les points de litiges et les résoudre par le dialogue
Un ou plusieurs membres de l'équipe tombent malade	Grave	Peu probable	Ne pas laisser les gens malades venir travailler au bureau pour ne pas contaminer d'autres employés	- Si la durée est longue, engager un développeur en freelance - Mettre en place un système de télétravail.
Un des membres de l'équipe perd sa motivation	Mineur	Peu probable	Mettre en place des meetings hebdomadaires pour partager les avancées de tous	Le rencontrer en tête-à-tête pour essayer de comprendre pourquoi il est démotivé
Client a du mal à utiliser la nouvelle interface	Majeur	Peu probable	- Ne pas trop changer l'interface. - Engager un ingénieur QA	- Proposer un tuto lors de la première connexion - Faire un sondage et présenter les résultats à l'équipe UI

FIGURE 6 – Objectifs

## 9 Critères d'acceptation et procédures

### 9.1 Critères d'acceptation

Indicateur	Changement souhaité pour l'indicateur
Nombre d'adhésions d'utilisateurs par jour	Augmentation de 10 %
Adhésion de producteurs alimentaires	Passer de 1,4/mois à 4/mois
Délai moyen de parution*	Réduit de 3,5 semaines à moins d'une semaine
Taux d'incidents de production P1	Pour commencer : réduit de >25/mois à moins de 1/mois.

FIGURE 7 – Indicateurs de réussite

### 9.2 Procédure d'acceptation

- Vérifications des chiffres du nombre de consommateur et des producteurs par la direction (CEO, CIO, CFO, CPO, CMO)
- Tous les incidents lors du déploiement d'une nouvelle mise à jour seront notifiés sur la plateforme JIRA.
- Vérification de l'historique des déploiements des mises à jour par la CIO et la CEO

## 10 Approbations signées

Validateur	Domaine de responsabilité	Date
Ash Callum	CEO	

FIGURE 8 – Approbateur et signature