

Déclaration de Travail d'Architecture

19 juin 2021

Aftis Saadi



Table des matières

1	<u>Informations sur le document</u>	1
2	<u>Objet de ce document</u>	1
3	<u>Déclaration de travail d'architecture</u>	1
3.1	Requête du projet et contexte	1
3.2	Description du projet et périmètre	2
3.3	Vue d'ensemble	2
3.4	Alignement stratégique	2
4	<u>Objectifs et périmètre</u>	3
4.1	Objectifs	3
4.2	Périmètre	3
4.3	Approche managériale	3
5	<u>Rôles et responsabilités</u>	4
5.1	Structure de gouvernance	4
5.2	Process du projet	5
5.3	Rôles et responsabilités	6
6	<u>Approche architecturale</u>	7
6.1	Différences entre l'ancien et le nouveau système	7
7	<u>Plan de travail</u>	8
7.1	Livrable 2	8
8	<u>Plan de communication</u>	8
8.1	Effort	9
9	<u>Risques et facteurs de réduction</u>	9
10	<u>Critères d'acceptation et procédures</u>	10
10.1	Critères d'acceptation	10
10.2	Procédure d'acceptation	10
11	<u>Approbations signées</u>	11

1 Informations sur le document

<i>Nom du projet</i>	Architecture Foosus
<i>Préparé par :</i>	AFTIS Saadi
<i>N° de version du document :</i>	1.0
<i>Titre :</i>	<i>Déclaration de travail d'architecture</i>
<i>Email :</i>	Aftis.Saadi@gmail.com
<i>Types d'action :</i>	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion.

FIGURE 1 – Informations

2 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.

3 Déclaration de travail d'architecture

3.1 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 de 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.

3.2 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 garder ce qui fonctionne déjà correctement.
- Prendre en charge tous les types d'utilisateurs.

3.3 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 leurs proposer le meilleur service.

3.4 Alignement stratégique

Pour commencer tel que 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" sont des Frameworks très utilisés et super efficaces 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 géographique et DistanceMatrix Api pour lister les magasins les plus proches. Finalement, pour que le système supporte plusieurs types d'utilisateurs à savoir fournisseurs, back-office, consommateurs... , la sécurité de Spring-Boot permet l'implémentation d'un système d'accès basé sur les rôles.

4 Objectifs et périmètre

4.1 Objectifs

<i>Objectif Business</i>	<i>Notes</i>
Augmenter le nombre d'inscrit	Nouvelle interface, plus fluide grâce au CDN
Avoir une meilleure image	Mieux sécuriser le système, innover souvent et ne plus avoir de downtime
Viser des clients partout dans le monde	utilisation de la géolocalisation

FIGURE 2 – Objectifs

4.2 Périmètre

Plusieurs conditions définissent le périmètre de ce projet :

- Ne pas mettre le système actuel Hors-service, les deux systèmes doivent coexister le temps de la transition.
- Ne pas dépasser le budget de 50.000 \$ et la durée de six mois.
- Réutiliser dans le nouveau système les fonctionnalités qui fonctionnent déjà correctement.

4.3 Approche managériale

Foosus a remarqué que le déploiement de patch ou de mises à jour volumineuses était contre-productif, d'un côté 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 qui a été fait la veille et ce qui est supposé être fait le jour même. Concernant les livrables réguliers, la méthode "Scrum" utilise les "Sprints" qui sont des périodes d'une vingtaine de jours où on doit à la fin remettre un livrable ce qui en fait des livrable régulier et rarement volumineux.

5 Rôles et responsabilités

5.1 Structure de gouvernance

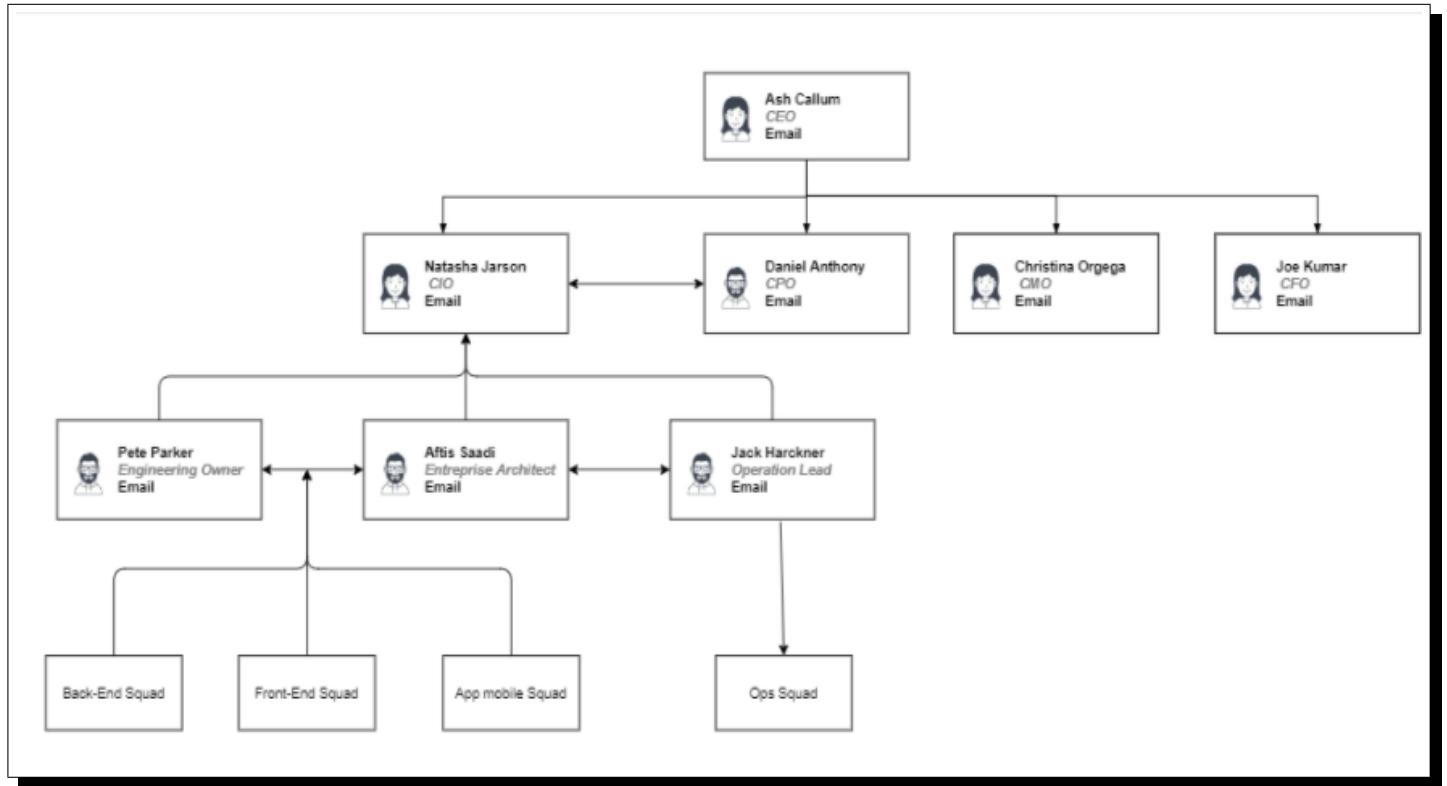


FIGURE 3 – Organisation

5.2 Process du projet

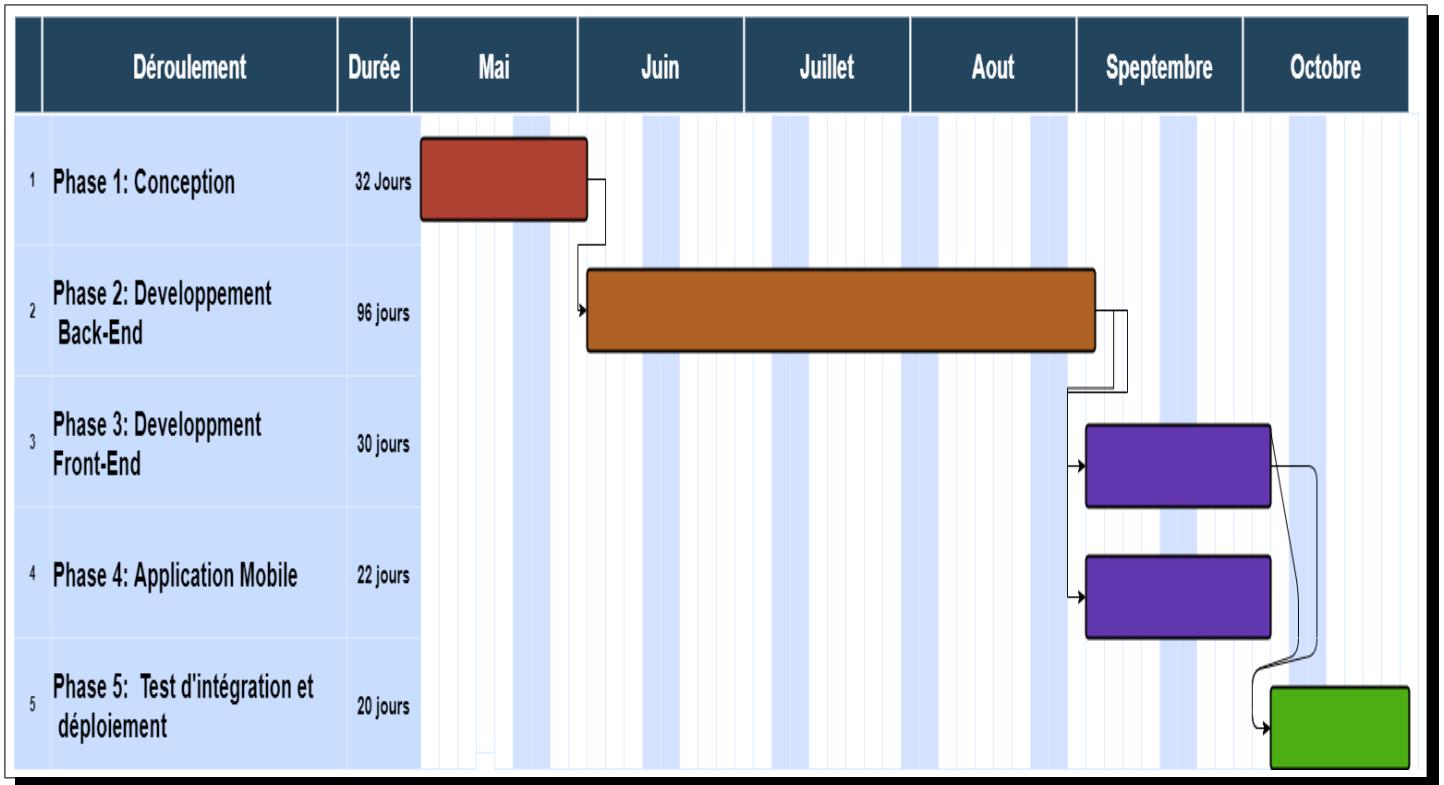


FIGURE 4 – Diagramme de Gantt

5.3 Rôles et responsabilités

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

FIGURE 5 – Matrice de "RACI" montrant les rôles et responsabilités

6 Approche architecturale

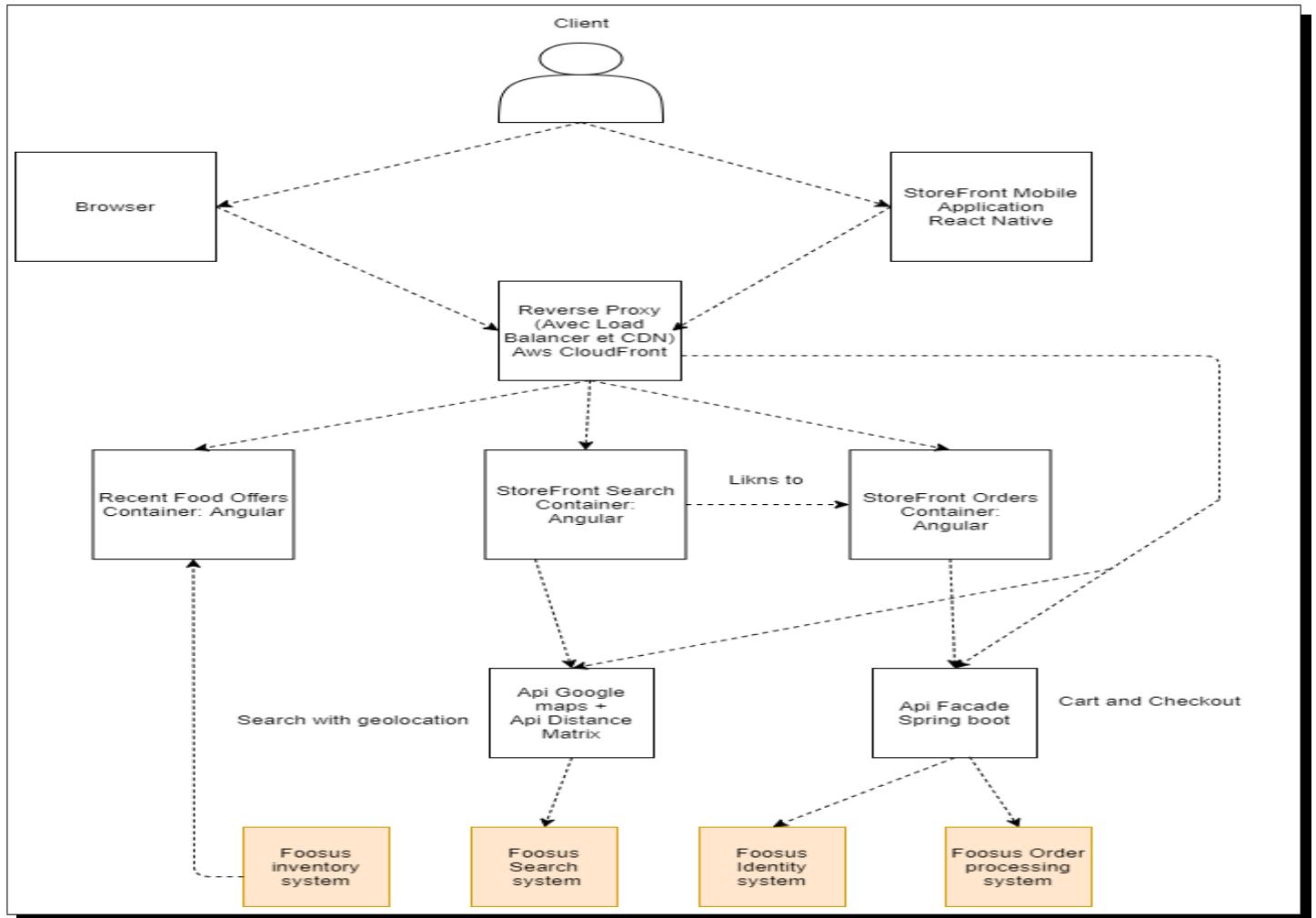


FIGURE 6 – Container Diagram Level 2

6.1 Différences entre l'ancien et le nouveau système

- Le nouveau système est standardisé avec seulement trois 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 "Blue-Green deployment" pour ne plus mettre le site hors service lorsqu'on doit le mettre à jour.

7 Plan de travail

Activité 1

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

Activité 2

Recoder l'api façade "Back-End for frontend" pour lui mettre de nouvelles associations et développer l'api "Google Maps"

7.1 Livrable 2

Système de géolocalisation ainsi que le back-end standardisé avec une seule technologie.

Activité 3

Implémenter le service CloudFront d'amazon qui propose un CDN et un reverse proxy pour avoir des temps de chargements plus court.

Activité 4

Développer le nouveau Front-End avec les nouvelles interfaces.

Livrable 4

Des livrables ".ts, .html et .css" qui composeront le Front-End.

8 Plan de communication

Évènements

- Réunions Régulières "Daily Scrum"
- Réunions tous les 20 jours "Sprint"
- Réunions mensuels avec le comité de direction
- Réunions d'urgences.

Canaux

- Présentiel
- A distance (Teams).

Formats

- Mail.
- Messagerie Teams.

Durée

tel que montré dans le diagramme de Gantt la durée du projet est d'environ cinq mois et demi.

8.1 Effort

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

- Le Back-End pour une durée de 3 mois demandera 3 développeurs.
- Le Front-End pour une durée de 1 mois demandera 3 développeurs.
- L'application mobile pour une durée de 20 jours demandera 2 programmeur.

NB : Le modèle COCOMO se base sur le nombre de lignes de code pour estimer l'effort, ce n'est donc pas applicable pour la conception et les tests.

9 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
Délais impossible	Grave	Probable	Les "Daily Scrum" nous donneront un suivi précis du travail fait et du retard pris .	Si le problème vient de l'organisation alors mettre en place une réunion d'urgence pour réattribuer les tâches. Rallonger les délais, engager des développeurs en plus
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.
Perte de 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
Utilisateurs ne semblent pas comprendre 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 7 – Risques

10 Critères d'acceptation et procédures

10.1 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 8 – Indicateurs de réussite

10.2 Procédure d'acceptation

- Vérifications des chiffres sur les taux de nouvelles inscriptions des consommateurs et producteurs par la direction (CEO, CIO, CFO, CPO, CMO)
- Vérification de l'historique des incidents lors des déploiements d'une nouvelle mise à jour (tous les incidents seront notifiés sur la plateforme JIRA).
- Vérification de l'historique des déploiements des mises à jour par la CIO et la CEO. (Pour vérifier la régularité des maj)

11 Approbations signées

Validateur	Domaine de responsabilité	Date
<i>Ash Callum</i>	CEO	

FIGURE 9 – Approbateur et signature