

SPECIFICATION DES CONDITIONS REQUISES POUR L'ARCHITECTURE



Nom de l'entreprise : Foosus

Nom du projet : Développement d'entreprise

Adresse :

Email :

Tel :

SOMMAIRE

1. INTRODUCTION.....	1
2. CONTEXTE.....	1
3. MESURE DU SUCCES.....	1
4. CONDITIONS REQUISES POUR L'ARCHITECTURE.....	1
5. CONTRATS DE SERVICE BUSINESS.....	2
5.1. Accord de niveau de service.....	2
6. LIGNES DIRECTRICES POUR L'IMPLÉMENTATION.....	2
7. SPECIFICATIONS POUR L'IMPLÉMENTATION.....	3
8. STANDARDS POUR L'IMPLÉMENTATION.....	3
9. CONDITIONS REQUISES POUR L'INTEROPÉRABILITÉ.....	3
10. CONDITIONS REQUISES POUR LE MANAGEMENT DU SERVICE IT.....	4
11. CONTRAINTES.....	4
12. HYPOTHESES.....	5
13. CONCLUSION.....	5

1. INTRODUCTION

Ce document décrit la spécification des conditions requises pour l'architecture pour le projet de développement de l'entreprise FOOSUS. Il supprime toute ambiguïté en décrivant les conditions requises pour l'implémentation de l'architecture mais aussi des conditions pour la conformité de l'implémentation.

2. CONTEXTE

Foosus est une start-up de 3 ans dans le secteur de l'alimentation durable qui souhaite faire passer son entreprise à un niveau supérieur. Son système actuel, ne répondant plus à ses nouvelles exigences, elle souhaite développer une nouvelle architecture .

Le projet consiste à développer une nouvelle architecture d'une plate-forme de commerce électronique sécurisée polyvalente tenant compte des différents objectifs de l'entreprise dont les objectifs Business, les contraintes organisationnelles, fonctionnelles et techniques.

3. MESURE DU SUCCES

Les métriques et KPIs utilisées pour déterminer le succès de ce travail d'architecture sont définies au point 8.1 du document de «déclaration de travail d'architecture».

4. CONDITIONS REQUISES POUR L'ARCHITECTURE

Ci-dessous, une liste de conditions requises pour l'architecture:

1. **Évolutivité** : L'architecture doit être évolutive afin que l'entreprise Foosus puisse déployer ses services sur diverses régions dans des villes et des pays donnés. Cette évolutivité permettra à l'entreprise d'éviter les pannes lors des modifications.
2. **Disponibilité** : L'architecture doit assurer l'utilisation en continue de la plate-forme même durant la maintenance d'un service. En effet, cette disponibilité doit permettre d'éviter de désactiver la plate-forme à chaque installation d'une nouvelle version ou à chaque modification du schéma de la base de données
3. **Temps de réponse** : L'architecture doit réduire, voire éliminer les problèmes de lenteurs de la plate-forme. Cette réduction du temps de réponse est obligatoire notamment pour la fonctionnalité de «recherche fournisseurs alimentaires».
4. **Accessibilité** : La plate-forme doit être accessible de partout dans le monde et avec le même niveau de performance. En effet, même si le système est surchargé, les utilisateurs connectés doivent pouvoir continuer à accéder à tous les services de façon dégradée.
5. **Confidentialité** : La gestion des catégories d'utilisateurs ainsi que les données

auxquelles ils ont accès doit être assurée par la plate-forme. En effet, elle doit pouvoir prendre en charge divers types d'utilisateurs (par exemple, fournisseurs, back-office, consommateurs), avec des fonctionnalités et des services spécifiques pour ces catégories.

6. **Sécurité** : La protection des données qui transitent doit être assurée par la mise en place d'un protocole de sécurité qui permet de crypter les données.

5. CONTRATS DE SERVICE BUSINESS

5.1. Accord de niveau de service

Par cette architecture, nous garantissons :

1. L'évolutivité de la plate-forme permettant aux services de Foosus de se déployer sur diverses régions à travers des villes et des pays donnés.
2. La disponibilité de la plate-forme pour les fournisseurs et les consommateurs où qu'ils se trouvent, ainsi que la gestion des charges.
3. La confidentialité par la gestion des utilisateurs de la plate-forme.
4. La sécurité des données qui transitent sur la plate-forme.
5. La gestion des hauts niveaux de trafics (monter en charges)
6. La géolocalisation des utilisateurs de la plate-forme afin de proposer des produits disponibles près des lieux de résidence des consommateurs.

6. LIGNES DIRECTRICES POUR L'IMPLEMENTATION

Ci-après figure une liste des directives relatives au projet approuvé.

1. Les solutions open source sont préférables aux solutions payantes.
2. Le support continu des composants doit être pris en compte lors de leur sélection ou lors des prises de décision de création ou d'achat.
3. 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 continu.
4. Mise en place de l'intégration continue pour permettre d'effectuer des petits changements et faciliter les rollback.

7. SPECIFICATIONS POUR L'IMPLÉMENTATION

Une spécification est un ensemble explicites d'exigence à satisfaire par un produit. *L'ensemble des exigences de l'architecture ont été définie au point 6.1.3 du document de «déclaration de travail d'architecture».*

8. STANDARDS POUR L'IMPLÉMENTATION

La liste ci-dessous nous présente les standards pour l'implémentation de l'architecture :

1. **Réunion du comité de pilotage ou COPIL** : Elle permet de définir une stratégie d'orientation pour le projet. Elle permet notamment de décider des Go/No Go au démarrage du projet. Elle doit être organisée en amont du Kick-off et renouvelée tout au long du projet en fonction des besoins.
2. **Kick-off ou réunion de lancement du projet** : elle sert à informer et à donner le contexte et les enjeux du projet. C'est lors de ce rendez-vous incontournable que sont fixés les objectifs, la structure et le timing du projet. Cette réunion ne sera organisée qu'au début de la phase de lancement du projet.
3. **La revue de projet** : Cette réunion spécifique permet de préparer le Copil à venir et de suivre la vie du projet en live. Nous allons organiser une revue de projet de manière régulière, surtout avant chaque COPIL.
4. **Point d'avancement** : c'est la réunion de fonctionnement de l'équipe. Elle a pour objectif de valider l'avancement du projet et de déclencher une réunion de travail. Nous allons organiser un point d'avancement régulièrement.
5. **Réunion de clôture** : Elle permet un retour d'expérience et une reconnaissance du travail effectué par les différents contributeurs. Elle est organisée une seule fois, à la fin du projet.

9. CONDITIONS REQUISES POUR L'INTEROPÉRABILITÉ

Pour assurer l'interopérabilité un ensemble de conditions requises dont :

1. Utilisation des micro-services :
2. Communication à faible couplage

10. CONDITIONS REQUISES POUR LE MANAGEMENT DU SERVICE IT

La mise en place de la méthode agile :

La méthode agile est un mode de gestion des projets privilégiant le dialogue entre toutes les parties prenantes ainsi que la souplesse en cours de réalisation et la capacité à modifier les plans et la rapidité de livraison.

La mise en place d'une telle méthode se passe en 7 points :

1. **Définir une équipe SCRUM** : l'équipe peut-être composé de 5 à 9 membres.
2. **Définir la longueur du Sprint** : Un sprint est un intervalle de temps qui va de 7 à 30 jours.
3. **Nommer un Scrum master** : Le Scrum Master est le catalyseur du groupe de Scrum. Il veille à ce que le groupe de Scrum soit efficace et progressif.
4. **Nommer le Product Owner** : Le Product Owner doit être une personne qui peut être chargée de s'assurer que l'équipe produit de la valeur à partir du projet pour l'entreprise.
5. **Créer le backlog du produit final** : Le backlog de produit est une liste de souhaits de tous les récits d'utilisateur (exigences) qui devraient être réalisés dans le projet.
6. **Planifier et lancer le premier Sprint** : Sur la base de la hiérarchisation du backlog, l'équipe peut maintenant choisir des éléments de la liste (généralement à partir du haut). Elle doit faire un brainstorming et décide de ce qu'elle peut réaliser au cours du prochain sprint.
7. **Clôturer le Sprint en cours et commencer le Sprint suivant** : Lorsqu'un sprint est terminé, tout le travail prévu devrait, on l'espère, être terminé. Si ce n'est pas le cas, c'est à l'équipe de décider si le travail restant doit être transféré au sprint suivant ou remis dans le backlog.

11. CONTRAINTES

Les contraintes de l'architectures sont citées au point 6.1.2 du document de «déclaration de travail d'architecture».

12. HYPOTHESES

Li-dessous la liste d'hypothèses :

1. La plate-forme existante sera conservée en mode maintenance. Aucune nouvelle fonctionnalité n'y sera ajoutée.
2. Cette nouvelle architecture (architecture en microservices) est construite autour des technologies actuelles et a pour avantage de pouvoir s'adapter aux nouvelles technologies futures.
3. Les dirigeants doivent éviter les faux raccourcis pour éviter tout nouveau comportement dans le système existant car les équipes sont attachées à la plate-forme existante.
4. L'offre initiale impliquera la coexistence de deux plate-formes et la montée en puissance empirique du volume d'utilisateurs qui migreront vers la nouvelle plateforme à mesure que le produit évoluera. Cette augmentation sera proportionnelle à l'évolution des fonctionnalités. Par exemple, les utilisateurs précoces pourront choisir d'utiliser les nouvelles fonctionnalités de recherche intégrées au processus de paiement existant.
5. La géolocalisation permettra d'introduire d'autres innovations en fonction de l'emplacement de l'utilisateur ou du fournisseur alimentaire, si elle est modélisée suffisamment tôt dans la nouvelle plate-forme.
6. L'élaboration sur mesure d'une approche architecturale de type « lean » pourra contribuer à la réalisation de cette feuille de route, ce qui évitera de priver les équipes de leur autonomie et de compromettre la rapidité des cycles de versions.

13. CONCLUSION

Ce document a été réalisé, conformément à la déclaration du travail d'architecture du projet de développement de l'entreprise Foosus, dans le but de spécifier l'ensemble des conditions requises pour l'implémentation de l'architecture recommandée. En effet, La Spécification des Conditions requises pour l'Architecture fournit une vision quantitative de la solution, énumérant des critères mesurables qui doivent être remplis durant l'implémentation de l'architecture.