



# Spécification des Conditions requises pour l'Architecture

---

*Projet : Foosus*

*Client : Foosus*

*Préparé par : Degruson Julien*

*N° de Version du Document : 0.1*

*Titre : Spécification des Conditions requises pour l'Architecture*

*Date de Version du Document : 12/03/2023*

*Revu par :*

*Date de Révision :*

## Table des Matières

---

1. Objet de ce document
2. Mesures du succès
3. Conditions requises pour l'architecture
4. Contrats de service business

5. Contrats de service application
6. Lignes directrices pour l'implémentation
7. Spécifications pour l'implémentation
8. Standards pour l'implémentation
9. Conditions requises pour l'interopérabilité
10. Conditions requises pour le management du service IT
11. Contraintes
12. Hypothèses

## Objet de ce document

---

*La Spécification des Conditions requises pour l'Architecture fournit un ensemble de déclarations quantitatives qui dessinent ce que doit faire un projet d'implémentation afin d'être conforme à l'architecture.*

*Une Spécification des Conditions requises pour l'Architecture constitue généralement un composant majeur du contrat d'implémentation, ou du contrat pour une Définition de l'Architecture plus détaillée.*

*Comme mentionné ci-dessus, la Spécification des Conditions requises pour l'Architecture accompagne le Document de Définition de l'Architecture, avec un objectif complémentaire : le Document de Définition de l'Architecture fournit une vision qualitative de la solution et tâche de communiquer l'intention de l'architecte.*

*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.*

## Mesures du succès

---

### Indicateurs de réussite

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.

# Conditions requises pour l'architecture

---

Foosus ayant l'ambition de s'agrandir pour suivre les tendances du marché, l'architecture qu'ils avaient lorsqu'ils ont commencé n'est plus adéquate. Leurs objectifs sont les suivants :

## **Évoluer avec notre base de clientèle**

La pile technologique doit être conçue de façon à évoluer naturellement au même rythme que notre base de clientèle. Nous avons constaté des pannes causées par l'incapacité de notre système logiciel d'absorber les pics d'utilisation par les clients ou les programmes marketing. Les contraintes de performances du système existant empêchaient de supporter le niveau d'engagement et de croissance attendus de nos futurs programmes marketing. Même si le système est surchargé, les utilisateurs connectés doivent pouvoir continuer à accéder à tous les services de façon dégradée.

## **Une plateforme sécurisée, utilisable et réactive**

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 souhaitons en outre qu'elle soit facile à adapter aux particularités locales et qu'elle réponde aux exigences d'utilisation de nos clients. Par le passé, nous avons fait primer la facilité d'utilisation sur la sécurité. Ce parti pris a failli nuire plusieurs fois à notre réputation. Nous voulons éviter tout risque pour notre image de marque et nous avons besoin d'une approche qui garantisse la sécurité chaque fois que nous élargirons la plateforme.

## **Une technologie transparente**

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. Notre marché cible des 4 villes dans le monde entier. Le temps où nous pouvions bénéficier d'une accalmie au cœur de la nuit est révolu. Nous sommes actifs 24 h/24 ! Chaque nouvelle version doit être de taille réduite, présenter peu de risques, être transparente pour nos utilisateurs et rester accessible en tout lieu et à tout moment. C'est quand nos utilisateurs peuvent accéder facilement à nos services et apprécient notre produit que nous réussissons. Les utilisateurs situés dans différentes régions doivent pouvoir espérer des performances similaires. Nous voulons cibler les consommateurs dans des zones géographiques spécifiques, sur des connexions lentes (par exemple, avec des téléphones portables) aussi bien que sur des réseaux haut débit. Toutes les solutions doivent pouvoir répondre à cette exigence. Une évolutivité capable d'accompagner la croissance. L'an dernier, 12 de nos pannes ont été provoquées par la publication par une ou plusieurs équipes de modifications lourdes qui n'ont pas eu les résultats escomptés. Nous avons également eu des difficultés à intégrer les travaux réalisés par différentes équipes sur des modifications de notre plateforme qui n'avaient pas de lien entre elles. Nous ne devrions pas avoir ce genre de problème en tant que petite entreprise. La difficulté vient du temps nécessaire pour que chaque nouvelle version logicielle soit vue par les autres équipes ou testée dans nos environnements de production. Nous devons combler le fossé entre le moment où une ligne de code est écrite et celui où elle est validée dans un environnement intégré. Cela peut également nous aider à déterminer les réactions de nos clients vis-à-vis de nouvelles fonctionnalités à mesure que nous développons ces dernières.

# Contrats de service business

---

## Accords de niveau de service

---

Les investisseurs et les clients ont différentes métriques pour mesurer la valeur de Foosus. Les investisseurs vont se concentrer sur le nombre de nouveaux inscrits, tandis que les clients sur la qualité de l'algorithme pour le choix des fournisseurs, ainsi que la disponibilité de l'application.

## Contrats de service application

---

### Objectifs de niveau de service

---

#### Indicateurs de réussite

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.

## Lignes directrices pour l'implémentation

---

Il faut garder en tête les principes togap :

Principe 2 : Maximiser les avantages pour l'entreprise

Principe 4 : Continuité des activités

Principe 20 : Contrôle de la diversité technique

# Spécifications pour l'implémentation

---

	Principe 2 : Maximiser les avantages pour l'entreprise	Principe 4 : Continuité des activités	Principe 20 : Contrôle de la diversité technique
Augmentation du nombre d'adhésion	✓		
Augmentation du taux d'adhésion des producteurs alimentaire	✓		
Délai des mises à jour		✓	
Diminution du taux d'incident en production		✓	
Changement de paradigme d'architecture			✓

## Standards pour l'implémentation

---

Les standards pour l'implémentation peuvent inclure l'utilisation de l'architecture REST pour faciliter la communication entre les différents services de l'application, l'adoption d'un coding style pour garantir la cohérence et la maintenabilité du code, et l'utilisation d'un framework front pour accélérer le développement de l'interface utilisateur et assurer la compatibilité avec les différents navigateurs.

L'utilisation de l'architecture REST (Representational State Transfer) permet de créer des services web qui peuvent être facilement consommés par d'autres parties de l'application. Un coding style standard assure la lisibilité et la cohérence du code, ce qui facilite la maintenance

et le débogage. Enfin, l'utilisation d'un framework front (comme React, Angular ou Vue.js) fournit des outils et des bibliothèques qui accélèrent le développement et assurent la compatibilité avec les navigateurs modernes.

## Conditions requises pour l'interopérabilité

---

Pour assurer l'interopérabilité d'une application basée sur REST avec des microservices, il est essentiel de suivre certaines conditions :

**Utiliser des APIs standardisées :** Les APIs REST standardisées permettent aux différents composants d'une application de communiquer entre eux. Il est important de suivre les conventions de nommage, les méthodes HTTP et les formats de données pour assurer l'interopérabilité.

**Respecter les principes de conception REST :** Les principes de REST, tels que l'utilisation de ressources et de méthodes HTTP standardisées, facilitent l'interopérabilité en fournissant une architecture uniforme pour les interactions entre les différents composants.

**Assurer la compatibilité des formats de données :** JSON est un format de données couramment utilisé avec REST, il est donc important de s'assurer que les différentes parties de l'application utilisent les mêmes conventions de formatage pour les données échangées.

**Implémenter des microservices :** Les microservices offrent une approche modulaire pour la conception d'applications, permettant aux différentes parties de l'application de communiquer entre elles via des APIs REST. Cela facilite l'interopérabilité et permet à chaque microservice de fonctionner indépendamment.

**Assurer la sécurité :** L'utilisation de SSL/TLS pour chiffrer les données échangées entre les différents composants de l'application est essentielle pour assurer la confidentialité et l'intégrité des données.

## Conditions requises pour le management du service IT

---

Pour assurer un management efficace des services IT en utilisant une approche Agile et Scrum, il est important de suivre certaines conditions clés:

**Adopter une approche itérative :** L'approche Agile et Scrum est basée sur des itérations courtes et régulières pour la planification, la conception, le développement et le déploiement.

Utiliser les méthodes de planification Agile : Les méthodes de planification Agile telles que le Backlog Produit et le Backlog Sprint sont essentielles pour une planification efficace. Il est important de hiérarchiser les exigences et les fonctionnalités pour garantir une livraison de manière itérative et incrémentale.

Favoriser la collaboration : Le travail en équipe et la collaboration entre les différents départements sont des éléments clés d'une approche Agile et Scrum. Cela permet de partager très rapidement les informations pour éviter des surprises et du temps dépensé inutilement.

Suivre les principes de Scrum : Le cadre de travail Scrum est basé sur des rôles clés tels que le Product Owner, le Scrum Master et l'équipe de développement et de test.

Adopter une approche d'amélioration continue : L'approche Agile et Scrum met l'accent sur l'amélioration continue pour garantir que les services IT répondent aux besoins de l'entreprise.

## Contraintes

---

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.

## Hypothèses

---

- La nouvelle architecture sera construite en fonction des technologies actuelles et avec la capacité de s'adapter à de nouvelles technologies lorsque celles-ci seront disponibles.
- L'offre initiale implique la coexistence de deux plateformes 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.
- La géolocalisation, si elle est modélisée suffisamment tôt dans la nouvelle plateforme, permettra d'introduire d'autres innovations en fonction de l'emplacement de l'utilisateur ou du fournisseur alimentaire.
- 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.

L'architecture de type lean est tout à fait compatible avec une architecture microservices. En fait, l'approche lean est souvent considérée comme étant un prérequis pour une implémentation réussie de l'architecture microservices.

Les deux approches partagent de nombreuses similitudes, notamment en ce qui concerne l'accent mis sur la modularité, la simplicité, la collaboration et l'amélioration continue. Les microservices sont conçus pour être facilement déployables, évolutifs et remplaçables, ce qui est en ligne avec l'approche lean.

En adoptant une architecture de type lean, les équipes de développement peuvent s'assurer que chaque microservice est conçu pour répondre à un besoin spécifique et qu'il est facilement intégrable avec les autres services du système. En outre, les équipes peuvent utiliser des outils d'analyse de données pour mesurer la performance du système et identifier les zones qui nécessitent des améliorations, ce qui est en ligne avec l'approche lean.