

# Spécification des Conditions requises pour l’Architecture

Projet : Foosus Géoconscient

Client : Foosus

Préparé par : Ludovic SOUPLET

N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l’Architecture

Date de Version du Document : 20 septembre 2023

Revu par :

Date de Révision :

Table des matières

[Spécification des Conditions requises pour l’Architecture 1](#_Toc146115595)

[Objet de ce document 4](#_Toc146115596)

[Mesures du succès 1](#_Toc146115597)

[Conditions requises pour l’architecture 1](#_Toc146115598)

[Contrats de service business 1](#_Toc146115599)

[Accords de niveau de service 1](#_Toc146115600)

[Contrats de service application 1](#_Toc146115601)

[Objectifs de niveau de service 1](#_Toc146115602)

[Indicateurs de niveau de service 1](#_Toc146115603)

[Lignes directrices pour l’implémentation 1](#_Toc146115604)

[Spécifications pour l’implémentation 1](#_Toc146115605)

[Standards pour l’implémentation 1](#_Toc146115606)

[Conditions requises pour l’interopérabilité 1](#_Toc146115607)

[Conditions requises pour le management du service IT 1](#_Toc146115608)

[Contraintes 1](#_Toc146115609)

[Hypothèses 2](#_Toc146115610)

# 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

La mesure du succès de la nouvelle plateforme se fera au travers des indicateurs ci-dessous :

|  |  |  |  |
| --- | --- | --- | --- |
| **Métrique** | **Technique de mesure** | **Valeur cible** | **Justification** |
| Nombre d’adhésions d’utilisateurs par jour | Requêtage sur la plateforme et affichage sur un dashboard de suivi | Augmentation de 10% | Recul des inscriptions sur les derniers mois. C’est l’indicateur clé de la réussite du projet |
| Adhésion de producteurs alimentaires | Requêtage sur la plateforme et affichage sur un dashboard de suivi | Passer de 1,4/mois à 4/mois | Plus de producteurs c’est une attractivité supplémentaire pour attirer de nouveaux clients |
| Délai moyen de mise en production | Modification automatique de la date de « dernière MEP » sur le dashboard de suivi. Alerte en cas de dépassement de la valeur cible et valeur cible +2j | Réduit de 3,5 semaines à moins d'une semaine | Réduire la taille des MEP afin de réduire les impacts sur la plateforme  Retour utilisateur plus aisé sur les fonctionnalités car moins « perdues » dans un lot de nouvelle fonctionnalités |
| Taux d'incidents de production P1 | Suivi de l’uptime via les outils du clouder choisi (cloudwatch par exemple pour AWS) | Réduit de >25/mois à moins de 1/mois | Augmentation de la satisfaction clientèle et de la réputation de l’application |
| Délai de réponse des requêtes | Suivi du délai de réponses via les outils du clouder choisi (cloudwatch par exemple pour AWS) | Moins de 1 secondes en moyennes | Satisfaction utilisateurs  Compatibilité maximum avec les connexions lentes et performances similaires entre les utilisateurs |
| Taux de conversion après une recherche | Requêtage sur la plateforme et affichage sur un dashboard de suivi | 75% | Actuellement, le taux de conversion après une recherche est de 52%, c’est trop bas |

# Conditions requises pour l’architecture

* L'architecture doit être conçue pour optimiser le rapport qualité-prix
* L'architecture peut inclure de nouveaux composants personnalisés ou des composants commerciaux pour favoriser la flexibilité, la stabilité et l'extensibilité
* Les solutions open source sont préférées aux solutions payantes
* Il faut tenir compte du support continu des composants lors de leur sélection ou de la décision d'achat
* Dans la mesure du possible, toutes les solutions, qu'elles soient commerciales ou open source, doivent faire partie d'une même pile technologique afin de réduire les coûts de maintenance et de support continus
* L’architecture doit prendre en compte le double run avec l’architecture historique le temps que la migration soit effective
* L’architecture doit prendre en compte l’internationalisation de la plateforme
* L’architecture doit être scalable
* Les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.
* Nos fournisseurs et nos consommateurs doivent pouvoir accéder à notre solution 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.

# Contrats de service business

## Objectifs de niveau de service

Permettre aux parties prenantes qui interagissent avec le process business :

* Les fournisseurs alimentaires soumettront à Foosus un inventaire des produits alimentaires disponibles
* Recherche dans l’interface client et commande de produits de consommation.
* Les clients des produits de consommation trouveront et commanderont des produits alimentaires
* Les fournisseurs alimentaires recevront des commandes
* L’équipe finance de Foosus recevra les paiements

## Accords de niveau de service

1. Disponibilité de l'Application :

- Objectif : Assurer que l'application est disponible pour les utilisateurs

- SLA : Une disponibilité de 99,9 %, ce qui signifie que l'application ne doit pas être hors service plus de 0,1 % du temps (environ 8,76 heures par an).

2. Temps de Réponse :

- Objectif : Garantir des temps de réponse rapides aux demandes des utilisateurs en toute circonstance

- SLA : Un temps de réponse moyen inférieur à 1 secondes pour les requêtes de recherche.

3. Sécurité des Données :

- Objectif : Protéger les données des utilisateurs contre les violations de sécurité.

- SLA : Cryptage des données, audits de sécurité réguliers, et une réponse rapide en cas de violation.

4. Support Client :

- Objectif : Fournir un support client efficace pour résoudre les problèmes et répondre aux questions.

- SLA : Délais de réponse pour les demandes d'assistance doit être inférieur à 10h

5. Délais de Résolution des Problèmes :

- Objectif : Minimiser les temps d'arrêt et les perturbations de service.

- SLA : Délais de résolution pour les incidents et les problèmes techniques inférieur à 4h

6. Notification des Utilisateurs :

- Objectif : Informer les utilisateurs des interruptions de service planifiées ou imprévues.

- SLA : Le délais de notification en cas de maintenance planifiée doit être d’une semaine ou pour un incident majeur doit être d’une heure

# Contrats de service application

## Objectifs de niveau de service

Maintenir un haut niveau de qualité afin de satisfaire au contrat de service business

## Indicateurs de niveau de service

1. Disponibilité de l'Application :

- Objectif : Assurer que l'application est disponible pour les utilisateurs

- SLA : Une disponibilité de 99,9 %, ce qui signifie que l'application ne doit pas être hors service plus de 0,1 % du temps (environ 8,76 heures par an).

2. Temps de Réponse :

- Objectif : Garantir des temps de réponse rapides aux demandes des utilisateurs en toute circonstance

- SLA : Un temps de réponse moyen inférieur à 1 secondes pour les requêtes de recherche.

3. Sécurité des Données :

- Objectif : Protéger les données des utilisateurs contre les violations de sécurité.

- SLA : Cryptage des données, audits de sécurité réguliers, et une réponse rapide en cas de violation.

4. Mises à Jour et Correctifs :

- Objectif : Maintenir l'application à jour et sécurisée.

- SLA :

* Fréquence des mises à jour et des correctifs d’une fois par semaine
* Délais de déploiement en cas de vulnérabilités critiques inférieur à 2h

5. Sauvegarde et Récupération des Données :

- Objectif : Assurer la sauvegarde régulière des données et la capacité de les récupérer en cas de besoin.

- SLA :

* Fréquence des sauvegardes de 12h
* Délais de récupération des données en cas d’avarie de 12h
* Les sauvegardes doivent être testés pour être validées
* Les sauvegardes ne doivent pas être stockés au même endroit que les données d’origines

6. Conformité Réglementaire :

- Objectif : Assurer la conformité aux réglementations applicables.

- SLA : Respect du RGPD

7. Plan de Reprise d'Activité (PRA) :

* Objectif : Établir un plan de reprise d'activité en cas de catastrophe ou de perturbation majeure.
* SLA : Un exercice de PRA devra être réalisé une fois par an

# Lignes directrices pour l’implémentation

La nouvelle plateforme devra répondre à des exigences de scalabilité, performances et de livraisons rapides. De plus, dans un premier temps elle devra fonctionner en parallèle de l’ancienne plateforme avant son décommissionnement.

Afin de répondre à ces exigences, une solution Micro-Service avec hébergement dans le cloud est proposé. Les détails de cette solution se trouve dans la section suivante.

# Spécifications pour l’implémentation

L'architecture **micro services** est un style de conception logicielle où une application est décomposée en petits services autonomes qui travaillent ensemble pour fournir une fonctionnalité complète. Chaque service est indépendant, communique via des interfaces, peut être développé et déployé séparément, et offre une meilleure évolutivité et résilience. Cette approche est particulièrement adaptée aux applications complexes et évolutives.

Pour ce faire, chaque micro service doit être conteneurisé. La **conteneurisation** est une méthode qui permet d'encapsuler des applications et leurs dépendances dans des environnements isolés et portables, appelés conteneurs. Cela facilite le déploiement, la gestion et la portabilité des applications tout en offrant une isolation et une flexibilité accrues. Elle est essentielle pour le développement et le déploiement d'applications modernes.

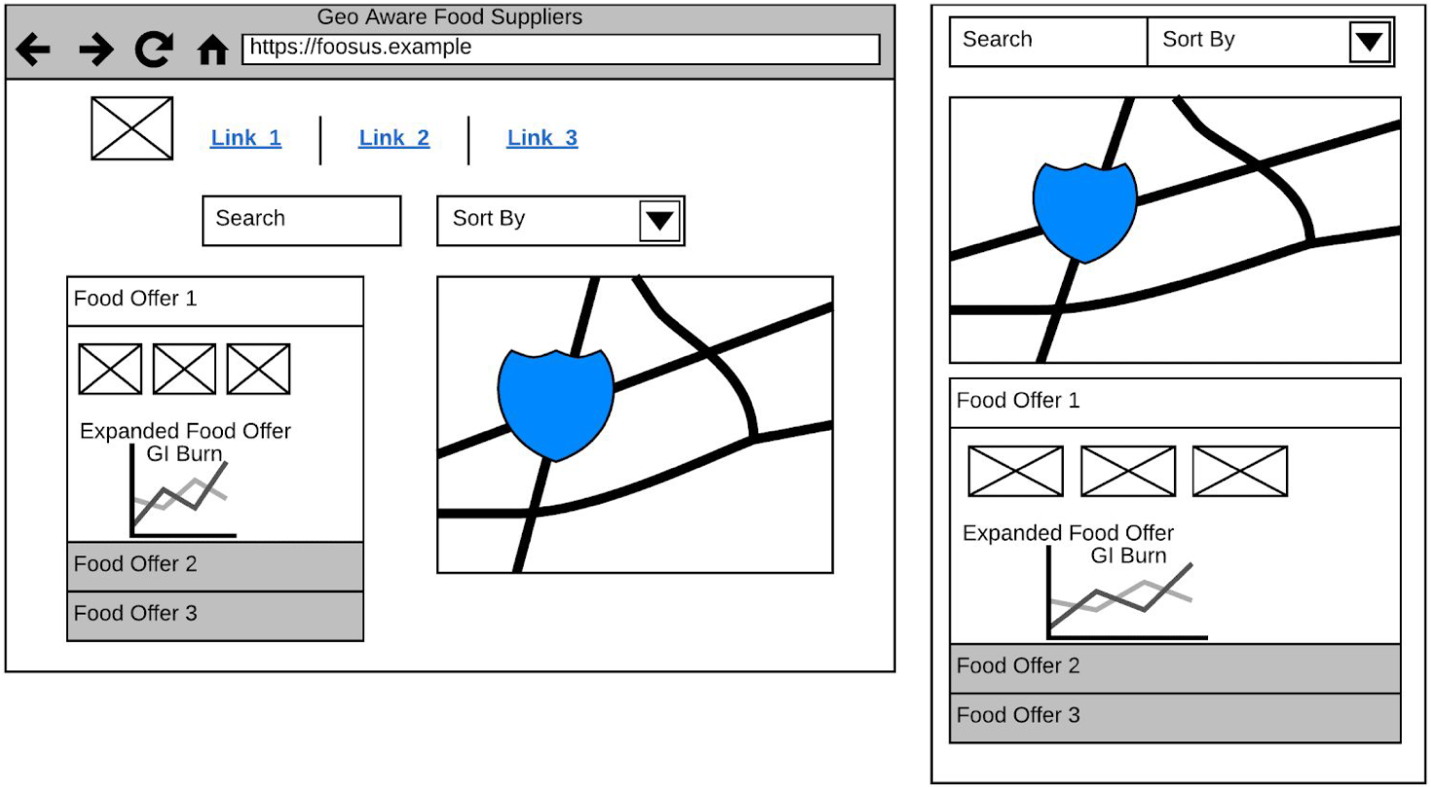
Afin de gérer l’ensemble des conteneurs (un conteneur = un micro service) qui constitue notre application, nous utiliserons l’orchestration. **L'orchestration** pour une application micro services consiste à coordonner et à gérer les différents services micro services qui composent l'application, en s'assurant qu'ils fonctionnent harmonieusement ensemble. Elle inclut la gestion du déploiement, de la mise à l'échelle, de la supervision et de la coordination des services pour garantir la performance et la fiabilité de l'ensemble de l'application.

Afin de répondre au besoin de performance tout autour du monde, nous utiliserons le principe du cloud régionalisé. Le **cloud régionalisé** pour une application micro services signifie que l'application est déployée sur des centres de données dans différentes régions géographiques du cloud. Cela permet d'améliorer la disponibilité, la résilience et les performances de l'application en fonction de la localisation des utilisateurs, tout en garantissant que les services micro services sont répartis de manière stratégique pour servir efficacement différentes régions.

Afin d’automatiser le déploiement des nouvelles itérations nous utiliserons une **chaîne CI/CD**.

Une CI/CD (Intégration Continue/​Livraison Continue) pour une application micro services est un processus automatisé qui permet de développer, tester et déployer rapidement et de manière cohérente chaque micro service, garantissant ainsi une mise à jour continue et efficace de l'application. Cela favorise l'agilité, la qualité et la stabilité de l'ensemble de l'application.

Une spécification **d’UI** pour l’implémentation de la nouvelle fonctionnalité est également fournie. Voir les maquettes ci-dessous.



# Standards pour l’implémentation

* Les standards pour l'implémentation sont des règles, des normes ou des conventions strictes qui doivent être suivies de manière rigoureuse. Ils établissent un cadre précis pour l'exécution d'une tâche ou la création d'un produit.
* Ces standards sont hautement contraignants et visent à garantir une cohérence et une qualité uniformes. Ils sont généralement utilisés dans des domaines où la conformité est essentielle.
* Exemple : Les standards pour l'implémentation de la gestion de la qualité ISO 9001 imposent des exigences spécifiques en matière de documentation, de processus, de suivi et d'amélioration continue.

# Conditions requises pour l’interopérabilité

API rest

Utilisation de produit respectant les normes de l’industrie (process de validation avant utilisation)

# Conditions requises pour le management du service IT

# Contraintes

# Hypothèses