

Spécification des Conditions requises pour l'Architecture

Projet : Approvisionnement alimentaire géographiquement responsable

Client : Foosus

Préparé par : Evan GUEGUEN

N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l'Architecture

Date de Version du Document : 09/05/2025

Revu par : [À compléter]

Date de Révision : [À compléter]

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
-

1. Objet de ce document

Ce document fournit une liste de conditions quantitatives et techniques que le projet "Approvisionnement alimentaire géographiquement responsable" doit satisfaire pour être conforme à l'architecture cible. Il complète le document de définition de l'architecture en explicitant des critères mesurables pour guider l'implémentation technique de la nouvelle plateforme Foosus.

Cette spécification vise à répondre aux problèmes structurels de la plateforme Foosus actuelle, identifiés dans la vision architecturale, en apportant une architecture cible qui soutient l'innovation, l'extensibilité, la stabilité, et la gouvernance de la performance métier et technique.

2. Mesures du succès

Indicateur	Objectif
Adhésions utilisateurs/jour	Augmentation de 10 %
Adhésion producteurs alimentaires	Passer de 1,4 à 4 par mois
Délai moyen de mise en ligne	Réduction de 3,5 semaines à < 1 semaine
Incidents critiques (P1)	Réduction de >25/mois à <1/mois

- Permettre la **coexistence** avec l'ancienne plateforme sans dégrader l'expérience utilisateur.
- Offrir une **expérience stable et géolocalisée**, pour regagner la confiance des utilisateurs et des investisseurs.
- Créer une architecture qui permet l'**expérimentation contrôlée et rapide**, supportant l'apprentissage agile.
- Offrir une visibilité temps réel sur les **performances métiers et techniques** (observabilité).
- Réduire la **dette technique** structurelle en construisant une plateforme "future-proof".

3. Conditions requises pour l'architecture

- Architecture **modulaire et évolutive** basée sur microservices.

- Prise en charge native de la **géolocalisation** pour le tri et la recherche.
 - **Haute disponibilité** (24/7), sans downtime pendant les déploiements.
 - Compatibilité **mobile-first**, avec UX responsive pour bandes passantes réduites.
 - Systèmes d'**authentification sécurisée**, séparés selon les rôles : consommateurs, fournisseurs, back-office.
 - Capacité à absorber des **pics de charge** sans rupture de service.
 - Plateforme **multi-régionale**, localisable et optimisée pour différents marchés.
 - Observabilité intégrée (logs, métriques, alertes).
-
- Pression forte sur les **taux d'inscriptions** comme métrique vitale aux yeux des parties prenantes.
 - Besoin impératif de **réduire les interruptions de service visibles** et de sortir des mises à jour sans downtime.
 - Continuer à opérer avec des **équipes surchargées** qui doivent maintenir l'existant tout en construisant le nouveau.

4. Contrats de service business

Principes Business

- L'architecture doit être un **levier d'innovation et de réputation**.
- Favoriser une gouvernance claire et **évolutive** des services.

Principes Techniques

- **Fail fast, learn fast** : itération, rollback, versionning facilité.
- Préférence pour les solutions **prédictibles, catalogables, et modifiables**.
- Favoriser des **interfaces découplées** (API ouvertes, pas de dépendance cyclique).
- Alignement technologique avec la **capacité organisationnelle**.

Principes Data

- **Protection native des données personnelles.**
 - Cohérence adaptée au besoin métier, pas de cohérence forte par défaut.
 - Représenter explicitement le **modèle de domaine** dans l'architecture applicative.
 -
-

5. Contrats de service application

- Taux d'erreur < 0,1 % sur les opérations critiques (paiement, panier, recherche).
 - Temps de réponse moyen < 2s sur les opérations principales.
 - Support de la **coexistence entre ancienne et nouvelle plateforme** jusqu'à migration complète.
 - Suivi des fonctionnalités expérimentales (tests A/B, rollback rapides).
 - Recherche client → commande → réception fournisseur → paiement finance.
 - Flux à modéliser pour **permettre la personnalisation géographique**.
-

6. Lignes directrices pour l'implémentation

- Favoriser le **déploiement incrémental** avec livraison continue.
- Prioriser la séparation des responsabilités pour limiter les dépendances entre modules.
- Intégration progressive d'un **design system UX** unifié.
- Support des pratiques **DevSecOps** dès le début du cycle.
- Toute la chaîne technologique (infra, runtime, devtools) doit être **réarchitecturée** pour soutenir l'état cible.

- Objectif : plateforme **dev-friendly**, compatible avec les feuilles de route produit et technique.
-

7. Spécifications pour l'implémentation

- Mise en œuvre d'un **calculateur de distance géographique** basé sur la localisation utilisateur.
 - Indexation géospatiale des offres fournisseurs.
 - Interface utilisateur personnalisée selon le profil (fournisseur, consommateur).
 - Implémentation de **mécanismes de cache et de CDN** pour les zones à faible débit.
-

8. Standards pour l'implémentation

- Architecture RESTful avec support GraphQL en option.
 - Stockage sur **base de données relationnelle** pour les transactions, NoSQL pour les recherches/filtres.
 - API sécurisées (OAuth2, JWT).
 - Utilisation des normes d'**accessibilité numérique** (WCAG 2.1).
-

9. Conditions requises pour l'interopérabilité

- Intégration avec des **systèmes de paiement tiers** (Stripe, PayPal, etc.).
 - Ouverture aux systèmes de messagerie pour communication fournisseur/utilisateur.
 - API documentées pour **interopérabilité avec des partenaires externes**.
-

10. Conditions requises pour le management du service IT

- Tableau de bord de **suivi de performance et incidents**.
 - Mise en place d'un **pipeline CI/CD** avec automatisation des tests.
 - Logs centralisés avec alertes proactives.
 - Référentiel de **tests d'intégrité architecturale**.
-

11. Contraintes

- Nécessité de maintenir **l'ancienne plateforme en parallèle**.
 - Ressources internes limitées : coordination requise pour éviter les conflits inter-équipes.
 - Déploiement dans des zones avec **connectivité inégale** (priorité mobile & UX fluide).
-

12. Hypothèses

- Aucune nouvelle fonctionnalité ne sera ajoutée à l'ancienne plateforme.
- Montée en charge progressive des utilisateurs vers la nouvelle architecture.
- La géolocalisation constituera un **pivot d'innovation** pour d'autres cas d'usage.
- Une approche **lean et modulaire** est requise pour préserver l'agilité des équipes.