

Spécification des Conditions requises pour l’Architecture

*Projet :* Projet FOOSUS

*Préparé par :* Julien GUTTER

*N° de Version du Document :* 0.1

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

*Date de Version du Document : Août 2025*

*Revu par :*

*Date de Révision :*

Table des matières

[Table des Matières 2](#_Toc206612219)

[Objet de ce document 3](#_Toc206612220)

[Mesures du succès 3](#_Toc206612221)

[Conditions requises pour l’architecture 3](#_Toc206612222)

[Contrats de service business 5](#_Toc206612223)

[Accords de niveau de service 5](#_Toc206612224)

[Contrats de service application 6](#_Toc206612225)

[Objectifs de niveau de service 6](#_Toc206612226)

[Indicateurs de niveau de service 6](#_Toc206612227)

[Lignes directrices pour l’implémentation 6](#_Toc206612228)

[Spécifications pour l’implémentation 7](#_Toc206612229)

[Standards pour l’implémentation 7](#_Toc206612230)

[Conditions requises pour l’interopérabilité 7](#_Toc206612231)

[Conditions requises pour le management du service IT 8](#_Toc206612232)

[Contraintes 8](#_Toc206612233)

[Hypothèses 8](#_Toc206612234)

# **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**

Les mesures suivantes, alignées sur les KPI du projet Foosus, seront utilisées pour évaluer la conformité de l’architecture :

* Adhésions d’utilisateurs : Augmentation de 10 % des adhésions quotidiennes par rapport à la plateforme existante.
* Adhésions de producteurs : Passage de 1,4 à 4 adhésions mensuelles de producteurs alimentaires.
* Délai de parution : Réduction du délai moyen de parution de 3,5 semaines à <1 semaine.
* Incidents P1 : Réduction des incidents critiques de >25/mois à <1/mois.
* Abandons de recherche : Réduction des abandons de recherche de 48 % à <30 %, grâce à une recherche géolocalisée optimisée.

# **Conditions requises pour l’architecture**

Ce chapitre définit les exigences quantitatives et mesurables que l’architecture de la plateforme d’e-commerce Foosus doit satisfaire pour répondre aux objectifs commerciaux et aux KPI. Ces exigences englobent les aspects fonctionnels, non fonctionnels, réglementaires, et opérationnels, et sont alignées sur les contraintes et les directives. Elles complètent la vision qualitative du Document de Définition de l’Architecture et servent de base pour valider l’implémentation de l’architecture lors du déploiement pilote.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exigence** | **Métrique** | **Justification** | **Validation** |
| Recherche géolocalisée : La plateforme doit permettre aux consommateurs de trouver des fournisseurs locaux dans un rayon de 50 km, avec une précision de localisation <1 km. | Précision <1 km, abandons de recherche <30 % (Google Analytics). | Améliore l’expérience utilisateur, augmente les adhésions (+10 %/jour, 4 producteurs/mois). | Tests UX en phase C, API OpenStreetMap. |
| Processus de commande : Supporter la recherche de produits, l’ajout au panier, et l’envoi d’instructions de livraison par e-mail, avec une intégration future pour les paiements tiers. | Temps de traitement moyen <2 minutes (tests UX). | Simplifie l’expérience client, réduit les abandons. | Tests UX via Figma en phase C. |
| Support multi-utilisateurs : Fournir des interfaces spécifiques pour les fournisseurs (gestion des produits), consommateurs (recherche, panier), et back-office (gestion des commandes). | Satisfaction utilisateur >85 % (enquêtes UX post-ateliers). | Répond aux besoins variés des parties prenantes. | Ateliers UX bihebdomadaires, validation par les utilisateurs. |
| Performance : Les interfaces utilisateur (React) doivent se charger en <3 secondes sur une connexion 3G. | Temps de chargement <3 secondes (Prometheus/Grafana). | Assure l’accessibilité multi-appareils, réduit les abandons. | Tests de performance en phase D. |
| Disponibilité : La plateforme doit garantir une disponibilité de 99,9 % (temps d’arrêt <8,76 heures/an). | Disponibilité 99,9 % (Prometheus/Grafana). | Minimise les interruptions, atteint le KPI d’incidents P1 <1/mois. | Monitoring en phase G. |
| Évolutivité : L’architecture cloud (AWS/Azure) doit supporter un déploiement pilote et une capacité pour 1 million d’utilisateurs supplémentaires. | Capacité testée pour 1 million d’utilisateurs (tests de charge). | Soutient la croissance multi-régions. | Tests de charge en phase D. |
| Sécurité : L’architecture doit respecter les standards ISO 27001, avec une authentification sécurisée. | Aucune faille critique détectée (audit phase D). | Garantit la protection contre les cyberattaques. | Tests de sécurité, Keycloak. |
| Respect du budget : Le coût total de la conception de l’architecture ne doit pas dépasser 50 000 USD. | Budget suivi mensuellement (tableau de bord). | Aligne l’architecture sur les contraintes financières. | Validation par le chef de projet. |
| Cohérence technologique : Utiliser une pile open source (Node.js, PostgreSQL, React) avec un support communautaire actif. | Compatibilité des composants validée (phase D). | Réduit les coûts de maintenance, respecte les directives. | Analyse des solutions en phase E. |

# **Contrats de service business**

Les Contrats d’Architecture sont les accords communs entre les partenaires de développement et les sponsors sur les livrables, la qualité, et la correspondance à l’objectif d’une architecture. En ce qui concerne l’aspect business, les objectifs sont les suivants :

* Recherche géolocalisée : La plateforme doit permettre aux utilisateurs de trouver des fournisseurs dans un rayon de 50 km, avec un calculateur de distance précis à <1 km.
* Processus de commande : Support de la recherche de produits, ajout au panier, et envoi d’instructions de livraison par e-mail au fournisseur, avec une intégration future pour les paiements tiers.
* Support multi-utilisateurs : Interfaces spécifiques pour les fournisseurs (gestion des produits), les consommateurs (recherche, panier), et le back-office (gestion des commandes), validées par des tests UX.

## **Accords de niveau de service**

Les niveaux à atteindre afin de s’assurer d’avoir bien complété les objectifs sont les suivants :

* Disponibilité : La plateforme doit garantir une disponibilité de 99,9 % (temps d’arrêt <8,76 heures/an).
* Temps de réponse : Les API (par exemple, recherche géolocalisée) doivent répondre en <500 ms pour 95 % des requêtes, mesuré via Prometheus/Grafana.
* Évolutivité : Support d’un déploiement pilote dans une région, avec une capacité d’évolution pour 1 million d’utilisateurs supplémentaires sans dégradation.

# **Contrats de service application**

* Module de recherche : Doit intégrer une API de géolocalisation (par exemple, OpenStreetMap) pour afficher les fournisseurs locaux avec des informations secondaires (par exemple, indice glycémique).
* Module de panier : Doit permettre l’ajout de produits et la gestion des instructions de livraison, avec une interface utilisateur responsive (mobile/fixe).
* Module back-office : Doit fournir des rapports sur les commandes et les adhésions, accessibles via une interface sécurisée.
* Sécurité, communication par HTTPS, SSL, sécurité des informations bancaires

## **Objectifs de niveau de service**

* Temps de chargement : Interfaces utilisateur (React) avec un temps de chargement <3 secondes sur une connexion 3G.
* Taux d’abandon : Réduction des abandons de recherche à <30 % via des tests UX.
* Délai de déploiement : Mises à jour des microservices en <1 semaine, sans interruption de service.

## **Indicateurs de niveau de service**

* Temps de réponse API : Mesuré à <500 ms pour 95 % des requêtes via Prometheus/Grafana.
* Taux d’erreurs API : <1 % des requêtes échouent (mesuré via Grafana).
* Taux d’abandon de recherche : Mesuré via des outils d’analyse UX (par exemple, Google Analytics) à <30 %.
* Incidents P1 : Nombre d’incidents critiques <1/mois, suivi via Jira.

# **Lignes directrices pour l’implémentation**

* Solutions open source : Prioriser les technologies open source (Node.js pour le back-end, PostgreSQL pour les données, React pour le front-end, Keycloak pour l’authentification) avec un support communautaire actif.
* Microservices : Adopter une architecture de microservices pour permettre des déploiements continus sans interruption, alignés sur le KPI de délai de parution (<1 semaine).
* Cohérence technologique : Utiliser une pile technologique cohérente (Node.js, PostgreSQL, React) pour réduire les coûts de maintenance.
* Approche lean : Intégrer des itérations agiles (sprints de 2 semaines) pour tester les fonctionnalités (par exemple, recherche géolocalisée) et préserver l’autonomie des équipes.
* Conformité : ISO 27001 pour la sécurité.

# **Spécifications pour l’implémentation**

Technologies recommandées :

* Back-end : Node.js avec API REST (spécifications OpenAPI).
* Base de données : NoSQL avec indexation spatiale pour la géolocalisation.
* Front-end : React pour des interfaces responsive sur mobile/fixe.
* Authentification : Keycloak pour une gestion sécurisée des utilisateurs.
* Géolocalisation : OpenStreetMap ou Google Maps API pour la recherche géolocalisée.
* Infrastructure : Déploiement sur une plateforme cloud (AWS ou Azure) pour l’évolutivité et la résilience.
* Tests : Tests UX via Figma/Miro, tests de performance via Prometheus/Grafana, simulations de déploiement pour valider les microservices.

# **Standards pour l’implémentation**

Normes techniques :

* REST/OpenAPI : API conformes aux spécifications OpenAPI pour l’interopérabilité.
* NoSQL : Schémas de données non relationnels (MongoDB, ElasticSearch) pour les produits, clients, et localisations.
* SQL : pour gestion des utilisateurs et la gestion des commandes
* Microservices : Architecture modulaire pour des mises à jour fréquentes et transparentes.

Normes réglementaires :

* ISO 27001 : Standards de sécurité pour protéger la plateforme.

Normes UX :

* WCAG 2.1 : Accessibilité des interfaces pour mobile/fixe, optimisées pour faible bande passante.

# **Conditions requises pour l’interopérabilité**

* Intégration avec systèmes externes :
  + API REST pour une intégration future avec des prestataires de paiement tiers.
  + API de géolocalisation (par exemple, OpenStreetMap) pour la recherche basée sur la localisation.
* Standards d’interopérabilité : Utilisation d’OpenAPI pour documenter les API et garantir la compatibilité avec des systèmes externes.
* Coexistence des plateformes : Support de l’intégration entre la plateforme existante (en maintenance) et la nouvelle plateforme pendant le déploiement pilote.

# **Conditions requises pour le management du service IT**

* Surveillance : Utilisation de Prometheus/Grafana pour monitorer les performances (temps de réponse API, taux d’erreurs) et les incidents P1 (<1/mois).
* Gestion des exigences : Suivi continu des exigences, avec des mises à jour hebdomadaires des backlogs.
* Rapports : Rapports mensuels sur SharePoint pour documenter les progrès, les métriques.
* Gouvernance : Revues trimestrielles par le comité de gouvernance pour valider les livrables et résoudre les escalades.

# **Contraintes**

* Budget : Le coût total de la conception de l’architecture ne doit pas dépasser 50 000 USD, suivi via un tableau de bord mensuel.
* Délai : Tous les livrables doivent être finalisés d’ici 6 mois.
* Solutions open source : Priorité aux technologies open source avec un support communautaire actif pour minimiser les coûts.
* Cohérence technologique : Utilisation d’une pile technologique unifiée (Node.js, PostgreSQL, React) pour réduire la complexité de maintenance.
* Coexistence : La plateforme existante doit rester opérationnelle en mode maintenance pendant le déploiement pilote.

# **Hypothèses**

Les hypothèses suivantes sous-tendent la conception de l’architecture :

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Hypothèse** | **Impact** | **Propriétaire** |
| 1 | La plateforme existante peut être maintenue en mode maintenance pendant la conception de la nouvelle architecture, sans développement de nouvelles fonctionnalités. | Si fausse, des ressources supplémentaires seront nécessaires pour maintenir la plateforme existante, dépassant le budget de 50 000 USD. | CIO |
| 2 | La coexistence des plateformes existante et nouvelle est réalisable pendant le déploiement pilote, avec une migration progressive des utilisateurs. | Si fausse, des interruptions de service pourraient survenir, augmentant les incidents P1 et nuisant aux KPI (adhésions, abandons). | Architecte principal |
| 3 | Les solutions open source (par exemple, Node.js, PostgreSQL, React) offrent un support communautaire suffisant pour garantir la stabilité et l’évolutivité. | Si fausse, des coûts de maintenance imprévus pourraient compromettre le budget et la cohérence technologique. | Équipe d’architecture |
| 4 | La géolocalisation, modélisée tôt dans l’architecture (phase C), permettra des innovations futures (par exemple, recommandations basées sur la localisation). | Si fausse, la plateforme pourrait ne pas répondre aux attentes des utilisateurs pour la recherche géolocalisée, augmentant les abandons (48 % actuellement). | Équipe d’architecture |
| 5 | Les équipes techniques et produits resteront autonomes grâce à une approche lean et des outils agiles. | Si fausse, des retards dans les itérations UX ou les validations pourraient compromettre le délai de 6 mois. | CIO |