

# **Spécification des Conditions requises pour l'Architecture**

**Projet : Foosus, Conception d'une nouvelle architecture**



**Date : 14/05/2025**

**Document réaliser par : Eric Gigondan**

## **Table des matières**

<b>Information sur le document .....</b>	<b>3</b>
<b>Objet de ce document.....</b>	<b>4</b>
<b>Mesure du succès .....</b>	<b>5</b>
Indicateurs de réussites .....	5
<b>Conditions requises pour l'architecture.....</b>	<b>6</b>
<b>Contrats de services.....</b>	<b>8</b>
Accords de niveau de service SLA.....	8
Objectifs de niveau de service SLO .....	9
<b>Conditions requises pour l'interopérabilité.....</b>	<b>10</b>

## Information sur le document

<i>Nom du projet</i>	Foosus : Conception d'une nouvelle architecture
<i>Préparé par :</i>	Eric gigondan
<i>N° de version du document :</i>	1.0
<i>Titre :</i>	<i>Spécification des Conditions requises pour l'Architecture</i>
<i>Date de version du document :</i>	14/05/2025
<i>Types d'action :</i>	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion

## 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.

## Mesure du succès

### Indicateurs de réussites

Indicateur	Changement souhaité
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. miro

Les Accords de Niveau de Service (SLA), les Objectifs de Niveau de Service (SLO) et les Indicateurs de Niveau de Service (SLI) sont détaillés dans des sections spécifiques de ce document. Ils servent à établir le niveau de service attendu ainsi que les indicateurs permettant de le mesurer.

## Conditions requises pour l'architecture

L'architecture cible de la plateforme Foosus doit répondre à un ensemble d'exigences fonctionnelles, techniques, de sécurité et d'exploitabilité.

D'un point de vue fonctionnel, elle doit permettre la mise en relation entre producteurs et consommateurs en exploitant la géolocalisation, tout en proposant des parcours adaptés selon les profils utilisateurs. La plateforme devra être accessible sur mobile et desktop, avec une interface fluide, réactive et cohérente sur tous les supports. Elle doit intégrer les fonctionnalités essentielles liées à la commande, à la géolocalisation, au paiement, à la gestion de comptes et à la notification en temps réel, tout en permettant l'ajout rapide de nouvelles offres ou variantes sans perturber les parcours existants.

Sur le plan technique, l'architecture devra être entièrement modulaire et construite selon une approche microservices, avec des services autonomes communiquant via des interfaces ouvertes de type API REST ou GraphQL. Un Backend for Frontend sera mis en place afin d'adapter les flux de données aux spécificités de chaque interface cliente, ce qui permettra de maîtriser la latence et de garantir une bonne performance, même sur des connexions limitées. Les services devront être déployables de manière indépendante, avec une intégration complète dans un pipeline d'automatisation (CI/CD) permettant les tests, les mises à jour et les rollbacks sans interruption.

L'architecture cible reposera sur une conteneurisation systématique des services applicatifs à l'aide de Docker, afin de garantir leur portabilité, leur isolation et leur cohérence entre les différents environnements (développement, test, production). L'orchestration de ces conteneurs sera assurée via Kubernetes, déployé dans un environnement cloud fourni par un partenaire tel qu'AWS (EKS), Google Cloud (GKE) ou Azure (AKS), selon les critères de coûts, de localisation des données et de disponibilité. Ce choix permettra de gérer dynamiquement la montée en charge, l'équilibrage de charge, la résilience et le redémarrage automatique des services défaillants.

La supervision de l'ensemble des composants sera confiée à un ensemble d'outils intégrés incluant Datadog pour la collecte et l'analyse des métriques, Sentry pour le suivi des erreurs applicatives, et Grafana pour la visualisation des indicateurs techniques clés. Ces outils permettront un suivi en temps réel de l'état de santé des services, avec des alertes configurées selon les niveaux de criticité, et une historisation complète pour l'analyse post-incident. Ce socle technique assurera un haut niveau de fiabilité, de maintenabilité et de réactivité dans l'exploitation de la plateforme.

Les exigences de sécurité sont également centrales. Toutes les données sensibles, qu'il s'agisse d'informations personnelles ou de données de facturation, devront être chiffrées à la fois en transit et au repos, en utilisant des standards reconnus comme TLS pour les communications et AES pour le stockage. L'ensemble de la plateforme devra être conforme au RGPD, avec des mécanismes clairs de gestion du consentement, d'anonymisation des données et de suppression à la demande. Les accès aux ressources critiques devront être sécurisés par une authentification forte, et chaque action sensible devra être tracée dans un système de journalisation centralisé.

Enfin, l'architecture devra être pensée pour faciliter l'exploitation, le monitoring et la maintenance. Les services devront exposer des métriques techniques permettant leur supervision en temps réel, avec des alertes déclenchées automatiquement en cas d'anomalie. La traçabilité des déploiements, la centralisation des logs et la documentation à jour de l'ensemble des interfaces seront des éléments indispensables pour assurer la fiabilité et l'évolutivité du système dans le temps.

# Contrats de services

## Accords de niveau de service SLA

Pour garantir un service fiable et performant, la plateforme Foosus s'engage sur plusieurs niveaux de service mesurables. Ces engagements portent sur la disponibilité, la performance, la gestion des incidents et la stabilité des déploiements. Ils servent de référence pour évaluer la qualité opérationnelle de l'architecture une fois mise en production.

Aspect de service	Engagement	Description
Disponibilité globale	$\geq 99,5$ % par mois	La plateforme doit rester accessible sans interruption majeure pendant 99,5 % du temps mensuel. Les interruptions planifiées doivent être annoncées et limitées.
Performance des APIs	$\leq 200$ ms de temps de réponse moyen	Les services exposés via API doivent répondre en moins de 200 millisecondes dans 95 % des cas, afin d'assurer une navigation fluide.
Temps de chargement	$\leq 2$ secondes sur mobile	Les pages critiques doivent s'afficher en moins de 2 secondes sur un appareil mobile en réseau 3G ou 4G.
Support des incidents	Réaction $\leq 2$ heures	Un incident critique doit être pris en charge dans les 2 heures suivant la détection ou la remontée.
Résolution des incidents	Résolution $\leq 4$ heures	Les incidents critiques (niveau P1) doivent être résolus dans un délai maximal de 4 heures.
Déploiement continu	$\geq 98$ % de taux de réussite	Les déploiements via la chaîne CI/CD doivent réussir sans erreur dans au moins 98 % des cas.
Intégrité des données	100 % des données chiffrées	Toutes les données personnelles et de facturation doivent être chiffrées en transit et au repos avec des algorithmes reconnus (ex. TLS, AES-256).
Suivi des erreurs	Journalisation centralisée continue	Tous les incidents techniques doivent être automatiquement enregistrés dans un système de log consultable et historisé.

miro



## Objectifs de niveau de service SLO

Les objectifs de niveau de service (SLO) sont définis à partir d'indicateurs mesurables (SLI) liés à la performance, la disponibilité, la fiabilité et la sécurité du système. Ces indicateurs sont suivis en continu à l'aide des outils de monitoring mis en place (Datadog, Sentry, CI/CD...) et servent de base pour évaluer la stabilité et la qualité de l'architecture cible.

Aspect de service	SLI (Indicateur mesuré)	SLO (Objectif visé)
Disponibilité	Pourcentage de disponibilité mensuelle (Datadog)	≥ 99,9 %
Latence API	Temps de réponse moyen des requêtes (Grafana, Datadog APM)	≤ 200 ms pour 95 % des requêtes
Temps de chargement	Temps d'affichage d'une page sur mobile	≤ 2 secondes pour 90 % des sessions
Taux d'erreur	Pourcentage de requêtes échouées (5xx, timeout) (Sentry, Datadog)	< 1 % sur la période de mesure
Temps de réaction	Délai entre détection et prise en charge incident P1 (PagerDuty)	≤ 2 heures
Temps de résolution	Durée entre l'ouverture et la résolution d'un incident critique	≤ 4 heures
Fiabilité des déploiements	Pourcentage de déploiements réussis (GitLab CI/CD, Github CI/CD)	≥ 98 % des déploiements automatisés
Intégrité des données	Données personnelles chiffrées à 100 %	100 % des données sensibles chiffrées au repos et en transit
Rétention utilisateurs	Taux d'utilisateurs actifs après 30 jours (Mixpanel )	≥ 60 % des nouveaux utilisateurs
Taux de disponibilité du monitoring	Disponibilité ( Datadog/Sentry)	≥ 99,9 %

miro

## Conditions requises pour l'interopérabilité

Les exigences d'interopérabilité établissent les critères indispensables pour assurer une collaboration fluide et efficace entre les différents composants de l'architecture de Foosus, ainsi qu'avec les systèmes externes.

Cela englobe les mécanismes de communication, les protocoles utilisés, ainsi que les formats de données pris en charge.

Couche d'interopérabilité	Standards / Protocoles utilisés	Objectif dans le projet
Communication réseau	HTTP / HTTPS	Assurer des échanges sécurisés entre services internes et externes
Interfaces d'API	REST, GraphQL, OpenAPI (Swagger)	Permettre l'accès aux fonctionnalités via des interfaces normalisées et documentées
Format d'échange de données	JSON (principal), XML (si nécessaire)	Garantir la lisibilité et la compatibilité des données entre systèmes hétérogènes
Authentification inter-systèmes	OAuth2, JWT	Contrôler l'accès aux API et sécuriser les échanges entre services ou partenaires
Gestion des erreurs	Codes HTTP standardisés, messages d'erreur structurés	Faciliter le diagnostic et l'intégration côté client ou partenaires
Supervision / observabilité	Datadog	Intégrer les services au système de monitoring central pour un suivi unifié
Documentation technique	Portail d'API, OpenAPI spec versionnée	Rendre les interfaces accessibles, compréhensibles et maintenables
CI/CD et intégration externe	GitHub Actions, Docker, Ansible, Terraform, Kubernetes, Sonarqube	Permettre l'automatisation des tests, déploiements et intégrations tiers
Base de données / stockage	NoSQL, ORM	Assurer la compatibilité entre les services applicatifs et les systèmes de stockage
Services externes partenaires	API fournisseurs (ex : Stripe, Google Maps)	Intégrer les services tiers sans dépendance forte ou verrou propriétaire