

FOOSUS

# Contrat d'architecture des utilisateurs business

*Projet : XXXX Client : YYYY*



Louis ZEPHIR  
14/11/2024

## Table des matières

<b>Objet de ce document</b> .....	2
<b>Introduction et contexte</b> .....	3
<b>Comparaison des architectures</b> .....	3
Architecture actuelle (En couche) .....	4
Nouvelle architecture (Microservices) : .....	4
Alignement stratégique .....	5
<b>La Nature de l'accord</b> .....	7
<b>Gestions des risques prioritaires</b> .....	8
<b>Objectifs et périmètre</b> .....	9
<b>Objectifs</b> .....	9
<i>Objectif Business 1 : Augmenter l'engagement des utilisateurs grâce à une expérience fluide et évolutive</i> .....	9
<i>Objectif Business 2 : Réduire la dette technique pour accélérer l'innovation et maintenir la flexibilité</i> .....	10
<b>Périmètre</b> .....	10
<b>Parties prenantes, préoccupations et visions</b> .....	11
<b>Conditions requises pour la conformité</b> .....	13
<b>Personnes adoptant l'architecture</b> .....	15
<b>Fenêtre temporelle</b> .....	16
Plan de Migration .....	16
1. Stratégie de transition .....	16
2. Phases de migration .....	16
3. Gestion de la coexistence .....	17
4. Suivi et validation .....	17
Phase temporelle .....	18
<b>Métriques Business de l'architecture</b> .....	18
<b>Accords de service pour l'architecture (y compris accord du niveau de service [ANS] ...</b>	20
<b>Processus CI/CD</b> .....	21
Objectifs du processus CI/CD.....	21
Pipeline CI/CD proposé .....	22
Règles de déploiement.....	22
Suivi et monitoring.....	23
Avantages du processus CI/CD.....	23
<b>Personnes approuvant ce plan</b> .....	24

# Objet de ce document

---

*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. L'implémentation réussie de ces accords sera livrée grâce à une gouvernance de l'architecture efficace (voir TOGAF Partie VII, Gouvernance de l'architecture). En implémentant une approche dirigée du management de contrats, les éléments suivants seront garantis :*

- *Un système de contrôle continu pour vérifier l'intégrité, les changements, les prises de décisions, et l'audit de toutes les activités relatives à l'architecture au sein de l'organisation.*
- *L'adhésion aux principes, standards et conditions requises des architectures existantes ou en développement*
- *L'identification des risques dans tous les aspects du développement et de l'implémentation des/de l'architecture(s), y compris le développement interne en fonction des standards acceptés, des politiques, des technologies et des produits, de même que les aspects opérationnels des architectures de façon à ce que l'organisation puisse poursuivre son business au sein d'un environnement résistant.*
- *Un ensemble de process et de pratiques qui garantissent la transparence, la responsabilité et la discipline au regard du développement et de l'utilisation de tous les artefacts architecturaux*
- *Un accord formel sur l'organe de gouvernance responsable du contrat, son degré d'autorité, et le périmètre de l'architecture sous la gouvernance de cet organe*

*Ceci est une déclaration d'intention de se conformer à l'architecture d'entreprise, délivrée par les utilisateurs business entreprise. Lorsque l'architecture d'entreprise aura été implémentée (à la fin de la Phase F), un Contrat d'Architecture sera normalement établi entre la fonction architecture (ou la fonction de gouvernance IT, englobant la fonction architecture) et les utilisateurs business qui par la suite construiront et déploieront des applications système dans l'environnement créé par l'architecture.*

# Introduction et contexte

---

Foosus, une start-up spécialisée dans l'alimentation durable, vise à transformer la façon dont les consommateurs interagissent avec les producteurs et artisans locaux.

En mettant en avant une approche géociblée et en favorisant une alimentation locale, Foosus souhaite répondre à des défis stratégiques tels que l'amélioration de l'expérience utilisateur, l'augmentation de l'engagement des parties prenantes, et l'évolution rapide de sa plateforme.

Depuis sa création, Foosus a fait face à une accumulation de dette technique et à une incohérence dans ses choix d'architecture, ralentissant son innovation et sa capacité à s'adapter aux besoins des utilisateurs business et finaux.

Ce **Contrat d'architecture des utilisateurs business** établit un cadre clair pour répondre aux attentes des utilisateurs business, en alignant les décisions architecturales avec les objectifs stratégiques de l'entreprise.

L'objectif principal de ce contrat est de garantir que l'architecture développée :

- Soutient les besoins fonctionnels des utilisateurs business, comme la mise en relation rapide et fiable des consommateurs avec les producteurs locaux.
- Assure une évolutivité pour accompagner la croissance rapide de Foosus.
- Réduit la dette technique tout en maintenant une flexibilité pour l'innovation future.
- Fournit une base stable et conforme aux exigences réglementaires, notamment en matière de protection des données.

Ce document vise également à établir des engagements clairs entre les équipes techniques et les utilisateurs business afin de garantir une collaboration efficace et un alignement constant sur les priorités business.

Il servira de référence pour évaluer le succès de l'architecture et guider les adaptations nécessaires pour répondre aux attentes des utilisateurs dans un environnement en constante évolution.

## ***Comparaison des architectures***

L'évolution de l'architecture actuelle vers une architecture microservices marque une transformation stratégique pour Foosus.

Cette transition vise à résoudre les limites actuelles tout en permettant une scalabilité accrue et

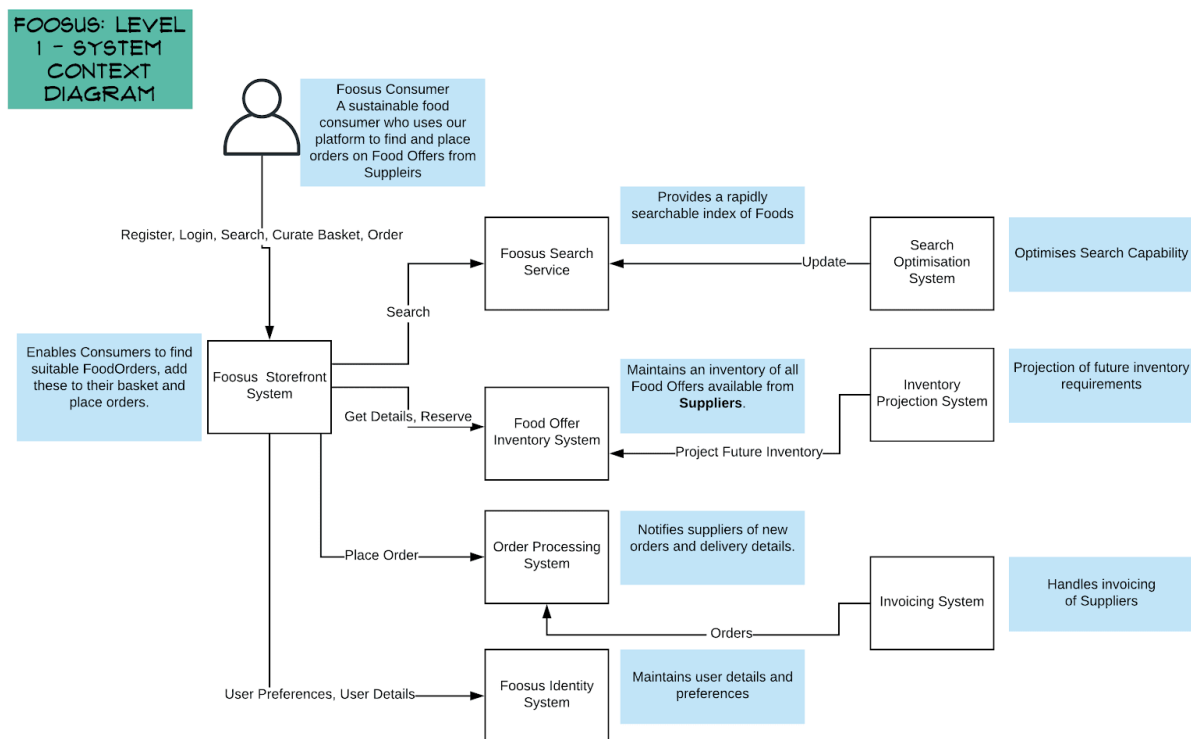
une flexibilité adaptée aux besoins de l'entreprise. Voici une vue d'ensemble des deux architectures :

## Architecture actuelle (En couche)

Une architecture en couche, où toutes les fonctionnalités sont regroupées dans un seul système.

### Inconvénients :

- Difficultés à maintenir et à faire évoluer rapidement le système.
- Un impact global en cas de panne sur un composant.
- Dépendances fortes entre les différentes parties du système, limitant la modularité.

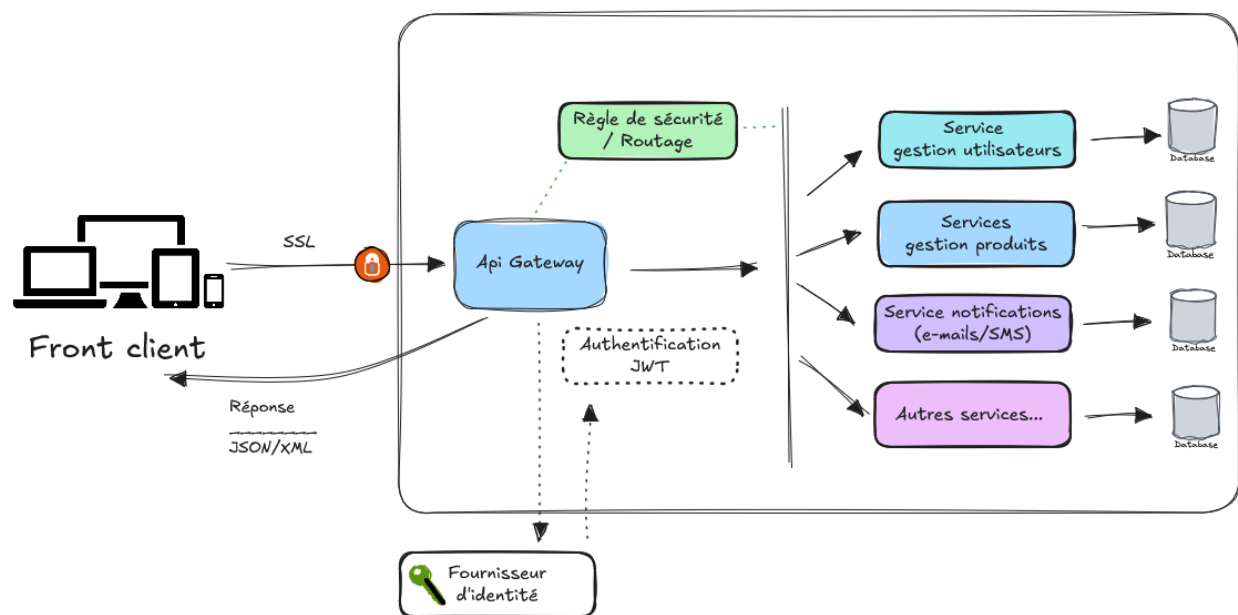


## Nouvelle architecture (Microservices) :

Une architecture décomposée en services indépendants, chacun ayant une responsabilité spécifique.

### Avantages clés :

- L'évolutivité car chaque service peut être modifié, mis à jour ou redimensionné indépendamment des autres.
- La fiabilité car en cas de panne dans un service, le service en question n'affectera pas l'ensemble du système.
- La flexibilité qui facilite l'intégration de nouvelles technologies et la mise en œuvre rapide de nouvelles fonctionnalités.



## Alignement stratégique

Foosus adopte une approche stratégique basée sur les principes **LEAN**, parfaitement alignée avec sa mission : **réduire les intermédiaires**, rapprocher les consommateurs des producteurs locaux, et optimiser les processus pour minimiser les pertes dans la chaîne d'approvisionnement alimentaire.

L'architecture cible soutient cette philosophie en mettant en place des outils et des pratiques qui favorisent : la réduction du gaspillage, l'optimisation des processus, la création de valeur directe et l'amélioration continue.

Principe LEAN	Description	Impact pour Foosus
<b>Réduction du gaspillage</b>	<ul style="list-style-type: none"> <li>- Interactions entre consommateurs et producteurs pour limiter les processus manuels et réduire les erreurs.</li> <li>- Outils analytiques pour optimiser la gestion des stocks et éviter les excédents ou les ruptures.</li> </ul>	<ul style="list-style-type: none"> <li>- Moins de pertes alimentaires.</li> <li>- Réduction des erreurs de commande et des délais liés aux processus manuels.</li> </ul>
<b>Optimisation des processus</b>	<ul style="list-style-type: none"> <li>- Architecture microservices pour un déploiement rapide et modulable des améliorations.</li> <li>- API Gateway pour rationaliser les flux d'information et garantir une communication efficace.</li> </ul>	<ul style="list-style-type: none"> <li>- Meilleur délai de mise sur le marché pour les nouvelles fonctionnalités.</li> <li>- Simplification des flux opérationnels.</li> </ul>
<b>Création de valeur directe</b>	<ul style="list-style-type: none"> <li>- Accès simplifié aux offres locales en temps réel via une interface fluide et</li> </ul>	<ul style="list-style-type: none"> <li>- Amélioration de l'expérience utilisateur.</li> </ul>

	personnalisée. - Suppression des intermédiaires pour réduire les coûts pour les consommateurs et améliorer les marges des producteurs.	- Augmentation des bénéfices pour les producteurs locaux.
<b>Amélioration continue</b>	- Intégration de pipelines CI/CD pour des itérations rapides et ajustements fréquents. - Suivi continu des performances grâce aux métriques mises en place.	- Services améliorés en fonction des retours utilisateurs. - Performances mesurées et optimisées en permanence.

# La Nature de l'accord

---

Cet accord entre l'équipe d'architecture de Foosus et les utilisateurs business définit les engagements mutuels et les mécanismes de collaboration nécessaires pour aligner les décisions architecturales avec les objectifs stratégiques de l'entreprise.

Il vise à garantir que l'architecture en cours de développement répond efficacement aux besoins des utilisateurs business tout en soutenant les priorités techniques, réglementaires et opérationnelles de Foosus.

## Portée de l'accord

### 1. Alignement stratégique

Cet accord garantit que l'architecture soutient les objectifs business tels que l'augmentation de l'engagement des utilisateurs, l'élargissement de la base client, et l'optimisation de la mise en relation entre consommateurs et producteurs locaux.

### 2. Collaboration continue

Il établit un cadre pour une communication et une collaboration régulière entre les équipes techniques et les utilisateurs business, favorisant une compréhension mutuelle des priorités et des contraintes.

### 3. Adaptabilité

L'accord permet une flexibilité pour répondre aux changements dans les besoins business tout en limitant l'impact sur les délais, la qualité, et la scalabilité de l'architecture.

## Engagements clés

### 1. De la part de l'équipe d'architecture :

- Fournir une architecture évolutive et performante qui soutient les processus business critiques.
- Assurer la conformité aux standards réglementaires (ex. RGPD) et techniques.
- Communiquer régulièrement l'avancement des travaux et les impacts des décisions prises.
- Offrir un soutien technique pour l'intégration des solutions dans les processus existants.

### 2. De la part des utilisateurs business :

- Fournir des exigences claires et précises pour guider les choix architecturaux.



- Participer activement aux revues périodiques pour valider les livrables et ajuster les priorités si nécessaire.
- Respecter les délais de validation des livrables pour éviter des retards dans le projet.
- S'engager à adopter les solutions proposées, à condition qu'elles répondent aux critères validés.

### Objectif final de l'accord

Cet accord constitue une base commune pour s'assurer que l'architecture livrée par Foosus répond de manière mesurable aux besoins des utilisateurs business tout en respectant les contraintes techniques et organisationnelles.

Il définit également un mécanisme clair pour gérer les changements et ajuster les priorités en fonction des besoins émergents.

En établissant une compréhension commune et des engagements partagés, cet accord contribue à minimiser les risques, à maximiser l'impact business, et à garantir la réussite globale de l'initiative d'architecture de Foosus.

## Gestions des risques prioritaires

Risque	Gravité	Probabilité	Criticité	Facteur de réduction	Propriétaire
<b>Retards dans la validation des livrables</b>	5	3	15	- Réunions hebdomadaires de suivi avec les parties prenantes. - Validation progressive par étape.	Responsable de projet
<b>Manque de compétences techniques</b>	4	4	16	- Organisation de sessions de formation sur les microservices et pipelines CI/CD. - Mentorat interne.	Directeur technique (CTO)
<b>Dépendance à des services tiers critiques</b>	4	3	12	- Mise en place de solutions alternatives en cas de panne (backup system). - Analyse des SLA tiers.	Responsable des opérations
<b>Difficultés de communication entre équipes</b>	3	5	15	- Adoption d'outils collaboratifs (Slack, Trello, etc.). - Mise en place d'un facilitateur technique.	Responsable de l'équipe

## Interprétation du tableau

### **Gravité (1-5) :**

- 1 = faible impact
- 5 = impact très important.

### **Probabilité (1-5) :**

- 1 = risque peu probable.
- 5 = risque très probable.

### **Criticité :**

Plus le score de criticité est élevé, plus le risque doit être priorisé et surveillé.

Méthode de calcul : Gravité **X** Probabilité = **Criticité**

# Objectifs et périmètre

---

## Objectifs

---

Les objectifs business de ce travail d'architecture visent à répondre aux priorités stratégiques de Foosus, en assurant une base technique solide pour soutenir ses ambitions de croissance et d'innovation.

Ce travail d'architecture est conçu pour résoudre les problèmes actuels liés à la dette technique, améliorer la cohérence des systèmes, et garantir une expérience utilisateur optimisée.

Ces objectifs s'articulent autour de deux priorités principales, détaillées ci-dessous :

### ***Objectif Business 1 : Augmenter l'engagement des utilisateurs grâce à une expérience fluide et évolutive***

Cet objectif consiste à offrir une plateforme performante et intuitive qui facilite l'interaction entre les consommateurs et les producteurs locaux. Les actions incluent :

- **Amélioration des performances**

Réduire le temps de réponse moyen des services pour assurer une expérience utilisateur rapide, même en cas de forte charge.

- **Scalabilité de l'architecture**

Permettre à la plateforme de gérer une augmentation significative du nombre d'utilisateurs sans compromettre la qualité du service.

- **Géolocalisation optimisée**

Intégrer des services de géolocalisation fiables pour connecter efficacement les

utilisateurs aux producteurs proches, en maximisant la pertinence des résultats.

Une architecture évolutive et performante permettra d'améliorer la satisfaction utilisateur, d'augmenter le taux de rétention et d'attirer de nouveaux utilisateurs, contribuant directement à la croissance de Foosus.

## ***Objectif Business 2 : Réduire la dette technique pour accélérer l'innovation et maintenir la flexibilité***

Cet objectif vise à refondre l'architecture existante pour minimiser la dette technique, standardiser les pratiques de développement et garantir une base modulaire pour les futures fonctionnalités. Les actions incluent :

- **Standardisation des microservices**

Adopter des pratiques et des standards de développement qui facilitent la maintenance, la réutilisation et la modularité.

- **Réduction des dépendances historiques**

Identifier et migrer les composants obsolètes pour éviter qu'ils ne freinent l'ajout de nouvelles fonctionnalités.

- **Déploiement continu**

Mettre en place des pipelines CI/CD robustes pour accélérer les cycles de livraison, tout en maintenant une haute qualité de service.

En réduisant la dette technique et en adoptant une architecture modulaire, Foosus pourra innover rapidement et rester compétitif dans un secteur en constante évolution, tout en garantissant une stabilité opérationnelle.

## **Périmètre**

Catégorie	Description
Composants principaux	- Conception et développement des microservices prioritaires : gestion des utilisateurs, géolocalisation, gestion des producteurs et offres. - Mise en place des outils de support (CI/CD, monitoring, alerting).
Intégrations externes	- Intégration avec les services tiers pour la géolocalisation, paiements et notifications.

	- Interopérabilité avec les systèmes existants pour une transition progressive.
<b>Performance et scalabilité</b>	- Architecture capable de gérer les pics de trafic. - Optimisation des services pour les environnements à faible bande passante.
<b>Sécurité et conformité</b>	- Protection des données utilisateur en conformité avec les réglementations (ex. RGPD). - Mise en œuvre des protocoles de sécurité (authentification, chiffrement).
<b>Déploiement et gestion</b>	- Gestion du cycle de vie des microservices : déploiement, maintenance, mises à jour. - Transition vers les équipes de maintenance après validation et déploiement.
<b>Limites du périmètre</b>	- Exclusion des fonctionnalités secondaires ou extensions non critiques, qui seront traitées dans des phases ultérieures.

Ce tableau synthétise les points clés du périmètre de manière concise et indiquant les détails nécessaires pour comprendre les priorités et limites de cette phase du travail d'architecture.

## Parties prenantes, préoccupations et visions

*Le tableau suivant montre les parties prenantes qui utiliseront ce document, leurs préoccupations, et la façon dont le travail d'architecture répondra à ces préoccupations par l'expression de plusieurs visions, ou perspectives.*

Partie prenante	Préoccupation	Vision
<b>CEO (Ash Callum)</b>	Alignement de l'architecture avec les objectifs stratégiques de croissance et d'engagement des utilisateurs.	Une architecture évolutive qui soutient la montée en charge et renforce la compétitivité de Foosus.
<b>CIO (Natasha Jarson)</b>	Réduction de la dette technique et standardisation des pratiques pour améliorer la maintenabilité.	Une architecture modulaire basée sur les microservices, permettant une maintenance simplifiée et une innovation rapide.
<b>CPO (Daniel Anthony)</b>	Amélioration de l'expérience utilisateur et livraison rapide de nouvelles fonctionnalités.	Une architecture performante, supportant des déploiements fréquents et une UX fluide pour les utilisateurs finaux.
<b>Responsable Ingénierie (Pete Parker)</b>	Gestion efficace des ressources techniques et minimisation des incidents de production.	Une infrastructure robuste, avec des pipelines CI/CD et des tests rigoureux pour garantir la disponibilité et la fiabilité.
<b>Équipe Produit</b>	Accélération du temps de mise	Une architecture agile, facilitant

	sur le marché des nouvelles fonctionnalités.	le développement et le déploiement de fonctionnalités prioritaires.
<b>Équipe de Développement</b>	Adopter de nouveaux standards tout en minimisant les impacts sur la productivité.	Un cadre architectural clair avec des standards bien définis et une documentation complète pour guider les équipes.
<b>Clients finaux (utilisateurs de la plateforme)</b>	Performances de la plateforme, disponibilité constante et pertinence des résultats de géolocalisation.	Une plateforme rapide, fiable, et connectant efficacement les utilisateurs aux producteurs locaux.

Ci-dessous un tableau concluant un **accord sur les aspects de l'architecture** qui concernent chaque partie prenante et listant les **visions nécessaires pour répondre à leurs préoccupations**.

Partie prenante	Aspects de l'architecture concernés	Visions nécessaires
<b>CEO (Ash Callum)</b>	<ul style="list-style-type: none"> <li>- Alignement stratégique avec les objectifs business.</li> <li>- Scalabilité de la plateforme pour soutenir la croissance.</li> </ul>	Vision d'architecture stratégique montrant comment l'architecture soutient la montée en charge et la compétitivité.
<b>CIO (Natasha Jarson)</b>	<ul style="list-style-type: none"> <li>- Réduction de la dette technique.</li> <li>- Standardisation des pratiques de développement.</li> </ul>	Vision technique détaillant les composants modulaires, les standards, et les processus pour minimiser la dette technique.
<b>CPO (Daniel Anthony)</b>	<ul style="list-style-type: none"> <li>- Optimisation de l'expérience utilisateur.</li> <li>- Flexibilité pour intégrer rapidement de nouvelles fonctionnalités.</li> </ul>	Vision produit illustrant comment l'architecture améliore l'UX et accélère la mise sur le marché des nouvelles fonctionnalités.
<b>Responsable Ingénierie (Pete Parker)</b>	<ul style="list-style-type: none"> <li>- Disponibilité et fiabilité de l'infrastructure.</li> <li>- Mise en place de pipelines CI/CD pour les déploiements continus.</li> </ul>	Vision technique opérationnelle décrivant les mécanismes de déploiement continu et de surveillance des performances.
<b>Équipe Produit</b>	<ul style="list-style-type: none"> <li>- Simplification de l'ajout de nouvelles fonctionnalités.</li> <li>- Amélioration de la collaboration entre les équipes produit et développement.</li> </ul>	Vision fonctionnelle montrant l'intégration des workflows produit-développement pour une meilleure agilité.
<b>Équipe de Développement</b>	<ul style="list-style-type: none"> <li>- Adoption des nouveaux standards.</li> <li>- Documentation et outils facilitant le développement de micro services.</li> </ul>	Vision de développement clarifiant les standards, la documentation et les outils nécessaires pour garantir une transition fluide.
<b>Clients finaux</b>	<ul style="list-style-type: none"> <li>- Performance et disponibilité de la plateforme.</li> <li>- Pertinence et précision des services de géolocalisation.</li> </ul>	Vision utilisateur démontrant comment l'architecture soutient des performances optimales et connecte efficacement les utilisateurs.

## Synthèse

Pour chaque partie prenante, cet accord garantit que l'architecture sera conçue pour répondre à leurs préoccupations spécifiques. Les visions proposées (stratégique, technique, produit, opérationnelle, fonctionnelle, développement et utilisateur) permettront de :

1. Clarifier les impacts des décisions architecturales sur leurs objectifs.
2. Créer des représentations spécifiques pour expliquer comment l'architecture soutient leurs besoins.
3. Aligner les attentes et assurer une validation claire des résultats à chaque étape du projet.

Cet accord servira de base pour maintenir la transparence et la collaboration tout au long du travail d'architecture.

# Conditions requises pour la conformité

Pour garantir que l'architecture développée pour Foosus respecte les objectifs business et techniques ainsi que les standards établis, un cadre clair de conformité a été défini.

Ce cadre s'assure que les livrables architecturaux répondent aux exigences réglementaires, techniques, et fonctionnelles tout au long du projet.

## Exigences réglementaires

- **Conformité au RGPD (Règlement Général sur la Protection des Données)**

Toutes les données personnelles collectées et traitées par la plateforme doivent être sécurisées et protégées conformément aux normes européennes. Cela inclut le chiffrement des données personnelles, la gestion des droits des utilisateurs (ex : droit à l'effacement), et la transparence sur les traitements des données.

- **Respect des normes de sécurité**

Les mécanismes d'authentification, de contrôle d'accès, et de protection contre les menaces doivent être conformes aux standards de sécurité de l'industrie, tels que OWASP (Open Web Application Security Project).

## Exigences techniques

- **Performance et scalabilité**

L'architecture doit démontrer sa capacité à répondre aux objectifs de performance (temps de réponse < 200 ms) et à gérer des montées en charge importantes (10 000 requêtes/minute).

- **Interopérabilité et modularité**

Tous les composants de l'architecture doivent être conçus pour fonctionner ensemble de manière cohérente et flexible, facilitant ainsi les futures extensions ou remplacements de services.

<b>Exigences fonctionnelles</b>
---------------------------------

- **Fiabilité des services de géolocalisation**

Les services doivent fournir des résultats pertinents avec une précision suffisante et une disponibilité garantie à 99 %.

- **Expérience utilisateur optimisée**

L'architecture doit soutenir une interface fluide, accessible sur différents types de dispositifs et performante, même en cas de bande passante limitée.

<b>Mécanismes de vérification de conformité</b>
---

1. **Audits périodiques**

Des audits de conformité seront réalisés à intervalles réguliers (tous les 3 mois) pour évaluer les performances, la sécurité, et la conformité des livrables avec les exigences définies.

2. **Tests techniques**

Chaque composant sera soumis à des tests rigoureux (tests de charge, de sécurité, et d'intégration) pour valider sa conformité avant sa mise en production.

3. **Revue par les parties prenantes**

Les livrables seront présentés et revus par le comité d'architecture, incluant les représentants business, produit, et technique, pour garantir leur alignement avec les objectifs stratégiques.

4. **Documentation**

Chaque livrable sera accompagné d'une documentation claire et détaillée, assurant une traçabilité complète des décisions et des validations.

<b>Gestion des écarts</b>
---------------------------

En cas de non-conformité identifiée lors des audits ou des tests, des actions correctives seront planifiées et exécutées dans des délais définis. Les écarts majeurs devront être

approuvés par le comité d'architecture avant toute mise en production

## Personnes adoptant l'architecture

Personnes adoptant	Rôle	Responsabilités
Équipe Produit	Exploiter l'architecture pour concevoir et prioriser les fonctionnalités alignées avec les besoins utilisateurs.	<ul style="list-style-type: none"><li>- Traduire les exigences business en fonctionnalités techniques.</li><li>- Valider que les services livrés répondent aux besoins identifiés.</li></ul>
Équipe de Développement	Implémenter et maintenir les microservices et autres composants de l'architecture.	<ul style="list-style-type: none"><li>- Suivre les standards de l'architecture.</li><li>- Intégrer de nouvelles fonctionnalités.</li><li>- Documenter le code pour assurer la continuité technique.</li></ul>
Équipe d'Ingénierie et DevOps	Superviser le déploiement, la surveillance, et la maintenance des composants architecturaux.	<ul style="list-style-type: none"><li>- Mettre en place des pipelines CI/CD pour des déploiements sécurisés.</li><li>- Surveiller les performances en production.</li><li>- Optimiser l'infrastructure cloud.</li></ul>
Direction Business (CEO, CPO)	Utiliser l'architecture pour atteindre les objectifs stratégiques de Foosus.	<ul style="list-style-type: none"><li>- S'assurer que les livrables architecturaux sont alignés avec les priorités stratégiques.</li><li>- Valider les décisions clés d'adaptation.</li></ul>
Clients finaux	Interagir avec la plateforme et bénéficier des fonctionnalités supportées par l'architecture.	<ul style="list-style-type: none"><li>- Fournir des retours sur l'expérience utilisateur.</li><li>- Adopter les nouvelles fonctionnalités proposées.</li></ul>

### Engagement à l'adoption

Pour garantir l'adoption réussie :

- Des formations et supports seront fournis pour accompagner les utilisateurs internes.
- Les retours des parties prenantes et utilisateurs finaux seront intégrés dans les itérations pour ajuster l'architecture si nécessaire.

Ce tableau synthétise les rôles et responsabilités de chaque personne ou groupe impliqué dans l'adoption de l'architecture, facilitant leur implication et leur collaboration.



# Fenêtre temporelle

---

## Plan de Migration

Le plan de migration décrit les étapes nécessaires pour passer de l'architecture actuelle à l'architecture cible basée sur les microservices.

Il prend en compte les objectifs de continuité opérationnelle, la minimisation des risques, et une expérience utilisateur fluide

### 1. Stratégie de transition

La migration sera effectuée de manière incrémentale pour éviter toute interruption des services existants.

Les deux systèmes (ancien et nouveau) coexisteront temporairement durant la phase de transition.

- **Maintien en Condition Opérationnelle (MCO)**

L'ancien système sera maintenu uniquement pour corriger les bugs critiques et assurer une disponibilité minimale.

- **Pas de développement sur l'ancien système**

Aucune nouvelle fonctionnalité ne sera développée sur l'ancienne architecture.

- **Notifications aux utilisateurs**

Les utilisateurs seront régulièrement informés de la disponibilité de la nouvelle interface et des nouvelles fonctionnalités via des notifications dans l'interface et/ou par email.

### 2. Phases de migration

Phase	Activités principales	Livrables	Estimation
<b>Préparation</b>	- Analyse détaillée de l'architecture actuelle. - Identification des dépendances critiques. - Définition des microservices prioritaires.	- Rapport d'analyse. - Feuille de route technique.	2 semaines.

<b>Infrastructure</b>	<ul style="list-style-type: none"> <li>- Mise en place des environnements (Kubernetes/Docker).</li> <li>- Déploiement de l'API Gateway.</li> <li>- Configuration des bases de données indépendantes.</li> </ul>	<ul style="list-style-type: none"> <li>- API Gateway fonctionnelle.</li> <li>- Bases configurées.</li> </ul>	3 semaines.
<b>Migration des services</b>	<ul style="list-style-type: none"> <li>- Migration des services critiques (authentification, gestion des commandes).</li> <li>- Tests d'intégration sur des environnements de staging.</li> </ul>	<ul style="list-style-type: none"> <li>- Services migrés.</li> <li>- Rapport de tests.</li> </ul>	4 semaines (progressif).
<b>Déploiement progressif</b>	<ul style="list-style-type: none"> <li>- Déploiement des microservices sur l'environnement de production.</li> <li>- Communication aux utilisateurs.</li> <li>- Formation des équipes internes.</li> </ul>	<ul style="list-style-type: none"> <li>- Microservices actifs.</li> <li>- Documentation mise à jour.</li> </ul>	3 semaines.
<b>Clôture</b>	<ul style="list-style-type: none"> <li>- Arrêt progressif des composants de l'ancien système.</li> <li>- Validation finale par les parties prenantes.</li> <li>- Rapport de clôture du projet.</li> </ul>	<ul style="list-style-type: none"> <li>- Système ancien désactivé.</li> <li>- Rapport final.</li> </ul>	2 semaines.

### 3. Gestion de la coexistence

Pendant la phase de migration, un plan de coexistence est mis en place :

- Les services critiques de l'ancien système resteront disponibles pour garantir la continuité des opérations.
- Un processus de synchronisation des données sera assuré pour que les bases de données de l'ancien et du nouveau système soient alignées.

### 4. Suivi et validation

- **Indicateurs de suivi :**

- Pourcentage des fonctionnalités migrées avec succès.
- Taux de disponibilité pendant la migration.
- Retour des utilisateurs sur les nouvelles fonctionnalités.

- **Tests et validations :**

- Chaque phase inclura des tests d'intégration et des tests utilisateurs avant de passer à la phase suivante.
- Les parties prenantes valideront chaque étape clé pour garantir l'alignement avec les objectifs.

## Phase temporelle

Fenêtre temporelle	Description	Durée ou date cible	Commentaires
Début des travaux d'architecture	Validation des besoins business et techniques, et lancement des travaux de conception.	Mois 1	Comprend l'identification des priorités et des standards d'architecture.
Livraison des microservices prioritaires	Livraison des services de gestion des utilisateurs et de géolocalisation.	Mois 4	Ces microservices sont essentiels pour valider l'architecture et initier les tests d'intégration.
Tests d'intégration complets	Réalisation de tests de charge, de performance, et d'intégration sur l'ensemble des composants.	Mois 6	Ces tests garantissent que l'architecture répond aux exigences de scalabilité et de performance.
Déploiement initial en production	Mise en production des premiers microservices validés.	Mois 7	Inclut la surveillance post-déploiement et les optimisations nécessaires en production.
Phase d'extension des fonctionnalités	Développement des microservices pour la gestion des transactions et des offres alimentaires.	Mois 8 à 10	Cette phase permet d'ajouter des fonctionnalités avancées en optimisant la dette technique.
Validation finale de l'architecture	Audit de conformité, validation des KPIs, et validation officielle de l'architecture livrée.	Mois 11	Le comité d'architecture validera les résultats par rapport aux objectifs initiaux.
Clôture du projet	Finalisation de la documentation, transfert aux équipes de maintenance, et clôture administrative.	Mois 12	Comprend une évaluation finale des résultats et une rétrospective avec les parties prenantes.

- **Fenêtre temporelle** : Ce tableau détaille les jalons critiques et leurs dates cibles, offrant une vue d'ensemble du calendrier et des priorités.
- **Alignement stratégique** : Ces dates permettent de coordonner les efforts de toutes les parties prenantes et de suivre les progrès du projet en temps réel.
- **Flexibilité** : Si des ajustements sont nécessaires, ces fenêtres peuvent servir de points de contrôle pour reprogrammer les étapes sans compromettre les objectifs globaux.

## Métriques Business de l'architecture

Métrique	Description	Valeur cible	Justification	Commentaires
Taux de rétention des	Pourcentage d'utilisateurs qui	≥ 70 %	Indique la satisfaction	Analyse mensuelle basée sur les

<b>utilisateurs</b>	continuent d'utiliser la plateforme après leur inscription.		utilisateur et l'efficacité de l'architecture à répondre aux besoins des clients.	données de connexion et d'interaction des utilisateurs.
<b>Taux d'engagement utilisateur</b>	Pourcentage d'utilisateurs actifs interagissant régulièrement avec la plateforme (par mois).	≥ 50 %	Mesure l'attractivité et la pertinence des fonctionnalités offertes par l'architecture.	Collecte des données via les journaux d'interaction et outils analytiques.
<b>Croissance du nombre d'utilisateurs</b>	Augmentation du nombre total d'utilisateurs actifs sur la plateforme.	+ 20 % par an	Mesure l'impact de l'architecture sur la capacité de Foosus à attirer de nouveaux utilisateurs.	Calculée sur une base annuelle avec des rapports trimestriels pour suivre la progression.
<b>Temps moyen de mise sur le marché</b>	Temps nécessaire pour concevoir, développer, et déployer une nouvelle fonctionnalité.	< 4 semaines	Mesure la rapidité de l'innovation grâce à l'architecture modulaire et aux pipelines CI/CD.	Évaluée à chaque sprint de développement pour identifier les points d'amélioration.
<b>Taux de conversion</b>	Pourcentage d'utilisateurs visitant la plateforme qui interagissent avec un producteur ou un artisan local.	≥ 25 %	Indique l'efficacité de l'architecture à connecter les utilisateurs avec les producteurs.	Analyse trimestrielle basée sur les données d'interactions et de transactions enregistrées.
<b>Disponibilité de la plateforme</b>	Pourcentage de temps où la plateforme est pleinement opérationnelle sans interruption.	≥ 99,9 %	Reflète la fiabilité de l'architecture à fournir un service constant aux utilisateurs.	Surveillance en temps réel via des outils de monitoring.
<b>Satisfaction des parties prenantes internes</b>	Score de satisfaction des équipes business et techniques envers l'architecture.	≥ 80 %	Mesure l'alignement de l'architecture avec les attentes et priorités des équipes internes.	Évaluée semestriellement par des enquêtes auprès des équipes produit et développement.

## Explications

- **Métriques quantitatives et qualitatives** : Les métriques combinent des indicateurs directs (comme la rétention et l'engagement) et des retours qualitatifs (comme la satisfaction des parties prenantes).
- **Mesure de l'impact business** : Ces métriques montrent comment l'architecture contribue directement aux objectifs de Foosus en termes de croissance, engagement, et satisfaction.
- **Cycle de suivi** : Chaque métrique est associée à une fréquence de mesure (mensuelle, trimestrielle, ou semestrielle) pour assurer un suivi constant et des ajustements si nécessaires.

# Accords de service pour l'architecture (y compris accord du niveau de service [ANS])

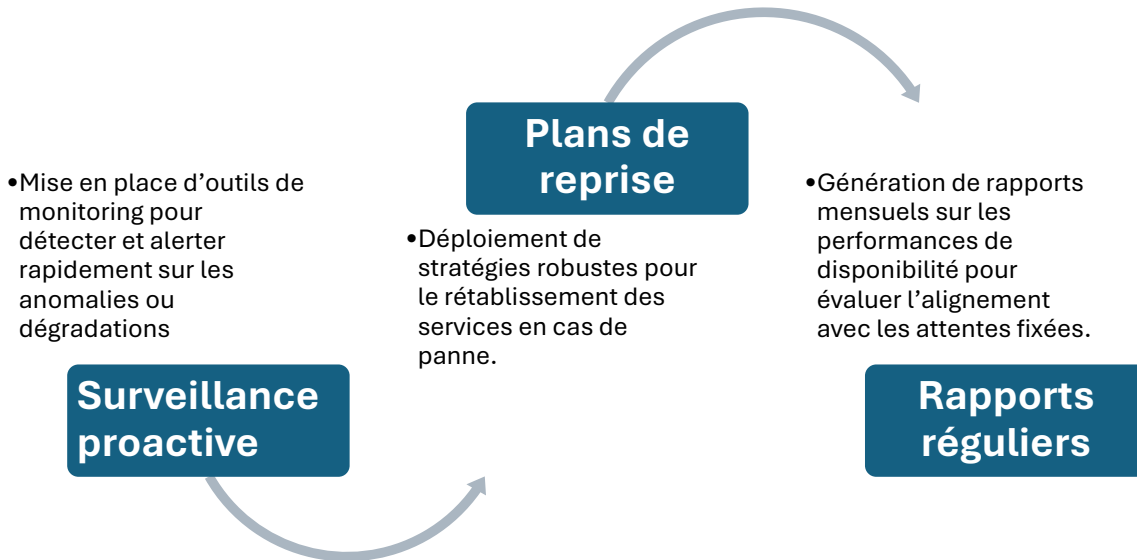
## Attentes de disponibilité

Les attentes de disponibilité suivantes ont été déterminées en accord avec les parties prenantes business de Foosus.

Ces objectifs garantissent une expérience utilisateur fluide et une fiabilité adaptée aux besoins critiques du business

Aspect de disponibilité	Attente	Justification
Disponibilité générale de la plateforme	≥ 99,9 % sur une base mensuelle	Assure un accès continu pour les utilisateurs finaux, minimisant les interruptions de service.
Temps de rétablissement après incident (RTO)	≤ 2 heures pour les incidents critiques	Réduction de l'impact des pannes critiques sur l'expérience utilisateur et les opérations business.
Durée maximale de dégradation des performances (MTTR)	≤ 4 heures pour rétablir les performances normales en cas de dégradation	Garantit que les utilisateurs peuvent rapidement retrouver une expérience optimale.
Disponibilité des microservices prioritaires	≥ 99,95 % pour les services critiques (ex : géolocalisation, gestion des utilisateurs)	Ces services sont essentiels à la fonctionnalité principale de la plateforme.
Fenêtre de maintenance planifiée	1 fois par mois, durée maximale de 2 heures, avec notification préalable	Limite les interruptions planifiées à des périodes de faible utilisation pour minimiser les impacts.
Notification en cas d'incident	≤ 15 minutes après la détection d'un incident majeur	Assure une communication proactive avec les parties prenantes internes et externes.

## Processus pour atteindre les attentes



## Engagement

Ces attentes de disponibilité définissent des objectifs mesurables qui alignent les besoins du business avec la capacité de l’architecture à fournir des services fiables.

En cas d’écart significatif, des ajustements seront planifiés en collaboration avec les équipes business et techniques.

# Processus CI/CD

Le processus d’intégration et de déploiement continu (CI/CD) est conçu pour assurer une livraison rapide, fiable et automatisée des fonctionnalités dans l’architecture microservices. Voici les étapes clés et les outils impliqués dans ce processus.

## Objectifs du processus CI/CD

Avantage	Description
<b>Automatisation complète</b>	Réduction des interventions manuelles pour minimiser les erreurs.
<b>Fiabilité</b>	Déploiement uniquement après validation par des tests automatisés.
<b>Rapidité</b>	Déploiements fréquents pour intégrer rapidement les retours utilisateurs.
<b>Standardisation</b>	Adoption de pratiques uniformes pour tous les microservices.

# Pipeline CI/CD proposé

Le pipeline est structuré en plusieurs étapes, chacune avec des objectifs spécifiques.

Étape	Description	Outils utilisés
<b>Commit et intégration</b>	- Les développeurs soumettent leur code dans un dépôt Git. - Déclenchement automatique du pipeline.	- GitLab CI/CD, GitHub Actions
<b>Compilation</b>	- Compilation et vérification du code. - Validation des dépendances.	- Docker, Maven, Gradle
<b>Tests automatisés</b>	- Exécution des tests unitaires et d'intégration. - Vérification des performances et de la sécurité.	- JUnit, Selenium, SonarQube
<b>Build des conteneurs</b>	- Création d'images Docker pour les microservices.	- Docker, Podman
<b>Déploiement en staging</b>	- Déploiement sur un environnement de préproduction pour validation manuelle ou tests supplémentaires.	- Kubernetes, Helm, Ansible
<b>Déploiement en production</b>	- Déploiement progressif dans l'environnement de production après validation de l'étape précédente.	- Kubernetes, AWS CodeDeploy

## Règles de déploiement

- **Conditionnement par les tests :**
  - Si les tests automatisés échouent, le pipeline s'arrête immédiatement, et les développeurs reçoivent une alerte.
  - Si tous les tests réussissent, le pipeline continue automatiquement vers les étapes de build et de déploiement.

Cliquez [ici](#) pour voir le diagramme complet des règles de déploiement

- **Progression des environnements**  
DEV => PREPROD => PROD
  - **Environnement de développement (DEV)** : Déploiement initial pour validation technique.
  - **Environnement de préproduction (PREPROD)** : Validation finale par les parties prenantes.
  - **Environnement de production (PROD)** : Déploiement progressif pour limiter les risques.
- **Déploiement bleu/vert ou canary**
  - Utilisation d'une stratégie de déploiement "bleu/vert" ou "canary" pour minimiser les interruptions de service.

## Comparaison des deux stratégies

Critère	Déploiement Bleu/Vert	Déploiement Canary
Vitesse de déploiement	Basculé d'un coup (rapide).	Progressif (plus lent).
Réversibilité	Très rapide (retour au Bleu possible immédiatement).	Plus complexe, nécessite une gestion par étapes.
Risque d'impact	Très faible une fois basculé.	Limité à une petite portion d'utilisateurs.
Usage idéal	Changements majeurs.	Changements mineurs ou fréquents.

## Pourquoi les utiliser les déploiement bleu/vert ou Canary pour Foosus ?

Foosus, en tant que plateforme critique pour les producteurs et consommateurs locaux, doit minimiser les risques liés aux mises à jour en production.

- **Déploiement bleu/vert**  
Idéal pour des changements majeurs, comme l'introduction de nouveaux microservices ou des modifications structurelles importantes, avec une réversibilité rapide en cas de problème.
- **Déploiement Canary**  
Adapté aux mises à jour mineures, il permet de tester de nouvelles fonctionnalités sur une petite portion des utilisateurs avant un déploiement global, limitant ainsi les impacts.

Ces stratégies garantissent continuité de service et fiabilité tout en favorisant l'innovation rapide.

Cliquez [ici](#) pour voir le diagramme complet des règles de déploiement

## Suivi et monitoring

Pour garantir le bon fonctionnement du pipeline, des outils de monitoring et de logs sont intégrés :

**Monitoring** : Prometheus, Grafana.

**Logs** : ELK Stack (Elasticsearch, Logstash, Kibana).

**Alertes** : Configuration d'alertes automatiques pour détecter et résoudre rapidement les problèmes.

## Avantages du processus CI/CD

Avantages	Description
-----------	-------------



<b>Qualité accrue</b>	Identification rapide des erreurs grâce aux tests automatisés.
<b>Gain de temps</b>	Moins de délais grâce à l'automatisation des étapes.
<b>Flexibilité</b>	Déploiements fréquents permettant d'intégrer rapidement les nouvelles fonctionnalités.
<b>Fiabilité</b>	Réduction des interruptions de service grâce aux stratégies de déploiement progressif.

## Personnes approuvant ce plan

---

Valideur	Domaines de responsabilité	Date
<b>Ash Callum</b>	CEO – Validation stratégique et alignement business	-
<b>Natasha Jarson</b>	CIO – Supervision de l'architecture et des systèmes IT	-
<b>Daniel Anthony</b>	CPO – Gestion des produits et priorités fonctionnelles	-
<b>Pete Parker</b>	Responsable Ingénierie – Implémentation et performance	-
<b>Architecte Principal</b>	Architecture – Conception, standards, et alignement technique	-