



## Déclaration de Travail d'Architecture - FOOSUS

Version : 1.0

Auteur : Clément Hindié

Date : 20 janvier 2026

Document : Déclaration de Travail d'Architecture

Projet : Nouvelle architecture

## *Historique des modifications*

## Tables des matières

1.	Objet du document .....	4
2.	Vision d'Architecture .....	4
2.1	Intention.....	4
2.2	Principes structurants .....	5
3.	État Cible de l'Architecture.....	5
3.1	Architecture cible.....	5
3.1.1	Digramme d'Architecture .....	6
3.1.2	Composants de l'architecture cible .....	7
3.2	Domaines fonctionnels.....	8
3.3	Coexistence avec l'existant.....	8
4.	Portée du Travail d'Architecture .....	8
5.	Parties Prenantes.....	9
6.	Approche d'Architecture .....	9
7.	Livrables .....	10
8.	Critères de réussite.....	10
9.	Risques et Atténuations .....	11
10.	Engagements .....	12

## 1. Objet du document

Ce document définit le cadre de travail architectural pour la nouvelle plateforme FOOSUS.

Il présente la vision, les objectifs, les responsabilités et l'approche retenue pour construire une architecture cohérente, évolutive et adaptée à la croissance de l'entreprise.

Il constitue une adaptation sur mesure du cadre TOGAF, appliquée de manière pragmatique et lean : seules les activités, vues et décisions réellement utiles à FOOSUS sont retenues. L'objectif est de fournir une base technique saine et durable, capable de soutenir l'innovation produit tout en garantissant la stabilité opérationnelle.

## 2. Vision d'Architecture

### 2.1 Intention

FOOSUS doit disposer d'une plateforme géo-centrée, modulaire et fiable, capable de connecter efficacement consommateurs et producteurs locaux.

Cette plateforme doit permettre aux équipes produit d'expérimenter rapidement, tout en assurant que l'architecture soutienne la croissance plutôt que de la freiner.

Cette vision correspond à la phase Architecture Vision du cadre TOGAF, adaptée à la culture lean de FOOSUS.

## 2.2 Principes structurants

Les principes suivants guident toutes les décisions architecturales :

- **Responsabilités claires** : chaque service couvre un périmètre fonctionnel limité et non ambigu.
- **Normalisation pragmatique** : des normes existantes, mais ne brident pas la créativité.
- **Évolutivité horizontale** : la plateforme doit absorber les pics de charge sans dégrader l'expérience utilisateur.
- **Livraisons continues** : versions petites, peu risquées, déployées sans interruption.
- **Sécurité intégrée dès la conception** : aucun compromis entre facilité d'usage et protection des données.
- **Observabilité native** : logs structurés, métriques, traces et alertes dès le départ.
- **Réversibilité** : les décisions doivent être faciles à inverser tant que possible.
- **Coexistence maîtrisée** : migration progressive depuis l'existant, sans bascule brutale.

Ces principes constituent le framework d'architecture sur mesure de FOOSUS.

## 3. État Cible de l'Architecture

### 3.1 Architecture cible

L'architecture cible repose sur un ensemble de micro-services organisés par domaines métier et exposant des API standardisées.

Elle doit :

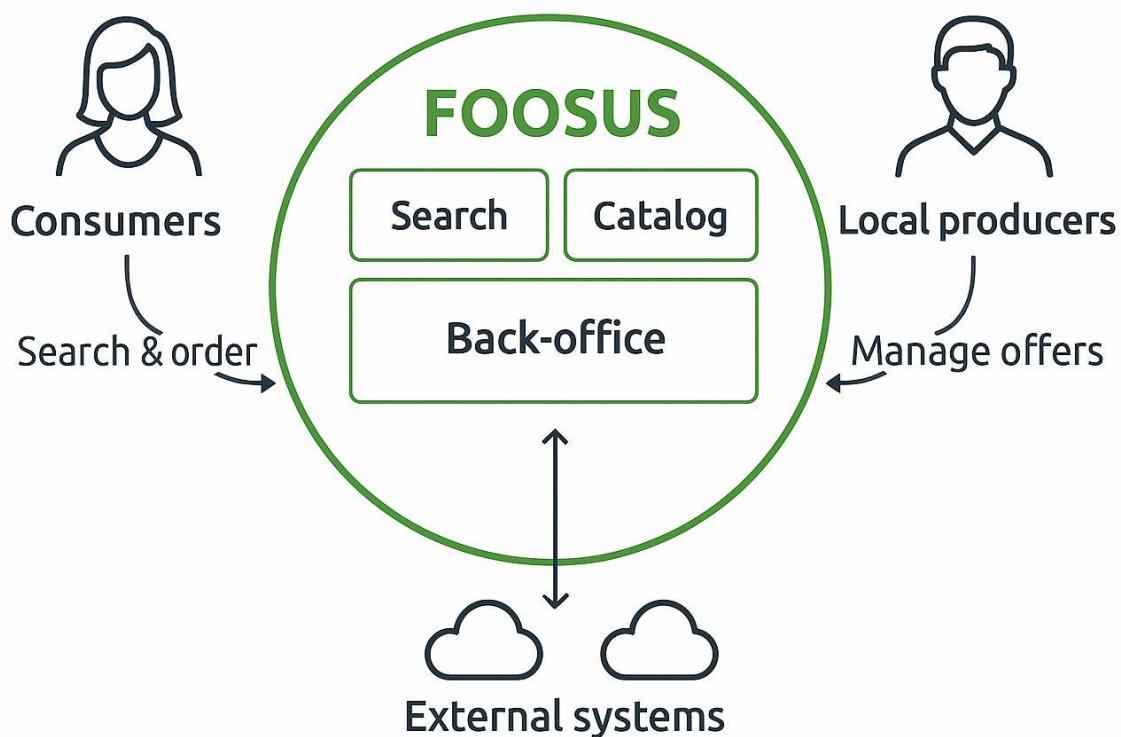
- Fonctionner dans un environnement cloud.
- Intégrer la géolocalisation comme capacité native.
- Supporter plusieurs régions géographiques.
- Fonctionner en continu sans interruption lors des déploiements.
- Coexister temporairement avec la plateforme actuelle.
- Offrir une expérience fluide sur mobile même avec une faible bande passante.

Cette cible servira de référence pour les phases ultérieures du cadre TOGAF (Business, Data, Application et Technology Architecture), qui seront détaillées au fil des itérations.

### 3.1.1 Diagramme d'Architecture

La plateforme FOOSUS repose sur une architecture modulaire organisée autour de domaines métiers clairement définis.

Le diagramme qui suit, illustre les principaux composants, leurs responsabilités et leurs interactions avec les parties prenantes internes et externes.



### 3.1.2 Composants de l'architecture cible

L'architecture cible s'appuie sur un ensemble de composants spécialisés, chacun responsable d'un périmètre fonctionnel précis.

Ces composants sont conçus pour être faiblement couplés, observables et déployables indépendamment.

- **Search Service :**

- Rôle : recherche géolocalisée des offres et des producteurs
- Capacités : indexation, filtrage, système de notation, géo-distance
- Interfaces : API REST /search
- Justification : composant critique pour l'expérience utilisateur

- **Catalog Service :**

- Rôle : gestion des offres alimentaires et métadonnées associées
- Capacités : CRUD des offres, disponibilité, catégorisation
- Interfaces : API REST /catalog
- Justification : cœur de métier de FOOSUS

- **User & Supplier Service :**

- Rôle : gestion des comptes consommateurs et producteurs
- Capacités : profils, préférences, vérifications
- Interfaces : API REST /users et /suppliers
- Justification : séparation claire des responsabilités

- **Back-Office Service :**

- Rôle : supervision, qualité, support, modération
- Capacités : tableau de bord, alertes, gestion des incidents
- Interfaces : API REST /admin
- Justification : indispensable pour l'exploitation

- **Experimentation Service :**

- Rôle : gestion des feature flags et variantes
- Capacités : activation ciblée, tests A/B, suivi des métriques
- Interfaces : API REST /experiments
- Justification : support à l'innovation produit

- **Geo Service :**

- Rôle : géolocalisation, calculs de distance, normalisation d'adresses
- Capacités : géocodage, reverse-géocodage, clustering
- Interfaces : API REST /geo
- Justification : capacité native de la plateforme

- **External Systems :**
  - Rôle : intégration avec services tiers (paiements, cartographie, notifications)
  - Justification : modularité et évolutivité

### 3.2 Domaines fonctionnels

Les domaines concernés incluent :

- Recherche géolocalisée.
- Catalogue et offres alimentaires.
- Gestion des fournisseurs et des consommateurs.
- Opérations de back-office.
- Capacités d'expérimentation (feature flags, variantes).
- Paiements (phase ultérieure).

### 3.3 Coexistence avec l'existant

La plateforme actuelle sera maintenue en mode maintenance.

La migration vers la nouvelle architecture se fera progressivement, fonctionnalité par fonctionnalité, selon un strangler pattern.

Cette approche réduit les risques, évite les interruptions de service et permet une montée en charge progressive.

## 4. Portée du Travail d'Architecture

Cette première phase couvre :

- La définition de l'architecture cible,
- L'identification des exigences architecturales fonctionnelles et non fonctionnelles,
- L'élaboration des contrats architecturaux,
- La définition des standards techniques initiaux,
- La production des vues d'architecture nécessaires à la compréhension globale,
- La mise en place d'une gouvernance LEAN adaptée à FOOSUS.

Ne sont pas inclus :

- L'implémentation détaillée des micro-services.
- La migration complète de l'existant.<sup>1</sup>
- L'intégration des paiements tiers.
- Le choix définitif du cloud provider (à instruire ultérieurement).

## 5. Parties Prenantes

Partie prenante	Rôle et responsabilités
<b>Direction générale (CEO)</b>	Porte la vision business, arbitre les priorités stratégiques et valide les orientations structurantes.
<b>CPO</b>	Définit les capacités métier, pilote la priorisation produit et garantit l'alignement avec la feuille de route.
<b>CIO</b>	Assure la gouvernance technique, l'alignement organisationnel et la cohérence des pratiques.
<b>Responsable Ingénierie</b>	Encadre les standards de développement, garantit la qualité d'implémentation et soutient les équipes techniques.
<b>Équipes de développement</b>	Conçoivent, implémentent et testent les services ; apportent un retour d'expérience continu.
<b>Fonction Architecture</b>	Porte la vision globale, définit les garde-fous, documente les décisions et accompagne les équipes.

## 6. Approche d'Architecture

L'approche retenue est incrémentale, LEAN et collaborative.

Elle repose sur :

- Des cycles courts.
- Des artefacts utiles et accessibles.
- Des décisions documentées mais non figées.
- Une implication continue des équipes techniques et produit.

Les activités principales incluent :

- Analyse du système existant et identification des points de rupture.
- Définition des domaines métier (DDD).
- Définition des micro-services et de leurs frontières.
- Conception des API et contrats interservices.
- Définition des exigences non fonctionnelles.
- Stratégie de migration.
- Définition des standards techniques.
- Mise en place d'une gouvernance lean.

## 7. Livrables

Avant de pouvoir engager l'ensemble des équipes dans une trajectoire commune, il est essentiel de rendre l'architecture visible, compréhensible et partageable. Les livrables suivants constituent la base documentaire qui permettra à chacun — direction, produit, design et développement — de s'aligner sur une vision claire et d'avancer de manière cohérente.

- Déclaration de Travail d'Architecture
- Spécification des Conditions Requises pour l'Architecture
- Contrats architecturaux (business, design, développement)
- Vues d'architecture :
  - Vue de contexte
  - Vue des domaines
  - Vue micro-services
  - Vue de coexistence ancien/nouveau
  - Vue de déploiement
- Diagrammes UML locaux
- Modèle de gouvernance

## 8. Critères de réussite

Avant de pouvoir affirmer que l'architecture proposée remplit son rôle, il est nécessaire de définir des repères concrets et mesurables. Ces critères permettront d'évaluer objectivement les progrès réalisés et de valider que la plateforme soutient réellement la croissance et l'innovation attendues.

- Réduction des incidents critiques à moins d'un par mois.
- Lead time inférieur à une semaine.
- Mise en production d'une recherche géolocalisée opérationnelle dès le premier trimestre.
- Validation de l'architecture par les parties prenantes.
- Adoption des standards par les équipes.
- Premiers micro-services déployés sans interruption de service.

## 9. Risques et Atténuations

Avant de détailler les mesures d'atténuation, il est utile d'évaluer chaque risque selon une échelle simple. Le système de notation permet de prioriser les efforts et de rendre explicite le niveau d'attention requis pour chaque point sensible.

Échelle de notation :

- Probabilité : 1 (faible) → 5 (très élevée)
- Impact : 1 (faible) → 5 (très élevé)
- Score global = Probabilité × Impact
- 1–8 : risque maîtrisable
- 9–15 : risque significatif
- 16–25 : risque critique

Risque	Probabilité (1–5)	Impact (1–5)	Score	Stratégie d'atténuation
<b>Fragmentation technologique</b>	4	5	20	Standards clairs, accompagnement des équipes, revue régulière des pratiques.
<b>Résistance au changement</b>	3	5	15	Communication transparente, cadre non contraignant, implication progressive.
<b>Dette technique persistante</b>	5	5	25	Migration progressive, priorisation conjointe, visibilité sur la dette.
<b>Décisions difficiles à inverser</b>	3	5	15	Patterns réversibles, expérimentation contrôlée, documentation des choix.
<b>Charge de travail excessive</b>	3	4	12	Découpage en phases, priorisation stricte, synchronisation régulière.

## 10. Engagements

La réussite de cette initiative repose sur un engagement mutuel. La Fonction Architecture doit fournir un cadre clair et soutenant, tandis que les parties prenantes doivent contribuer activement à la cohérence et à la discipline nécessaires pour faire évoluer la plateforme.

La Fonction Architecture s'engage à :

- Fournir une vision claire et cohérente.
- Documenter les décisions structurantes.
- Accompagner les équipes dans la montée en maturité.
- Garantir la cohérence globale de la plateforme.

Les parties prenantes s'engagent à :

- Participer aux ateliers.
- Respecter les contrats architecturaux.
- Contribuer à la normalisation.
- Prioriser les travaux nécessaires à la santé du système.