

FOOSUS

# Spécification des Conditions Requises

CLEMENT HINDIE

27/01/2026

## Historique des modifications

[illegible]

## Tables des matières

1.	Objet du document .....	3
2.	Contexte .....	3
3.	Parties prenantes .....	3
4.	Exigences fonctionnelles d'architecture .....	3
4.1	Géolocalisation et recherche.....	3
4.2	Gestion des rôles et parcours.....	3
4.3	Coexistence avec l'existant.....	3
5.	Exigences non fonctionnelles .....	5
5.1	Performance et scalabilité .....	5
5.2	Disponibilité et déploiement .....	5
5.3	Sécurité .....	5
5.4	Observabilité.....	5
5.5	Réversibilité et expérimentation.....	6
5.6	Engagements de service (SLA/SLO/SLI).....	6
6.	Contraintes et hypothèses.....	7
7.	Conditions de conformité .....	7
7.1	Revue d'architecture .....	7
7.2	Lignes directrices d'implémentation.....	8
7.3	Normes d'implémentation.....	9
7.4	2 Standards techniques .....	9
7.5	Tests et observabilité.....	9
7.6	Documentation minimale .....	10
8.	Validation.....	10

## 1. Objet du document

## 2. Contexte

FOOSUS doit concevoir une plateforme géo-centrée, évolutive et fiable, capable de connecter efficacement les différents acteurs pour l'architecture de la nouvelle plateforme FOOSUS.

La plateforme doit intégrer la géolocalisation comme capacité fondamentale.

La plateforme doit supporter un modèle d'évolution à long terme, capable de soutenir la croissance ni les besoins de distribution.

F-5 Les exigences fonctionnelles et non fonctionnelles,

Les exigences de performance de calcul, la distance entre un utilisateur et un fournisseur.

Le processus doit inclure une phase de sélection, de validation, de confirmation et d'ajustement futur avec des prestataires.

Le processus doit inclure une phase de sélection, de validation, de confirmation et d'ajustement futur avec des prestataires.

Le processus doit inclure une phase de sélection, de validation, de confirmation et d'ajustement futur avec des prestataires.

Le processus doit inclure une phase de sélection, de validation, de confirmation et d'ajustement futur avec des prestataires.

## 3. Parties prenantes

Partie prenante	Rôle et responsabilités
<b>Direction générale (CEO)</b>	Porte la vision business et valide les orientations structurantes.
<b>CPO</b>	Définit les capacités métier et pilote la priorisation produit.
<b>CIO</b>	Assure la gouvernance technique et l'alignement organisationnel.
<b>Responsable Ingénierie</b>	Encadre les standards de développement et garantit la qualité d'implémentation.
<b>Équipes de développement</b>	Conçoivent, implémentent et testent les services.
<b>Fonction Architecture</b>	Porte la vision globale, définit les garde-fous et documente les décisions

## 4. Exigences fonctionnelles d'architecture

### 4.1 Géolocalisation et recherche

### 4.2 Gestion des rôles et parcours

### 4.3 Coexistence avec l'existant

La nouvelle plateforme doit fonctionner en parallèle de l'ancienne.

**F-7 Redirection progressive :**

Certaines fonctionnalités (ex. recherche) doivent pouvoir être migrées indépendamment.

## 5. Exigences non fonctionnelles

### 5.1 Performance et scalabilité

#### NF-1 Scalabilité horizontale :

Les services doivent pouvoir être répliqués horizontalement.

#### NF-2 Temps de réponse :

#### NF-4 Sécurité des données :

Le système doit être protégé des origines d'attaque, Validité et de l'interaction des données.

#### NF-8 Disponibilité sans interruption : Données personnelles :

Des données de confiance doivent être disponibles pour les utilisateurs réglementations.

#### NF-6 Livraisons Fréquentes :

Le délai moyen de mise en production doit être inférieur à une semaine.

Les opérations critiques doivent répondre en moins d'un seuil défini (à préciser avec le produit).

#### NF-3 Comportement dégradé :

Le système doit rester utilisable même en surcharge.

### 5.2 Disponibilité et déploiement

### 5.3 Sécurité

### 5.4 Observabilité

## 5.5 Réversibilité et expérimentation

### 5.6 Engagements de service (SLA/SLO/SLI)

Les engagements de service suivants définissent les niveaux de performance, de fiabilité et de qualité attendus pour la plateforme FOOSUS. Ils permettent d'évaluer objectivement la conformité des implémentations et de garantir une expérience utilisateur cohérente, même en période de croissance ou de forte charge.

- SLA Disponibilité globale de la plateforme : 99.95 % (mensuel)
- NF-11 Décisions réversibles :
  - Taux d'erreur critique (5xx) :  $\leq 0.1$  %
  - Les choix doivent être faciles à inverser lorsque possible
- Les déploiements sans interruption : 100 % des mises en production

NF-12 Feature flags :

La plateforme doit permettre l'activation/désactivation de fonctionnalités.

Les SLA représentent les engagements de disponibilité et de performance perçus par les utilisateurs finaux.

Ces engagements constituent la base de référence pour les équipes produit, design et développement.

#### SLO — Service Level Objectives (objectifs internes par service)

Les SLO définissent les objectifs opérationnels permettant d'atteindre les SLA. Ils sont mesurés par service et servent de garde-fous techniques.

Service	Disponibilité cible	Latence cible
<b>Search API</b>	$\geq 99.95$ %	$\leq 300$ ms
<b>Catalogue API</b>	$\geq 99.9$ %	$\leq 400$ ms
<b>Geo API</b>	$\geq 99.9$ %	$\leq 250$ ms
<b>Bookings API</b>	$\geq 99.9$ %	$\leq 500$ ms
<b>Notifications</b>	$\geq 99.5$ %	$\leq 1$ s

Ces objectifs sont révisables à chaque itération majeure, en fonction des retours d'expérience et de la montée en charge.

**SLI — Service Level Indicators (indicateurs mesurés)**

Les SLI sont les métriques observées en production permettant de vérifier l'atteinte des SLO.

- L'alignement avec les domaines métier,
- Le respect des exigences non fonctionnelles,
- La stratégie de déploiement et de migration.

Indicateur	Description
<b>Disponibilité</b>	Pourcentage de requêtes réussies sur le total
<b>Latence</b>	P95 des temps de réponse mesurés
<b>Taux d'erreur</b>	Ratio des erreurs 5xx sur le total des requêtes
<b>Déploiement</b>	Pourcentage de mises en production réalisées sans interruption
<b>Observabilité</b>	Présence de logs structurés, métriques et traces distribuées pour chaque service

Ces indicateurs sont collectés via les outils d'observabilité définis dans la section NF9.

## 6. Contraintes et hypothèses

### C-1 Plateforme existante en maintenance uniquement :

Aucune nouvelle fonctionnalité ne sera développée sur l'ancien système.

### C-2 Hébergement cloud privilégié :

Le choix du fournisseur reste à instruire.

### C-3 Culture LEAN :

Les artefacts doivent rester légers, utiles et révisables.

## 7. Conditions de conformité

### 7.1 Revue d'architecture

Toute capacité structurante doit passer par une revue courte validant :



## 7.2 Lignes directrices d'implémentation

- Respect strict des frontières de domaine :

Chaque service doit implémenter uniquement les règles métier de son domaine.  
Aucun accès direct à la base de données d'un autre service n'est autorisé.

- API explicites et stables :

Les interactions entre services doivent passer exclusivement par des API documentées.  
Les évolutions doivent être rétro-compatibles.

- Feature flags systématiques :

Toute nouvelle fonctionnalité doit être activable/désactivable indépendamment du déploiement.

- Déploiements incrémentaux :

Les mises en production doivent être petites, fréquentes et sans interruption.

- Gestion des erreurs standardisée :

Les services doivent renvoyer des erreurs structurées, avec un format commun (code, message, cause).

- Tolérance aux pannes :

Les services doivent gérer les indisponibilités temporaires via des mécanismes tels que réessaie automatique, timeout, circuit breaker.

- Stockage indépendant :

Chaque service possède son propre stockage, adapté à son besoin (SQL, NoSQL, cache).

- Observabilité intégrée dès le développement :

Logs structurés, métriques, traces distribuées doivent être présents avant toute mise en production.

## 7.3 Normes d'implémentation

Les implémentations doivent respecter les normes suivantes :

- **APIs RESTful normalisées :**

La documentation est obligatoire, au format OpenAPI, avec un format JSON unique, une gestion de version explicite (ex. /v1/...) et l'utilisation de codes HTTP standardisés.

- Des tests automatisés à l'API,
- Des règles de sécurité,
- Des logs et des traces d'observabilité,
- Des traces distribuées.

- **Sécurité :**

L'authentification doit s'appuyer sur OAuth2 / OIDC, l'autorisation sur un modèle RBAC, le chiffrement TLS doit être systématique et les actions sensibles doivent être journalisées.

- **Qualité logicielle :**

Les règles métier critiques doivent être couvertes par des tests unitaires ( $\geq 80\%$ ), chaque API doit disposer de tests d'intégration, le code doit être analysé statiquement et toute modification doit faire l'objet d'une revue de code

- **Observabilité :**

Chaque service doit exposer des métriques via Prometheus, produire des traces distribuées (OpenTelemetry), générer des logs structurés au format JSON et disposer de tableaux de bord Grafana dédiés.

- **Intégration et déploiement continus :**

Les chaînes de traitement doivent être entièrement automatisées (construction, tests, analyse, déploiement), les mises en production doivent être progressives (blue/green ou rolling update) et la promotion en production doit être conditionnée à la validation automatique des SLO.

- **Nomenclature :**

Les services doivent suivre la convention foosus-<domaine>-service, les API doivent être exposées sous /api/<domaine>/<ressource> et les métriques doivent respecter le format foosus\_<service>\_<mesure>.

## 7.4 2 Standards techniques

Les implémentations doivent respecter :

## 7.5 Tests et observabilité

Chaque service doit fournir :

## 7.6 Documentation minimale

Chaque service doit être documenté par :

## 8. Validation

- Un diagramme de contexte local.
- Ce document est validé conjointement par :
  - Ses dépendances principales.
  - Ses Fonctionnaires.
  - Le CPO.
  - Le CIO.
  - Le Responsable Ingénierie.

Il constitue la référence pour toutes les décisions architecturales de la phase 1.