

# Déclaration de Travail d'Architecture



## Foosus Géoconscient

<b>Nom du projet</b>	Foosus Géoconscient
<b>Préparé par</b>	JOUDAR Mohamed
<b>Version N°</b>	1.2
<b>Titre</b>	Déclaration de Travail d'Architecture
<b>Date de version</b>	30/09/2025
<b>Revu par</b>	N/A
<b>Date de révision</b>	N/A
<b>Historique de version</b>	v1.0 (24/09/2025) v1.2 (01/10/2025)

<b>1. Objet de ce document</b>	<b>5</b>
<b>2. Déclaration de travail d'architecture</b>	<b>5</b>
2.1. Requête du projet et contexte	5
2.2. Description du projet et périmètre	5
2.2.1. Description du projet	5
2.2.2. Périmètre du projet	6
2.3. Vue d'ensemble	6
2.4. Alignement stratégique	6
<b>3. Objectifs et périmètre</b>	<b>7</b>
3.1. Objectifs	7
3.2. Périmètre	7
3.3. Parties prenantes, préoccupations et visions	8
3.4. Approche managériale	9
3.4.1. L'approche Lean modernisée	9
3.4.2. L'approche Agile orientée platform	9
3.5. Procédures de changement de périmètre	9
<b>4. Rôles et responsabilités</b>	<b>10</b>
4.1. Structure de gouvernance	10
4.2. Process du projet	11
4.2.1. Process	11
4.2.2. Indicateurs	13
4.3. Rôles et responsabilités	14
<b>5. Approche architecturale</b>	<b>15</b>
5.1. Principes architecturaux	15
5.1.1. Une architecture priorisant l'utilisateur	15
5.1.2. Une architecture à l'implémentation itérative	16
5.1.3. Une architecture permettant la cohabitation entre deux solutions	16
5.2. Process d'architecture	16
5.3. Contenu de l'architecture	19
<b>6. Plan de travail</b>	<b>20</b>
6.1. Plan d'architecture	21
6.1.1. Phase préliminaire	21
6.1.2. Phase A – Vision de l'architecture	21
6.1.3. Phase B – Architecture Business	22
6.1.4. Phase C – Architecture des systèmes d'information	23
6.1.5. Phase D – Architecture technologique	24
6.1.6. Phase E – Opportunités et solutions	24
6.1.7. Phase F – Planning de Migration	25
6.1.8. Phase G – Gouvernance de l'implémentation	26
6.1.9. Phase H – Management du changement d'architecture	27
6.1.10. Phase N – Management des conditions requises	27
6.2. Plan de communication	27
6.2.1. Création ou mise à jour d'un document ou d'un artefact	28
6.2.2. Réunion du groupe projet	28
6.2.3. Réunion des sponsors	28
6.3. Collaboration	28

<b>7. Risques et facteurs de réduction</b>	<b>29</b>
7.1. Analyse des risques	29
7.2. Hypothèses	31
<b>8. Critères d'acceptation et procédures</b>	<b>32</b>
8.1. Métriques et KPI	32
8.2. Procédure d'acceptation	32
<b>9. Personnes approuvant ce plan</b>	<b>33</b>

# 1. Objet de ce document

Ce document est une Déclaration de travail d'architecture pour le projet de **Plateforme E-commerce Cloud-Native de Foosus**.

La Déclaration de travail d'architecture définit le périmètre et l'approche qui seront utilisés pour mener à bien un projet d'architecture. La Déclaration de travail d'architecture constitue habituellement le document qui permet de mesurer la réussite de l'exécution du projet d'architecture et peut former la base de l'accord contractuel entre le fournisseur et le consommateur de services d'architecture. En général, toutes les informations de ce document doivent se situer à un haut niveau.

## 2. Déclaration de travail d'architecture

### 2.1. Requête du projet et contexte

Foosus évolue dans l'écosystème dynamique de l'alimentation durable, porté par une mission claire : démocratiser l'accès aux produits alimentaires locaux grâce à une plateforme digitale de mise en relation. L'entreprise a démontré sa capacité d'innovation en connectant efficacement consommateurs et producteurs locaux sur un marché en forte croissance.

Les analyses comportementales et les études de marché confirment l'accélération des tendances de consommation responsable, validant la stratégie de Foosus. Cette dynamique s'accompagne d'exigences technologiques accrues : géolocalisation intelligente en temps réel, personnalisation contextuelle, et expérience utilisateur omnicanale.

L'architecture technique actuelle, conçue pour un contexte géographique limité, constitue désormais un goulot d'étranglement critique face aux ambitions d'expansion mondiale. Les limitations observées incluent une scalabilité horizontale insuffisante, une géolocalisation rudimentaire, et l'absence d'architecture event-driven nécessaire pour supporter les volumes de données et transactions anticipés.

Foosus initie donc une transformation architecturale majeure vers une plateforme cloud-native de nouvelle génération, capable de rivaliser avec les géants internationaux de l'e-commerce tout en préservant son positionnement différenciant sur l'économie circulaire locale.

### 2.2. Description du projet et périmètre

#### 2.2.1. Description du projet

Le projet vise la conception et l'implémentation d'une architecture cloud-native moderne, optimisée pour l'hypercroissance et l'expansion géographique. Cette transformation s'appuie sur les paradigmes technologiques de 2025 : serverless-first, edge computing, et intelligence artificielle distribuée.

L'architecture cible doit répondre aux exigences fonctionnelles suivantes :

- **Géolocalisation intelligente edge-native** : Proposer des produits hyper-localisés via un système de géofencing sophistiqué, exploitant l'edge computing pour des latences sub-100ms et une précision GPS augmentée par ML

- **Déploiement multi-région zero-downtime** : Permettre le déploiement de services spécialisés par région géographique via une approche GitOps et infrastructure-as-code, avec activation instantanée de nouveaux marchés
- **Architecture évolutive API-first** : Faciliter l'extension fonctionnelle par une approche microservices événementielle, avec versioning automatique et backward compatibility garantie
- **Resilience by design** : Éliminer les interruptions de service grâce aux patterns circuit breaker, bulkhead, et deployment canary automatisés

Les capacités techniques émergentes incluent :

- Recommandations alimentaires personnalisées par IA générative
- Optimisation logistique multi-modale en temps réel
- Carbon footprint tracking automatisé per transaction
- Blockchain intégrée pour la traçabilité alimentaire

### 2.2.2. Périmètre du projet

Le périmètre englobe l'écosystème complet de la marketplace Foosus, conçu pour une audience globale multi-culturelle. L'architecture supportera nativement :

**Acteurs métier** : Consommateurs finaux, producteurs/artisans locaux, équipes back-office Foosus, partenaires logistiques, et régulateurs alimentaires régionaux

**Channels digitaux** : Applications natives iOS/Android, Progressive Web App (PWA), interfaces conversationnelles (chatbots/voice), API publiques pour partenaires, et tableaux de bord temps réel

**Zones géographiques** : Architecture multi-région avec data sovereignty compliance (RGPD, CCPA, etc.), adaptabilité linguistique et culturelle automatisée, et intégration des systèmes de paiement locaux

**Innovation technologique** : Intégration native d'IoT agricole, réalité augmentée pour la présentation produits, et blockchain pour la traçabilité end-to-end

## 2.3. Vue d'ensemble

Cette transformation architecturale constitue l'initiative technologique stratégique de Foosus, alignée sur sa mission de leader de l'e-commerce alimentaire responsable. En tant qu'entreprise pure-play, Foosus peut ainsi concentrer l'intégralité de ses ressources d'innovation sur cette architecture unifiée, sans complexité de coordination multi-domaines.

L'approche adoptée privilégie une architecture cloud-native découplée, optimisant time-to-market et capacité d'expérimentation. Cette stratégie technologique soutient directement les objectifs de différenciation concurrentielle et d'agilité business.

## 2.4. Alignement stratégique

L'architecture proposée répond directement aux trois piliers stratégiques de Foosus :

**Excellence de l'expérience utilisateur** : La priorisation d'une architecture user-centric garantit des temps de réponse optimaux via edge computing, une personnalisation poussée par ML/AI, et une

disponibilité 99.99% grâce aux patterns de resilience distribuée. L'approche mobile-first et PWA assure une accessibilité universelle.

**Compétitivité technologique** : L'adoption d'une stack cloud-native moderne (serverless, containerisée, event-driven) permet d'égaliser les performances des plateformes globales tout en conservant l'agilité d'une scale-up. L'architecture API-first facilite l'innovation continue et l'intégration d'écosystèmes partenaires.

**Expansion internationale accélérée** : La conception multi-région native, couplée à une infrastructure-as-code et des déploiements GitOps, permet l'activation de nouveaux marchés en quelques heures plutôt qu'en mois. L'edge computing distribué garantit des performances homogènes à l'échelle planétaire, condition sine qua non de la scalabilité internationale.

## 3. Objectifs et périmètre

### 3.1. Objectifs

Cette transformation architecturale vise à repositionner Foosus comme leader technologique de l'e-commerce alimentaire responsable. Les objectifs business suivants définissent les exigences de performance et de compétitivité :

Objectif business	Capacités techniques cibles
<b>Hyperscalabilité cloud-native</b>	<ul style="list-style-type: none"><li>• Scaling horizontal automatique via Kubernetes et serverless</li><li>• Absorption des pics de trafic par auto-scaling intelligent (99.99% uptime)</li><li>• Architecture event-driven résiliente avec circuit breakers</li><li>• Capacité multi-région avec basculement automatique</li></ul>
<b>Sécurité zero-trust omnicanale</b>	<ul style="list-style-type: none"><li>• Conformité réglementaire automatisée (RGPD, CCPA, SOC2)</li><li>• Chiffrement end-to-end avec rotation automatique des clés</li><li>• Identity &amp; Access Management moderne (OAuth 2.0/OIDC)</li><li>• Sécurité by-design avec threat modeling intégré</li></ul>
<b>Disponibilité globale edge-native</b>	<ul style="list-style-type: none"><li>• Déploiements blue-green zero-downtime via GitOps</li><li>• Edge computing distribué pour latences sub-100ms</li><li>• CDN intelligent avec caching contextuel</li><li>• Progressive Web App (PWA) avec offline-first capability</li></ul>
<b>Agilité développement platform-driven</b>	<ul style="list-style-type: none"><li>• API-first design avec versioning automatique</li><li>• Intégration continue/déploiement continu (CI/CD) moderne</li><li>• Feature flags et A/B testing natifs</li><li>• Microservices découplés avec service mesh</li></ul>
<b>Innovation data-driven</b>	<ul style="list-style-type: none"><li>• Plateformes d'expérimentation cloud-natives</li><li>• ML/AI intégrés pour personnalisation temps réel</li><li>• Observabilité complète (telemetry, traces, logs)</li><li>• Infrastructure-as-Code pour reproductibilité</li></ul>

### 3.2. Périmètre

Le périmètre englobe l'écosystème complet de la plateforme Foosus, conçu pour supporter l'expansion internationale et l'innovation continue.

**Périmètre métier :** L'architecture couvre l'intégralité de la chaîne de valeur e-commerce, depuis l'onboarding des producteurs jusqu'à la livraison finale. Cette approche end-to-end inclut la gestion d'inventaire intelligent, les recommandations personnalisées par IA, l'optimisation logistique multi-modale, et le suivi de l'empreinte carbone automatisé.

**Périmètre utilisateurs :** La plateforme servira nativement une audience globale multi-culturelle incluant consommateurs finaux, producteurs/artisans locaux, équipes opérationnelles Foosus, partenaires logistiques, et écosystème de développeurs tiers. L'architecture supportera des profils d'usage différenciés avec personnalisation contextuelle par segment.

**Périmètre géographique :** Conception multi-région native avec data sovereignty compliance, permettant l'activation instantanée de nouveaux marchés. L'architecture supportera la localisation automatisée (langues, devises, réglementations) et l'intégration des systèmes de paiement régionaux via une approche plugin-based.

**Périmètre technologique :** Stack cloud-native moderne optimisée pour tous les canaux digitaux : applications natives iOS/Android, Progressive Web App (PWA), interfaces conversationnelles (chatbots/voice), API publiques pour partenaires, et tableaux de bord temps réel. L'architecture intégrera nativement IoT agricole, réalité augmentée, et blockchain pour traçabilité.

### 3.3. Parties prenantes, préoccupations et visions

L'écosystème des parties prenantes reflète l'ambition de transformation digitale de Foosus, avec des préoccupations alignées sur les enjeux de scalabilité internationale et d'innovation technologique.

Partie prenante	Préoccupation stratégique	Vision architecturale
<b>Ash CALLUM</b> (CEO)	Compétitivité internationale et croissance accélérée	Validation de la vision stratégique cloud-native pour leadership marché
<b>Natasha JARSON</b> (CIO)	Excellence technologique et gouvernance moderne	Approbation de l'architecture cloud-native et des choix platform engineering
<b>Daniel ANTHONY</b> (CPO)	Innovation produit et expérience utilisateur premium	Définition de la vision business orientée data-driven et personnalisation
<b>Jo KUMAR</b> (CFO)	Optimisation ROI et FinOps cloud	Approbation des investissements technologiques et modèle économique cloud
<b>Christina ORGEGA</b> (CMO)	Différenciation concurrentielle et expansion géographique	Validation du positionnement technologique et capacités marketing automation
<b>Pete PARKER</b> (Engineering Owner)	Architecture moderne et vitesse développement	Application de la vision technologique cloud-native et DevOps avancés
<b>Jack HARKNER</b> (Operations Lead)	Fiabilité opérationnelle et observabilité	Mise en œuvre de l'infrastructure cloud-native et monitoring intelligent
<b>Enterprise Architect Owner</b>	Cohérence architecturale et gouvernance technique	Définition des patterns cloud-native et standards d'architecture
<b>Squads techniques</b>	Productivité développement et qualité logicielle	Application des principes cloud-native et conformité à l'état de l'art actuel
<b>Équipes produit</b>	Time-to-market et feedback loops rapides	Exploitation des capacités d'expérimentation et A/B testing



## 3.4. Approche managériale

Foosus adopte une approche managériale hybride combinant les principes éprouvés du Lean et de l'Agile, enrichis des pratiques émergentes du Platform Engineering et du DevOps moderne.

### 3.4.1. L'approche Lean modernisée

Les principes Lean sont adaptés aux réalités du développement cloud-native 2025, intégrant les concepts de waste reduction dans le contexte des architectures distribuées :

- **Value Stream Optimization** : Élimination des goulots d'étranglement dans les pipelines CI/CD avec observabilité end-to-end des déploiements
- **Continuous Improvement** : Feedback loops automatisés via telemetry et métriques business en temps réel, permettant l'optimisation continue des performances
- **Quality at Source** : Shift-left security et testing, avec validation automatisée de la qualité à chaque étape du pipeline de développement
- **Flow Efficiency** : Réduction du lead time via automation intelligente et infrastructure-as-code pour éliminer les handoffs manuels

### 3.4.2. L'approche Agile orientée platform

L'agilité moderne transcende les frameworks traditionnels pour embrasser les principes du Platform Engineering et de l'autonomie des équipes :

- **Product Thinking** : Chaque composant de l'architecture est conçu comme un produit interne avec ses propres métriques de succès et feedback loops
- **Developer Experience Excellence** : Plateforme de développement auto-service permettant aux équipes de déployer et itérer de manière autonome
- **Continuous Delivery** : Déploiements fréquents et sécurisés via GitOps, avec capacité de rollback instantané et feature flags granulaires
- **Data-Driven Decisions** : Expérimentation A/B native et analytics en temps réel pour valider les hypothèses produit et orienter les développements

Cette approche managériale s'appuie sur les outils modernes de collaboration (Slack, Notion, Figma) et les pratiques de remote-first collaboration, optimisant la productivité des équipes distribuées internationales.

## 3.5. Procédures de changement de périmètre

Les évolutions de périmètre suivent un processus de gouvernance agile adapté aux réalités des architectures cloud-native évolutives, privilégiant la rapidité d'adaptation tout en maintenant la cohérence architecturale.

**Déclenchement et évaluation** : Tout changement significatif fait l'objet d'une architecture decision record (ADR) documentant le contexte, les options considérées, et les impacts anticipés. Cette approche RFC (Request for Comments) permet une évaluation collaborative et transparente.

**Validation bi-niveau** : Le comité de direction valide l'alignement stratégique et business, tandis que l'architecture review board évalue la faisabilité technique et l'impact sur l'architecture cible. Cette double validation garantit la cohérence des décisions.

**Documentation requisite** : Chaque demande de changement comprend une analyse d'impact multi-dimensionnelle (technique, financière, opérationnelle, sécurité, compliance) avec quantification des risques et définition des critères d'acceptation.

**Communication et mise en œuvre** : Les changements approuvés sont communiqués via les canaux Slack dédiés et les tableaux de bord centralisés. La mise à jour de la documentation technique s'effectue automatiquement via les pipelines GitOps, assurant la cohérence entre architecture documentée et implémentée.

**Suivi post-implémentation** : Chaque changement fait l'objet d'un monitoring spécifique avec métriques de performance et feedback utilisateur, permettant l'apprentissage continu et l'optimisation des processus de gouvernance.

## 4. Rôles et responsabilités

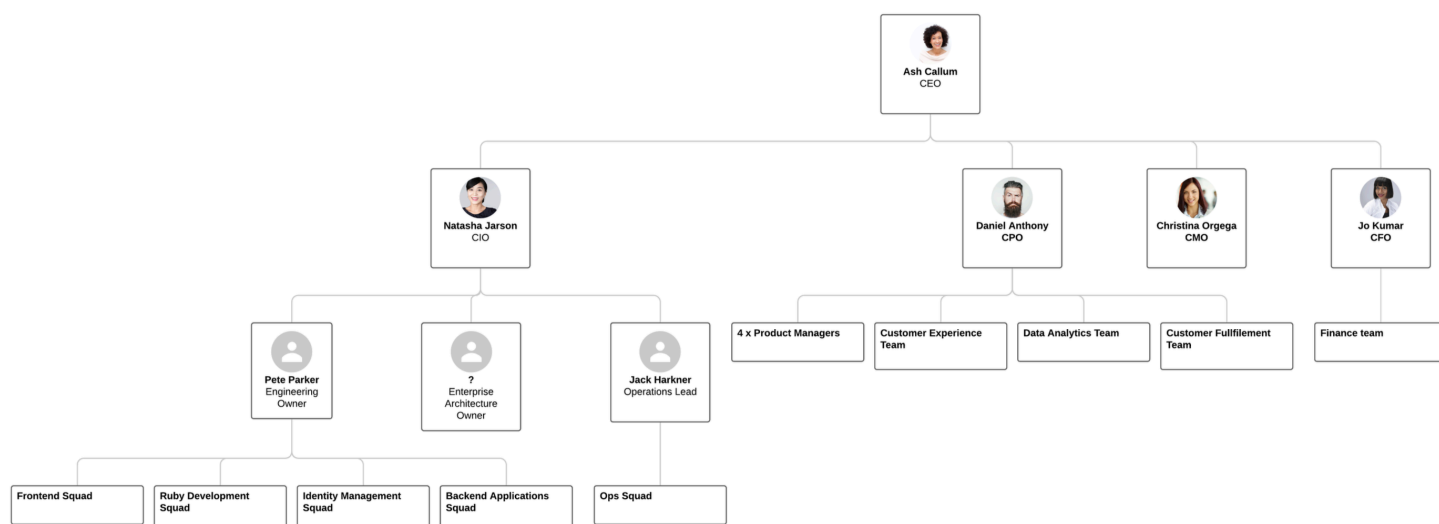
### 4.1. Structure de gouvernance

La transformation cloud-native de Foosus s'appuie sur une structure de gouvernance moderne, optimisée pour l'agilité et la prise de décision distribuée. Cette organisation reflète les meilleures pratiques en matière de platform engineering et de DevOps à l'échelle.

Partie prenante	Responsabilités architecturales
<b>Ash CALLUM</b> (CEO)	<ul style="list-style-type: none"><li>• Sponsor exécutif de la transformation cloud-native</li><li>• Validation des investissements technologiques stratégiques</li><li>• Arbitrage des orientations d'expansion géographique</li></ul>
<b>Natasha JARSON</b> (CIO)	<ul style="list-style-type: none"><li>• Définition de la stratégie technologique cloud-native</li><li>• Supervision de l'architecture d'entreprise moderne</li><li>• Gouvernance des choix de platform engineering</li></ul>
<b>Daniel ANTHONY</b> (CPO)	<ul style="list-style-type: none"><li>• Alignement produit avec les capacités techniques cloud</li><li>• Définition des exigences d'expérience utilisateur multi-canal</li><li>• Priorisation des fonctionnalités selon l'impact business</li></ul>
<b>Jo KUMAR</b> (CFO)	<ul style="list-style-type: none"><li>• Optimisation FinOps et gouvernance des coûts cloud</li><li>• Validation ROI des investissements technologiques</li><li>• Supervision des métriques de performance financière</li></ul>
<b>Christina ORGEGA</b> (CMO)	<ul style="list-style-type: none"><li>• Alignement marketing avec les capacités techniques globales</li><li>• Définition des exigences de géolocalisation et personnalisation</li><li>• Supervision des performances d'acquisition utilisateur</li></ul>
<b>Pete PARKER</b> (Engineering Owner)	<ul style="list-style-type: none"><li>• Implémentation de l'architecture cloud-native</li><li>• Coordination des équipes techniques cross-fonctionnelles</li><li>• Gouvernance des pratiques DevSecOps et CI/CD</li></ul>
<b>Jack HARKNER</b> (Operations Lead)	<ul style="list-style-type: none"><li>• Excellence opérationnelle cloud-native</li><li>• Supervision SRE et observabilité</li><li>• Gestion des déploiements multi-région</li></ul>

<b>Enterprise Architect Owner</b>	<ul style="list-style-type: none"> <li>• Définition des patterns et standards cloud-native</li> <li>• Gouvernance de la cohérence architecturale</li> <li>• Évaluation et adoption des technologies émergentes</li> </ul>
<b>Platform Engineering Teams</b>	<ul style="list-style-type: none"> <li>• Développement et maintenance de l'infrastructure as-code</li> <li>• Création des outils de développement auto-service</li> <li>• Optimisation continue de la developer experience</li> </ul>
<b>Product Teams</b>	<ul style="list-style-type: none"> <li>• Développement features selon les standards cloud-native</li> <li>• Implémentation des pratiques shift-left security</li> <li>• Participation à l'amélioration continue des processus</li> </ul>
<b>Data &amp; Analytics Teams</b>	<ul style="list-style-type: none"> <li>• Architecture data moderne (data mesh, real-time analytics)</li> <li>• Implémentation ML/AI pour personnalisation</li> <li>• Gouvernance des données et privacy by design</li> </ul>
<b>Customer Experience Teams</b>	<ul style="list-style-type: none"> <li>• Définition des exigences d'expérience omnicanale</li> <li>• Validation des parcours utilisateur multi-région</li> <li>• Optimisation des interfaces conversationnelles</li> </ul>

Figure 1 : Organigramme hiérarchique



## 4.2. Process du projet

### 4.2.1. Process

La transformation cloud-native de Foosus s'appuie sur un process de gestion de projet moderne, alliant rigueur méthodologique et agilité opérationnelle. Cette approche garantit la maîtrise des risques tout en préservant la capacité d'innovation.

#### A. LANCEMENT DU PROJET

L'initialisation de la transformation cloud-native repose sur une analyse exhaustive des besoins business et techniques, documentée dans la Requête de travail d'architecture. Cette phase établit les fondations de gouvernance moderne nécessaires à un projet d'envergure internationale.

**Repository cloud-native centralisé :** L'ensemble des livrables architecturaux est centralisé dans un repository GitOps sur GitHub/GitLab Enterprise, avec versioning automatique selon la norme

SemVer 2.0. Cette approche Infrastructure-as-Code garantit la traçabilité complète et l'accès transparent pour toutes les parties prenantes. Chaque Architecture Decision Record (ADR) est horodaté et signé numériquement.

**Validation de viabilité technique** : L'évaluation de faisabilité intègre les contraintes cloud-native spécifiques (multi-région, compliance, performance edge) via des proof-of-concepts automatisés. Les critères d'acceptation incluent la validation des SLAs de performance et des exigences de sécurité zero-trust.

## B. PLANIFICATION

La planification adopte une approche outcome-driven, privilégiant la valeur business plutôt que les outputs techniques. Cette méthodologie assure l'alignement stratégique tout en préservant l'agilité d'exécution.

**Objectifs OKR cloud-native** : Chaque objectif suit la méthodologie OKR (Objectives and Key Results) adaptée aux réalités cloud-native. Les Key Results incluent des métriques d'observabilité (SLIs/SLOs), de developer experience, et d'impact business. L'analyse des risques intègre les spécificités cloud (vendor lock-in, data sovereignty, resilience).

**Gouvernance distribuée** : Le Product Owner Architecture coordonne les tribes techniques (Platform Engineering, Security, SRE) selon les principes Spotify Model modernisés. Le Architecture Review Board valide les décisions techniques majeures via des ADR collaboratifs.

**Comité de pilotage exécutif** : Réunions bimensuelles avec dashboard temps réel présentant les métriques de progression (deployment frequency, lead time, MTTR) et l'alignement avec les objectifs business. Les décisions sont tracées via des tools collaboratifs (Notion, Confluence) avec notification automatique.

## C. RÉALISATION

L'exécution privilégie l'automation et l'observabilité continue, permettant une vélocité élevée tout en maintenant la qualité et la sécurité.

**Rituals agiles modernisés** : Sprint reviews bi-hebdomadaires avec démonstrations live des features déployées en production. Les retrospectives incluent l'analyse des métriques DORA (deployment frequency, lead time, MTTR, change failure rate) pour l'amélioration continue.

**Governance by automation** : Les standards architecturaux sont enforced via des politiques-as-code (Open Policy Agent, Conftest) dans les pipelines CI/CD. Les dérives architecturales sont détectées automatiquement via des fitness functions et remontées dans les dashboards de gouvernance.

## D. SUIVI

Le monitoring s'appuie sur une observabilité complète (traces, métriques, logs) avec alerting intelligent pour la détection proactive des anomalies et dérives.

**Observabilité de la transformation** : Dashboards Grafana/DataDog centralisés présentant les métriques techniques (infrastructure, application) et business (conversion, rétention, NPS). Les alertes automatisées déclenchent les procédures de remédiation selon les playbooks SRE.

**Health checks automatisés** : Validation continue de la conformité architecturale via des tests automatisés (architecture testing, chaos engineering). Les déviations déclenchent automatiquement des workflows de correction ou d'escalation selon leur criticité.

## E. CONTRÔLE DE FIN DE PROJET

La validation finale s'appuie sur des critères mesurables et des tests automatisés, garantissant la qualité avant la mise en production.

**Testing cloud-native** : Validation via des test suites automatisées incluant performance testing (load, chaos), security testing (SAST, DAST, penetration), et compliance testing (GDPR, SOC2). Les critères d'acceptation incluent les benchmarks de performance et les certifications de sécurité.

**Go/No-Go automatisé** : Décision de production basée sur des critères objectifs (code coverage > 85%, security scan passed, performance benchmarks atteints). Les rollback plans sont pré-validés via des blue-green deployments et feature flags.

## F. REQUÊTE DE CHANGEMENT

La gestion du changement s'appuie sur des processus légers et une documentation automatisée, maintenant l'agilité tout en assurant la traçabilité.

**Change management moderne** : Les changements suivent un workflow GitOps avec review collaborative via pull requests. Chaque modification majeure fait l'objet d'un RFC (Request for Comments) avec évaluation d'impact automatisée (blast radius, dependencies, rollback complexity).

**Communication transparente** : Notification automatique via Slack/Teams avec résumé exécutif des impacts. Les changements critiques déclenchent des war rooms virtuelles avec dashboard temps réel pour le suivi de déploiement. La documentation est mise à jour automatiquement via les pipelines CI/CD.

### 4.2.2. Indicateurs

#### A. INDICATEURS DE RÉUSSITE DU PROJET

Ces métriques déterminent l'atteinte des objectifs stratégiques de la transformation cloud-native et valident l'alignement avec les ambitions business de Foosus.

Thématique	Indicateur cible	Impact business attendu
<b>Performance globale</b>	<ul style="list-style-type: none"><li>API response time P95 &lt; 100ms</li><li>Multi-region availability 99.99%</li><li>Edge latency &lt; 50ms</li></ul>	Expérience utilisateur premium et compétitivité internationale
<b>Scalabilité intelligente</b>	<ul style="list-style-type: none"><li>Auto-scaling efficiency &gt; 95%</li><li>Traffic spike absorption 10x</li><li>Zero-downtime deployments 100%</li></ul>	Absorption des pics et croissance sans friction
<b>Sécurité moderne</b>	<ul style="list-style-type: none"><li>Zero-trust compliance score &gt; 95%</li><li>Security vulnerabilities MTTR &lt; 4h</li><li>Data sovereignty compliance 100%</li></ul>	Confiance utilisateur et conformité réglementaire globale
<b>Agilité développement</b>	<ul style="list-style-type: none"><li>Lead time feature-to-production &lt; 2h</li><li>Developer satisfaction score &gt; 4.5/5</li><li>Time-to-market nouvelles régions &lt; 24h</li></ul>	Innovation accélérée et expansion géographique rapide
<b>Excellence opérationnelle</b>	<ul style="list-style-type: none"><li>Mean Time to Recovery &lt; 15min</li><li>Infrastructure cost optimization 15%/an</li><li>Carbon footprint reduction 20%/an</li></ul>	Résilience opérationnelle et responsabilité environnementale

#### B. INDICATEURS DE SUIVI DU PROJET

Ces métriques permettent le pilotage opérationnel de la transformation et l'identification proactive des risques et blocages.

Thématique	Indicateur de suivi
<b>Progression architecture (TOGAF)</b>	<ul style="list-style-type: none"> <li>• Nombre de livrables ADM planifiés / réalisés</li> <li>• Phases TOGAF complétées / phases totales</li> <li>• Architecture Decision Records validés / en attente</li> <li>• Fitness functions implémentées / définies</li> </ul>
<b>Vélocité développement</b>	<ul style="list-style-type: none"> <li>• Story points delivered vs planned (sprint burndown)</li> <li>• Features cloud-native déployées / planifiées</li> <li>• Pipeline CI/CD automatisés / manuels</li> <li>• Technical debt reduction rate</li> </ul>
<b>Qualité &amp; sécurité</b>	<ul style="list-style-type: none"> <li>• Bugs critiques détectés / résolus</li> <li>• Security scans passed / total</li> <li>• Test coverage evolution (target &gt; 85%)</li> <li>• Infrastructure drift detection / remediation</li> </ul>
<b>Gouvernance financière</b>	<ul style="list-style-type: none"> <li>• Budget cloud optimisé / alloué par service</li> <li>• FinOps forecast accuracy vs actual</li> <li>• Cost per transaction trend</li> <li>• ROI milestone achievement rate</li> </ul>
<b>Adoption &amp; formation</b>	<ul style="list-style-type: none"> <li>• Teams onboarded sur platform / total teams</li> <li>• Developer certification completion rate</li> <li>• Platform self-service adoption rate</li> <li>• Knowledge transfer sessions completed</li> </ul>

### 4.3. Rôles et responsabilités

La matrice RACI modernisée reflète la distribution des responsabilités dans un environnement cloud-native, privilégiant l'autonomie des équipes et la transparence décisionnelle.

ID	Activités cloud-native	C E O	C I O	C P O	C F O	Engineering Owner	Ops Lead	Architect Owner	Platform Teams	Product Teams
<b>Phase 1 : Lancement du projet</b>										
P101	Définition des besoins cloud-native	A	R	C	C	C	I	C	I	I
P102	Étude de viabilité technique & business	A	C	C	R	C	C	C	I	I
<b>Phase 2 : Planification</b>										
P201	Définition des objectifs d'architecture	A	C	C	I	C	C	R	C	I
P202	Analyse des risques cloud & sécurité	I	C	I	C	C	C	R	C	I

P203	Définition des moyens (stack tech, équipes)	I	A	C	C	R		C	C	C	I
P204	Définition des KPI cloud-native	C	C	C	A	C		C	R	C	C
<b>Phase 3 : Réalisation</b>											
P301	Définition de l'architecture cloud-native	I	C	C	I	C		C	R	C	C
P302	Développement de la plateforme	I	I	I	I	A		C	C	R	R
P303	Contrôle des KPI de suivi	I	C	C	C	A		C	C	C	C
<b>Phase 4 : Contrôle</b>											
P401	Recettage & validation qualité	I	I	A	I	C		C	C	C	R
P402	Contrôle des critères d'acceptation	I	C	A	C	C		C	C	C	R
<b>Phase 5 : Production</b>											
P501	Mise en production multi-région	I	C	I	I	C		A	C	R	C
P502	Contrôle des KPI de réalisation	A	C	C	C	C		C	C	C	C

#### Légende RACI :

- **R (Responsible)** : Exécute la tâche et assume la responsabilité de sa réalisation
- **A (Accountable)** : Rend compte du résultat et valide la complétion
- **C (Consulted)** : Consulté avant la décision, apporte son expertise
- **I (Informed)** : Informé après la décision ou l'action

Cette structure de gouvernance moderne garantit l'agilité nécessaire à l'innovation tout en maintenant les standards de qualité et de sécurité requis pour une plateforme internationale de classe mondiale.

## 5. Approche architecturale

### 5.1. Principes architecturaux

#### 5.1.1. Une architecture priorisant l'utilisateur

La conception de l'architecture cloud-native de Foosus, son développement et ses évolutions futures s'articulent autour des besoins exprimés par l'ensemble des parties prenantes, en plaçant l'expérience

utilisateur au cœur des décisions techniques. Cette approche user-centric garantit l'alignement entre innovations technologiques et valeur business délivrée.

L'excellence de l'expérience utilisateur s'appuie sur une infrastructure edge-native distribuée mondialement, permettant des latences sub-100ms et une personnalisation contextuelle en temps réel. L'approche managériale mise en œuvre sur le projet et sa maintenance s'inspire des principes Lean modernisés, enrichis des pratiques Platform Engineering pour optimiser la developer experience et accélérer la vélocité de livraison.

La compréhension de l'ensemble des parties prenantes et la maintenabilité évolutive de la plateforme reposent sur l'adoption de patterns cloud-native standards et l'application rigoureuse des principes d'architecture simple et cohérente. Cette simplicité architecturale masque la complexité opérationnelle tout en préservant la flexibilité d'évolution.

### **5.1.2. Une architecture à l'implémentation itérative**

La transformation cloud-native sera déployée par itérations successives autour de cycles courts alignés sur l'approche managériale agile, avec pour première évolution majeure la mise en place d'un système de géolocalisation intelligente exploitant l'edge computing et les capacités de machine learning distribuées.

Cette approche itérative intègre nativement la cohabitation entre l'architecture legacy et la nouvelle plateforme cloud-native. Les étapes de migration suivront une stratégie de transformation progressive basée sur les patterns strangler fig et blue-green deployment, documentée dans le plan de migration architectural. L'architecture supportera les déploiements parallèles avec routage intelligent du trafic.

La gestion de ces cycles courts s'appuie sur une infrastructure GitOps moderne avec pipelines CI/CD automatisés, permettant le déploiement sécurisé et traçable des releases. L'intégration de feature flags granulaires et de capacités de rollback instantané garantit la stabilité opérationnelle tout en maximisant la fréquence de livraison.

### **5.1.3. Une architecture permettant la cohabitation entre deux solutions**

La migration entre l'architecture existante et la nouvelle solution cloud-native est conçue pour garantir une continuité de service absolue pour les utilisateurs. La stratégie de transformation adopte une approche zero-downtime s'appuyant sur des patterns de migration éprouvés et des mécanismes de basculement automatique.

La nouvelle architecture intègre dès sa conception les contraintes de cohabitation, que ce soit dans la priorisation des évolutions fonctionnelles et techniques ou dans l'orchestration des déploiements multi-environnements. L'architecture API-first facilite l'interopérabilité entre anciens et nouveaux services pendant la période de transition.

Le plan de migration architectural définit précisément les étapes de transition, les critères de basculement, et les procédures de rollback pour chaque composant. Cette approche méthodique minimise les risques opérationnels tout en permettant une transformation progressive et contrôlée vers l'architecture cible.

## **5.2. Process d'architecture**



La méthode ADM constitue le framework méthodologique de référence pour la transformation cloud-native de Foosus. Adaptée aux réalités des architectures distribuées et des pratiques DevSecOps, cette approche garantit la rigueur méthodologique tout en préservant l'agilité d'exécution.

Toutes les phases TOGAF ne présentent pas la même pertinence pour la transformation cloud-native spécifique de Foosus. Le tableau ci-dessous décrit l'utilisation modernisée de l'ADM pour ce projet, intégrant les spécificités du développement cloud-native, des pratiques GitOps, et de la gouvernance automatisée.

Phase	Entrée	Version	Sortie	Version
<b>P</b>	Architecture Repository	V.0	Architecture Repository	V.P
			Request for Architecture Work (including Cloud-Native Business Principles, Goals and Drivers)	V.P
<b>A</b>	Architecture Repository	V.P	Architecture Vision	V.A
	Request for Architecture Work	V.P	Statement of Architecture Work including Communication Plan et Cloud Architecture RoadMap	V.A
	Architecture Building Blocks	V.0	Architecture Definition Document	V.A
<b>B</b>	Architecture Building Blocks	V.0	Architecture Definition Document	V.B
	Architecture Repository	V.P	Architecture Requirement Specification	V.B
	Architecture Vision	V.A	Statement of Architecture Work	V.B
	Statement of Architecture Work	V.A		
	Architecture Definition Document	V.A		
<b>C</b>	Architecture Building Blocks	V.0	Architecture Definition Document	V.C
	Architecture Repository	V.P	Architecture Requirement Specification	V.C
	Architecture Vision	V.A	Statement of Architecture Work	V.C
	Architecture Definition Document	V.B		
	Architecture Requirement Specification	V.B		
	Statement of Architecture Work	V.B		
<b>D</b>	Architecture Building Blocks	V.0	Architecture Definition Document	V.D
	Architecture Repository	V.P	Architecture Requirement Specification	V.D
	Architecture Vision	V.A	Statement of Architecture Work	V.D
	Architecture Definition Document	V.C		
	Architecture Requirement Specification	V.C		

	Statement of Architecture Work	V.C		
<b>E</b>	Architecture Building Blocks	V.0	Architecture Vision	V.E
	Architecture Repository	V.P	Architecture Definition Document	V.E
	Architecture Vision	V.A	Architecture Requirement Specification	V.E
	Architecture Definition Document	V.D	Statement of Architecture Work	V.E
	Architecture Requirement Specification	V.D	Implementation and Migration Plan	V.E
	Statement of Architecture Work	V.D		
<b>F</b>	Architecture Repository	V.P	Architecture Requirement Specification	V.F
	Architecture Requirement Specification	V.E	Architecture Vision	V.E
	Implementation and Migration Plan	V.E	Implementation and Migration Plan (including Implementation Governance Model)	V.F
	Architecture Definition Document	V.E	Architecture Definition Document	V.F
	Architecture Requirement Specification	V.E	Request for Architecture Work	V.F
	Statement of Architecture Work	V.E	Statement of Architecture Work	V.F
	Implementation and Migration Plan	V.E		
<b>G</b>	Architecture Repository	V.P	Architecture Contract	V.G
	Architecture Requirement Specification	V.F	Compliance Assessment	V.G
	Implementation and Migration Plan	V.F	Statement of Architecture Work	V.G
	Architecture Definition Document	V.F		
	Request for Architecture Work	V.F		
	Statement of Architecture Work	V.F		
	Architecture Vision	V.E		
	Architecture Contract	V.0		
<b>H</b>	Architecture Repository	V.P	Change Request	V.H
	Architecture Definition Document	V.F	Request for Architecture Work	V.H
	Implementation and Migration Plan	V.F	Statement of Architecture Work	V.H
	Architecture Requirement Specification	V.F		

	Architecture Vision	V.E		
	Architecture Contract	V.G		
	Compliance Assessment	V.G		
	Statement of Architecture Work	V.G		
N	Requirement Impact Assessment	V.O	Requirement Impact Assessment	V.X
	Architecture Repository	V.P	Architecture Requirement Specification	V.X
	Architecture Requirement Specification	V.X	Architecture Vision	V.X
			Statement of Architecture Work	V.X

Cette approche TOGAF modernisée intègre les spécificités cloud-native à chaque phase : Architecture Decision Records (ADR) automatisés, validation continue par fitness functions, et gouvernance par politiques-as-code. Les livrables traditionnels sont enrichis des artefacts DevSecOps et Platform Engineering nécessaires à une transformation réussie.

### 5.3. Contenu de l'architecture

Le cadre de contenu d'architecture TOGAF (ACF - Architecture Content Framework) modernisé fournit une catégorisation cloud-native des meilleures pratiques pour le contenu architectural. Adapté aux réalités 2025, tous les éléments ne sont pas également pertinents pour chaque projet. Le tableau ci-dessous décrit les zones de contenu pertinentes pour la transformation cloud-native de Foosus.

Les artefacts présentés intègrent les paradigmes cloud-native, DevSecOps, et Platform Engineering comme éléments indispensables de la démarche moderne. D'autres artefacts pourront être ajoutés au fil de la transformation pour compléter ou souligner des éléments critiques de l'architecture distribuée.

Zone de contenu	Artefacts cloud-native modernisés
<b>Principes, vision, et conditions requises de l'Architecture</b>	<ul style="list-style-type: none"> <li>• Cloud-Native Stakeholder Map Matrix</li> <li>• Digital Value Stream Map</li> <li>• Platform Engineering Charter</li> <li>• Zero Trust Security Principles</li> <li>• Developer Experience Requirements</li> </ul>
<b>Architecture business</b>	<ul style="list-style-type: none"> <li>• Digital Organization/Actor Catalog</li> <li>• Cloud Economics Driver/Goal Catalog</li> <li>• DevOps Role Catalog</li> <li>• Cloud-Native Business Service Catalog</li> <li>• Multi-Region Location Catalog</li> <li>• Event-Driven Process Catalog</li> <li>• SLA/SLO Contract Catalog</li> <li>• Digital Value Stream Stages</li> <li>• Cross-Functional Team Matrix</li> <li>• API-First Business Service Diagram</li> <li>• Cloud Journey Lifecycle Diagram</li> <li>• Cloud-Native Business Use-Case</li> <li>• Automated Value Stream Map</li> </ul>

<b>Architecture des systèmes d'information — Données</b>	<ul style="list-style-type: none"> <li>• Data Mesh Entity/Component Catalog</li> <li>• Event Sourcing Conceptual Data Model</li> <li>• CQRS Logical Data Diagram</li> <li>• Zero Trust Data Security Architecture</li> <li>• Cloud-Native Data Migration Strategy</li> <li>• Real-time Data Pipeline Architecture</li> <li>• Data Sovereignty Compliance Matrix</li> </ul>
<b>Architecture des systèmes d'information — Applications</b>	<ul style="list-style-type: none"> <li>• Microservices/Function Matrix</li> <li>• Cloud-Native Application Portfolio</li> <li>• API-First Use-Case Diagram</li> <li>• Container/Serverless Engineering Diagram</li> <li>• GitOps Application Migration Plan</li> <li>• Edge Computing Distribution Diagram</li> <li>• Service Mesh Communication Architecture</li> </ul>
<b>Architecture technologique</b>	<ul style="list-style-type: none"> <li>• Cloud-Native Technology Portfolio</li> <li>• Kubernetes Platform Decomposition</li> <li>• Serverless Processing Architecture</li> <li>• Edge Computing Infrastructure Map</li> <li>• CI/CD Pipeline Technology Stack</li> <li>• Observability Technology Framework</li> <li>• Infrastructure as Code Templates</li> </ul>
<b>Réalisation de l'architecture</b>	<ul style="list-style-type: none"> <li>• Cloud Migration Context Diagram</li> <li>• Platform Engineering Benefits Map</li> <li>• Cloud-Native Requirements Catalog</li> <li>• DevSecOps Compliance Framework</li> <li>• FinOps Cost Optimization Plan</li> <li>• SRE Runbook Repository</li> <li>• Architecture Decision Records (ADR)</li> </ul>

## Évolution des artefacts TOGAF pour le cloud-native

**Modernisation des catalogues traditionnels :** Les catalogues TOGAF classiques sont enrichis pour capturer les spécificités cloud-native : microservices, APIs, événements, et infrastructure distribuée. Chaque artefact intègre les dimensions de sécurité, performance, et conformité dès la conception.

**Integration DevSecOps native :** Les artefacts incluent systématiquement les aspects de sécurité (Zero Trust), d'automatisation (GitOps), et d'observabilité (telemetry) pour garantir une architecture production-ready dès les phases de design.

**Architecture as Code :** L'ensemble des artefacts est versionné et maintenu sous forme de code (YAML, JSON, HCL) dans des repositories GitOps, permettant la traçabilité, la collaboration, et l'automation de la documentation architecturale.

Cette approche modernisée du framework TOGAF garantit à Foosus une gouvernance architecturale rigoureuse tout en embrassant les paradigmes cloud-native nécessaires à sa transformation digitale et son expansion internationale.

## 6. Plan de travail

## 6.1. Plan d'architecture

Cette section recense l'intégralité des activités et livrables architecturaux requis pour la transformation de Foosus, en précisant l'implication des équipes design et développement dans chaque étape. L'approche peut s'adapter aux insights révélés lors du déroulement de chaque phase.

L'architecte d'entreprise pilote l'ensemble du processus, assumant la responsabilité du bon déroulement de la définition architecturale et de sa cohérence globale.

### 6.1.1. Phase préliminaire

#### A. ACTIVITÉS

La phase préliminaire établit les fondations du projet en définissant clairement les attentes et objectifs de la nouvelle architecture, tout en structurant l'initialisation du processus de transformation.

#### B. LIVRABLES

##### Architecture Repository

Créé sur infrastructure Git et accessible via GitHub, ce repository centralise et versionne l'ensemble du travail de définition architecturale tout au long du processus de développement.

- L'architecture repository garantit l'accessibilité aux équipes de développement pour consultation des documents de travail architectural les concernant

##### Request for Architecture Work

Document initié par l'organisation sponsor pour déclencher le cycle de développement architectural. Il confirme le contexte organisationnel et les objectifs projet, incluant les spécifications relatives aux principes business et aux objectifs stratégiques de Foosus.

- Produit en collaboration avec la CIO qui valide la viabilité technique du projet

### 6.1.2. Phase A – Vision de l'architecture

#### A. ACTIVITÉS

La vision architecturale fournit une perspective stratégique sur les transformations attendues grâce au déploiement de l'architecture cible, en délimitant le périmètre et en définissant les moyens nécessaires pour l'atteindre.

#### B. LIVRABLES

##### Architecture Vision

Document synthétisant les changements organisationnels induits par le déploiement de l'architecture cible et définissant les résultats attendus pour l'ensemble des parties prenantes. Une matrice des parties prenantes (Stakeholder Map Matrix) identifie ces acteurs et leur niveau d'engagement. Une Value Stream Map cartographie la chaîne de fonctionnement produit et les étapes de création de valeur pour se concentrer sur les éléments critiques au bon fonctionnement organisationnel.

## Statement of Architecture Work

Cette déclaration formalise l'accord sur le périmètre du travail architectural et structure la production de l'architecture cible via un plan de communication et une roadmap détaillée. Ce livrable évolue à chaque phase importante pour intégrer les évolutions de la définition architecturale ou du contexte organisationnel.

## Architecture Definition Document

La définition architecturale rassemble les principaux artefacts architecturaux développés durant le projet et leurs informations associées. À ce stade, elle comprend la définition du périmètre projet, la description des objectifs et l'identification des contraintes initiales.

### 6.1.3. Phase B – Architecture Business

#### A. ACTIVITÉS

La phase d'architecture business développe une définition complète de l'architecture métier cible et spécifie l'ensemble des exigences que l'implémentation doit respecter pour s'y conformer.

#### B. LIVRABLES

##### Architecture Definition Document

Cette phase enrichit la définition architecturale en établissant les principes et le modèle d'architecture business cible. Elle permet de conduire l'analyse des écarts avec l'architecture business actuelle et de proposer une vision business initiale des étapes de transition nécessaires. La production de ce document s'appuie sur :

L'analyse du produit organisationnel et de son cycle de vie via les artefacts suivants :

- **Driver / Goal / Objective Catalog**, pour identifier les objectifs stratégiques de l'entreprise
- **Business Service / Function Catalog**, pour cartographier les capacités fonctionnelles actuelles et souhaitées
- **Process / Event / Control / Product Catalog**, pour identifier et hiérarchiser les processus, leurs événements, livrables et éléments de contrôle
- **Contract / Measure Catalog**, pour identifier les engagements de service de l'entreprise
- **Value Stream Stages Catalog**, pour identifier les étapes de création de valeur vers le produit final utilisateur
- **Business Service / Information Diagram**, pour identifier les informations, données et sources nécessaires aux services métiers et visualiser les dépendances fonctionnelles
- **Product Lifecycle Diagram**, pour identifier le cycle de vie produit et assurer les contrôles et processus architecturaux nécessaires
- **Business Use-Case Diagram**, pour identifier les relations clients-services métiers et illustrer l'utilisation des capacités fonctionnelles
- **Value Stream Map**, pour identifier la chaîne de fonctionnement produit et les étapes de création de valeur

La compréhension des interactions parties prenantes via ces artefacts :

- **Organization / Actor Catalog**, pour identifier les participants aux interactions avec les équipes informatiques
- **Role Catalog**, pour identifier les niveaux d'autorisation disponibles

- **Actor / Role Matrix**, pour optimiser l'analyse des besoins formatifs, paramètres de sécurité utilisateur et anticiper les évolutions organisationnelles et managériales

La délimitation du périmètre au-delà des parties prenantes grâce aux artefacts :

- **Location Catalog**, pour identifier les lieux d'activité organisationnelle et éléments architecturaux associés, notamment la présence de Data Centers

### **Architecture Requirement Specification**

Document formalisant l'ensemble des éléments quantitatifs que l'implémentation projet doit respecter pour se conformer à l'architecture business, énumérant des critères mesurables à satisfaire durant l'implémentation architecturale.

Sa réalisation s'appuie sur les mêmes artefacts énumérés pour la définition architecturale, qui propose une vision qualitative de l'architecture business.

### **Statement of Architecture Work**

La déclaration de travail architectural est mise à jour selon les éléments révélés pendant la phase. L'architecture roadmap peut être enrichie d'une vision business du déroulement des étapes vers l'architecture cible.

## **6.1.4. Phase C – Architecture des systèmes d'information**

### **A. ACTIVITÉS**

La phase d'architecture SI développe une définition complète de l'architecture des systèmes d'information cible et spécifie l'ensemble des exigences que l'implémentation doit respecter pour s'y conformer.

### **B. LIVRABLES**

#### **Architecture Definition Document**

Cette phase enrichit la définition architecturale en établissant les principes et le modèle d'architecture SI cible. Elle permet de conduire l'analyse des écarts avec l'architecture SI actuelle et de proposer une vision SI initiale des étapes de transition nécessaires. La production de ce document s'appuie sur les artefacts suivants :

- **Data Entity / Data Component Catalog**, pour identifier l'inventaire complet des données organisationnelles, leur usage et leur gouvernance
- **Conceptual Data Diagram**, pour identifier les relations entre entités de données critiques à destination des parties prenantes métier
- **Logical Data Diagram**, pour identifier les relations logiques entre entités de données organisationnelles
- **Data Security Diagram**, pour identifier les autorisations d'accès aux données et les règles applicables en matière de gouvernance des données
- **Data Migration Diagram**, pour identifier le parcours des données depuis leur source vers leur destination et les transformations requises
- **Application / Function Matrix**, pour identifier les relations entre applications et fonctions métier organisationnelles
- **Application Use-Case Diagram**, pour identifier les relations entre consommateurs et fournisseurs de services applicatifs en illustrant leur utilisation fonctionnelle

- **Software Engineering Diagram**, pour identifier l'architecture logicielle et les relations entre composants logiciels
- **Application Migration Diagram**, pour identifier la stratégie de migration applicative vers l'architecture cible
- **Software Distribution Diagram**, pour identifier la distribution des composants logiciels dans l'infrastructure

### **Architecture Requirement Specification**

Document formalisant l'ensemble des éléments quantitatifs que l'implémentation projet doit respecter pour se conformer à l'architecture SI, énumérant des critères mesurables à satisfaire durant l'implémentation architecturale.

### **Statement of Architecture Work**

La déclaration de travail architectural est mise à jour selon les éléments révélés pendant la phase. L'architecture roadmap peut être enrichie d'une vision SI du déroulement des étapes vers l'architecture cible.

## **6.1.5. Phase D – Architecture technologique**

### **A. ACTIVITÉS**

La phase d'architecture technologique développe une définition complète de l'architecture technologique cible et spécifie l'ensemble des exigences que l'implémentation doit respecter pour s'y conformer.

### **B. LIVRABLES**

#### **Architecture Definition Document**

Cette phase enrichit la définition architecturale en établissant les principes et le modèle d'architecture technologique cible. Elle permet de conduire l'analyse des écarts avec l'architecture technologique actuelle et de proposer une vision technologique initiale des étapes de transition nécessaires.

#### **Architecture Requirement Specification**

Document formalisant l'ensemble des éléments quantitatifs que l'implémentation projet doit respecter pour se conformer à l'architecture technologique, énumérant des critères mesurables à satisfaire durant l'implémentation architecturale.

#### **Statement of Architecture Work**

La déclaration de travail architectural est mise à jour selon les éléments révélés pendant la phase. L'architecture roadmap peut être enrichie d'une vision technologique du déroulement des étapes vers l'architecture cible.

## **6.1.6. Phase E – Opportunités et solutions**

### **A. ACTIVITÉS**



Cette phase identifie et concrétise les opportunités et solutions générées par l'architecture cible définie précédemment, en vue de leur matérialisation et de l'ajustement éventuel des livrables antérieurs.

## **B. LIVRABLES**

### **Architecture Definition Document**

La définition architecturale intègre l'analyse d'impact et ses conclusions en complément des éléments précédemment évalués, considérant la création des artefacts suivants :

- **Project Context Diagram**, pour identifier les organisations, fonctions, services, processus, applications, données et technologies à ajouter, supprimer ou impacter par le projet
- **Benefits Diagram**, pour illustrer les opportunités identifiées par la définition architecturale et les hiérarchiser selon leur ampleur, bénéfices et complexité

### **Architecture Vision**

L'Architecture Vision est révisée pour synthétiser l'ensemble de la définition architecturale désormais achevée et préciser les impacts issus de la modification architecturale, notamment les bénéfices attendus.

### **Architecture Requirement Specification**

Document formalisant l'ensemble des éléments quantitatifs que l'implémentation projet doit respecter pour se conformer à l'architecture définie et aux opportunités identifiées pendant cette phase, énumérant des critères mesurables à satisfaire durant l'implémentation architecturale.

Sa réalisation s'appuie sur les mêmes artefacts énumérés pour la définition architecturale, qui propose une vision qualitative de l'architecture et des opportunités associées.

### **Statement of Architecture Work**

La déclaration de travail architectural définit le périmètre et l'approche utilisés pour compléter le cycle de développement architectural, ainsi que le plan de communication et le calendrier associé.

Cette phase permet d'intégrer les opportunités identifiées et de mettre à jour les éléments révélés pendant la définition architecturale.

### **Implementation and Migration Plan**

Document focalisé sur l'implémentation et le déploiement par étapes de l'architecture cible. Il présente le calendrier détaillé projet avec l'exposition des risques, solutions et opportunités à chaque étape de migration vers l'architecture cible.

## **6.1.7. Phase F – Planning de Migration**

### **A. ACTIVITÉS**

Basée sur le travail des phases précédentes et les décisions prises pour l'architecture cible, cette phase définit les étapes de transition architecturale et l'ensemble des éléments encadrant cette transformation.

## **B. LIVRABLES**

## **Architecture Definition Document**

La définition architecturale finalise les artefacts relatifs à la transition architecturale en s'appuyant sur la vision d'ensemble développée lors des phases précédentes.

## **Architecture Requirement Specification**

Document formalisant l'ensemble des éléments quantitatifs que l'implémentation projet doit respecter pour se conformer à l'architecture définie et aux étapes de migration spécifiées pendant cette phase, énumérant des critères mesurables à satisfaire durant l'implémentation architecturale.

## **Statement of Architecture Work**

La déclaration de travail architectural définit le périmètre et l'approche utilisés pour compléter le cycle de développement architectural, ainsi que le plan de communication et le calendrier associé.

Cette phase permet d'intégrer la définition des étapes de migration architecturale, notamment le calendrier associé à son implémentation (Architecture RoadMap).

## **Request for Architecture Work**

Document produit par l'organisation sponsor pour déclencher le cycle de développement architectural. Il confirme le contexte et les objectifs projet, incluant les spécifications relatives aux principes business et aux objectifs généraux de Foosus.

Pendant cette phase, ce document valide la vision finale développée pendant le travail architectural, notamment auprès des sponsors, pour lancer le cycle de migration architecturale.

## **Implementation and Migration Plan**

Document focalisé sur l'implémentation et le déploiement par étapes de l'architecture cible. Il présente le calendrier détaillé projet avec l'exposition des risques, solutions et opportunités à chaque étape de migration vers l'architecture cible.

Pendant cette phase, outre la révision du planning d'implémentation, la planification de la gouvernance de l'architecture de transition pendant l'implémentation est cruciale (Implementation Governance Model).

## **6.1.8. Phase G – Gouvernance de l'implémentation**

### **A. ACTIVITÉS**

Cette phase assure que le projet d'implémentation respecte l'architecture définie et que toute modification reste sous contrôle architectural. C'est durant cette phase que la mise en œuvre de l'architecture définie est réalisée.

### **B. LIVRABLES**

#### **Architecture Contract**

Le contrat architectural spécifie les accords entre développement et architecture pour garantir que l'implémentation respecte les exigences architecturales.

#### **Compliance Assessment**

L'évaluation de conformité vérifie que l'implémentation respecte l'architecture définie et identifie les déviations nécessitant une action corrective.

#### **Statement of Architecture Work**

La déclaration de travail peut être mise à jour selon les éléments révélés pendant la mise en œuvre de l'architecture définie.

### **6.1.9. Phase H – Management du changement d'architecture**

#### **A. ACTIVITÉS**

Cette phase garantit la gestion appropriée des changements de l'architecture de base. Elle comprend l'évaluation continue de l'architecture existante et la recommandation des changements nécessaires.

#### **B. LIVRABLES**

##### **Change Request**

Les demandes de changement documentent les modifications architecturales nécessaires et assurent le suivi de leur implémentation.

##### **Request for Architecture Work**

Ce document peut être mis à jour pour refléter les nouveaux besoins identifiés pendant la phase de gestion du changement.

##### **Statement of Architecture Work**

La déclaration de travail peut être mise à jour pour refléter les modifications apportées à l'architecture.

### **6.1.10. Phase N – Management des conditions requises**

#### **A. ACTIVITÉS**

Cette phase gère les exigences tout au long du cycle ADM, garantissant que les exigences sont identifiées, stockées et gérées efficacement pendant et après le processus de développement architectural.

#### **B. LIVRABLES**

##### **Requirement Impact Assessment**

Cette évaluation analyse l'impact des nouvelles exigences sur l'architecture existante.

##### **Architecture Requirement Specification**

Document maintenant les spécifications des exigences architecturales à jour tout au long du cycle de développement.

## **6.2. Plan de communication**

### 6.2.1. Création ou mise à jour d'un document ou d'un artefact

Les équipes informatiques peuvent être impliquées selon le document finalisé ou mis à jour.

**Public cible :** Parties prenantes concernées selon le sujet - Voir liste de distribution en en-tête des documents ou le RACI de la déclaration de travail

**Modalités :** • Diffusion sur le repository GitHub à la finalisation ou validation du document • Envoi par mail aux parties prenantes concernées. En cas de mise à jour, préciser les changements apportés

**Fréquence/Durée :** Dès la finalisation / validation du document

**Auteur :** Auteur du document

### 6.2.2. Réunion du groupe projet

Le groupe projet intègre notamment l'Engineer Owner et l'Operation Lead. D'autres membres des équipes informatiques peuvent être impliqués selon le sujet traité.

**Public cible :** Parties prenantes concernées selon le sujet - Voir liste de distribution en en-tête des documents ou le RACI de la déclaration de travail

**Modalités :** • Réunion de travail en présentiel ou distanciel • Revue des avancées, problèmes, besoins, objectifs de manière détaillée

**Fréquence/Durée :** 1 fois par semaine - Minimum 2 heures, peut prévoir plus sur certains sujets spécifiques

**Auteur :** Chef du projet

### 6.2.3. Réunion des sponsors

La CIO sera conviée à cette réunion en tant que membre du CODIR.

**Public cible :** Sponsors et membres du CODIR

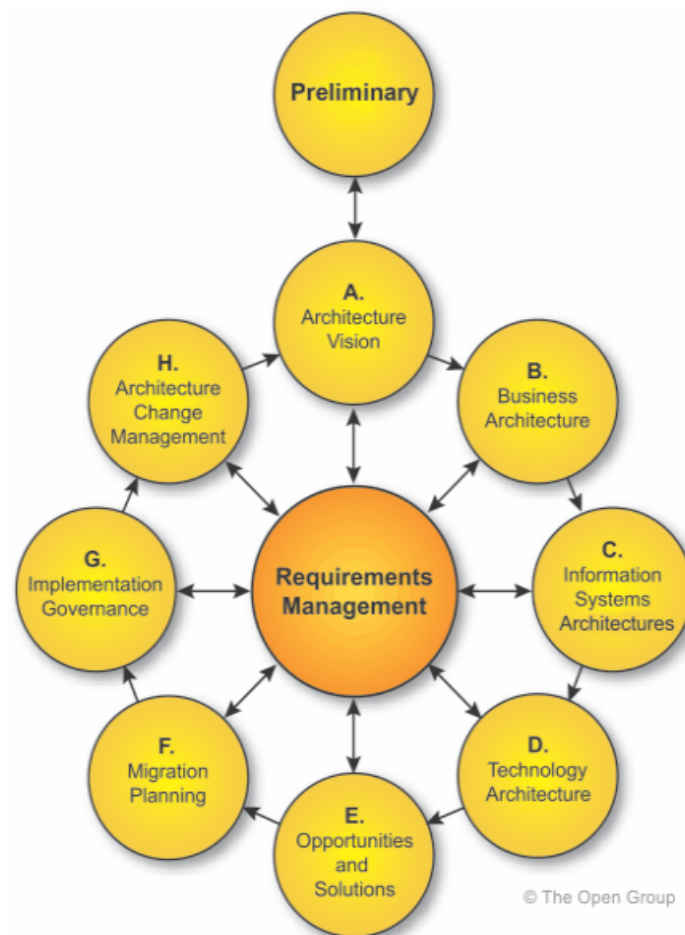
**Modalités :** • Réunion de travail en présentiel ou distanciel • Revue des avancées, problèmes, besoins, objectifs de manière synthétique • Validation finale des éléments nécessitant approbation

**Fréquence/Durée :** 1 fois par semaine - 30 minutes, sauf sujet particulier à traiter

**Auteur :** Chef du projet

## 6.3. Collaboration

La planification du projet s'appuiera sur le cycle ADM tel que défini par TOGAF. Il conviendra d'apporter des précisions complémentaires à ce cycle ultérieurement.



## 7. Risques et facteurs de réduction

### 7.1. Analyse des risques

La transformation cloud-native de Foosus présente des défis spécifiques nécessitant une approche de gestion des risques adaptée aux réalités des architectures distribuées modernes. L'analyse suivante identifie les risques stratégiques majeurs et leurs facteurs de mitigation.

#### MATRICE DES RISQUES STRATÉGIQUES

Catégorie de risque	Probabilité	Impact Business	Criticité	Actions de mitigation
<b>Transformation organisationnelle</b>	Probable	Élevé	Critique	<ul style="list-style-type: none"> <li>• Accompagnement au changement structuré</li> <li>• Formation équipes aux pratiques cloud-native</li> <li>• Communication transparente sur les bénéfices</li> </ul>
<b>Complexité architecturale</b>	Probable	Élevé	Critique	<ul style="list-style-type: none"> <li>• Adoption progressive par phases</li> <li>• Architecture patterns éprouvés</li> <li>• Gouvernance technique renforcée</li> </ul>
<b>Performance globale</b>	Possible	Très élevé	Critique	<ul style="list-style-type: none"> <li>• Architecture distribuée optimisée</li> <li>• Tests de charge continus</li> <li>• Monitoring business en temps réel</li> </ul>

<b>Sécurité distribuée</b>	Possible	Très élevé	Critique	<ul style="list-style-type: none"> <li>• Stratégie Zero Trust</li> <li>• Sécurité by design</li> <li>• Conformité réglementaire automatisée</li> </ul>
<b>Dépendance technologique</b>	Probable	Moyen	Modéré	<ul style="list-style-type: none"> <li>• Architecture multi-cloud</li> <li>• Standards ouverts privilégiés</li> <li>• Stratégie de portabilité</li> </ul>
<b>Adoption utilisateur</b>	Possible	Élevé	Élevé	<ul style="list-style-type: none"> <li>• Approche user-centric</li> <li>• Tests utilisateurs réguliers</li> <li>• Déploiements progressifs</li> </ul>
<b>Conformité réglementaire</b>	Probable	Très élevé	Critique	<ul style="list-style-type: none"> <li>• Privacy by design</li> <li>• Data sovereignty compliance</li> <li>• Audit continu des pratiques</li> </ul>
<b>Scalabilité internationale</b>	Possible	Élevé	Élevé	<ul style="list-style-type: none"> <li>• Architecture multi-région native</li> <li>• Localisation automatisée</li> <li>• Infrastructure élastique</li> </ul>

## ÉCHELLE D'ÉVALUATION

### Probabilité :

- **Peu probable** : Risque faible d'occurrence
- **Possible** : Risque modéré nécessitant surveillance
- **Probable** : Risque élevé nécessitant mitigation active
- **Certain** : Risque inévitable nécessitant préparation complète

### Impact Business :

- **Faible** : Impact opérationnel mineur
- **Moyen** : Impact sur les objectifs court terme
- **Élevé** : Impact significatif sur la croissance
- **Très élevé** : Impact critique sur la viabilité business

### Criticité :

- **Faible** : Surveillance passive
- **Modéré** : Surveillance active et préparation
- **Élevé** : Plan de mitigation détaillé requis
- **Critique** : Action immédiate et ressources dédiées

## FACTEURS DE RÉDUCTION GLOBAUX

### Excellence opérationnelle :

- Mise en place d'une culture DevSecOps transverse
- Automatisation maximale des processus critiques
- Observabilité complète de la plateforme et des métriques business

### Gouvernance moderne :

- Architecture Decision Records (ADR) pour traçabilité des choix
- Revues architecturales régulières avec les équipes techniques
- Politique de gestion des changements agile et documentée

### Innovation contrôlée :

- Adoption progressive des technologies émergentes
- Proof of concepts systématiques avant industrialisation
- Veille technologique continue et évaluation d'impact

## 7.2. Hypothèses

Le succès de cette transformation cloud-native repose sur des hypothèses stratégiques clés qui conditionnent l'approche et les décisions architecturales.

Hypothèse stratégique	Impact sur l'architecture	Propriétaire	Validation requise
<b>Disponibilité des compétences cloud-native</b>	Les décisions architecturales présupposent l'existence ou l'acquisition rapide d'expertise en architecture distribuée moderne	CIO	Formation/recrutement validé avant Phase C
<b>Budget transformation suffisant</b>	L'architecture cible ne tient pas compte de contraintes budgétaires restrictives sur les coûts d'infrastructure cloud et d'outillage	CFO	Approbation budgétaire finalisée avant Phase E
<b>Adhésion organisationnelle</b>	La transformation présuppose l'acceptation du changement par l'ensemble des équipes et l'adaptation des processus métier	CEO	Conduite du changement validée avant Phase F
<b>Conformité réglementaire évolutive</b>	L'architecture supporte l'évolution des réglementations sans refonte majeure grâce à sa conception flexible	CPO	Veille juridique continue activée
<b>Maturité écosystème partenaires</b>	Les intégrations avec les partenaires locaux sont réalisables via des APIs modernes sans développements spécifiques majeurs	CMO	Audit capacités partenaires avant Phase D
<b>Acceptation utilisateur</b>	Les utilisateurs finaux adopteront les nouvelles expériences digitales sans résistance significative au changement	CPO	Tests utilisateurs validant l'UX avant Phase G
<b>Stabilité technologique</b>	Les technologies cloud-native sélectionnées conserveront leur pertinence et support sur la durée du projet	CTO	Évaluation vendor et roadmaps tech avant Phase D
<b>Capacité de migration</b>	La migration depuis l'architecture existante vers l'architecture cible est réalisable sans interruption critique de service	Engineering Owner	Stratégie de migration validée avant Phase F

### SUIVI DES HYPOTHÈSES

**Revue périodique :** Les hypothèses font l'objet d'une réévaluation trimestrielle lors des comités de pilotage, avec mise à jour des plans de mitigation si nécessaire.

**Seuils d'alerte :** Chaque hypothèse dispose d'indicateurs de validation permettant de déclencher des actions correctives avant impact critique sur le projet.

**Escalation :** Toute invalidation d'hypothèse critique déclenche automatiquement une revue d'architecture et potentiellement une adaptation du périmètre ou de l'approche.

## 8. Critères d'acceptation et procédures

### 8.1. Métriques et KPI

La réussite de la transformation cloud-native de Foosus sera mesurée selon des indicateurs business validés et des critères architecturaux essentiels à l'atteinte des objectifs stratégiques.

#### MÉTRIQUES DE PERFORMANCE BUSINESS

Les indicateurs de réussite ont été définis précédemment lors de la rédaction de la Requête de travail d'architecture :

Indicateur business	Changement attendu	Délai de validation
Croissance utilisateurs globale	Augmenter de 10%	3 mois post-déploiement
Adhésion de producteurs locaux	Passer de 1,4/mois à 4/mois	6 mois post-déploiement
Délai moyen de parution (Time-to-market)	Passer de 3,5 semaines à moins d'une semaine	Immédiat post-déploiement
Taux d'incident de production	Passer de >25/mois à <1/mois	3 mois de stabilisation

#### MÉTRIQUES D'EXCELLENCE ARCHITECTURALE

Ces critères garantissent la qualité technique et opérationnelle de l'architecture cloud-native déployée :

Dimension architecturale	Critère d'acceptation	Objectif d'excellence
Disponibilité globale	>99% uptime multi-région	99,9% avec basculement automatique
Performance internationale	Support croissance 1M+ utilisateurs	Latences optimisées par géolocalisation
Sécurité et conformité	100% conformité réglementaire	Certification sécurité par région
Scalabilité élastique	Absorption pics de charge sans interruption	Auto-scaling transparent

### 8.2. Procédure d'acceptation

L'acceptation de la transformation suit un processus structuré en deux niveaux de validation garantissant la conformité technique et l'alignement business.

**Validation technique** : Chaque livrable architectural est validé par le groupe projet technique selon les critères d'acceptation définis, puis soumis au comité sponsors pour approbation stratégique et allocation des ressources de déploiement.

**Acceptation définitive** : Une fois la validation sponsor obtenue, les éléments approuvés sont communiqués à l'ensemble des parties prenantes et déployés selon le planning de migration validé. L'acceptation finale intervient après validation des métriques business sur une période de stabilisation de 3 mois, confirmant l'atteinte des objectifs de transformation.



# 9. Personnes approuvant ce plan

**Ash CALLUM**

Chief Executive Officer

---

**Natasha JARSON**

Chief Information Officer

---

**Daniel ANTHONY**

Chief Product Officer

---

**Jo KUMAR**

Chief Financial Officer

---

**Christina ORGEGA**

Chief Marketing Officer

---

Date de signature :