

# **Contrat d'Architecture avec le Développement et le Design**



## **Foosus Géoconscient**

<b>Nom du projet</b>	Foosus Géoconscient
<b>Préparé par</b>	JOUDAR Mohamed
<b>Version N°</b>	1.2
<b>Titre</b>	Contrat d'Architecture avec le Développement et le Design
<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>4</b>
<b>2. Introduction et contexte</b>	<b>4</b>
<b>3. La Nature de l'accord</b>	<b>5</b>
<b>4. Objectifs et Périmètre</b>	<b>5</b>
4.1. Objectifs	5
4.2. Périmètre	6
4.3. Parties prenantes, préoccupations et visions	6
<b>5. Description de l'architecture, principes stratégiques et conditions requises</b>	<b>7</b>
5.1. Description	7
5.2. Principes stratégiques	9
5.3. Conditions requises pour l'architecture	10
<b>6. Livrables architecturaux</b>	<b>11</b>
<b>7. Plan de travail commun priorisé</b>	<b>14</b>
7.1 Phase préliminaire	14
7.2 Phase A – Vision de l'architecture	15
7.3 Phase B – Architecture Business	15
7.4 Phase C – Architecture des systèmes d'information	16
7.5 Phase D – Architecture technologique	18
7.6 Phase E – Opportunités et solutions	19
7.7 Phase F – Planning de Migration	20
7.8 Phase G – Gouvernance de l'implémentation	21
7.9 Phase H – Management du changement d'architecture	22
7.10 Phase N – Management des conditions requises	22
<b>8. Structure de gouvernance</b>	<b>23</b>
<b>9. Plan de communication</b>	<b>23</b>
9.1 Création ou mise à jour d'un document ou d'un artefact	23
9.2 Réunion du groupe projet	24
9.3 Réunion des sponsors	24
<b>10. Risques et facteurs de réduction</b>	<b>24</b>
<b>11. Hypothèses</b>	<b>26</b>
<b>12. Critères d'acceptation et procédures</b>	<b>27</b>
12.1 Métriques et KPI	27
12.2 Procédure d'acceptation	28
<b>13 Procédures de changement de périmètre</b>	<b>28</b>
<b>14. Calendrier</b>	<b>30</b>
<b>15. Phases de livrables définies</b>	<b>30</b>
<b>16. Personnes approuvant ce plan</b>	<b>32</b>

# 1. Objet de ce document

Les Contrats d'Architecture sont les accords communs entre les partenaires de développement et les sponsors sur les livrables, la qualité, et la correspondance à l'objectif d'une architecture.

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

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

Ceci est une déclaration d'intention signée sur la conception et le développement de l'architecture d'entreprise, ou de parties significatives de celles-ci, de la part d'organisations partenaires, y compris les intégrateurs système, fournisseurs d'applications, et fournisseurs de service.

De plus en plus, le développement d'un ou plusieurs domaine(s) d'architecture (business, données, application, technologie) peut être externalisé, avec la fonction d'architecture de l'entreprise fournissant une vue d'ensemble de l'architecture d'entreprise globale, ainsi que la coordination et le contrôle de l'effort total. Dans certains cas, même ce rôle de supervision peut être externalisé, bien que la plupart des entreprises préfèrent conserver cette responsabilité clé en interne.

Quelles que soient les spécificités des dispositions d'externalisation, les dispositions elles-mêmes seront normalement gouvernées par un Contrat d'Architecture qui définit les livrables, la qualité, et la correspondance à l'objectif de l'architecture développée, ainsi que les processus de collaboration pour les partenaires du développement de l'architecture.

## 2. Introduction et contexte

Foosus s'impose comme un acteur innovant de l'écosystème alimentaire durable, orchestrant la connexion directe entre consommateurs conscients et producteurs locaux à travers sa marketplace géolocalisée. Cette mission répond à une demande marché confirmée par les analyses comportementales qui révèlent l'aspiration croissante des consommateurs vers la proximité alimentaire et le soutien aux circuits courts.

L'entreprise développe actuellement sa stratégie d'expansion géographique en s'appuyant sur des technologies de géolocalisation intelligente, permettant aux utilisateurs d'accéder instantanément

aux offres alimentaires disponibles dans leur zone de proximité. Cette approche hyperlocale constitue le socle de la différenciation concurrentielle de Foosus sur le marché de l'e-commerce alimentaire.

Cependant, l'infrastructure technologique existante atteint ses limites de scalabilité et contraint la vélocité de croissance envisagée. L'architecture monolithique actuelle ne supporte plus les exigences de performance, de résilience et d'agilité nécessaires à l'expansion internationale projetée. Cette contrainte technique représente un risque stratégique majeur face aux géants de l'e-commerce global.

Pour lever ces verrous technologiques, Foosus engage une transformation architecturale vers une plateforme cloud-native distribuée, conçue pour soutenir sa compétitivité internationale tout en préservant sa capacité d'innovation et d'adaptation aux spécificités des marchés locaux.

### 3. La Nature de l'accord

Ce contrat établit le cadre de gouvernance et les engagements mutuels pour la conception et le développement de la nouvelle architecture cloud-native de Foosus. Il définit les responsabilités, les livrables attendus, et les critères de succès que doivent respecter l'ensemble des équipes techniques impliquées dans cette transformation.

L'accord s'adresse spécifiquement aux squads de développement, aux équipes DevSecOps, aux architectes d'entreprise, et aux responsables opérationnels qui collaboreront pour matérialiser cette vision architecturale ambitieuse. Il constitue le socle contractuel garantissant l'alignement entre les objectifs business stratégiques et leur implémentation technique via une approche cloud-native moderne.

### 4. Objectifs et Périmètre

#### 4.1. 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
Hyperscalabilité cloud-native	<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>
Sécurité zero-trust omnicanale	<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>
Disponibilité globale edge-native	<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>

## 4.2. 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.

## 4.3. 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 ORGEA</b> (CMO)	Différenciation concurrentielle et expansion géographique	Validation du positionnement technologique et capacités marketing automation
<b>Pete PARKER</b> (Engineering Owner)	Architecture moderne et vélocité 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 2025
<b>Équipes produit</b>	Time-to-market et feedback loops rapides	Exploitation des capacités d'expérimentation et A/B testing

## 5. Description de l'architecture, principes stratégiques et conditions requises

### 5.1. Description

#### Standardisation technologique

**Langage de développement principal** : Les services backend utilisent prioritairement Java, capitalisant sur l'expertise des équipes tout en répondant aux exigences techniques de la plateforme.

**Homogénéité de stack** : L'ensemble des choix technologiques respecte une cohérence de pile, tant pour les technologies de développement que d'infrastructure.

#### Orchestration et scalabilité

**Gestion de conteneurs** : Kubernetes constitue la solution standard pour l'orchestration, offrant auto-scaling automatique et gestion optimisée des ressources.

**Scaling automatique** : Les conteneurs se multiplient automatiquement lors des pics de charge et se libèrent quand la demande diminue, optimisant l'utilisation des ressources.

#### Distribution géographique

**Répartition multi-région** : L'orchestration Kubernetes distribue la charge entre serveurs géographiquement répartis, garantissant performance globale et résilience.

**Optimisation régionale** : La solution adapte automatiquement la distribution selon la localisation des utilisateurs et la qualité des connexions.

#### Déploiements sans interruption

**Basculement progressif** : Les déploiements supportent temporairement deux versions simultanément, dirigeant progressivement le trafic vers la nouvelle version.

**Transition transparente** : Les nouvelles requêtes sont orientées vers les nouveaux conteneurs tandis que les anciens traitent les sessions en cours jusqu'à leur terme naturel.

## Gestion de versions et rollback

**Workflow Git standardisé** : Le développement suit une approche trunk-based avec merge fast-forward, évitant les commits superflus et les historiques complexes.

**Traçabilité des versions** : Chaque version est identifiée par des tags permettant une localisation rapide et un rollback efficace.

**Infrastructure-as-Code** : La configuration d'infrastructure versionnée accompagne le code applicatif, facilitant les déploiements reproductibles et les rollbacks complets.

## Environnements normalisés

**Environnement de développement** : Chaque merge request dispose d'un environnement isolé permettant la validation indépendante des modifications.

**Environnement d'intégration** : La branche principale est déployée automatiquement pour tester l'intégration des fonctionnalités.

**Environnement de qualification** : Les versions candidates sont validées via des tests d'acceptation avant passage en production.

**Environnement de production** : Versioning rigoureux avec capacité de rollback vers versions antérieures validées.

## Pipeline de qualité

**Intégration continue** : Pipeline CI/CD automatisé avec indicateurs de qualité obligatoires (couverture de code, tests automatisés).

**Validation métier** : Tests d'acceptation systématiques en environnement de qualification avant déploiement production.

**Livraisons fréquentes** : Cycles de livraison courts pour feedback utilisateur rapide et réactivité aux dysfonctionnements.

## Architecture microservices

**Pattern de communication** : Architecture découplée utilisant des patterns modernes (Gateway Pattern) pour la communication inter-services.

**Développement indépendant** : Services développés de manière autonome avec intégration via gateway centralisée.

## Sécurité et accès



**Gestion des droits** : Solution intégrée de gestion des rôles et permissions avec granularité appropriée selon les types d'utilisateurs.

**Déploiement géographique** : Configuration de services par zone géographique selon les contraintes business et réglementaires.

**Protocoles sécurisés** : Adoption systématique des derniers standards de sécurité, notamment HTTPS pour toutes les communications web.

**Conformité continue** : Respect des dernières normes et versions en vigueur pour tous les éléments critiques de sécurité.

## 5.2. Principes stratégiques

Les principes architecturaux suivants guident toutes les décisions techniques et orientent l'implémentation cloud-native :

**Standardisation technologique stratégique** : La solution privilégie l'harmonisation des technologies cloud-native pour réduire la complexité et optimiser la maintenance. L'adoption d'une stack cohérente facilite la montée en compétences et la collaboration entre équipes.

**Évolutivité fonctionnelle adaptative** : L'architecture permet l'ajout et la modification des services selon le rythme business de l'entreprise. Cette flexibilité garantit l'adaptation aux besoins métier évolutifs sans refonte majeure de la plateforme.

**Déploiement géographique intelligent** : La solution supporte la création et le déploiement de services pour des régions ou utilisateurs spécifiques. Cette capacité constitue un avantage concurrentiel majeur pour l'expansion internationale de Foosus.

**Sécurité omniprésente** : La plateforme intègre la sécurité par conception, garantissant la protection des données et transactions en tout lieu et circonstance. Cette approche zero-trust renforce la confiance utilisateur et la conformité réglementaire.

**Scalabilité utilisateur transparente** : L'architecture absorbe automatiquement les pics de trafic et supporte la croissance continue du nombre d'utilisateurs. Cette élasticité préserve la performance lors des montées en charge imprévues.

**Performance géographique uniforme** : La solution maintient disponibilité et performance optimales quelle que soit la zone géographique d'accès. Cette universalité facilite l'adoption globale et l'expérience utilisateur cohérente.

**Déploiements sans interruption** : L'implémentation élimine les interruptions de service lors des mises à jour, avec capacité de retour en arrière rapide en cas de dysfonctionnement. Cette continuité opérationnelle préserve l'activité business critique.

**Innovation technologique continue** : La plateforme encourage l'expérimentation et l'adoption de nouvelles technologies. Cette ouverture maintient l'avantage concurrentiel et facilite l'évolution technologique.

**Livraisons fiables et régulières** : L'architecture garantit des cycles de déploiement stables et prévisibles. Cette régularité améliore la planification business et la réactivité face aux évolutions marché.

## 5.3. Conditions requises pour l'architecture

La transformation cloud-native de Foosus s'appuie sur un ensemble de principes architecturaux modernes, adaptés aux enjeux de scalabilité internationale et d'innovation technologique. Ces conditions garantissent la cohérence et l'excellence de l'architecture distribuée.

### Choix technologiques

**Stratégie open-source première :** Les solutions cloud-native open source sont privilégiées pour leur innovation continue, leur communauté active, et leur évitement du vendor lock-in. L'écosystème CNCF (Cloud Native Computing Foundation) constitue le référentiel de choix.

**Support et pérennité :** La sélection technologique intègre systématiquement l'évaluation de la maturité projet (graduation CNCF), la roadmap vendor, et la taille de la communauté. Chaque composant doit garantir un support long terme et une évolutivité compatible avec les enjeux d'expansion internationale.

**Cohérence de stack :** L'architecture adopte une approche platform engineering avec une stack technologique unifiée : Kubernetes pour l'orchestration, service mesh (Istio/Envoy) pour la communication, GitOps (ArgoCD/Flux) pour le déploiement, et observabilité intégrée (Prometheus/Grafana/Jaeger). Cette homogénéité réduit la complexité opérationnelle et optimise les coûts de maintenance.

### Principes data

**Data mesh et distributed architecture :** La modélisation des données adopte une approche domain-driven design avec event sourcing et CQRS patterns, permettant l'évolutivité indépendante des domaines métier tout en préservant la cohérence globale.

**Privacy by design et data sovereignty :** La protection des données personnelles est intégrée dès la conception avec chiffrement end-to-end, tokenisation des PII, et architecture respectant les contraintes de résidence des données par région (GDPR, CCPA, LGPD).

**Cohérence éventuelle intelligente :** L'architecture data privilégie la disponibilité et la partition tolerance (théorème CAP) avec des patterns de cohérence éventuelle adaptés au contexte business. Les données critiques (paiements, stocks) maintiennent une cohérence forte, tandis que les données d'analytics acceptent une cohérence relâchée.

**Domain boundaries et context mapping :** Chaque bounded context dispose de son modèle de données optimisé, avec API contrats bien définis pour l'intégration inter-domaines. Cette approche facilite l'évolution indépendante et la scalabilité horizontale.

### Principes d'application

**Microservices et single responsibility :** Chaque service respecte le principe de responsabilité unique avec couplage faible et cohésion forte. L'architecture event-driven permet la communication asynchrone et la résilience distribuée.

**API-first design :** Toutes les interfaces suivent une approche contract-first avec spécifications OpenAPI, versioning sémantique, et backward compatibility. Les APIs publiques incluent rate limiting, authentification OAuth 2.0/OIDC, et monitoring complet.

**Contract-driven development :** Les interfaces inter-services reflètent uniquement les données et opérations nécessaires à l'intégration, suivant les principes de least privilege et information hiding. Les consumer-driven contracts garantissent la stabilité des APIs.

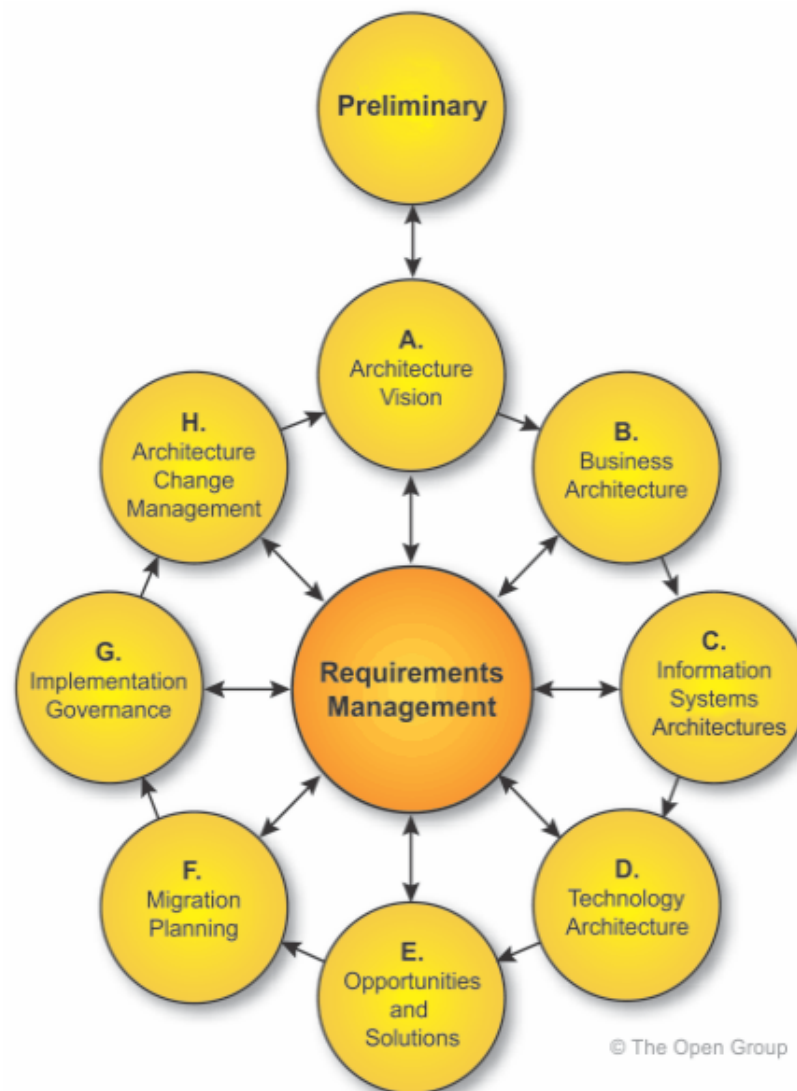
**Dependency management :** L'architecture évite les dépendances cycliques via une hiérarchisation claire des couches : présentation → application → domaine → infrastructure. Les patterns anti-corruption layer protègent contre les couplages non désirés.

**Resilience patterns :** L'implémentation native des patterns de résilience (circuit breaker, retry, timeout, bulkhead) garantit la fault tolerance et la graceful degradation en cas de défaillance partielle.

Ces conditions requises constituent le socle architectural sur lequel s'appuient les contrats de service et spécifications d'implémentation détaillés dans les sections suivantes.

## 6. Livrables architecturaux

Le développement de cette architecture s'appuie sur la méthode ADM suivante :



Le section plus bas **12. Critères d'acceptation et procédures** décrit les **métriques de l'architecture cible**.

Quant aux **phases de livraisons**, la méthode ADM constitue le framework méthodologique de référence. 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.0	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.

## 7. Plan de travail commun priorisé

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.

### 7.1 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

## 7.2 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

#### 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

#### 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

## 7.3 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 des 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 de formation, les 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.

## 7.4 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

Lors de l'élaboration de ce document :

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner fournit les éléments techniques nécessaires.
- Les équipes de développement peuvent être consulté sur le sujet pour leur expertise sur les aspects techniques.

### 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.

Lors de l'élaboration de ce document :

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner fournit les éléments techniques nécessaires.

- Les équipes de développement peuvent être consulté sur le sujet pour leur expertise sur les aspects techniques.

### 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

## 7.5 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.

Il est possible de se baser sur les artefacts suivants :

- **Technology Portfolio Catalog**, afin d'identifier la liste des technologie utilisé dans l'entreprise, y compris la gestion de leur cycle de vie et de leur versionning
- **Platform Decomposition Catalog**, afin d'illustrer les plateformes technologiques qui vont supporter les opérations de l'architecture des systèmes d'information
- **Processing Diagram**, afin d'identifier des unité de code et de configuration déployables et les technologies sur lesquels elles sont déployées

Lors de l'élaboration de ce document :

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

#### 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.

Il est possible de se baser sur les artefacts énumérés pour la définition d'architecture

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.

## 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

## 7.6 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

Lors de l'élaboration de ce document :

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

#### 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.

- La CIO valide les décisions techniques prises dans le document avec le support de l'Engineering Owner et de l'Operations Lead.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

#### 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.

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

### **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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

### **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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

## **7.7 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.

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

#### **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.

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

### **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).

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

### **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.

- Il est produit conjointement avec la CIO qui valide la vision finale technique du projet.

### **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).

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

## **7.8 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

### **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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

## **7.9 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.

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

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

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

- Il est produit conjointement avec la CIO qui valide la vision finale technique du projet.

### **Statement of Architecture Work**

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

- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.

## **7.10 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.

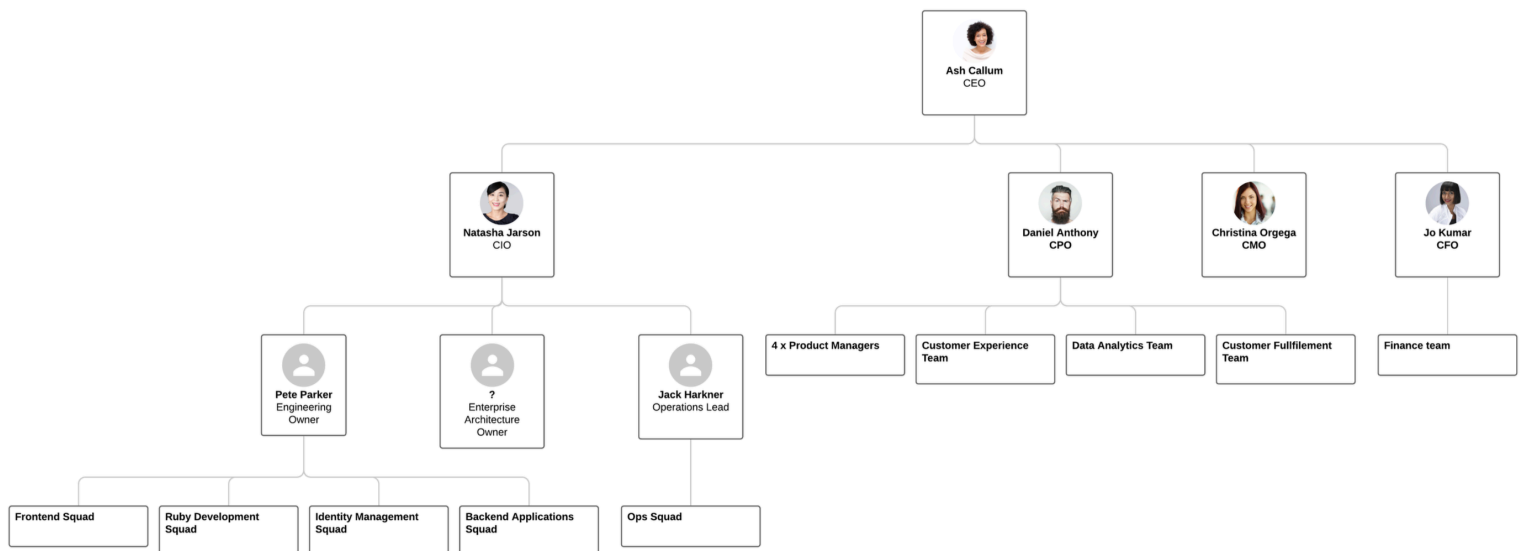
- La CIO définit et valide l'orientation stratégique technique énoncé dans le document avec le support de l'Engineering Owner et de l'Operations Lead.
- Les équipes de développement peuvent s'appuyer sur ce document pour apporter les corrections nécessaires dans leur travail.

### Architecture Requirement Specification

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

- La CIO valide les décisions stratégiques techniques énoncées dans le document
- Le Engineering Owner et l'Operations Lead fournissent les éléments techniques nécessaires.
- Les équipes de développement et d'Ops peuvent être consultés sur le sujet pour leur expertise sur les aspects techniques.

## 8. Structure de gouvernance



## 9. Plan de communication

### 9.1 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

## 9.2 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

## 9.3 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

# 10. Risques et facteurs de réduction

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

## 11. 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.

# 12. Critères d'acceptation et procédures

## 12.1 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
<b>Croissance utilisateurs globale</b>	Augmenter de 10%	3 mois post-déploiement
<b>Adhésion de producteurs locaux</b>	Passer de 1,4/mois à 4/mois	6 mois post-déploiement
<b>Délai moyen de parution (Time-to-market)</b>	Passer de 3,5 semaines à moins d'une semaine	Immédiat post-déploiement
<b>Taux d'incident de production</b>	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
<b>Disponibilité globale</b>	>99% uptime multi-région	99,9% avec basculement automatique
<b>Performance internationale</b>	Support croissance 1M+ utilisateurs	Latences optimisées par géolocalisation

<b>Sécurité et conformité</b>	100% conformité réglementaire	Certification sécurité par région
<b>Scalabilité élastique</b>	Absorption pics de charge sans interruption	Auto-scaling transparent

## 12.2 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.

## 13 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, conformité) 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.

## 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.

## 14. Calendrier

La transformation architecturale s'articule sur 6 mois selon le cycle ADM TOGAF, avec validation continue et ajustements selon les retours des parties prenantes. Le planning intègre des sprints de 3 semaines et des jalons de validation espacés de 4 à 6 semaines.

**Phase Préliminaire (Semaines 1-2)** : Architecture Repository et gouvernance framework

**Phase A - Vision (Semaines 3-5)** : Architecture Vision et stakeholder alignment validé

**Phases B-C - Business/SI (Semaines 6-13)** : Domain patterns et event-driven architecture

**Phase D - Technologique (Semaines 14-17)** : Stack cloud-native et service mesh finalisés

**Phase E - Solutions (Semaines 18-20)** : Implementation patterns et migration strategy

**Phase F - Migration (Semaines 21-24)** : Deployment automation et production readiness

**Phase G - Gouvernance (Semaines 25-26)** : Go-live et monitoring opérationnel

Les jalons critiques s'appuient sur des Architecture Decision Records automatisés, des fitness functions de validation, et des critères d'acceptation mesurables. La gouvernance intègre des processus de révision bi-hebdomadaires et des capacités d'ajustement via change management GitOps.

## 15. Phases de livrables définies

La méthode ADM produit des livrables architecturaux structurés par phase, permettant la validation progressive et l'implémentation de l'architecture cloud-native cible.

**Phase Préliminaire** : Architecture Repository (Git), Request for Architecture Work, governance framework cloud-native établi.

**Phase A - Vision** : Architecture Vision synthétique, Statement of Architecture Work avec roadmap détaillée, Stakeholder Map Matrix et Value Stream Map validés par les parties prenantes.

**Phases B-C-D** : Architecture Definition Document enrichi par phase (business, SI, technologique), Architecture Requirement Specification avec critères mesurables, artefacts TOGAF spécialisés (Business Service Catalog, Data Entity Catalog, Technology Portfolio).

**Phase E - Solutions** : Implementation patterns finalisés, Project Context Diagram et Benefits Diagram, opportunités d'optimisation identifiées et hiérarchisées.

**Phase F - Migration** : Implementation and Migration Plan opérationnel avec calendrier détaillé, stratégies de déploiement blue-green validées, governance model pour la transition.

**Phases G-H** : Architecture Contract entre équipes, Compliance Assessment avec métriques de conformité, Change Request workflows pour l'évolution continue.

L'ensemble des livrables constitue le socle documentaire et technique permettant aux équipes de matérialiser la plateforme cloud-native selon les spécifications validées, avec traçabilité complète des décisions architecturales.

## 16. 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 ORGEA**

Chief Marketing Officer

---

Date de signature :