

FOOSUS

Contrat de Conception et de Développement de l'Architecture



v 0.1

Ce document ainsi que toutes les informations qu'il contient sont des informations confidentielles appartenant à Foosus.

Table des matières

Information sur le document.....	3
Objet de ce document.....	4
Introduction et Contexte.....	6
La Nature de l'accord.....	7
Objectifs et Périmètre.....	7
Objectifs.....	8
Périmètre.....	8
Description de l'architecture, principes stratégiques et conditions requises.....	10
Description de l'Architecture.....	10
Principes Stratégiques.....	11
Conditions Requises.....	11
Architecture Cible.....	12
Technologies Utilisées.....	13
Livrables architecturaux.....	16
Développement de l'architecture.....	16
Mesures de l'architecture cible.....	16
Livraison de l'architecture et métriques business.....	17
Phases de livraison définies.....	17
Plan de travail commun priorisé.....	18
Plan de Communication.....	20
Risques et facteurs de réduction.....	21
Analyse des risques.....	21
Hypothèses.....	23
Critères d'acceptation et procédures.....	24
Procédures de changement de périmètre.....	27
Conditions requises pour la conformité.....	29
Processus.....	30
Rôles et Responsabilités.....	31
Calendrier.....	32
Durée et Effort.....	32
Approbations signées.....	33

Information sur le document

Nom du projet	Conception d'une nouvelle architecture
Préparé par :	Clément Bastion
N° de version du document :	0.1
Titre :	Contrat de Conception et de Développement de l'Architecture
Date de version du document :	02/08/2024
Revu par :	
Date de révision :	

Objet de ce document

Les Contrats d'Architecture sont des accords établis entre les partenaires de développement et les sponsors, définissant les livrables, la qualité, et l'adéquation aux objectifs de l'architecture. Une gouvernance efficace de l'architecture (cf. TOGAF Partie VII, Gouvernance de l'architecture) est essentielle pour la mise en œuvre réussie de ces accords. En adoptant une approche rigoureuse de gestion des contrats, les éléments suivants seront garantis :

- **Système de contrôle continu** : Ce système vérifie l'intégrité, les modifications, les prises de décisions, et l'audit de toutes les activités liées à l'architecture au sein de l'organisation.
- **Adhésion aux principes et standards** : Les architectures existantes ou en développement doivent respecter les principes, standards, et conditions requises établies.
- **Identification des risques** : Les risques sont évalués pour tous les aspects du développement et de l'implémentation de l'architecture, incluant le développement interne selon les standards acceptés, les politiques, les technologies, et les produits, ainsi que les aspects opérationnels pour assurer la résilience de l'organisation.

Les pratiques incluent :

- **Transparence, responsabilité et discipline** : Mise en place de processus garantissant ces principes dans le développement et l'utilisation de tous les artefacts architecturaux.
- **Accord formel de gouvernance** : Définition de l'autorité de l'organe de gouvernance, ainsi que du périmètre de l'architecture sous sa supervision.

Les Contrats d'Architecture constituent une déclaration d'intention signée par les organisations partenaires, incluant les intégrateurs de systèmes, les fournisseurs d'applications, et les fournisseurs de services, concernant la conception et le développement de l'architecture d'entreprise ou de parties significatives de celle-ci.

De plus en plus, le développement de certains domaines d'architecture (business, données, application, technologie) peut être externalisé. L'entreprise conserve une vue d'ensemble de l'architecture globale et coordonne l'effort total, même si dans certains cas, la supervision peut aussi être externalisée. Cependant, la plupart des entreprises préfèrent conserver cette responsabilité clé en interne.

Quelles que soient les spécificités des arrangements d'externalisation, ceux-ci sont normalement régis par un Contrat d'Architecture. Ce contrat définit les livrables, la qualité, et la correspondance à l'objectif de l'architecture développée, ainsi que les processus de collaboration avec les partenaires de développement de l'architecture.

Introduction et Contexte

Foosus fait face à des défis critiques avec sa plateforme historique, construite sans une architecture cohérente. Cette plateforme est devenue coûteuse à maintenir et limite l'innovation. Les interruptions de service et l'instabilité ont conduit à une baisse des inscriptions d'utilisateurs, affectant la compétitivité de l'entreprise.

Le projet vise à créer une nouvelle architecture, permettant à Foosus de se moderniser et de s'aligner sur ses objectifs stratégiques. La nouvelle plateforme coexistera temporairement avec l'ancienne, facilitant une transition progressive. L'objectif est de garantir une infrastructure stable et évolutive, capable de soutenir une innovation rapide et une expansion sur les marchés locaux.

Ce document décrit les exigences et les plans pour le développement de l'architecture de la nouvelle plateforme Foosus. Il définit les objectifs, les périmètres, les livrables, les processus de gouvernance et de gestion des risques, ainsi que les hypothèses et les critères d'acceptation.

La Nature de l'accord

L'accord porte sur la collaboration structurée entre Foosus, ses partenaires de développement, et les sponsors, pour la conception et la mise en œuvre d'une nouvelle architecture de plateforme. Cet accord vise à établir des engagements clairs en termes de livrables, de qualité, et d'alignement avec les objectifs stratégiques de l'entreprise.

Objectifs de l'accord

1. **Définir les responsabilités et les attentes** : L'accord clarifie les rôles et les responsabilités de chaque partie prenante, assurant une compréhension commune des attentes et des obligations.
2. **Garantir la qualité des livrables** : Il précise les standards de qualité auxquels les livrables doivent se conformer, incluant les critères de performance, de sécurité, et de conformité aux réglementations.
3. **Aligner l'architecture avec les objectifs business** : L'accord s'assure que l'architecture développée soutient les objectifs stratégiques de Foosus, tels que l'amélioration de l'expérience utilisateur, l'augmentation des inscriptions et le soutien aux marchés locaux.

Éléments clés de l'accord

- **Livrables** : Détail des produits finaux attendus, incluant la documentation technique, les systèmes intégrés, et les rapports de conformité.
- **Qualité** : Définition des standards de qualité, couvrant les aspects de robustesse, de sécurité, et de performance des systèmes.
- **Processus de collaboration** : Établissement de procédures pour la communication, la prise de décision, et la résolution de conflits entre les partenaires.
- **Gouvernance** : Mise en place d'une structure de gouvernance pour superviser le projet, assurer le suivi des progrès et valider les étapes clés.

Gouvernance et conformité

Cet accord inclut des dispositions pour une gouvernance efficace, en s'assurant que toutes les activités relatives à l'architecture sont en conformité avec les principes directeurs établis par l'entreprise. Il précise également les mécanismes de contrôle pour surveiller les changements, garantir la transparence et la responsabilité, et auditer les processus pour assurer l'intégrité des résultats.

Objectifs et Périmètre

Objectifs

Objectif	Notes
Accélérer le Time-to-Market	Lancer rapidement de nouvelles fonctionnalités et produits pour répondre aux besoins évolutifs des consommateurs et du marché.
Améliorer l'efficacité opérationnelle	Optimiser les processus internes pour réduire les coûts et améliorer la productivité.
Soutenir la croissance de l'entreprise	Faciliter l'expansion dans de nouveaux marchés géographiques et l'augmentation de la base de clients.
Promouvoir l'innovation	Créer un environnement technologique flexible permettant une expérimentation rapide et sûre de nouvelles idées.
Renforcer l'expérience utilisateur	Offrir une performance élevée, une disponibilité continue et des fonctionnalités personnalisées pour les utilisateurs.
Faciliter l'intégration avec les partenaires externes	Assurer une intégration facile et rapide avec les systèmes des partenaires logistiques, des fournisseurs et des plateformes de paiement.
Assurer la conformité et la sécurité des données	Intégrer des mécanismes robustes de sécurité et de protection des données pour se conformer aux réglementations en vigueur.

Périmètre

Les parties prenantes, leurs préoccupations et visions sont décrites dans le tableau suivant. Ce tableau montre les parties prenantes qui utilisent ce document, leurs préoccupations, et la façon dont le travail d'architecture répondra à ces préoccupations par l'expression de plusieurs visions.

Partie prenante	Préoccupation	Vision
-----------------	---------------	--------

Ash Callum, CEO	Maintenir un taux positif d'inscriptions de nouveaux utilisateurs, expansion dans les marchés locaux	Une architecture scalable qui soutient la croissance de la base d'utilisateurs en fournissant du géociblage et de la flexibilité pour toucher une gamme plus large d'utilisateurs.
Jo Kumar, CFO	Maximiser les investissements en IT tout en assurant un retour sur investissement rapide	Une plateforme évolutive qui permet de nouvelles fonctionnalités sans augmentation significative des coûts opérationnels.
Natasha Jarson, CIO	Garantir la stabilité et la sécurité du système tout en facilitant l'innovation technique	Une architecture robuste avec des mécanismes de sécurité intégrés, supportant l'innovation et les expérimentations rapides.
Christina Orgega, CMO	Améliorer la réputation de Foosus sur le marché et garantir la satisfaction des clients	Une plateforme fiable et stable, minimisant les interruptions de service et offrant une expérience utilisateur exceptionnelle.
Daniel Anthony, CPO	Accélérer le développement de nouvelles fonctionnalités et améliorer l'expérience produit	Un environnement de développement flexible et rapide, avec des micro-services modulaires permettant des itérations rapides.
Pete Parker, Ingénieur	Assurer la maintenabilité et la cohérence du code, tout en réduisant la dette technique	Une architecture modulaire et standardisée facilitant la maintenance et l'évolution continue du système sans complexité accrue.

Description de l'architecture, principes stratégiques et conditions requises

Description de l'Architecture

La nouvelle architecture de la plateforme Foosus est conçue pour être évolutive, flexible et résiliente, tout en répondant aux besoins actuels et futurs de l'entreprise. Elle repose sur une approche de microservices, permettant une modularité et une scalabilité accrues. Cette architecture intègre plusieurs composants clés :

1. **Microservices** : Chaque service est décomposé en modules indépendants, facilitant les déploiements et les mises à jour sans perturber l'ensemble du système. Cette structure permet également une maintenance ciblée et une évolutivité facile.
2. **Géolocalisation** : Un composant central de la nouvelle plateforme est l'intégration de la géolocalisation, permettant aux utilisateurs de trouver des producteurs et des artisans locaux. Un calculateur de distance aide à proposer des options proches des utilisateurs, optimisant ainsi l'expérience utilisateur.
3. **Accessibilité multi-plateformes** : La plateforme est conçue pour être accessible à partir de divers appareils, y compris les ordinateurs, les smartphones, et les tablettes. Elle est optimisée pour fonctionner sur des réseaux à faible bande passante ainsi que sur des connexions haut débit, garantissant une expérience utilisateur cohérente.
4. **Sécurité renforcée** : La sécurité des données est une priorité, avec l'implémentation de protocoles de chiffrement avancés, une gestion stricte des accès et une surveillance continue pour détecter et répondre aux menaces potentielles.
5. **Cloud computing et haute disponibilité** : La plateforme utilise des solutions cloud pour assurer une haute disponibilité et une résilience face aux pannes. Cela inclut des capacités de récupération rapide en cas de désastre et une distribution de contenu optimisée via des réseaux de distribution de contenu (CDN).

Principes Stratégiques

Les principes stratégiques qui guident la conception et le développement de cette architecture incluent :

1. **Modularité et Flexibilité** : L'architecture doit être facilement extensible et modifiable, permettant d'ajouter de nouvelles fonctionnalités ou de modifier les existantes sans impact majeur sur l'ensemble du système.
2. **Scalabilité** : La plateforme doit pouvoir évoluer pour supporter une augmentation du nombre d'utilisateurs et de transactions, sans compromettre les performances ou la disponibilité.
3. **Sécurité et Conformité** : Assurer la protection des données des utilisateurs et se conformer aux réglementations internationales en matière de sécurité des données, comme le RGPD.
4. **Performance et Réactivité** : Maintenir des temps de réponse rapides et une haute disponibilité, même en période de forte demande.
5. **Expérience Utilisateur** : Offrir une interface utilisateur intuitive et fluide, avec un accès facile aux services et une personnalisation basée sur la localisation et les préférences des utilisateurs.

Conditions Requises

Pour réaliser les objectifs et respecter les principes stratégiques, plusieurs conditions requises ont été définies :

1. **Infrastructure Technique** :
 - Utilisation de micro-services déployés sur une infrastructure cloud pour assurer une haute disponibilité et une scalabilité flexible.
 - Intégration d'une solution de gestion de bases de données capables de gérer des données géolocalisées et de supporter des volumes élevés de transactions.
2. **Sécurité** :
 - Implémentation de mécanismes d'authentification et d'autorisation robustes, incluant l'utilisation de protocoles OAuth 2.0 et JWT (JSON Web Tokens).
 - Surveillance proactive des systèmes pour détecter les menaces et répondre aux incidents de sécurité.
3. **Interopérabilité** :
 - Les systèmes doivent pouvoir s'intégrer avec des solutions tierces, y compris les systèmes de paiement, les services de logistique, et d'autres partenaires externes.

- Support des API ouvertes et conformes aux standards pour faciliter l'intégration et l'extension des services.

4. Gestion des Performances :

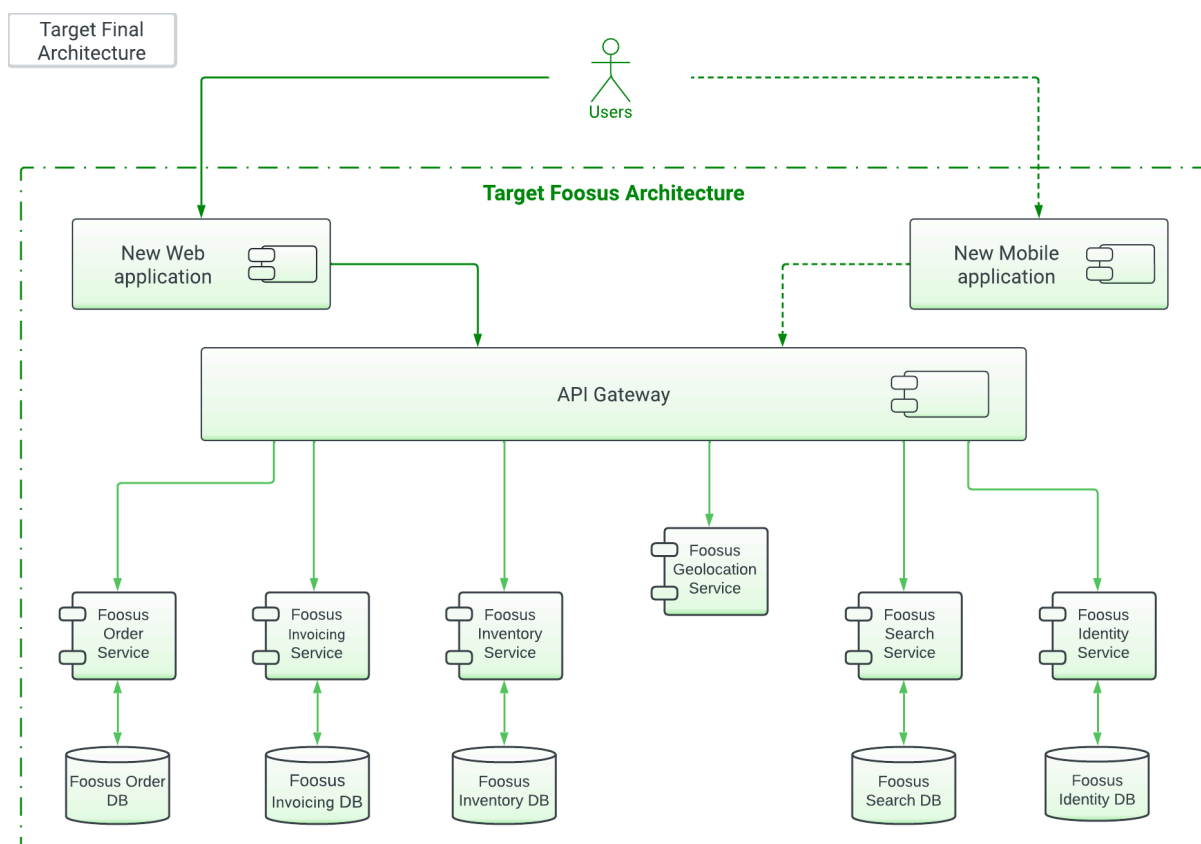
- Utilisation d'outils de monitoring pour suivre les performances du système en temps réel et assurer une disponibilité continue.
- Mise en place de solutions de mise en cache et d'optimisation des requêtes pour maintenir des temps de réponse rapides.

5. Innovation et Expérimentation :

- Capacité à lancer rapidement de nouvelles fonctionnalités ou à expérimenter des variantes de produits, avec une infrastructure qui supporte des cycles de développement et de déploiement rapides.

Architecture Cible

Nous avons adapté notre architecture source à notre nouvelle base d'architecture en respectant le principe de micro-services. Cette transition permettra de décomposer les applications monolithiques en composants plus petits, indépendants et plus faciles à gérer, alignant ainsi notre infrastructure avec les besoins actuels et futurs de l'entreprise.



Technologies Utilisées

L'adoption de technologies appropriées est cruciale pour atteindre les objectifs de l'architecture modulaire basée sur les microservices. Voici un aperçu des technologies clés qui seront utilisées :

Backend

- **Spring Boot (Java) :**
 - **Robustesse :** Spring Boot est un framework mature et largement adopté, reconnu pour sa robustesse et sa stabilité.
 - **Micro-services :** Conçu pour faciliter la création de micro-services, avec des fonctionnalités intégrées pour la gestion des configurations et des dépendances.
 - **Communauté :** Une large communauté et une excellente documentation, facilitant le support et la résolution de problèmes.
 - **Sécurité :** Intégration facile avec des standards de sécurité et des mécanismes d'authentification.

Frontend

- **React.js :** Bibliothèque JavaScript pour construire des interfaces utilisateur interactives.
 - **Performances :** Virtual DOM et rendu efficace, améliorant les performances des applications interactives.
 - **Écosystème :** Un écosystème riche avec des outils comme Redux pour la gestion de l'état et une forte adoption dans l'industrie.
 - **Réutilisabilité :** Composants réutilisables qui facilitent la maintenance et la scalabilité du code.
 - **Support :** Soutenu par Facebook et une grande communauté, garantissant une évolution continue et un bon support.

Bases de Données

- **MongoDB :**
 - MongoDB est une base de données NoSQL qui permet de stocker des données non structurées et semi-structurées de manière flexible. Elle est adaptée aux applications nécessitant une scalabilité horizontale et une gestion de données évolutive.
- **PostgreSQL :**
 - PostgreSQL est une base de données relationnelle open source connue pour sa robustesse, sa conformité ACID, et ses capacités d'extension. Elle est idéale

pour des applications nécessitant des transactions complexes et une intégrité des données.

Communication entre Services

- **Kafka :**
 - Apache Kafka est une plateforme de streaming distribuée qui permet de gérer le flux de données en temps réel. Il est adapté aux systèmes nécessitant une grande scalabilité et une tolérance aux pannes, en particulier pour les applications de traitement de données en temps réel.
- **RabbitMQ :**
 - RabbitMQ est un message broker qui facilite la communication asynchrone entre les micro-services. Il est simple à configurer et offre des fonctionnalités avancées de routage des messages, ce qui est essentiel pour des systèmes distribués.

Infrastructure et Déploiement

- **Docker :**
 - Docker permet de conteneuriser les applications, assurant ainsi leur portabilité et une scalabilité simplifiée. Les conteneurs garantissent que les applications fonctionnent de manière cohérente dans différents environnements.
- **Kubernetes :**
 - Kubernetes est un orchestrateur de conteneurs qui automatise le déploiement, la gestion et la mise à l'échelle des applications conteneurisées. Il est indispensable pour gérer des clusters de conteneurs et assurer une haute disponibilité et une tolérance aux pannes.
- **AWS/GCP/Azure :**
 - Les services cloud comme AWS, GCP, et Azure offrent une infrastructure scalable et fiable, avec une multitude de services gérés pour le déploiement, la gestion et la sécurisation des applications. Le choix du fournisseur dépendra des besoins spécifiques et des accords commerciaux existants.

Sécurité

- **OAuth 2.0 :**
 - OAuth 2.0 est un standard d'autorisation sécurisé largement adopté pour permettre aux applications d'accéder aux ressources sans partager les informations de connexion des utilisateurs. Il est essentiel pour protéger les API et les données utilisateur.
- **JWT (JSON Web Tokens) :**

- Les JWT sont utilisés pour créer des tokens d'accès sécurisés et stateless, facilitant l'authentification et l'autorisation dans des environnements distribués. Ils permettent une vérification rapide et sécurisée des utilisateurs.

Monitoring et Logging

- **Prometheus :**

- Prometheus est un système de monitoring et d'alerte open source conçu pour la scalabilité et la fiabilité. Il est capable de surveiller des milliers de services et de serveurs, ce qui est crucial pour des applications en production.

- **ELK Stack (Elasticsearch, Logstash, Kibana) :**

- La suite ELK offre une solution complète pour la gestion et l'analyse des logs. Elasticsearch permet une recherche rapide, Logstash collecte et transforme les données, et Kibana offre des visualisations puissantes. Cette combinaison est idéale pour le suivi et le diagnostic des performances des applications.

Livrables architecturaux

Les livrables architecturaux sont des documents et des artefacts qui détaillent l'architecture de la plateforme, mesurent les performances attendues, et guident la livraison et l'évolution du système. Ces livrables doivent satisfaire aux conditions requises pour le business et assurer que la solution développée est alignée avec les objectifs stratégiques de Foosus.

Développement de l'architecture

Livrables :

- **Schémas d'architecture globale** : Ces schémas illustrent la structure de haut niveau de l'architecture, y compris les composants principaux, les micro-services, les bases de données, et les interfaces externes. Ils servent de guide de référence pour les équipes de développement et d'opérations.
- **Documentation des micro-services** : Détails des API, des flux de données, des protocoles de communication, et des interconnexions entre les différents services. Cette documentation inclut également des spécifications pour les contrats de données et les points de terminaison des services.
- **Manuel des standards technologiques** : Liste des technologies, des frameworks et des bibliothèques approuvés, ainsi que des normes de codage et de sécurité à respecter. Ce manuel assure une cohérence dans le développement et la maintenance des services.

Objectif : Fournir une base solide et documentée pour le développement de la nouvelle architecture, garantissant une compréhension claire des composants et de leur interaction.

Mesures de l'architecture cible

Livrables :

- **Rapport de performance des systèmes** : Ce rapport inclut des indicateurs de performance clés (KPI) tels que les temps de réponse, la capacité de traitement des requêtes, et la disponibilité. Ces mesures sont comparées aux objectifs définis pour évaluer l'efficacité de l'architecture.
- **Tests de charge et de stress** : Documentation des scénarios de test, des résultats, et des analyses pour évaluer la robustesse de l'architecture sous différentes charges. Cela inclut des tests de simulation de pannes pour vérifier la résilience des systèmes.
- **Analyse de la sécurité** : Évaluation des vulnérabilités potentielles, des protocoles de sécurité mis en place, et des mesures correctives recommandées. Inclut une vérification de conformité aux normes de sécurité et aux réglementations (comme le RGPD).

Objectif : Évaluer si l'architecture cible répond aux exigences de performance, de sécurité, et de résilience, et identifier les améliorations nécessaires.

Livraison de l'architecture et métriques business

Livrables :

- **Plan de déploiement :** Détails sur les étapes de déploiement, les environnements de pré-production et de production, et les stratégies de transition pour éviter les interruptions de service. Inclut des procédures pour les mises à jour continues et les déploiements de nouvelles fonctionnalités.
- **Tableau de bord des métriques business :** Outil de visualisation des KPI liés aux objectifs business, tels que le nombre d'adhésions d'utilisateurs, le taux de rétention, et l'augmentation des transactions. Ce tableau de bord permet de suivre l'impact des changements architecturaux sur les performances business.
- **Rapport de conformité finale :** Documentation de la conformité de l'architecture livrée aux spécifications et aux exigences réglementaires. Ce rapport est essentiel pour l'acceptation finale du projet par les parties prenantes.

Objectif : Assurer une livraison ordonnée et mesurable de l'architecture, tout en fournissant des outils pour évaluer l'atteinte des objectifs business.

Phases de livraison définies

Livrables :

- **Feuille de route du projet :** Chronologie détaillée des phases de livraison, incluant les jalons importants, les livrables intermédiaires, et les dates clés pour les revues et les tests. Cette feuille de route aide à coordonner les efforts entre les équipes et à suivre l'avancement du projet.
- **Journal des versions :** Historique des versions déployées, incluant les nouvelles fonctionnalités, les améliorations, et les corrections de bogues. Ce document est crucial pour la traçabilité des changements et pour la gestion des mises à jour.
- **Plan de migration :** Stratégie pour la transition des utilisateurs et des données de l'ancienne plateforme à la nouvelle. Inclut des étapes détaillées pour minimiser les risques et assurer une continuité de service.

Objectif : Planifier et exécuter les livraisons de l'architecture de manière contrôlée, en minimisant les risques et en maximisant la transparence pour les parties prenantes.

Plan de travail commun priorisé

Voici le plan de travail commun priorisé sous forme de tableau, basé sur les éléments disponibles dans le document fourni :

Élément de Travail	Activités	Livrables
Analyse Initiale	<ul style="list-style-type: none">- Collecte des besoins auprès des parties prenantes.- Analyse des systèmes existants.- Évaluation des technologies.	<ul style="list-style-type: none">- Déclaration de travail d'architecture.- Rapport d'analyse initiale.
Conception de l'Architecture	<ul style="list-style-type: none">- Développement de la vision architecturale- Conception des modèles d'architecture cible.- Définition des interfaces et des interactions entre les composants.	<ul style="list-style-type: none">- Document de définition d'architecture.- Diagrammes d'architecture.
Spécification des Conditions Requises	<ul style="list-style-type: none">- Définition des exigences quantitatives pour la mise en œuvre de l'architecture.- Validation des spécifications avec les parties prenantes.	<ul style="list-style-type: none">- Spécification des conditions requises pour l'architecture.
Contrats d'Architecture	<ul style="list-style-type: none">- Élaboration des contrats d'architecture.- Négociation des accords de niveau de service (SLA).	<ul style="list-style-type: none">- Contrat d'architecture avec la conception et le développement.- Contrat d'architecture avec les utilisateurs professionnels.
Développement et Implémentation	<ul style="list-style-type: none">- Développement des micro-services.- Intégration continue et déploiement continu (CI/CD).- Mise en place de l'infrastructure de monitoring et de logging.	<ul style="list-style-type: none">- Code source des micro-services.- Scripts de déploiement.- Infrastructure de monitoring et de logging.

Tests et Validation	<ul style="list-style-type: none"> - Conduite de tests unitaires, d'intégration, de performance et de sécurité. - Validation des fonctionnalités avec les parties prenantes. 	<ul style="list-style-type: none"> - Rapports de tests. - Plan de validation.
Formation et Transition	<ul style="list-style-type: none"> - Formation des équipes internes. - Documentation des processus opérationnels - Transition progressive des systèmes existants. 	<ul style="list-style-type: none"> - Documentation de formation. - Plans de transition.
Mise en Production et Support Post-Lancement	<ul style="list-style-type: none"> - Mise en production de la nouvelle architecture. - Surveillance post-lancement. - Réception des feedbacks et ajustements continus. 	<ul style="list-style-type: none"> - Documentation de mise en production. - Rapports de surveillance post-lancement. - Plan de support post-lancement.

Plan de Communication

Évènements	Canaux	Formats	Contenu
Réunions de lancement	En personne/visioconférence	Présentations PowerPoint	Objectifs et avancement du projet
Réunions hebdomadaires de suivi	En personne/visioconférence	Présentations PowerPoint	Décisions prises et actions à suivre
Comités de pilotage mensuels	En personne/visioconférence	Présentations PowerPoint	Changements de périmètre et ajustements nécessaires
Séances de validation avec les parties prenantes	En personne/visioconférence	Présentations PowerPoint	Suivi des livrables et des jalons
Emails	Email	Emails récapitulatifs	Résumé des réunions et décisions importantes
Intranet d'entreprise	Intranet	Tableaux de bord en ligne	Suivi des progrès et documentation à jour
Outils de gestion de projet (e.g., JIRA, Confluence)	Outils de gestion de projet	Documents Word/PDF, Tableaux de bord en ligne	Planification des tâches et suivi des actions

Risques et facteurs de réduction

Analyse des risques

Identification des Risques

ID	Risque	Conséquences
1	Dépassement du budget	<ul style="list-style-type: none">- Projets retardés ou annulés- Réduction de la portée du projet- Perte de crédibilité
2	Exigences imprécises ou en constante évolution	<ul style="list-style-type: none">- Délais prolongés- Coûts supplémentaires- Insatisfaction des parties prenantes
3	Vulnérabilité des données	<ul style="list-style-type: none">- Violations de données- Perte de confiance des clients- Amendes réglementaires
4	Complexité accrue de l'architecture	<ul style="list-style-type: none">- Maintenance difficile- Risque accru de dysfonctionnements- Délais de développement plus longs
5	Retards dans les délais de livraison	<ul style="list-style-type: none">- Dépassement du budget- Insatisfaction des clients- Perte de parts de marché

Stratégies de réduction

ID	Probabilité (P)	Gravité (G)	Criticité (C) = P x G	Responsable	Facteur de réduction
1	Moyenne	Haute	15	Chef de Projet	Bonne planification budgétaire et suivi régulier des dépenses
2	Moyenne	Haute	15	Chef de Projet	Validation continue des exigences avec les parties prenantes et gestion des changements
3	Moyenne	Très haute	20	Expert en Sécurité	Mise en place de mesures de sécurité robustes, chiffrement des données et audits réguliers
4	Moyenne	Moyenne	9	Architecte en Chef	Validation de l'architecture par des experts externes, adoption de standards de codage
5	Moyenne	Haute	15	Chef de Projet	Suivi rigoureux du planning, gestion proactive des risques et des dépendances

Hypothèses

Ces hypothèses doivent être vérifiées régulièrement tout au long du projet. Si l'une d'elles se révèle incorrecte, cela peut entraîner des ajustements dans la planification et la gestion du projet pour garantir le succès de l'initiative.

ID	Hypothèse	Impact	Propriétaire
1	Les ressources nécessaires (humaines et financières) seront disponibles tout au long du projet	Une disponibilité continue des ressources garantit l'achèvement du projet dans les délais prévus et prévient les retards et les dépassements de budget.	Chef de Projet
2	Les parties prenantes resteront engagées et fourniront des retours rapides et constructifs	Un engagement continu des parties prenantes assure la pertinence et l'acceptation des livrables, facilitant une prise de décision rapide et efficace.	Responsable Produit
3	Les technologies sélectionnées seront compatibles avec les systèmes existants	La compatibilité technologique réduit les risques de réintégration, minimise les coûts supplémentaires et assure une transition en douceur.	Architecte en Chef
4	Les besoins des utilisateurs finaux ne changeront pas significativement au cours du projet	Une stabilité des besoins utilisateurs facilite la gestion du périmètre et aide à respecter les délais et les budgets.	Responsable Produit

Critères d'acceptation et procédures

Les critères d'acceptation définissent les standards de qualité et les objectifs de performance que la nouvelle architecture doit atteindre pour être approuvée. Ces critères sont essentiels pour garantir que les livrables répondent aux exigences des parties prenantes et aux besoins stratégiques de l'entreprise. Les procédures d'acceptation incluent des tests, des revues, et l'approbation finale des livrables par les parties prenantes concernées.

Procédures d'acceptation

1. **Revue de Conformité Technique :**

- Chaque composant de l'architecture sera vérifié pour s'assurer qu'il respecte les spécifications techniques et les exigences fonctionnelles. Les équipes techniques réaliseront des revues détaillées pour vérifier la conformité des livrables.

2. **Tests de Validation :**

- Des tests rigoureux seront menés pour valider les performances, la sécurité, l'interopérabilité, et la résilience des systèmes. Ces tests incluent des tests de charge, de sécurité (y compris des tests de pénétration), et de récupération après sinistre.

3. **Tests d'Acceptation Utilisateur (UAT) :**

- Les utilisateurs finaux participeront à des sessions de test pour valider que la solution répond à leurs besoins et est conviviale. Les feedbacks seront utilisés pour apporter les ajustements nécessaires avant la mise en production.

4. **Approbation Finale :**

- Une réunion de validation finale impliquant toutes les parties prenantes clés sera organisée. La signature des documents d'acceptation sera requise pour officialiser l'achèvement et l'approbation des livrables.

Métriques et KPIs de l'État Cible de l'Architecture

Les métriques et les KPIs (Key Performance Indicators) suivants seront utilisés pour évaluer l'efficacité de la nouvelle architecture et s'assurer qu'elle répond aux objectifs définis.

Métrique	Technique de mesure	Valeur cible	Justification	Notes supplémentaires
Temps de réponse des systèmes	Tests de performance	< 200 ms	Assurer une expérience utilisateur fluide et réactive.	Mesuré sous différentes conditions de charge.
Disponibilité des services	Monitoring de l'infrastructure	99.9%	Minimiser les interruptions de service pour maximiser la satisfaction utilisateur.	Calculé sur une période de 12 mois.
Taux d'erreur	Surveillance des logs	< 0.1%	Réduire les erreurs système pour améliorer la fiabilité.	Inclut les erreurs critiques.
Capacité de traitement	Tests de charge	10 000 transactions/minute	Assurer la capacité de traiter de grandes quantités de données simultanément.	Basé sur des scénarios de pointe.
Conformité aux normes de sécurité	Audits de sécurité	100%	Garantir la protection des données et la conformité réglementaire.	Inclut les audits de conformité RGPD.

Métriques de livraison de l'architecture et du business

Ces métriques mesurent l'impact de la nouvelle architecture sur les objectifs business de Foosus, en particulier en termes de croissance et d'engagement utilisateur.

Métrique	Technique de mesure	Valeur cible	Justification	Notes supplémentaires
Croissance des inscriptions	Suivi des nouvelles inscriptions	+20% dans les 6 mois	Augmenter la base d'utilisateurs grâce aux améliorations de la plateforme.	Comparaison des données avant/après.
Engagement utilisateur	Analyse de l'utilisation	50% d'utilisateurs actifs par mois	Améliorer l'engagement en proposant des fonctionnalités adaptées.	Mesuré par la fréquence d'utilisation.
Retour sur investissement (ROI)	Analyse financière	> 150%	Justifier l'investissement dans la nouvelle architecture par les gains financiers.	Inclut l'analyse des coûts et des revenus.
Satisfaction des utilisateurs	Enquêtes de satisfaction	> 85%	Renforcer la fidélité des utilisateurs grâce à une meilleure expérience.	Inclut les scores NPS et autres indices.

Procédures de changement de périmètre

Les procédures de changement de périmètre (ou **scope change procedures**) sont mises en place pour gérer les modifications apportées aux objectifs, aux livrables ou aux ressources du projet. Elles garantissent que tous les changements sont documentés, évalués, approuvés, et implémentés de manière contrôlée. Cela permet de minimiser les risques associés aux changements imprévus et d'assurer que les parties prenantes sont informées et d'accord avec les modifications apportées. Voici les étapes clés du processus de changement de périmètre :

1. Identification du Besoin de Changement

- **Origine des changements** : Les demandes de changement peuvent provenir de diverses sources, telles que des parties prenantes, des équipes techniques, des réglementations nouvelles, ou des évolutions du marché.
- **Description du changement** : La première étape consiste à formuler clairement le changement proposé, en précisant les nouvelles exigences ou les ajustements souhaités.

2. Documentation et Soumission

- **Formulaire de demande de changement** : Le changement proposé doit être documenté dans un formulaire de demande de changement, qui inclut une description détaillée, les raisons du changement, et les impacts potentiels.
- **Soumission de la demande** : Le formulaire est soumis au comité de gestion de projet ou à l'équipe de gouvernance pour examen.

3. Évaluation de l'Impact

- **Évaluation de l'impact** : Une évaluation complète est réalisée pour comprendre les implications du changement sur le calendrier, le budget, les ressources, la qualité des livrables, et les risques associés.
 - **Impact sur le calendrier** : Analyse de l'impact sur les échéances projetées et ajustement des jalons.
 - **Impact sur le budget** : Estimation des coûts supplémentaires ou des économies résultant du changement.
 - **Impact sur les ressources** : Identification des besoins en ressources supplémentaires, en termes de personnel, de matériel ou de logiciels.
 - **Impact sur la qualité** : Vérification des répercussions sur les standards de qualité et la conformité aux exigences initiales.

4. Approbation et Décision

- **Révision par le comité de gestion de projet** : Le comité examine l'évaluation de l'impact et décide si le changement doit être approuvé, modifié ou rejeté.
- **Approbation des parties prenantes** : Les principales parties prenantes doivent approuver le changement proposé, surtout s'il implique des modifications majeures aux objectifs du projet.

5. Mise en Œuvre du Changement

- **Plan d'implémentation** : Un plan détaillé est élaboré pour mettre en œuvre le changement, incluant les étapes nécessaires, les responsables, et les délais.
- **Communication** : Les équipes concernées et les parties prenantes sont informées du changement, des raisons derrière celui-ci, et de la manière dont il sera implémenté.
- **Mise en œuvre** : Le changement est intégré dans le projet selon le plan établi.

6. Suivi et Contrôle

- **Suivi de l'implémentation** : La mise en œuvre du changement est surveillée pour s'assurer qu'elle se déroule comme prévu. Les écarts sont corrigés en temps opportun.
- **Évaluation post-implémentation** : Après l'intégration du changement, une évaluation est réalisée pour vérifier si les objectifs du changement ont été atteints et pour identifier toute leçon apprise.

7. Mise à Jour de la Documentation

- **Mise à jour des documents de projet** : Toutes les documentations, y compris le plan de projet, les spécifications des exigences, et les plans de test, sont mises à jour pour refléter les changements approuvés.
- **Archivage des demandes de changement** : Les formulaires de demande de changement et les décisions associées sont archivés pour référence future et pour l'audit du projet.

Conditions requises pour la conformité

La conformité dans le cadre du projet d'architecture pour la plateforme Foosus implique le respect des normes, des réglementations, et des standards internes et externes. Ces conditions requises pour la conformité garantissent que l'architecture développée adhère aux exigences légales, sécuritaires et de qualité. L'accord sur un cadre de conformité permet de structurer l'évaluation, le suivi et la validation de cette conformité tout au long du projet.

1. Conformité Légale et Réglementaire

- **Régulations locales et internationales** : Respect des lois et réglementations en vigueur dans les différentes juridictions où la plateforme sera opérée. Cela inclut, mais ne se limite pas à, les réglementations sur la protection des données (comme le RGPD en Europe), les lois sur le commerce en ligne, et les réglementations financières.
- **Protection des données personnelles** : Mise en place de mécanismes de sécurité pour protéger les informations personnelles des utilisateurs, conformément aux réglementations de confidentialité et de protection des données. Cela inclut l'anonymisation, le chiffrement, et la gestion des consentements.
- **Conformité aux standards industriels** : Adoption de standards reconnus de l'industrie pour la sécurité, la qualité, et l'interopérabilité, tels que ISO/IEC 27001 pour la gestion de la sécurité de l'information.

2. Conformité Sécuritaire

- **Normes de sécurité informatique** : Application de normes de sécurité rigoureuses pour protéger l'infrastructure et les données contre les cybermenaces. Cela inclut les tests de pénétration réguliers, les audits de sécurité, et la gestion des vulnérabilités.
- **Gestion des accès et identités** : Implémentation de protocoles d'authentification et d'autorisation robustes, incluant l'utilisation de méthodes modernes comme l'authentification multi-facteurs (MFA) et les protocoles d'autorisation comme OAuth 2.0.

3. Conformité de Qualité et Performance

- **Garantie de qualité** : Développement de l'architecture selon des standards de qualité bien définis, garantissant des performances optimales, la résilience, et la capacité d'évoluer avec les besoins de l'entreprise.
- **Tests de conformité** : Conduite de tests exhaustifs pour vérifier que les systèmes respectent les critères de performance, de sécurité, et de fiabilité définis dans les spécifications.

Développement et propriété de l'architecture

Processus

1. Planification et Conception

- **Collecte des Exigences** : Identification des besoins des utilisateurs et des exigences techniques, menée par le Responsable Produit et l'Architecte en Chef.
- **Développement de la Vision d'Architecture** : Création d'une vision d'ensemble de l'architecture cible, incluant les principes directeurs, les objectifs et les contraintes.

2. Développement

- **Conception des Microservices** : Développement des microservices conformément aux spécifications techniques et fonctionnelles.
- **Intégration Continue (CI)** : Mise en place d'un pipeline de CI pour automatiser les tests et l'intégration des nouveaux codes.
- **Déploiement Continu (CD)** : Utilisation de pratiques de CD pour automatiser le déploiement des microservices dans les environnements de test et de production.

3. Test et Validation

- **Tests de Performance et de Sécurité** : Réalisation de tests pour s'assurer que l'architecture répond aux critères de performance et de sécurité établis.
- **Validation par les Utilisateurs** : Conduite de tests d'acceptation par les utilisateurs finaux pour vérifier la satisfaction des exigences fonctionnelles.

4. Mise en Production et Suivi

- **Déploiement en Production** : Mise en œuvre de la nouvelle architecture dans l'environnement de production, avec une surveillance continue pour détecter et résoudre rapidement les problèmes.
- **Suivi et Support Post-Lancement** : Surveillance continue des performances, gestion des incidents et planification des améliorations futures.

Rôles et Responsabilités

Rôle	Responsabilités	Responsable Hiérarchique
Chef de Projet	<ul style="list-style-type: none">- Coordination globale du projet- Gestion des ressources et des délais- Suivi du budget	Ash Callum, CEO
Architecte en Chef	<ul style="list-style-type: none">- Définition de l'architecture globale- Supervision technique- Choix des technologies et frameworks	Natasha Jarson, CIO
Responsable DevOps	<ul style="list-style-type: none">- Gestion des environnements de développement et production- Implémentation des pipelines CI/CD- Surveillance des infrastructures	Natasha Jarson, CIO
Expert en Sécurité	<ul style="list-style-type: none">- Mise en place des mesures de sécurité- Gestion des accès et audits de sécurité- Gestion des incidents de sécurité	Natasha Jarson, CIO
Responsable Produit	<ul style="list-style-type: none">- Définition des fonctionnalités produit- Priorisation des développements- Validation des livrables	Daniel Anthony, CPO
Analyste de Données	<ul style="list-style-type: none">- Analyse des données- Fourniture d'insights stratégiques- Optimisation des processus	Daniel Anthony, CPO
Responsable Support Client	<ul style="list-style-type: none">- Gestion des interactions avec les utilisateurs finaux- Collecte des feedbacks- Assistance technique et résolution des problèmes	Christina Orgega, CMO

Calendrier

Durée et Effort

Phase	Durée Estimée	Effort Estimé
Analyse Initiale	1 mois	4 personnes x 1 mois = 4 PM (Person-Months)
Conception de l'Architecture	2 mois	4 personnes x 2 mois = 8 PM
Spécification des Conditions Requises	1 mois	3 personnes x 1 mois = 3 PM
Élaboration des Contrats	1 mois	2 personnes x 1 mois = 2 PM
Développement et Implémentation	4 mois	6 personnes x 4 mois = 24 PM
Tests et Validation	2 mois	4 personnes x 2 mois = 8 PM
Formation et Transition	1 mois	3 personnes x 1 mois = 3 PM
Mise en Production et Support Post-Lancement	2 mois	3 personnes x 2 mois = 6 PM
Total	14 mois	58 PM

Approbations signées

Nom	Date	Signature
Ash Callum, CEO		
Daniel Anthony, CPO		
Natasha Jarson, CIO		