



Spécification des conditions requises pour l'architecture

LOUIS ZEPHIR

Historique

Projet	Client	Préparé par
XXX	YYYY	

—

Titre	Version	Commentaires	Auteur	Date
Spécification des Conditions requises pour l'Architecture	0.1	Rédaction du document	Louis ZEPHIR	09/2024

Table des Matières

Table des matières

Historique	1
Table des Matières	2
Objet de ce document.....	3
Mesures du succès	3
Conditions requises pour l'architecture.....	4
Contrats de service business	6
Accords de niveau de service	6
Contrats de service application	7
Objectifs de niveau de service	7
Indicateurs de niveau de service.....	8
Lignes directrices pour l'implémentation	9
Spécifications pour l'implémentation.....	10
Standards pour l'implémentation	11
Conditions requises pour l'interopérabilité	13
Conditions requises pour le management du service IT.....	14
Contraintes	15
Hypothèses	16

Objet de ce document

La Spécification des Conditions requises pour l'Architecture fournit un ensemble de déclarations quantitatives qui dessinent ce que doit faire un projet d'implémentation afin d'être conforme à l'architecture.

Une Spécification des Conditions requises pour l'Architecture constitue généralement un composant majeur du contrat d'implémentation, ou du contrat pour une Définition de l'Architecture plus détaillée.

Comme mentionné ci-dessus, la Spécification des Conditions requises pour l'Architecture accompagne le Document de Définition de l'Architecture, avec un objectif complémentaire : le Document de Définition de l'Architecture fournit une vision qualitative de la solution et tâche de communiquer l'intention de l'architecte.

La Spécification des Conditions requises pour l'Architecture fournit une vision quantitative de la solution, énumérant des critères mesurables qui doivent être remplis durant l'implémentation de l'architecture.

Mesures du succès

Les mesures de succès pour le projet d'architecture de Foosus sont définies pour évaluer l'efficacité de l'architecture à répondre aux objectifs stratégiques, techniques et business de l'entreprise.

Ces métriques fourniront une base pour valider les livrables et garantir que l'architecture permet à Foosus d'atteindre son état cible.

Mesure	Description	Valeur cible	Justification
Temps moyen de déploiement	Durée entre le début du développement et la mise en production d'une fonctionnalité.	≤ 4 semaines	Permet de valider l'agilité de l'architecture et sa capacité à supporter des cycles d'innovation rapides.
Disponibilité de la plateforme	Pourcentage de temps où la plateforme est opérationnelle et accessible par les utilisateurs finaux.	≥ 99,9 %	Garantit une expérience utilisateur fiable et continue, essentielle pour la satisfaction des clients.
Temps de réponse	Temps nécessaire pour répondre aux	≤ 200 ms	Assure une expérience utilisateur fluide et soutient les objectifs de

moyen des microservices	requêtes des utilisateurs via les services critiques.		performance de l'architecture.
Taux de satisfaction des utilisateurs	Pourcentage des utilisateurs finaux satisfaits, mesuré via des enquêtes ou des retours directs.	≥ 80 %	Permet d'évaluer l'impact de l'architecture sur l'expérience utilisateur globale.
Réduction de la dette technique	Pourcentage de réduction des composants obsolètes ou non standards dans le système existant.	≥ 25 % la première année	Confirme que l'architecture favorise une meilleure maintenabilité et réduit les obstacles au développement futur.
Scalabilité en période de forte charge	Nombre maximal de requêtes que la plateforme peut traiter sans interruption ou dégradation des performances.	≥ 10 000 requêtes/minut ^e	Valide la capacité de l'architecture à gérer la croissance de l'activité et les pics de trafic.
Interopérabilité des microservices	Pourcentage des microservices compatibles avec les standards d'interopérabilité définis.	100 %	Garantit que l'architecture est cohérente et modulable, facilitant les ajouts futurs.

Suivi et validation

- Ces mesures seront suivies régulièrement (mensuellement ou trimestriellement) pour évaluer la progression et ajuster les priorités si nécessaire.
- Chaque mesure est directement alignée avec les objectifs stratégiques de Foosus, garantissant une évaluation claire de la réussite du projet.

Conditions requises pour l'architecture

Les conditions requises pour l'architecture de Foosus définissent les critères fondamentaux qui doivent être respectés pour garantir que l'architecture soutient efficacement les objectifs stratégiques, business et techniques de l'entreprise.

Ces exigences couvrent plusieurs domaines, incluant la performance, la scalabilité, la sécurité, et la modularité.

Catégorie	Condition requise	Description
Performance	Temps de réponse des services : ≤ 200 ms	Les services doivent répondre rapidement pour offrir une expérience utilisateur optimale.
	Disponibilité de la plateforme : $\geq 99,9$ %	La plateforme doit rester accessible à tout moment pour éviter des interruptions critiques.
Scalabilité	Gestion des pics de trafic : $\geq 10\,000$ requêtes/minute	L'architecture doit être capable de s'adapter aux pics de trafic sans compromettre les performances.
Modularité et flexibilité	Architecture microservices : Standardisation et responsabilité unique pour chaque service.	Garantit une maintenance simplifiée et la possibilité d'ajouter de nouvelles fonctionnalités facilement.
	Interopérabilité des services : Tous les services doivent être compatibles entre eux et extensibles.	Permet une intégration fluide entre les différents composants et services tiers.
Sécurité et conformité	Conformité RGPD : Chiffrement des données sensibles et gestion des droits des utilisateurs.	Protège les données des utilisateurs et garantit le respect des normes européennes.
	Contrôles d'accès robustes : Implémentation de mécanismes de sécurité (authentification et autorisation).	Empêche les accès non autorisés aux données et services critiques.
Facilité de maintenance	Documentation complète : Chaque service doit être accompagné d'une documentation technique détaillée.	Facilite la transition des équipes et l'amélioration continue de l'architecture.
	Tests automatisés : Mise en place de pipelines CI/CD avec tests intégrés.	Réduit les erreurs et accélère les cycles de développement et de déploiement.
Expérience utilisateur	Géolocalisation précise : Intégration de services géociblés fiables avec une précision suffisante.	Connecte efficacement les consommateurs avec les producteurs locaux.
	Interface utilisateur performante : Temps de chargement des pages ≤ 2 secondes.	Offre une navigation fluide et rapide, augmentant la satisfaction utilisateur.

Priorisation

- Les conditions liées à la performance et à la sécurité seront prioritaires pour garantir une base solide.
- Les exigences de modularité et de scalabilité seront intégrées dès la phase de conception pour faciliter les évolutions futures.

Engagement

Ces conditions requises définissent le cadre à respecter pour assurer que l'architecture livrée par Foosus réponde aux attentes de toutes les parties prenantes et soutienne efficacement les

objectifs stratégiques de l'entreprise.

Des revues régulières seront réalisées pour garantir leur respect tout au long du projet.

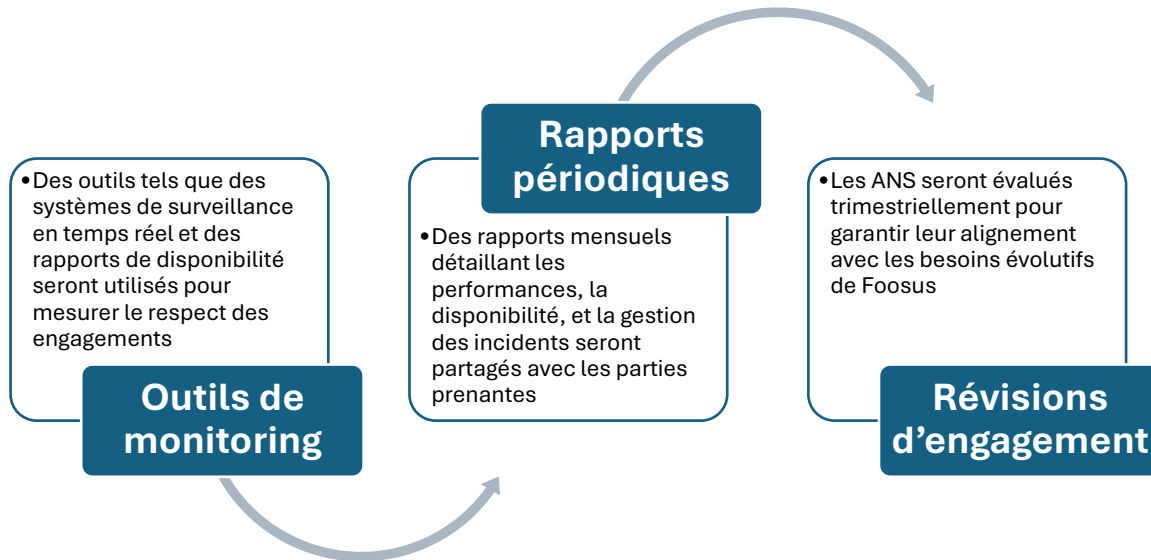
Contrats de service business

Les **accords de niveau de service (ANS)** définissent les attentes et engagements en termes de performance, disponibilité, et support pour l'architecture, afin de garantir une expérience utilisateur optimale et de répondre aux besoins business de Foosus.

Accords de niveau de service

Aspect de service	Engagement	Description
Disponibilité générale	$\geq 99,9 \%$	La plateforme doit être opérationnelle et accessible à tout moment pour les utilisateurs finaux.
Temps de réponse des services	≤ 200 ms pour les requêtes critiques	Assure une performance optimale pour des services tels que la géolocalisation et la gestion des utilisateurs.
Temps de rétablissement (RTO)	≤ 2 heures pour les incidents critiques	Limite l'impact des pannes critiques sur les opérations business et l'expérience utilisateur.
Durée de résolution (MTTR)	≤ 4 heures pour les problèmes non critiques	Garantit un rétablissement rapide des performances en cas de dégradation mineure.
Précision des services géolocalisés	$\geq 95 \%$ de précision pour les résultats de géolocalisation	Les services géociblés doivent offrir une connectivité fiable entre les consommateurs et les producteurs locaux.
Fenêtres de maintenance	1 fois par mois, avec une durée maximale de 2 heures, et notification préalable de 48 heures.	Les interruptions planifiées doivent être minimisées et programmées lors de périodes d'utilisation réduite.
Notification des incidents	≤ 15 minutes après la détection d'un incident critique	Communication proactive avec les parties prenantes pour limiter l'impact des interruptions.

Processus de suivi et validation



Engagement mutuel

- **Équipe technique**
Responsable de respecter les niveaux de service définis en assurant une maintenance proactive et des interventions rapides en cas de problème.
- **Équipe business**
Responsable de fournir des retours réguliers sur les priorités stratégiques pour ajuster les niveaux de service si nécessaire.

Contrats de service application

Objectifs de niveau de service

Les **contrats de service application** définissent les engagements en matière de performance, disponibilité, et qualité des services applicatifs fournis par l'architecture.

Ces objectifs garantissent une expérience utilisateur optimale tout en soutenant les priorités business de Foosus.

Performance
<ul style="list-style-type: none"> Assurer des temps de réponses rapides pour les services applicatifs critiques, tels que la géolocalisation et la gestion des utilisateurs
Disponibilité
<ul style="list-style-type: none"> Fournir une disponibilité continue des services avec des interruptions minimales, même en période de forte utilisation.
Scalabilité
<ul style="list-style-type: none"> Permettre une montée en charge fluide pour gérer les pics d'activité sans dégradation des performances.
Fiabilité
<ul style="list-style-type: none"> Maintenir la cohérence et l'exactitude des résultats fournis par les services.
Interopérabilité
<ul style="list-style-type: none"> Garantir que les applications fonctionnent sans problème avec les systèmes internes et externes

Indicateurs de niveau de service

Indicateur	Description	Valeur cible	Justification	Commentaires
Temps de réponse moyen	Temps moyen pour exécuter une requête sur un service applicatif critique.	≤ 200 ms	Assure une expérience utilisateur fluide, essentielle pour des services comme la géolocalisation.	Mesuré en temps réel via des outils de monitoring.
Taux d'erreurs applicatives	Pourcentage d'erreurs survenant lors de l'utilisation des services applicatifs.	≤ 0,1 %	Réduit les interruptions et les frustrations des utilisateurs finaux.	Analyse des logs pour identifier les causes des erreurs et les corriger rapidement.
Disponibilité des services applicatifs	Pourcentage de temps où les services sont accessibles.	≥ 99,9 %	Garantit une disponibilité continue, même en cas de forte utilisation.	Mesuré sur une base mensuelle avec des rapports automatisés.
Capacité de montée en	Nombre maximal de requêtes simultanées	≥ 10 000	Permet de gérer les pics d'activité liés à	Testée périodiquement

charge	gérées sans dégradation des performances.	requête s/minut e	l'augmentation du nombre d'utilisateurs.	via des simulations de charge.
Interopérabilité des services	Pourcentage des microservices intégrés respectant les standards d'interopérabilité.	100 %	Garantit une cohérence et une compatibilité entre les différents composants de l'architecture.	Validation via des tests d'intégration automatisés.
Temps moyen de récupération (MTTR)	Temps nécessaire pour résoudre un incident affectant un service applicatif.	≤ 4 heures	Limite l'impact des interruptions sur l'activité business et l'expérience utilisateur.	Suivi des temps de résolution via un système de gestion des incidents.
Temps de disponibilité après mise à jour	Pourcentage de temps de disponibilité après une mise à jour applicative.	≥ 99,95 %	Garantit que les mises à jour n'entraînent pas de perturbations significatives pour les utilisateurs.	Mesuré immédiatement après chaque déploiement.

Lignes directrices pour l'implémentation

Principes et recommandations clés pour garantir une implémentation efficace et alignée sur les objectifs business et techniques de Foosus.

Ces lignes directrices couvrent les approches méthodologiques, les bonnes pratiques, et les exigences spécifiques liées au déploiement de l'architecture

Aspect	Lignes directrices	Description
Modularité	Adopter une architecture basée sur les microservices.	Chaque service doit être conçu pour une responsabilité unique, facilitant la maintenance et l'évolution.
Interopérabilité	Utiliser des standards ouverts et des protocoles d'intégration reconnus (ex. REST, JSON, OAuth).	Garantit la compatibilité des composants et services, internes comme externes.
Performance et scalabilité	Concevoir les services pour être scalable horizontalement.	Utiliser des solutions cloud natives pour gérer efficacement la montée en charge.
Déploiement continu	Mettre en place des pipelines CI/CD avec des tests automatisés.	Assure des déploiements fréquents, fiables et sécurisés, sans interruption significative des services.
Sécurité	Implémenter des mécanismes	Garantit que seuls les

	d'authentification et d'autorisation robustes (OAuth 2.0, JWT).	utilisateurs et services autorisés accèdent aux données sensibles.
Conformité	S'assurer que tous les services respectent les exigences du RGPD.	Intégrer des mécanismes de gestion des droits des utilisateurs et de chiffrement des données sensibles.
Surveillance et maintenance	Configurer des outils de monitoring (ex. Prometheus, Grafana) pour suivre les performances et détecter les anomalies.	Garantit une supervision proactive pour minimiser les temps d'arrêt et optimiser les performances.
Documentation	Créer une documentation technique complète pour chaque composant.	Inclut des guides d'utilisation, de déploiement et de maintenance pour les équipes techniques.
Environnement de test	Mettre en place un environnement de test complet et isolé pour valider les services avant déploiement.	Permet de détecter les erreurs et garantir la fiabilité des composants livrés.
Mises à jour et évolutions	Planifier des fenêtres de maintenance régulières avec des mécanismes de rollback en cas d'échec.	Réduit les interruptions de service lors des mises à jour ou améliorations.

Spécifications pour l'implémentation

Les spécifications pour l'implémentation définissent les exigences détaillées pour le développement, le déploiement, et la gestion des composants architecturaux.

Elles assurent que chaque étape de l'implémentation respecte les standards, les objectifs business et techniques de Foosus.

Composant/Aspect	Spécifications	Description
Microservices	<ul style="list-style-type: none"> - Chaque service doit avoir une responsabilité unique (principe SRP). - Utiliser des APIs REST conformes aux standards. 	Assure une modularité et une interopérabilité maximales entre les services.
Frameworks et technologies	<ul style="list-style-type: none"> - Utiliser des technologies éprouvées : <i>Backend</i> : Node.js, Python. <i>Frontend</i> : React, Vue.js. <i>Base de données</i> : PostgreSQL, MongoDB. 	Garantit la cohérence et la maintenabilité grâce à des choix technologiques standards.
Sécurité	<ul style="list-style-type: none"> - Implémenter OAuth 2.0 et JWT pour l'authentification. 	Garantit la confidentialité et la sécurité des données des

	- Utiliser TLS pour le chiffrement des communications.	utilisateurs.
Scalabilité	- Configurer le scaling horizontal pour les services critiques. - Utiliser des services cloud natifs (ex. Kubernetes).	Permet à l'architecture de s'adapter aux pics de trafic et à la croissance de l'activité.
Pipelines CI/CD	- Mettre en place une intégration continue avec tests automatisés. - Configurer des déploiements continus sécurisés.	Réduit les délais de livraison tout en augmentant la qualité des livrables.
Surveillance et monitoring	- Intégrer des outils comme Prometheus, Grafana pour suivre la disponibilité et les performances.	Permet une supervision proactive pour détecter et résoudre rapidement les anomalies.
Base de données	- Optimiser les requêtes pour un temps de réponse ≤ 50 ms. - Mettre en œuvre des sauvegardes automatiques quotidiennes.	Assure une performance élevée et une protection des données en cas d'incident.
Géolocalisation	- Intégrer des services tels que Google Maps API ou OpenStreetMap avec une précision ≥ 95 %.	Fournit une fonctionnalité géociblée fiable pour connecter les utilisateurs aux producteurs locaux.
Tests	- Automatiser les tests unitaires, d'intégration et de charge. - Atteindre une couverture de code ≥ 80 %.	Garantit la qualité et la robustesse des services développés.
Documentation	- Produire une documentation technique complète pour chaque composant et API.	Facilite la collaboration et la prise en charge des services par différentes équipes.

Standards pour l'implémentation

Les standards pour l'implémentation établissent les règles et pratiques essentielles pour garantir une architecture cohérente, maintenable et alignée avec les objectifs stratégiques de Foosus.

Ces standards s'appliquent à tous les aspects du développement, du déploiement, et de la gestion des services.

Domaine	Standard	Description
Développement	Respect des principes SOLID dans la conception des services.	Garantit une architecture modulaire et maintenable.
	Utilisation des conventions de codage	Maintient une cohérence dans le

	reconnues (ex. PEP 8 pour Python, ESLint pour JavaScript).	code et facilite la collaboration entre équipes.
Services et APIs	Adopter les standards REST (GET, POST, PUT, DELETE).	Assure une interopérabilité et une standardisation dans la communication entre services.
	Utilisation de JSON pour les échanges de données.	Facilite l'intégration avec des systèmes tiers et la compréhension des données échangées.
Gestion des versions	Utilisation de Git avec des branches standardisées (ex. <i>main</i> , <i>feature</i> , <i>release</i> , <i>hotfix</i>).	Permet une gestion efficace des versions et des contributions.
	Suivi des versions via un système de gestion sémantique (Semantic Versioning).	Aide à identifier clairement les changements majeurs, mineurs et correctifs dans les services.
Sécurité	Implémentation de mécanismes de sécurité conformes aux recommandations OWASP.	Protège les services contre les vulnérabilités courantes comme l'injection SQL et les attaques XSS.
	Utilisation obligatoire du HTTPS pour toutes les communications.	Garantit la confidentialité et l'intégrité des données échangées.
Base de données	Suivi des meilleures pratiques SQL pour les bases relationnelles (ex. PostgreSQL).	Optimise les performances et réduit les risques d'erreurs.
	Normalisation des bases de données avec des clés primaires et étrangères bien définies.	Assure la cohérence et la robustesse des données.
Tests	Couverture de code minimale : $\geq 80\%$.	Garantit une validation approfondie des fonctionnalités et réduit les régressions.
	Utilisation de frameworks de tests standardisés (ex. Pytest, Jest).	Facilite l'automatisation et la répétabilité des tests.
Déploiement	- Utilisation de pipelines CI/CD avec validation automatisée.	Accélère les cycles de développement et minimise les risques d'erreurs humaines.
	Standardisation des environnements de staging et production pour éviter les incohérences.	Garantit que les tests sont représentatifs de la réalité en production.
Documentation	Adoption de formats standardisés pour la documentation des APIs (ex. OpenAPI/Swagger).	Assure une documentation claire et accessible pour les développeurs internes et externes.
Interopérabilité	Respect des standards ouverts (ex. OAuth 2.0, JSON Web Tokens).	Facilite l'intégration avec d'autres systèmes et technologies.

Surveillance	- Utilisation de métriques standardisées pour surveiller les performances (temps de réponse, disponibilité).	Aide à identifier les goulots d'étranglement et à optimiser les performances.
---------------------	--	---

Conditions requises pour l'interopérabilité

Les conditions requises pour l'interopérabilité définissent les critères nécessaires pour garantir que les différents composants de l'architecture de Foosus, ainsi que les systèmes externes, fonctionnent ensemble de manière cohérente et efficace.

Cela inclut les mécanismes de communication, les protocoles, et les formats de données.

Catégorie	Condition requise	Description
Standards ouverts	Utiliser des standards reconnus (ex. REST, GraphQL, JSON, OAuth 2.0).	Assure une communication universelle et réduit les barrières à l'intégration avec des systèmes tiers.
Format des données	Adopter JSON comme format principal pour les échanges de données.	Simplifie les échanges grâce à un format lisible et standardisé.
Protocole de communication	Implémenter HTTPS pour sécuriser toutes les communications entre les microservices et avec les systèmes externes.	Garantit la confidentialité et l'intégrité des données échangées.
Gestion des API	Standardiser les APIs avec des spécifications OpenAPI/Swagger pour une documentation claire et précise.	Facilite l'intégration et l'utilisation des services par les développeurs internes et externes.
Interopérabilité des microservices	Assurer que chaque microservice respecte les interfaces contractuelles définies et des schémas d'entrée/sortie cohérents.	Maintient la compatibilité et minimise les risques d'intégration entre services.
Sécurité des intégrations	Mettre en œuvre OAuth 2.0 pour l'authentification et l'autorisation des intégrations externes.	Protège les services contre les accès non autorisés.
Versionnement des APIs	Suivre un système de versionnement sémantique (Semantic Versioning) pour les interfaces d'API.	Permet une gestion claire des évolutions et rétrocompatibilités des services.
Tests d'intégration	Automatiser les tests d'intégration pour valider les échanges entre les microservices et avec des services tiers.	Garantit une compatibilité continue lors des mises à jour ou des évolutions des services.

Support multi-plateforme	S'assurer que les services fonctionnent de manière optimale sur les environnements principaux (web, mobile).	Augmente l'accessibilité et l'utilisabilité des services pour les utilisateurs finaux.
Interopérabilité avec les systèmes tiers	Prendre en charge les intégrations avec les principaux outils et plateformes externes (ex. Google Maps API).	Étend les capacités de la plateforme grâce à des fonctionnalités avancées intégrées.

Conditions requises pour le management du service IT

Les **conditions requises pour le management du service IT** définissent les critères nécessaires pour garantir une gestion efficace, proactive et durable des services IT de Foosus.

Ces exigences couvrent les outils, les processus, et les standards à adopter pour assurer une supervision et une maintenance optimales.

Catégorie	Condition requise	Description
Surveillance et monitoring	Mettre en place des outils de monitoring (ex. Prometheus, Grafana) pour surveiller la disponibilité et la performance des services.	Assure une supervision proactive et la détection rapide des anomalies ou des dégradations de performance.
Gestion des incidents	Utiliser un système de gestion des incidents (ex. Jira Service Management) pour suivre et résoudre les problèmes.	Garantit une résolution rapide et structurée des problèmes, avec une traçabilité complète.
Gestion des configurations	Adopter un outil de gestion des configurations (ex. Ansible, Terraform) pour centraliser et standardiser les configurations des systèmes.	Facilite la cohérence et la reproductibilité des environnements IT.
Mises à jour et maintenance	Planifier des fenêtres de maintenance régulières avec notification préalable (ex. 48 heures avant l'intervention).	Minimise les impacts des mises à jour et garantit une communication claire avec les parties prenantes.
Automatisation	Automatiser les processus récurrents, comme les sauvegardes, les mises à jour, et les déploiements.	Réduit les risques d'erreur humaine et accélère les opérations IT.
Gestion des	Implémenter des sauvegardes	Protège les données critiques

données	automatiques quotidiennes avec une conservation des données d'au moins 30 jours.	contre les pertes et assure une reprise rapide en cas d'incident.
Gestion des accès	Mettre en œuvre un contrôle d'accès basé sur les rôles (RBAC) pour les systèmes et services IT.	Garantit que seuls les utilisateurs autorisés peuvent accéder aux ressources sensibles.
Analyse des logs	Centraliser et analyser les logs des services (ex. avec ELK Stack : Elasticsearch, Logstash, Kibana).	Fournit une visibilité sur les activités des systèmes et aide à identifier les problèmes ou anomalies.
Formation continue	Organiser des sessions de formation régulières pour les équipes IT sur les nouveaux outils et standards adoptés.	Renforce les compétences des équipes pour une gestion efficace des services IT.
Rapports périodiques	Générer des rapports mensuels sur les performances, les incidents et les activités de maintenance.	Permet de suivre les indicateurs clés et de partager les informations pertinentes avec les parties prenantes.

Contraintes

Les contraintes identifiées pour le projet d'architecture de Foosus représentent les limites techniques, organisationnelles, et financières qui doivent être prises en compte lors de la conception, de l'implémentation, et de la maintenance de l'architecture.

Ces contraintes influencent directement les choix stratégiques et opérationnels.

Catégorie	Contrainte	Description
Techniques	Dette technique héritée	L'architecture actuelle comporte des dépendances et composants obsolètes, limitant l'agilité et la maintenabilité.
	Compatibilité avec les systèmes existants	L'architecture doit s'intégrer avec les systèmes internes et externes déjà en place.
	Scalabilité technique	Les solutions doivent être conçues pour gérer efficacement des pics de trafic importants sans redéveloppement majeur.
Financières	Budget limité	Les choix technologiques doivent respecter un cadre budgétaire strict, limitant l'adoption de solutions coûteuses.
Organisationnelles	Disponibilité des ressources	Les équipes internes ont des capacités limitées, nécessitant une planification rigoureuse et

	humaines	potentiellement l'implication de partenaires externes.
	Culture agile	L'architecture doit respecter les pratiques de Kanban et d'amélioration continue adoptées par Foosus.
Sécuritaires et légales	Conformité RGPD	L'architecture doit garantir la confidentialité et la sécurité des données personnelles des utilisateurs finaux.
	Sécurité des systèmes	Les solutions doivent intégrer des mécanismes robustes pour protéger contre les cyberattaques.
Dépendances externes	Services tiers	L'architecture dépend de services externes (ex. géolocalisation, paiements), ce qui impose des contraintes de compatibilité et de performance.
	Disponibilité des fournisseurs	Les interruptions des services tiers pourraient avoir un impact sur les performances globales.
Temps	Délais serrés	Le projet doit respecter un calendrier serré pour répondre rapidement aux besoins stratégiques de Foosus.

Hypothèses

Les **hypothèses** établissent les fondements sur lesquels reposeront la conception et l'implémentation de l'architecture de Foosus.

Elles permettront de clarifier les conditions présumées, afin d'aligner les parties prenantes et de guider les décisions tout au long du projet.

Ces hypothèses doivent être validées ou ajustées à mesure que le projet avance.

Hypothèse	Impact	Propriétaire
L'équipe technique actuelle est suffisante pour démarrer le projet.	Les capacités initiales seront suffisantes pour l'implémentation, mais pourraient nécessiter des renforts en cas de montée en charge.	Responsable ingénierie (Pete Parker)
Les fournisseurs tiers (géolocalisation, paiements) resteront disponibles et performants.	Toute indisponibilité ou défaillance pourrait entraîner un ralentissement ou des interruptions des services critiques.	Responsable produit (Daniel Anthony)
Les parties prenantes valideront rapidement les livrables.	Les validations rapides réduiront les retards dans le calendrier global du projet.	Comité d'architecture
Le budget initial est suffisant pour couvrir les phases critiques du projet.	Un dépassement budgétaire pourrait nécessiter une re-priorisation des fonctionnalités.	CEO (Ash Callum)

La dette technique existante est bien définie et gérable dans le cadre du projet.	Une mauvaise estimation pourrait entraîner des retards ou un dépassement des ressources nécessaires.	CIO (Natasha Jarson)
Les utilisateurs finaux accepteront et adopteront les nouvelles fonctionnalités.	Un faible taux d'adoption pourrait compromettre la valeur business de l'architecture livrée.	Équipe produit
L'approche microservices répondra aux besoins de modularité et de scalabilité.	Une défaillance dans la conception des microservices pourrait limiter les performances globales.	Architecte logiciel
Les outils de monitoring et de CI/CD fonctionneront comme prévu.	Des défaillances dans ces outils pourraient ralentir les cycles de déploiement et de résolution.	Responsable DevOps