

Spécification des Conditions requises pour l'Architecture



Projet : Approvisionnement alimentaire géographiquement responsable

Client : Foosus

Préparé par : Clément MARCEL

N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l'Architecture

Date de Version du Document : 01/02/2024

Table des Matières

Objet de ce document	3
Mesures du succès	3
Conditions requises pour l'architecture	4
Contrats de service business	4
Contrats de service application	4
Lignes directrices pour l'implémentation	5
Spécifications pour l'implémentation	6
Standards pour l'implémentation	7
Conditions requises pour l'interopérabilité	7
Conditions requises pour le management du service IT	8
Contraintes	8
Hypothèses	9

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

Métrique	Technique de mesure	Valeur cible	Justification
Nombre d'adhésions d'utilisateurs par jour	Suivi quotidien du nombre d'adhésions d'utilisateurs	Augmentation de 10 %	Stimuler la croissance des utilisateurs de la plateforme
Adhésion de producteurs alimentaires	Analyse des inscriptions de producteurs alimentaires	Atteindre la cible mensuelle	Augmenter la variété des producteurs sur la plateforme
Délai moyen de parution*	Suivi du temps entre la demande et la parution	Réduit de 3,5 semaines à moins d'une semaine	Accélérer le processus de publication des produits
Taux d'incidents de production P1	Suivi des incidents de production de priorité P1	Réduit de >25/mois à moins de 1/mois	Minimiser les incidents critiques de production

Conditions requises pour l'architecture

L'architecture doit :

- Évoluer avec notre base de clientèle
- Utiliser une technologie transparente
- Avoir une évolutivité capable d'accompagner la croissance

Contrats de service business

Les contrats de service business seront accompagnés d'Accords de niveau de service. Ils sont généralement appelés le *service-level agreement* – **SLA** – c'est un document qui définit la qualité de service, prestation prescrite entre par l'entreprise et qui répond aux prérogatives du ou des et un client, dans notre les cas les clients sont les utilisateurs.

Autrement dit, il s'agit de clauses basées sur un contrat définissant les objectifs précis attendus et le niveau de service que souhaite obtenir les utilisateurs de la part de l'entreprise et fixe les responsabilités.

1. Accords de niveau de service (SLA)

Les SLA prévoient généralement des indicateurs afin de mesurer la performance des services dans le but de réaliser une comparaison avec le niveau de qualité des services défini.

SLA	Objectif	Méthode de Mesure	Période d'Évaluation
Disponibilité du Système	99.9% d'uptime mensuel	Surveillance en temps réel	Mensuel
Performance du Temps de Réponse	Temps de réponse moyen < 500 ms	Mesure des requêtes utilisateur	Hebdomadaire
Réussite des Transactions	Taux de réussite global > 99%	Suivi continu des transactions	Quotidien
Capacité et Évolutivité	Supporter 10,000 utilisateurs simultanés	Surveillance de la charge utilisateur	Mensuel

Contrats de service application

Cette section décrit le contrat de service application qui sera fourni dans le cadre de ce projet et intégré au répertoire d'architecture. Dans le cadre de document, la présentation des parties suivantes guidera le lecteur sur ce qui fourni dans le document finale de contrat de service d'application, tout en lui indiquant les objectifs de niveau de service.

1. Objectifs de niveau de service (SLO)

Les Objectifs de niveau de service (SLO) définis pour le projet Foosus représentent les cibles spécifiques que l'architecture et les services associés s'engagent à atteindre. Ces objectifs fixent des seuils quantifiables pour garantir la qualité, la fiabilité et la performance conformément aux attentes des utilisateurs et aux standards de l'architecture d'entreprise.

SLO	Objectif	Critère d'Évaluation	Fréquence d'Évaluation
Disponibilité du Système	99.9% d'uptime	Pourcentage d'uptime effectif	Mensuel
Performance du Temps de Réponse	Temps de réponse moyen < 500 ms	Moyenne des temps de réponse	Hebdomadaire
Réussite des Transactions	Taux de réussite global > 99%	Taux de réussite moyen	Quotidien
Capacité et Évolutivité	Supporter 10,000 utilisateurs simultanés	Charge utilisateur maximale	Mensuel

2. Indicateurs de niveau de service (SLI)

Les Indicateurs de niveau de service (SLI) sont des mesures quantifiables qui permettent d'évaluer la performance et la qualité des services offerts dans le cadre du projet Foosus. Ces indicateurs fournissent une base objective pour évaluer la réalisation des objectifs fixés en termes de disponibilité, de performances et d'expérience utilisateur.

SLI	Méthode de Mesure	Fréquence de Mesure	Outil de Surveillance
Disponibilité du Système	Pourcentage effectif d'uptime	En temps réel	Outils de monitoring système
Performance du Temps de Réponse	Mesure du temps de réponse des requêtes utilisateur	Continu	Analyseurs de performances web
Réussite des Transactions	Suivi continu du taux de réussite	En temps réel	Outils de suivi des transactions
Capacité et Évolutivité	Surveillance du nombre d'utilisateurs simultanés	Continu	Outils de scalabilité

Lignes directrices pour l'implémentation

Les lignes directrices pour l'implémentation du projet Foosus établissent un ensemble de principes directeurs visant à encadrer le processus de mise en œuvre de l'architecture d'entreprise. Ces lignes directrices définissent les règles et les recommandations qui orienteront les équipes tout au long du cycle de vie du projet, assurant ainsi une cohérence, une efficacité et une conformité continues avec les objectifs stratégiques.

Elles s'articulent autour des meilleures pratiques et des normes spécifiques à l'industrie, fournissant un cadre robuste pour guider les décisions et les actions des parties prenantes impliquées dans le processus d'implémentation. Ces lignes directrices visent à garantir la qualité, la flexibilité et l'évolutivité des solutions déployées, tout en favorisant une communication ouverte et continue entre la fonction architecture et les utilisateurs business.

Ci-dessous une liste de bonnes pratiques à mettre en place pour le projet :

- Privilégier des échanges courts mais réguliers plutôt que des échanges longs mais ponctuels (Ateliers de conceptions, Example Mapping, Pair programming, etc). Cette directive permettra aux équipes de rester alignées dans le temps, et de stimuler l'intelligence collective.
- Définir, partager et respecter des règles de développements, de syntaxe, de nommage des fichiers et du code, nomenclatures, qui soient communes afin de renforcer la lisibilité et la clarté.
- Les processus de conception et développement doivent adopter une approche agile tel que DDD (Domain-Driven Design), BDD (Behaviour-Driven Design) et TDD (Test-Driven Development).
- Le code doit refléter le langage du domaine métier de Foosus, de ses consommateurs, fournisseurs, et tout autres utilisateurs des services. Un glossaire des termes du domaine devra être tenu, modifié et discuté autant de fois que nécessaire afin d'accroître une compréhension partagée des termes métiers.
- **Livraison continue** : Livrer en production le plus souvent et plus rapidement possible afin d'avoir du feedback

Spécifications pour l'implémentation

Dans cette partie, nous définirons les spécifications techniques et technologiques pour mieux piloter l'implémentation.

- Langage de programmation : La proposition de langage de programmation est détaillée dans le Document de Développement de l'Architecture, mais l'avis des équipes de développement sera pris en compte
- Frameworks : La solution doit être développée en utilisant des frameworks modernes pour la gestion de la couche d'accès aux données, ainsi que des frameworks de test comme Jest.
- Base de données : La solution doit être compatible avec les bases de données relationnelles telles que MySQL et PostgreSQL. L'utilisation d'une base de données NoSQL telle que MongoDB peut être envisagée pour des cas d'utilisation spécifiques^{7/11}
- API : La solution doit fournir une API RESTful pour permettre l'intégration avec d'autres applications et services tiers. L'API doit être sécurisée à l'aide d'un protocole d'authentification standard tel que OAuth2.
- Interface utilisateur : La solution doit avoir une interface utilisateur conviviale et intuitive. L'utilisation de frameworks de développement d'interface peut être envisagée.
- Sécurité : La solution doit respecter les normes de sécurité de l'industrie pour garantir la confidentialité et l'intégrité des données. Les mesures de sécurité suivantes doivent être mises en place : l'authentification et l'autorisation, le chiffrement des données en transit et au repos, la gestion des certificats et l'audit des logs ainsi que le respect de la RGPD
- Évolutivité : La solution doit être conçue pour être évolutive et capable de gérer des charges élevées. Les mesures d'évolutivité telles que l'utilisation de conteneurs et d'orchestrateurs tels que Docker et Kubernetes peuvent être envisagées.

Standards pour l'implémentation

Les standards sont reconnus comme des solutions appropriées au delà d'une organisation spécifique. De même, les normes sont des cadres internationaux, plus orientés sur le réglementaire.

- Les systèmes d'exploitation utilisés pour toutes nos opérations sont des distributions GNU/Linux, ou leurs images de conteneurs associées.
- Le processus de développement logiciel doivent adopter une approche de **protection des données personnelles** dites "*Privacy by design*"
- Les requêtes et réponses de nos serveurs web, applications et API doivent respecter les normes des codes HTTP définis dans la RFC2616.
- Le développement de nos applications web doivent se prémunir des risques de sécurité les plus critiques, partagées internationalement par OWASP Top 10.
- La qualité de l'entreprise tend à appliquer la norme de management de la qualité ISO 9001 pour améliorer ses processus.
- La sécurité de nos systèmes d'informations doivent appliquer, du moins tendre vers, les normes internationales ISO 27001/27002/27005.

Conditions requises pour l'interopérabilité

L'interopérabilité est cruciale pour le projet Foosus, facilitée par l'utilisation de l'API REST pour standardiser les échanges de données. Pour garantir une intégration fluide, les conditions requises comprennent :

- **Normes et spécifications** : Mise en œuvre de l'API REST en respectant les normes industrielles et les meilleures pratiques architecturales. Utilisation de formats de données standard tels que JSON ou XML.
- **Documentation complète** : Fourniture d'une documentation détaillée comprenant les points de terminaison, les méthodes HTTP, les paramètres acceptés, les formats de données et les restrictions d'accès.
- **Conformité aux standards de sécurité** : Intégration de mesures de sécurité, notamment HTTPS, l'authentification basée sur des jetons, et une gestion fine des autorisations.
- **Support des formats de données courants** : Capacité à traiter différents formats, tels que JSON et XML, pour une interopérabilité avec les systèmes existants.
- **Versioning** : Mise en place d'une stratégie de versioning, incluant un numéro de version dans l'URL de l'API ou l'utilisation d'en-têtes HTTP spécifiques.
- **Tests d'interopérabilité** : Réalisation de tests approfondis pour garantir la conformité, identifier les éventuels problèmes et assurer une intégration harmonieuse.

En respectant ces conditions, Foosus sera en mesure de créer des échanges de données efficaces, améliorant l'intégration et favorisant la collaboration avec les partenaires externes, fournisseurs et autres acteurs de l'écosystème alimentaire durable.

Conditions requises pour le management du service IT

Pour assurer un management efficace du service IT, il est essentiel de mettre en place des pratiques solides pour le monitoring, l'observabilité et la gestion des incidents, ainsi que pour la gestion du personnel. Voici les conditions requises pour le management du personnel dans le cadre du projet de Foosus :

- **Rôles et responsabilités** clairement définis : Chaque membre du personnel doit avoir des rôles et des responsabilités clairement définis en fonction de ses compétences et de son domaine d'expertise. Cela garantit une répartition claire des tâches et des responsabilités, ce qui facilite la coordination et la collaboration au sein de l'équipe.
- **Communication efficace** : Une communication ouverte et transparente est essentielle pour un bon management du service IT. Il est important de mettre en place des canaux de communication clairs et accessibles à tous les membres de l'équipe. Cela peut inclure des réunions régulières, des outils de messagerie instantanée, des plateformes de collaboration en ligne, etc. Une communication efficace favorise la collaboration, résout les problèmes rapidement et maintient tous les membres de l'équipe alignés sur les objectifs du projet.
- **Formation et développement des compétences** : Le management du service IT doit encourager la formation continue et le développement des compétences de son personnel. Cela peut inclure des programmes de formation internes ou externes, des sessions de partage des connaissances, des opportunités de certification, etc. En investissant dans le développement des compétences, l'équipe sera mieux équipée pour gérer les défis techniques et améliorer la qualité globale du service.
- **Gestion du rendement** : Il est important d'établir des critères de performance clairs et d'évaluer régulièrement le rendement de chaque membre de l'équipe. Cela permet d'identifier les forces et les faiblesses individuelles, de reconnaître les performances exceptionnelles et d'offrir un soutien ou une formation supplémentaire si nécessaire. Une gestion efficace du rendement encourage la motivation, l'engagement et l'amélioration continue au sein de l'équipe.
- **Collaboration et esprit d'équipe** : Favoriser un environnement de travail collaboratif et axé sur l'équipe est crucial pour une gestion efficace des services. Cela comprend la promotion d'une culture de partage des connaissances, l'encouragement du travail d'équipe et de la collaboration, et la promotion d'une atmosphère de soutien et d'inclusion. Une dynamique d'équipe et une collaboration solide contribuent à une meilleure résolution des problèmes, à une productivité accrue et à la réussite globale du projet.

En mettant en place ces conditions requises pour le management du personnel, Foosus pourra gérer efficacement son équipe, favoriser la collaboration et le développement des compétences, et maintenir un environnement de travail productif et motivant. Cela contribuera à la réussite globale du projet et à l'atteinte des objectifs de service.

Contraintes

Contraintes	Description
Contraintes de temps	Le projet peut être soumis à des contraintes de délais stricts imposées par les clients, des impératifs commerciaux ou des événements spécifiques. La gestion efficace de ces contraintes est cruciale pour respecter les délais de livraison.

Contraintes	Description
Contraintes budgétaires	Le projet peut être limité par des contraintes budgétaires avec un budget fixe. La gestion optimale des ressources financières est essentielle pour respecter les contraintes budgétaires tout en assurant la qualité et l'efficacité du projet.
Contraintes de ressources humaines	Les ressources humaines disponibles peuvent être limitées en nombre, compétences spécifiques ou disponibilité. Une gestion efficace des ressources humaines garantit que les bonnes personnes sont affectées aux bonnes tâches et que les compétences nécessaires sont présentes.
Contraintes technologiques	Le projet peut être confronté à des contraintes technologiques telles que des limitations matérielles, des incompatibilités entre systèmes ou des exigences techniques spécifiques. Ces contraintes doivent être prises en compte lors de la planification et de l'implémentation.
Contraintes réglementaires	Le projet peut être soumis à des contraintes réglementaires spécifiques, telles que des lois sur la protection des données, des normes de sécurité ou des exigences de conformité sectorielle. La conformité à ces réglementations est essentielle tout au long du projet.
Contraintes de sécurité	La sécurité des données et des systèmes peut imposer des contraintes importantes sur le projet. La mise en place de mesures de sécurité adéquates, telles que des protocoles d'authentification et des mécanismes de protection des données, est essentielle pour assurer la confidentialité et l'intégrité des informations.

Hypothèses

ID	Hypothèse	Impact	Propriétaire
1	Plutôt que d'investir davantage dans la plateforme existante, nous la conserverons en mode de maintenance. Aucune nouvelle fonctionnalité ne sera développée.	Fort	Chef des opérations
2	La nouvelle architecture sera construite en fonction des technologies actuelles et avec la capacité de s'adapter à de nouvelles technologies lorsque celles-ci seront disponibles.	Modéré	Ingénieur en chef

3	Les équipes étant attachées à la plateforme existante, les dirigeants devront éviter de prendre de faux raccourcis en intégrant un nouveau comportement dans le système existant.	Modéré	Directrice des systèmes informatiques
4	L'offre initiale impliquera la coexistence de deux plateformes et la montée en puissance empirique du volume d'utilisateurs qui migreront vers la nouvelle plateforme à mesure que le produit évoluera. Cette augmentation sera proportionnelle à l'évolution des fonctionnalités. Par exemple, les utilisateurs précoces pourront choisir d'utiliser les nouvelles fonctionnalités de recherche intégrées au processus de paiement existant.	Faible	Ingénieur en chef
5	La géolocalisation, si elle est modélisée suffisamment tôt dans la nouvelle plateforme, permettra d'introduire d'autres innovations en fonction de l'emplacement de l'utilisateur ou du fournisseur alimentaire.	Faible	Architecte Logiciel