
Foosus

Spécification des Conditions requises pour l'Architecture

Projet : Développement d'architecture pour soutenir le développement des activités.

Client : Foosus

Préparé par : *Mickaël D. P.*

N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l'Architecture

Date de Version du Document : 27/04/2024

Revu par : [En attente]

Date de Révision : [Date]

Table des Matières

1. 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.

2. Mesures du succès

Pour consulter les mesure du succès veuillez vous référer au document « Déclaration de travail d'architecture » à la section « 8. Critère d'acceptation et procédures – Métriques et KPIs ».

3. Conditions requises pour l'architecture

L'architecture de la plateforme Foosus représente la base essentielle sur laquelle repose toute l'activité visant à connecter les consommateurs avec des producteurs alimentaires locaux de manière durable et efficace. Les conditions requises pour cette architecture sont définies selon deux aspects clés : les exigences fonctionnelles et les exigences techniques. Les critères suivants serviront à s'assurer que la nouvelle architecture réponde aux besoins actuels des utilisateurs ainsi qu'à ceux des besoins futurs.

Exigences Fonctionnelles :

1. **Géolocalisation :** La plateforme doit offrir une fonctionnalité de géolocalisation permettant aux utilisateurs de bénéficier d'une expérience personnalisée et pertinente en fonction de leur position géographique. Cette fonctionnalité est cruciale pour l'activité de proximité de Foosus.

2. **Authentification par rôle :** Les utilisateurs doivent pouvoir se connecter en fonction de leur rôle (consommateurs, fournisseurs, backoffice) et accéder à un ensemble de fonctionnalités dédiées, telles que la personnalisation de leur profil, la gestion des commandes, les offres, ainsi que des outils d'analyse et de recherche.

3. **Outil efficace de recherche :** La plateforme doit proposer un outil de recherche efficace permettant de relier facilement les consommateurs aux fournisseurs selon un principe de démarche simplifiée.

Exigences Techniques :

1. **Scalabilité :** La nouvelle architecture doit être conçue pour supporter une croissance importante du nombre d'utilisateurs et de transactions, tout en étant facilement évolutive en termes de capacités.

2. **Performance :** Dans des conditions normales de fonctionnement, le temps de réponse pour les interactions utilisateurs doit être inférieur à 2 secondes, avec une disponibilité du système de 99,9%.

3. **Sécurité :** Les interactions et transactions des utilisateurs doivent être sécurisées et les données doivent respecter les réglementations applicables, telles que le RGPD pour les données personnelles des utilisateurs.

4. **Mutliversions :** La nouvelle architecture doit permettre la coexistence de différentes versions des modules pour faciliter les transitions sans interruption lors des modifications et innovations.

5. **Innovation :** Comme la plateforme doit être conçue avec une capacité d'adoption rapide de nouvelles technologies afin de maintenir un avantage concurrentiel et d'assurer la meilleure qualité de service, elle sera orientée vers les micro services.

6. **Accessibilité en bande passante limité et lors des pics de fréquentations:** La plateforme doit être accessible même avec une faible bande passante et utiliser

des méthodes d'équilibrage de charge pour assurer un scaling horizontal et vertical en utilisant éventuellement des ressources cloud élastiques pour supporter les pics d'activité.

Ces exigences constituent les fondements des spécifications du projet d'architecture de Foosus. Leur respect permettra de développer une architecture répondant aux besoins actuels et futurs des utilisateurs, assurant ainsi une base solide pour la croissance et l'innovation continue de la plateforme.

4. Contrats de service business

Accords de niveau de service

Les Accords de niveau de service (SLA) jouent un rôle crucial dans l'engagement de Foosus envers la qualité et la fiabilité de son système de production. Imposés pour répondre aux exigences de ses partenaires et utilisateurs, ces SLA déterminent les normes de performance à atteindre pour garantir un fonctionnement optimal de la plateforme. En cas de non-respect de ces engagements, des mesures correctives seront mises en place.

1. **Une haute disponibilité :** visant les 99,9 % et des interruptions ne dépassant pas 2 heures par mois. Tout incident provoquant une perte de disponibilité fera l'objet d'une analyse mesurée.

2. **Assurer l'accessibilité de la plateforme :** sur une gamme variée de connexions Internet, même celles de faible débit. Des tests réguliers sont effectués pour garantir un design responsive adapté, offrant des temps de réponse inférieurs à 2 secondes pour la plupart des types de connexion (notamment au niveau ADSL), ainsi qu'une expérience utilisable même dans des conditions de connexion plus limitées.

3. **Garantir la sécurité de la solution :** contre les fuites de données personnelles en traçant les logs des échanges systèmes et API.

4. **Définir les attentes (livraison, qualité, performance) :** Foosus s'engage à respecter les délais de livraison convenus, à garantir la qualité des produits locaux proposés sur la plateforme, et à assurer des performances optimales en termes de temps de chargement et de réactivité.

5. **Objectif de Point de Restauration (RPO) :** Le temps de perte de données ne devrait pas excéder 10 minutes. Des tests seront effectués pour garantir la

robustesse de la plateforme.

6. **Objectif de Temps de Récupération (RTO) :** La plateforme ne devrait pas subir de perturbation excédant une heure. Des tests seront effectués pour garantir la robustesse de la plateforme.

5. Contrats de service application

Objectifs de niveau de service

Les objectifs de niveau de service pour les applications Foosus sont conçus pour garantir une expérience utilisateur de qualité. Ils incluent :

1. **Disponibilité :** Les applications doivent avoir une disponibilité de 99,9 % du temps, en dehors des fenêtres de maintenance planifiées, afin d'assurer un accès constant aux services.

2. **Performance :** Le temps de réponse, en dehors des pics de charge exceptionnelle, doit être maintenu sous les 2 secondes pour garantir une expérience agréable. En cas de pic particulièrement élevé, le maintien du service prime sur le délai de réponse.

3. **Sécurité :** Les applications doivent garantir la sécurité des transactions avec la plateforme. Elles seront régulièrement mises à jour pour combler les failles connues et respecter les normes de sécurité les plus récentes, notamment en ce qui concerne les données utilisateurs.

4. **Maintenance :** Les applications doivent être maintenues à jour pour intégrer les dernières technologies et fonctionnalités, éviter les bugs et les défaillances de services. En cas de maintenance nécessaire, une fenêtre d'indisponibilité sera communiquée à l'avance aux utilisateurs.

Indicateurs de niveau de service

Pour chaque objectif, des indicateurs spécifiques permettront de mesurer leur réalisation par les applications, tels que :

1. **Taux de Disponibilité :** Calculé en temps réel, ce taux indiquera le pourcentage de disponibilité des applications.

2. **Temps de Réponse Moyen :** Des audits périodiques sur le temps de réponse seront effectués et l'ensemble des informations sur les échanges du système d'information disponibles seront analysées pour mesurer ce temps.

3. **Incidents de Sécurité :** Le monitoring des activités et des journaux d'activités permettra de détecter d'éventuels problèmes de sécurité et d'en identifier les origines.

6. Lignes directrices pour l'implémentation

Ces lignes directrices fournissent un cadre général pour la mise en œuvre de l'architecture de Foosus afin de conserver une approche cohérente et efficace :

- **Règles pour les choix des solutions :**
  - Privilégier les solutions open source par rapport aux solutions payantes.
  - Tenir compte du support continu des composants lors de leur sélection.
  - Favoriser l'utilisation de solutions faisant partie d'une même pile technologique pour réduire les coûts de maintenance.
- **Remboursement de la dette technique :**
  - Ne plus engranger de dette technique.
  - Identifier et rembourser progressivement la dette technique des pratiques passées.
  - Réaliser des actions de refonte de composants système et l'amélioration de la qualité du code.
- **Approche agile :**
  - Adopter une méthodologie agile pour permettre une adaptation rapide aux changements et une livraison continue de valeur aux utilisateurs.
- **Philosophie Lean :**
  - Maintenir une approche Lean dans les processus de développement pour maximiser la valeur tout en minimisant le gaspillage.
- **Revue des modules et révision de Code :**
  - Effectuer une révision systématique des modules et du code pour garantir la qualité et la sécurité des éléments intégrés à la plateforme**.**
- **Intégration Continue/Déploiement Continue (CI/CD) :**
  - Développer et maintenir une suite complète de tests automatisés pour garantir la fiabilité et la robustesse des applications.
  - Mettre en place des pipelines CI/CD pour automatiser les tests et les déploiements, optimisant ainsi les cycles d'intégration.

- **Conteneurisation :**
  - Adopter une approche basée sur les conteneurs, tels que Docker ou Kubernetes, pour faciliter le déploiement et la gestion des applications de manière portable et scalable.
  - Mettre en place des pratiques de conteneurisation pour isoler les applications, leurs dépendances et leurs configurations, permettant ainsi une gestion efficace des environnements de développement, de test et de production.
- **Standardisation des Technologies :**
  - Utiliser des technologies standardisées pour garantir la cohérence et la compatibilité de la stack technologique.
- **Sécurité des Données :**
  - Mettre en place des mesures de sécurité strictes pour protéger les utilisateurs et leurs données personnelles, en utilisant des protocoles sécurisés (comme HTTPS) et en chiffrant les données sensibles.
- **Internationalisation :**
  - Prendre en compte les besoins d'internationalisation dès la conception de la plateforme, en tenant compte des différences de langues, de devises et de fuseaux horaires pour assurer une expérience utilisateur cohérente à l'échelle mondiale.
- **Exploitation des données pour la Business Intelligence :**
  - Exploiter les données générées par la plateforme, en particulier les données de géolocalisation, pour améliorer les services et identifier de nouvelles opportunités, tout en respectant strictement la vie privée des utilisateurs et en se conformant à la réglementation locale.

7. Spécifications pour l’implémentation

Ces spécifications détaillent les exigences techniques précises pour la mise en œuvre de l'architecture de la solution Foosus, en intégrant les meilleures pratiques et les innovations pertinentes.

Architecture :

- **Microservices :** La nouvelle architecture de Foosus reposera sur une structure en microservices, favorisant la scalabilité, la maintenance des applications et l'intégration de nouvelles technologies. Les composants seront conçus avec un couplage faible et une forte cohésion pour permettre l'ajout de nouvelles fonctionnalités sans perturber les existantes.

- **Conteneurisation :** Les microservices seront conteneurisés à l'aide de Docker et Kubernetes pour faciliter le déploiement, la gestion et la mise à l'échelle des applications de manière efficace et portable.
- **Base de données :** Chaque microservice de Foosus disposera de sa propre base de données, configurée selon les besoins spécifiques de son domaine fonctionnel. Pour garantir la sécurité et la confidentialité des données, des mesures strictes seront mises en place, notamment :
  - Chaque base de données sera isolée et accessible uniquement par le microservice associé, réduisant ainsi les risques de fuites de données et de violations de sécurité.
  - Les politiques de gestion des accès seront strictement appliquées, limitant l'accès aux données uniquement aux utilisateurs autorisés selon leurs rôles et responsabilités.
- **APIs et Intégration :** Les microservices nécessiteront l'utilisation d'API, privilégiant une architecture RESTful pour assurer la compatibilité et la sécurité des échanges. Les services externes de Google Maps Platform pourrot par exemple être intégrés pour la géolocalisation des offres alimentaires.

Sécurité :

- **Contrôle d'Accès :** Un contrôle d'accès basé sur les rôles (Oauth et connectID) sera implémenté, avec une communication chiffrée et le chiffrement des données au repos pour garantir la sécurité des utilisateurs et de leurs données.
- **Intégration de Solutions et Sécurité Cloud :** Les services cloud, notamment AWS, seront utilisés pour améliorer la scalabilité et la flexibilité de la plateforme, avec AWS RDS pour les bases de données relationnelles, AWS Lambda pour l'exécution de microservices sans serveur et AWS IAM pour la gestion sécurisée des accès.

Patterns architecturaux :

- **Gateway Routing Pattern :** Un API Gateway servira de point d'entrée unique pour router les requêtes vers les microservices appropriés, assurant ainsi une gestion centralisée des flux**.**
- **Strangler Fig Pattern :** Ce pattern facilitera la migration progressive du système vers une nouvelle architecture en remplaçant progressivement les fonctionnalités existantes par de nouvelles implémentations au sein des microservices.

8. Standards pour l’implémentation

Les standards d’implémentation assurent une meilleure compatibilité et réduisent les risques lors des intégrations. Ils sont essentiels pour garantir la cohérence et la fiabilité de la nouvelle architecture de la solution Foosus.

- **Conventions de codage:** Il est crucial d’établir des conventions de codage pour assurer la cohérence à travers l’ensemble des éléments de la plateforme. Cela facilite la maintenabilité grâce à des normes sur la nomenclature, la structure et les commentaires. En favorisant l’interopérabilité entre les différentes parties du système, cela permet également l’intégration de nouvelles fonctionnalités et les interactions avec les services tiers.
- **Utilisation de frameworks populaires:** L’utilisation de frameworks et de bibliothèques soutenant l’architecture de microservices avec une communauté active et nombreuse garantit une meilleure robustesse du système. De plus, cela favorise la compatibilité avec d’autres productions basées sur le même framework, facilitant ainsi l'évolutivité et la maintenance de la solution.
- **Documentation du code:** La documentation du code et des éventuelles API du système est indispensable pour assurer une littérature claire et accessible. L’usage d’outils et de conventions pour la rédaction de ces documents, comme Swagger pour les API REST, est recommandé. Ces documents doivent être mis à jour périodiquement pour permettre aux équipes d’avoir des informations à jour et de garantir la compréhension et l'évolutivité du système.
- **Normes de sécurité:** Pour contribuer à l’amélioration continue de la sécurité de la plateforme, l’intégration de normes de sécurité telles que OWASP est primordiale. Cela permet de suivre de près les besoins de sécurisation des applications web contre les vulnérabilités communes et de garantir la protection des données sensibles de l'entreprise et de ses utilisateurs.
- **Normes d’interopérabilité:** Adhérer aux standards ouverts et aux protocoles bien établis assure une grande interopérabilité avec des éléments ou systèmes tiers. Cela facilite l’intégration de divers modules et l’échange de données avec d’autres services. L’adoption de patterns architecturaux peut également contribuer à renforcer l'interopérabilité de la solution Foosus et à assurer sa compatibilité avec l'écosystème technologique existant.

9. Conditions requises pour l’interopérabilité

L’interopérabilité est cruciale pour Foosus, permettant à notre plateforme de se connecter et d’interagir efficacement avec une variété de services externes, outils et données, tout en assurant une intégration harmonieuse entre nos propres services internes. Pour réaliser cette vision, les conditions suivantes doivent être mises en place :

- **Standards ouverts et protocoles :** Adopter des standards ouverts et des protocoles bien établis pour les communications entre services, tels que HTTP, REST pour les APIs web, afin de pouvoir interagir avec une large gamme d'autres systèmes et services.
- **Formats de données normalisés :** Utiliser des formats de données normalisés et largement adoptés, tels que JSON pour les échanges de données via API, pour faciliter l'intégration et l’interopérabilité avec d’autres systèmes et services.
- **Authentification et autorisation :** Mettre en œuvre des mécanismes d'authentification et d'autorisation standardisés, tels qu'OAuth 2.0 et OpenID Connect, voire AWS IAM, pour sécuriser les communications entre différents services tout en permettant une gestion flexible des accès.
- **APIs documentées et versionnées :** Fournir une documentation claire et accessible pour toutes les APIs, en utilisant des outils comme Swagger, afin de permettre des mises à jour et des évolutions sans interruption des services existants.
- **Patterns de conception pour l’interopérabilité :** Employer des patterns de conception architecturaux qui favorisent l’interopérabilité, comme les API gateways pour centraliser et gérer les requêtes externes ou le pattern de Strangler fig pour faciliter la migration et l'intégration progressive de nouveaux systèmes ou services.
- **Tests d’Intégration :** Développer des suites de tests d'intégration pour vérifier la compatibilité et la communication correcte entre les différents composants et services, à la fois internes et externes.
- **La conteneurisation des microservices :** A l’aide de Docker et Kubernetes pour offrir un environnements portable pour le déploiement et a gestion des applications implémentés à la plateforme

Cette approche devrait garantir une évolutivité durable et une capacité d’innovation continue en permettant l’utilisation généralisée des meilleures pratiques à un moment donné.

10. Conditions requises pour le management du service IT

La gestion efficace des services IT est indispensable pour garantir la disponibilité, la performance et la sécurité de la plateforme Foosus. Pour atteindre ces objectifs, les conditions suivantes doivent être remplies :

- **Monitoring et reporting en temps réel:** Utilisation d'outils de surveillance en temps réel pour suivre la performance et l'utilisation de la plateforme, permettant ainsi de réagir rapidement aux incidents de performance ou de sécurité. Cela inclut l'automatisation des processus de mitigation.
- **Gestion des incidents :** Mise en place de processus clairs pour la gestion des incidents, comprenant chaque étape, de la détection à la résolution à court et long terme. Un système de tickets peut être utilisé pour gérer les incidents utilisateurs.
- **Gestion des besoins de changements :** Établissement de procédures claires pour la gestion des changements une fois identifiés, incluant l'évaluation, l'approbation et l'implémentation des modifications tout en minimisant les interruptions de service.
- **Planification et Automatisation de la Scalabilité :** Préparation et automatisation des mécanismes de scalabilité pour permettre à la plateforme Foosus de s'adapter automatiquement à la demande, aussi bien lors de pics de charge que de façon continue pour accompagner la croissance. L'utilisation de services tels qu'AWS Auto Scaling, Google Cloud AutoML et Azure Autoscale prendre en charge cette automatisation.
- **Adhésion à des politiques de sécurité informatique strictes :** Conformité aux standards de l'industrie et aux réglementations légales pour établir des routines de tests de sécurité et corriger rapidement les vulnérabilités.
- **Maintenance proactive :** Mise en place d'une maintenance proactive et d'un soutien technique pour tous les systèmes et services, incluant les mises à jour régulières des logiciels et du système d'exploitation.

11. Contraintes

La réalisation de l'architecture IT de Foosus est soumise à plusieurs contraintes critiques qui doivent être prises en compte tout au long du processus de développement pour assurer le succès du projet. Ces contraintes comprennent :

- **Budget :** Un budget prédéfini limite les investissements dans de nouvelles technologies, des outils et du personnel. Cette contrainte exige une allocation efficace des ressources et la recherche de solutions offrant un bon rapport qualité-prix. De plus, le budget du prototype est pertinent uniquement s'il exclut la rémunération des forces de travail qui lui seront attribuées en dehors de sa conception.
- **Délais :** Les délais pour le déploiement de certaines fonctionnalités ou pour la réalisation complète du projet peuvent restreindre la portée des travaux de développement ou exiger des phases d'implémentation accélérées.
- **L'existant :** La nécessité de s'intégrer ou de coexister avec des systèmes IT existants peut limiter les choix de conception ou imposer l'utilisation de certaines technologies, même si de meilleures alternatives sont disponibles.
- **Réglementations :** Les exigences légales et réglementaires, telles que le RGPD pour la protection des données personnelles, la réglementation sur la protection des consommateurs, notamment le droit de rétractation, ainsi que la réglementation sur la sécurité et la traçabilité alimentaire, doivent être strictement respectées.
- **Dépendances :** L'usage de services tiers ou open source peut entraîner des contraintes liées à la disponibilité, à la performance et aux délais de mise à jour, entre autres.

14. Hypothèses

Pour consulter les hypothèses veuillez vous référer au document « Déclaration de travail d'architecture » à la section « 8. Risques et facteurs de réductions – Hypothèses».