



## Spécification des Conditions requises pour l'Architecture



Descriptif du projet : Spécification des Conditions requises pour  
l'Architecture  
*Foosus géo-conscient*

Conception d'une nouvelle architecture afin de soutenir le développement d'une entreprise	
<b>Client</b>	Foosus
<b>Nom du projet</b>	Projet OC-5-19042021
<b>Préparé par</b>	Mitra IZADI
<b>N° de version du document</b>	0.4
<b>Titre</b>	Spécification des Conditions requises pour l'Architecture
<b>Date de version du document</b>	20/07/2021
<b>Revu par</b>	Monsieur Eric LEGBA
<b>Date de révision</b>	23/05/2021
<b>Liste de distribution</b>	
<b>De</b>	
<b>Date</b>	
<b>Email</b>	mitraizaid65@gmail.com
<b>Pour Action</b>	Information
<b>Date de rendu</b>	23/10/2021
<b>Types d'action</b>	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion, Autre (à spécifier)
<b>Historique de versions du document</b>	Voir git



<b>1</b>	<b>Objet de ce document.....</b>	<b>4</b>
<b>2</b>	<b>Mesures du succès.....</b>	<b>4</b>
<b>3</b>	<b>Conditions requises pour l'architecture .....</b>	<b>4</b>
<b>4</b>	<b>Contrats de service business.....</b>	<b>5</b>
4.1	Accords de niveau de service .....	5
<b>5</b>	<b>Contrats de service application .....</b>	<b>5</b>
5.1	Objectifs de niveau de service.....	5
5.2	Indicateurs de niveau de service .....	6
<b>6</b>	<b>Lignes directrices pour l'implémentation .....</b>	<b>6</b>
<b>7</b>	<b>Spécifications pour l'implémentation .....</b>	<b>7</b>
<b>8</b>	<b>Standards pour l'implémentation .....</b>	<b>9</b>
<b>9</b>	<b>Conditions requises pour l'interopérabilité.....</b>	<b>9</b>
<b>10</b>	<b>Conditions requises pour le management du service IT .....</b>	<b>10</b>
<b>11</b>	<b>Contraintes.....</b>	<b>10</b>
<b>12</b>	<b>Hypothèses.....</b>	<b>10</b>
<b>13</b>	<b>Annexes.....</b>	<b>11</b>

## 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

### Indicateurs de réussite

Indicateur	Changement souhaité pour l'indicateur
Nombre d'adhésions d'utilisateurs par jour	Augmentation de 10 %
Adhésion de producteurs alimentaires	Passer de 1,4/mois à 4/mois
Délai moyen de parution*	Réduit de 3,5 semaines à moins d'une semaine
Taux d'incidents de production P1	Pour commencer : réduit de >25/mois à moins de 1/mois.

## 3 Conditions requises pour l'architecture

1. Tirer parti de la géolocalisation pour relier des fournisseurs et des consommateurs et pour proposer des produits disponibles près des lieux de résidence de ces derniers. Un calculateur de distance devra être inclus pour permettre aux consommateurs de trouver les fournisseurs les plus proches d'eux.
2. L'architecture devra être évolutive pour que nous puissions déployer nos services sur diverses régions, dans des villes et des pays donnés.
3. Les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

E-mail: [contact@foosus.com](mailto:contact@foosus.com) - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24

4. Nos fournisseurs et nos consommateurs doivent pouvoir accéder à notre solution où qu'ils se trouvent. Cette solution doit être utilisable avec des appareils mobiles et fixes. Elle doit tenir compte des contraintes de bande passante pour les réseaux cellulaires et les connexions Internet haut débit.
5. Elle doit pouvoir prendre en charge divers types d'utilisateurs (par exemple, fournisseurs, back-office, consommateurs), avec des fonctionnalités et des services spécifiques pour ces catégories.
6. Les livrables doivent pouvoir être fournis à intervalles réguliers pour que le nouveau système soit rapidement opérationnel et puisse être doté de nouvelles fonctionnalités au fil du temps.

## 4 Contrats de service business

### 4.1 Accords de niveau de service

Le service-level agreement (SLA) ou « entente de niveau de service » est un document qui définit la qualité de service, prestation prescrite entre un fournisseur de service et un client. Autrement dit, il s'agit de clauses basées sur un contrat définissant les objectifs précis attendus et le niveau de service que souhaite obtenir un client de la part du prestataire et fixe les responsabilités.

1. Tirer parti de la géolocalisation pour relier des fournisseurs et des consommateurs et pour proposer des produits disponibles près des lieux de résidence de ces derniers. Un calculateur de distance devra être inclus pour permettre aux consommateurs de trouver les fournisseurs les plus proches d'eux.
2. Les utilisateurs situés dans différentes régions doivent pouvoir espérer des performances similaires. Nous voulons cibler les consommateurs dans des zones géographiques spécifiques, sur des connexions lentes (par exemple, avec des téléphones portables) aussi bien que sur des réseaux haut débit. Toutes les solutions doivent pouvoir répondre à cette exigence.

## 5 Contrats de service application

### 5.1 Objectifs de niveau de service

Le but est de libérer la créativité et l'expérience de l'équipe technique. Il faut leur permettre de donner le meilleur d'elles-mêmes en créant une nouvelle plateforme qui pourra faire franchir le prochain million d'utilisateurs inscrits à la base de clientèle. On va impulser des campagnes de marketing Foosus dans plusieurs grandes villes en étant sûrs que la plateforme restera utilisable et réactive, tout en offrant une expérience utilisateur

de premier plan.

On ne peut pas abandonner les outils actuels pendant qu'on en élabore de nouveaux car cela impliquerait la mise hors service de la plateforme existante. Pour pouvoir continuer à accepter de nouvelles adhésions de fournisseurs et de consommateurs, il faut dissocier les nouvelles livraisons de l'architecture et de l'infrastructure existantes afin de limiter les interruptions de service.

La nouvelle plateforme devra également permettre à l'équipe produite d'innover rapidement en réorientant des solutions existantes, en expérimentant de nouvelles modifications et en facilitant l'intégration avec des partenaires internes et externes.

## 5.2 Indicateurs de niveau de service

Le niveau de service requis, ou en anglais service-level requirement (SLR), est une définition, en fonction d'indicateurs prédéfinis, du niveau de service convenu entre un fournisseur de service et son client.

Lors de la création de back logs, les équipes produites peuvent développer à partir de comportements spécifiques, mais toute nouvelle conception doit cependant tenir compte des éléments suivants :

- 1- Emplacement des offres alimentaires proposées par les fournisseurs.
- 2- Proximité de l'utilisateur effectuant la recherche en cours.
- 3- Visualisation des informations statistiques secondaires et sectorielles relatives au produit alimentaire concerné. Par exemple, détails sur son indice glycémique.

## 6 Lignes directrices pour l'implémentation

- 1- Pour pouvoir continuer à accepter de nouvelles adhésions de fournisseurs et de consommateurs, nous devons en outre dissocier les nouvelles livraisons de l'architecture et de l'infrastructure existantes afin de limiter les interruptions de service.
- 2- L'an dernier, 12 des pannes ont été provoquées par la publication par une ou plusieurs équipes de modifications lourdes qui n'ont pas eu les résultats escomptés. L'entreprise a également eu des difficultés à intégrer les travaux réalisés par différentes équipes sur des modifications de la plateforme qui n'avaient pas de lien entre elles. Il ne faut pas avoir ce genre de problème en tant que petite entreprise. La difficulté vient du temps nécessaire pour que chaque nouvelle version logicielle soit vue par les autres équipes ou testée dans nos environnements de production. Il faut combler le fossé entre le moment où une ligne de code est écrite et celui où elle est validée dans un environnement intégré. Cela permet

également d'aider l'équipe à déterminer les réactions des clients vis-à-vis de nouvelles fonctionnalités au fur et à mesure qu'elle développe ces dernières.

## 7 Spécifications pour l'implémentation

1. **Déploiements indépendants.** Il est tout à fait possible de mettre à jour un service sans redéployer toute l'application et annuler ou restaurer par progression une mise à jour en cas de problème. Les résolutions de bogues et les publications de fonctionnalités doivent être plus faciles à gérer et moins risquées.
2. **Développement indépendant.** Il est nécessaire de créer des équipes de développement uniques pour chaque service afin de favoriser l'innovation continue et un rythme de publication plus élevé.
3. **Résilience.** Le système doit résister aux pannes de n'importe quel composant, telles que les pannes de serveur, les pannes de disque dur, les partitions réseau, etc. Tirer parti de la réplication empêche la perte de données et permet à un service de continuer à utiliser les instances restantes. Tirer parti de l'isolation empêche les échecs en cascade.
4. **Élasticité.** Vous pouvez vous attendre à ce que la charge varie considérablement au cours de la durée de vie du service. Il est essentiel de mettre en œuvre une scalabilité dynamique et automatique, à la fois vers le haut et vers le bas, en fonction de la charge.
5. **Utilisation simple.** La simplicité de l'architecture améliore son efficacité, sa gestion et sa maintenance.
6. **Sécurité.** Une architecture logicielle doit reposer sur un système qui dispose de contrôles de sécurité intégrés aux composants matériels et logiciels, ainsi que de protections supplémentaires contre les menaces telles que les attaques DDoS (distributed denial-of-service attack).
7. **Le coût** est l'un des aspects pertinents à prendre en compte lors du choix d'une architecture. Nous devons choisir une architecture dont le système dans lequel elle est intégrée nous offre des avantages pour une utilisation efficace et à moindre coût des ressources.

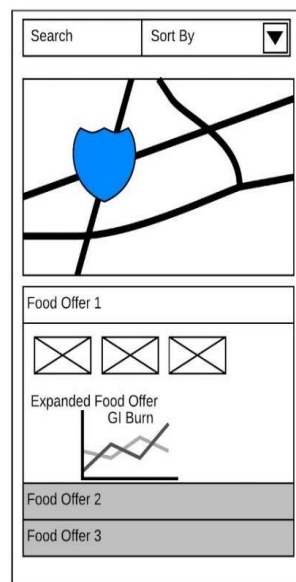
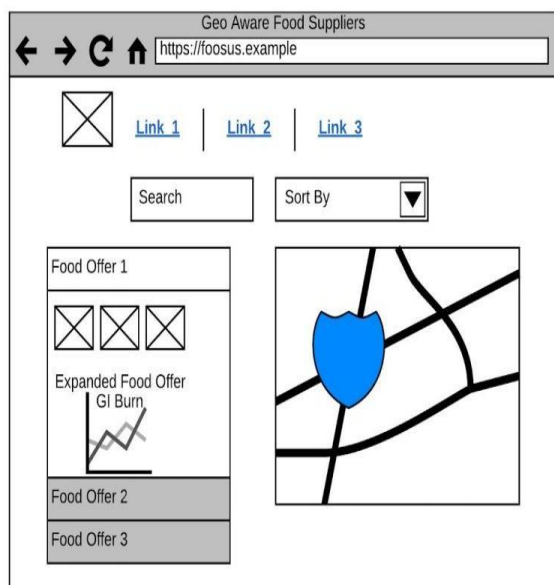
Les premiers tests de wireframes se sont traduits par les deux structures suivantes de point de rupture pour les appareils fixes et mobiles, tous deux privilégiant la proximité entre un

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

E-mail: [contact@foosus.com](mailto:contact@foosus.com) - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24

fournisseur et l'utilisateur.



Lors de la création de backlogs, les équipes produites peuvent développer à partir de comportements spécifiques, mais toute nouvelle conception doit cependant tenir compte des éléments suivants :

- Emplacement des offres alimentaires proposées par les fournisseurs.
- Proximité de l'utilisateur effectuant la recherche en cours.
- Visualisation des informations statistiques secondaires et sectorielles relatives au produit alimentaire concerné. Par exemple, détails sur son indice glycémique.

Nous prévoyons de faire reposer les nouvelles conceptions sur le processus tri des offres alimentaires existant, qui comporte les étapes suivantes :

- Recherche et identification des produits alimentaires requis.
- Ajout des offres alimentaires au panier.
- Recherche d'un accord pour payer à la livraison.
- Instructions de livraison et facture de la commission par e-mail au fournisseur alimentaire.

La vision du produit à long terme consiste à modifier ces deux dernières étapes afin que nous puissions :



- L'intégrer à des prestataires de paiement tiers ;
- Gérer toutes les communications avec les fournisseurs alimentaires au sein d'une interface utilisateur personnalisée.

## 8 Standards pour l'implémentation

Les premières études sur les meilleures pratiques en matière d'architecture en font apparaître plusieurs qui présentent des risques techniques réduits. Il s'agit notamment de micro services potentiels, de normes prenant en charge des solutions Web et mobiles, de bases de données standard et d'autres approches similaires.

## 9 Conditions requises pour l'interopérabilité

L'interopérabilité sera garantie avec l'adoption de l'architecture des micro-services. Avec cette architecture, la migration de l'application actuelle vers l'application cible peut être réalisée par étapes puisque chaque micro-service peut utiliser différentes technologies, langages de programmation et bases de données.

Il est très important d'examiner attentivement le choix du mécanisme de messagerie lorsqu'il s'agit d'architecture de micro-service. Si cet aspect est ignoré, cela peut compromettre l'ensemble de l'objectif de la conception d'une architecture de micro-service.

Pour implémenter ce projet, mon choix se porte sur la Messagerie Asynchrone.

Si le couplage faible est important, en particulier dans un système qui nécessite une résilience élevée et dont l'échelle est imprévisible, la meilleure option est la messagerie asynchrone.

La messagerie asynchrone est une approche fondamentale pour intégrer des systèmes indépendants ou pour construire un ensemble de systèmes faiblement couplés qui peuvent fonctionner, évoluer et évoluer de manière indépendante et flexible. La messagerie asynchrone et la communication pilotée par les événements sont des fonctionnalités essentielles lors de la propagation de changements sur plusieurs micro-services et leurs modèles de domaine connexes.

## 10 Conditions requises pour le management du service IT

Les équipes produites aimeraient pouvoir exécuter diverses variantes ou réaliser des comparaisons de différentes solutions auprès des utilisateurs.

Pour y parvenir, l'entreprise a besoin de visibilité sur la façon dont les logiciels sont utilisés et il faut pouvoir inverser des décisions d'architecture tant que cela reste peu onéreux. Ou alors répliquer sur une plateforme qui permette à l'entreprise d'essayer de nouveaux produits d'une façon compatible avec ses objectifs commerciaux fondamentaux.

## 11 Contraintes

Ci-après figure une liste des contraintes relatives au projet approuvé.

1. Le projet initial est approuvé pour un coût de 50 000 USD (45 190 €) et une période de 6 mois est prévue pour définir l'architecture et préparer un projet de suivi afin de développer un prototype.
2. L'architecture doit permettre d'obtenir le meilleur rapport qualité-coût.
3. L'architecture peut inclure de nouveaux composants personnalisés ou des composants du commerce pour favoriser la flexibilité, la stabilité et l'extensibilité.

L'objectif de cette phase du projet étant la définition de l'architecture, des projets de suivi seront créés pour compléter les détails avec les équipes internes.

## 12 Hypothèses

1. Plutôt que d'investir davantage dans la plateforme existante, nous la conserverons en mode de maintenance. Aucune nouvelle fonctionnalité ne sera développée.
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.
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.
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 au fur et 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.
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.



## 13 Annexes

[https://en.wikipedia.org/wiki/Service-level\\_agreement](https://en.wikipedia.org/wiki/Service-level_agreement)

[https://en.wikipedia.org/wiki/Denial-of-service\\_attack](https://en.wikipedia.org/wiki/Denial-of-service_attack)