



Spécification des Conditions requises pour
l'Architecture

Projet : Foosus Géoconscient

Client : Foosus

Préparé par : Ludovic SOUPLET

N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l'Architecture

Date de Version du Document : 20 septembre 2023

Revu par :

Date de Révision :

Table des matières

<i>Objet de ce document</i>	4
<i>Mesures du succès</i>	5
<i>Conditions requises pour l'architecture</i>	6
<i>Contrats de service business</i>	7
Objectifs de niveau de service.....	7
Accords de niveau de service	8
Informers les utilisateurs des interruptions de service planifiées ou imprévues.....	8
<i>Contrats de service application</i>	8
Objectifs de niveau de service.....	8
Indicateurs de niveau de service	9
Assurer la conformité aux réglementations applicables	9
<i>Lignes directrices pour l'implémentation</i>	10
<i>Spécifications pour l'implémentation</i>	10
<i>Standards pour l'implémentation</i>	11
<i>Conditions requises pour l'interopérabilité</i>	13
<i>Conditions requises pour le management du service IT</i>	13
<i>Contraintes</i>	14
<i>Hypothèses</i>	15

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

La mesure du succès de la nouvelle plateforme se fera au travers des indicateurs ci-dessous :

Métrique	Technique de mesure	Valeur cible	Justification
Nombre d'adhésions d'utilisateurs par jour	Requêtage sur la plateforme et affichage sur un dashboard de suivi	Augmentation de 10%	Recul des inscriptions sur les derniers mois. C'est l'indicateur clé de la réussite du projet
Adhésion de producteurs alimentaires	Requêtage sur la plateforme et affichage sur un dashboard de suivi	Passer de 1,4/mois à 4/mois	Plus de producteurs c'est une attractivité supplémentaire pour attirer de nouveaux clients
Délai moyen de mise en production	Modification automatique de la date de « dernière MEP » sur le dashboard de suivi. Alerte en cas de dépassement de la valeur cible et valeur cible +2j	Réduit de 3,5 semaines à moins d'une semaine	Réduire la taille des MEP afin de réduire les impacts sur la plateforme Retour utilisateur plus aisé sur les fonctionnalités car moins « perdues » dans un lot de nouvelle fonctionnalités
Taux d'incidents de production P1	Suivi de l'uptime via les outils du clouder choisi (cloudwatch par exemple pour AWS)	Réduit de >25/mois à moins de 1/mois	Augmentation de la satisfaction clientèle et de la réputation de l'application
Délai de réponse des requêtes	Suivi du délai de réponses via les outils du clouder choisi (cloudwatch par exemple pour AWS)	Moins de 1 seconde en moyennes	Satisfaction utilisateurs Compatibilité maximum avec les connexions lentes et performances similaires entre les utilisateurs
Taux de conversion après une recherche	Requêtage sur la plateforme et affichage sur un dashboard de suivi	75%	Actuellement, le taux de conversion après une recherche est de 52%, c'est trop bas

Conditions requises pour l'architecture

- L'architecture doit être conçue pour optimiser le rapport qualité-prix
- L'architecture peut inclure de nouveaux composants personnalisés ou des composants commerciaux pour favoriser la flexibilité, la stabilité et l'extensibilité
- Les solutions open source sont préférées aux solutions payantes
- Il faut tenir compte du support continu des composants lors de leur sélection ou de la décision d'achat
- Dans la mesure du possible, toutes les solutions, qu'elles soient commerciales ou open source, doivent faire partie d'une même pile technologique afin de réduire les coûts de maintenance et de support continu
- L'architecture doit prendre en compte le double run avec l'architecture historique le temps que la migration soit effective
- L'architecture doit prendre en compte l'internationalisation de la plateforme
- L'architecture doit être scalable
- Les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.
- 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.

Contrats de service business

Objectifs de niveau de service

Permettre aux parties prenantes qui interagissent avec le process business :

- Les fournisseurs alimentaires soumettront à Foosus un inventaire des produits alimentaires disponibles
- Recherche dans l'interface client et commande de produits de consommation.
- Les clients des produits de consommation trouveront et commanderont des produits alimentaires
- Les fournisseurs alimentaires recevront des commandes
- L'équipe finance de Foosus recevra les paiements

Accords de niveau de service

ID	Objectifs	SLA
1.	Assurer que l'application est disponible pour les utilisateurs	Une disponibilité de 99,9 %, ce qui signifie que l'application ne doit pas être hors service plus de 0,1 % du temps (environ 8,76 heures par an)
2.	Garantir des temps de réponse rapides aux demandes des utilisateurs en toute circonstance	Un temps de réponse moyen inférieur à 1 seconde pour les requêtes de recherche
3.	Protéger les données des utilisateurs contre les violations de sécurité	Cryptage des données et audits de sécurité tous les deux ans
4.	Fournir un support client efficace pour résoudre les problèmes et répondre aux questions	Délais de réponse pour les demandes d'assistance doit être inférieur à 10h
5.	Minimiser les temps d'arrêt et les perturbations de service	Délais de résolution pour les incidents et les problèmes techniques inférieur à 4h
6.	Informar les utilisateurs des interruptions de service planifiées ou imprévues	Le délai de notification en cas de maintenance planifiée doit être d'une semaine ou pour un incident majeur doit être d'une heure

Contrats de service application

Objectifs de niveau de service

Maintenir un haut niveau de qualité afin de satisfaire au contrat de service business

Indicateurs de niveau de service

ID	Objectifs	SLA
1.	Assurer que l'application est disponible pour les utilisateurs	Une disponibilité de 99,9 %, ce qui signifie que l'application ne doit pas être hors service plus de 0,1 % du temps (environ 8,76 heures par an)
2.	Garantir des temps de réponse rapides aux demandes des utilisateurs en toute circonstance	Un temps de réponse moyen inférieur à 1 secondes pour les requêtes de recherche
3.	Protéger les données des utilisateurs contre les violations de sécurité	Cryptage des données et audits de sécurité tous les deux ans
4.	Maintenir l'application à jour et sécurisée	Fréquence des mises à jour et des correctifs d'une fois par semaine Délais de déploiement en cas de vulnérabilités critiques inférieur à 2h
5.	Assurer la sauvegarde régulière des données et la capacité de les récupérer en cas de besoin	Fréquence des sauvegardes de 12h Délais de récupération des données en cas d'avarie de 12h Les sauvegardes doivent être testées pour être validées Les sauvegardes ne doivent pas être stockés au même endroit que les données d'origines (géographiquement)
6.	Assurer la conformité aux réglementations applicables	Respect du RGPD
7.	Établir un plan de reprise d'activité (PRA) en cas de catastrophe ou de perturbation majeure.	Un exercice de PRA devra être réalisé une fois par an

Lignes directrices pour l'implémentation

La nouvelle plateforme devra répondre à des exigences de scalabilité, performances et de livraisons rapides. De plus, dans un premier temps elle devra fonctionner en parallèle de l'ancienne plateforme avant son décommissionnement.

Afin de répondre à ces exigences, une solution Micro-Service avec hébergement dans le cloud est proposé. Les détails de cette solution se trouve dans la section suivante.

Spécifications pour l'implémentation

L'architecture **micro services** est un style de conception logicielle où une application est décomposée en petits services autonomes qui travaillent ensemble pour fournir une fonctionnalité complète. Chaque service est indépendant, communique via des interfaces, peut être développé et déployé séparément, et offre une meilleure évolutivité et résilience. Cette approche est particulièrement adaptée aux applications complexes et évolutives.

Pour ce faire, chaque micro service doit être conteneurisé. La **conteneurisation** est une méthode qui permet d'encapsuler des applications et leurs dépendances dans des environnements isolés et portables, appelés conteneurs. Cela facilite le déploiement, la gestion et la portabilité des applications tout en offrant une isolation et une flexibilité accrues. Elle est essentielle pour le développement et le déploiement d'applications modernes.

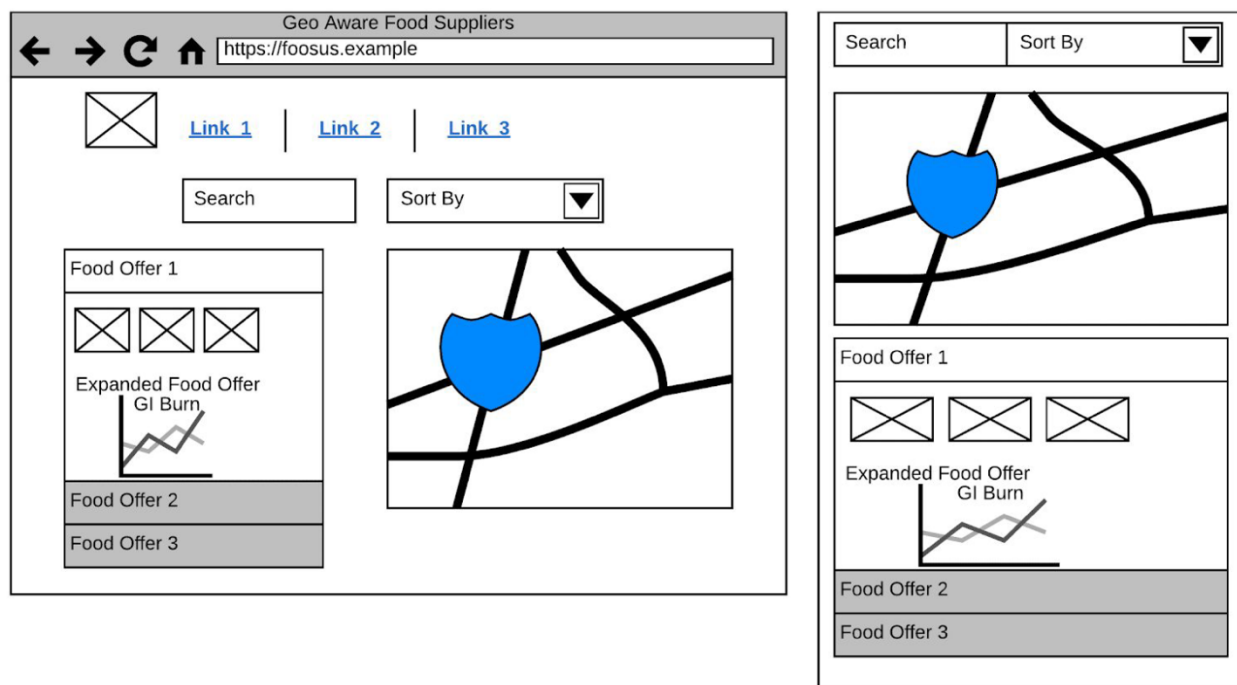
Afin de gérer l'ensemble des conteneurs (un conteneur = un micro service) qui constitue notre application, nous utiliserons l'orchestration. **L'orchestration** pour une application micro services consiste à coordonner et à gérer les différents services micro services qui composent l'application, en s'assurant qu'ils fonctionnent harmonieusement ensemble. Elle inclut la gestion du déploiement, de la mise à l'échelle, de la supervision et de la coordination des services pour garantir la performance et la fiabilité de l'ensemble de l'application.

Afin de répondre au besoin de performance tout autour du monde, nous utiliserons le principe du cloud régionalisé. Le **cloud régionalisé** pour une application micro services signifie que l'application est déployée sur des centres de données dans différentes régions géographiques du cloud. Cela permet d'améliorer la disponibilité, la résilience et les performances de l'application en fonction de la localisation des utilisateurs, tout en garantissant que les services micro services sont répartis de manière stratégique pour servir efficacement différentes régions.

Afin d'automatiser le déploiement des nouvelles itérations nous utiliserons une **chaîne CI/CD**. Une CI/CD (Intégration Continue/Livraison Continue) pour une application micro services est un processus automatisé qui permet de développer, tester et déployer rapidement et de manière cohérente chaque micro service, garantissant ainsi une mise à jour continue et efficace de l'application. Cela favorise l'agilité, la qualité et la stabilité de l'ensemble de l'application.

Le front-end sera développé en single page application et s'appuiera sur le principe du micro front-end.

Une spécification **d'UI** pour l'implémentation de la nouvelle fonctionnalité est également fournie. Voir les maquettes ci-dessous :



- 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

Standards pour l'implémentation

Voici la liste des standards à respecter :

API REST : Une API REST est un style architectural pour la conception de services web qui utilise des ressources HTTP pour représenter les données et des méthodes standardisées (GET, POST, PUT, DELETE) pour effectuer des opérations sur ces ressources. Elle favorise la simplicité, la scalabilité et l'indépendance des composants, permettant aux clients de communiquer avec un serveur de manière cohérente et légère, généralement en échangeant des données au format JSON ou XML. Les API REST sont largement utilisées pour développer des applications web et mobiles, ainsi que des services web accessibles via Internet.

HTTPS : HTTPS (HyperText Transfer Protocol Secure) est un protocole de communication sur Internet qui sécurise les échanges de données entre un client et un serveur. Il utilise un chiffrement pour garantir la confidentialité et l'intégrité des informations transmises, empêchant ainsi les tiers indésirables d'accéder ou de modifier les données pendant la transmission. Cette

technologie est essentielle pour protéger les informations sensibles lors de la navigation sur des sites web, notamment pour les transactions en ligne, les mots de passe et les données personnelles. Le protocole HTTPS est reconnaissable par le cadenas dans la barre d'adresse du navigateur, indiquant une connexion sécurisée.

Authentification : L'authentification est le processus par lequel un système ou une application vérifie l'identité d'un utilisateur ou d'une entité pour s'assurer qu'ils sont bien qui ils prétendent être. Cela implique généralement la fourniture de données d'identification, telles que des noms d'utilisateur et des mots de passe, qui sont ensuite vérifiées par le système pour permettre ou refuser l'accès à des ressources ou des fonctionnalités spécifiques. L'authentification est un élément clé de la sécurité informatique et est couramment utilisée pour protéger l'accès aux comptes en ligne, aux systèmes informatiques et à d'autres données sensibles.

Responsive Design : Le responsive design est une approche de conception web qui vise à créer des sites web et des applications qui s'adaptent automatiquement à différentes tailles d'écran, notamment sur des appareils mobiles tels que les smartphones et les tablettes. Cette technique utilise des techniques de mise en page flexibles et des médias queries pour ajuster dynamiquement la disposition et la présentation du contenu en fonction de la résolution et de la taille de l'écran de l'utilisateur. Le responsive design garantit une expérience utilisateur optimale quel que soit le dispositif utilisé pour accéder au site, améliorant ainsi l'accessibilité et la convivialité.

Lean : Le Lean est une philosophie de gestion axée sur l'élimination du gaspillage dans les processus et l'optimisation de la valeur pour le client. Il s'agit de minimiser les ressources inutiles tout en maximisant l'efficacité et la qualité.

Normes ISO : Les normes ISO sont des standards internationaux établis par l'Organisation internationale de normalisation (ISO) pour garantir la qualité, la sécurité, la durabilité, et d'autres aspects dans divers domaines tels que la gestion de la qualité (ISO 9001), la gestion environnementale (ISO 14001), etc.

RGPD : Le Règlement Général sur la Protection des Données (RGPD) est une réglementation européenne qui vise à protéger la vie privée et les données personnelles des individus. Il impose des obligations strictes aux organisations en ce qui concerne la collecte, le traitement, et la sécurité des données personnelles.

Règle ANSSI : Les règles de l'Agence nationale de la sécurité des systèmes d'information (ANSSI) en France sont des directives de sécurité informatique qui visent à renforcer la protection des systèmes d'information contre les menaces et les cyberattaques. Elles couvrent divers aspects de la sécurité informatique.

OWASP : L'Open Web Application Security Project (OWASP) est une organisation à but non lucratif qui se consacre à l'amélioration de la sécurité des logiciels. L'OWASP publie notamment une liste des 10 principales vulnérabilités de sécurité des applications web (OWASP Top Ten) pour sensibiliser à ces problèmes et promouvoir les bonnes pratiques de sécurité.

Conditions requises pour l'interopérabilité

Pour assurer l'interopérabilité, plusieurs conditions requises doivent être remplies. Ces conditions garantissent que différents composants, systèmes ou logiciels peuvent fonctionner ensemble de manière transparente et efficace. Voici quelques-unes des conditions requises pour l'interopérabilité :

Normes et Protocoles Communs : L'utilisation de normes et de protocoles de communication communs est essentielle. Tous les systèmes impliqués doivent comprendre et respecter ces normes pour échanger des données de manière cohérente.

API REST : Fourniture d'API documentées afin de permettre aux systèmes de communiquer de manière structurée.

Formats de Données Standard : Utilisation de JSON

Protocoles de Sécurité commun : Afin de faciliter les échanges entre services, les protocoles de sécurité doivent être les mêmes sur toute la plateforme, sauf cas particulier

Conventions de Nommage : Des conventions de nommage cohérentes et compréhensibles facilitent l'identification et la compréhension des données et des services entre les systèmes.

Interopérabilité Bimodale : La coexistence de l'ancienne et la nouvelle plateforme nécessitera de prendre en charge l'interopérabilité entre les anciennes technologies et les nouvelles, permettant aux systèmes hérités de fonctionner avec les systèmes modernes.

Conditions requises pour le management du service IT

Ces éléments sont présents dans la Déclaration de travail d'architecture que vous pouvez trouver dans le repository d'artefacts architecturaux.

Chapitres :

- Approche managériale
- Procédures de changement de périmètre
- Process du projet

Contraintes

- Le projet initial est approuvé avec un budget de 50 000 USD et une période de 6 mois est allouée pour définir l'architecture et préparer un projet pilote en vue de développer un prototype.
- L'architecture doit être conçue pour optimiser le rapport qualité-prix.
- L'architecture peut inclure de nouveaux composants personnalisés ou des composants commerciaux pour favoriser la flexibilité, la stabilité et l'extensibilité.
- Les solutions open source sont préférées aux solutions payantes.
- Il faut tenir compte du support continu des composants lors de leur sélection ou de la décision d'achat.
- Dans la mesure du possible, toutes les solutions, qu'elles soient commerciales ou open source, doivent faire partie d'une même pile technologique afin de réduire les coûts de maintenance et de support continus.

Hypothèses

ID	Hypothèse	Impact
1.	Maintien de la plateforme actuel	Plus de développement sur la plateforme historique
2.	Intégrer les technologies historiques de la plateforme et permettre l'utilisation de nouvelles technologies	Concevoir une plateforme évolutive
3.	Accès des utilisateurs aux nouvelles fonctionnalités de manière progressive	Prévoir un double run (ancienne et nouvelle application) Prévoir les ressources nécessaires pour la montée en charge
4.	Élaboration sur mesure d'une approche architecturale type « lean »	Cela contribue à la bonne exécution de la feuille de route et cela évitera de priver les équipes de leur autonomie et de compromettre la rapidité des cycles