17/04/2023

Spécification des Conditions Requises pour l'architecture

Ouelaa Icham
FOOSUS

Spécification des Conditions requises pour l’Architecture

Projet : Foosus

Client : Foosus

Préparé par : Ouelaa Icham

N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l’Architecture

Date de Version du Document : 03/2023

Revu par :

Date de Révision :

Table des Matières

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 s'effectue via les métriques suivantes qui permettront de déterminer le si projet est un succès ou si les résultats voulus ne sont pas attendus :

Métrique	Technique de mesure	Valeur cible (objectifs voulus)
Nouvelle adhésion de client journaliers	Analyse des nouvelles adhésions + sondage	+ 10%
Nouvelle adhésion de producteurs	Analyse des nouvelles adhésions + sondage	De 1,4 / mois à 4 / mois
Taux d'incidents après la mise en production	Mise en place d'un système de monitoring avec un suivi des logs d'erreurs	Pour commencer moins de 25 par mois puis moins d'un incident par mois
Réduction du délai moyen de parution	Vérification des dates de parution et donc de leurs délais	Réduit de 3.5 semaines à moins d'une semaine

Conditions requises pour l'architecture

Dans le cadre de ce projet les conditions requises en terme d'architecture sont les suivantes :

- Les solutions open source sont préférables aux solutions payantes.
- Le support continu des composants doit être pris en compte lors de leur sélection ou lors des prises de décision de création ou d'achat.
- Toutes les solutions du commerce ou open source doivent, dans la mesure du possible, faire partie d'une même pile technologique afin de réduire les coûts de maintenance et de support continus.

Contrats de service business

Accords de niveau de service

En terme business foosus à définit des objectifs et des besoins précis qui permettront une évolution de l'entreprise stable et pérenne dans le temps.
L'objectif des accords de niveau de service seront de mettre en place un accord entre un fournisseur de services et son client, définissant les niveaux de prestation convenus entre eux.

En définissant la qualité et la performance des prestations attendues, ainsi que les délais d'intervention et de résolution d'éventuels problèmes techniques, l'accord de niveau de service aidera foosus à mieux gérer les interactions avec leurs fournisseurs, ce qui impacte positivement de nombreux secteurs de leur activité.

En mettant en place des accords précis et bien développés sur les niveaux de service, en planifiant avec soin leur application et en assurant leur mise en œuvre pertinente, l'objectif de foosus sera :

- De garantir un service optimal et la satisfaction de ses clients
- D'offrir aux utilisateurs finaux une idée réaliste de la qualité de service qu'ils sont en droit d'attendre
- D'éviter toute confusion concernant les tâches à exécuter et les objectifs à atteindre
- De fournir une source fiable d'information au sujet de l'accord
- D'obliger le client et le fournisseur à être attentifs aux détails
- De protéger les fournisseurs contre des critères ambigus ou non déclarés

Contrats de service application

Objectifs de niveau de service

Ce projet vise à libérer la créativité et l'expérience des équipes techniques Foosus afin de pouvoir donner le meilleur d'elles-mêmes en créant une nouvelle plateforme qui pourra suivre le business, faire évoluer la base clientèle, améliorer l'expérience des utilisateurs (clients, consommateurs, producteur et artisans locaux.

La nouvelle solution doit être développée sans interruption de service la solution existante.
La nouvelle plateforme devra permettre aux équipes produites d'innover rapidement en réorientant des solutions existantes, en expérimentant de nouvelles modifications et en facilitant l'intégration avec des partenaires internes et externes.

Indicateurs de niveau de service

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.

Lignes directrices pour l'implémentation

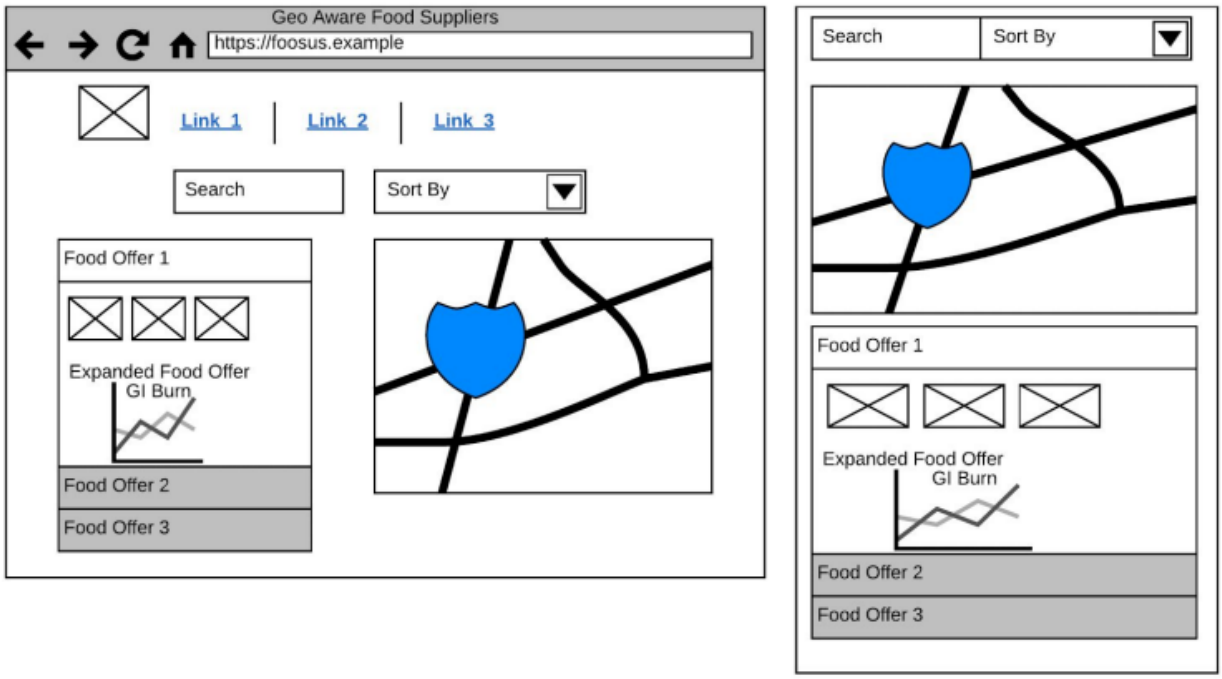
Les lignes directrices principales et standards de l'implémentation sont :

- Décisions pilotées par le feed-back et l'apprentissage.
- Faire des choix qui soutiennent les objectifs long terme.
- Accepter le fait que les erreurs se produisent.
- S'assurer que nous concevons l'architecture pour s'améliorer.
- Concevoir des interfaces ouvertes et extensibles en systèmes, sur lesquelles il est facile d'itérer.
- Appliquer une approche pilotée par le contrat client, où les interfaces entre les systèmes reflètent uniquement les données et opérations nécessaires à leur intégration
- Les solutions open source sont préférables aux solutions payantes.
- Le support continu des composants doit être pris en compte lors de leur sélection ou lors des prises de décision de création ou d'achat.
- Toutes les solutions du commerce ou open source doivent, dans la mesure du possible, faire partie d'une même pile technologique afin de réduire les coûts de maintenance et de support continus.

Spécifications pour l’implémentation

La fonctionnalité la plus critiquée de la plateforme existante est la fonctionnalité de recherche de fournisseurs alimentaires. Les enseignements actuels montrent que 48 % des parcours commencent par une recherche, mais sont abandonnés avant qu'un client ait même consulté une Offre alimentaire de l'un de nos fournisseurs.
Les équipes UX et CX ont testé une nouvelle conception qui attribue une forte priorité aux producteurs et aux artisans locaux en fonction de critères de géolocalisation.
Le résultat attendu pour le premier trimestre est l'amélioration de notre capacité de recherche en utilisant la pile technologique de l'état recherché.

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 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.

Conditions requises pour l’interopérabilité

L’interopérabilité peut être garanti grâce à l’adoption d’une architecture micro-service pour le nouveau système.

Les architectures micro-services permettent de développer, de déployer et de gérer opérationnellement des applications distribuées, constituées de services aux fonctionnalités complémentaires, potentiellement hétérogènes et interopérables.

Les micro-services étant indépendants et interopérables, ils ne sont pas contraints à une uniformité technologique. Les choix technologiques peuvent ainsi être adaptés aux besoins spécifiques de chaque micro-service. Il est ainsi possible d’utiliser un serveur de bases de données non-relationnel plus performant en écriture et plus extensible pour un micro-service de gestion d’historique.

Contraintes

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

- 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.
- L'architecture doit permettre d'obtenir le meilleur rapport qualité-coût.
- 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.

Hypothèses

Il serait possible de baser l’architecture sur un modèle orientée services :

Définition :

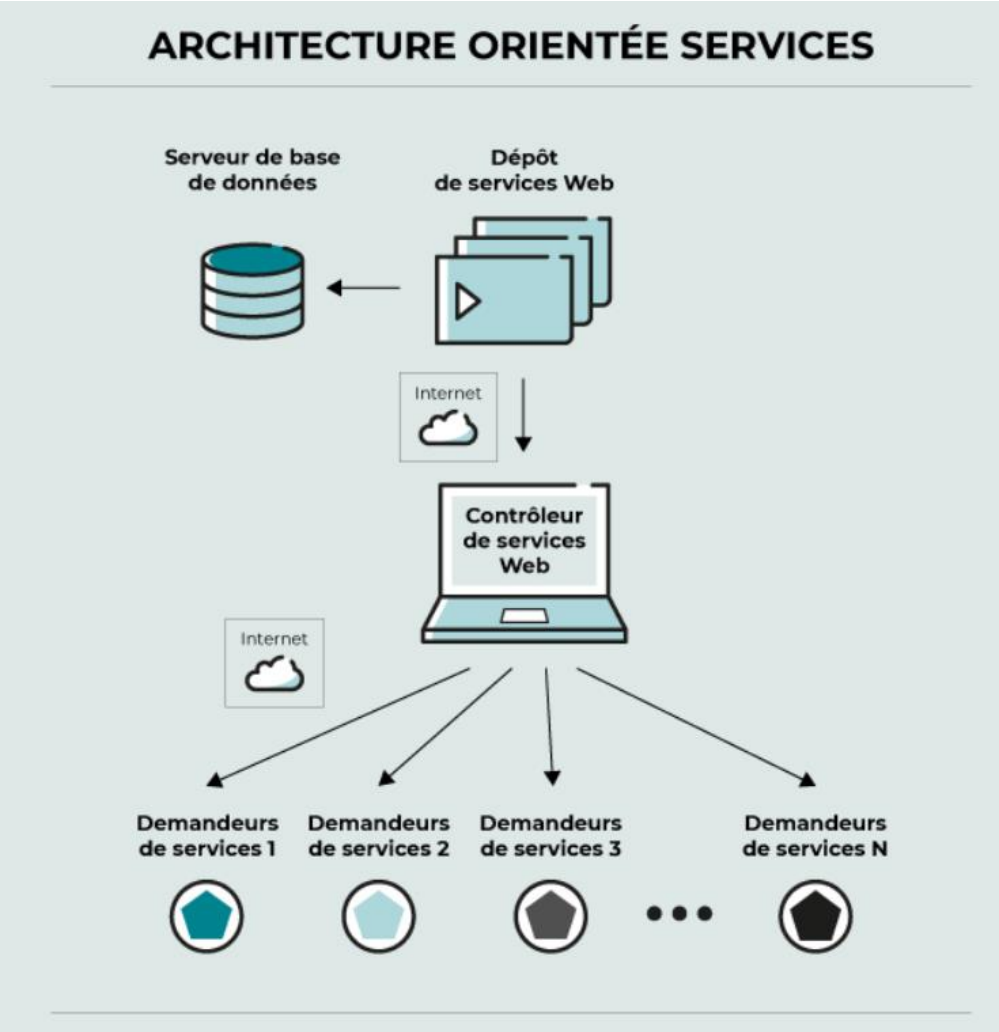
L'architecture orientée services (ou SOA, Service-Oriented Architecture) est un modèle de conception qui rend des composants logiciels réutilisables, grâce à des interfaces de services qui utilisent un langage commun pour communiquer via un réseau.

Un service est une unité autonome de fonctionnalité logicielle, ou d'un ensemble de fonctionnalités, conçue pour réaliser une tâche précise comme récupérer des

informations ou exécuter une opération. Il contient les intégrations de code et de données nécessaires pour exécuter une fonction métier distincte et complète. Vous pouvez y accéder à distance, et interagir avec lui ou le mettre à jour de manière indépendante.

En d'autres termes, l'architecture SOA permet à des composants logiciels déployés et gérés séparément de communiquer et de fonctionner ensemble sous la forme d'applications logicielles communes à différents systèmes.

Schéma et composants :



Comme vous pouvez le voir dans le schéma ci-dessus, une architecture client-serveur standard comporte quatre parties :

- Le dépôt de services web (Web service repository) : Il s'agit d'une bibliothèque de services web conçue pour répondre à des demandes d'informations externes. L'information fournie est généralement un petit élément, comme un numéro, un mot, quelques variables, etc. Par exemple, un numéro de vol, un numéro de suivi

de colis, le statut d'une commande (une lettre), etc. Cette bibliothèque est généralement documentée de manière très détaillée, car des applications externes font appel aux fonctions qu'elle contient.

- Le contrôleur de services web (Web service controller) : Ce module communique les informations contenues dans le dépôt de services web aux demandeurs de services. Lorsqu'un demandeur de service externe appelle une certaine fonction du dépôt de services web, le contrôleur de services web interprète la demande et recherche la fonction dans le dépôt de services web. Il exécute ensuite cette fonction et renvoie une valeur au demandeur.
- Le serveur de base de données (Database Server) : Ce serveur contient les tables, les index et les données gérés par l'application. Les recherches et les opérations d'insertion/suppression/mise à jour sont exécutées ici.
- Les demandeurs de services (Service Requester) : Il s'agit d'applications externes qui demandent des services au dépôt de services web par l'intermédiaire d'Internet, comme une organisation demandant des informations sur les vols à une compagnie aérienne, ou une autre entreprise demandant à un transporteur la localisation d'un colis à un moment donné.

Notez que l'architecture orientée services est créée par l'entreprise qui fournit le service et non par celle qui le consomme, cela permet de simplifier les connexions avec les clients éventuels.

Exemple d'utilisation :

Prenons l'exemple de Visa qui fournit un service à une entreprise connue de jeans :

- Le site web de Levi's a besoin des informations de Visa (fournisseur de services) pour effectuer une transaction.
- Le site web de Levi's appelle la fonction « Validate_Credit_Card_Number 5555 4567 8901 3456 » à partir du dépôt de services web du site Visa.
- Le site web de Visa reçoit la demande, exécute la fonction « Validate_Credit_Card_Number 5555 4567 8901 3456 » et renvoie la valeur « OK », ce qui signifie que la carte de crédit est valide.
- Le site web de Levi's reçoit la valeur « OK » et l'utilise pour conclure la transaction.

Avantages et inconvénients :
L'utilisation d'une architecture orientée services présente plusieurs avantages :

- Ce modèle permet de collaborer avec des acteurs externes sans les laisser accéder à nos systèmes. Dans l'exemple ci-dessus, Visa ne veut pas ouvrir sa base de données à des tiers pour des raisons de sécurité, mais elle souhaite néanmoins que les clients puissent obtenir les informations dont ils ont besoin.

- Il permet également de partager avec le monde entier, de manière ordonnée et contrôlée, la sélection des fonctions les plus populaires du site. Visa dit : « Si quelqu'un veut connaître l'état de la carte X, appelez cette fonction, donnez la valeur de X et je répondrai par oui ou par non. » La fonction est ouverte au monde en tant que service dans un environnement très strict et contrôlé ; les clients ne peuvent pas accéder à ce qu'ils veulent.

Il existe également quelques inconvénients majeurs :

- Les services web peuvent représenter une faiblesse au niveau du site pour les pirates qui veulent engorger le système. Certaines formes d'attaques sont des « dénis de service ». Elles consistent à demander le même service web des millions de fois par seconde, jusqu'à ce que le serveur tombe en panne. Il existe aujourd'hui une technologie permettant de résoudre ce problème, mais c'est toujours un problème à prendre en compte dans les architectures de services web.

- Le propriétaire du service web aide d'autres sites, mais reçoit une petite rémunération pour ce faire.