



# Spécification des Conditions requises pour l'Architecture

Par

Alois GRIMM

Projet : Nouvelle plate-forme d'e-commerce

Client : Foosus

## TABLE DES MATIERES

Information sur le document .....	2
Objet de ce document.....	3
Mesures du succès .....	3
Conditions requises pour l'architecture .....	4
Contrats de service business .....	4
Accords de niveau de service .....	4
Contrats de service application .....	4
Objectifs de niveau de service.....	4
Indicateurs de niveau de service .....	5
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 .....	8



## INFORMATION SUR LE DOCUMENT

Nom du projet	Projet Nouvelle plate-forme d'e-commerce
Préparé par :	Aloïs GRIMM
N° de version du document :	1.0
Titre :	Spécification des Conditions requises pour l'Architecture
Date de version du document :	11/08/2022
Revu par :	
Date de révision :	
Liste de distribution :	
De :	
Date :	
Email :	
Pour Action :	
Date de rendu :	
Email :	
Types d'action :	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion
Historique de versions du document	<a href="#">Lien vers github</a>



## 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 SUCCES

Métrique	Technique de mesure	Valeur cible	Notes
Nombre d'adhésions d'utilisateurs par jour	Requête SQL	+ 10 %	
Adhésion de producteurs alimentaires	Requête SQL	4 / mois	V.A. : 1,4 / mois
Remplacer l'ancienne plateforme	Suivi de projet	1 an	
Nouvelle plateforme conforme aux recommandations	Test validation	Tous les objectifs sont atteints	

## CONDITIONS REQUISES POUR L'ARCHITECTURE

- Créer une plateforme de commerce électronique polyvalente
- Efficacité, flexibilité et approches cohérentes dans la prise de décision
- Pouvoir concurrencer les grandes entreprises mondiales de commerce électronique sur le marché de l'alimentation durable.
- Innover de façon responsable
- Poursuivre la croissance de l'entreprise
- Innover rapidement
- Plateforme scalable
- Permettre des mises à jour à chaud, sans interruption de service

## CONTRATS DE SERVICE BUSINESS

### ACCORDS DE NIVEAU DE SERVICE

- Tirer parti de la géolocalisation pour relier des fournisseurs et des consommateurs et pour proposer des produits disponibles à proximité des lieux de résidence de ces derniers.
- L'architecture devra être évolutive pour permettre à nos services de se déployer sur diverses régions à travers des villes et des pays donnés.
- Notre solution doit être disponible pour nos fournisseurs et nos consommateurs, 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.
- Elle doit pouvoir prendre en charge différents types d'utilisateurs (par exemple, fournisseurs, back-office, consommateurs) avec des fonctionnalités et des services spécifiques pour ces catégories.

## CONTRATS DE SERVICE APPLICATION

### OBJECTIFS DE NIVEAU DE SERVICE

- L'architecture doit être évolutive et pouvoir intégrer de nouvelles technologies lorsqu'elles sont disponibles.
- Les mises à jour doivent être transparentes, et ne pas interrompre le service pendant des heures.
- La plateforme doit être disponible 24h/24.
- La sécurité ne doit pas être négligée au profit de la simplicité d'utilisation.
- 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.
- Intégrer des prestataires de paiement tiers ;
- Gérer toutes les communications avec les fournisseurs alimentaires au sein d'une interface utilisateur personnalisée.

#### INDICATEURS DE NIVEAU DE SERVICE

Métrique	Technique de mesure	Valeur cible	Notes
Délai moyen de parution	Logiciel gestion de projet	Moins d'une semaine	V.A. : 3,5 semaines
Taux d'incidents de production P1	Formulaire de déclaration d'incident et supervision	Moins 1 / mois	V.A. : 25 / mois

#### LIGNES DIRECTRICES POUR L'IMPLEMENTATION

- 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.

## SPECIFICATIONS POUR L'IMPLEMENTATION

Type	Nom	Avantage
Modèle d'architecture	Micro-service	<ul style="list-style-type: none"> <li>• Scalable</li> <li>• Peu fonctionner en mode dégradé</li> <li>• Développement rapide</li> <li>• S'adapte aux méthodes Agile et Lean</li> </ul>
OS	Debian	<ul style="list-style-type: none"> <li>• Licences diverses (DFSG)</li> <li>• Libre et Gratuit</li> <li>• Communauté active</li> <li>• Stable</li> <li>• Support longue durée</li> </ul>
Virtualisation	Docker Kubernetes	<ul style="list-style-type: none"> <li>• Licence Apache version 2.0</li> <li>• Libre et Gratuit</li> <li>• Permet de crée des clusters hautes disponibilités</li> <li>• Automatise la montée en charge</li> <li>• Auto-guérison, redémarre les conteneurs qui ne répondent plus</li> </ul>
Messagerie interservice	Kafka	<ul style="list-style-type: none"> <li>• Licence Apache v2</li> <li>• Libre et Gratuit</li> <li>• Supporte la haute disponibilité</li> <li>• Supporte le système distribué</li> </ul>
SGBD	Postgres	<ul style="list-style-type: none"> <li>• Licence BSD</li> <li>• Libre et Gratuit</li> <li>• Communauté active</li> <li>• Bonne gestion de base de données volumineuse</li> <li>• Gestion de données spatiales</li> <li>• Supporte la haute disponibilité en hot standby / warm standby ou réplication logique</li> </ul>
Géolocalisation	Openroute service	<ul style="list-style-type: none"> <li>• Licence MIT/X11</li> <li>• Utilise la base de données openstreetmap</li> <li>• Installable sur notre infrastructure sans restriction</li> </ul>

Format d'échange de données	JSON	<ul style="list-style-type: none"> <li>• Structure normée</li> <li>• Encodage UTF8</li> <li>• Compact</li> </ul>
Sécurité	Traefik	<ul style="list-style-type: none"> <li>• Licence MIT</li> <li>• Libre et Gratuit</li> <li>• Sécurise les transactions entre DMZ publique et privée</li> </ul>
Journalisation Suppervision	Grafana Promtail Loki	<ul style="list-style-type: none"> <li>• Licence AGPL-3.0</li> <li>• Gratuit</li> <li>• Libre</li> <li>• Peut fonctionner en système distribué, et en haute disponibilité</li> </ul>
	Prometheus	<ul style="list-style-type: none"> <li>• Licence Apache version 2.0</li> <li>• Gratuit</li> <li>• Libre</li> <li>• Plugin Kubernetes pour suivre les démarrages de conteneur</li> <li>• Peut fonctionner en système distribué, et en haute disponibilité</li> </ul>

## STANDARDS POUR L'IMPLEMENTATION

- Assuré la qualité avec ISO 9001
- Réussir la transformation avec TOGAF
- Hébergement certifié ISO 50001 et suivi des bonnes pratiques du Green IT

## CONDITIONS REQUISES POUR L'INTEROPERABILITE

- Afin de faciliter la migration et la création de lien entre l'ancien et le nouveau système, les deux plateformes utiliseront des tokens compatible.
- Une fois les langages, les technologies et le SGBD choisi, un modèle de microservice sera créé et servira à tous les autres microservice. Cela garantira l'agilité des développeurs et simplifiera le développement et la maintenance.
- Les microservices communiqueront via une messagerie asynchrone de type publish / suscribe afin d'assurer la transmission des messages.



## CONDITIONS REQUISES POUR LE MANAGEMENT DU SERVICE IT

- Développement agile
- Management Lean
- Intégré le green IT

## CONTRAINTES

- 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.

## HYPOTHESES

- Plutôt que d'investir davantage dans la plateforme existante, nous la conserverons en mode de maintenance. Aucune nouvelle fonctionnalité ne sera développée.
- 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.
- 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.
- 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.
- Les utilisateurs précoces pourront choisir d'utiliser les nouvelles fonctionnalités de recherche intégrées au processus de paiement existant.
- 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.
- L'élaboration sur mesure d'une approche architecturale de type « lean » pourra contribuer à la réalisation de cette feuille de route, ce qui évitera de priver les équipes de leur autonomie et de compromettre la rapidité des cycles de versions.

