



# **Spécification des Conditions requises pour l'Architecture**

*Projet : Foosus géoconscient v2 Client : Foosus*



## Table des matières

Objet de ce document.....	4
Mesures du succès.....	5
Conditions requises pour l'architecture.....	6
Contrats de service business.....	7
Accords de niveau de service.....	7
Contrats de service application.....	7
Objectifs de niveau de service.....	7
Indicateurs de niveau de service.....	8
Lignes directrices pour l'implémentation.....	9
Standards pour l'implémentation.....	9
Développement Back-end/Front-end.....	9
Spring boot.....	9
Angular.....	10
DevSecOps.....	11
GitHub.....	11
Docker.....	11
Communication/gestion de projet.....	12
Trello.....	12
Discord.....	12
Infrastructure.....	13
Amazon AWS.....	13
Conditions requises pour l'interopérabilité.....	14
Application.....	14
Infrastructure.....	14
Conditions requises pour le management du service IT.....	15
Contraintes.....	16
Hypothèses.....	16



Foosus – Projet *géoconscient* V2

## Information sur le document

<i>Nom du projet</i>	Projet <i>Foosus</i> <i>géoconscient</i> v2
<i>Préparé par :</i>	Stiven GUILLAUME
<i>N° de version du document :</i>	0.1
<i>Titre :</i>	<i>Spécification des Conditions requises pour l'Architecture</i>
Types d'action :	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion, Autre (à spécifier)
Historique de versions du document	Voir git



## 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

---

Métrique	Valeur cible	Justification	Notes supplémentaires
Backlog projet	0		Le backlog projet est vide. Le projet n'a plus de fonctionnalités non livrés
Utilisateur sur la nouvelle plateforme	>95 %		Les utilisateur sont en grande majorité sur la nouvelle plateforme. Quelques utilisateurs historique/inactif ne sont pas encore migrés
Taux d'incident de prod P1	< 1 par mois	Actuel >25 par mois	La mise en place d'un outil de ticketing qui soit «ITIL compliant » est a prévoir
Adhésion de producteurs alimentaires	4 par mois	Actuel : 1,4 par mois	
Délais moyen de parution	<1 semaine	Actuel : 3,5 semaine	
Nombre d'adhésion par jour	+10 %		Est considéré comme étant une adhésion valide une inscription avec validation du



			compte par mail
Temps moyen de résolution	<30 minutes		La prise en charge et le temps de résolution d'un incident doit devenir le plus bas possible.

## Conditions requises pour l'architecture

---

Le nouveau projet à pour objectif la création et le déploiement d'une nouvelle plateforme de commerce électronique, dont le but est de permettre à l'entreprise de renouer avec la croissance, en améliorant son image de marque, et de pouvoir accueillir des nouveaux clients, mais aussi des nouvelles régions et services.

Pour arriver à répondre à ces objectifs, la plateforme possède une série d'exigences techniques ainsi qu'un contrat de développement et de conception.

Les principaux objectifs de l'entreprise en matière d'architecture sont les suivants.

- 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,



Foosus – Projet *géoconscient* V2

fournisseurs, back-office, consommateurs) avec des fonctionnalités et des services spécifiques pour ces catégories.

## Contrats de service business

### Accords de niveau de service

---

Les services suivants sont couverts par cet accord :

- Couverture de service 24/7.
- La mise en place d'un outils de gestion des incidents respectant les bonnes pratiques ITIL doit être faite, avec une priorisation des incidents en fonction de la gravité de l'impact, du nombre d'utilisateurs impactés et de la récurrence des incidents.
- Tout changement à la suite du projet doit être fait en respectant le workflow préconisé par ITIL :
  - Soumission de la demande
  - Planification
  - Approbations
  - Mise en production
  - Examen
  - Fermeture

## Contrats de service application

### Objectifs de niveau de service

---

Une indisponibilité maximum mensuelle de l'application est mise en place, suivant les critères



suivants :

Module	Disponibilité mensuelle
Infrastructure	Inférieur à 99,99 %, mais égal ou supérieur à 99,0 %
Support utilisateur N1 → N3	Inférieur à 99,0 %, mais égal ou supérieur à 95,0 %
Fonctionnalités applicatives	Inférieur à 99,99 %, mais égal ou supérieur à 99,0 %

## Indicateurs de niveau de service

les indicateurs de niveau de services suivant doivent être mis en place

Indicateur	Information
Fiabilité des livraisons	Les livraisons ne doivent pas provoquer d'interruption de service ou de baisse de qualité de l'application
Score de satisfaction client	Mettre en place sur l'application une possibilité de « noter » de 0 à 5 étoiles le processus de commande
Temps de réponse de l'application	Une réponse <200 ms est acceptable. Au delà, une analyse doit être fait pour savoir comment optimiser l'application





## Lignes directrices pour l'implémentation

---

Le projet doit respecter les directives 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 continu.

Pour arriver à respecter ces lignes directrices, le projet devra respecter des standards pour l'implémentation.

## Standards pour l'implémentation

---

Dans le cadre d'une volonté d'homogénéisation des différents développements, certaines pratiques devront être communes à toutes les équipes de développement, aussi bien d'un point de vue écriture de code que d'un point de vue utilisation d'outils.

### Développement Back-end/Front-end

Le développement de l'application s'appuiera sur les technologies suivantes :

Spring boot



## Foosus – Projet *géoconscient* V2

Les développeurs actuels étant habitués au langage Java, l'utilisation de Spring Boot réduit les risques de retard due au changement de langage.

Spring Boot est un framework Java qui facilite le développement d'applications basées sur Spring, offrant des outils permettant d'obtenir une application packagée en .jar, totalement autonome.

Ce qui représente l'intérêt principal dans une architecture micro services.

Ce sera le framework utilisé pour le back-end.



## Angular

Angular permet la création d'applications Web, et plus particulièrement des Single Page Applications, qui sont des applications web accessibles via une page web unique.

Il est basé sur le langage TypeScript.

Il sera là pour fournir le Front End.





Foosus – Projet *géoconscient* V2

## DevSecOps

### GitHub

GitHub est un service d'hébergement et de gestion de développement de logiciel, basé sur le système de version Git.

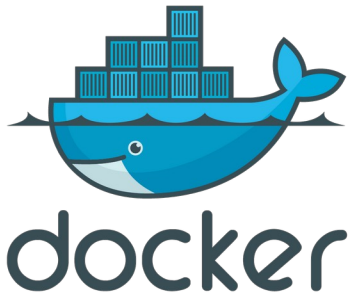


### Docker

Docker est une plateforme permettant l'exécution de certaines application dans des conteneurs logiciels. Il permet d'empaqueter une application et ses dépendances dans un conteneur isolé qui pourra être exécuté sur n'importe quel serveur.



Foosus – Projet *géoconscient* V2



## Communication/gestion de projet

### Trello

Trello est une application de gestion de projet en ligne et inspiré de la méthode Kanban. Il repose sur une organisation en planches listant des cartes représentant des tâches. Ces cartes seront adressées à des utilisateurs.



### Discord

Une communication « informelle » et dans le but d'entraide ou de partage de ressource pourra être mise en place via l'utilisation de discord.

Cet outil est un outil de chat/viéoconférence. Il permet l'utilisation de bot pouvant être développés et connectés via des hooks à des outils comme GitHub pour permettre l'information lors de commit ou de



Foosus – Projet *géoconscient* V2

push par exemple.



## Infrastructure

### Amazon AWS

Dans la mesure où un besoin d'environnement pouvant s'adapter à des pics de charge ou la croissance du nombre d'utilisateur est requis, la mise en place d'une infrastructure cloud est à envisager.

Nous préconisons la mise en place de l'infrastructure sur le cloud Amazon AWS.





# Conditions requises pour l'interopérabilité

---

## Application

Chaque micro-services devront communiquer les uns avec les autres. Pour cela, ils fonctionneront à l'aide d'une API RESTful (« Representational State Transfert »).

L'usage d'une API REST signifie que les micro services vont communiquer entres eux à partir d'une URL.

Cette communication va se faire en utilisant les méthodes GET, POST, PUT ou DELETE, pour l'obtention des données, l'envoi de données.

Cette utilisation permet une complète interopérabilité des micro services et l'ajout de nouvelles fonctionnalités.

## Infrastructure

L'infrastructure peut être vue comme ayant deux couches différentes

- Une couche physique
- Une couche virtuelle

La couche physique correspond au cloud fourni par Amazon AWS.

La couche virtuelle, elle, correspond aux données ou aux services virtualisés à l'aide de conteneurs Docker.

Dans le cadre d'un changement de fournisseur cloud (exemple : passage de Amazon AWS au cloud OVH), la couche virtuelle devra et pourra être redéployée sans difficultés.

Il y a donc plus de réelle dépendances complexes avec le fournisseur de l'infrastructure,



## Conditions requises pour le management du service IT

---

Le management du service IT, au-delà du développement et à la suite du projet devrait, dans l'idéal, respecter des bonnes pratiques de qualité (standard ISO 9001) et ITIL

Le couplage de ce standard qualité et ce framework se font s'approcher du standard ISO/IEC 20000

Cette norme est la norme internationale sur les services de gestion IT, permettant aux entreprise IT de garantir que leur processus de gestion IT sont conformes aux besoins de l'entreprise et aux bonnes pratiques internationales.

De ce fait, le management du service IT devra répondre aux critères suivants :

- **Orientation client** : Les processus utilisés prennent en compte l'environnement « business »
- **Leadership** : La direction doit faire preuve de son engagement. Il peut lui être demandé un arbitrage dans des problématiques techniques
- **Participation du personnel** : Chaque parties prenantes doivent être engagées et responsabilisées
- **Approche processus** : Les processus et les responsabilités doivent être définie pour pouvoir atteindre les objectifs
- **Approche système à la gestionnaire** : L'entièreté des activités doivent être couvertes par les processus. Pas seulement l'IT
- **Amélioration continue** : L'organisation doit continuellement améliorer l'efficacité et le rendement de son système, via des objectifs de gestion, des audits, des actions préventives...
- **Approche objective à la prise de décision** : Les décisions doivent être prise de manière objective, en se basant sur des faits ou des données chiffrées (KPI)
- **Relations mutuellement favorables entre fournisseurs** : La relation entre les prestataires de service et ses fournisseurs doit être de qualité et sans failles



## Contraintes

---

D'un point de vue architecturale, les directives suivantes doivent être si possible appliquées, ou a défaut s'en rapprocher le plus possible :

- Solution open source préférables aux solutions non open sources
- Support continu des composants à prendre en compte à la sélection
- rester dans la même pile technologique pour tous les modules composant l'application

D'autres contraintes sont à prendre en compte, au niveau de la phase d'étude comme au niveau du projet en lui même :

- Projet initial approuvé pour un coût de 50 000\$
- Période de 6 mois pour la définition de l'architecture, et la préparation d'un projet de suivi afin de développer un prototype
- Une architecture qui doit permettre d'obtenir le meilleur rapport qualité-coût
- Une architecture pouvant inclure de nouveaux composants personnalisés ou des composants déjà existant pour favoriser la flexibilité, la stabilité et l'extensibilité

## Hypothèses

---

I D	Hypothèse	Impact	Propriétaire
1	Aucune nouvelle fonctionnalité ne sera développée	Obligation de garder l'ancien système en mode de maintenance.  Chaque micro service ajoutés devront prendre le pas sur l'ancienne plateforme	Architecte logiciel Peter Parker (CTO)





## Foosus – Projet *géoconscient* V2

2	Technologies actuelles et capacité à s'adapter aux technologies futures	<p>L'architecture devra être en capacité d'accepter l'utilisation de technologies non existantes actuellement.</p> <p>Là encore, l'usage ne micro service permet de séparer chaque fonctionnalités, qui deviens totalement indépendante technologiquement</p>	Architecte logiciel Pater Parker (CTO)
3	Pas de faux raccourcis		Architecte logiciel
4	Montée en puissance empirique du volume d'utilisateur qui migrerons vers la nouvelle plateforme	<p>Chaque micro service devra être en capacité de travailler avec l'ancien système</p> <p>Les utilisateurs devront pouvoir être migrés fonctionnalité par fonctionnalité</p>	Architecte logiciel Peter Parker (CTO) Natasha Jarson (CIO)
5	Nouvelles fonctionnalités introduites grâce à la géolocalisation	L'architecture en micro service rend l'ajout de nouvelles fonctionnalité totalement possible, sans interruption de service, et permet l'innovation	Architecte logiciel Pater Parker (CTO)
6	Approche architecturale type Lean	Une approche mêlant Lean et Agile, mais totalement sur mesure pour être adapté aux équipes et au projet sera mise en place	Architecte logiciel Pater Parker (CTO) Natasha Jarson (CIO)
7	Impact environnemental	Développement respectant les bonnes pratique et s'incluant dans une démarche Green IT	Architecte logiciel Pater Parker (CTO) Natasha Jarson (CIO)