

# Contrat d'Architecture avec les Fonctions Développement et Design

*Projet : Conception d'une nouvelle  
architecture*

Version du document	Date	Diffusion
v0.1	23/06/2021	Structure et plan
v0.2	30/06/2021	Recherches et études + description de l'architecture
v0.3	07/07/2021	Avancée de parties du document
v0.4	21/07/2021	Diagrammes
v0.5	1/08/2021	Description de l'architecture
v0.6	11/08/2021	Périmètre et livrables
v0.7	18/08/2021	Critères d'acceptation
v1.0	01/09/2021	Finalisation du document

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Objet de ce document</b>	<b>3</b>
<b>Introduction et Contexte</b>	<b>3</b>
<b>La Nature de l'accord</b>	<b>4</b>
<b>Objectifs et périmètre</b>	<b>4</b>
Objectifs	4
Objectif Business 1	4
Objectif Business 2	4
Objectif Business 3	4
Objectif Business 4	5
Périmètre	5
Parties prenantes, préoccupations et visions	5
<b>Description de l'architecture, principes stratégiques et conditions requises</b>	<b>6</b>
Description	6
Principes stratégiques	7
Référence aux Conditions requises pour l'architecture	7
<b>Livrables architecturaux</b>	<b>9</b>
Développement de l'architecture	9
Mesures de l'architecture cible	10
Livraison de l'architecture et métriques business	11
Phases de livraison définies	11
<b>Plan de travail commun priorisé</b>	<b>11</b>
Architecture microservice	11
Architecture microservice : Activités	11
Architecture microservice : Livrables	11
Définition des ressources de l'API	11
Diagramme de composants	12
Diagramme d'infrastructure	12
Architecture event driven	12
Architecture event driven : Activités	12
Architecture event driven : Livrables	12
Diagramme d'architecture event-driven	12
Architecture cloud native	12
Architecture cloud native : Activités	12
Architecture cloud native : Livrables	13
Mise en place d'un cycle de vie SI	13
Mise en place d'un cycle de vie : Activités	13
Mise en place d'un cycle de vie : Livrables	13

Sécurisation du Système d'Information	13
Sécurisation du SI : Activités	13
Sécurisation du SI : Livrables	13
<b>Plan de communication</b>	<b>14</b>
<b>Risques et facteurs de réduction</b>	<b>15</b>
Structure de gouvernance	15
Analyse des risques	15
<b>Hypothèses</b>	<b>15</b>
<b>Critères d'acceptation et procédures</b>	<b>16</b>
Métriques et KPIs de l'État Cible de l'Architecture	16
Métriques de livraison de l'architecture et du business	16
Procédure d'acceptation	17
<b>Procédures de changement de périmètre</b>	<b>17</b>
Conditions requises pour la conformité	17
Développement et propriété de l'architecture	18
<b>Calendrier</b>	<b>20</b>
Diagramme de Gantt	20
<b>Personnes approuvant ce plan</b>	<b>22</b>

# Objet de ce document

Les Contrats d'Architecture sont les accords communs entre les partenaires de développement et les sponsors sur les livrables, la qualité, et la correspondance à l'objectif d'une architecture. L'implémentation réussie de ces accords sera livrée grâce à une gouvernance de l'architecture efficace (voir TOGAF Partie VII, Gouvernance de l'architecture). En implémentant une approche dirigée du management de contrats, les éléments suivants seront garantis :

- Un système de contrôle continu pour vérifier l'intégrité, les changements, les prises de décisions, et l'audit de toutes les activités relatives à l'architecture au sein de l'organisation.
- L'adhésion aux principes, standards et conditions requises des architectures existantes ou en développement
- L'identification des risques dans tous les aspects du développement et de l'implémentation des/de l'architecture(s), y compris le développement interne en fonction des standards acceptés, des politiques, des technologies et des produits, de même que les aspects opérationnels des architectures de façon à ce que l'organisation puisse poursuivre son business au sein d'un environnement résilient.
- Un ensemble de processus et de pratiques qui garantissent la transparence, la responsabilité et la discipline au regard du développement et de l'utilisation de tous les artefacts architecturaux
- Un accord formel sur l'organe de gouvernance responsable du contrat, son degré d'autorité, et le périmètre de l'architecture sous la gouvernance de cet organe

Ceci est une déclaration d'intention signée sur la conception et le développement de l'architecture d'entreprise, ou de parties significatives de celles-ci, de la part d'organisations partenaires, y compris les intégrateurs système, fournisseurs d'applications, et fournisseurs de service.

De plus en plus, le développement d'un ou plusieurs domaine(s) d'architecture (business, données, application, technologie) peut être externalisé, avec la fonction d'architecture de l'entreprise fournissant une vue d'ensemble de l'architecture d'entreprise globale, ainsi que la coordination et le contrôle de l'effort total. Dans certains cas, même ce rôle de supervision peut être externalisé, bien que la plupart des entreprises préfèrent conserver cette responsabilité clé en interne.

Quelles que soient les spécificités des dispositions d'externalisation, les dispositions elles-mêmes seront normalement gouvernées par un Contrat d'Architecture qui définit les livrables, la qualité, et la correspondance à l'objectif de l'architecture développée, ainsi que les processus de collaboration pour les partenaires du développement de l'architecture.

# Introduction et Contexte

Foosus est une start-up spécialisée dans la vente de produits alimentaires durables qui souhaite moderniser sa plate-forme et s'inscrire dans la tendance mondiale de modernisation des outils informatiques. Une transition vers une architecture et une manière de travailler mieux dédiée à l'évolutivité et l'adaptabilité est nécessaire après avoir longtemps privilégié la croissance et la créativité.

Ainsi, le but est de concevoir une nouvelle architecture qui permettra de poser des bases solides au développement informatique de la société et de sa plateforme en ligne.

## La Nature de l'accord

L'architecte a pour mission de concevoir une nouvelle architecture pour Foosus.

Cette architecture doit répondre aux besoins business établis lors du brief business haut niveau tout en permettant d'évoluer facilement vers de nouveaux objectifs qui seront établis dans la suite de ce projet.

## Objectifs et périmètre

### Objectifs

Les objectifs business de ce Travail d'Architecture sont les suivants :

#### Objectif Business 1

Le premier objectif business de la nouvelle solution architecture est d'utiliser la géolocalisation afin d'établir un lien entre les fournisseurs et les consommateurs.

#### Objectif Business 2

Les contraintes de réseau liées à la position géographique ou à l'utilisation d'un appareil mobile ne devront pas être un frein à l'accessibilité de la plate-forme.

#### Objectif Business 3

Des livrables devront être fournis régulièrement et sans interruption de service afin de permettre aux utilisateurs d'accéder rapidement aux nouvelles fonctionnalités ajoutées au système

## Objectif Business 4

Un système de rôle rendra disponible des fonctionnalités personnalisées sur la plateforme en fonction du type d'utilisateur (fournisseur, back-office, consommateur..).

## Périmètre

Le but de ce projet est d'établir un nouveau système et architecture cible vers laquelle itérer afin de permettre à Foosus d'entamer une transition vers des méthodes de travail plus rigoureuses et professionnelles. Au travers de contrats d'architecture, d'une déclaration de travail d'architecture et des spécifications des conditions requises pour l'architecture, nous détaillerons les différents objectifs, contraintes et étapes nous permettant de mettre en place une architecture résiliente, efficace et évolutive.

La mise en place d'environnements de développement continu, la construction d'une nouvelle architecture centrée autour de l'évolutivité et la résilience, l'ajout de nouvelles fonctionnalités identifiées par le business ainsi que la sécurisation du parc informatique seront les ajouts majeurs au nouveau système.

Cette première partie du projet sera effectuée dans les six prochains mois.

## Parties prenantes, préoccupations et visions

Le tableau suivant montre les parties prenantes qui utilisent ce document, leurs préoccupations, et la façon dont le travail d'architecture répondra à ces préoccupations par l'expression de plusieurs visions, ou perspectives.

Partie prenante	Préoccupation	Vision
Ash Callum	PDG (CEO)	Augmentation des inscriptions, système évolutif et flexible

<i>Daniel Anthony</i>	<i>Directeur Produit (CPO)</i>	Périmètre architectural clair, visibilité des données et indicateurs de l'architecture, amélioration de la stabilité du service
<i>Jack Harkness</i>	<i>Responsable Opérations</i>	Visibilité des données et indicateurs de l'architecture
<i>Christina Ortega</i>	<i>Directeur Marketing (CMO)</i>	Système évolutif et flexible, Visibilité des données et indicateurs de l'architecture
<i>Natasha Jarson</i>	<i>Directeur Informatique (CIO)</i>	Périmètre architectural clair, système évolutif et flexible

## Description de l'architecture, principes stratégiques et conditions requises

### Description

L'architecture est une architecture micro service possédant un nombre évolutif de services entièrement indépendants et qui communiquent entre eux par le biais d'une API Rest. Les services pourront être améliorés, remplacés ou supprimés au fur et à mesure de la croissance du système et de l'émergence de nouveaux besoins.

L'architecture sera orientée événements afin d'avoir un système très réactif aux demandes utilisateurs, tout en préservant (voir améliorant) la "scalabilité".

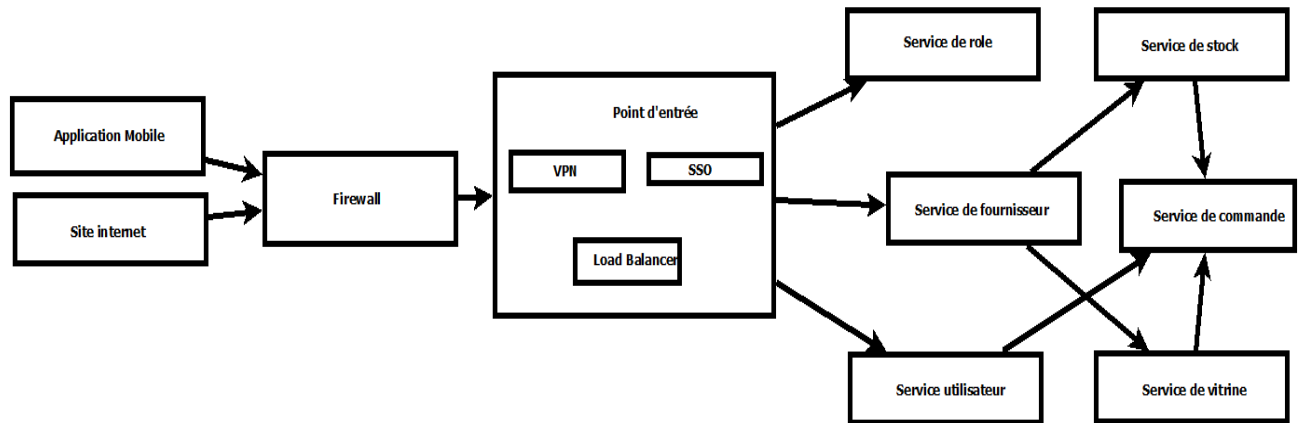


Diagramme de composants

## Principes stratégiques

Les principes stratégiques pour une architecture microservice orientée événement sont :

- des fonctionnalités découpées en service indépendants par leurs fonctionnalités et leur stockage de données;
- des événements qui déclenchent des réactions à travers le système en temps réel.

Ces principes permettent de faire évoluer avec énormément de liberté l'architecture et ses composants.

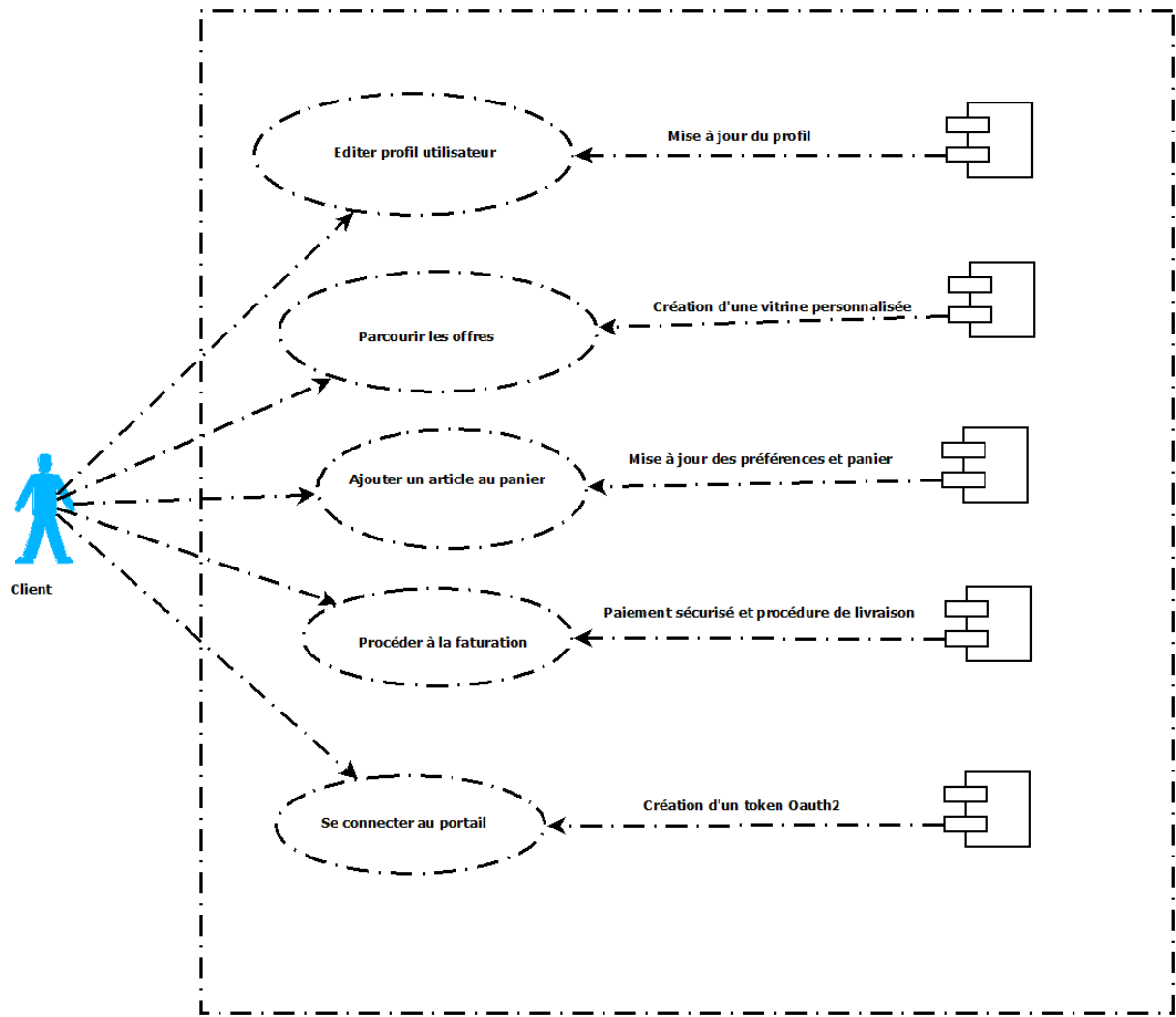
## Référence aux Conditions requises pour l'architecture

Les conditions requises pour cette architecture sont :

- Chaque service possède sa propre base de données indépendante, son propre serveur d'application, librairies..
- Les consommateurs et producteurs d'événements doivent être entièrement découplés;
- L'architecture doit être Cloud native afin de profiter de l'évolutivité rendue possible par l'architecture microservice event driven;
- La sécurité du système doit être assurée par le suivi des pratiques conseillées par un audit de sécurité informatique.

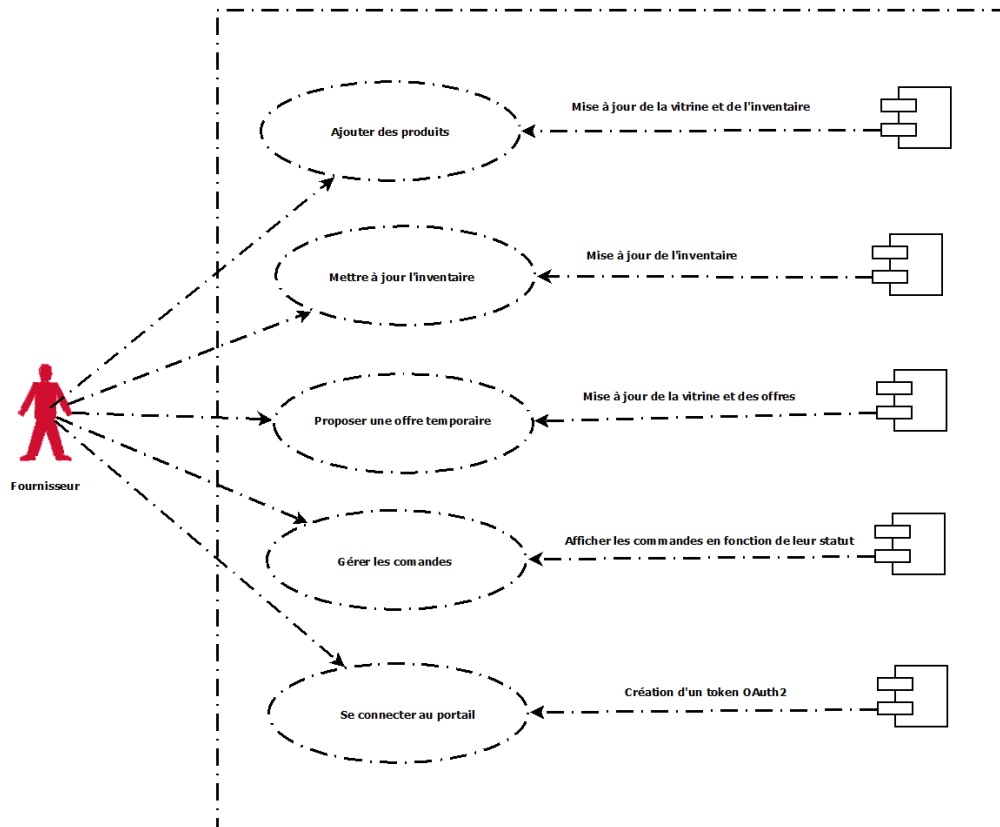
Les utilisateurs auront globalement les fonctionnalités de haut niveau suivantes :





Use case : Client

Les fournisseurs ayant accès à des fonctionnalités différentes sur l'application, le diagramme suivant décrit les fonctionnalités haut niveau accessibles :



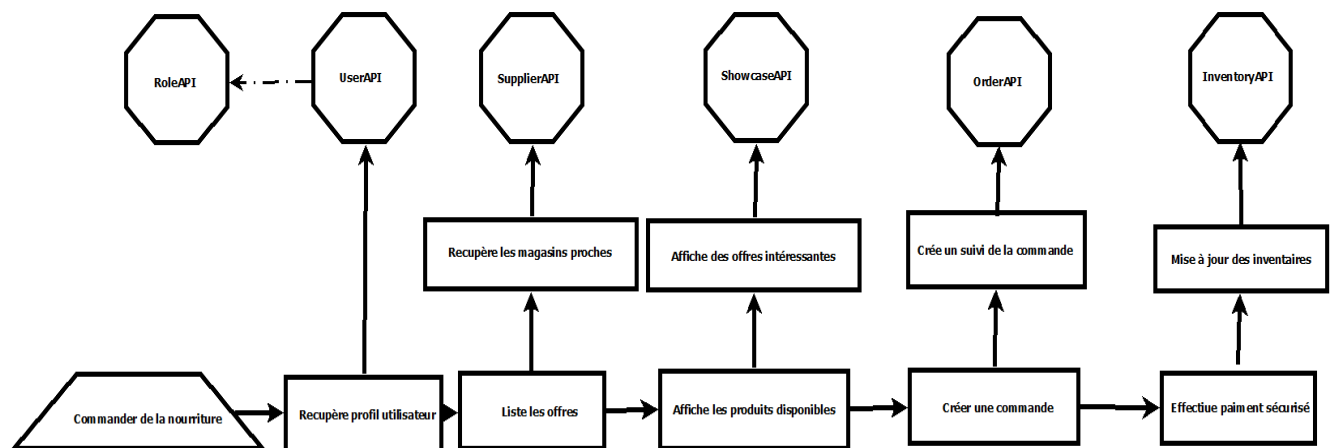
Use case : Fournisseur

## Livrables architecturaux

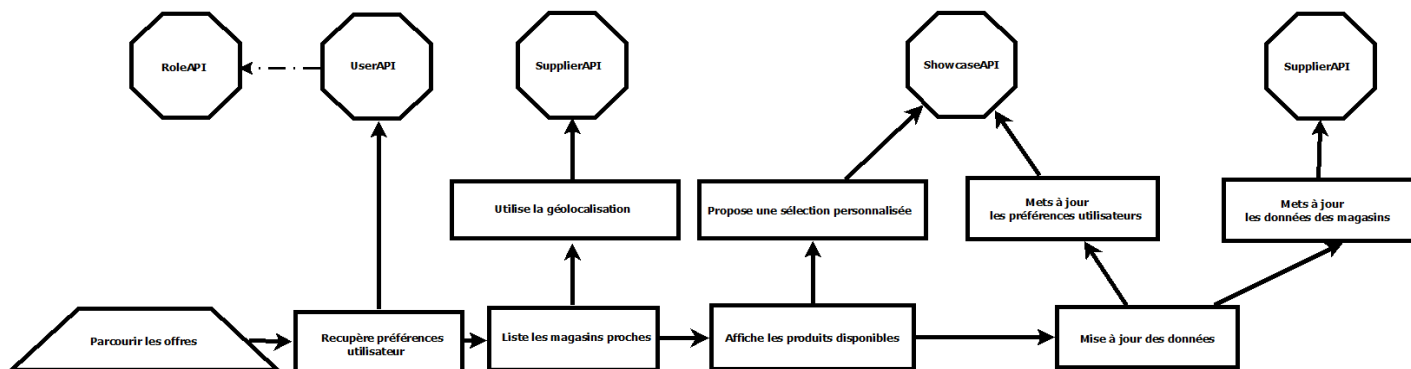
Livrables architecturaux qui satisfont les conditions requises pour le business.

## Développement de l'architecture

L'architecture devra être développée en utilisant les différents services via une API Rest. Nous avons réalisé deux exemples de cas utilisateur et avons détaillé les différents appels aux API qui existeront.



Business Services and Information Diagram - Order Food



Business Services and Information Diagram - Browse Offers

## Mesures de l'architecture cible

Métrique	Valeur cible	Justification	Notes supplémentaires
Erreurs par microservices		Les erreurs interrompent et perturbent le bon fonctionnement du système	
Traffic		Le trafic permet d'évaluer l'utilisation et la performance d'un service	Un trafic trop élevé/trop bas donnera lieu à une augmentation/diminuti

			on des ressources allouées
Latence		La latence permet d'évaluer le bon fonctionnement et l'efficacité d'un service	

## Livraison de l'architecture et métriques business

### Phases de livraison définies

Se référer au calendrier;

## Plan de travail commun priorisé

Cette section décrit toutes les activités et tous les livrables pour le travail d'architecture.

### Architecture microservice

#### Architecture microservice : Activités

Les activités relatives à la conception et implémentation d'une architecture microservices sont :

- la conception d'un diagramme de composants incluant la fragmentation des monolithes (si existants) en microservices;
- la définition des ressources de l'API REST;
- la définition des infrastructures pour chaque microservice;
- l'écriture des recommandations quant aux futures évolutions de l'architecture.

#### Architecture microservice : Livrables

Les produits de travail suivants seront créés en résultat de ce travail d'architecture :

#### *Définition des ressources de l'API*

Un tableau définissant pour chaque requête (POST, GET, PUT, DELETE) les paramètres, le micro service associé, l'endpoint et les code de retour HTTP possibles.

### ***Diagramme de composants***

Un diagramme présentant clairement et de manière accessible les microservices de l'architecture cible et leurs interactions.

### **Diagramme d'infrastructure**

Un diagramme présentant pour chaque micro service les infrastructures qui leur sont associées en termes notamment de serveur et de base de données.

## **Architecture event driven**

### **Architecture event driven : Activités**

Une architecture orientée événements se distingue par sa construction autour d'événements, de consommateurs et de producteurs. Les activités seront donc de définir clairement le statut de chacun des composants : est-il un consommateur ou un producteur ?

- Si il est un consommateur, quels types d'événements doit -il consommer et pour quelles conséquences ?
- Si il est un producteur, quels types d'événements produit-il ?

### **Architecture event driven : Livrables**

Les produits de travail suivants seront créés en résultat de ce travail d'architecture :

### ***Diagramme d'architecture event-driven***

Un diagramme décrivant le rôle des composants et leurs interactions. Chaque composant décrira le contexte dans lequel il produit un événement ou l'action effectuée après la consommation d'un événement.

## **Architecture cloud native**

### **Architecture cloud native : Activités**

Une architecture Cloud native se distingue par un design conçu pour interagir et fonctionner avec un cloud. Les activités requises afin de pouvoir profiter de l'offre Cloud sont :

- une étude de la RGPD et des contraintes associées à celle-ci, particulièrement lors de l'utilisation d'une offre Cloud;
- une étude des avantages, inconvénients et coûts de l'offre Cloud;
- le choix d'une offre Cloud avantageuse et la justification de ce choix;
- une étude et un choix parmi les différentes options offertes par le fournisseur Cloud

## **Architecture cloud native : Livrables**

Les produits de travail suivants seront créés en résultat de ce travail d'architecture :

## **Mise en place d'un cycle de vie SI**

### **Mise en place d'un cycle de vie : Activités**

Afin de mettre en place un cycle de vie de SI dans l'entreprise, différentes activités sont nécessaires :

- Mise en place d'un environnement de développement, où les développeurs pourront directement implémenter les fonctionnalités qu'ils développent. Une fois la fonctionnalité finie, elle est ajoutée à l'environnement de test.
- Mise en place d'un environnement de test, où l'équipe de test pourra tester dès que possible les nouvelles fonctionnalités implémentées par les développeurs. Ici, les bugs sont détectés, résolus et finalement validés par l'équipe de test qui s'assure que le problème n'existe plus.
- Mise en place d'un environnement de production qui héberge les fonctionnalités prêtes à être utilisées par les utilisateurs finaux.

La prochaine amélioration est d'ajouter un environnement de recette où des utilisateurs testeurs valident la bonne configuration et le bon fonctionnement des nouvelles fonctionnalités.

### **Mise en place d'un cycle de vie : Livrables**

Les livrables sont les trois environnements de recette accompagnés d'une documentation sur la manière dont ils fonctionnent et comment les utiliser.

## **Sécurisation du Système d'Information**

### **Sécurisation du SI : Activités**

Les activités liées à la sécurisation du SI sont les suivantes :

- Audit informatique par un organisme neutre;
- Mise en place de contrôles cycliques de sécurité;
- Contrôle des accès aux ressources;
- Mettre en place des éléments de sécurité tels que pare-feu, VPN..
- Mettre en place une gestion centralisée du système d'information.

### **Sécurisation du SI : Livrables**

Les livrables correspondants aux activités de sécurisation du SI sont :

- Un manuel de bonnes pratiques de sécurité à mettre en place;
- Une documentation sur les composants de sécurité mis en place;
- Un rapport d'audit (lorsque l'audit aura été effectué);
- Un diagramme de contrôle des accès aux ressources.

## Plan de communication

Description	Fréquence	Méthode	Audience	Responsable
Rapport de statut du projet	Hebdomadaire	Réunion	Développeurs, équipe de test	Responsable développeur
Rapport d'équipe	Journalier	Réunion	Développeurs	Responsable développeur
Rapport de projet	Milestones	Réunion	Développeurs, équipe de test	Responsable développeur
Rapport de fin de projet	Fin de projet	Réunion	Développeurs, équipe de test, responsable qualité	Responsable développeur
Vision	Mensuelle	Réunion	Responsable produit, commerce, développeur, architecte Manager général	Manager général
Roadmap	Toutes les deux semaines	Réunion	Equipe de management produit, responsable dev, responsable architecte	Responsable produit

Les communications non planifiées entre les membres de l'entreprise pourront être effectuées via un outil de communication commun. Nous recommandons fortement l'utilisation de [Slack](#) ou [Discord](#).

# Risques et facteurs de réduction

## Analyse des risques

ID	Risque	Gravité	Probabilité	Facteur de réduction
1.	Cyberattaques	8/10	Faible	Mesures de sécurité globale
2.	Coût du Cloud	3/10	Haute	Identification des coûts cachés
3.	Protection des données	5/10	Moyenne	Respect de la RGPD

## Hypothèses

Le tableau suivant résume les hypothèses pour cette Déclaration de travail d'architecture.

ID	Hypothèse	Impact	Propriétaire
1	Plateforme actuelle en mode maintenance	Arrêt de l'amélioration de la plateforme actuelle (jusqu'au changement de plateforme)	
2	Coexistence des plateformes	Mise à disposition de nouvelle fonctionnalités et montée en charge progressive	
3	Approche lean	Efficacité et autonomie des équipes conservée	



# Critères d'acceptation et procédures

## Métriques et KPIs de l'État Cible de l'Architecture

Les métriques suivantes seront utilisées pour déterminer le succès de ce travail d'architecture :

### Métriques de livraison de l'architecture et du business

Métrique	Valeur cible	Justification
Rapidité d'implémentation de la géolocalisation	Un mois	La géolocalisation est une feature phare de l'architecture cible et doit être disponible rapidement.
Rapidité de déploiement	Moins d'une demi-heure	La complexité du déploiement freine largement l'évolutivité du système.
Rapidité de modification du hardware	Moins d'une demi-heure	Il doit être facile d'ajouter ou de supprimer de l'architecture des serveurs, bases de données..
Accessibilité et résilience	Sur une semaine : accessible à 100% Rapidité de remise en service après perte d'un serveur : moins d'une heure	Le site doit être accessible tous les jours à toutes heures malgré la perte d'un élément de l'architecture
Réduction du nombre de bugs en environnement de production	Réduction du nombre de bugs en production par 100%	L'évolutivité du système requiert une stabilité importante face aux bugs.
Réduction des dépendances entre les services	Services touchés par la modification d'un service : 1 maximum (le service modifié)	L'évolutivité du système doit être améliorée par l'absence de dépendance entre les services

## Procédure d'acceptation

La validation de la conformité du système aux valeurs cibles sera acceptée par "sample", c'est-à-dire en créant des valeurs de tests suffisants afin d'évaluer le niveau global pour chaque métrique.

## Procédures de changement de périmètre

Nous recommandons la procédure suivante en cas de changement de périmètre.

1. Définir le changement de périmètre (nature et périmètre);
2. Etablir les acteurs clés, leurs rôles et responsabilités et les impliquer le plus tôt possible;
2. Faire un inventaire informatique détaillé des composantes informatiques (applications, infrastructures, licenses..)
3. Etablir un plan d'action afin de rendre indépendant le nouveau système d'information;
4. Analyser les impacts sur les contrats et projets existants

## Conditions requises pour la conformité

Démontrer la conformité signifie qu'une organisation s'engage à gérer son activité conformément aux réglementations et aux normes en vigueur. Il est important d'être capable de démontrer que l'entreprise est à la hauteur des exigences de conformité et que des procédures sont en place afin de garder un niveau au moins aussi élevé.

Pour cela, nous établirons une carte de conformité pour le COBIT 5 et une carte de conformité pour la RGPD; nous testerons ensuite les conformités via des projets.

## Développement et propriété de l'architecture

L'architecture est Cloud native et héberge une solution micro-services composée d'une multitude de services indépendants possédant chacun une base de données. La redirection vers le service voulu sera effectuée par l'API Gateway.

Les serveurs et les bases les plus vitales seront multipliés pour réduire les risques d'interruption de service.

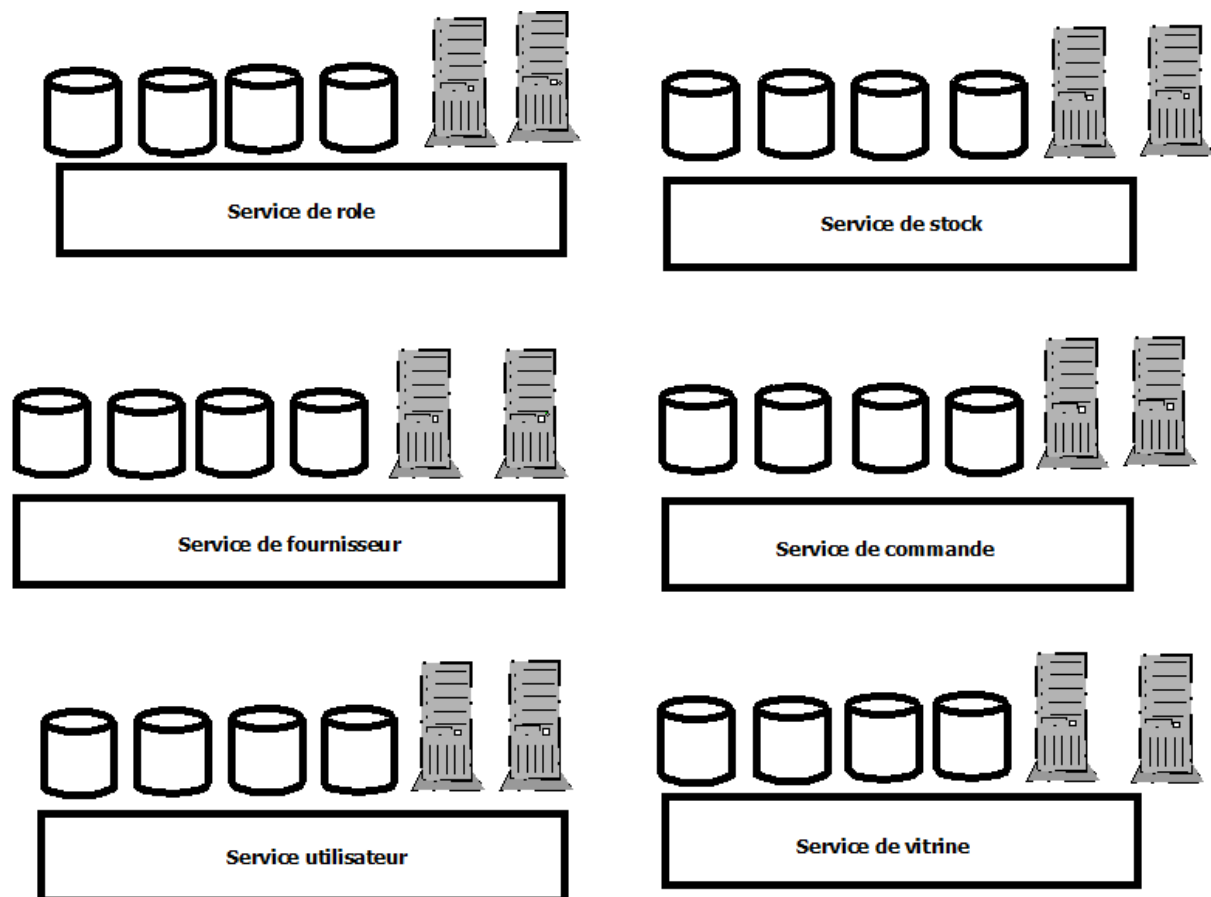
En terme de sécurité, nous utiliserons les outils et infrastructures suivants :

- Un SSO afin de contrôler et faciliter les accès;
- Une pare-feu afin de contrôler les accès au réseau;
- Un point d'entrée qui contrôlera les requête HTTP;

- un load balancer afin de gérer efficacement la charge sur les serveurs;

L'utilisation d'un orchestrateur de déploiement tel que Kubernetes permettra de gérer pleinement les différents serveurs et leur montée en charge. Le but de cet outil est d'avoir un contrôle total sur les

Les infrastructures requises pour le bon développement du parc informatique sont décrites dans le diagramme d'infrastructure suivant :



*Diagramme d'infrastructure*

Nous recommandons ainsi la stratégie de back-up 4-3-2 qui consiste à établir pour tous les composants importants de l'application :

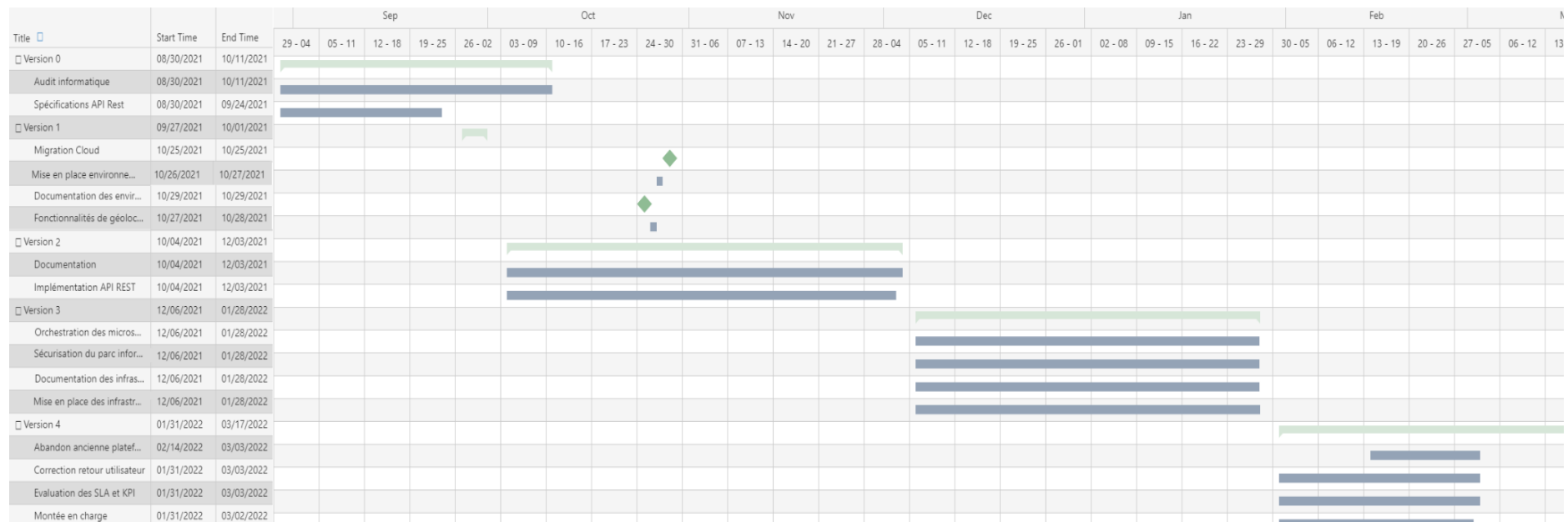
- Quatres copies des données;
- Trois lieux de stockage différents (sur place, dans un datacenter, dans un cloud);
- Deux lieux sur trois sont "off-site".

Cette stratégie permet une très forte résilience du système face aux attaques ou aux erreurs informatiques tout en permettant une récupération rapide des données quelle que soit la situation.

Nous ajouterons à ces précautions la présence de deux serveurs afin de pouvoir compenser la perte de l'un d'entre eux. L'objectif est d'avoir dès que possible plusieurs serveurs actifs pour profiter du load-balancing. La solidité du système sera assurée en ayant en plus plusieurs serveurs de remplacement prêts à prendre le relai au besoin.

# Calendrier

## Diagramme de Gantt



Voici une prévision du déroulement des six premiers mois du projet de changement d'architecture de Foosus. Nous avons décidé de délivrer les premières phases sous forme de quatre versions :

Le début du projet commencera par le début de l'audit informatique par un organisme indépendant ainsi que le commencement des spécifications des APIs. Une fois les spécifications faites, il sera possible de coder un "Minimum Viable Product" assez rapidement, afin de mettre rapidement en production le nouveau produit. Les nouvelles fonctionnalités, la montée en charge et la couverture de l'ensemble des fonctionnalités proposées par le système précédent seront ajoutées petit à petit lors de la version 2 et 3.

La version 1.0 permettra de mettre en place les outils les plus importants et de commencer rapidement à travailler avec ces nouveaux éléments. Ainsi, la mise en place des environnements et la migration Cloud permettront, dès les premières semaines, de livrer un produit plus abouti plus rapidement. Nous implémenterons également la géolocalisation car c'est une feature majeure et très attendue.

La version 2.0 sera principalement concentrée sur la spécification, le développement et la documentation et l'implémentation de l'API REST. Celle-ci sera sous-divisée en microservices et fonctionnera selon le modèle d'architecture "event-driven". Nous profiterons ensuite de la fin du développement de l'API pour commencer l'audit informatique.

La version 3.0 permettra de finaliser la mise en place d'une architecture sécurisée, résiliente et évolutive. Les éléments architecturaux de sécurité recommandés par l'audit informatique seront mis en place et documentés, la montée en charge du nouveau système sera automatisée grâce à la configuration d'un API Gateway et d'un orchestrateur de conteneur.

Finalement, la version 4.0 permettra l'évaluation finale de la première partie du projet et, si elle est concluante, l'arrêt définitif du système précédent. Le but ici sera d'effectuer des retours utilisateurs et business, d'évaluer les performances et les indicateurs de réussite en utilisant les concepts de la méthode Lean pour perfectionner le système en apprenant des erreurs constatées.

## Personnes approuvant ce plan

Valideur	Domaine de responsabilité	Date