

# **Contrat de Conception et de Développement de l’Architecture**

* *Projet :* nouvelle application
* *Client :* Foosus
* *Préparé par :* Morgan Facorat

*Note : Ce document fournit un modèle générique. Il pourra nécessiter des modifications pour correspondre à un client et une situation de projet spécifiques.*

# **Table des matières**

1. Objet de ce document
2. Introduction et contexte
3. La nature de l’accord
4. Objectifs et périmètre
5. Description de l’architecture, principes stratégiques et conditions requises
6. Livrables architecturaux
7. Plan de travail commun priorisé
8. Plan de communication
9. Risques et facteurs de réduction
10. Hypothèses
11. Critères d’acceptation et procédures
12. Procédures de changement de périmètre
13. Calendrier
14. Phases de livrables définies
15. Personnes approuvant ce plan

# **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**

La plateforme actuelle de Foosus a atteint un point au-delà duquel elle ne peut plus soutenir les projets de croissance et d'expansion de l'entreprise. Après plusieurs années de développement, notre solution technique complexe n'évolue plus au rythme de l'activité et risque d'entraver notre croissance. Les études de marché et les analyses commerciales montrent que nos clients souhaitent acheter local et soutienne les producteurs locaux.

La dette technologique accumulée durant toutes ces années nous force à nous orienter vers une solution d’actualité, stable, et évolutive.

# **La Nature de l’accord**

La nature de l’accord entre les partenaires de développement et les sponsors est tacite. Les développeurs se doivent de concevoir une architecture pouvant atteindre les différends business définis par les sponsors.

# **Objectifs et périmètre**

## **Objectifs**

Les objectifs business de ce Travail d’architecture sont les suivants :

### ***Objectif Business 1***

Foosus recherche une augmentation journalière du nombre de ses clients, l’objectif donné est de 10%.

### ***Objectif Business 2***

Foosus recherche à obtenir une utilisation régulière de l’application vis-à-vis des clients, l’objectif donné est que 70% des utilisateurs consultent l’application au minium 2 fois par semaine.

## **Périmètre**

A consulter dans le document « Statement of Architecture Work FR » disponible dans le drive

## **Parties prenantes, préoccupations et visions**

*Le tableau suivant montre les parties prenantes qui utiliseront 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** |
| Pete Parker  *Engineering Owner* | Applicatif | Technique |
| Natasha Jarson  *CIO* | Applicatif | Technique et technologique |
| Ash Callum  *CEO* | Globale | Business |
| Daniel Anthony  *CPO* | Utilisateur | Technique |
| Christina Orgega  *CMO* | Marketing | Business |
| Jo Kumar  *CFO* | Financière | Business |

# **Description de l’architecture, principes stratégiques et conditions requises**

## **Description**

L’architecture sélectionnée est une architecture micro-service sous java. Nous avons choisi java car son interopérabilité est ce qui convient parfaitement à notre projet. La logique micro-service nous permets d’encapsuler différentes fonctionnalités et les modifier sans pour autant impacter les autres. L’architecture micro-service est une architecture facilement maintenable et évolutive, elle répond à tous nos besoins.

## **Principes stratégiques**

Notre stratégie est basée sur différents points détaillés dans le tableau suivant :

|  |  |
| --- | --- |
| Point clé | Description |
| Clarté | L’architecture se doit d’être développé d’une façon claire et logique |
| Efficacité | L’architecture est la plus optimiser et efficace possible |
| Maniabilité | L’architecture est conçue de maniable |

## **Référence aux Conditions requises pour l’architecture**

Afin de pouvoir assurer une certaine cohérence du code, chaque développeur sera formé et aura à disposition différents documents liés à la syntaxe.

Nous avons trouvé judicieux de mettre à disposition de nos équipe un serveur de test, où ils pourront tester et déployer l’architecture avant la mise en service.

# **Livrables architecturaux**

Nous allons concevoir différents types de livrables architecturaux durant le projet. La liste des différents livrables produits est disponible dans le document déclaration de travail d’architecture.

## **Développement de l’architecture**

Pour rappel voici les caractéristiques majeures du projet.

|  |  |
| --- | --- |
|  |  |
| Langage | Java |
| Framework | Sprint boot |
| Architecture type | Micro-service |
| Méthode | Agile |
| SGBD | SQL server |

Le fait d’utiliser les différentes caractéristiques citées, nous force à concevoir des livrables spécifiques. Nous avons décidé de lister une majeure partie d’entre eux.

* Script SQL
* Fichier bin / script de l’application
* Documentation technique et fonctionnelle
* Code source

## **Mesures de l’architecture cible**

Nous pouvons retrouver les différentes métriques de mesures dans le document « Spécification des Conditions requises pour l’Architecture

## **Livraison de l’architecture et métriques business**

Nous avons décidé de livrer l’architecture et les métriques business en respectant le macro-planning définis dans le document « Déclaration de travail d’architecture ». Tout cela sera effectuer pendant la phase finale qui est le déploiement.

## **Phases de livraison définies**

Les différentes phases de livraison ont été définis dans le document « déclaration de travail d’architecture »

# **Plan de travail commun priorisé**

Nous nous sommes basés sur la méthodologie ADM comme norme, chaque livrable conçue par chacune des étapes, ont été définie dans le document « déclaration de travail d’architecture »

# **Plan de communication**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Évènements | Réunion hebdomadaire | Tous les jeudis après-midi |
| Canaux | WhatsApp, Email, iMessage | Différents groupes seront créés avec les parties prenantes différentes |
| Formats | PDF, DOCX, PNG | Différents documents seront disponibles, en fonction des autorisations |
| Contenu | Livrable, artefacts | Disponible depuis le drive du projet |
| Durée et effort | Sprint de 2 semaines maximum |  |
| Collaboration | Les équipes du CPO et CIO |  |

## 

## **Analyse des risques**

Les différents risques du projet ont été hiérarchisés par rapport à leurs criticités.

Criticité = Probabilité \* Gravité

Tableau d’analyse des risques :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risque | Probabilité | Gravité | Criticité | Responsable | Prévention | Réparation |
| Dépassement budgétaire | 4 | 5 | 20 | Architecte logiciel | Gestion du temps | -Négocier avec la direction une augmentation budgétaire |
| Dépassement du délai de livraison | 4 | 5 | 20 | Architecte logiciel | Mise en place de réunions hebdomadaires afin de vérifier l’avancement du projet | -Application de pénalités contractuelles |
| -Mise en place d’astreinte |
| Ajout d’une nouvelle fonctionnalité | 4 | 5 | 20 | CIO et CPO | Concevoir l’application de façon évolutive | -Augmentation du budget  -Augmenter les délais |
| Climat conflictuel dans l’équipe | 2 | 4 | 8 | Architecte logiciel | Mise en place de session de dialogue récurrente | -Changement d’équipes |
| Membre de l’équipe indisponible | 1 | 4 | 4 | Architecte logiciel | Pair programming | -Faire appel à un remplaçant |
| Le projet ne répond pas aux attentes de la direction | 1 | 5 | 5 | Architecte logiciel | Réalisation de points réguliers avec la direction. | -Revoir les documents |

# **Critères d’acceptation et procédures**

## **Métriques et KPIs de l’État Cible de l’Architecture**

De plus, les métriques suivantes seront utilisées pour déterminer le succès de ce travail d’architecture :

|  |  |
| --- | --- |
| Indicateur | Changement souhaité pour l'indicateur |
| Compatibilité web | L’application devra être compatible avec 99% des navigateur disponible sur le marché |
| Compatibilité support | L’architecture devra être compatible avec les différents os disponible |
| Diminution du nombre d’incident | L’architecture est confrontée à 1 incident maximum par moi |

## **Métriques de livraison de l’architecture et du business**

|  |  |
| --- | --- |
| **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. |

## **Procédure d’acceptation**

La procédure d’acception de la déclaration de travail d’architecture est une procédure contenue en 2 phases.

Phase 1 : elle consiste à déployer la nouvelle solution ; et attendre une période de 3 mois.

Phase 2 : Après cette période une campagne de satisfaction utilisateur débutera. Un résultat supérieur à 80% de satisfaction, nous permettra de valider le projet.

# **Procédures de changement de périmètre**

Afin d’être préparés à toutes éventualités, nous avons trouvé judicieux de concevoir une procédure destinée aux changements de périmètre.

Celle-ci est divisée en deux phases distinctes.

Phase préliminaire : Évaluation du nouveau périmètre, cette phase a pour but de déterminer si le changement de périmètre est réellement nécessaire. Le nouveau périmètre se doit de répondre aux différentes questions liées au formulaire de changement de périmètre, disponible sur le drive.

Phase 1 : Approbation, après avoir compléter et répondu aux questions du formulaires détaillant et validant le nouveau périmètre, la signature du CIO et du CEO est nécessaire avant de pouvoir enclencher un changement de périmètre.

Phase 2 : Application du nouveau périmètre.

## **Conditions requises pour la conformité**

Nous avons décidé d’utiliser un qcm à point, afin de déterminer si toutes les conditions requises pour la conformité sont réunis. Ce formulaire a été conçue avec les différentes parties intéressées.

Grace à celui-ci nous avons la possibilité d’évaluer le niveau de conformité (le qcm est disponible dans le drive).

## **Développement et propriété de l’architecture**

Nous avons trouvé judicieux d’ajouter une matrice RACI des différentes parties prenantes. Grâce à cette matrice, nous pouvons identifier rapidement les parties prenantes exerçantes un pouvoir important sur le projet.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Pete | Morgan | Natasha | Ash | Daniel | Christina | Jo |
| Déterminer les besoins applicatifs | C | R | A | C | C | I | I |
| Évaluation des risques | C | R | A | C | C | I | C |
| Définition du périmètre | C | R | A | C | C | C | C |
| Modéliser les processus | C | R | A | I | C | I | I |

# **Calendrier**

# 

# **Personnes approuvant ce plan**

|  |  |  |
| --- | --- | --- |
| **Validateur** | **Domaine de responsabilité** | **Date** |
| Ash Callum | *Domaine business* |  |
|  |  |  |