



Déclaration de Travail d'Architecture

Projet : Foosus Client : Foosus

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. Déclaration de travail d'architecture
3. Objectifs et périmètre
4. Rôles et responsabilités
5. Approche architecturale
6. Plan de travail
7. Risques et facteurs de réduction
8. Critères d'acceptation et procédures
9. Approbations signées

Information sur le document

<i>Nom du projet</i>	Foosus

<i>Préparé par :</i>	Degruson Julien
<i>N° de version du document :</i>	0.1
<i>Titre :</i>	<i>Déclaration de travail d'architecture</i>
<i>Date de version du document :</i>	08/03/2023
<i>Revu par :</i>	
<i>Date de révision :</i>	
<i>Liste de distribution :</i>	
De :	
Date :	
Email :	
Pour Action :	
Date de rendu :	
Email :	
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

Ce document est une Déclaration de travail d'architecture pour le Foosus.

La Déclaration de travail d'architecture définit le périmètre et l'approche qui seront utilisés pour mener à bien un projet d'architecture. La Déclaration de travail d'architecture constitue habituellement le document qui permet de mesurer la réussite de l'exécution du projet d'architecture et peut former la base de l'accord contractuel entre le fournisseur et le consommateur de services d'architecture. En général, toutes les informations de ce document doivent se situer à un haut niveau.

La Déclaration de travail d'architecture peut être documentée sur un wiki ou l'intranet plutôt que par un document texte. Pour faire encore mieux, vous pouvez utiliser un outil sous licence TOGAF pour restituer cette production.

Ce modèle montre les contenus « typiques » d'une Déclaration de travail d'architecture et peut être adapté pour être aligné sur toute adaptation TOGAF implémentée.

Déclaration de travail d'architecture

Requête du projet et contexte

Foosus est une start-up âgée de 2 ans, spécialisée dans le secteur de l'alimentation durable. L'incertitude est un facteur important de ce secteur et il faut s'assurer que les risques de prise de décisions techniques difficiles à inverser ont été minimisés !

Ils ont besoin d'une nouvelle architecture qui peut évoluer avec l'entreprise, prendre en charge un nouvel emplacement géographique et fournir la fiabilité nécessaire à leurs clients, fournisseurs et consommateurs.

Vue d'ensemble

Mieux servir leurs clients et innover de façon responsable, de manière à maximiser leurs capacités et aider l'organisation à continuer à croître en accord avec leur feuille de route générale. Ils souhaitent revenir à un niveau où ils pourront innover rapidement. Il est important de repartir sur une nouvelle base pour leur initiative la plus récente, et d'éviter de répéter des choix d'architecture qui ne sont plus adaptés à une plateforme scalable.

Créer une plateforme de commerce électronique polyvalente pour faire passer l'entreprise à un niveau supérieur. L'efficacité, la flexibilité et des approches cohérentes dans la prise de décision sont nécessaires pour qu'ils puissent concurrencer les grandes entreprises mondiales de commerce électronique qui dominent le marché de l'alimentation durable.

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.

Objectifs et périmètre

Objectifs

Les objectifs business de ce travail d'architecture sont les suivants :

<i>Objectif Business</i>	<i>Notes</i>
Évoluer avec leur base de clientèle	
Améliorer les performances du système actuel	
Une plateforme sécurisée	
Une technologie transparente	
Expérimentation	

Indicateurs de réussite

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.

Périmètre

Le scope de ce travail d'architecture concerne l'ensemble des projets de Foosus. Il s'agit d'un travail de fond pour restructurer l'architecture globale de l'entreprise.

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.

Partie prenante	Préoccupation	Vision
Direction	Réussite du projet	Technique ou Fonctionnelle en fonction du scope de la personne
Équipes de développement	Développement de la solution	Technique
Utilisateurs	Achat de produit	Fonctionnelle
Fournisseurs	Vente de produit	Fonctionnelle

Approche managériale

L'approche managériale privilégiée jusqu'ici est le format agile, il n'y a pas de contre indication à continuer sur ce format. L'approche agile va permettre de concentrer les ressources sur les projets prioritaires et pouvoir changer rapidement de priorité au fur et à mesure des besoins.

Procédures de changement de périmètre

Identifier le changement de périmètre : Tout d'abord, il est important d'identifier clairement la nature du changement de périmètre et les raisons qui le motivent. Cette étape permet de comprendre l'impact potentiel du changement sur le système et de déterminer les parties prenantes qui seront affectées.

Évaluer l'impact : Une fois le changement de périmètre identifié, il est important d'évaluer son impact potentiel sur le système. Cette étape peut inclure l'analyse des exigences du système, la

modélisation du flux de données, l'identification des risques, etc. Cette évaluation permet de comprendre les implications du changement sur le système et de déterminer les actions à entreprendre.

Proposer des solutions : Après avoir évalué l'impact, il est temps de proposer des solutions pour répondre au changement de périmètre. Les solutions peuvent inclure l'ajout ou la modification de fonctionnalités, l'adaptation des exigences, la mise à jour de la documentation, etc. Il est important d'impliquer les parties prenantes dans ce processus pour garantir que les solutions proposées répondent à leurs besoins.

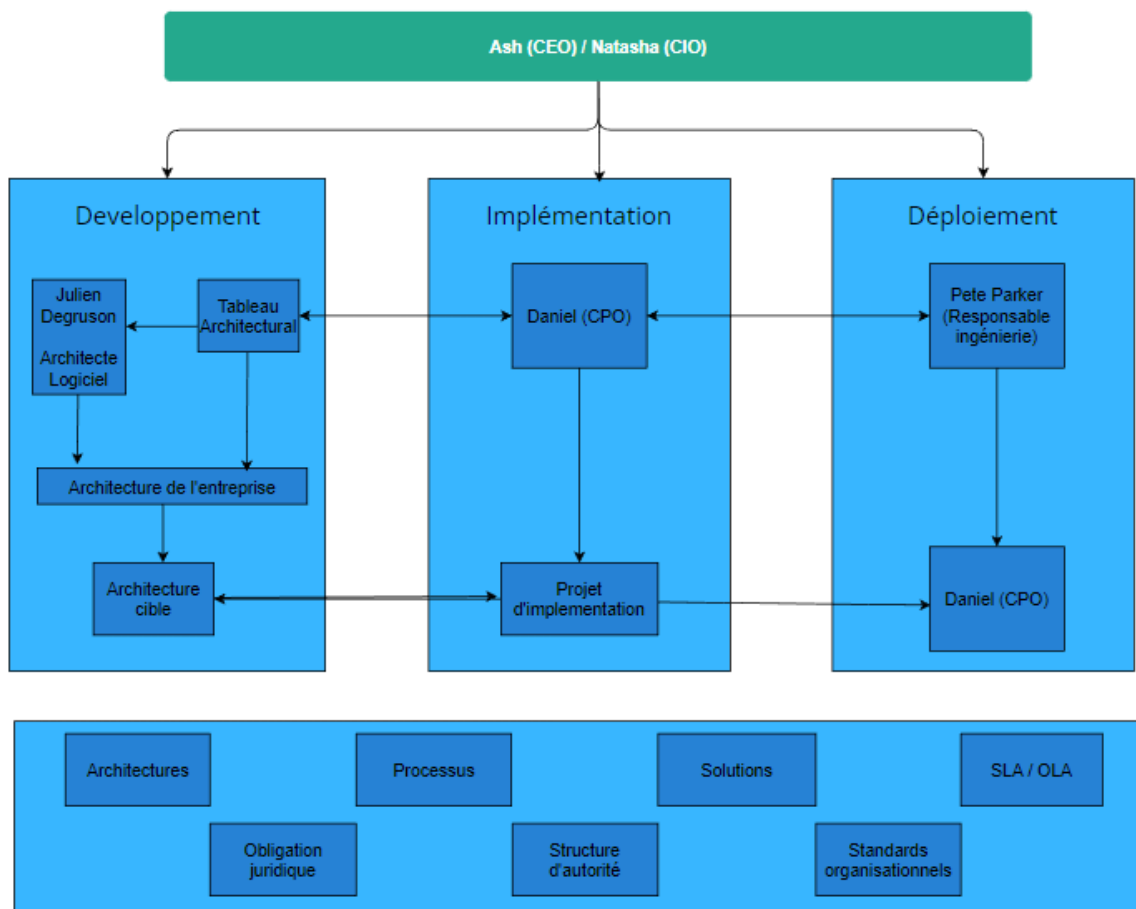
Obtenir l'approbation : Une fois les solutions proposées, il est nécessaire d'obtenir l'approbation des parties prenantes avant de procéder à toute modification. Cela peut impliquer l'approbation de la direction, des clients, des utilisateurs finaux, etc.

Mettre en œuvre le changement : Une fois que le changement de périmètre est approuvé, il est temps de le mettre en œuvre. Cela peut inclure la modification du code source, la mise à jour de la documentation, la formation des utilisateurs finaux, etc. Il est important de suivre des processus de contrôle de qualité rigoureux pour garantir que la modification est réalisée de manière fiable et sans affecter la stabilité du système.

Vérifier l'efficacité : Enfin, il est important de vérifier l'efficacité du changement de périmètre. Cette étape peut inclure des tests d'acceptation, des tests de performance, des tests de sécurité, etc. Cette vérification permet de s'assurer que le changement répond aux objectifs fixés et qu'il n'a pas d'effets secondaires imprévus.

Rôles et responsabilités

Structure de gouvernance



Process du projet

La documentation ainsi que le code doivent être commit et push sur un système de version de contrôle tel que git. Afin de pouvoir très rapidement importer et modifier ceux-ci.

Comme précisé plus haut, les équipes s'organiseront autour de la méthode agile, et devront suivre au maximum les préceptes de celle-ci.

Assurance qualité :

Planification de l'assurance qualité : Le plan d'assurance qualité est développé pour

décrire les normes de qualité à suivre, les activités de test et d'inspection, ainsi que les outils et techniques de gestion de la qualité à utiliser tout au long du projet.

Exécution de l'assurance qualité : Les activités d'assurance qualité sont effectuées pendant tout le projet pour garantir que les normes de qualité sont respectées. Cela peut inclure des activités de test, d'inspection et de vérification pour s'assurer que les livrables répondent aux spécifications et aux attentes.

Surveillance et rapport : Les résultats des activités d'assurance qualité sont surveillés et rapportés à l'équipe de projet et aux parties prenantes. Les problèmes de qualité sont signalés, les actions correctives sont prises et les améliorations sont identifiées et mises en œuvre.

Gestion des modifications :

Identification des changements : Les changements potentiels sont identifiés et documentés. Cela peut inclure des demandes de modification de la portée, des changements de calendrier ou des modifications de budget.

Évaluation des changements : L'impact des changements proposés est évalué pour déterminer s'ils peuvent être intégrés dans le projet sans compromettre la qualité ou la livraison.

Approbation des changements : Les changements sont soumis aux parties prenantes pour approbation. Les changements qui sont approuvés sont documentés et intégrés dans le plan de gestion de projet.

Mise en œuvre des changements : Les changements approuvés sont mis en œuvre conformément au plan de gestion de projet et aux normes de qualité établies.

Vérification des résultats : Les résultats des changements sont vérifiés pour s'assurer qu'ils répondent aux attentes. Les résultats sont documentés et rapportés à l'équipe de projet et aux parties prenantes.

Rôles et responsabilités (RACI)

Activités / Rôles	Responsable	Approbateur	Consulté	Informé
Politique qualité	Direction	Architecte logiciel	Développeurs, Utilisateurs	-
Conception	Architecte logiciel	Direction	Développeurs, Utilisateurs	-
Développement	Développeurs	Architecte logiciel, Direction	-	Utilisateurs
Déploiement	Développeurs	Architecte logiciel,	Utilisateurs	-

		Direction		
Maintenance	Développeurs	Architecte logiciel, Direction	Utilisateurs	-

Approche architecturale

Process d'architecture

La méthode de développement d'architecture TOGAF (ou ADM pour « Architecture Development Method ») décrit une méthodologie des meilleures pratiques pour le développement architectural. Néanmoins, toutes les phases ne sont pas également pertinentes pour chaque projet. Le tableau ci-dessous décrit l'utilisation de l'ADM pour ce projet spécifique.

Phase	Entrée/Sortie	Notes
Préliminaire	Principes architecturaux	
A —Vision de l'architecture	Documents de définition d'architecture	
B —Architecture business	Documents de définition d'architecture	<<Ligne de base et/ou cible>>
C — Architecture des systèmes d'information	Documents de définition d'architecture	
D — Architecture technologique	Documents de définition d'architecture	<<Ligne de base et/ou cible ?>>
E —Opportunités et solutions	Roadmap de l'architecture	

F —Planning de migration	Plan de migration et d'implémentation	
G —Gouvernance de l'implémentation	Contrats d'architecture	
H —Management du changement d'architecture	Mise à jour du document de définition d'architecture (si nécessaire) Mise à jour des contrats d'architecture (si nécessaire)	
Management des conditions requises	Spécification des conditions requises pour l'architecture	

Solution

Pour une entreprise qui s'est dispersée dans plusieurs petits projets différents, l'architecture microservices peut aider à rationaliser les efforts de développement. Elle peut permettre de réduire la complexité en cassant l'application en services plus petits et plus simples. Cela peut également aider à réduire les coûts de maintenance et de développement en éliminant les dépendances inutiles entre les différents projets.

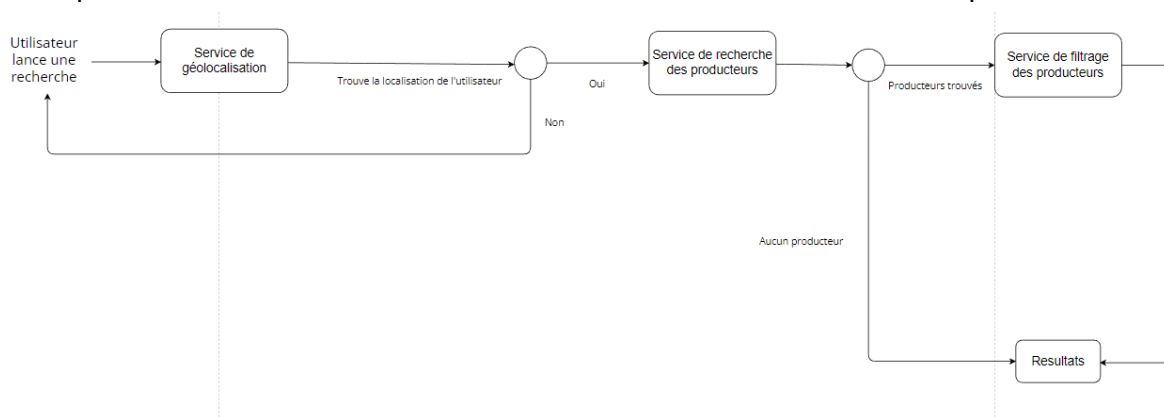
L'architecture microservices est un style d'architecture logicielle qui divise une application en services indépendants, chaque service étant responsable d'une tâche spécifique. Ces services communiquent entre eux via des API bien définies, généralement en utilisant des protocoles légers comme HTTP ou REST.

L'un des avantages clés de l'architecture microservices est qu'elle permet une évolutivité plus facile et une meilleure gestion des risques. Étant donné que chaque service est indépendant, il est plus facile de modifier ou de mettre à jour un service spécifique sans affecter l'ensemble de l'application. Cela signifie également qu'il est plus facile d'ajouter de nouveaux services pour répondre à des besoins spécifiques.

L'architecture microservices facilite également la collaboration entre les équipes de développement. Les équipes peuvent travailler sur des services indépendants les uns des autres, ce qui peut aider à réduire les conflits de fusion et à permettre une livraison plus rapide de nouvelles fonctionnalités.

Enfin, l'architecture microservices peut aider à améliorer la résilience de l'application. En cas de panne ou de dysfonctionnement d'un service, l'ensemble de l'application ne sera pas affecté. Au lieu de cela, seuls les services qui dépendent du service défectueux seront touchés. Cela peut aider à réduire les temps d'arrêt et à améliorer la fiabilité de l'application dans son ensemble.

En conclusion, l'architecture microservices peut être une solution intéressante pour une entreprise qui souhaite restructurer plusieurs petits projets différents. Elle peut aider à améliorer l'évolutivité, la collaboration, la simplicité, la résilience et la réduction des coûts, ce qui peut se traduire par une meilleure satisfaction des clients et une meilleure efficacité opérationnelle.



Contenu de l'architecture

Le cadre de contenu d'architecture TOGAF (ou ACF pour « Architecture Content Framework ») fournit une catégorisation des meilleures pratiques pour le contenu de l'architecture. Néanmoins, tous les éléments ne sont pas également pertinents pour chaque projet. Le tableau ci-dessous décrit les zones de contenu pertinentes pour ce projet spécifique.

Zone de contenu	Entrée/Sortie	Notes
Principes, Vision, et Conditions requises de l'Architecture	Architecture Content framework	Introduction
Architecture Business	Architecture Content framework	Contenu de métamodèle Artéfacts architecturaux

		Délivrables architecturaux
Architecture des systèmes d'information — Données	Architecture Content framework	Contenu de métamodèle Artéfacts architecturaux Délivrables architecturaux
Architecture des systèmes d'information — Applications	Architecture Content framework	Contenu de métamodèle Artéfacts architecturaux Délivrables architecturaux
Architecture technologique	Architecture Content framework	Contenu de métamodèle Artéfacts architecturaux Délivrables architecturaux
Réalisation de l'architecture	Architecture Content framework	Building blocks

Méthodologies pertinentes et normes de l'industrie

La vision de Foosus depuis ses debuts a été de laisser libre court aux envies et idées de ses collaborateurs, sans aucune restrictions. Cela a permis de créer un environnement de travail agréable et de faire germer des projets intéressants. La contrepartie est qu'il n'y a aucune vision commune sur la partie technique de tous ces projets.

Le but de Foosus aujourd'hui est de réduire cette dette technique. Nous disposons donc de 6 mois et d'un budget de 50000\$ pour définir l'architecture et préparer un projet de suivi afin de développer un prototype.

La réussite de ce projet passera par la mise en place de normes de l'industrie ainsi que l'utilisation de technologie cohérente avec la vision de Foosus.

Plan de travail

Élément de travail 1

Activités

Étude de la dette technique de chaque projet.

Livrables

- Listing des projets ainsi que de leur dette technique, et optionnellement des possibilités d'amélioration.

Élément de travail 2

Activités

Étude des choix de technologies et des good practices applicables aux projets de Foosus.

Livrables

- Listing des principes architecturaux TOGAF
- Listing des choix de technologies et des good practices

Élément de travail 3

Activités

En se basant sur tous les éléments précédents, identifier les meilleurs choix techniques.

Livrables

Élément de travail 4

Activités

Préparation de l'organisation à mettre en place pour incorporer les choix de l'élément 3 au sein de Foosus.

Livrables

Plan d'action sur les étapes à suivre pour la mise en place du projet

Plan de communication

Reunion de lancement du projet : Attribution des roles et des objectifs de chacun. Présentation des objectifs et des principes architecturaux choisis. Explication du plan d'action de l'étape 4.

Reunion des equipes de developpement : Entrer plus en détails sur les choix technologiques, sur les principes agiles a suivre pour parvenir aux objectifs : Revue de code, pair programming..

Reunion mensuel : S'assurer que la direction est toujours la bonne.

Canaux

Mail pour les informations dont on doit garder l'historique

Reunion pour les besoins de brainstorming

Slack ou autre messagerie direct pour l'echange d'information rapide

Collaboration

Plan et calendrier du projet

Cartographie de l'existant

Etudes des possibilités

Décision sur les choix techniques

Gouvernance des implémentations

Risques et facteurs de réduction

Analyse des risques

ID	Risque	Gravité	Probabilité	Facteur de réduction	Propriétaire
1.	DDOS	Elevée	Moyenne	Load Balancing	Equipe infrastructure
2.	Infiltration malveillante dans le système	Elevée	Moyenne	Audit	Equipe cyber sécurité
3	Faillle dans la nouvelle architecture	Elevée	Moyenne	Garder les systèmes en parallèle le temps de tester totalement la nouvelle architecture	Equipe dev et test

Hypothèses

Le tableau ci-dessous résume les hypothèses pour cette Déclaration de travail d'architecture :

ID	Hypothèse	Impact	Propriétaire
1.	DDOS	Site qui met du temps à répondre, voir qui ne répond plus	Equipe infrastructure
2.	Infiltration malveillante dans le système	Mot de passe et information personnelle qui fuite. Arnaques.	Equipe cyber sécurité
3	Faillle dans la nouvelle architecture	Bug qui entraine l'incapacité d'utiliser certaines parties du site	Equipe dev et test

Critères d'acceptation et procédures

Métriques et KPIs

Il faudra garder en tête pour chaque étape les conditions et objectifs, ainsi que les indicateurs de réussites :

Indicateurs de réussite

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

Réception de la déclaration de travail d'architecture : Lorsque la déclaration de travail d'architecture est soumise, elle est examinée par l'équipe chargée du projet.

Évaluation préliminaire : Une première évaluation est effectuée pour déterminer si la déclaration de travail d'architecture est complète, si elle répond aux attentes et aux objectifs du projet, et si elle est réalisable dans le cadre du budget et des délais prévus.

Révision : Si des modifications sont nécessaires, la déclaration de travail d'architecture est renvoyée à l'architecte ou à l'équipe chargée de sa rédaction pour révision et mise à jour.

Validation : Une fois que la déclaration de travail d'architecture a été jugée satisfaisante, elle est validée par les parties prenantes du projet, telles que le client, le responsable du projet, l'équipe technique et le service informatique. Cette validation doit être

formelle et documentée.

Signature du contrat : Une fois que la déclaration de travail d'architecture est validée, un contrat est signé entre le client et l'équipe en charge du projet. Ce contrat précise les termes et les conditions de l'exécution de la déclaration de travail d'architecture, y compris les délais, les budgets, les responsabilités et les obligations.

Approbations signées

Date de signature