



# Déclaration de Travail d'Architecture

---

*Projet : XXXX Client : YYYY*

*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

---

Nom du projet	Projet XXX

<i>Préparé par :</i>	
<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>	
<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 <<projet XXX>>.

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

L'entreprise Foosus propose un service rapprochant les consommateurs aux fournisseurs locaux. Jusqu'à maintenant le développement de leur solutions se faisait via plusieurs équipes de développement et chacune était libre d'expérimenter dans son langage de programmation de prédilection afin de garder une innovation rapide et constante.

Le résultat d'un tel développement est que désormais l'architecture de la solution est morcelée en différentes technologies rendant l'ensemble difficile à maintenir et ralentissant le développement de nouvelles innovations.

Foosus a désormais besoin d'une nouvelle architecture permettant de réduire le besoin en maintenance, de limiter l'apparition de problème majeur dans la version déployée entraînant un arrêt complet du système ou un retard de déploiement.

## Description du projet et périmètre

Afin de limiter l'apparition de problème critique l'architecture du système traitant les requêtes client, soit la partie du système faisant l'interface entre le client et les services fonctionnel de l'entreprise, doit être repensé. En effet cette section est construite avec cinq technologies différentes ce qui entraîne fatalement une instabilité du système. De plus cela implique également d'avoir autant d'équipe spécialisée que de technologie, engendrant des coûts supplémentaires et un risque de fuite des connaissances nécessaires au développement du système.

---

## Vue d'ensemble

La section devant être repensé peut être découper en deux groupe distinct, d'une part la solution web et d'autre part l'application mobile. Le fait que ces deux groupe coexiste implique déjà un dédoublement de la charge de travail puisqu'une fonctionnalité développée pour un des deux groupe devra être redéveloppée pour l'autre. Cette séparation ainsi que le nombre important de technologie vont être les deux points d'amélioration ciblé.

## Alignement stratégique

Afin de limiter au mieux les risques d'apparition de problèmes critiques venant perturber la disponibilité du système, le système va être factorisé dans le but d'en réduire la complexité, qui est de loin son plus gros défaut. La nouvelle solution, après discussion avec les équipes de développement, sera construite selon le modèle de micro-service.

## Objectifs et périmètre

### Objectifs

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

<i><b>Objectif Business</b></i>	<i><b>Notes</b></i>
Tirer parti de la géolocalisation	L'objectif de Foosus étant de soutenir l'alimentation locale, la mise en relations des différents acteurs en fonction de leur position géographique est essentiel
Évolutivité vers diverses régions	Pour permettre le déploiement de la solution à travers le monde, elle devra être capable d'accepter facilement l'ajout de fonctionnalité afin de l'adapter aux besoins des régions où elle sera déployé
Utilisable via des appareils fixes et mobiles	Pour plus de disponibilité, la solution doit être accessible sur des appareils fixes et mobiles, cela implique la prise en compte de la bande passante disponible dans

	chacun de ces cas
Centrer l'architecture sur les rôles	La solutions devra proposé diverses fonctionnalité en fonction de l'utilisateur qui s'y connecte, afin de répondre au mieux aux besoins des différents acteurs de la solution

## Périmètre

Le projet de changement d'architecture porte sur les services traitant tout le fonctionnel et le visuel du site web, ainsi que de l'application mobile du produit de Foosus. Les opérations de traitement en dehors de ce cadre ne sont pas concerné par ce projet.

## 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
Ash Callum (CEO)	Prix, technologie employé, Durée de développement	Unification des technologies, factorisation des services, Opensources
Natasha Jarson (CIO)	Technologie employé, structure de la solution, maintenabilité et évolutivité, outil de déploiement	Micro-service, conteneurisation, répartition de la charge
Daniel Anthony (CPO)	Traitements des besoins utilisateur (accessibilité, suivi des besoins)	Factorisation des services, traçabilité des requêtes
Christina Orgega (CMO)	Disponibilité de la solution et capacité de charge (pendant une campagne marketing)	Conteneurisation, répartition de charge, lancement automatique de nouvelles instances

Jo Kumar (CFO)	Budget de la solution	Technologie opensource, unification des technologies, réutilisation de code, agiles du Lean
Pete Parker (Engineering Owner)	Technologie employé, structure de la solution, maintenabilité et évolutivité	Micro-service, conteneurisation, unification des technologies
Jack Harkner (Operations Lead)	Environnement nécessaire au fonctionnement de la solution (serveur, etc)	Conteneurisation, répartition de la charge
Frontend Squad	Interface utilisateur	Unification des technologies, micro-service
Ruby Development Squad	Changement de technologie	Unification des technologies, formation aux nouvelles technologies, micro-service
Identity Management Squad	Système de rôles	Unification des technologies, micro-service
Backend Applications Squad	Fonctionnel gérant les opérations	Unification des technologies, micro-service
Ops Squad	Environnement nécessaire au fonctionnement de la solution (serveur, etc)	Conteneurisation, répartition de la charge, redémarrage de service
Product Managers	Évolutivité fonctionnalité/besoin client	Micro-service
Customer Experience Team	Expérience utilisateur, accessibilité	répartition de la charge, redémarrage de service
Data Analytics	Gestion des données client	Micro-service, séparation de

Team		la base de données
Customer Fullfilement Team	Réponse aux besoins utilisateur, satisfaction client, disponibilité	Redémarrage automatique de service, répartition de la charge
Finance Team	Budget de la solution	Technologie opensource, unification des technologies, réutilisation de code, agiles du Lean

## Approche managériale

Dans ce changement d'architecture, les technologies vont être factorisée, certaines équipe de développement pourrait se retrouver avec des compétences obsolètes pour la nouvelle solution. La gestion Lean ayant été désigné comme au cœur de la ligne de conduite de Foosus, plutôt que se débarrasser des équipes aux compétences obsolètes, des formation visant à les faire monter en compétence dans les nouvelles technologies employé seront mené afin de maintenir une force de frappe identique si ce n'est plus efficace. De plus, les équipes ayant été décrites comme investi dans le produit, seront plus moteur que de nouvelles équipe déjà experte dans les technologies employés mais sans connaissance du produit.

## Procédures de changement de périmètre

Dans le but de migrer de la solution actuelle vers la solution cible de manière fluide, la solution actuelle sera maintenu à son état présent. Cela implique que l'apport de nouvelles fonctionnalités devra être stoppé afin de réduire le risque de défaillance due à leurs intégrations.

Par la suite, une fois que les équipes auront suivi les formations requises, elles pourront commencer à établir les fondations de la nouvelle solution dans son environnement. La migration des services et des bases de données seront ensuite graduellement transférées de l'une à l'autre en veillant à garder une continuité de service durant toute la transition.

Cette continuité pourra être assuré par le déploiement des nouveaux services en parallèles des anciens, redirigeant les nouvelles connexions vers eux, tandis que les anciens assureront la gestion des requêtes antérieur au déploiement. Cela devrait permettre une transition fluide et imperceptible au regard des utilisateurs.

## Rôles et responsabilités

### Structure de gouvernance

---

La structure de la hiérarchie ne changera pas par rapport au tableau des parties prenantes ci-dessus, en revanche les 4 équipes de développement vont, quant à elle, changer du fait de la factorisation des technologies. Il sera pertinent de fusionner les équipes selon leurs capacité puis leur rôles au sein du développement du produit.

## Process du projet

Pour mener ce projet à bien, nous devons nous assurer un processus de développement efficace et limitant les dérives, limitant ainsi la possibilité de dérive. La méthodologie Agile et celle du Lean seront utilisé afin de réussir au mieux cet objectif.

Tout d’abord, les semaines seront ponctué de 1 à 2 réunions de suivi de projet afin de détecter une dérive/un blocage potentiel le plus tôt possible. Ensuite, des relectures de pairs sera mise en place, elle permettra d’élévé l’esprit critique des équipes et de réduire la quantité d’incident au déploiement de version. Cette relecture se verra accompagné de test unitaires afin de s’assurer que chaque fonctionnalité possède bien le comportement désiré sans comportements inattendu. Enfin, dans l’optique de réduire les incidents au lancement d’une solution, un environnement de test reproduisant à l’identique l’environnement de production sera mis à disposition des équipes de développement afin d’augmenter au mieux l’efficacité des tests et de la détection de comportement inadéquat.

## Rôles et responsabilités (RACI)

Éléments de la solution	Ash Callum (CEO)	Natasha Jarson (CIO)	Daniel Anthony (CPO)	Christina Ortega (CMO)	Jo Kumar (CFO)	Pete Parker (Engineering Owner)	Jack Harkner (Operations Lead)	Équipes de développement	Équipe opérationnelle
Unification des technologies	C	A			A	A	I	R	
factorisation des services	C	A				A	I	R	I
Technologie opensource	A	C	I		A	R	R	C	C
conteneuri	C	A				I	A		R



sation									
répartition de charge	C	A	C	I		I	A		R
lancement automatique de nouvelles instances	C	A	C	I		I	A		R
agiles du Lean	A	R	I		A	I	I	I	I
séparation de la base de données	C	A				I	A		R
micro-service	C	A				A	A	R	I
Formation équipes	C	A			A	R	R	R	R

# Approche architecturale

## Méthodologies pertinentes et normes de l'industrie

Ce projet d'architecture sera mené en respectant certaine méthodologie importante en lien avec les besoins et la philosophie de Foosus.

Dans ce cadre le développement futur de la solution devra suivre la méthodologie du Lean visant principalement à améliorer l'efficacité et la qualité du développement de la solution, ainsi que d'en réduire le coût. De ce fait, les solutions opensource et gratuites seront préféré si ces dernières répondent aux besoins exprimés.

La solution traitant certaines données sensibles des clients, notamment les données de géolocalisation et bancaires, devra alors respecter le RGPD afin d'assurer la protection de ces données.

# Plan de travail

---

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

## Définition des conditions requises pour l'architectures et établissement des contrats avec les équipes cibles

---

### Activités

Les conditions spécifiques à cette architecture doivent être décrite dans un document propre afin de garder une trace des contraintes à respecter lors de l'établissement de la solution cible. Par ailleurs, un contrat d'architecture est nécessaire pour les équipes business, ainsi que celles de développement afin de leurs présenter la solution cible et les mesure nécessaire pour atteindre cette dernière.

### Livrables

- Document de Spécification des conditions requises pour l'architecture
- Contrat d'architecture avec les Utilisateurs Business
- Contrat d'architecture avec les Fonctions Développement et Design

## Définition de la stack technologiques

---

### Activités

Une étude sera menée afin déterminé quelles technologies devraient être privilégiés pour mettre en place la solution.

### Livrables

- Document de Définition de la stack technologique

## Restructuration des équipes de développement

---

### Activités

Les équipes de développement devront être réarrangé en fonction de la nouvelle stack technologique. Si une des équipes est déjà formé à ces technologie alors elles sera dispersé dans les futurs équipes afin que les employés ayant la connaissance de la technologie puissent apporté leur aide à ceux qui devront s'y former.

À noté qu'une équipe de maintenance de l'ancien systèmes devra être formé pour prévenir d'une interruption de ce dernier, quand bien même aucune nouvelle fonctionnalité ne sera déployé dessus.

## **Livrables**

- Un plan de restructuration des équipes détaillant la nouvelles constitution des équipes ainsi que le besoin en formations de ces dernières.

## **Définition de la procédure d'assurance qualité**

---

### **Activités**

Conceptualisation des solutions permettant d'éprouver d'une part la qualité des fonctionnalité développé, et d'autre part l'impact de l'intégration de ces dernières dans l'environnement de production.

### **Livrables**

- Document de définition de la procédure d'assurance qualité

## **Définition de l'environnement détaillé de la nouvelle solution**

---

### **Activités**

La nouvelle solution nécessitant la mise en place d'un environnement spécifique usant de technologies avancé de développement opérationnel nécessite un plan détaillé contenant les technologies utilisées, les sécurités mises en places ainsi que le plan des flux réseaux du nouvel environnement.

### **Livrables**

- Document de définition de l'environnement de la nouvelle solution

## **Formations des équipes de développement**

---

## Activités

La stack technologique ayant été factorisé, les employés n'ayant pas les connaissances correspondantes devront être formés. Des formations additionnels seront mené en vue de la mise en place de la procédure d'assurance qualité.

## Livrables

- Relevé des compétence des employés suite aux formations effectuées

## Formation de l'équipe opérationnel

---

### Activités

La nouvelle architecture va nécessité la mise en place de technologie avancé de développement opérationnel, pour cela l'équipe devra suivre les formations nécessaires à l'acquisition des compétence requises lui faisant défaut.

### Livrables

- Relevé des compétence des employés suite aux formations effectuées

## Plan de communication

---

### Évènements

Les événements surviendront à la fin des étapes du plan de travail afin que les éléments produits puissent être approuvés par qui de droit et présenté aux parties prenante concernées.

### Canaux

Les documents seront transmit aux parties prenante concernées par mail, puis déposé dans un répertoire de version en ligne afin de garder une traçabilité des changements effectués aux documents. Pour les parties décisionnaires, des présentations pourront avoir lieu afin de détailler les documents dans le but d'une validation et, le cas échéant, déterminer la liste des points à redéfinir.

### Formats

Les documents seront stocker sous formes de fichier ODT sur le répertoire de version en ligne, ainsi qu'en format PDF, ce qui facilitera leurs distribution aux parties prenantes.

# Contenu

Dans le but de facilité la compréhension des informations transmises par les différent livrables de ce travail d'architecture, lors de la distribution par mail, un avant-propos présentera les grandes lignes des sujets traités dans les documents ci-joints.

# Durée et effort

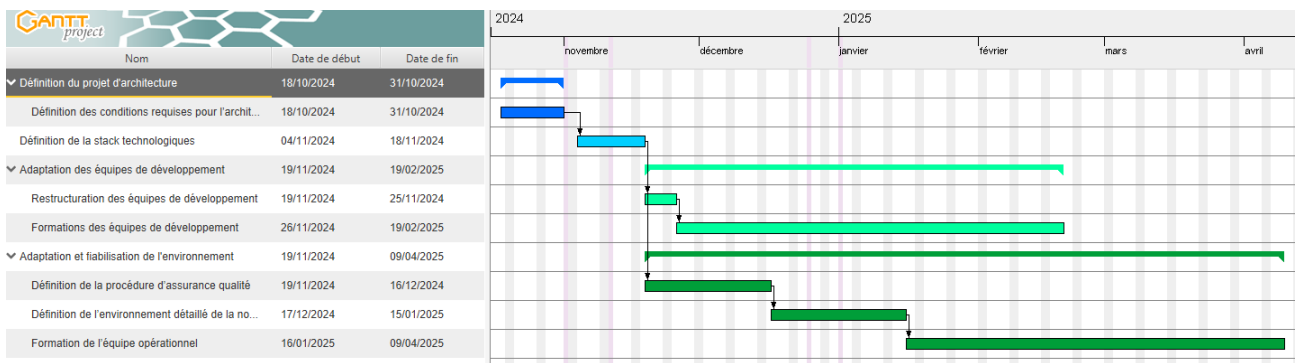
L'élaboration de l'architecture cible ainsi que la préparation du projet de suivi prendront six mois pour arriver à leurs termes. Cette période sera découper par les différents jalons du plan de travail, vous pouvez retrouver une représentation visuel de ces derniers ci-dessous.

# Collaboration

Pour la réalisation de ce projet, plusieurs phases et livrables nécessiteront des échanges avec les parties prenantes concerné, afin de s'assurer de la cohérence de la solution avec les besoins énoncés en amont, ainsi que de la compréhension et acceptation de la nouvelle solution par les parties prenantes.

# Plan et calendrier du projet

Voici un diagramme de Gantt détaillant le calendrier de ce projet d'architecture en fonction des jalons défini dans le plan de travail :



# Risques et facteurs de réduction

## Analyse des risques

ID	Risque	Gravité	Probabilité	Facteur de réduction	Propriétaire
----	--------	---------	-------------	----------------------	--------------

1.	Allongement de la durée de formation d'une équipe de développement	Fort	Probable	Management du Lean	Pete Parker (Engineering Owner)
2.	Perte de données lors de la migration	Fort	Peu probable	Anticipation des séquence de transfert de données et simulation dans un environnement de test	Jack Harkner (Operations Lead)

## Hypothèses

---

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

ID	Hypothèse	Impact	Propriétaire
1.	Une équipe obsolète formé à de nouvel compétence sera plus efficace qu'une nouvelle équipe due à l'implication dans le produit des équipes actuelles	Durée formation, efficacité des équipes	Natasha Jarson (CIO)

## Critères d'acceptation et procédures

### Métriques et KPIs

---

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

<b>Métrique</b>	<b>Technique de mesure</b>	<b>Valeur cible</b>	<b>Justification</b>
Nombre d'adhésions utilisateur par jour	Étude de la base de donnée utilisateur afin de détecter les nouveaux arrivants	+10 %	L'un des objectifs de Foosus est de développer la croissance de l'entreprise, pour cela, traquer la quantité d'adhésion permet de se rendre compte de l'attractivité de la solution
Adhésion de producteurs alimentaires	Étude de la base de donnée producteur afin de détecter les nouveaux arrivants	De 1,4 par mois à 4 par mois	Dans le but de croître et d'attirer de nouveaux consommateurs, Foosus a besoin de nouveaux producteurs afin de diversifier les offres de proximité de la solutions, aussi l'attractivité des producteurs doit être mesuré
Délai moyen de parution	Une mesure du temps de développement de chaque fonctionnalité devra être mis en place dans la gestion du projet	De 3,5 semaines à <1 semaines	Pour rétablir la dynamique d'innovation recherché par l'entreprise, le délai de parution de fonctionnalité doit être réduit au minimum
Taux d'incidents de production P1	Une étude constante des rapports d'incidents permettra de mesurer leur fréquence d'apparition	De >25/mois à <1/mois	Réduire le taux d'incident permet d'augmenter la confiance des utilisateurs envers la solution, que ce soit les producteurs ou les consommateurs. Ce facteur est essentiel puisqu'il va influencer sur l'attractivité des utilisateurs, il permettra également d'augmenter la production de fonctionnalité puisque moins de ressources seront alloué à la résolutions

			des incidents
--	--	--	---------------

## Procédure d'acceptation

---

Le projet d'architecture devra être validé par chaque chef de section, en tenant informé les équipes correspondantes, avant d'être validé par le CEO. Pour ce faire un contrat business et un contrat techniques seront fournis afin de présenter le projet de la meilleure manière aux parties intéressés, selon les éléments importants pour chacun.

## Approbations signées

---

Date de signature