



19 sept. 2022

DECLARATION DE TRAVAIL D'ARCHITECTURE

Projet Foosus géoconscient
Version 1.0

Auteur : Hervé Prevost - Architecte Logiciel - Foosus

Date version : 19/09/2022

Version : 1.0

SOMMAIRE

SOMMAIRE	2
OBJET DU DOCUMENT	4
DECLARATION DE TRAVAIL D'ARCHITECTURE	4
Requête du projet et contexte	4
Description du projet et périmètre	5
Nouvelle fonctionnalité : géolocalisation	5
Evolutivité	6
Disponibilité	6
Simplicité et Spécificité	7
Vue d'ensemble	7
Vision générale	7
Ecueils à éviter	8
Équipe développement et technologies	8
Alignement stratégique	11
OBJECTIFS ET PÉRIMÈTRE	11
Objectifs	11
Disponibilité	11
Evolutivité	12
Périmètre	12
Parties prenantes, préoccupations, et visions	12
Approche managériale	13
RÔLES ET RESPONSABILITÉS	14
Structure de gouvernance	14
Process du projet	14
Archives documentaires	14
Comités de pilotage (DI, équipe développement, Product Owner)	14
Gestion de la configuration (CIO, DO)	14
Assurance qualité (CPO, Architecte, CMO, DO)	15
Procédure en cas d'escalade	17
Procédure en cas de changement	18
Rôles et responsabilités (RACI)	19
APPROCHE ARCHITECTURALE	20
Process d'architecture	20
Mise à l'échelle	20
Cycle ADM	21

CONTENU DE L'ARCHITECTURE	30
PLAN DE TRAVAIL	31
Élément de travail 1	31
Activités	31
Livrables	31
Élément de travail 2	31
Activités	31
Livrables	31
Élément de travail 3	32
Activités	32
Livrables	32
Plan de communication	32
Durée et effort	33
Collaboration	34
Plan et calendrier du projet	34
RISQUES ET FACTEURS DE RÉDUCTION	35
Analyse des risques	35
Hypothèses	37
CRITÈRES D'ACCEPTATION ET PROCÉDURES	38
Métriques ou KPIs (indicateurs clés de performance)	38
APPROBATIONS SIGNÉES	39

OBJET DU DOCUMENT

Ce document est une *déclaration de travail d'architecture* pour la mise à niveau de la plateforme e-commerce de l'entreprise.

Ce document définit le périmètre et la méthode qui seront utilisés pour mener à bien ce projet d'architecture. Il permet d'en mesurer la réussite et forme la base contractuelle entre le fournisseur et les consommateurs de services d'architecture.

Le document est construit selon le cadre d'architecture TOGAF. Il correspond à la **phase A du cycle ADM** (Architecture Development Method) c'est-à-dire la Vision d'Architecture. Cette étape comporte les éléments suivants :

- Énoncé clair du travail d'architecture demandé
- Description de la liste des travaux à effectuer pour atteindre la nouvelle architecture
- Définition de l'architecture Actuelle et de l'architecture cible
 - Description des grands atouts de la nouvelle architecture, ses nouvelles possibilités et capacités, la valeur commerciale
 - Présentation des nouveaux travaux aux parties prenantes (stakeholder)
 - Construction d'un consensus avec les parties prenantes et obtention de la signature commune d'un document de démarrage des travaux
- Identification des parties prenantes, leur souhaits et contraintes
- Confirmation des objectifs énoncés dans la phase préliminaire
- Evaluation des capacités et ressources de l'entreprise pour mener à bien ce projet et sa préparation au changement
- Evaluation des risques liés à la transformation

DECLARATION DE TRAVAIL D'ARCHITECTURE

Requête du projet et contexte

L'objectif de Foosus est de soutenir l'alimentation locale et de mettre les consommateurs en contact avec des producteurs et des artisans locaux.

L'entreprise veut construire une solution géociblée s'appuyant sur une nouvelle architecture.

Le succès de Foosus est grandement lié à sa capacité d'innovation logicielle mais l'entreprise a accumulé une dette technique importante qui impacte aujourd'hui ses possibilités d'évolution.

Par ailleurs, le système actuel n'est pas dimensionné pour faire face à l'augmentation du nombre d'abonnés en cas de succès des développements commerciaux envisagés (1 million d'utilisateurs).

Il devient impératif de disposer d'un système dimensionné pour le marché et capable d'évoluer plus rapidement.

Description du projet et périmètre

Selon les mots de notre CEO, il s'agit de *“Créer un plan pour nous amener à un état cible d'architecture, qui réponde aux besoins et aux capacités du business et des équipes de Direction”*.

L'entreprise souhaite concurrencer les ténors de l'alimentation durable. Le projet consistera à définir l'organisation technique de la plateforme et de son équipe de développement permettant d'atteindre cet objectif.

La **géolocalisation** est la seule fonctionnalité nouvelle à développer dans l'immédiat. Les autres améliorations nécessaires sont d'ordre plus général et concernent en priorité **l'évolutivité et la disponibilité** de la plateforme.

Nouvelle fonctionnalité : géolocalisation

La nouvelle plateforme doit *“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.”*

Voici un descriptif plus détaillé des autres besoins et contraintes classés selon leur nature :

Evolutivité

- L'architecture devra être évolutive pour permettre à nos services de se déployer sur diverses régions à travers des villes et des pays donnés.
- Les livrables doivent pouvoir être fournis à intervalles réguliers pour que le nouveau système soit rapidement opérationnel et puisse être doté de nouvelles fonctionnalités au fil du temps.
- Possibilité de réorienter des solutions existantes, en expérimentant de nouvelles modifications et en facilitant l'intégration avec des partenaires internes et externes.
- Nous devons combler le fossé entre le moment où une ligne de code est écrite et celui où elle est validée dans un environnement intégré. Cela peut également nous aider à déterminer les réactions de nos clients vis-à-vis de nouvelles fonctionnalités à mesure que nous développons ces dernières.
- Nos équipes produits aimeraient pouvoir exécuter diverses variantes ou réaliser des comparaisons de différentes solutions auprès de nos utilisateurs. Pour y parvenir, nous avons besoin de visibilité sur la façon dont nos logiciels sont utilisés et nous devons pouvoir inverser des décisions d'architecture tant que cela reste peu onéreux. Ou alors répliquer sur une plateforme qui nous permette d'essayer de nouveaux produits d'une façon compatible avec nos objectifs commerciaux fondamentaux.

Disponibilité

- 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.
- Les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.
- Choisir une architecture permettant de limiter les risques techniques.

- Même si le système est surchargé, les utilisateurs connectés doivent pouvoir continuer à accéder à tous les services de façon dégradée.
- Chaque nouvelle version doit être de taille réduite, présenter peu de risques, être transparente pour nos utilisateurs et rester accessible en tout lieu et à tout moment.

Simplicité et Spécificité

- 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.
- Offrir une expérience utilisateur de premier plan (simplicité, UX Design).

Vue d'ensemble

Vision générale

L'architecture proposée pour le système cible est une architecture hybride afin de répondre aux objectifs principaux à priori contradictoires : capacité de mutation rapide et stabilité.

En tant que plateforme web, le système est naturellement apparenté à l'architecture **client-serveur**. Pour accélérer le développement de la partie Back-Office du système, qui constitue comme souvent, la partie la plus importante du travail à réaliser, nous allons également nous appuyer sur des principes d'**architecture en couche et orientée données**.

L'architecture en couche permet de standardiser les interfaces de saisie et industrialiser les opérations de gestion de données. Elle limite les tests nécessaires aux tâches génériques et permet de disposer d'un système rapidement fonctionnel et évolutif.

L'architecture orientée données permet de gérer les informations de structure (tables, formats de saisie, droits utilisateurs, procédures...) de la même manière que les autres données ordinaires. Un applicatif noyau côté

serveur et client se chargeant de gérer l'interface et la mise à jour simplement à partir de ces paramètres stockées en base de données.

Enfin, pour les opérations en front qui nécessitent des interfaces utilisateurs plus spécifiques ou qui présenteraient une chaîne d'accès complexe aux données nous utiliserons des **microservices**.

Bien que cette dernière architecture soit moins pertinente en B2C, elle offrira plus de souplesse à terme pour tester les fonctionnalités sans remettre en cause le noyau central du système.

L'architecture microservices est présentée comme la plus souple de toutes puisqu'elle permet d'encapsuler les développements et les données nécessaires sans se préoccuper des langages employés pour leur mise en œuvre. Le revers de la médaille est que l'orchestration d'un tel système (multiples rouages et modules autonomes) est beaucoup plus complexe qu'une application monolithique. Elle nécessite des compétences rares.

Ecueils à éviter

Un autre élément à l'origine de bien des erreurs de conception de produits ou services est d'oublier que 1) la clientèle ne peut disposer du même niveau d'information sur ces produits que le personnel interne à l'entreprise, 2) que cette clientèle est déjà noyée dans l'utilisation quotidienne d'une myriade d'applications et de services. Dans ce contexte proposer des services complexes avec un niveau élevé de raffinement pour attirer le consommateur a souvent l'effet inverse de celui escompté.

L'architecture doit donc viser la fidélisation du consommateur avant tout par la simplicité et la fiabilité des services qui lui sont fournis.

Équipe développement et technologies

Les équipes de développement ont elles-mêmes demandé de limiter les technologies utilisées pour coder les applications. Les équipes disposent d'un socle technique commun à savoir le langage Java, mais le rythme

important des développements souhaités par la direction a conduit à une multiplication des technologies.

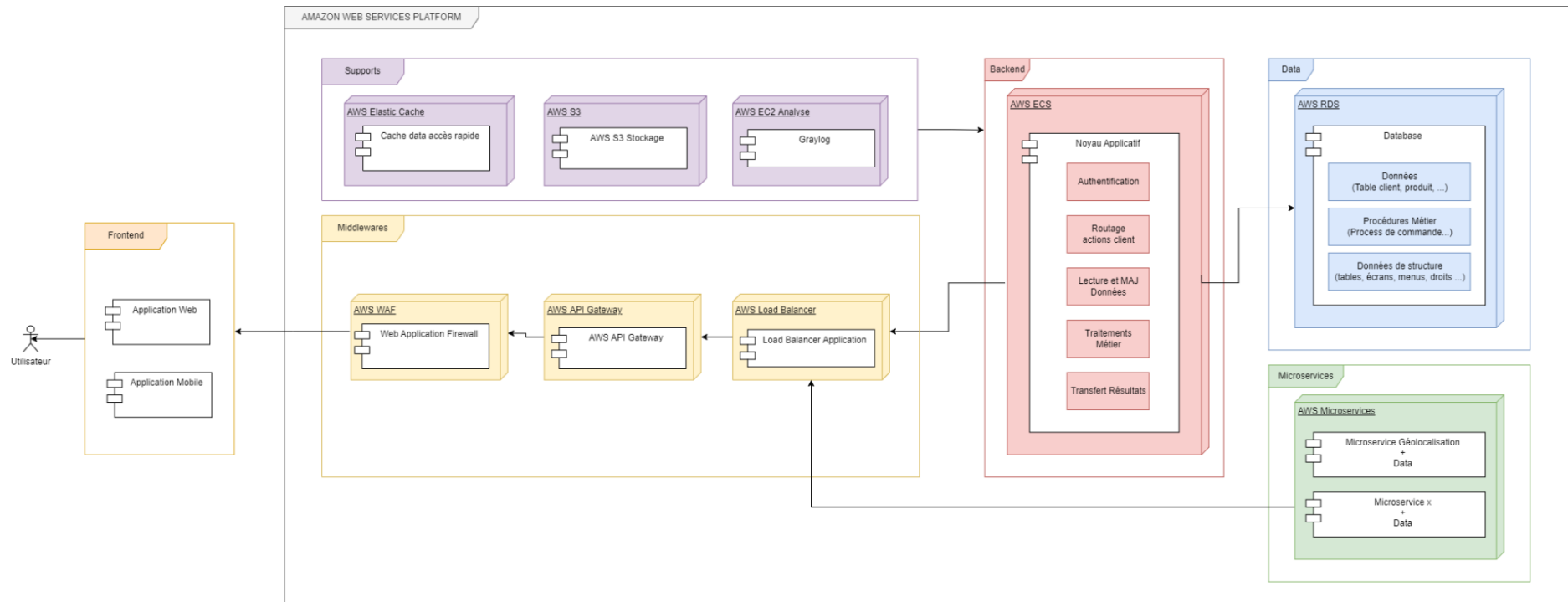
La disponibilité d'un système est fonction de son maillon le plus faible. Il est donc indispensable d'harmoniser ces techniques et de rendre chaque développeur opérant sur n'importe quel projet.

La spécialisation entre les compétences backend et front a également été remise en cause récemment. Il est en effet nécessaire que chaque développeur ait une connaissance globale des environnements afin d'orienter efficacement ses développements.

Pour toutes ces raisons, je préconise de migrer les équipes vers le langage **Javascript**. Celui-ci doit de toute façon être maîtrisé pour le développement Web. Il se rapproche du Java dans sa syntaxe (surtout si l'on y ajoute le typage avec **Typescript**). **Express Js** adossé à **Node JS** côté serveur permet de profiter des performances du développement asynchrone en annulant la complexité inhérente à ce type de développement (callback hell). S'agissant des possibilités de traitement asynchrone, **RXJava** était une alternative à NodeJS mais il ne règle pas le problème de l'hétérogénéité des langages (front/back).

La nouvelle architecture préconisée (en couche/orientée données) nous aurait permis d'économiser l'apprentissage d'un framework client s'il n'avait pas été nécessaire de développer une version mobile de l'application. A ce titre **REACT** semble s'imposer largement (avec React Native notamment) par rapport à Angular, ce dernier aurait dû être abandonné de toute façon à plus ou moins long terme.

Dans le Schéma ci-dessous la partie back-office est représentée par les blocs intitulés Backend et Data. Les autres modules de la plateforme disposant d'un fonctionnement autonome sont représentés dans le bloc Microservices. Selon ce principe d'architecture les composants encapsulent le code et les données (d'où la mention + data figurant dans chacun d'eux).



Alignement stratégique

La raison d'être de Foosus est de soutenir l'alimentation locale et de mettre les consommateurs en contact avec des producteurs et des artisans locaux.

Il lui manque pour cela des capacités de géolocalisation pour faciliter la mise en relation de consommateurs avec des fournisseurs d'une même zone géographique.

D'autre part, Foosus souhaite recouvrer une part importante des clients qu'elle a perdu du fait de l'indisponibilité récurrente du système et du manque d'attractivité de sa plateforme.

Idéalement, elle ambitionne de concurrencer les ténors de l'alimentation durable en atteignant le million d'utilisateurs.

Elle doit donc à la fois fiabiliser sa plateforme et la rendre plus rapidement évolutive.

OBJECTIFS ET PÉRIMÈTRE

Objectifs

Les objectifs business de ce travail d'architecture peuvent être résumés ainsi :

Disponibilité

- Garantir le fonctionnement nominal de la plateforme à toute heure, quels que soient la zone géographique de l'utilisateur, son rôle (client, fournisseur, collaborateur) ou son dispositif de consultation.
- Mettre en place un environnement de test et validation pour éviter toute régression lors de la mise en production
- Mise en production progressive des nouvelles fonctionnalités afin de limiter leur impact et localiser plus facilement les sources d'erreur.

Evolutivité

Augmenter le rythme des évolutions du système en prenant plusieurs mesures :

- Limiter et uniformiser les technologies de développement utilisées pour simplifier les développements
- Mise en place de fonctionnalités modulaires permettant d'en mesurer l'attrait auprès du public avant généralisation
- Possibilité de les retirer sans incidence sur le fonctionnement global du système (réversibilité).

Périmètre

Le périmètre du projet est la définition de l'architecture cible de la nouvelle plateforme, elle comprend le back office, la partie front de l'application mobile et l'application web ainsi que les composants backend.

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 Chief Executive Officer (CEO)	Rentabilité, Alignement à la stratégie de l'entreprise	Métier
Natasha JARON Chief Information Officer (CIO)	Sécurité, agilité et efficacité du système d'information	Logicielle, Infrastructure, Opérationnelle

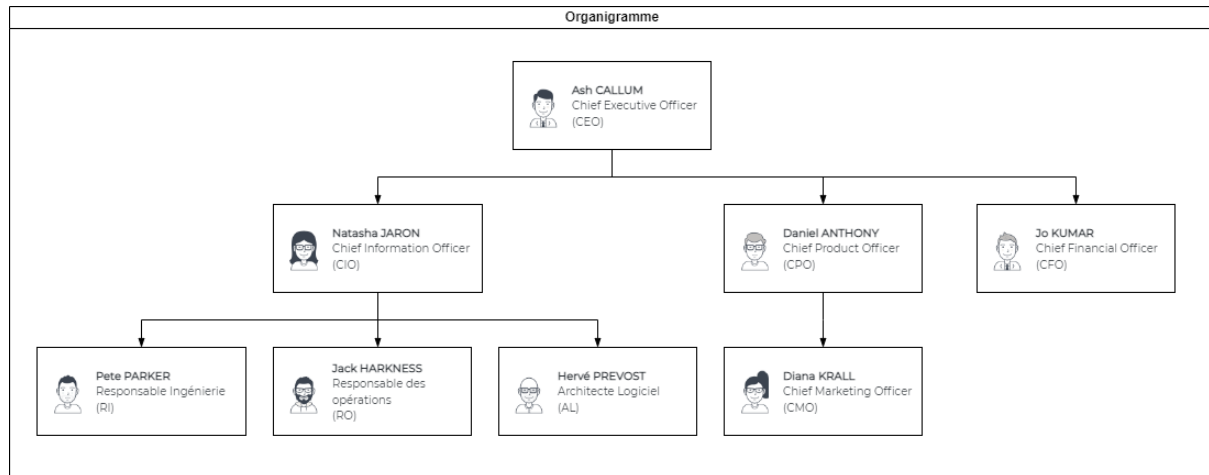
Daniel ANTHONY Chief Product Officer (CPO)	Gestion produit, stratégie, UX	Métier
Diana KRALL Chief Marketing Officer (CMO)	Marketing, stats et analyse ventes	Marketing
Jo KUMAR Chief Financial Officer (CFO)	Rentabilité, budgets, coûts	Financière
Jack HARKNESS Responsable des opérations (RO)	Performances, supervision et statistiques de la plateforme	Opérationnelle
Pete PARKER Responsable Ingénierie (RI)	Formation et management des équipes de développement, choix des technologies et méthodes	Logicielle
Hervé PREVOST Architecte Logiciel (AL)	Gouvernance architecture, Développement et évolutions	Logicielle, architecturale

Approche managériale

L'organisation suit les principes Agile. Nous recommandons d'utiliser SCRUM plutôt que KANBAN. SCRUM fonctionne avec des sprints qui permettent une meilleure définition des rôles au sein des équipes et impose des contraintes de temps, deux notions nécessaires pour améliorer l'organisation et l'efficacité des développements.

RÔLES ET RESPONSABILITÉS

Structure de gouvernance



Process du projet

Archives documentaires

Le dépôt créé pour abriter les documents d'architecture est :

<https://github.com/HPrevost/OCProjet5>

Comités de pilotage (DI, équipe développement, Product Owner)

- Suivi avancement de projet et des ressources (Hebdomadaire)
- Sprint planning + Sprint Review

Gestion de la configuration (CIO, DO)

La gestion des configurations est un processus qui permet de maintenir les systèmes informatiques, les serveurs et les logiciels dans l'état souhaité et d'en préserver la cohérence. C'est une façon de s'assurer qu'un système fonctionne comme prévu au fil des changements effectués.

Une revue hebdomadaire doit permettre de faire un état des lieux du fonctionnement des systèmes, des problèmes rencontrés, des modalités de résolutions et des ressources nécessaires.

Assurance qualité (CPO, Architecte, CMO, DO)

La qualité, c'est la capacité pour une entreprise à satisfaire les besoins exprimés, latents et implicites du client, ainsi que ceux des autres parties intéressées.

Les enjeux de la qualité sont :

- La satisfaction du client qui augmentent la notoriété de l'entreprise et la perspective de nouveaux contrats
- La confiance client c'est ce précisément ce qu'on appelle l'**assurance qualité**
- La différenciation par rapport à la concurrence qui s'obtient par la spécificité ou les performances des services proposés. Ces éléments permettent de pratiquer des prix plus élevés ou de profiter d'un segment de marché plus important.
- La réduction des coûts liés aux défauts et dysfonctionnements des processus de l'entreprise

Les indicateurs de mesure de la qualité définis par la direction sont résumés dans le tableau suivant :

Indicateur	Changement souhaité
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

Les enquêtes de satisfaction (avis utilisateurs) et les remontées concernant l'expérience (UX) faites au support technique sont également de bons moyens pour ajuster et améliorer la qualité du produit.

En dernier lieu nous pouvons citer le contrôle croisé des développements (pair review) et l'organisation de sessions régulières et systématiques de test pour chaque nouvelle correction ou évolution.

Procédure en cas d'escalade



Procédure en cas de changement



Rôles et responsabilités (RACI)

La matrice RACI est utilisée afin de déterminer et visualiser les rôles de chacun dans un projet.

Tâche	CEO	CIO	CPO	CMO	CFO	RO	RI	AL
Définition des objectifs	A	R	C	C	C	C	C	C
Gestion des risques		A				C		R
Définition architecture	I	A	C	I	I	C	C	R
Gestion des changements	I	A	A	C	A	C	I	R
Gestion Projet		A	C			C	R	C
Livraisons		A	I	I	I	I	R	I
Réalisation des Développement (*)		I					A	I

Lexique des rôles : (R)esponsable, (A)pprobateur, (C)onsulté, (I)nformé

(*) Les développeurs sont chargés de la réalisation, ils ne sont pas indiqués sur la matrice RACI.

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. Le tableau ci-dessous décrit l'utilisation de l'ADM pour ce projet spécifique. La phase préliminaire est représentée en fond gris car elle est déjà accomplie. C'est le cas également des phases G et H qui sont prématurées à ce stade de définition du projet (les phases B, C, D n'étant pas définies).

Mise à l'échelle

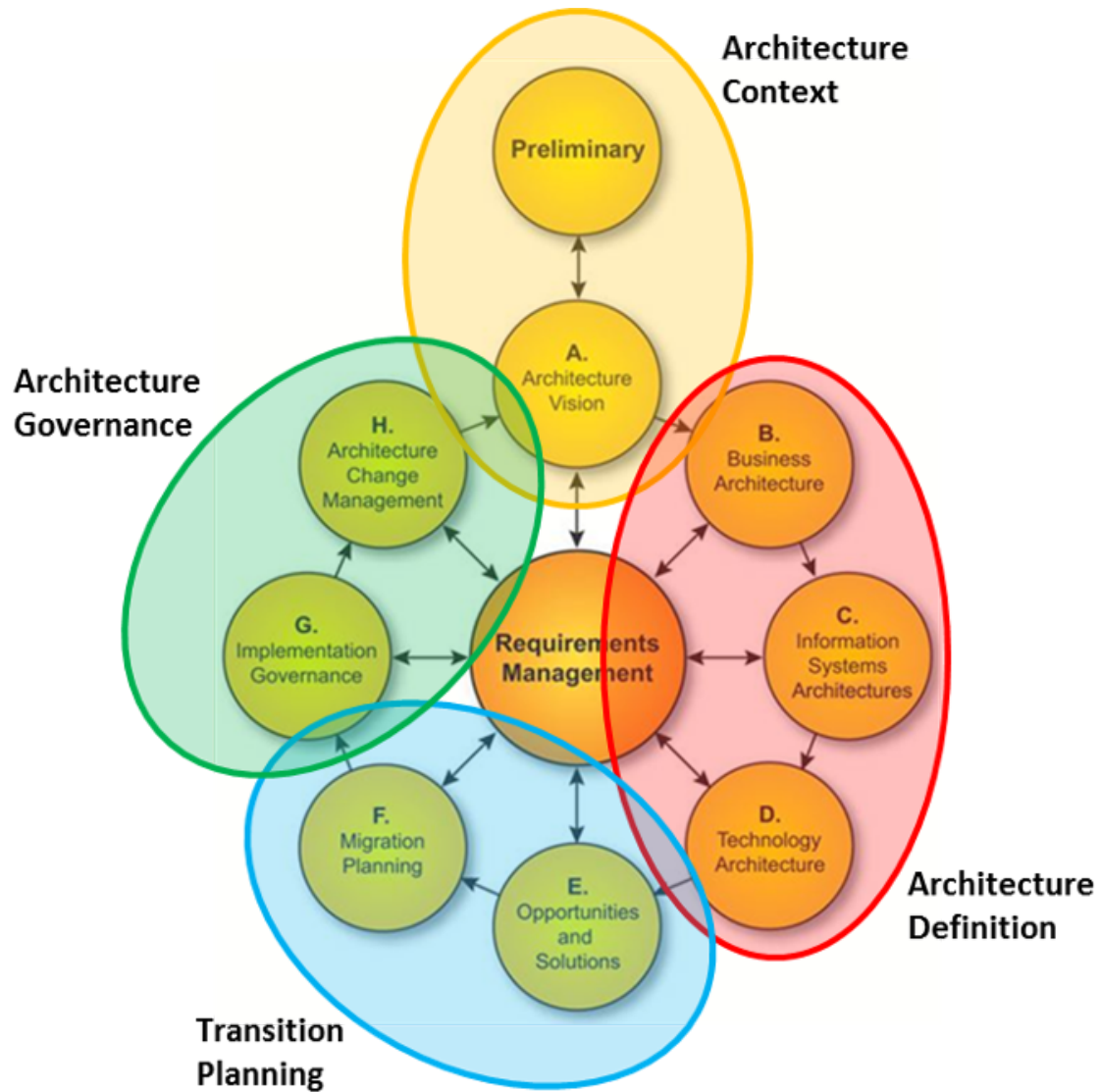
Comme nous l'avons vu précédemment, le présent document s'occupe de la phase A du cycle.

Les phases B, C et D étant essentielles à tout projet elles seront donc également utilisées.

Les phases E et F de la partie *Transition Planning* seront nécessaires puisque l'architecture cible présente des changements conséquents par rapport à l'architecture actuelle. Comme les différentes parties prenantes l'ont demandé, nous devons faire en sorte de maintenir la plateforme dans un fonctionnement nominal durant la migration et éviter les interruptions de service.

Les phases de la *gouvernance d'architecture* ne seront pas utilisées car prématurées à ce stade du projet.

Cycle ADM



Phase	Composantes	Détail
DÉFINITION DES BESOINS		
Préliminaire	<p>Demandes ou requête d'architecture</p> <p>Constitution équipe architecture</p> <p>Autorisation de projet</p>	Phase Terminée

<p>A —Vision de l'architecture</p>		
	Travaux	Énoncé clair du travail d'architecture demandé, décrit la liste des travaux à effectuer pour atteindre la nouvelle architecture
	Architecture cible	<p>Décrit les grands atouts qu'apportera la nouvelle architecture, ses nouvelles possibilités et capacités, la valeur commerciale</p> <p>Vend les nouveaux travaux aux parties prenantes (skateholder) Construit un consensus avec les parties prenantes et obtient la signature commune d'un document de démarrage des travaux</p>
	Parties prenantes	Identifie les parties prenantes, leur souhaits et contraintes
	Objectifs et alignement	Confirme les objectifs énoncés dans la phase préliminaire
	Capacités et ressources	Évalue les capacités et ressources de l'entreprise pour mener à bien ce projet et sa préparation au changement
	Gestion des risques	Évalue les risques liés à la transformation

ARCHITECTURE FUTURE		
B —Architecture business		<p>Définit l'architecture métier (fonctionnelle) cible : quels sont les composants à implémenter pour passer de l'architecture actuelle à la cible.</p> <p>Prérequis aux étapes d'architecture données, applicative, technologique.</p> <p>Les phases B C D (Métier, Systèmes, Technologie) appartiennent au développement d'architecture et suivent les mêmes étapes.</p> <p>En schématisant sur ces différents plans l'architecture actuelle et cible on peut mieux définir les étapes à réaliser pour parvenir à la migration</p>
	Atouts et rentabilité	Démontre aux parties prenantes le ROI et la valeur commerciale du développement
	Processus métier	Utilise les modèles suivant : activité, processus, cas d'utilisation pour décrire l'archi métier cible
C — Architecture des systèmes d'information		Décrit le SI sur un plan données et applicatif. Pour chaque partie on suit les étapes suivantes:

	<p>Étapes du processus</p>	<p>Définir le point de vue, les modèles et outils</p> <p>Analyser l'architecture actuelle</p> <p>Analyser l'architecture cible</p> <p>Définir l'écart entre les deux</p> <p>Définir les étapes de développement pour les composants candidats</p> <p>Résoudre les impacts potentiels quant à cette migration</p> <p>Le DDA (Document Definition Architecture) est ainsi établi</p>
	<p>Création des modèles</p>	<p>Données</p> <p>Entité-Relations</p> <p>Diagrammes de classe</p> <p>Applications</p> <p>Diagramme de communication d'application</p>

<p>D — Architecture technologique</p>	<p>Références</p> <p>Comparatifs</p>	<p>C'est la description de la structure et des interactions entre les services des plateformes, et les composants logiques et physiques</p> <p>On s'appuie sur les modèles de référence technologique</p> <p>On établit des comparatifs selon plusieurs critères : performance, maintenabilité, latence, disponibilité, localisation</p>
<p>LIVRAISON DE L'ARCHITECTURE</p>		
<p>E —Opportunités et solutions</p>	<p>Faisabilité technique et opérationnelle</p> <p>Roadmap</p>	<p>Définition d'une solution spécifique</p> <p>Lots de travaux et timeline</p>
<p>F —Planning de migration</p>		

	<p>Plan détaillé de migration</p> <p>4 cadres de travail impliqués</p> <p>Priorisation des tâches avec 3 critères d'évaluation</p> <p>Définition des facteurs de succès critiques</p> <p>Mesure d'efficacité</p>	<p>Définir les états transitoires entre l'architecture actuelle et la cible et vérifier la conformité à chaque étape</p> <p>Business planning</p> <p>Enterprise Architecture</p> <p>Portfolio and project management</p> <p>Operations Management</p> <p>Evaluation de performances</p> <p>Retour sur investissement</p> <p>Valeur commerciale</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	Conformité aux objectifs stratégiques	
GESTION DES CHANGEMENTS		
G —Gouvernance de l'implémentation	<p>Vérifier le périmètre et les priorités de développement</p> <p>Guider le développement et le déploiement de la solution</p> <p>Faire des revues régulières pour vérifier la compatibilité</p>	Vérifier que l'implémentation est en accord avec l'architecture cible
H —Management du changement d'architecture	Définir les procédures pour gérer le changement d'architecture. Consiste à monitorer :	

	<p>Les demandes de gouvernance</p> <p>Vérifie sur une demande correspond à une mise à jour de l'architecture ou bien nécessite un nouveau cycle ADM</p> <p>Vérifie si le changement représente une valeur économique</p> <p>Classification des demandes</p> <ul style="list-style-type: none"> Simplification (réduction de l'investissement) Incremental change (valeur ajoutée tirée d'un investissement) Re-Architecting : Accroître investissement et la valeur ajoutée <p>Les nouvelles technologies</p> <p>Les changements de l'environnement métier</p>	
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

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
A - Principes, Vision, et Conditions requises de l'Architecture	- Vision d'architecture
B - Architecture Business	- Dictionnaire métier - Acteurs et rôles - Diagrammes de cas d'utilisation métier - Diagramme de séquences
C - Architecture des SI — Données	- Diagramme Entités/Relation
C - Architecture des SI — Applications	- Diagramme de localisation des applications et utilisateurs - Diagramme de communication inter-applications - Diagramme de migration applicative
D - Architecture technologique	- Composants logiciels - Infrastructure matérielle et réseau - Interaction composants logiques et physiques
F - Réalisation de l'architecture	- Contrats d'architecture

PLAN DE TRAVAIL

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

Élément de travail 1

Activités

Demande de chantier d'architecture, la phase préliminaire et la phase A de l'ADM

Livrables

- Autorisation de projet
- Périmètre couvert
- Parties prenantes
- Vision macro
- Roadmap
- Risques

Élément de travail 2

Activités

Livraison de l'architecture, les phases B, C, D de l'ADM.

Livrables

- Objectifs stratégiques et opérationnels
- Fonctions et services métier
- Processus métier
- Lexique

- Architecture des données
- Architecture applicative

- Architecture technique
 - Composants logiciels
 - Infrastructures

Élément de travail 3

Activités

Planification de la transition, les phases E et F de l'ADM

Livrables

Faisabilité technique et organisationnelle

- Contraintes d'intégration
- Planning de migration
- Constitution des projets et mise en œuvre
- Organisation

Plan de communication

Voici les plannings de communication pour l'organisation du chantier d'évolution de l'architecture

Évènement	Participants	Contenu	Fréquence
Réunion quotidienne	Équipes projets / Product Owner	Suivi de l'avancement, des difficultés et des opportunités	Quotidien
Sprint planning	Équipes projets / Product Owner/ Utilisateur réfèrent	Définition des lots de travaux pour le sprint à venir	Bimensuel

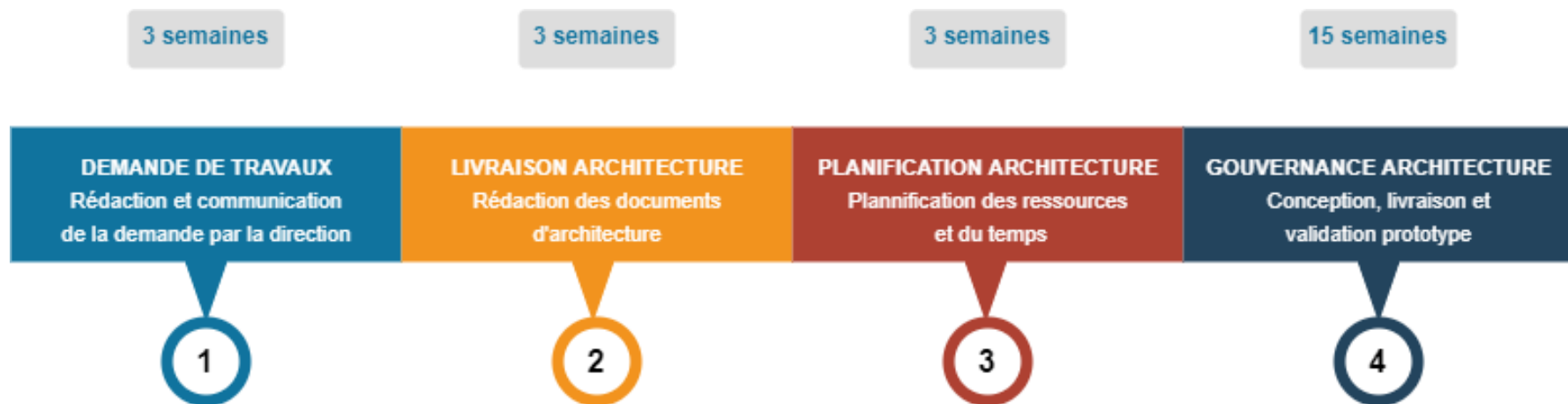
Sprint Review	Équipes projets / Product Owner/ Utilisateur référent	Présentation / Validation des lots de travaux terminés	Bimensuel
Comité de pilotage	Responsables projets/PO	Suivi de la planification et des ressources projet	Hebdomadaire
Comité opérationnel	Responsables d'unités / PO	Suivi des indicateurs qualités et validation expérience utilisateur	Bimensuel
Comité architecture	Equipe architecture	Gouvernance d'architecture	Bimensuel

Durée et effort

Le projet est approuvé à hauteur de 50 000 \$ sur une période de 6 mois pour définir l'architecture et préparer un projet de développement d'un prototype.

Collaboration

Plan et calendrier du projet



RISQUES ET FACTEURS DE RÉDUCTION

Analyse des risques

Risque	Effet	Fréquence	Impact	Criticité	Atténuation
Serveur d'une région indisponible	Indisponibilité de service, augmentation de la latence	1	4	4	Permettre d'utiliser les ressources du serveur le plus proche le temps de l'indisponibilité.
Indisponibilité des services suite à une mise en production	Perte de crédibilité, indisponibilité de service	2	5	10	Réduire les unités déployées au strict nécessaire et augmenter la fréquence des déploiements pour une meilleur progressivité. Faire des déploiements région par région.
Vol de données	Conséquences juridiques	1	5	5	Adaptation à la législation locale

Capacité de monter en charge sous-évaluée	Interruption service	3	5	15	<p>Évaluer en amont les capacités théoriques de connexion simultanée nécessaires.</p> <p>Supervision du trafic et des métriques stockage, réseau et processeurs.</p> <p>Configuration d'un cluster permettant l'adaptation dynamique à la montée en charge.</p> <p>Coder la plateforme pour réduction du poids des images, compression, optimisation des requêtes et adaptations d'affichage selon débit du dispositif</p>
Risque technique / Compétence Cloud	Architecture non adaptée aux besoins et exigences	4	5	20	Recruter un spécialiste services cloud

Hypothèses

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

ID	Hypothèse	Solution
H1	La plateforme existante est maintenue. Aucune nouvelle fonctionnalité n'y sera développée.	Un nouveau SI sera déployé et supportera uniquement les nouvelles fonctionnalités.
H2	La nouvelle architecture sera construite avec les technologies actuelles et pourra s'adapter à de nouvelles technologies lorsque celles-ci seront disponibles.	Développement selon le modèle de l'Architecture micro-services.
H3	La plateforme et les technologies actuelles sont conservées, les dirigeants devant adapter leurs demandes aux possibilités du système.	Les nouvelles fonctionnalités devront être adaptées pour s'intégrer dans la plateforme actuelle.
H4	<p>Redéveloppement complet de la plateforme, uniformisation des technologies employées et des savoirs-faire des équipes.</p> <p>Formation des équipes pour favoriser l'évolution des développeurs vers des profils full-stack interchangeables.</p>	<p>Redéveloppement complet de la plateforme avec un langage unique côté serveur et client (Node/JS) et selon des méthodes proches des compétences actuelles des équipes (Typescript, REACT..).</p> <p>Mise en place d'une architecture hybride prenant le meilleur de chaque modèle (en couche / orientée données / client-serveur) pour la partie Back-office.</p> <p>Architecture micro-services uniquement sur les fonctionnalités spécifiques ne pouvant être prises en charge par le noyau applicatif standardisé.</p>

CRITÈRES D'ACCEPTATION ET PROCÉDURES

Métriques ou KPIs (indicateurs clés de performance)

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

Id	Métrique	Mesure	Valeur initiale	Valeur cible
KPI1	Nombre d'adhésions d'utilisateurs par jour	Surveillance du nombre de nouveaux utilisateurs		Augmentation de 10 %
KPI2	Adhésion de producteurs alimentaires	Surveillance du nombre de nouveaux producteurs.	1,4/mois	> 4/mois
KPI3	Délai moyen de parution	Audit régulier du délai de parution d'une offre	3,5 semaines	< 1 semaine
KPI4	Taux d'incidents de production P1	Liste des incidents visibles par le client	> 25 / mois	< 1 / mois

APPROBATIONS SIGNÉES

Partie prenante	Date	Signature
Ash CALLUM Chief Executive Officer (CEO)		
Natasha JARON Chief Information Officer (CIO)		
Daniel ANTHONY Chief Product Officer (CPO)		
Jo KUMAR Chief Financial Officer (CFO)		