



19 sept. 2022

CONTRAT D'ARCHITECTURE AVEC LES FONCTIONS DESIGN ET DÉVELOPPEMENT

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
INTRODUCTION ET CONTEXTE	4
OBJECTIFS ET PÉRIMÈTRE	5
Objectifs	5
Périmètre	6
ARCHITECTURE, PRINCIPES STRATÉGIQUES, CONDITIONS REQUISES	7
FIG 1. Schéma des composants de l'architecture cible	8
Description	9
Applications front-end	9
Les middlewares	9
Les composants support	9
Le noyau applicatif en couches	9
Base de données	10
Les microservices	10
Principes stratégiques de l'architecture	10
Simplicité	11
Evolutivité	11
Rentabilité	11
Sécurité	12
Référence aux Conditions requises pour l'architecture	12
Processus de déploiement (CI/CD)	14
PLAN DE TRAVAIL COMMUN PRIORISE	15
Élément de travail 1	15
Activités	15
Livrables	15
Élément de travail 2	15
Activités	15
Livrables	15
Élément de travail 3	16
Activités	16
Livrables	16
EVALUATION DES DEMANDES DE CHANGEMENT	17
RISQUES ET FACTEURS DE RÉDUCTION	18

Analyse des risques	18
HYPOTHÈSES	20
CRITÈRES D'ACCEPTATION ET PROCÉDURES	21
Métriques et KPIs	21
Contrat de service application	22
ROADMAP PRÉVISIONNELLE (PROTOTYPE)	23
APPROBATIONS SIGNÉES	24

OBJET DU DOCUMENT

Le présent document 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.

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, la solution technique complexe n'évolue plus au rythme de l'activité et risque d'entraver la croissance de l'entreprise.

Les analyses de marché indiquent que la correspondance avec le marché a été éclipsée par l'instabilité de la plateforme et par une image de marque négative causée par des interruptions de service visibles par les clients.

Les équipes de développement concentrent leurs efforts à maintenir en condition opérationnelle la plateforme via l'introduction de correctif d'urgence.

Une nouvelle plateforme d'e-commerce est nécessaire afin d'améliorer sa compétitivité dans cet environnement concurrentiel intense. Elle devra tirer parti des possibilités offertes par les technologies de géolocalisation afin de faciliter les mises en relation clients / producteurs.

L'innovation et le développement rapide des produits sera placé au cœur de la stratégie de cette nouvelle plateforme, tout en maintenant une cohérence fonctionnelle avec la plateforme existante.

Notons qu'il n'est pas possible d'abandonner les outils actuels pendant l'élaborons des nouveaux car cela impliquerait la mise hors service de la plateforme existante qu'il est nécessaire de maintenir pour pouvoir continuer à accepter de nouvelles adhésions de fournisseurs et de consommateurs. Par ailleurs, il est nécessaire de dissocier les nouvelles livraisons de l'architecture et de l'infrastructure existantes afin de limiter les interruptions de service.

OBJECTIFS ET PÉRIMÈTRE

Objectifs

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

ID	Objectif	Description
O1	Tirer profit de la géolocalisation	La solution tirera parti de la géolocalisation pour relier des fournisseurs et des consommateurs et pour proposer des produits disponibles près des lieux de résidence de ces derniers. Un calculateur de distance sera inclus pour permettre aux consommateurs de trouver les fournisseurs les plus proches d'eux.
O2	Évolutivité	La solution sera évolutive, tant d'un point de vue fonctionnel que technologique. La solution pourra être déployée dans plusieurs villes / région / pays et adaptée à chacun d'eux.
O3	Sécurité	La sécurité de la nouvelle solution fait partie des priorités pour ce chantier d'architecture. La solution doit garantir la sécurité des utilisateurs, de leurs données personnelles, que la sécurité du système d'information dans sa globalité.

O4	Performance	La solution sera performante, peu importe le périphérique utilisé pour y accéder ou la vitesse de connexion internet disponible.
O5	Disponibilité	La solution sera disponible 24h/24h – 7j/7. Les interruptions de services seront inférieures à 2h / mois. Les temps de rétablissement de service seront au maximum d'1 heure. Les modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.
O6	Scalabilité	La solution devra être capable d'absorber les montées en charge ponctuelles ou régulières dû à l'augmentation du nombre d'utilisateurs.
O7	Innovation	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.
O8	Adapté à l'utilisateur	La solution doit être disponible sur l'ensemble des périphériques (mobile, tablette, PC). Elle doit pouvoir prendre en charge divers types d'utilisateurs (par exemple, fournisseurs, back-office, consommateurs), avec des fonctionnalités et des services spécifiques pour ces catégories.

Périmètre

D'un point de vue utilisateur, le périmètre de ce chantier d'architecture sera limité au développement des nouvelles fonctionnalités de géolocalisation et la refonte visuelle des applications web, mobile et tablette.

Toutefois, le chantier technique est bien plus vaste et vise à entamer la conception d'une nouvelle solution complète intégrant la partie Back-Office du système.

A cette fin, il est prévu de mettre en place une plateforme de développement sur laquelle sera placée la nouvelle structure de données ainsi que les requêtes de migration des données . Les différentes couches du noyau applicatif, puis l'ERP interne seront développées par une équipe dédiée sur cette plateforme.

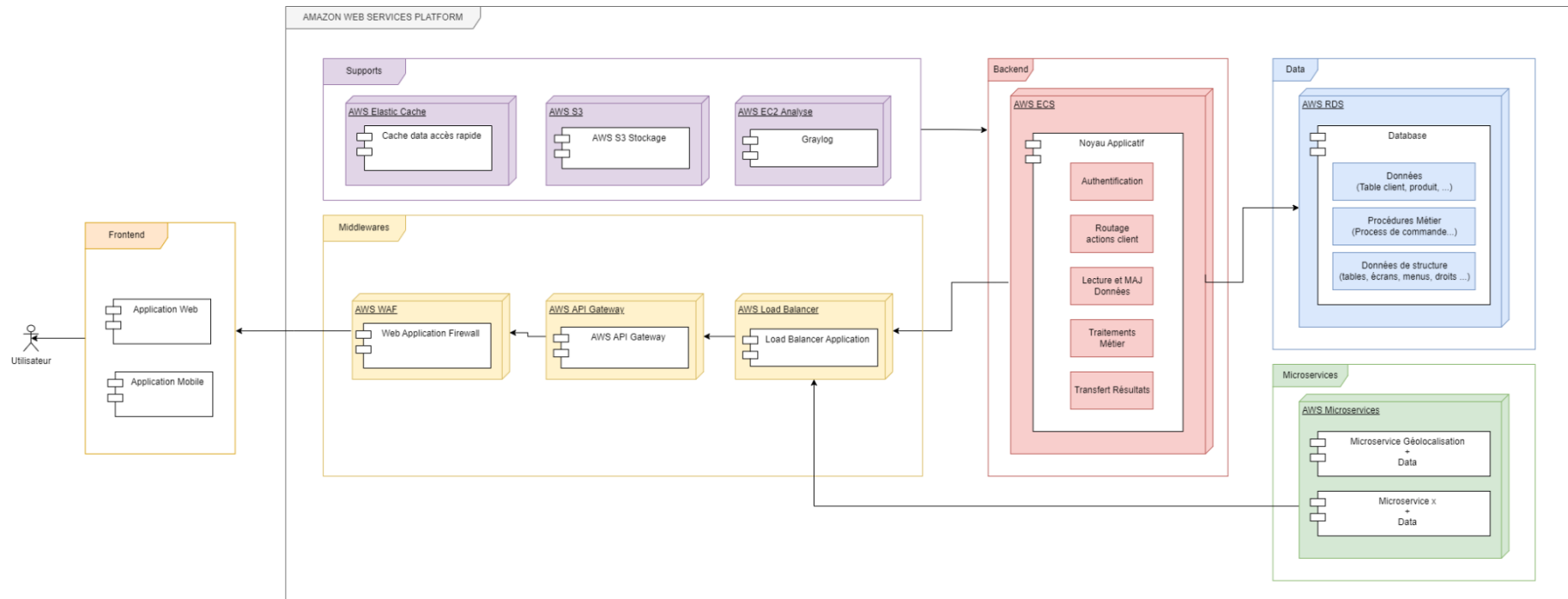
Les nouvelles fonctionnalités seront développées sous forme de Microservices et connectés à la base actuelle. Lorsque l'ERP interne sera en place nous migrerons les données sur la nouvelle plateforme.

ARCHITECTURE, PRINCIPES STRATÉGIQUES, CONDITIONS REQUISES

Schéma de l'architecture

Voir Fig 1. page suivante ...

FIG 1. Schéma des composants de l'architecture cible



Description

L'architecture présentée ci-après représente l'architecture cible. Pour la partie Back-office du système, il s'agit d'une architecture hybride à mi-chemin entre l'architecture client-serveur, l'architecture en couches et l'architecture orientée données.

Pour la partie Front, l'architecture sera à cheval entre une architecture client-serveur et orientée services (microservices plus exactement).

L'architecture est décomposée en 6 grandes parties

Applications front-end

Les applications sont réalisées à l'aide du framework REACT Native permettant la création des versions web et mobile toutes plateformes à l'aide d'un code unique (en Javascript).

Les middlewares

Ils regroupent les composants logiciels assurant la sécurité pour le routage (API Gateway), le contrôle d'accès (WAF) et enfin la répartition de charge (Load Balancing).

Les composants support

Il s'agit des composants fournis avec l'infrastructure AWS assurant le monitoring (Graylog via EC2), le stockage (S3) et l'optimisation des performances avec le cache applicatif (Elasticache).

Ils sont accessibles via les modules du back office ou les microservices spécifiques.

Le noyau applicatif en couches

Il s'agit du bloc désigné Backend.

Il contient les briques standardisées inhérentes à toute application (contrôle d'accès, navigation, recherche, affichage, mise à jour...).

Il accélère la construction et les évolutions du backoffice du système.

Base de données

Il s'agit du bloc intitulé Data d'AWS RDS (PostGre SQL plus spécifiquement). Il permet de stocker dans une base unique, les tables de données propres à l'application mais aussi les tables système (décrivant la structure des données, des formats de saisie, des menus, des procédures, etc).

Les microservices

Ce bloc est réservé aux microservices réalisant les tâches spécifiques qui ne sont pas prises en compte par le noyau applicatif.

Les microservices sont développées en JS et conteneurisées via docker.

Principes stratégiques de l'architecture

L'architecture préconisée se veut progressive pour commencer, se limitant aux développements sans équivalent dans le système actuel et relativement indépendants (Microservice Géolocalisation).

Ces développements permettront à l'entreprise d'innover rapidement via l'intégration continue (CI) et le déploiement continu (CD) sans nécessiter une remise en cause trop drastique du système actuel.

Dans un second temps un noyau applicatif sera mis en place en remplacement des composants actuels.

L'architecture finale préconisée consiste à limiter les développements informatiques à la conception d'un noyau standardisant les opérations communes à toute application. Les opérations d'adaptation aux besoins spécifiques de l'organisation se limiteront ainsi à du simple paramétrage (à l'instar des ERP).

En voici les caractéristiques :

Simplicité

La simplicité est une caractéristique essentielle à toute application. Elle constitue une demande forte des utilisateurs qui disposent de moins en moins de temps pour apprendre les règles de fonctionnement d'un service et de son interface.

Pour l'entreprise, elle consiste à centraliser ses données et piloter toute son activité à distance, avec un seul outil qui doit être spécifique à l'organisation et accessible à tous les acteurs du système d'information (clients, collaborateurs, partenaires ...).

Ces règles sont la condition nécessaire d'une automatisation rapide et complète de l'ensemble des processus métier de l'entreprise et donc de son efficacité et sa rentabilité. Elles favorisent également la participation active des entités externes à l'organisation, limitant la nécessité de recruter sans cesse de nouvelles ressources pour accompagner la croissance.

Evolutivité

Pour rester efficace en étant toujours conforme à l'organisation, un système doit aujourd'hui être capable d'évoluer très rapidement. Dans un contexte en mutation permanente, les outils créés avec les méthodes de développement classiques deviennent obsolètes avant d'avoir pu être exploités.

L'exigence d'obtenir un logiciel conforme au pixel près à ses attentes n'est plus compatible avec la réalité des marchés.

L'entreprise doit disposer d'un système avant tout sobre et efficace, modifiable directement en ligne, à tout moment, selon ses besoins, sans recourir systématiquement aux services d'un expert.

Rentabilité

La rentabilité va tout d'abord consister à disposer rapidement d'une application fonctionnelle et évolutive pour un budget limité.

Les possibilités de fonctionnement à distance et la simplicité de prise en main sont également des facteurs essentiels d'optimisation des ressources (locaux, matériel, personnel).

Sécurité

Cette qualité a été placée en dernier lieu car elle n'est pas spécifique au système préconisée mais reste une condition sine qua non à toute application. La sécurité est assurée grâce au chiffrement des flux et des données stockées, et en hébergeant le système sur une plateforme cloud à haute disponibilité, extensible dynamiquement et redondée : AWS.

Référence aux Conditions requises pour l'architecture

Voici un rappel des exigences d'architecture :

Catégorie	Id	Exigence
Evolutivité	AR1	L'architecture devra être évolutive pour permettre à nos services de se déployer sur diverses régions à travers des villes et des pays donnés.
	AR2	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.
	AR3	Offrir la possibilité de réorienter des solutions existantes, en expérimentant de nouvelles modifications et en facilitant l'intégration avec des partenaires internes et externes.
	AR4	Réduire le temps entre le développement et sa validation afin notamment d'analyser les réactions de nos clients vis-à-vis de

		nouvelles fonctionnalités au fur et à mesure de leur développement.
	AR5	Possibilité d'exécuter diverses variantes ou réaliser des comparaisons de différentes solutions auprès de nos utilisateurs.
	AR6	Besoin de visibilité sur la façon dont nos logiciels sont utilisés.
	AR7	Nous devons pouvoir inverser des décisions d'architecture tant que cela reste peu onéreux.
Disponibilité	AR8	Notre solution doit être disponible pour nos fournisseurs et nos consommateurs, où qu'ils se trouvent.
	AR9	Cette solution doit être utilisable avec des appareils mobiles et fixes.
	AR10	La solution doit tenir compte des contraintes de bande passante pour les réseaux cellulaires et les connexions Internet haut débit.
	AR11	Les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.
	AR12	Choisir une architecture permettant de limiter les risques techniques.
	AR13	Même si le système est surchargé, les utilisateurs connectés doivent pouvoir continuer à accéder à tous les services de façon dégradée.

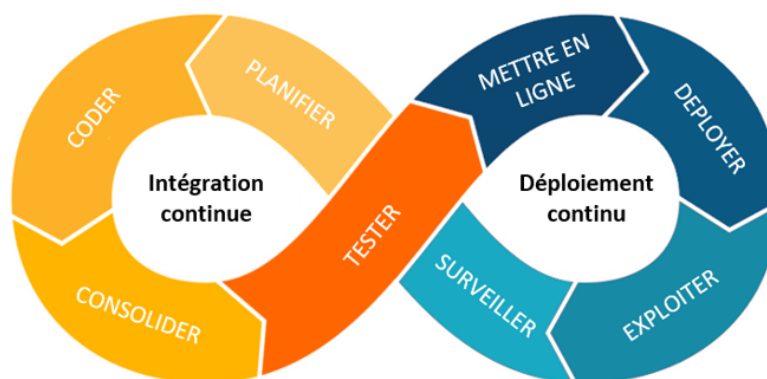
	AR14	Chaque nouvelle version doit être de taille réduite, présenter peu de risques, être transparente pour nos utilisateurs et rester accessible en tout lieu et à tout moment.
Spécificité	AR15	La solution doit pouvoir prendre en charge différents types d'utilisateurs (par exemple, fournisseurs, back-office, consommateurs) avec des fonctionnalités et des services spécifiques pour ces catégories.
Simplicité	AR16	La solution doit offrir une expérience utilisateur de premier plan (simplicité, UX Design).

Processus de déploiement (CI/CD)

Afin de satisfaire à l'objectif de déploiement rapide et favoriser l'innovation, le processus de déploiement s'appuiera sur des pipelines CI/CD intégrant :

- Des tests unitaires / fonctionnels automatisés
- De l'analyse de code automatisé,
- Des contrôles de conformité (sécurité, code qualité ...)
- Du déploiement continu automatisé sur les environnements de développement et pré-production

(avec validation humaine pour l'environnement de production)



PLAN DE TRAVAIL COMMUN PRIORISE

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

EVALUATION DES DEMANDES DE CHANGEMENT

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 référent	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

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	Évaluer en amont les capacités théoriques de connexion

					<p>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 et KPIs

Le catalogue ci-après définit les métriques de référence (KPI) utilisées pour mesurer le succès d'un point de vue business de la nouvelle 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

Contrat de service application

Les accords de niveau de service, imposés par Foosus pour satisfaire aux demandes des investisseurs, déterminent les niveaux de service à atteindre sur le système de production.

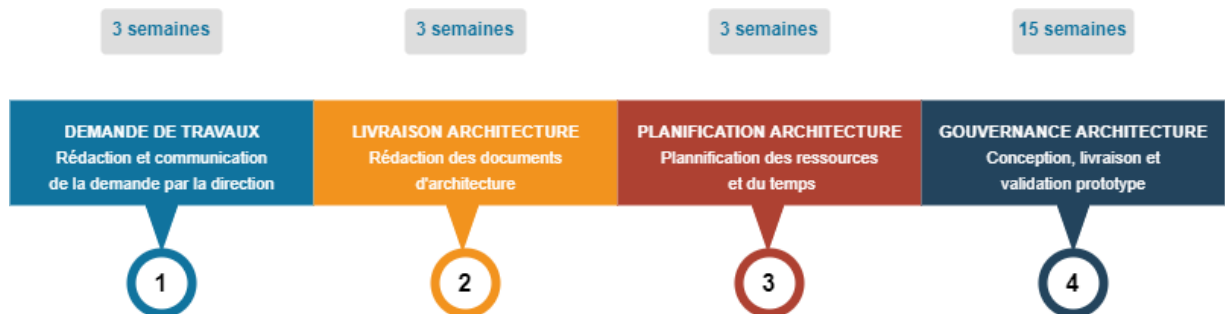
En cas de non-respect d'un ou plusieurs termes du contrat de service, des mesures coercitives devront être mises en œuvre sans délai.

Id.	Objectif de niveau de service	Mesure
A-SLA1	Le délai de déploiement d'une fonctionnalité achevée doit être ≥ 7 jours.	Délai moyen entre merge-request sur une branche « master » et déploiement de la branche.
A-SLA2	Le taux de couverture de code par des tests automatisés devra être $\geq 75\%$.	Rapport de couverture de code par les tests.
A-SLA3	Le nombre de vulnérabilités / mauvaises pratiques de sécurité détectées dans les applications devront être égal à 0.	Analyse statique et code-review.
A-SLA4	Le code des logiciels fournis devra être en conformité avec les chartes de développement établi et les mesures d'analyse automatisé de la « dette technique » devront être $< 1h$ / dépôt de code source.	Analyse statique et code review.
A-SLA5	Les logiciels livrés devront être totalement conformes aux principes de conception définis par la gouvernance d'architecture.	Validation des logiciels par la gouvernance d'architecture.
A-SLA6	Les navigateurs web disposant d'une part de marché $> 2\%$ des utilisateurs (Mesures Foosus ou GlobalStats, la plus haute étant retenue) mobile ou desktop seront supportés.	Analyse statistiques + Plan de test.

ROADMAP PRÉVISIONNELLE (PROTOTYPE)

La roadmap ci-après présente les grandes étapes pour ce chantier d'architecture.

Une seconde roadmap sera produite durant la phase de planification et affinée en phase de prototype pour le déploiement de la solution à grande échelle. Notons que cette phase de prototypage est impérative pour s'assurer du succès de l'architecture finale et supprimer l'ensemble des inconnues inhérentes aux projets de cette ampleur.



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)		
hervé PREVOST Architecte Logiciel (AL)		