



Contrat de Conception et de Développement de l'Architecture

Projet :

État Cible de l'Architecture et implémentation d'une nouvelle architecture

Rédaction : Nicolas Cervantes
Valideur : Eric Legba



Table des Matières

- 1. Objet de ce document**
- 2. Introduction et contexte**
- 3. Objectifs et périmètre**
- 4. Description de l'architecture, principes stratégiques et conditions requises**
- 5. Livrables architecturaux**
- 6. Plan de travail commun priorisé**
- 7. Diagramme de la nouvelle Architecture**
- 8. Plan de communication**
- 9. Critères d'acceptation et procédures**
- 10. Procédures de changement de périmètre**
- 11. Phases de livrables définies**
- 12. Personnes approuvant ce plan**



1. Objet de ce document

Les Contrats d'Architecture sont les accords communs entre les partenaires de développement et les sponsors sur les livrables, la qualité, et la correspondance à l'objectif d'une architecture. L'implémentation réussie de ces accords sera livrée grâce à une gouvernance de l'architecture efficace (voir TOGAF Partie VII, Gouvernance de l'architecture). En implémentant une approche dirigée du management de contrats, les éléments suivants seront garantis :

- *Un système de contrôle continu pour vérifier l'intégrité, les changements, les prises de décisions, et l'audit de toutes les activités relatives à l'architecture au sein de l'organisation.*
- *L'adhésion aux principes, standards et conditions requises des architectures existantes ou en développement*
- *L'identification des risques dans tous les aspects du développement et de l'implémentation des/de l'architecture(s), y compris le développement interne en fonction des standards acceptés, des politiques, des technologies et des produits, de même que les aspects opérationnels des architectures de façon à ce que l'organisation puisse poursuivre son business au sein d'un environnement résilient.*
 - *Un ensemble de processus et de pratiques qui garantissent la transparence, la responsabilité et la discipline au regard du développement et de l'utilisation de tous les artefacts architecturaux*
 - *Un accord formel sur l'organe de gouvernance responsable du contrat, son degré d'autorité, et le périmètre de l'architecture sous la gouvernance de cet organe*

Ceci est une déclaration d'intention signée sur la conception et le développement de l'architecture d'entreprise, ou de parties significatives de celles-ci, de la part d'organisations partenaires, y compris les intégrateurs système, fournisseurs d'applications, et fournisseurs de service.

De plus en plus, le développement d'un ou plusieurs domaine(s) d'architecture (business, données, application, technologie) peut être externalisé, avec la fonction d'architecture de l'entreprise fournissant une vue d'ensemble de l'architecture d'entreprise globale, ainsi que la coordination et le contrôle de l'effort total. Dans certains cas, même ce rôle de supervision peut être externalisé, bien que la plupart des entreprises préfèrent conserver cette responsabilité clé en interne.

Quelles que soient les spécificités des dispositions d'externalisation, les dispositions elles-mêmes seront normalement gouvernées par un Contrat d'Architecture qui définit les livrables, la qualité, et la correspondance à l'objectif de l'architecture développée, ainsi que les processus de collaboration pour les partenaires du développement de l'architecture.



2. Introduction et Contexte

La Nature de l'accord

Ce document traite les points les plus importants afin d'établir une communication entre les membres de la technique, celle-ci permettra une compréhension générale et spécifique sur les besoins techniques qui serviront de base et structure principale pour la correcte conception, réalisation et implémentation de la nouvelle architecture, afin de répondre à une évolution adaptée au marché et besoins ultimes des clients de la plateforme.

3. Objectifs et périmètre

Objectifs

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

1. Etablir une interface entre les consommateurs et fournisseurs.
 - a. Les consommateurs auront accès aux offres des fournisseurs
 - b. Les fournisseurs pourront connaître l'ensemble des habitudes d'achat de leurs produits via la plateforme
2. Augmenter les quantités de clients dans la plateforme
 - a. Des campagnes de marketing seront mises en place afin d'augmenter les visites dans la plateforme.
 - b. Les clients auront plus de facilités au moments de créer leurs profils, voir des remises pour leurs premiers achats
3. Réduire les contraintes de performance afin d'offrir un service sans interruptions.
4. Evolutivité adaptée à la croissance.
5. Adaptabilité aux demandes du marché.



Périmètre

Périmètre Business :

1. **Relier les clients**, augmenter le nombre de fournisseurs à la plateforme afin d'augmenter les offres de produits, et de façon directe la quantité de consommateur et client.
2. **Déploiement territorial**, rendre disponible les services dans plus de pays et région.
3. **Liberté d'innovation**, pour les équipes de marketing et produits afin de faire évoluer les besoins des campagnes publicitaires et tests des équipes chargés du marketing.

Périmètre Gestion :

1. **Gestion des livrables**, les documents et livrables devront être regroupés dans un répertoire et être actualisés de façon méthodique, suite à des changements ou incorporation de nouvelles fonctionnalités.
2. **Types de profils** : Pour chaque cas et types d'utilisateurs un type de profil devra être créé, les fournisseurs auront accès à des fonctionnalités différentes que les consommateurs, ainsi que le profil métier et technique qui permettront d'accéder, modifier et ou créer des rapports.

Périmètre Technique :

1. Etablir un **équilibre de charge** entre le flux de visiteurs accédant aux services de la plateforme et la disponibilité des infrastructures afin de ne jamais avoir des coupures de services.
2. Implémenter une **évolutivité et scalabilité** dans la plateforme, résistante aux demandes et besoin de croissance, pour ainsi offrir un service continu.
3. **Déploiement continu**, les équipes de développement devront avoir la liberté de tester et déployer de nouvelles fonctionnalités afin de permettre une évolution organique et innovantes des services de la plateforme Foosus.
4. **Proposer des produits** et fournisseurs basé sur la géolocalisation des utilisateurs, à un rayon modifiable et adaptable par l'utilisateur.
5. **Incorporer la sécurité**, l'architecture et système devront faire preuve de résilience, résistance et prévention face aux événements d'instabilité ou risques d'attaque informatique, authentification, rôles : attribution de fonctionnalités, échange d'information entre microservices.



Parties prenantes, préoccupations et visions		
Partie prenante	Préoccupation	Vision
Ash Callum CEO	Business	1. Innovation 2. Croissance 3. Communication 4. Améliorer les résultats et maximiser la rentabilité à long terme
	Data	
	Innovation technique rapide et expérimentation	
Natasha Jarson CIO	Business	1. Innovation 2. Enthousiasmer les équipes 3. Différencier la position sur le marché 4. Déploiement et gestion de systèmes maintenables, scalable, performant et sûr.
	Applicative	
	Technologique	
	Data	
	Innovation dans le périmètre d'une architecture	
Daniel Anthony CPO	Innovation technique rapide et expérimentation	1. Innovation 2. Améliorer l'expérience du produit. 3. Stratégie du produit 4. Management de l'équipe produits 5. Rapports basés sur les données
	Data	
	Business	
	Visibilité de la plateforme	
	Innovation dans le périmètre d'une architecture	
Christina Orgega CMO	Innovation technique rapide et expérimentation	1. Fixer des buts et objectifs marketing 2. Planifier, mettre en œuvre et gérer la stratégie marketing 3. Déterminer les KPI pour le service marketing 4. Créer et présenter le plan et la stratégie marketing annuels 5. Utiliser des données et des rapports pour prendre des décisions fondées sur des preuves
	Business	
	Applications	
	Taux d'inscription utilisateurs	
Jo Kumar CFO	Visibilité de la plateforme	1. Assurer le leadership et la gestion du département financier 2. Soutenir le service financier dans l'élaboration d'objectifs financiers à court et à long terme 3. Créer des plans financier 4. Superviser le processus les devis annuel. 5. Gérer la liquidité de l'entreprise. 6. Créer un système de contrôle des coûts
	Data	
	Applications	
	Taux d'inscription utilisateurs	



Pete Parker Engineering Owner	2	Applicative	1. Examens d'évaluation, de faisabilité et de planification du projet 2. Revues de conception technique et de planification 3. Analyse et optimisation des horaires 4. L'analyse coûts-avantages
		Data	
Jack Harner Operations Lead	3	Visibilité de la plateforme	1. Faciliter les interactions entre l'équipe interne et les plateformes hébergées pour le succès global de l'organisation 2. Encadrer les membres de l'équipe 3. Identifier les manquements 4. Organiser une formation pour doter le personnel des connaissances sur les tendances en développement afin de créer un service hautement qualifié
		Applicative	
		Business	
Nicolas Cervantes Solution Architect	4	Business	1. Créer une vision technique pour montrer comment le logiciel couvrira les besoins de l'entreprise. 2. Transmettre la vision technologique à l'équipe de développement 3. Assurer l'interface entre l'équipe business et développement. 4. Décider d'une pile (stack) technologique.
		Technologique	
		Applicative	
		Data	
Product Team	2	Business	1. Améliorer l'expérience du produit. 2. Stratégie du produit

4. Description de l'architecture, principes stratégiques et conditions requises

Description

L'architecture repose dans le cloud en utilisant des microservices, chaque microservice aura sa base de données.

Principes stratégiques

Stratégie d'implémentation

Objectifs et processus techniques à suivre pour une bonne implémentation de la nouvelle architecture

1. Découpler les systèmes de base.



Les systèmes existant devront être divisé afin de pouvoir intégrer les nouvelles fonctionnalités et répondre aux besoins techniques de la nouvelles architecture

2. Réduire la dépendance aux batch jobs



Les batch jobs devront être réduit et même éliminés des processus de développement pour ainsi faire place aux nouveaux service d'automatisation de taches de la plateforme AWS

3. Exposer les évènements business



Pour respecter les principaux besoins du projet, les évènement business seront intégrés dans les processus de développement, cela assurera la globalité de demandes métier.

4. Faire des données quelque chose d'accessible



Les equipes de developement, marketing, finances et autres parties prenantes ayant besoins d'accès à l'information nécessaire pour accomplir leurs missions.

5. Simplicité opérationnelle



Un processus de gestion et de développement fera partie des taches journalières des équipes, tout en gardant l'efficacité du Lean.

Intégration Continue



L'intégration continue est un processus de développement de logiciels où les développeurs intègrent le nouveau code qu'ils ont écrit plus fréquemment tout au long du cycle de développement, en ajoutant à la base de code au moins une fois par jour. Les tests automatisés sont effectués contre chaque itération de la construction pour identifier les problèmes d'intégration plus tôt, lorsqu'ils sont plus faciles à réparer, ce qui aide également à éviter des problèmes à la fusion finale de la libération. Globalement, l'intégration continue aide à rationaliser le processus de construction, ce qui entraîne un logiciel de qualité supérieure et des calendriers de livraison plus prévisibles.

Avantages de l'intégration continue :

- Progrès continus et démontré pour améliorer les commentaires
- Détection d'erreur et métriques d'erreur précoces et améliorées qui permettent d'aborder des erreurs, parfois quelques minutes d'enregistrement
- Une collaboration améliorée de l'équipe ; Toute l'équipe peut changer le code, intégrer le système et déterminer rapidement les conflits avec d'autres parties du logiciel.
- L'intégration du système améliorée, qui réduit les surprises à la fin du cycle de vie du développement de logiciels
- Moins de changements parallèles pour la fusion et les tests
- Nombre réduit d'erreurs lors des tests du système
- Systèmes constamment mis à jour pour tester entre eux

5. Livrables architecturaux

Architecture, caractéristiques et composants du système

Architecture

Architecture en microservices :

L'architecture du nouveau système sera centrée, gérée et développée sous des microservices.

Pourquoi une architecture en microservices ?



- **Délai de mise sur le marché plus courte.** Lorsque différentes équipes peuvent développer leurs microservices de manière indépendante, l'ensemble du produit a moins de blocages et le temps de mise sur le marché global diminue.
- **Évolutivité améliorée.** Lorsque la demande de fonctionnalités particulières augmente ou diminue, les microservices peuvent évoluer séparément sur plusieurs serveurs et fournisseurs.
- **Résilience des opérations.** Comme les microservices fonctionnent indépendamment les uns des autres, la défaillance d'un composant n'arrêtera pas l'ensemble du produit, comme c'est le cas avec les applications monolithiques, et le module défectueux est facile à redémarrer.
- **Efficacité de la gestion.** Les produits logiciels modulaires basés sur des microservices sont beaucoup plus faciles à déployer, mettre à jour et gérer.
- **Facilité d'évaluation du produit.** Lorsque des applications monolithiques sont divisées en morceaux indépendants, il est beaucoup plus facile de comprendre le code source de chaque microservice, ce qui permet aux développeurs de les évaluer et de les mettre à jour beaucoup plus rapidement.

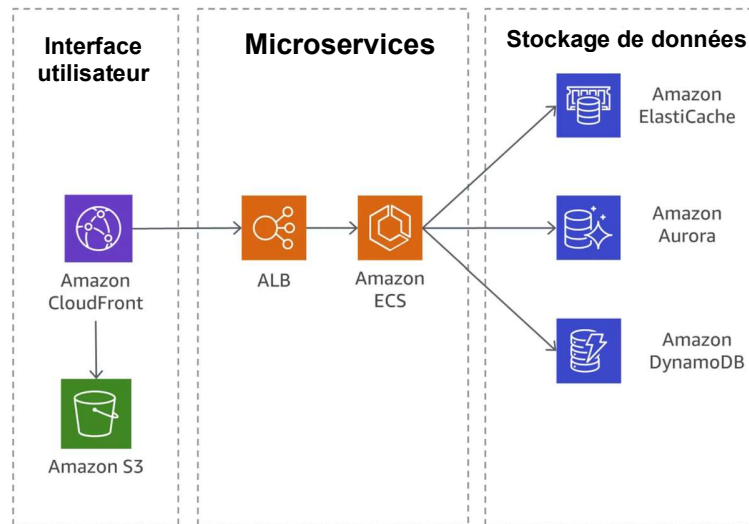
Cette architecture en microservice sera réalisée en utilisant les services offerts par la **plateforme Cloud de Amazon : AWS**.

Amazon Web Services (AWS)

AWS est un service cloud d'Amazon, qui fournit des services sous forme de blocs de construction, ces blocs de construction peuvent être utilisés pour créer et déployer tout type d'application dans le cloud.

Ces services ou blocs de construction sont conçus pour fonctionner les uns avec les autres et donnent lieu à des applications sophistiquées et hautement évolutives.

La plate-forme est développée avec une combinaison d'offres d'infrastructure en tant que service (IaaS), de plate-forme en tant que service (PaaS) et de logiciel en tant que service (SaaS).



Architecture en Microservices sous AWS

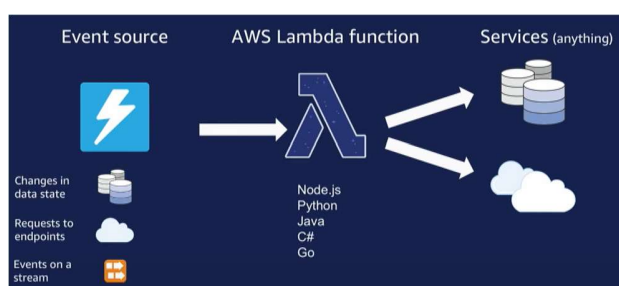
Caractéristiques de l'architecture

- **Hébergé dans le cloud (Plateforme AWS)**
- **Architecture en Microservices :**
Cette architecture permettra aux développeurs plus de flexibilité aux moments de créer et maintenir les différentes fonctions et services de la nouvelle architecture.
- **Méthodologie de développement et programmation asynchrone :**
 - Définit la relation entre deux ou plusieurs objets qui interagissent les uns avec les autres dans le même environnement ou système, mais à des phases temporelles ou à des instants différents, et ne s'affecte pas non plus les uns les autres.
 - Le processus asynchrone définit également un processus qui fonctionne indépendamment des autres processus.
 - Cela permettra d'éviter que l'écran reste figé, et donner au client l'impression que le processus prend des mesures visuellement, mais ce n'est généralement pas le cas, les calculs sont fait en arrière-plan.

Système Scalable et adaptable

- Développement évitant les points de défaillance unique.
- Utilisations de mécanismes automatisés de surveillance, de détection des pannes et de basculement pour les composants sans état et avec état

- Elimination des points de défaillance uniques (SPOF) avec une configuration de redondance N+1 ou 2N, où N+1, obtenu via l'équilibrage de charge entre les nœuds actifs-actifs, et 2N est obtenu par une paire de nœuds en configuration actif-veille.
- **Application Serverless :**
 - Aucun serveur à provisionner ou à gérer
 - Échelle avec l'utilisation
 - Aucun paiement pour les repos des systèmes
 - Disponibilité et tolérance aux pannes intégrées



Composants de l'architecture

Finalité	Composant Architectural	Détails
Microservices	Fonctions AWS Lambda	1. Outil de création de fonctions 2. Différentes options de langages 3. Coûts très bas
Microservices	AWS Step Functions	AWS Step Functions fournit une orchestration sans serveur pour les applications modernes. Orchestration gère de manière centralisée un workflow en le divisant en plusieurs étapes, en ajoutant une logique de flux et en suivant les entrées et les sorties entre les étapes.
Stockage	AWS S3	Permet le stockage des éléments du site web, afin qu'ils soient accessibles aux utilisateurs au moment d'accéder à la plateforme
Charge de travail	Amazon Application Load Balancer (ALB)	L'équilibreur de charge d'application fonctionne au niveau de la demande (couche 7), en acheminant le trafic vers les cibles (instances EC2, conteneurs, adresses IP et fonctions Lambda) en fonction du contenu de la demande. Idéal pour l'équilibrage de charge avancé du trafic HTTP et HTTPS, l'équilibreur



		de charge d'application fournit un routage de requête avancé ciblé sur la livraison d'architectures d'applications modernes, y compris des microservices et des applications basées sur des conteneurs.
Microservices	Amazon Elastic Container Service (ECS)	Amazon Elastic Container Service (Amazon ECS) est un service de gestion de conteneurs rapide et hautement évolutif qui facilite l'exécution, l'arrêt et la gestion des conteneurs sur un cluster. Amazon ECS peut être utilisé pour créer une expérience de génération et de déploiement cohérente, pour gérer et mettre à l'échelle les charges de travail par lots et Extract-Transform-Load (ETL), et pour créer des architectures d'applications sophistiquées sur un modèle de microservices.
Microservices	Docker	Application de conteneurisation des microservices
Interface Utilisateur	Amazon Cloud Front	Service Web qui accélère la distribution du contenu Web statique et dynamique, tel que .html, .css, .js et fichiers image, aux utilisateurs. CloudFront diffuse le contenu via un réseau mondial de centres de données appelés emplacements périphériques. Lorsqu'un utilisateur demande du contenu qui est servi avec CloudFront, la demande est acheminée vers l'emplacement périphérique qui offre la latence la plus faible (délai), afin que le contenu soit livré avec les meilleures performances possibles.
Data	Database Migration Service (DMS)	WS Database Migration Service (AWS DMS) aide à migrer des bases de données vers AWS. La base de données source reste pleinement opérationnelle pendant la migration, minimisant ainsi les temps d'arrêt des applications qui reposent sur la base de données. AWS Database Migration Service peut migrer les données vers et depuis les bases de données commerciales et open source les plus utilisées.
Data	Snowball	Application qui permet de transférer des téraoctets de données à l'intérieur et à l'extérieur de l'environnement AWS.



Data	OracleNoSQL	<ol style="list-style-type: none"> 1. Types de données intégrés pour Array, Range, UUID, Géolocalisation, etc. 2. Prise en charge native du stockage de documents (style JSON), XML et stockage de valeurs-clés (Hstore) 3. Réplication synchrone et asynchrone 4. Scriptable en PL, Perl, Python, etc. 5. Recherche en texte intégral
-------------	--------------------	---

Les microservices implémentés

	Types	Microservice	Description
1	Commandes	Pipeline des commandes	Toutes les commandes devront passer par un processus circuit de validation allant de la sélection des produits à la capture et validation du paiement.
2	Commandes	Validation de la commande	La dernière phase du circuit de prise d'achat et prise de commandes
3	Commandes	Paiement de la commande	L'avant dernière étape du circuit de prise de commande qui gère l'accès et demandes aux banques des consommateurs afin de réaliser les paiements.
4	Service Client	Ticketing demandes	Microservices qui gère les fonctionnalités du système de ticketing pour les demandes des utilisateurs afin de résoudre des conflits techniques ou logistiques.
5	Profil Utilisateur	Panier	La page et fonctionnalités du panier devront couvrir les besoins basiques, incluant les rappels des éléments du panier, des offres et promotions afin



			d'inviter l'utilisateur à achever les achats.
6	Localisation	Géolocalisation	Fonctions d'offres de produits et services adapté aux données GPS et rayon défini par le consommateur.
7	Profil Utilisateur	Favoris	En lien avec le panier les utilisateurs devront avoir la possibilité de mettre de côté les produits
8	Profil Utilisateur	Authentification	Gestion d'authentification incluant la capture des identifiant.
9	Profil Utilisateur	Récupération de compte	Services permettant aux utilisateurs de réinitialiser le mot de passe et récupérer l'accès au compte.
10	Profil Utilisateur	Création de profil consommateur	Les consommateur auront accès a des fonctionnalités spécifiques afin d'avoir une expérience d'achat idéale.
11	Profil Utilisateur	Création de profil fournisseur	Les fournisseurs auront accès a des fonctionnalités spécifiques afin d'avoir une expérience de vente, gestion de stock et clients adéquate.
12	Produits	Gestion des produits	La page de gestion de produits permettra modifier et gérer les prix, images et caractéristique de leurs produits.
13	Service Client	Messagerie consommateurs-fournisseurs	Service de communication entre les fournisseurs et consommateurs.
14	Service Client	Messagerie fournisseurs-métier	Service de communication entre les fournisseurs et le service client de Foosus.
15	Commandes	Transactions et paiements	Services



			Tokenisation et intégration de processus de paiement.
16	Marketing	Campagnes publicitaires	Les fournisseurs pourront établir et adapter des campagnes publicitaires selon des dates, horaires, produits et définir des saisonnalités.
17	Finance	Business Intelligence	Les équipes de marketing et finances auront accès à des fonctionnalités d'analyse de ventes, projection et autre outils couvrant le business intelligence
18	Service Client	Avis clients	Les consommateurs pourront laisser des avis des produits, services et commandes pour que les fournisseurs et le métier de Foosus puissent faire évoluer la qualité des prestations.

Sécurité de l'architecture

L'approche de modélisation des menaces (threat modeling)

La modélisation des menaces implique l'identification et la communication d'informations sur les menaces susceptibles d'avoir un impact sur un système ou un réseau particulier. La modélisation des menaces de sécurité permet à une équipe informatique de comprendre la nature des menaces, ainsi que leur impact potentiel sur le réseau. De plus, la modélisation des menaces peut être utilisée pour analyser les dangers que représentent les menaces pour les applications, en tenant compte de leurs vulnérabilités potentielles.

Les 14 points principaux pour implémenter la modélisation des menaces :

1. Construire une conception, un modèle de réseau ou un système de défense des applications sécurisé
2. S'assurer que les ressources sont investies efficacement pour éviter de jeter de l'argent ou des



personnes inutilement sur un problème

3. Faire de la sécurité une priorité, même en la plaçant avant la rentabilité à court terme, sachant que les bénéfices seront améliorés à long terme avec un système plus sûr
4. Tenir les parties prenantes informées de l'évolution du système
5. Spécification des menaces auxquelles le système est confronté
6. Identifier les exigences de conformité en ce qui concerne le système ou l'application traitée
7. S'assurer que les mesures prises sont conformes aux réglementations de conformité
8. Définir les contrôles nécessaires pour atténuer la menace avant, pendant et après son apparition
9. Construire ces contrôles et les déployer de manière transparente et claire pour toutes les parties prenantes
10. Évaluer les risques liés au système de gestion des menaces
11. Documenter les menaces qui affectent le système
12. Documenter les efforts d'atténuation appliqués pour gérer chaque menace
13. S'assurer que les objectifs de l'entreprise ne sont pas affectés par un acteur de menace ou un événement négatif
14. Déterminer les moyens de tester le système pour s'assurer qu'il fonctionnera compte tenu des menaces contre lesquelles il est conçu pour protéger l'organisation

Autres éléments pour accomplir la sécurité du nouveau système

- **Amazon API Gateway :**
 - Gère l'autorisation et l'authentification au niveau des microservices
- **Firewalls :**
 - Il est recommandé de positionner un firewall avant chaque microservice.
- **Processus DevSecOps :**
 - Automatise l'intégration de la sécurité à chaque phase du cycle de vie du développement logiciel, de la conception initiale à l'intégration, aux tests, au déploiement et à la livraison du logiciel.
 - Un avantage clé de DevSecOps est la rapidité avec laquelle il gère les vulnérabilités de sécurité nouvellement identifiées. Comme DevSecOps intègre l'analyse et la correction des vulnérabilités dans le cycle de publication, la capacité d'identifier et de corriger les vulnérabilités et les expositions communes (VEC) est diminuée. Cela limite la fenêtre dont dispose un acteur malveillant pour tirer parti des vulnérabilités des systèmes de production destinés au public.
- **APIs publiques et privées**
- **Tokenisation :**



- Demande et renvoi de token entre l'utilisateur, le pool d'utilisateurs et l'API Gateway, assurera une sécurité supplémentaire au moment d'initier une session.
- **MFA par SMS :**
 - L'authentification par multiple facteur via l'envoi de SMS permettra d'ajouter une couche de sécurité supplémentaire au moment où les utilisateurs accéderont à la plateforme.
- **Sécurisation des points d'accès avec OAUTH/OAUTH 2.0 :**
 - La plateforme et processus OAuth assureront l'autorisation d'accès à l'API Gateway.

Stack Technologique de la nouvelle Architecture

Eléments du Stack Technologique

	Type	Categorie	Fournisseur	Techno.	Open Source
1	Plateforme	Backend	Amazon	Amazon Web Services	non
2	Service	Backend	AWS	Fonctions AWS Lambda	non
3	Service	Backend	AWS	AWS Step Functions	non
	Service	Frontend	AWS	AWS S3 Bucket	non
4	Service	Backend	AWS	Amazon Application Load Balancer (ALB)	non
5	Service	Backend	AWS	Amazon Elastic Container Service (ECS)	non
6	Service	Backend	AWS	Amazon Cognito	non
7	Service	Frontend	AWS	Amazon Cloud Front	non
8	Service	Backend	AWS	Amazon Application Load Balancer (ALB)	non
9	Service	Data	AWS	Database Migration Service (DMS)	non
11	Langage de programmation	Data	Oracle	OracleNoSQL	oui
12	Plateforme	Système Gestion de commandes	Hydra OMS	Hydra OMS	oui



Date rev : 15/09/2021

13	Langage de programmation	Frontend	whatwg	Html	oui
	Langage de programmation	Système Gestion de commandes	Ruby on Rails	Ruby on Rails	oui
	Langage de programmation	Interfaces	Json	Json	non
	API	Interfaces	REST	REST	non
14	Langage de programmation	Frontend	ECMA / Oracle	JavaScript	oui
15	Langage de programmation	Frontend	W3C	Css	oui
16	Service	Géolocalisation	OpenCagedata	Javascript	non
17	Service	Protocole d'autorisation	Groupe OAuth	OAuth 2.0	oui
18	Service	Répertorisation	Software Freedom Conservacy	Git	oui
19	Service	Interface répertorisation	Microsoft	GitHub	non

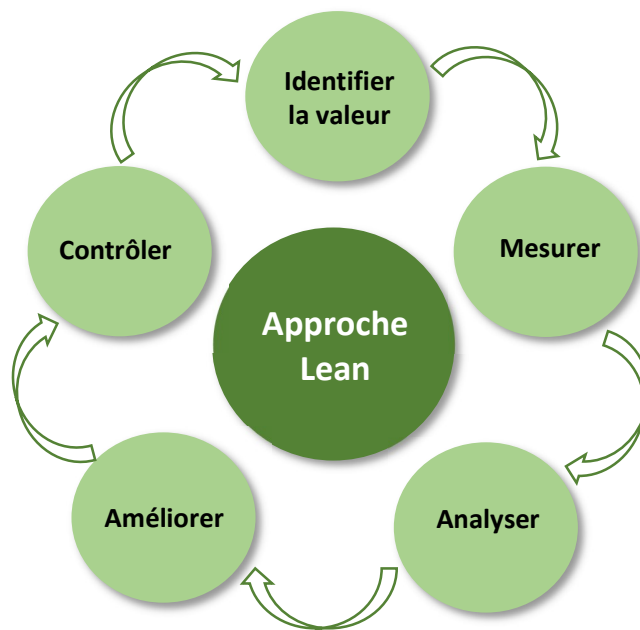


6. Plan de travail commun priorisé

Une approche *Lean* pour la gestion et le développement

Cette approche managériale pour le développement de la nouvelle plateforme permet de :

- **Prioriser l'expérience utilisateur :**
 - Les utilisateurs devront avoir une expérience de haute qualité, adapté aux besoins et cas spécifiques de la demande. Un étude approfondie devra être réalisé sur les étapes et trajets de chaque menu et boutons que les utilisateurs auront.
- **Éliminer les déchets :**
 - Tout surplus d'éléments dans le développement et documentation doivent être éliminés afin de rendre la gestion plus efficace.
- **Développement centré sur l'intégrité :**
 - Chaque élément crée (développement/documentation) fera partie d'une analyse rigoureuse d'intégrité pour éviter et prévenir des erreurs même avant les phases de tests, cela rendra le travail plus efficace.
- **Responsabiliser l'équipe :**
 - Avoir accès aux commentaires des clients
 - Être capable de communiquer entre les départements
 - Être habilité à suggérer des améliorations tout au long du cycle de vie du produit.
- **Livrer rapidement :**
 - Les équipes de développement de logiciels agiles doivent viser à créer une solution simple, livrer ce produit et continuer à apporter des améliorations incrémentielles sur ce produit en fonction des commentaires des clients.
 - Pour obtenir un développement logiciel plus rapide, il faut identifier toutes les étapes qui ralentissent et éliminer celles qui ne contribuent pas à la qualité de la solution logicielle à fournir.
- **Optimiser l'ensemble du processus :**
 - Any inefficiency should be quickly identified and cut out before it impacts workflows.
- **Amplifier l'apprentissage**
 - asserts that your development team should be continually acquiring new, relevant knowledge in order to create a product that is innovative and serves customer needs.



Les 5 piliers d'une approche Lean

Livrables de la nouvelle Architecture

Les Livrables

Les produits de travail suivants seront créés en résultat de ce travail d'architecture :

1. Diagramme complet de la nouvelle architecture
2. Stack Technologique utilisé
3. Répertoire GitHub avec accès aux documents du projet
4. Documents d'Architecture



Intégration d'un Système de Gestion de Commandes (OMS)

Description

Le logiciel de gestion des commandes (OMS) fournira à Foosus tous les outils et fonctionnalités nécessaires pour une gestion efficace des commandes. Le logiciel de gestion des commandes sera utilisé pour saisir et traiter divers types de commandes par les entreprises. La plupart des OMS sont une suite logicielle individuelle et autonome entièrement dédiée à la gestion des commandes uniquement. Certains OMS font partie ou font partie du système de planification des ressources de l'entreprise (ERP). Un OMS est conçu pour résoudre divers problèmes opérationnels liés au traitement et à l'exécution des commandes.

Éléments essentiels du futur OMS de Foosus :

1. Centralisation des processus :

Grâce à la centralisation des processus, les organisations peuvent avoir une vue à guichet unique du processus de commande complexe. Le système centralisé affiche les informations relatives aux commandes, aux stocks et aux expéditions sur un seul tableau de bord à partir duquel elles peuvent être contrôlées et coordonnées.

2. Intégration à canaux multiple :

Les partenaires commerciaux et les autres canaux peuvent être intégrés au logiciel de gestion des commandes pour fournir un tableau de bord unique des ventes à partir de plusieurs canaux de vente.

3. Automatisations :

Les organisations peuvent automatiser les activités régulières par OMS pour réduire le travail manuel. Ce système allège la charge des commandes, en particulier lorsque la rotation des stocks est élevée et que les entreprises doivent traiter des centaines de commandes par jour.

4. Expédition et logistique :

Cet élément et ses caractéristiques prennent en charge tous les travaux liés à l'expédition et à la logistique de l'entreprise.

5. Traitement des paiements :

Le logiciel peut être intégré à plusieurs canaux de paiement pour offrir aux clients un afflux facile de modes de paiement.

6. Analyses et rapports en temps réel :



Cet élément a des fonctionnalités pour fournir divers rapports analytiques concernant les commandes et les performances commerciales qui y sont liées.

Composants du Système de Gestion de Commandes

Le Système de Gestion de Commandes se compose principalement des éléments suivants :

Fonctionnalité OMS	Composants de la fonctionnalité
1. Gestion du dépôt	Inventaire physique, réception de commandes, nombre de cycles, changement des quantités et déplacement
2. Transactions Financières	Remboursement, comptes débiteurs, données des carte de crédit.
3. Contrôle des données	Rapports, Importation des données
4. Traitement des paiements	Tokenisation, intégration des processeurs de paiement
5. Traitement des commandes	Expédition des commandes, fonctions de services client. Impression des commandes
6. Marketing	Liste des prix, extraction de clients, campagnes publicitaires
7. Gestion de l'inventaire	Achats, personnalisation, fournisseurs.

Le Système de Gestion de Commandes Hydra

Point généraux sur le système :

- Open Source
- Automatiser le flux d'exécution des commandes
- Assistant d'exécution d'ordre convivial
- Processus d'exécution rapide des commandes
- Intégration puissante avec un logiciel externe
- Modifier les commandes - les modifications entrent en vigueur immédiatement pour toutes les nouvelles commandes
- Système de facturation Hydra
- Options de personnalisation disponibles

- Différents types d'ordres peuvent être exécutés : ordres de réclamation, ordres de retour, ordres de clients, refacturation, etc.
- Les commandes terminées sont disponibles pour les rapports et les analyses
- L'assistant d'exécution d'Hydra est hautement configurable via des fichiers de configuration
- Intégration de la base de données SQL

Caractéristique du système Hydra :

1. Automatisation du flux d'exécution d'ordres

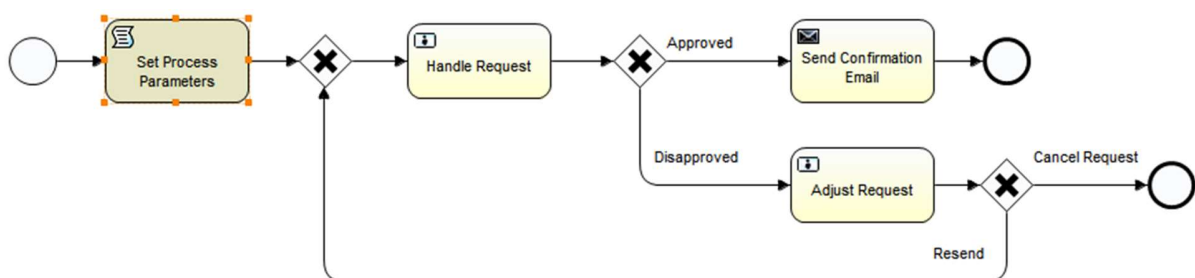
Hydra OMS utilise le moteur BPM qui effectue toutes les tâches commerciales. Evitant les commandes perdues, en retard et incorrectes. Modification de n'importe quel processus commercial à la volée - les modifications entrent en vigueur immédiatement pour toutes les nouvelles commandes.

2. Création de modèles de processus métier puissants

Conception de processus métier avec l'éditeur visuel compatible BPMN 2.0 normalisé ISO. Configuration des notifications d'escalade SLA, les branches et les conditions pour le flux de processus et intégration d'autres logiciels via l'API.

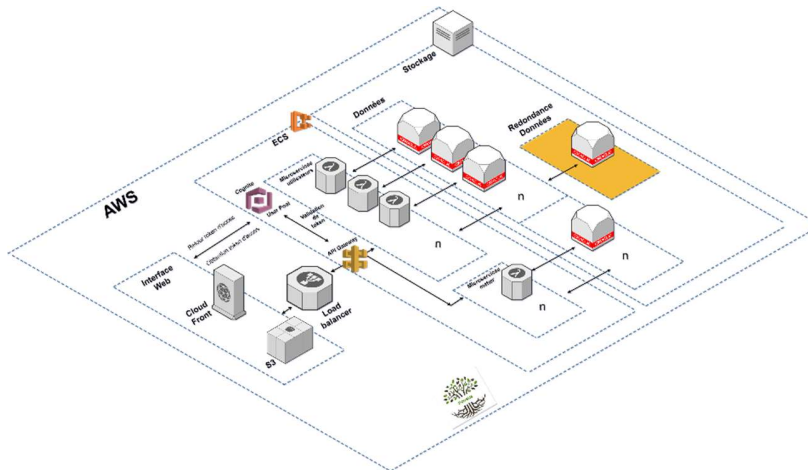
3. Fournir aux employés un assistant d'exécution de commande convivial

La fonctionnalité la plus douce d'Hydra est l'assistant d'exécution de processus configurable. Les employés Foosus n'auront pas besoin de se souvenir de tous les schémas de processus. Ils lancent simplement l'assistant, saisissent les données nécessaires (avec des suggestions de contexte configurables) et appuient sur le bouton Suivant.



Création d'une tâche automatisé sous Hydra OMS

7. Diagramme de la nouvelle architecture



8. Plan de communication

Canaux

1. Outils de communications existant dans Foosus
2. Trello
3. Courrier électronique
4. Slack
5. Répertoire de documentation
6. GitHub

Formats

1. Ecrit
2. Diagramme
3. Images

Contenu

Le contenu des communication sera principalement les changement, avances et évolution du plan de travail et document faisant partis du projet.



Rythme de communication

La documentation devra être constante et établie de façon discipliné, chaque semaine une réunion aura lieu afin de connaître l'état du projet, incluant les avances, obstacles et questions qui puissent survenir du côté technique et du métier.

9. Critères d'acceptation et procédures

Métriques et KPIs de l'État Cible de l'Architecture

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

Indicateurs de réussite	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.

Autres métriques business peuvent être prise en compte afin de réussir une processus business complet (liste non exhaustive) :

Métrique	Détails
Chiffre d'affaires	<u>Méthode de calcul :</u> Le chiffre d'affaires est calculé en additionnant tous les revenus des achats des clients, moins le coût associé aux produits retournés ou non livrables.
La marge bénéficiaire nette	La marge bénéficiaire nette est un bon moyen de prédire la croissance à long terme de l'entreprise et de voir si les revenus dépassent les coûts de gestion de l'entreprise. <u>Méthode de calcul :</u> Calculer les revenus mensuels et réduire toutes les dépenses de vente.
	Amélioration des processus et de la production



La marge brute	<p><u>Méthode de calcul :</u> <i>Marge brute = (total des revenus des ventes - coût des marchandises vendues) / total des revenus des ventes</i></p>
Croissance des ventes depuis le début de l'année	<p>Il est utile de surveiller la croissance des ventes sur différentes périodes : des mesures mensuelles, annuelles et à long terme permettront de mieux comprendre la situation de l'entreprise.</p>
Coût d'acquisition de clients	<p><u>Méthode de calcul :</u> Multiplier la valeur moyenne d'une vente par le nombre de transactions répétées et le temps de rétention moyen en mois pour un client type.</p>
Fidélisation et rétention de la clientèle	<p>Le taux de rétention indique le nombre de clients qui continuent à utiliser le produit sur une période prolongée et effectuent des achats répétés.</p> <p><u>Méthode de calcul :</u> <i>Taux de rétention = ((CE-CN)/CS)) X 100</i> <i>CE = nombre de clients au bout d'une certaine période (1 an, par exemple)</i> <i>CN = nombre de nouveaux clients acquis au cours de la même période</i> <i>CS = nombre de clients au début de la période</i></p>
Trafic mensuel de la plateforme Foosus	<p>Plus les gens entendent parler de la plateforme et des produits, plus ils sont susceptibles de consulter la page Web.</p> <p><u>Méthode de calcul :</u> <i>Utiliser un outil de marketing gratuit tel que Google Analytics pour suivre le trafic mensuel du site Web ainsi que les sources de trafic, afin de comprendre comment les gens trouvent le site et l'expérience.</i></p>
Bonheur des employés	<p>Afin d'avoir des services de qualité, les employés doivent être content pour être efficient.</p> <p><u>Méthode de calcul :</u> <i>Mener des enquêtes d'équipe ou utiliser un outil RH pour recueillir des commentaires rapides sur le travail d'équipe et les niveaux de satisfaction personnelle.</i></p>



Procédure d'acceptation

L'acceptation de la nouvelle plateforme :

1. Lister les besoins métier
2. Offres et propositions technique
3. Validation de l'offre technique par les directions
4. Validation des tests
5. Validation de la migration
6. Validation de l'implémentation
7. Analyse des besoins
8. GoLive

10. Procédures de changement de périmètre

1. Etudier les changement à réaliser
2. Définir la nouvelle architecture
3. Connaître les éléments existant afin de réaliser la migration des données.
4. Assurer la stabilité de la nouvelle implémentation en réalisant des test avant production.
5. Echanger avec les parties prenantes afin de connaître et signaler de défaillance.
6. Corriger les défaillance.
7. Alimenter la documentation avec les résultats de la mise en production et futurs changements.

Répertoire GitHub : <https://github.com/nikMilot/NouveauxSystemesFoosus>



11. Phases de livrables définies

Phase	Temps	Commentaires
Definition d'architecture	6 mois	Ces deux étapes sont faites en même temps
Création des livrables	6 mois	
Développement	5 mois	Le développement inclus la formation et l'acquisition de connaissance pour l'utilisation de la nouvelle plateforme. Le développement pourra également être fait en continu
Migration	5 mois	
Production	Temps total et approximatif du projet : 16 mois	

12. Personnes approuvant ce plan

Valideur	Domaine de responsabilité	Date
Direction Generale	DG	13-10-21
Natasha Jarson	CIO	13-10-21