

Déclaration de Travail d'architecture

Projet : Foosus, Conception d'une nouvelle architecture



Date : 14/05/2025

Document réaliser par : Eric Gigondan

Information sur le document

<i>Nom du projet</i>	Foosus : Conception d'une nouvelle architecture
<i>Préparé par :</i>	Eric gigondan
<i>N° de version du document :</i>	1.0
<i>Titre :</i>	<i>Déclaration de travail d'architecture</i>
<i>Date de version du document :</i>	14/05/2025
<i>Types d'action :</i>	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion

Objectif de ce document

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

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

Table des matières

Information sur le document	2
Objectif de ce document	2
Déclaration de travail d'architecture	4
Requête du projet et contexte	4
Description du projet et périmètre	4
Vue d'ensemble	5
Alignement stratégique	6
Rôles et responsabilités	7
Approche architecturale.....	8
Process d'architecture.....	8
Choix de l'architecture microservice face au monolithique	10
Contenu de l'architecture	11
Cadre de contenu d'architecture	11
Méthodologies pertinentes et norme de l'industrie	12
Méthode Lean	12
Architecture itérative	14
Déploiement as code	14
Architecture Cible.....	15
Sécurité.....	15
Diagramme de cas d'utilisation	15
Diagramme de déploiement	16
Pile technologique.....	17
Diagramme de donnée.....	18
Diagramme de composant	19
Critère d'acceptation	20
Métrique et KPIs	20
Procédure d'acceptation	20
Approbations signées	21

Déclaration de travail d'architecture

Requête du projet et contexte

Foosus est une start-up âgée de 3 ans, dans le secteur de l'alimentation durable. Leur objectif est de soutenir l'alimentation locale et de mettre les consommateurs en contact avec des producteurs et des artisans locaux.

Après plusieurs années de développement, le système d'information de Foosus présente une trop grande complexité. Il n'évolue plus au rythme de l'activité, risquant d'entraver la croissance de la start-up.

Le projet Foosus a pour but de concevoir une nouvelle architecture se présentant comme la création d'une nouvelle plateforme e-commerce géo-ciblée.

L'entreprise a besoin de frontières claires pour pouvoir développer une plateforme qui permette de l'innovation rapide et de se mettre à l'échelle.

Description du projet et périmètre

Le projet a pour but de créer une plateforme e-commerce polyvalente pour faire évoluer l'entreprise, en misant sur l'efficacité, la flexibilité et une prise de décision cohérente afin de rivaliser avec les leaders mondiaux dans le domaine de l'alimentation durable.

Objectifs :

- Exploiter la géolocalisation pour connecter fournisseurs de proximité et consommateurs, en mettant en avant les produits locaux.
- Concevoir une architecture évolutive, déployable dans plusieurs villes et pays.
- Garantir l'accessibilité de la plateforme sur mobile et ordinateur, en tenant compte des contraintes de bande passante pour les réseaux cellulaires.
- Catégorie de compte différents avec des fonctionnalités qui leur sont propres

Vue d'ensemble

La géolocalisation des consommateurs et des fournisseurs est une des fonctionnalités clé de la nouvelle architecture de départ. D'autres pourront être ajouter par la suite mais ne sont pas prioritaire. Parmi celle-ci, la modification des stocks et produits, le suivi des commandes, le système de promotion.

Ce schéma offre une vision plus long terme de l'application, qui est amené à évoluer tout en donnant une direction à Foosus.

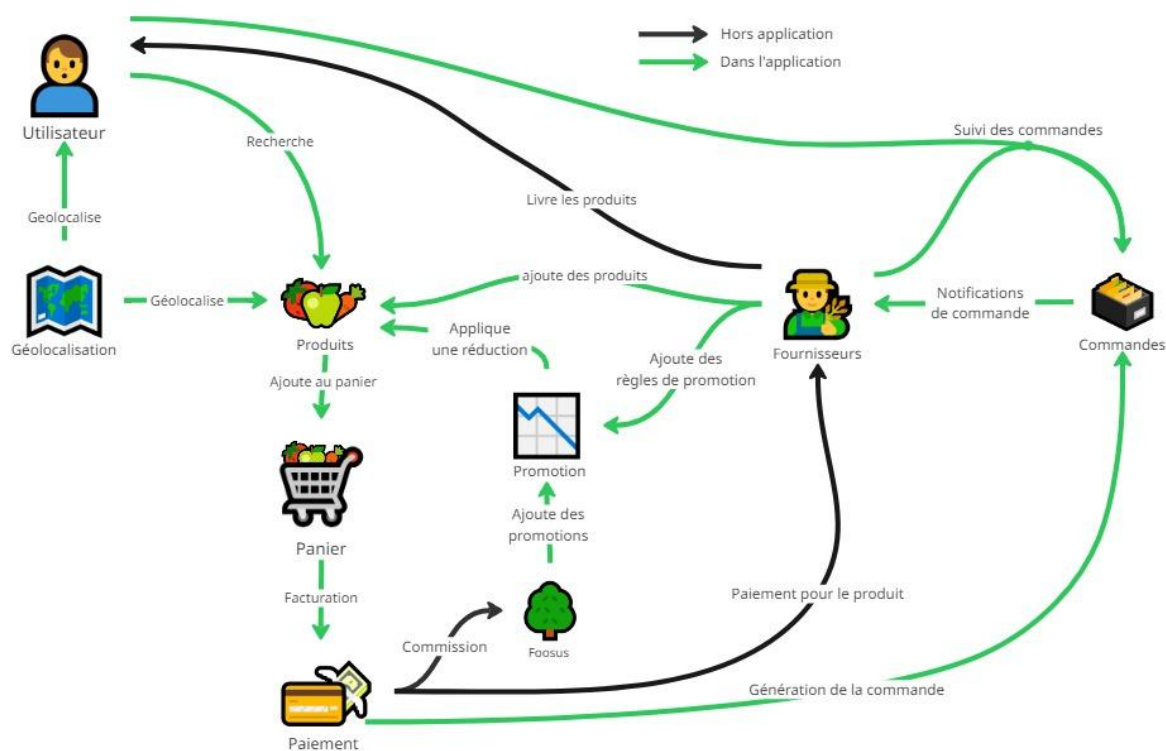


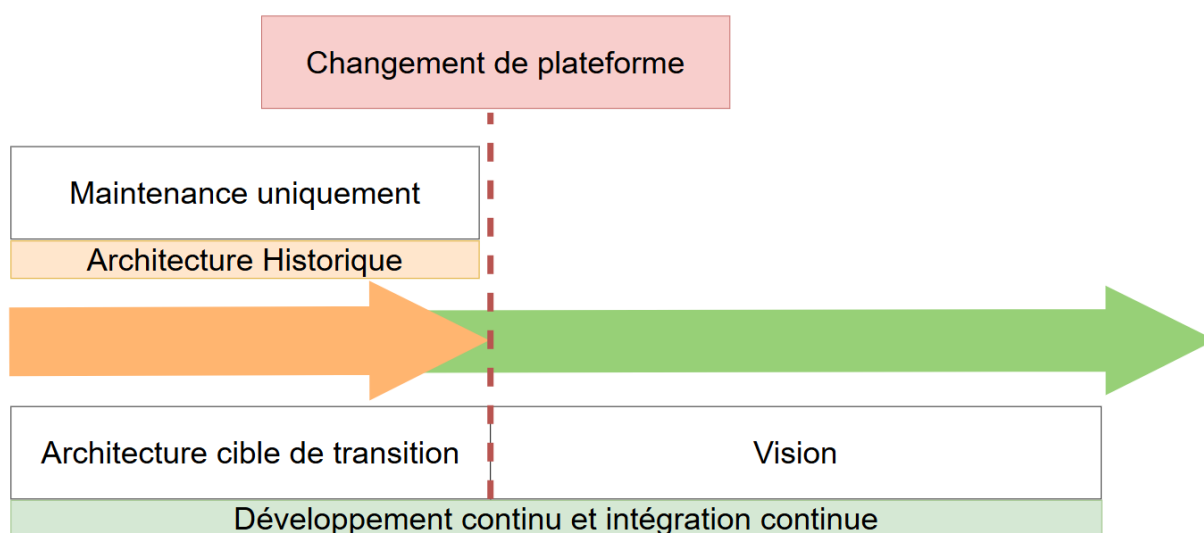
Figure 1 Vision de la plateforme e-commerce Foosus

Alignement stratégique

Le projet est contraint par des enjeux de budget et de délai. Il est donc pertinent d'adopter une vision d'architecture évolutive, s'appuyant sur une modernisation progressive de l'existant. Cette nouvelle architecture coexistera temporairement avec la plateforme actuelle, le temps de construire une architecture cible de transition.

Cette architecture transition, se présente comme une évolution de l'architecture historique déployée dans le cloud, avec l'ajout des fonctionnalités répondant aux exigences de Foosus via des micro-services.

Une fois cette étape atteinte, elle pourra remplacer l'ancienne plateforme et continuer à évoluer de manière incrémentale.



Dans l'attente de l'architecture de transition, l'application actuelle sera conteneurisée afin de simplifier son déploiement, sa maintenance et son portage sur les environnements cibles. Cette approche permet d'encapsuler les dépendances et de garantir un comportement homogène entre les différents environnements (développement, test, production). Elle facilite également l'intégration progressive des nouveaux microservices au sein d'une infrastructure orchestrée.

La vision d'une architecture évolutive repose sur l'adaptation continue de l'architecture existante. Cela inclut l'extraction, la réécriture, l'amélioration et le redéploiement de ses différents composants, en les transformant en micro-services standardisés, simples à maintenir, tout en permettant le développement constant de nouvelles fonctionnalités.

Cela implique la mise en place de processus et d'une approche architecturale sur mesure, mais suffisamment souple pour s'adapter à l'organisation des équipes et à la structure de Foosus. Le projet vise ainsi, entre autres, à définir un cadre architectural et des contraintes permettant d'itérer rapidement vers les objectifs métiers de Foosus.

Rôles et responsabilités



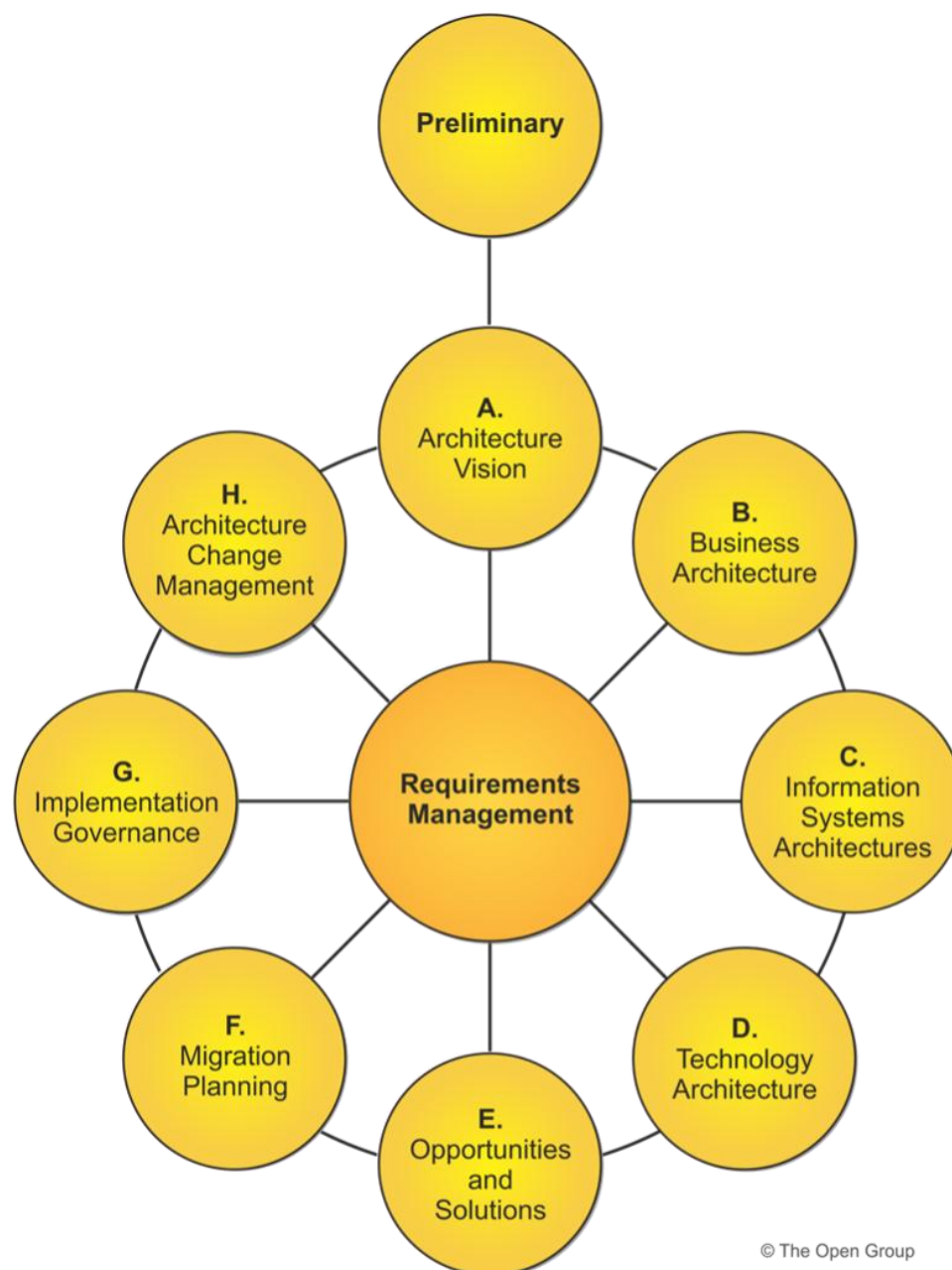
Le tableau suivant montre les rôles et responsabilités des parties prenantes clé en utilisant ce modèle.

Activités	Approbation du projet	Définitions des objectifs	Définition du budget	Définition de l'architecture	Identification et évaluation des risques	Développement
Ash Callum CEO	A	A	A	I	I	I
Natasha Jarson CIO	R	R	C	A	A	A
Daniel Anthony CPO	C	C	C	I	I	I
Christina Orgega CMO	C	C	C	I	I	I
Jo Kumar CFO	C	C	R	I	I	I
Pete Parker Engineering owner	C	C	C	C	C	R
Jack Harkner Operation Lead	C	C	C	C	C	R
Eric Gigondan Entreprise Architecture Owner	I	I	I	R	R	C

miro

Approche architecturale

Process d'architecture



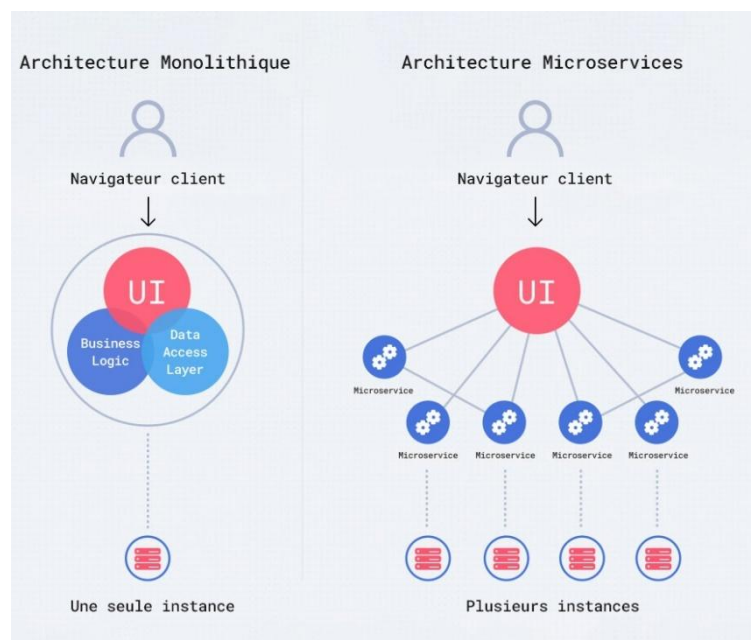
Le développement de l'architecture selon la méthode **TOGAF** s'appuie sur une méthodologie fondée sur les meilleures pratiques en matière d'architecture. Le tableau ci-dessous présente l'application de l'ADM dans le cadre de ce projet spécifique.

Phase	Sortie	Notes
Préliminaire (cadrage)	Définition des objectifs stratégiques, des contraintes, du cadre budgétaire et du plan stratégique.	Cette phase a pour but de préparer Foosus à la réalisation des travaux d'architecture. Il s'agit de mettre la startup en capacité de maîtriser la gestion et les transformations de son architecture
A - Vision de l'architecture (cadrage)	Vision de l'architecture avec une représentation générale des architectures initiales et cibles.	Cette phase a pour but de s'assurer de la compréhension commune de toutes les parties prenantes, afin d'obtenir un consensus sur les orientations et les résultats attendus.
B - Architecture business	Architecture métier avec une formalisation des éléments métier (exigences, processus et entités).	Ces trois phases (B, C et D) ont pour but de détailler l'architecture initiale et cible, de mesurer l'écart entre les deux puis d'évaluer l'impact des évolutions sur l'ensemble des facettes de la start-up.
C - Architecture des systèmes d'information	Architecture du système avec une définition des composants logiciels (applicatifs et données) qui supportent l'automatisation et l'exécution des capacités métiers.	
D - Architecture technologique	Architecture technique avec description des plateformes et environnement d'exécution sur lesquels vont s'appuyer les applications et les sources de données.	
E - Opportunités et solutions	Consolidation des résultats des phases B, C et D en termes d'architectures, d'exigences et d'écarts	Ces deux phases (E et F) traitent du planning et de l'organisation de la mise en œuvre de la nouvelle architecture
F - Planning de migration	Planning de migration avec détails sur la constitution des projets de mise en œuvre, leur organisation, les objectifs et les coûts.	
G - Gouvernance de l'implémentation	Gouvernance de l'implémentation avec les contrats relatifs aux projets d'implémentation incluant les recommandations architecturales.	Cette phase établit la version définitive des contrats d'architecture.
H - Management du changement d'architecture	Gestion de la maintenance et des évolutions avec la gestion des modifications, qui comprend l'évaluation des demandes de changements impactant l'architecture.	Cette phase prend en charge la gestion de l'architecture déployée
Management des conditions requises	Référentiel des exigences avec l'identification et la modification des exigences aux cours des différentes phases, ainsi que l'analyse et l'évaluation des impacts de celles-ci.	La gestion des exigences est une activité de rationalisation, de hiérarchisation et de suivi qui porte sur l'ensemble des exigences produites ou modifiées aux cours de chaque phase

Choix de l'architecture microservice face au monolithique

Dans le cadre du projet de refonte de la plateforme Foosus, le choix d'une architecture orientée microservices s'est imposé comme une réponse cohérente aux enjeux de modularité, de scalabilité et d'innovation rapide. Contrairement à une architecture monolithique, qui centralise l'ensemble des fonctionnalités dans un seul bloc applicatif, une approche par microservices permet de structurer le système en services indépendants, chacun étant responsable d'un domaine fonctionnel précis. Ce découplage logique favorise une meilleure isolation des responsabilités, une évolutivité granulaire et une capacité à déployer des fonctionnalités de manière incrémentale, sans impacter l'ensemble du système.

Cette orientation répond directement aux besoins spécifiques de Foosus, dont la plateforme vise à connecter en temps réel producteurs et consommateurs à l'échelle locale, dans un contexte géo ciblé et en croissance continue. Une architecture monolithique, bien que plus simple à mettre en œuvre initialement, aurait limité la capacité de l'entreprise à faire évoluer indépendamment les composants critiques tels que la géolocalisation, la gestion des utilisateurs ou encore les intégrations avec des services tiers. Elle aurait également introduit un risque accru de dette technique, en complexifiant la maintenance et les déploiements à mesure que la plateforme se développe.



À l'inverse, l'approche microservices permet non seulement de distribuer la charge de manière plus efficace, mais aussi d'aligner les équipes de développement sur des domaines fonctionnels autonomes, renforçant ainsi l'agilité organisationnelle. Ce modèle permet de mieux maîtriser les impacts des changements, d'adopter une stratégie de montée en charge progressive, et de faciliter la mise en œuvre de pipelines CI/CD pour automatiser les tests, les déploiements et le monitoring. Enfin, il s'inscrit pleinement dans les principes de modularité, de résilience et d'innovation continue qui sont au cœur de la vision stratégique de Foosus.

Contenu de l'architecture

Cadre de contenu d'architecture

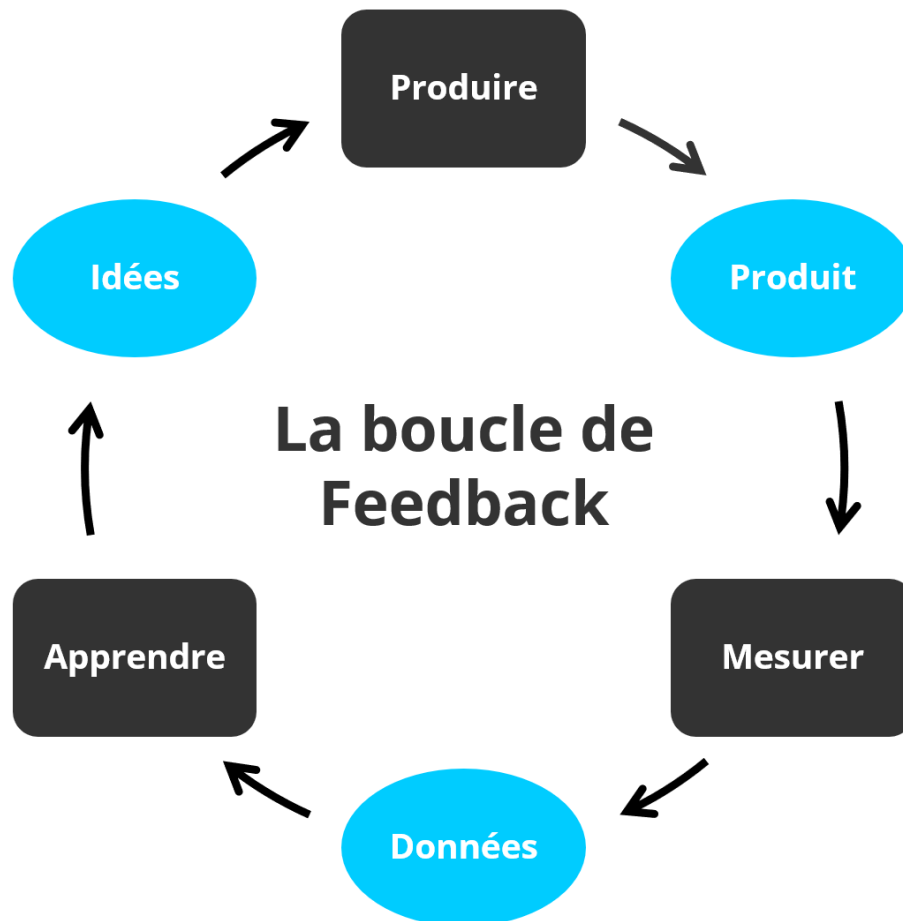
Le cadre de contenu d'architecture TOGAF (ou ACF pour « Architecture Content Framework ») fournit une catégorisation des meilleures pratiques pour le contenu de l'architecture. Néanmoins, tous les éléments ne sont pas également pertinents pour chaque projet. Le tableau ci-dessous décrit les zones de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

Phase	Entrée	Sortie
Preliminaire	<ul style="list-style-type: none"> Architecture historique. Besoin métier <u>Git actuel</u> 	<ul style="list-style-type: none"> Rôles et responsabilités Requête du projet Objectif
A - Vision de l'architecture (Cadrage)	<ul style="list-style-type: none"> Architecture historique. Demandes métiers Objectifs stratégiques Parties prenantes identifiées 	<ul style="list-style-type: none"> Demande de travaux d'architecture (ce document) Vue d'ensemble Alignement stratégique Contrat d'architecture métier Critère d'acceptation
B - Architecture business	<ul style="list-style-type: none"> Vision métier Besoins utilisateurs 	<ul style="list-style-type: none"> Diagramme de cas d'utilisation BFF (Backend For Frontend)
C - Architecture des systèmes d'information	<ul style="list-style-type: none"> Modèles métier Exigences fonctionnelles et non fonctionnelles 	<ul style="list-style-type: none"> Diagramme de composant Diagramme de donnée Contrat d'Architecture Développement et Design Pile technologique cible
D - Architecture technologique	<ul style="list-style-type: none"> Architecture SI Contraintes technologiques 	<ul style="list-style-type: none"> Diagramme de déploiement Déploiement As Code Principes de sécurité Choix microservice
G - Gouvernance de l'implémentation	<ul style="list-style-type: none"> Critères de validation Dossier d'architecture 	<ul style="list-style-type: none"> Procédure d'acceptation Approbation signée KPIs

Méthodologies pertinentes et norme de l'industrie

Méthode Lean

Foosus suit une méthode agile, combinant Lean et Kanban, centrée sur l'amélioration continue.



Les décisions d'architecture chez Foosus sont responsables et intègrent une démarche « Green IT ». Plusieurs principes viennent structurer cette approche :

Principes généraux :

- Décisions guidées par le retour d'expérience.
- Priorité au long terme et à l'apprentissage.
- Accepter les erreurs et concevoir pour échouer vite et s'améliorer.

Principes métier :

- Favoriser l'innovation et l'agilité grâce à l'extensibilité.
- Garantir la stabilité pour protéger l'image de marque.

Principes liés aux données :

- Concevoir sans dépendre d'une vision complète.
- Protéger les données personnelles.
- Adapter l'accès, la mutabilité et la cohérence au besoin métier.
- Représenter le domaine métier dans un contexte bien délimité.

Principes applicatifs :

- Applications à responsabilité unique et faiblement couplées.
- Interfaces ouvertes, extensibles, et pilotées par contrat.
- Éviter les dépendances cycliques.

Principes technologiques :

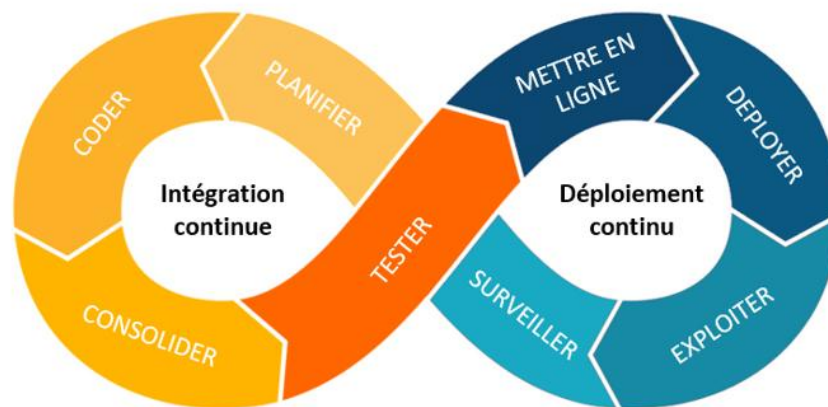
- Choix ouverts, adaptables et cohérents avec les besoins métier.
- Sorties logicielles fréquentes.
- Architecture traçable, prévisible et répétable.

Approche globale :

- Le projet s'étend sur 6 mois avec un prototype en ligne de mire.
- L'architecture est décomposée (stratégique, organisationnelle)
- Usage de composants sur mesure ou du marché, avec des micro-services pour un couplage faible.
- Une architecture de transition concrète est définie, avant d'atteindre une vision plus abstraite de l'architecture évolutive.
- La nouvelle plateforme montera progressivement en charge, l'actuelle étant seulement maintenue.

Architecture itérative

L'approche itérative de l'architecture adoptée dans le projet Foosus consiste à construire l'architecture cible par étapes successives, avec des validations régulières auprès des parties prenantes et des ajustements progressifs. Cette méthode permet de répondre aux contraintes de budget, de délai.



Au fil du temps, des composants de l'ancienne architecture sont extraits, réécrits, puis redéployés sous forme de microservices modulaires, en respectant des standards ouverts (REST, JSON, OAuth) et les principes de couplage faible.

Enfin, cette approche est renforcée par l'utilisation de méthodes Agile (Lean + Kanban), permettant un pilotage visuel du progrès, une priorisation continue des livrables, et une adaptation constante à l'évolution des besoins et des risques.

Déploiement as code

Le projet adopte une approche Infrastructure et Déploiement as Code, reposant sur des outils comme [Ansible](#), [Terraform](#), [Kubernetes](#) afin d'automatiser la création, la configuration et le déploiement des environnements techniques. Cette méthode garantit la répétabilité, la traçabilité et la standardisation des déploiements, tout en réduisant les erreurs humaines. Elle s'inscrit dans une stratégie DevOps alignée avec les pratiques de CI/CD, favorisant des mises en production fiables, rapides, contrôlées et surtout à moindre coût.

Architecture Cible

Sécurité

La plateforme intégrera des mécanismes de sécurité dès la conception. Les communications seront chiffrées (HTTPS / TLS), les données sensibles seront stockées de manière sécurisée et chiffrée. Des contrôles réguliers seront réalisés grâce à des tests automatisés et des outils d'analyse de code comme [Sonarqube](#) pour vérifier à chaque mise à jour les éventuelles failles de sécurité.

Diagramme de cas d'utilisation

Dans ce diagramme de cas, le client a accès après une authentification à une multitude de pages. De la modification de son compte au circuit complet pour une commande.

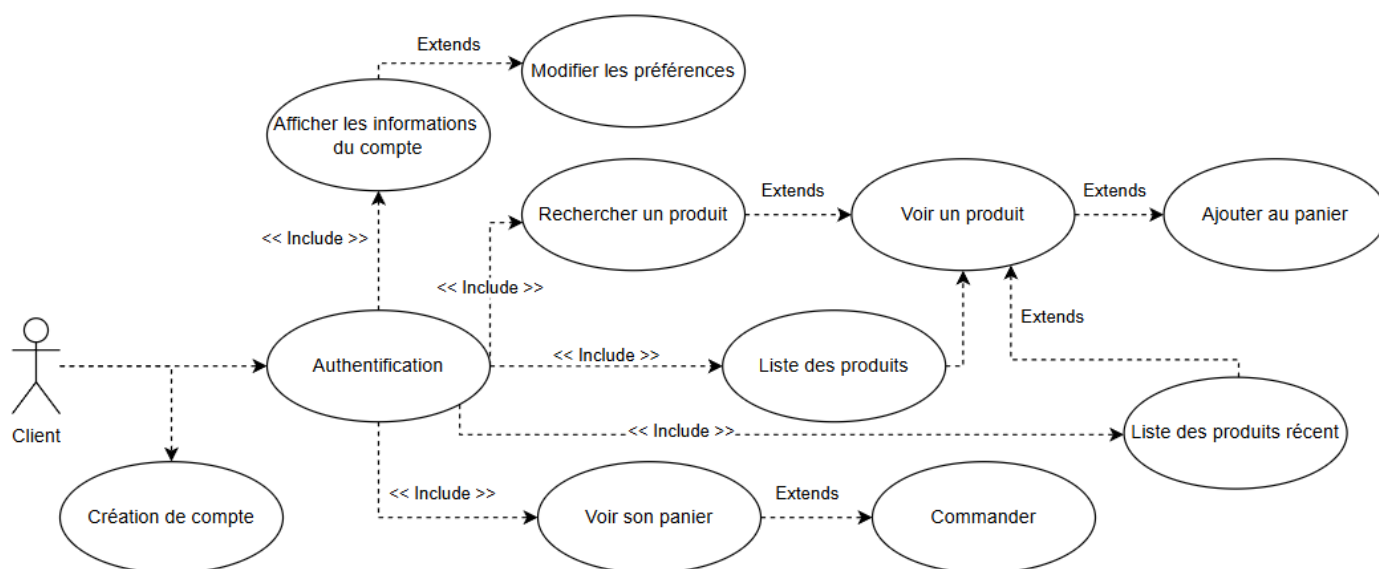
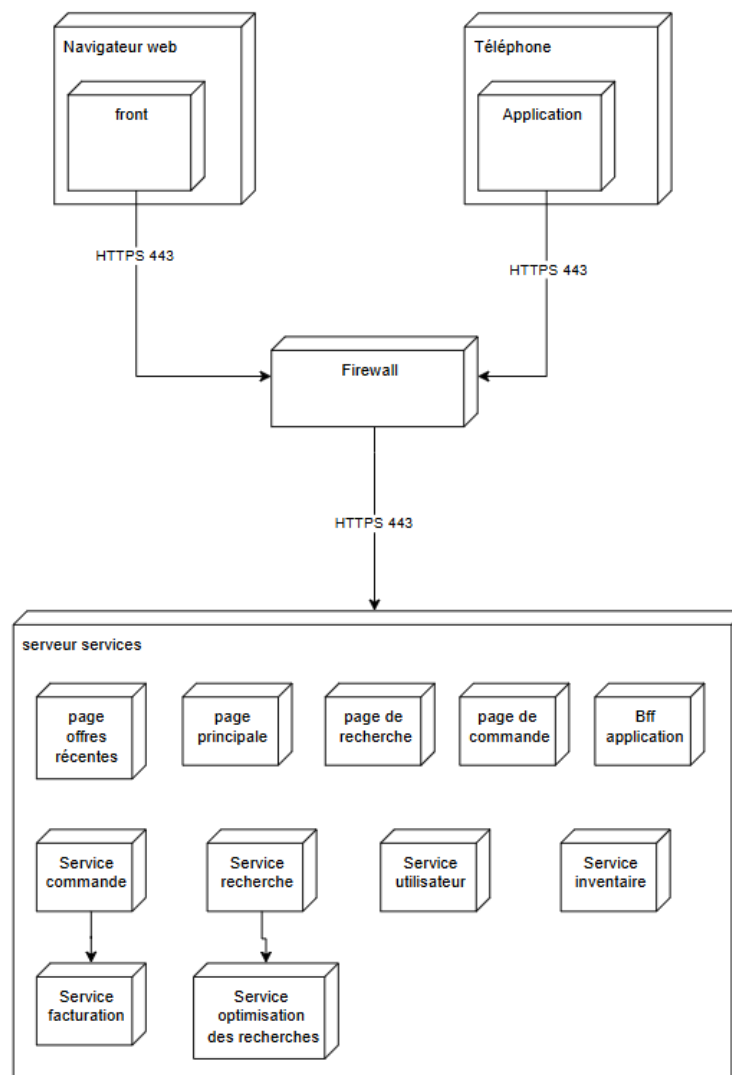


Diagramme de déploiement

Les différents services seront hébergés sur un serveur dédié. Dans le cloud ou sur une infrastructure privée.



Pile technologique

Dans cette architecture, la virtualisation dans des conteneur sera importante afin de réduire les coûts et permettre une évolution par ville / pays plus simplement.

Niveau 6 Frontend	Angular , ionic , html, css
Niveau 5 Protocole	HTTPS, REST,
Niveau 4 Backend	django / python / scala / play framwork
Niveau 3 Microservices	bff mobile, recherche, utilisateur, inventaire, commande, optimisation des recherche, facturation
Niveau 2 Orquestrateur	Kubernetes / terraform / ainsible
Niveau 1 Infrasturcture	Serveur virtuel, stockage cloud, serveur physique

Diagramme de donnée

Les différents services requièrent que les requêtes soit authentifié à l'aide d'un token JWT. La sécurité constitue un pilier fondamental de l'architecture cible de Foosus, en particulier pour la protection des données personnelles et de facturation. Afin de garantir la confidentialité, l'intégrité et la conformité réglementaire (notamment vis-à-vis du RGPD), toutes les données sensibles telles que les informations d'identification des utilisateurs et les données de paiement doivent être chiffrées au repos et en transit à l'aide de standards éprouvés (ex. AES-256, TLS 1.3).

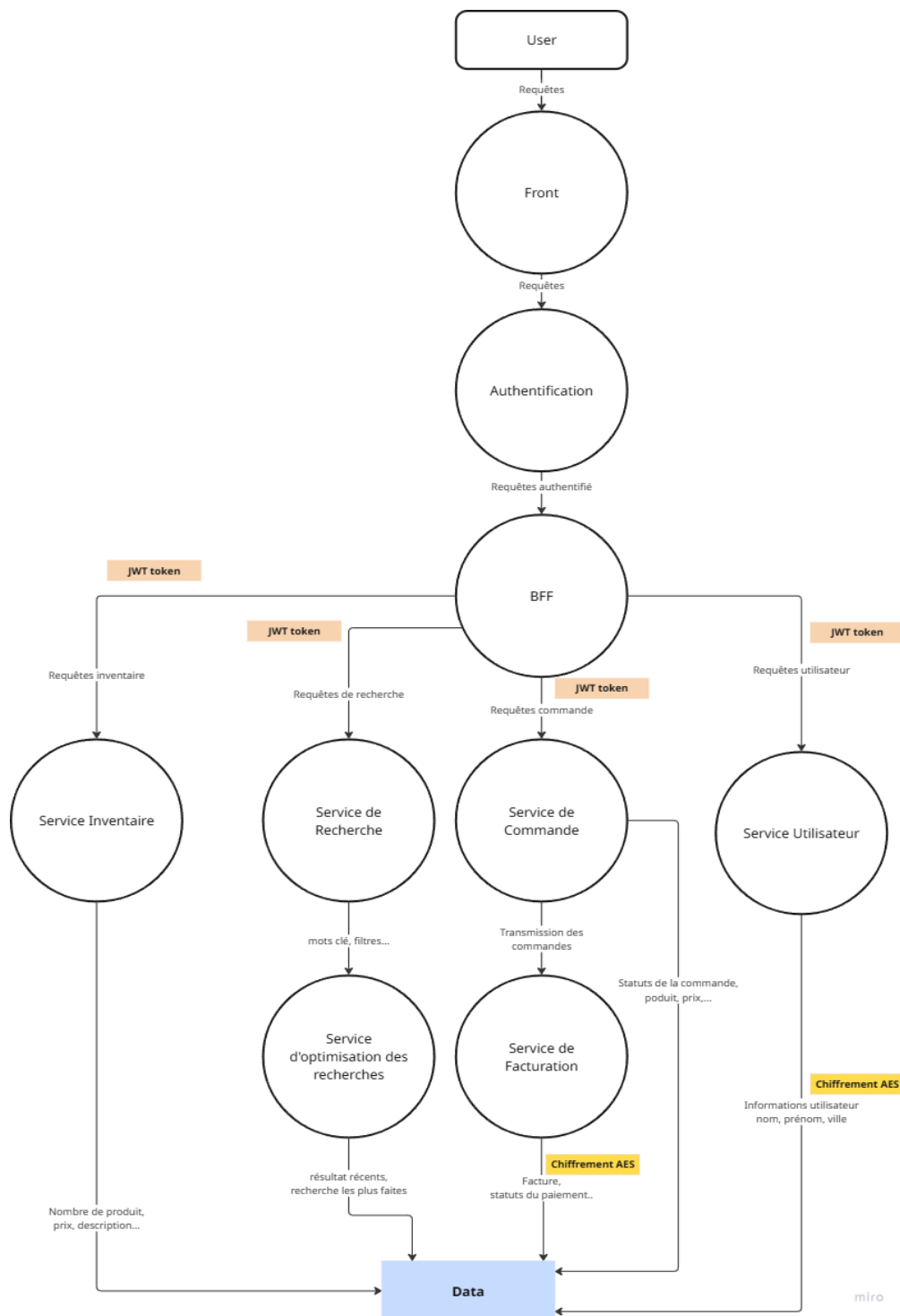
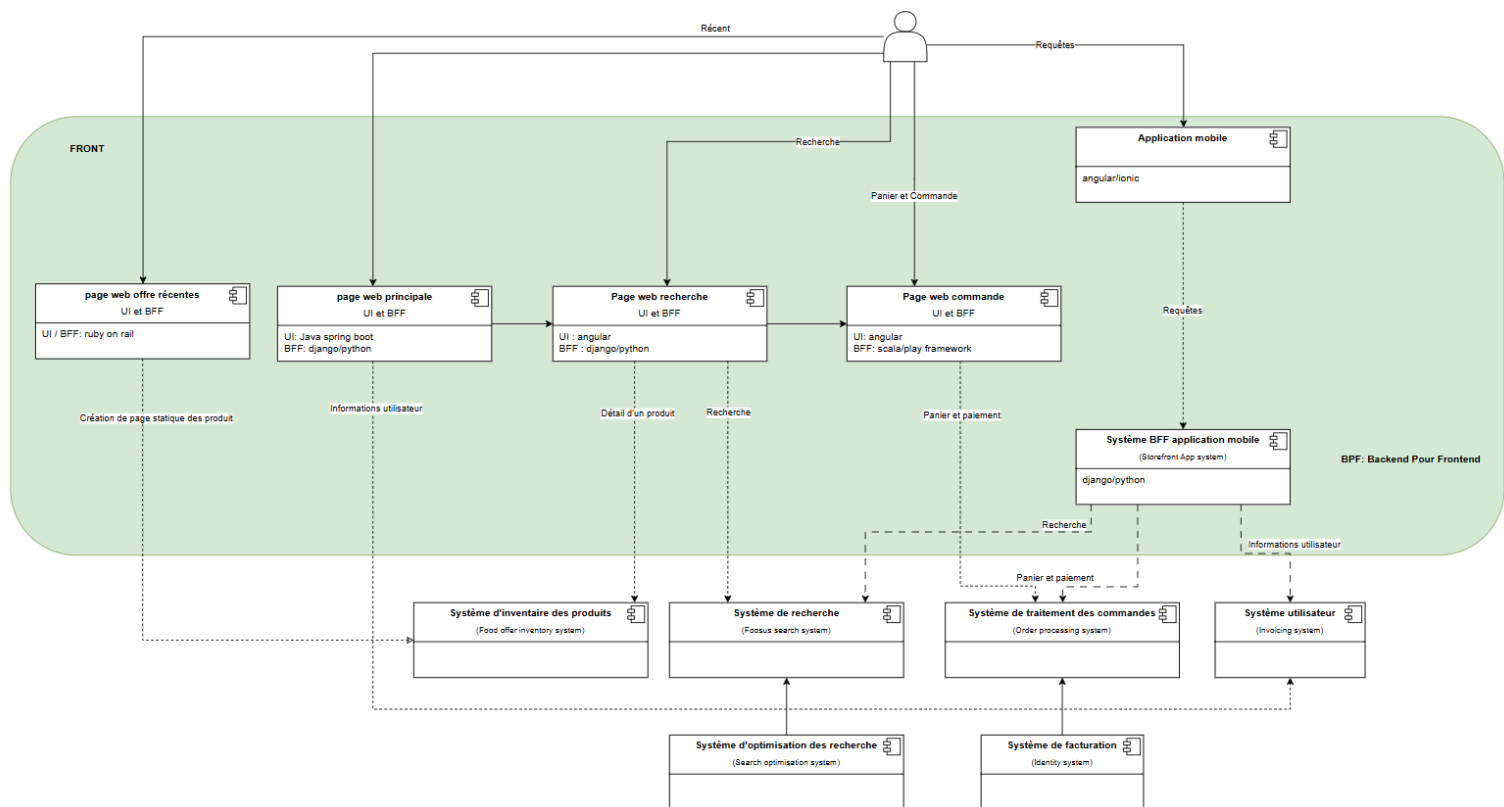


Diagramme de composant



L'adoption d'une architecture de type Backend for Frontend (BFF) s'impose comme une solution pertinente pour répondre à la diversité des usages et des profils utilisateurs.

Cette approche permet d'introduire une couche d'abstraction dédiée à chaque type de client, en tenant compte des contraintes propres aux terminaux utilisés (mobile, desktop) ainsi qu'aux rôles fonctionnels (producteurs, consommateurs). Elle facilite l'adaptation fine des données et des API en fonction des besoins spécifiques de chaque interface, ce qui contribue à réduire significativement la complexité côté client tout en optimisant les performances globales de l'application, notamment dans des environnements réseau hétérogènes.

Par ailleurs, le BFF joue un rôle clé dans la simplification de l'interaction avec les services métiers sous-jacents en masquant la complexité de l'architecture microservices, ce qui améliore la lisibilité, la maintenabilité et la cohérence fonctionnelle des échanges. Il offre également une indépendance forte entre les évolutions frontend et backend, permettant aux équipes de développement de livrer de nouvelles fonctionnalités plus rapidement, sans contrainte sur les couches basses. Enfin, cette couche intermédiaire constitue un point central pour l'implémentation des aspects transverses de sécurité, de conformité réglementaire (notamment **RGPD**), de gestion des erreurs et de cache, renforçant ainsi la robustesse, la traçabilité et la gouvernance de l'ensemble du système. En intégrant cette approche, l'architecture cible gagne en modularité, en agilité et en résilience, en parfaite adéquation avec les objectifs stratégiques et opérationnels de Foosus.

Critère d'acceptation

Métrique et KPIs

KPIs		
Indicateur	Technique de mesure	changement souhaité
Taux de conversion des utilisateurs	utilisation de Matomo	+10% après refonte
Nombre d'incidents critiques en prod	utilisation de datadog	< 1/mois
Taux de rétention (30 jours)	Mixpanel ou Amplitude	+15% à horizon 3 mois
Disponibilité de la plateforme	outil de monitoring (datadog, nagios ...)	99.99%
Scalabilité de l'infrastructure	outil de test (gatling, JMeter...)	> 10 000 requêtes/min <small>miro</small>

Sources : [Datadog](#) , [Nagios](#), [Gatling](#), [JMeter](#), [Mixpanel](#), [Amplitude](#), [Matomo](#)

Procédure d'acceptation

Chaque livrable produit dans le cadre du projet sera présenté oralement aux parties prenantes représentant les différents intérêts de Foosus. Cette présentation permettra d'expliquer le contenu du document, de répondre aux questions et de clarifier les points qui pourraient manquer de précision.

À l'issue de cette présentation, un exemplaire du livrable sera remis au comité de direction, avec une zone prévue pour la signature. Ce document devra être signé dans un délai raisonnable pour valider officiellement le livrable. En cas de refus, une motivation écrite devra être fournie afin que les ajustements nécessaires puissent être apportés avant une nouvelle soumission.

Approbations signées

Commentaire, date, nom, prénom, signature

