

Contrat d'Architecture avec les Fonctions Développement et Design



Conception d'une nouvelle architecture afin de soutenir le développement d'une entreprise	
Client	Foosus
Nom du projet	Projet OC-5-19042021
Préparé par	Mitra IZADI
N° de version du document	0.4
Titre	Contrat d'Architecture avec les Fonctions Développement et Design terminé
Date de version du document	19/05/2021
Revu par	Monsieur Eric LEGBA
Date de révision	23/05/2021
Liste de distribution	
De	
Date	
Email	mitraizaid65@gmail.com
Pour Action	Information
Date de rendu	23/10/2021
Types d'action	Approbation, Révision, Information, Classement, Action requise, Participation à une réunion, Autre (à spécifier)
Historique de versions du document	Voir git



1	Introduction et contexte.....	4
2	La nature de l'accord	5
3	Objectifs et périmètre	5
4	Description de l'architecture, principes stratégiques et conditions requises	5
4.1	Description du projet	5
4.2	Principes stratégiques	6
4.3	Conditions requises	7
4.3.1	Évoluer avec la base de clientèle de Foosus	7
4.3.2	Une plateforme sécurisée, utilisable et réactive	7
4.3.3	Une technologie transparente	7
4.3.4	Une évolutivité capable d'accompagner la croissance.....	7
4.3.5	Expérimentation	7
4.3.6	Pistes pour un approvisionnement alimentaire géographiquement responsable	7
4.3.6.1	Recherche de fournisseurs alimentaires.....	7
4.3.6.2	Tri des offres alimentaires	7
5	Livrables architecturaux.....	8
5.1	Développement de l'architecture	9
5.1.1	Domaine utilisateur	9
5.1.1.1	Processus fonctionnels	9
5.1.1.2	Description.....	9
5.1.2	Client	10
5.1.2.1	Processus fonctionnels	10
5.1.2.2	Description.....	10
5.1.3	Fournisseur	11
5.1.3.1	Processus fonctionnels	11
5.1.3.2	Description.....	11
5.2	Choix technologique	11
5.2.1	Avantages.....	12
5.2.2	Inconvénients	13
5.3	Azure Service Fabric.....	14
5.4	Livraison de l'architecture et métriques business.....	14

Indicateurs de réussite

Indicateur	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. 14
5.5 Phases de livraison définies		14
Principales étapes du processus métier		17
6 Plan de travail commun priorisé		18
La conception :		18
Le développement :		18
7 Plan de communication		19
8 Risques et facteurs de réduction.....		19
8.1 Structure de gouvernance.....		19
8.2 Analyse des risques		23
9 Hypothèses.....		23
10 Critères d'acceptation et procédures		23
11 Procédures de changement de périmètre		23
12 Calendrier.....		23
13 Phases de livrables définies		23
14 Personnes approuvant ce plan		24
15 Annexes.....		24

1 Introduction et contexte

L'objectif d'une architecture d'entreprise telle que proposée par TOGAF est d'optimiser les processus existants, souvent fragmentés, pour construire un environnement intégré, capable de répondre aux changements et de soutenir la stratégie de l'entreprise.

Le cœur de TOGAF est aujourd'hui la méthode de développement de l'architecture (ADM). Cette méthode couvre les 4 domaines de l'architecture :

1. le métier (business)

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

E-mail: contact@foosus.com - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24



2. les données
3. les applications
4. l'infrastructure technique

2 La nature de l'accord

L'Architecte logiciel fournit au CIO (Natasha Jarson) les documents correspondant à chaque phase de l'étude ainsi que toutes les informations utiles sur le déroulement de sa mission. Si le budget annoncé par le CIO est manifestement insuffisant pour la réalisation des travaux projetés, l'Architecte l'en informe sans délai.

Au cours des études, l'Architecte informe le CIO de toute évolution significative du budget prévisionnel de l'opération. Au cours des travaux, et sauf urgence liée à la sécurité des personnes et/ou des biens, toute décision entraînant un supplément de dépenses fait l'objet d'un accord du CIO.

(L'essentiel du contenu des trois phases suivantes B (métier), C (système d'information) et D (technique) consiste à détailler l'architecture cible et initiale, à mesurer l'écart entre les deux, puis à évaluer les impacts des évolutions sur l'ensemble des facettes de l'entreprise. La combinaison de ces éléments permet d'établir un premier scénario de la feuille de route de transition.)

3 Objectifs et périmètre

Voir « Objectif et périmètre » dans la Déclaration de Travail d'Architecture

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

4.1 Description du projet

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 notre croissance. Les études de marché et les analyses commerciales montrent que nos clients souhaitent acheter local et soutiennent les producteurs locaux.

Les concurrents n'ont pas ciblé cette niche. L'entreprise veut s'appuyer sur les connaissances acquises ces trois dernières années et créer une plateforme qui mettra en contact des consommateurs avec des producteurs et des artisans locaux dans toutes les catégories de besoins.

L'objectif est de mieux servir nos clients et d'innover de façon responsable, de manière à maximiser nos capacités et aider l'organisation à continuer à croître en accord avec notre feuille de route générale. L'entreprise n'attire plus de nouveaux utilisateurs, et il faut revenir à un niveau où l'entreprise peut innover rapidement. Il importe de repartir sur une nouvelle base pour l'initiative la plus récente, et d'éviter de répéter des choix d'architecture qui ne sont plus adaptés à une plateforme évolutif. L'incertitude est un facteur important du secteur et il faut assurer que l'entreprise a minimisé les risques de prise de décisions techniques difficiles à inverser!

Le précédent Responsable de l'Architecture, nourrissait une culture où les équipes de développement étaient encouragées à expérimenter et essayer librement de nouvelles approches techniques. Cela avait, jusqu'à récemment, résulté dans la construction d'une équipe de 15 développeurs qui aiment travailler ici et sont aussi investis dans la satisfaction des clients!

Cette culture impliquait beaucoup d'expérimentations divergentes, mais a donné lieu à peu de modèles ou d'idées réutilisables. Alors que les équipes ont atteint leurs premiers objectifs, l'application dispose aujourd'hui d'un ensemble hétérogène de technologies. Aujourd'hui, on demande de mettre en place une certaine standardisation pour la maintenance des développements futurs et à venir.

L'architecture et les systèmes informatiques ont permis de croître très rapidement soutenir la croissance et éviter de se mettre en travers de la route. Il y a un lien direct entre la capacité d'innovation et l'attrait que l'entreprise peut créer pour de nouvelles inscriptions.

4.2 Principes stratégiques

En réponse à un fort déclin des inscriptions utilisateurs, la direction de l'entreprise souhaite conserver la plateforme existante en mode maintenance et restructurer les équipes afin de livrer une plateforme à l'architecture travaillée, qui lui permette de grandir de manière alignée sur la vision business de soutien aux marchés *locaux* de l'entreprise. Les inscriptions constituent une métrique clé aux yeux des investisseurs et ne peuvent être améliorées que par l'agilité nécessaire pour innover rapidement et expérimenter avec des variantes d'offres produit existantes.

L'entreprise ne peut pas abandonner les outils actuels pendant qu'elle en élabore de nouveaux car cela impliquerait la mise hors service de la plateforme existante. Pour pouvoir continuer à accepter de nouvelles adhésions de fournisseurs et de consommateurs, Foosus doit en outre dissocier les nouvelles livraisons de l'architecture et de l'infrastructure existantes afin de limiter les interruptions de service.

Le but est de libérer la créativité et l'expérience des équipes techniques. Elle veut leur permettre de donner le meilleur d'elles-mêmes en créant une nouvelle plateforme qui pourra faire franchir le prochain million d'utilisateurs inscrits à la base de clientèle. Foosus veut également impulser des campagnes de marketing dans plusieurs grandes villes en étant

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

E-mail: contact@foosus.com - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24

sûrs que la plateforme restera utilisable et réactive, tout en offrant une expérience utilisateur de premier plan.

Il y a notamment plusieurs objectifs généraux qui doivent être satisfaits quelle que soit la nouvelle direction technique adoptée pour améliorer la capacité opérationnelle. La nouvelle plateforme devra également permettre à nos équipes produites d'innover rapidement en réorientant des solutions existantes, en expérimentant de nouvelles modifications et en facilitant l'intégration avec des partenaires internes et externes.

4.3 Conditions requises

4.3.1 Évoluer avec la base de clientèle de Foosus

La pile technologique doit être conçue de façon à évoluer naturellement au même rythme que notre base de clientèle.

4.3.2 Une plateforme sécurisée, utilisable et réactive

Eviter tout risque pour l'image de marque de l'entreprise et une approche qui garantisse la sécurité chaque fois que la plateforme s'élargie.

4.3.3 Une technologie transparente

Chaque nouvelle version doit être de taille réduite, présenter peu de risques, être transparente pour les utilisateurs et rester accessible en tout lieu et à tout moment.

4.3.4 Une évolutivité capable d'accompagner la croissance

Comblent le fossé entre le moment où une ligne de code est écrite et celui où elle est validée dans un environnement intégré.

4.3.5 Expérimentation

Avoir la visibilité sur la façon dont les logiciels sont utilisés et pouvoir inverser des décisions d'architecture tant que cela reste peu onéreux

4.3.6 Pistes pour un approvisionnement alimentaire géographiquement responsable

4.3.6.1 Recherche de fournisseurs alimentaires

- I. Emplacement des offres alimentaires proposées par les fournisseurs
- II. Proximité de l'utilisateur effectuant la recherche en cours
- III. Visualisation des informations statistiques secondaires et sectorielles relatives au produit alimentaire concerné. Par exemple, détails sur son indice glycémique

4.3.6.2 Tri des offres alimentaires

- I. Recherche et identification des produits alimentaires requis.
- II. Ajout des offres alimentaires au panier.
- III. Recherche d'un accord pour payer à la livraison.
- IV. Instructions de livraison et facture de la commission par e-mail au fournisseur alimentaire.

5 Livrables architecturaux

Voici le processus de vendre des produits dans le logiciel Foosus:

1. Le fournisseur partage les informations nécessaires à la vente de ces produits : Photos, descriptions etc. Ces informations sont reprises sur le site Foosus sous forme de fiche produit.
2. Les produits sont validés/approuvés par les administrateurs Foosus.
3. Le producteur paie les frais d'inscription.
4. Un accord de partenariat Foosus-producteur est conclu
5. Le client arrive sur le site et trouve le produit qu'il recherchait sur le site et le commande.
6. Il passe la commande sur le site. Foosus transmet les informations de sa commande au fournisseur.
7. A la réception des informations, le fournisseur expédiera directement le colis au client sans avoir à passer par Foosus.

Côté Foosus :

Le logiciel reçoit une commande Client qui génère à son tour une commande Fournisseur.

La commande est ensuite livrée au client. Le bon de livraison Fournisseur est identique au bon de livraison Client.

Pour le Fournisseur, la gestion du logiciel implique de pouvoir spécifier une adresse de livraison du Client Final, différente de celle du Revendeur, lors d'une commande client.

Côté fournisseur :

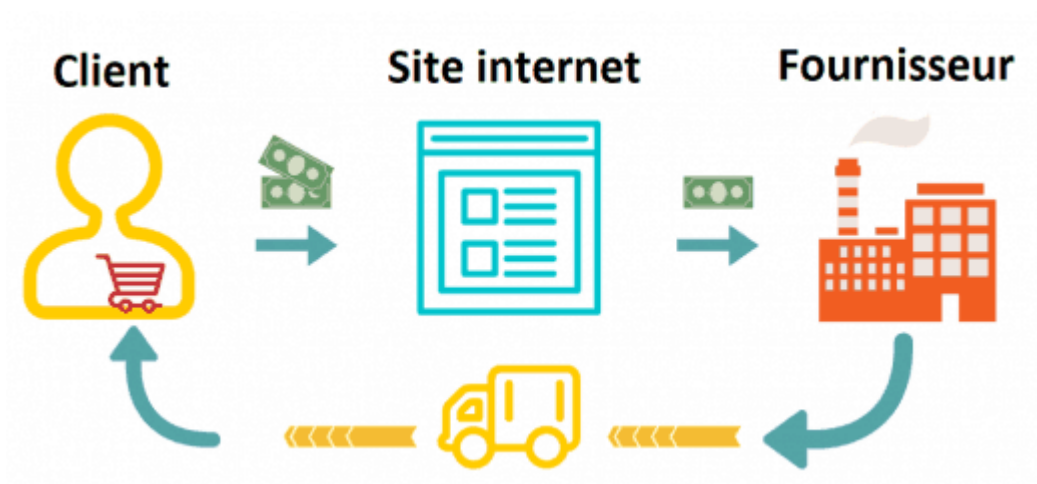
Le Fournisseur reçoit une commande Client qui, après livraison, génère un bon de livraison Client.

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

E-mail: contact@foosus.com - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24

Pour le Revendeur, la gestion du logiciel implique de pouvoir générer les commandes fournisseurs associées aux produits à expédier, tout en lui précisant l'adresse de livraison souhaitée.



5.1 Développement de l'architecture

5.1.1 Domaine utilisateur

5.1.1.1 Processus fonctionnels

1. Inscription
2. Authentification

5.1.1.2 Description

Avec une solution de session distribuée, les informations d'authentification utilisateur sont stockées dans un magasin de données partagé, souvent un simple dictionnaire distribué dont la clé est la session utilisateur. Lorsqu'un utilisateur accède à un microservice, les données utilisateur peuvent alors être extraites du magasin de données. Un autre avantage de cette solution est que l'état de connexion de l'utilisateur est opaque. Dans le cadre de l'utilisation d'une base de données distribuée, il s'agit également d'une solution hautement disponible et évolutive. Les inconvénients comprennent le fait que le magasin de données doit être protégé et donc seulement accessible via une connexion sécurisée et que la mise en œuvre de la solution présente souvent une complexité assez élevée.

En utilisant des jetons côté client, l'utilisateur est authentifié et un jeton est créé côté client. Ce jeton est signé par un service d'authentification et doit contenir suffisamment d'informations pour que l'identité de l'utilisateur puisse être établie dans tous les microservices. Le jeton est lié à chaque demande, donnant au service la possibilité de vérifier

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

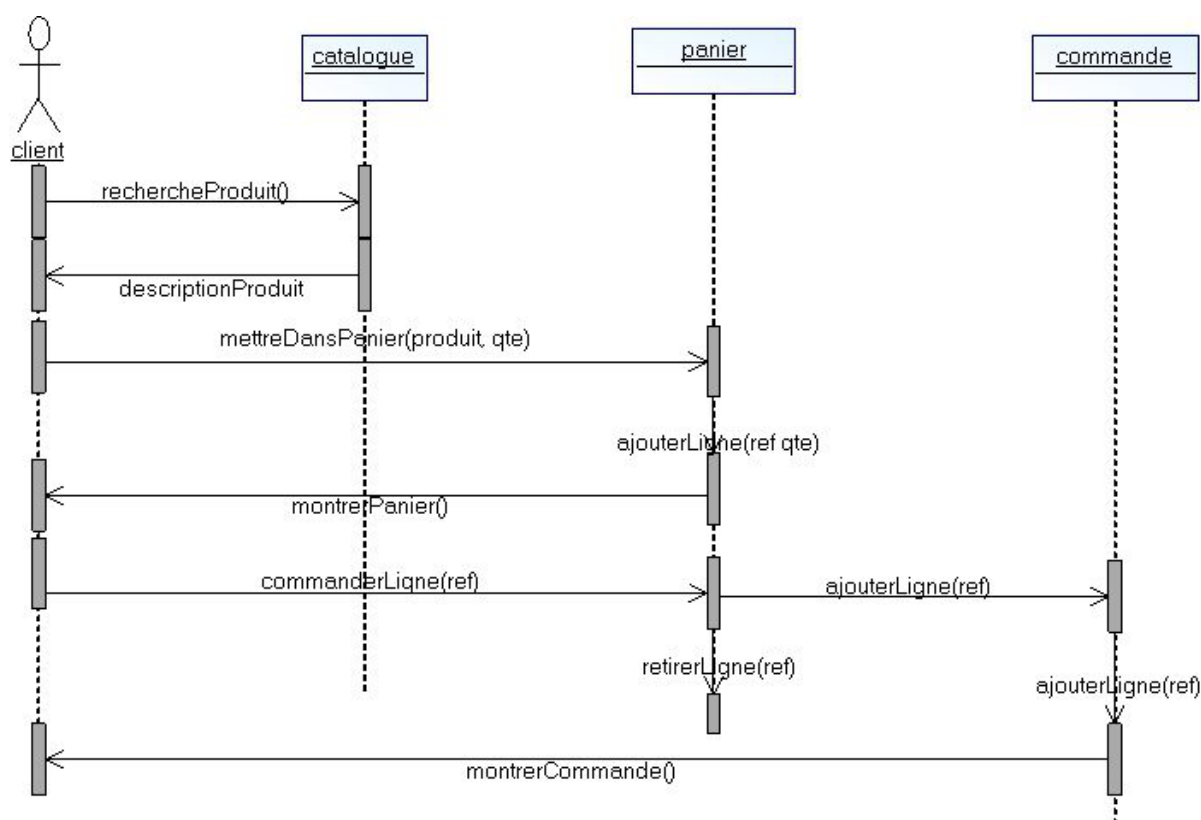
E-mail: contact@foosus.com - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24

l'utilisateur. La sécurité est relativement bonne avec cette solution mais un problème important concerne la difficulté pour se déconnecter. Les moyens de remédiation comprennent l'utilisation de jetons de courte durée et des vérifications fréquentes avec le service d'authentification. Pour l'utilisation de jetons côté client, Borsos préfère utiliser JSON Web Tokens (JWT), entre autres pour sa simplicité et ses bonnes bibliothèques de support.

5.1.2 Client

5.1.2.1 Processus fonctionnels

1. Rechercher un produit
2. Gestion du panier
3. Gestion de la commande



5.1.2.2 Description

Le microservice catalogue gère les informations relatives à tous les produits, notamment leur prix.

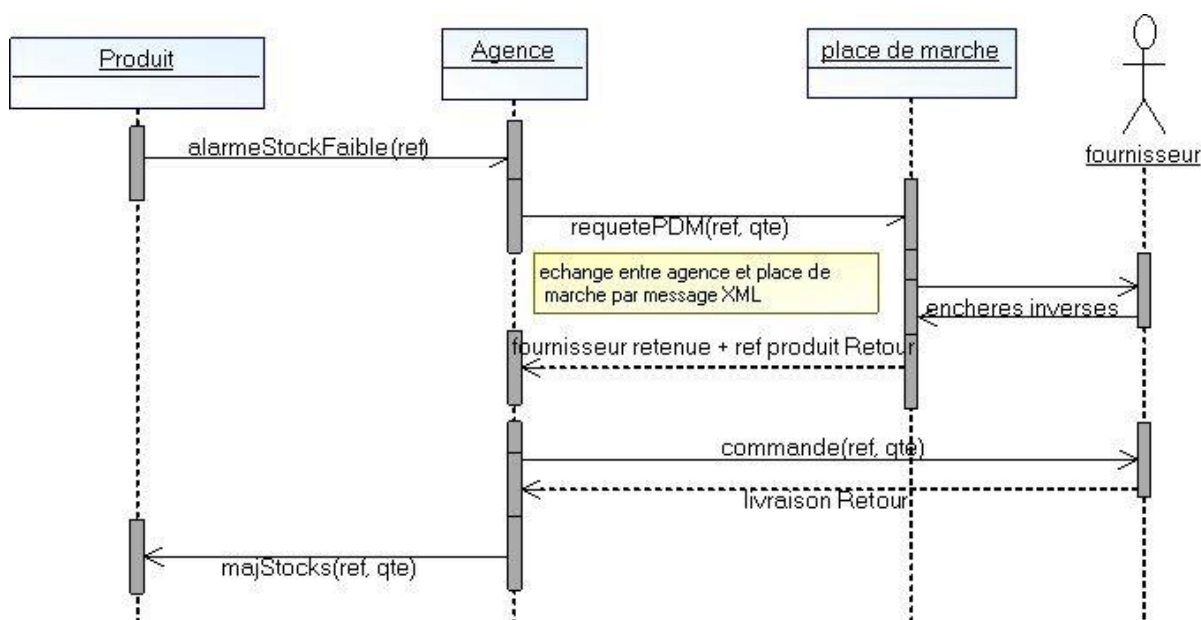
Le microservice Basket (Panier) gère les données temporelles relatives aux articles que les utilisateurs ajoutent à leurs paniers d'achat, notamment le prix des articles au moment où ils ont été ajoutés au panier. Quand le prix d'un produit est mis à jour dans le catalogue, ce prix doit l'être également dans les paniers actifs qui contiennent le produit correspondant. De

plus, le système doit en principe avertir l'utilisateur en lui indiquant que le prix d'un article particulier a changé depuis son ajout au panier.

5.1.3 Fournisseur

5.1.3.1 Processus fonctionnels

1. Faire proposition
2. Consulter demande
3. Gestion de la livraison



5.1.3.2 Description

Chaque fournisseur doit proposer tous les produits du catalogue. Pour que chaque fournisseur soit autonome, les stocks font l'objet d'une gestion et d'une comptabilité locale. Si exceptionnellement, deux fournisseurs souhaitent faire un mouvement de stocks pour des raisons qui leur sont propres, on considérera que l'un vend un produit à un client et que l'autre achète à un autre fournisseur. Le service comptabilité facturera la transaction pour le compte de chaque agence qui intervient dans le mouvement de stock. Dès qu'un produit possède une quantité en stock insuffisante dans une agence, celle-ci émet un appel d'offres sur une place de marché. La PDM (Product data management) gère alors un processus d'enchères inversées pour sélectionner le fournisseur le moins disant.

5.2 Choix technologique

La transition de l'application monolithique actuelle vers une architecture de type microservice:

Foosus: secteur de l'alimentation durable.

Adresse: 24, Boulevard Foosus, 75050 Paris - Tél.: 00 80 25 30 35 15

E-mail: contact@foosus.com - SARL au capital de 46 500 EUROS - SIRET N° 005 025 222 000 24



5.2.1 Avantages

Hétérogénéité Technologique

Avec un système composé de multiples services collaboratifs, nous pouvons décider d'utiliser différentes technologies à l'intérieur de chacun d'eux. Cela nous permet de choisir le bon outil pour chaque travail, plutôt qu'avoir à choisir une approche plus standardisée et unique. Cela finit souvent par être le plus petit dénominateur commun.

Résilience

Un concept clé dans l'ingénierie de la résilience est la cloison. Si l'un des composants d'un système échoue, mais cet échec ne cascade pas, vous pouvez isoler le problème et le reste du système peut continuer à fonctionner. Les limites de Service deviennent vos cloisons évidentes. Dans un service monolithique, si le service échoue, tout cesse de fonctionner. Avec un monolithique système, nous pouvons fonctionner sur plusieurs machines pour réduire notre risque de défaillance, mais avec microservices, nous pouvons construire des systèmes qui gèrent la défaillance totale des services et dégrader fonctionnalité en conséquence.

Mise à L'échelle

Avec un grand service monolithique, nous devons tout mettre à l'échelle ensemble. Une petite partie de notre système global est limité dans les performances, mais si ce comportement est bloqué dans une géante application monolithique, nous devons gérer la mise à l'échelle tout comme une pièce. Avec plus petits services, nous pouvons simplement mettre à l'échelle les services qui ont besoin de mise à l'échelle, ce qui nous permet de fonctionner d'autres parties du système sur un matériel plus petit et moins puissant.

Facilité de Déploiement

Un changement d'une ligne à une application monolithique d'un million de lignes nécessite l'ensemble application à déployer afin de libérer la modification. Cela pourrait avoir un impact important, déploiement à haut risque. En pratique, des déploiements à fort impact et à haut risque finissent par se produire rarement en raison de la peur compréhensible.

Avec les microservices, nous pouvons apporter une modification à un service unique et le déployer indépendamment le reste du système. Cela nous permet d'obtenir notre code déployé plus rapide. Si un problème ne se produit, il peut être isolé rapidement à un service individuel, rapide restauration facile à atteindre.

Alignement Organisationnel

Les Microservices nous permettent de mieux aligner notre architecture sur notre organisation, en nous aidant réduire le nombre de personnes travaillant sur une base de code pour atteindre le point idéal de la taille et de la productivité de l'équipe. Nous pouvons également changer la propriété des services entre les équipes pour essayer de garder les gens qui travaillent sur un service Co-localisé.

Composabilité

Une des clés les promesses des systèmes distribués et des architectures axées sur les services est que nous ouvrons des possibilités de réutilisation des fonctionnalités. Avec microservices, nous permettons à notre fonctionnalité à consommer de différentes manières à des fins différentes. Cela peut être particulièrement important lorsque nous pensons à la façon dont nos consommateurs utilisent notre logiciel.

Les microservices nous permettent de mieux aligner notre architecture avec notre organisation, nous aidant à réduire le nombre de personnes travaillant sur une base de code pour atteindre le point idéal de la taille et de la productivité de l'équipe.

Inconvénients de l'Architecture Micro-Services

Tandis que les principes de l'Architecture Micro-Services sont généralement partagés par tous, car ils constituent autant d'avantages, les inconvénients sont plus rarement évoqués. Ces inconvénients sont bien souvent les conséquences de ces principes et constituent parfois des contradictions. Il est toutefois possible d'atténuer ces inconvénients : c'est en fait là que réside le talent de l'architecte.

5.2.2 Inconvénients

- 1- Terme « micro » très relatif : la difficulté est de bien définir la granularité des micro-services. Cette granularité est variable d'un micro-service à l'autre. Ce n'est pas une bonne idée d'essayer d'obtenir une granularité uniforme pour tous les micro-services.
- 2- Problème de mise à jour des bases de données cloisonnées (chaque micro-service ayant sa propre base et pouvant être instancié plusieurs fois par le mécanisme de mise à l'échelle). Le mécanisme usuel de transaction de type « commit » n'est plus suffisant ; un mécanisme plus complexe appelé « saga » est nécessaire.
- 3- Défi en matière de débogage, test, déploiement des applications constituées de micro-services. Chaque micro-service aura été testé individuellement au préalable, mais ensuite, il faudra bien les tester tous ensemble, de façon automatisée de préférence.
- 4- Changements compliqués à cause des éventuelles dépendances entre les micro-services. Les micro-services supportent toutefois le versioning et plusieurs versions d'un même micro-service peuvent coexister, permettant une migration progressive vers la dernière version.
- 5- Application globale moins performante, car dépendante du réseau (éventuellement moins fiable). Un protocole de communication asynchrone sera souvent préféré à un protocole synchrone, afin d'éviter d'attendre trop longtemps une réponse « immédiate » à chaque

requête. De plus, l'usage d'un mécanisme de lecture via une mémoire cache est recommandé, afin d'optimiser la performance.

6- Besoin d'authentification, voire de chiffage, pour diminuer les failles de sécurité du réseau.

5.3 Azure Service Fabric

Il s'agit d'une plate-forme qui nous aide à emballer, déployer et gérer facilement des microservices évolutifs et fiables.

C'est une plateforme de systèmes distribués qui facilite le packaging, le déploiement et la gestion de conteneurs et de microservices évolutifs et fiables. Service Fabric résout également les problèmes non négligeables du développement et de la gestion des applications natives au cloud.

La singularité de Service Fabric tient au fait qu'il est axé sur la création de services avec état. Vous pouvez utiliser le modèle de programmation Service Fabric ou exécuter des services avec état en conteneur écrits dans n'importe quel langage ou code. Vous pouvez créer des clusters Service Fabric n'importe où, y compris Windows Server et Linux en local et d'autres clouds publics, en plus d'Azure.

5.4 Livraison de l'architecture et métriques business

Indicateurs de réussite

Indicateur	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.

5.5 Phases de livraison définies



Étape	Détails
1. Le fabricant crée les produits.	En fonction des besoins des grossistes, le fabricant essaie de produire le bon nombre de produits.
2. Le grossiste commande les produits.	Le grossiste essaie d'approvisionner la quantité exacte de produits nécessaires pour traiter les commandes provenant de ses détaillants (votre boutique, par exemple). Le grossiste paie les produits au fabricant.
3. Le fabricant expédie les produits au grossiste.	Le fabricant remet les produits à un transporteur cargo pour qu'il les achemine jusqu'au grossiste. Le fabricant paie le transporteur.
4. Le transporteur livre les produits au grossiste.	En fonction du mode d'expédition choisi et des emplacements des entreprises, les produits peuvent être acheminés par bateau, avion ou camion. Les produits peuvent mettre de quelques jours à quelques semaines pour arriver.
5. Le grossiste place les produits dans un entrepôt.	Le grossiste met à jour ses stocks afin que les détaillants sachent ce qui est disponible.

<p>6. Un client passe une commande dans votre boutique.</p>	<p>Vous recevez un e-mail de Shopify avec les détails de la commande.</p> <p>L'argent de la commande est ajouté à votre prochain versement.</p> <p>Le client reçoit un e-mail de confirmation de commande.</p>
<p>7. Vous passez la commande auprès de votre grossiste.</p>	<p>Si votre grossiste reçoit des notifications de commande automatiques de votre boutique, vous n'avez pas besoin de l'informer manuellement de la commande.</p> <p>Si vous configurez votre grossiste en tant que service de traitement des commandes personnalisé, vous devez marquer manuellement la commande comme traitée pour pouvoir envoyer un e-mail à votre grossiste. En faisant cela, vous envoyez également à votre client un e-mail de confirmation d'expédition.</p> <p>Si votre grossiste souhaite recevoir vos commandes d'une autre manière, assurez-vous de bien comprendre comment les lui envoyer.</p> <p>Vous payez les produits à votre grossiste.</p>
<p>8. Votre grossiste traite la commande.</p>	<p>Votre grossiste expédie la commande à votre client au moyen d'un service de coursier.</p> <p>Votre grossiste paie le service de coursier.</p> <p>Votre client reçoit un e-mail de confirmation d'expédition.</p> <p>Si le transporteur fournit un numéro d'expédition, vous pouvez l'envoyer à votre client dans un e-mail de mise à jour du statut d'expédition.</p>

9. Le service de coursier livre le produit à votre client.	En fonction du mode d'expédition choisi, cela peut prendre de quelques jours à quelques semaines.
10. Votre client reçoit sa commande.	Le produit passe ainsi du fabricant au client.

Livrables architecturaux qui satisfont aux conditions requises pour le business.

Principales étapes du processus métier

1. **Processus d'approvisionnement:** acquisition d'équipements et de services, préparation des prestataires de services et coordination des expéditions.
2. **Tarification:** identification de nouveaux producteurs locaux potentiels, renouvellement des contrats existants et signature de nouveaux contrats avec des producteurs. Dans cette étape les spécialistes parcourront diverses sources pour déterminer le meilleur prix du marché pour chaque article.
3. **Inventaire :** Un nouvel ajout d'inventaire est créé pour chaque nouveau produit, ce qui obligera l'équipe chargée de l'inventaire à télécharger l'image sur le système, à saisir les détails pertinents du pack d'actifs et à marquer l'article comme prêt pour le stock mais pas généralement disponible à l'achat.
4. **Stockage:** Chaque producteur met à jour les données sur les quantités de produits qu'il a en stock disponibles à la vente.
5. **Service Clients:** Au fur et à mesure que les clients recherchent des produits en ligne, les commandes sont rassemblées et les vendeurs répondent aux besoins des clients.
6. **Livraison des commandes:** Les services de transport des producteurs ou d'un prestataire livrent les commandes.
- 7.

6 Plan de travail commun priorisé

Pour mener à bien le projet Foosus, nous avons retenus 4 grandes étapes : l'étude, la conception, le développement et l'accompagnement :

L'étude :

- Réflexion sur l'objectif du projet, sa ou ses cible(s), sa rentabilité, les moyens financiers à engager
- Réflexion sur l'impact du site en interne et mesurer les moyens humains pour faire vivre le site
- Réflexion sur le contenu, son organisation, l'apparition des pages, les services attendus
- Synthèse de toutes ces réflexions, rédaction et validation d'une feuille de route, ou d'un cahier des charges

La conception :

- Définition de la navigation et de l'ergonomie du site
- Création graphique
- Définition d'une stratégie de référencement
- Définition des contenus
- Réflexion sur l'accessibilité du site
- Définition des services et des outils
- Définition des contenus dynamiques
- Choix du type d'hébergement
- Synthèse et validation

Le développement :

- Intégration
- Création ou fourniture des contenus
- Développement d'outils spécifiques si besoin
- Intégration éventuelle à un système d'information
- Test
- Formation des utilisateurs à l'outil de gestion de contenus choisi
- Mise en ligne

L'accompagnement :

- Lancement du site, référencement, publicité, inscription dans des annuaires, tout ce qui peut faire connaître le site
- Etude du comportement des internautes via des outils statistiques puissants, analyse et préconisation d'évolution
- Maintenance du site, choix de nouveaux contenus, corrections éventuelles, amélioration continue de la qualité du site

7 Plan de communication

Voir « Plan de communication » dans la Déclaration de Travail d'Architecture

8 Risques et facteurs de réduction

8.1 Structure de gouvernance

Type	Description du risque	Actions préventives
1. Ressources humaines	Les compétences de l'équipe sont insuffisantes	Structuration de l'équipe Redistribution des rôles Renforcement de l'encadrement Formation, entraide, motivation
2. Ressources humaines	Turn over de l'équipe très important	Redistribution des rôles Formation, entraide, motivation Période de recouvrement Assistance externe
3. Ressources humaines	Absence de motivation des équipes	Redistribution des rôles Responsabilisation Formation
4. Planification	Les prévisions sont optimistes, les ressources et les budgets sont sous-estimés	Recoupement de plusieurs estimations détaillées des charges, coûts et plannings Remise en cause des demandes Développement incrémental Réutilisation de logiciels
5. Management	Le projet est de taille excessive	Création de sous-projets avec responsables Développement incrémental Calcul des retours sur investissement, analyse de la valeur (enjeux des fonctions)
6. Planification	Délais tendus	Planification détaillée de tout le projet Identification du chemin critique Suivi hebdomadaire de l'avancement
7. Moyens	Budget serré	Suivi régulier Réutilisation de logiciels



8. Moyens	Indisponibilité des locaux lors des montées en charge	Report d'activités chez des prestataires externe
9. Démarche	Des modifications fréquentes sont demandées pendant le développement	Seuil d'acceptation des changements Développement incrémental, gestion de lots Report des modifications en fin de projet, gestion de versions
10. Démarche	Perfectionnisme	Examen critique des spécifications Maquettage Calcul des retours sur investissement, analyse de la valeur (enjeux des fonctions)
11. Démarche	Dépendance d'autres adaptations du système d'information	Réunions de coordination Synchronisation pour la diffusion Plate-forme de test dédiée Réception commune
12. Démarche	Flou de l'organisation maîtrise d'ouvrage/maîtrise d'œuvre	Rédaction d'une charte de projet Création de comités de suivi
13. Démarche	Déficiences du maître d'ouvrage	Assistance externe Assistance par le maître d'œuvre Création de groupes de travail d'utilisateurs
14. Management	Le suivi est insuffisant et ne permet pas de détecter des dérives	Réunions de suivi hebdomadaires Planning détaillé Fiches d'activités
15. Management	Dépendance de la sous-traitance	Demande d'engagements sur la qualité de service Contenu du contrat Suivi des réalisations sous-traitées Livraisons intermédiaires Audit qualité



16. Contractuel	Les fournitures externes sont défaillantes	Mise en concurrence Contrôle des références Analyse de compatibilité Tests de réception
17. Contractuel	Les travaux sous-traités sont défaillants.	Contrôle des références Audit de qualification
18. Fonctionnel	Le cahier des charges fonctionnel est incomplet	Développement incrémental Formation des concepteurs Création de groupes de travail d'utilisateurs Revue du cahier des charges
19. Fonctionnel	Le produit final ne correspond pas aux attentes des utilisateurs	Analyse du travail (démarche d'ergonomie) : organisation, missions... Maquettage Rédaction anticipée des tests de réception ou des manuels utilisateurs Exploitation des retours des sites pilotes Mesure de l'insatisfaction
20. Fonctionnel	Les interfaces utilisateurs ne sont pas bonnes	Analyse du travail (démarche d'ergonomie) : prise en compte de l'utilisateur (fonction, comportement, charge de travail) Production de règles d'ergonomie Maquettage et évaluation ergonomique
21. Fonctionnel	Défaut de formalisation des informations, des processus	Maquettage Développement incrémental
22. Fonctionnel	Incompréhension des spécifications	Rencontre d'utilisateurs Maquettage
23. Fonctionnel	Perte des données lors du changement de système	Développement d'un outil d'analyse des bases Sauvegardes préalables Sites pilotes
24. Fonctionnel	Anomalies de fonctionnement	Renforcement de la démarche de test Participation de l'AU à la réception

		Sites pilotes
25. Technique	Importance des changements technologiques (innovation)	Identification d'experts internes Formation Assistance externe
26. Technique	Les temps de réponse ne sont pas satisfaisants	Simulation Essais comparatifs Modélisation, prototypage Observation en sites pilotes et en formation Instrumentation, réglages Suivi de tableaux de bord
27. Technique	Il y a blocage sur les limites technologiques des plates-formes	Analyse technique Vérification a priori des performances Analyse des coûts Enquête sur les configurations
28. Technique	Instabilité de l'environnement	Recensement des bugs connus Choix de versions des logiciels Recommandations de configuration Accord avec les constructeurs/éditeurs
29. Fonctionnel ou Technique	Les interfaces avec d'autres systèmes sont nombreuses ou complexes.	Spécifications détaillées des interfaces Réunions de coordination Plate-forme de test dédiée Réception commune Observation en sites pilotes et en début d'exploitation
30. Organisationnel	L'organisation des structures internes de l'organisme est profondément modifiée	Communication Formation fonctionnelle des utilisateurs Assistance aux utilisateurs
31. Organisationnel	Les utilisateurs finals ne sont pas impliqués	Organisation de groupes de travail pour valider les spécifications, une maquette... Création d'un comité des utilisateurs Communication Choix de sites pilotes
32. Organisationnel	Attitude hostile des futurs utilisateurs	Organisation de groupes de travail d'utilisateurs pour valider les spécifications, une



		maquette... Création d'un comité des utilisateurs Communication Formation
--	--	---

8.2 Analyse des risques

Voir « Analyse des risques » dans la Déclaration de Travail d'Architecture

9 Hypothèses

Voir « Hypothèses » dans la Déclaration de Travail d'Architecture

10 Critères d'acceptation et procédures

Voir « Critères d'acceptation et procédures » dans la Déclaration de Travail d'Architecture

11 Procédures de changement de périmètre

Voir « Procédures de changement de périmètre » dans la Déclaration de Travail d'Architecture

12 Calendrier

Etant donné que chaque livrable aura en moyen une durée maximale de 3 semaines et sachant qu'environ 30 livrables ont été définis, il nous faudra 90 semaines pour un seul développeur, soit 630 jours ouvrables, pour mener à bien le processus de migration.

Cependant, en fonction du nombre d'équipes technique que Foosus sera capable de créer, si nous embauchons une équipe de 3 développeurs, ce temps sera réduit à environ 210 jours soit 7 mois.

13 Phases de livrables définies

Voir « Hypothèses » dans la Déclaration de Travail d'Architecture



14 Personnes approuvant ce plan

Validateurs	Date
La Direction del'Entreprise	05/08/2021
Natasha Jarson	10/08/2021
Le comité d'architecture	15/08/2021

15 Annexes

<https://espaces-numeriques.org/wp-content/uploads/2019/01/l81p09.pdf>

<https://www.sebweb.fr/expertise-agence?projet-web>

<https://docplayer.fr/8975018-Suivi-des-risques-d-un-projet.html>

<https://espaces-numeriques.org/wp-content/uploads/2019/01/l22p20.pdf>

<https://marcautran.developpez.com/tutoriels/uml/conception-uml/#LII>

<https://ichi.pro/fr/architecture-de-microservice-et-modeles-de-conception-pour-les-microservices-247274173093258>