

Nouvelle plateforme Foosus

Spécification des conditions requises pour l'architecture



ARCHITECTE LOGICIEL

PROJET

DATE
DÉPÔT

DATE
FIN

P5

Concevez une nouvelle architecture afin de soutenir le développement de votre entreprise

Livrable 2/4

Spécification des conditions requises pour l'architecture

04/04/2023

04/04/2023

Marc LEFRANÇOIS

mlefrancois.contact@gmail.com

CONTRÔLE DES RÉVISIONS

DATE	STATUT	AUTEUR	RÔLE	ACTIONS
10/02/2023	Brouillon	Marc LEFRANÇOIS	Créateur	Création du document et mise en place du plan.
10/02/2023	Brouillon	Marc LEFRANÇOIS	Créateur	Enrichissement du document → V1.
26/03/2023	Brouillon	Marc LEFRANÇOIS	Créateur	Envoi pour validation.
28/03/2023	Validé	Grégoire CATAN	Correcteur	Relecture du document et validation.
28/03/2023	Validé	Marc LEFRANÇOIS	Créateur	Envoi pour relecture et remarques.
29/03/2023	Relu	Jérôme POCHÉLU	Correcteur	Relecture du document et remarques.
31/03/2023	Finalisé	Marc LEFRANÇOIS	Créateur	Ajustement du contenu → V2.

Légendes :

Statut = Brouillon, Validé, Relu et Finalisé.

Rôle = Créateur / Modificateur, relecteur/correcteur, validateur.

CONTRÔLE DES RÉVISIONS.....	2
1. OBJET.....	5
2. MISSION D'ARCHITECTURE	5
2.1. CONTEXTE	5
2.2. OBJECTIFS	5
2.3. PERIMETRE	6
2.4. PARTIES PRENANTES	6
2.5. DOCUMENTS FOURNIS.....	6
3. SITUATION À DATE	7
4. PROBLÉMATIQUE & IMPACTS.....	7
4.1. CAUSES TECHNIQUES	7
4.2. CONSÉQUENCES	7
4.3. METHODE DES 5 POURQUOI (BUSINESS)	7
5. MESURE DU SUCCÈS	9
6. CONDITIONS REQUISES POUR L'ARCHITECTURE	9
6.1. ÉVOLUTIVITE.....	9
6.2. SIMPLICITE	9
6.3. MAINTENABILITE.....	10
6.4. DISPONIBILITE	10
6.5. PERFORMANCE	10
6.6. SECURITE.....	11
6.7. COMPATIBILITE	11
7. CONTRATS DE SERVICE.....	11
7.1. BUSINESS.....	11
7.2. APPLICATION	11
8. LIGNES DIRECTRICES POUR L'IMPLÉMENTATION.....	13
8.1. ÉVOLUTIVITE.....	13
8.2. SIMPLICITE	14
8.3. MAINTENABILITE.....	14
8.4. DISPONIBILITE	14
8.5. PERFORMANCE	14
8.6. SECURITE.....	14
8.7. ACCESSIBILITE	15
9. SPÉCIFICATIONS POUR L'IMPLÉMENTATION.....	15
9.1. ÉVOLUTIVITE.....	15
9.2. SIMPLICITE	15
9.3. MAINTENABILITE.....	15
9.4. DISPONIBILITE	16
9.5. PERFORMANCE	16
9.6. SECURITE.....	16
9.7. ACCESSIBILITE	17
10. STANDARDS POUR L'IMPLÉMENTATION	17
10.1. ÉVOLUTIVITE.....	17
10.2. SIMPLICITE	18
10.3. MAINTENABILITE.....	18
10.4. DISPONIBILITE ET PERFORMANCE	18
10.5. SECURITE.....	19
10.6. ACCESSIBILITE	19

11. CONDITIONS REQUISES POUR L'INTEROPÉRABILITÉ.....	20
12. CONDITIONS REQUISES POUR LE MANAGEMENT DU SERVICE IT	20
12.1. PRODUCT MANAGEMENT	21
12.2. GOUVERNANCE ARCHITECTURALE.....	21
12.3. PRODUCTION LOGICIELLE	21
12.4. INFRASTRUCTURE	21
12.5. EXPLOITATION.....	22
13. CONTRAINTES.....	22
14. HYPOTHÈSES DE SOLUTIONS	23
14.1. TAUX D'INSCRIPTIONS UTILISATEURS	23
14.2. ALIGNEMENT COMMERCIAL & TECHNIQUE.....	23
14.3. COMPETITIVITE CONCURRENTIEL	23
14.4. ALIGNEMENT TECHNICO-COMMERCIAL DE LA PERFORMANCE	23
14.5. LA REPUTATION DE LA MARQUE	23

1. Objet

Le présent document spécifie les conditions requises pour l'architecture, son implémentation et la conformité de cette dernière, dans le cadre du projet initialisant la définition de la nouvelle plateforme Foosus et de sa nouvelle architecture.

Son objectif est de décrire les exigences, principes, règles et métriques à respecter ou à atteindre pour s'assurer que la nouvelle architecture réponde à la vision stratégique de l'entreprise.

Aussi assurera-t-il également un rôle de référentiel quant à la conformité de la réponse architecturale apportée (*voir le document de déclaration de travail d'architecture*).

2. Mission d'architecture

Cette section resitue les éléments clés nécessaires à la bonne compréhension de ce qui est attendu, ici dans ce projet, la mise en place d'une nouvelle architecture pour soutenir les activités commerciales de la plateforme, et ceci afin de fournir une réponse adéquate aux attentes.

2.1. Contexte

Foosus, une start-up de 3 ans positionnée dans le secteur de l'alimentation durable, s'est donnée pour objectif de soutenir l'alimentation locale en mettant en relation les consommateurs avec des producteurs et des artisans locaux via à la fois une application mobile, et un site Web d'e-commerce. Cependant, elle rencontre des problèmes techniques et fonctionnels qui respectivement écornent son image de marque auprès de ses utilisateurs, et l'empêchent d'évoluer pour répondre aux exigences du marché.

Foosus a donc besoin de répondre rapidement et activement à ces problématiques, si elle souhaite garder la confiance de ses utilisateurs. Et c'est ce qu'elle souhaite faire avec une nouvelle plateforme basée sur une nouvelle architecture technique.

2.2. Objectifs

Les objectifs désignés ci-après ont été énoncés respectivement dans les courriels de **Ash CALLUM** cofondatrice et CEO de **Foosus** et ayant pour objet « **Fonction Architecture** », et de Natasha JARSON cofondatrice et CIO de **Foosus**, ainsi que dans les documents attenants (*voir la section « 2.5. Documents fournis »*).

Il s'agit de définir une nouvelle architecture standardisée permettant globalement :

1. d'effacer la dette technique et ne plus en avoir à l'avenir ;
2. d'évoluer avec l'entreprise, id. soutenir sa croissance de manière continue ;
3. de prendre en charge un nouvel emplacement géographique, afin de tirer parti de la géolocalisation pour relier fournisseurs et consommateurs, pour proposer des produits disponibles à proximité des lieux de résidence de ces derniers ;
4. de faciliter la maintenance et les développements futurs (*maitrise des coûts et des délais*) ;
5. de fournir la fiabilité nécessaire aux clients, fournisseurs et consommateurs de la Société.

L'état final recherché pour la société est de créer une plateforme de commerce électronique polyvalente lui permettant de franchir le million d'utilisateurs, et de concurrencer ainsi les grandes entreprises mondiales de commerce électronique qui dominent actuellement le marché de l'alimentation durable. Aussi, la définition d'une nouvelle architecture est au cœur des préoccupations sur la réponse à fournir pour ce faire, en étant capable de supporter les exigences de la nouvelle plateforme.

2.3. Périmètre

Le périmètre porte sur la définition et la mise en place d'une toute nouvelle architecture pour la plateforme e-commerce de **Foosus** représentant son cœur de métier.

2.4. Parties prenantes

Ci-dessous l'ensemble des parties prenantes pour le projet de la nouvelle architecture de la plateforme e-commerce de Foosus :

PARTIE PRENANTE	RÔLE / QUALITÉ	VISION
Ash CALLUM Chief Executive Officer	Responsable de la stratégie de l'entreprise.	Business
Daniel ANTHONY Chief Product Officer	En charge du référentiel métier produit : -> définition fonctionnelle, -> cas d'utilisations, -> métriques.	Business
Christina ORGEGA Chief Marketing Officer	En charge du référentiel métier clients : -> définition fonctionnelle, -> cas d'utilisations, -> métriques.	Business
Jo KUMAR Chief Financial Officer	En charge du référentiel métier clients : -> définition fonctionnelle, -> cas d'utilisations, -> métriques.	Financière
Natasha JARSON Chief Information Officer	Encadrant supérieur et pilote du processus SI : -> Définition des processus SI, -> Suivi projet, -> Définition des métriques des processus SI.	IT
Jack HARKNESS / HARKNER Chief Operational Officer	En charge du suivi : -> de la maintenance de l'actuelle plateforme, -> de l'implémentation de la nouvelle.	IT
Peter Parker Engineering Owner	Référent technique de la plateforme e-commerce.	IT
Marc LEFRANÇOIS Software Architect	En charge de la définition et du support de l'implémentation de la nouvelle architecture.	IT

2.5. Documents fournis

- ☛ l'autorisation de projet de Ash CALLUM ;
- ☛ les conditions requises pour le business de haut niveau de Daniel ANTHONY et Natasha JARSON ;
- ☛ La vision architecturale ;
- ☛ Les flux de valeurs organisationnelle ou VSM pour Value Stream Map ;
- ☛ Organigramme de FOOSUS ;
- ☛ Le document de référence de l'architecture actuelle « Legacy C4 FR ».

3. Situation à date

À l'heure actuelle l'architecture logicielle de l'entreprise est composée d'un ensemble hétérogène de technologies ne lui permettant pas d'évoluer avec la société en fonction de ses besoins, d'une fait une dette technique. Aussi, l'architecture actuelle entrave la croissance de la société, qui ne peut innover dans ses produits, et ainsi attirer de nouveaux utilisateurs.

En effet, le système, dans sa partie logicielle, est en incapacité d'évoluer au même rythme que la base clientèle de la société, id. d'absorber les pics d'utilisation, et ce faisant, cela crée des pannes, et donc des interruptions de service.

4. Problématique & impacts

4.1. Causes techniques

- ☛ Architecture logicielle composée d'un ensemble hétérogène de technologies.
- ☛ Incapacité du système logiciel d'absorber les pics d'utilisation.

4.2. Conséquences

- ☛ L'architecture actuelle entrave la croissance de la société, qui ne peut innover dans ses produits, et ainsi attirer de nouveaux utilisateurs.
- ☛ Pannes créant des interruptions de service.

4.3. Méthode des 5 pourquoi (business)

4.3.1. Taux d'inscriptions utilisateurs #1

1. La société ne peut attirer de nouveaux utilisateurs ! Pourquoi ?
2. La société ne peut innover ! Pourquoi ?
3. L'architecture entrave la croissance de la société ! Pourquoi ?
4. L'architecture logicielle est technologiquement hétérogène ! Pourquoi ?
5. L'ancienne DSI a laissé l'équipe technique expérimenter de nouvelles approches techniques sans cadre standard ni vision à long terme. ➔ **root cause**.

4.3.2. Taux d'inscriptions utilisateurs #2

1. La société ne peut attirer de nouveaux utilisateurs ! Pourquoi ?
2. La correspondance de la société avec le marché a été éclipsée ! Pourquoi ?
3. L'image de marque auprès des clients / utilisateurs est écornée ! Pourquoi ?
4. Des pannes créent des interruptions de service de la plateforme ! Pourquoi ?
5. L'architecture logicielle est technologiquement hétérogène ! Pourquoi ?
6. L'ancienne DSI a laissé l'équipe technique expérimenter de nouvelles approches techniques sans cadre standard ni vision à long terme. ➔ **root cause**.

4.3.3. Alignement commercial & technique

1. Pas d'alignement entre la partie commerciale et la partie technique ! Pourquoi ?
2. La plateforme a naturellement évolué vers la complexité avec une architecture hétérogène, induisant de facto un manque de clarté et donc de compréhension de cette dernière ! Pourquoi ?
3. L'ancienne DSI a laissé l'équipe technique expérimenter de nouvelles approches techniques sans cadre standard ni vision à long terme. ➔ **root cause**.

4.3.4. Compétitivité concurrentielle

1. Les concurrents directs prennent l'avantage, des parts de marché, sur la société ! Pourquoi ?

2. Contrairement à ses concurrents, la société n'arrive pas à s'adapter à la demande du marché, aux attentes des clients / utilisateurs ! Pourquoi ?
3. L'architecture peut difficilement évoluer / s'étendre ! Pourquoi ?
4. L'architecture est verrouillée par des solutions passées, accusant de fait une dette technique importante ! Pourquoi ?
5. L'ancienne DSI a laissé l'équipe technique expérimenter de nouvelles approches techniques sans cadre standard ni vision à long terme. ➔ **root cause**.

4.3.5. Alignement technico-commercial de la performance

1. La vision de l'intelligence business, id. l'alignement entre la performance technique et commerciale n'est pas clair ! Pourquoi ?
2. La recherche de l'intelligence business nécessite au préalable une lourde étape d'analyse de données ! Pourquoi ?
3. Les données à analyser nécessitent de passer par des outils tiers (*registres et feuilles de calcul*) ! Pourquoi ?
4. La plateforme n'offre pas de moyens de supervision à la fois sur sa performance technique que sur sa performance business ! ➔ **root cause**.

4.3.6. La réputation de la marque #1

1. Les livraisons créent des interruptions de service de la plateforme, impactant les utilisateurs ! Pourquoi ?
2. Incapacité de livrer de nouvelles versions / évolutions de la plateforme sans interruptions de service ! Pourquoi ?
3. Pas de processus qualité quant aux dites livraisons ! ➔ **root cause**.

4.3.7. La réputation de la marque #2

1. Des pannes créent des interruptions de service de la plateforme, impactant les utilisateurs ! Pourquoi ?
2. Le système logiciel est en incapacité d'absorber les pics d'utilisation ! Pourquoi ?
3. L'architecture logicielle est technologiquement hétérogène ! Pourquoi ?
4. L'ancienne DSI a laissé l'équipe technique expérimenter de nouvelles approches techniques sans cadre standard ni vision à long terme ! ➔ **root cause**.

5. Mesure du succès

Le catalogue définissant les métriques de références, à savoir les indicateurs clés de performance ou KPI, a été défini dans le troisième livrable traitant du contrat d'architecture avec les utilisateurs business à la section « **9. Conformité** ».

6. Conditions requises pour l'architecture

Afin de répondre aux besoins, exigences et niveaux de service business, le catalogue des exigences a été défini ci-après, et répertoriant les attentes de la nouvelle architecture, incluant la nouvelle fonctionnalité. Pour des soucis de lisibilité et de correspondance avec les bonnes pratiques architecturales définies dans le premier livrable traitant, lesdites exigences ont été réparties entre les sept catégories suivantes :

1. Évolutivité
2. Simplicité
3. Maintenabilité
4. Disponibilité
5. Performance
6. Sécurité
7. Compatibilité

6.1. Évolutivité

IDENTIFIANT	INTITULÉ
NDS-A-1-1	L'architecture peut inclure de nouveaux composants personnalisés ou des composants du commerce pour favoriser la flexibilité, la stabilité et l'extensibilité.
NDS-A-1-2	La nouvelle architecture, découlant de la nouvelle plateforme de commerce électronique, doit être évolutive (scalable).
NDS-A-1-3	La future plateforme doit s'adapter aux particularités locales et répondre aux exigences d'utilisation des clients.
NDS-A-1-4	Les livrables doivent pouvoir être fournis à intervalles réguliers pour que le nouveau système soit rapidement opérationnel et puisse être doté de nouvelles fonctionnalités au fil du temps.
NDS-A-1-5	Inclure la fonctionnalité de géolocalisation.

6.2. Simplicité

IDENTIFIANT	INTITULÉ
NDS-A-2-1	La future plateforme doit s'exécuter dans un environnement Cloud.
NDS-A-2-2	Possibilité d'inverser les décisions d'architecture tant que cela reste peu onéreux.

La simplification par le standardisation et notamment via une architecture microservices et par une implémentation au sein d'un environnement Cloud

6.3. Maintenabilité

IDENTIFIANT	INTITULÉ
NDS-A-3-1	Les solutions open source sont préférables aux solutions payantes.
NDS-A-3-2	Le support continu des composants doit être pris en compte lors de leur sélection ou lors des prises de décision de création ou d'achat.
NDS-A-3-3	Toutes les solutions du commerce ou open source doivent, dans la mesure du possible, faire partie d'une même pile technologique, afin de réduire les coûts de maintenance et de support continus.
NDS-A-3-4	La solution doit permettre de passer d'un taux d'incidents de production de plus de 25% par mois à un taux inférieur à 1% par mois.

6.4. Disponibilité

IDENTIFIANT	INTITULÉ
NDS-A-4-1	La plateforme doit être active 24/7 (24 heures 7 jours sur 7) quelle que soit la situation de celle-ci.
NDS-A-4-2	Les améliorations et autres modifications apportées aux systèmes de production devront limiter ou supprimer la nécessité d'interrompre le service pour procéder au déploiement.
NDS-A-4-3	En cas de surcharge du système, les utilisateurs connectés doivent pouvoir continuer à accéder à tous les services de façon dégradée.
NDS-A-4-4	La nouvelle solution de plateforme de commerce électronique doit être disponible pour les clients de la Société (fournisseurs et les consommateurs) quelle que soit la zone géographique où qu'ils se trouvent.
NDS-A-4-5	Accessibilité et performance "similaire" de la plateforme aux utilisateurs et ce quelle que soit la zone géographique considérée.
NDS-A-4-6	Chaque nouvelle version doit être de taille réduite, présenter peu de risques, être transparente pour nos utilisateurs et rester accessible en tout lieu et à tout moment

6.5. Performance

IDENTIFIANT	INTITULÉ
NDS-A-5-1	Accessibilité et performance "similaire" de la plateforme aux utilisateurs et ce quelle que soit la zone géographique considérée.
NDS-A-5-2	Prise en compte des bandes passantes : réseaux fixe et mobile.
NDS-A-5-3	La future plateforme doit supporter quotidiennement une augmentation d'adhésions utilisateurs de 10%.
NDS-A-5-4	La future plateforme doit absorber le trafic, mais également gérer les augmentations de charges.
NDS-A-5-5	Tirer parti de la géolocalisation pour relier des fournisseurs et des consommateurs et pour proposer des produits disponibles à proximité des lieux de résidence de ces derniers.
NDS-A-5-6	Prise en compte des catégories d'utilisateurs (fournisseurs, consommateurs, ...) et des fonctionnalités et services associés (spécifiques).

6.6. Sécurité

IDENTIFIANT	INTITULÉ
NDS-A-6-1	Garantir la sécurité de la plateforme à chaque fois qu'elle évoluera / s'élargira.
NDS-A-6-2	Chaque nouvelle version doit être de taille réduite, présenter peu de risques, être transparente pour nos utilisateurs et rester accessible en tout lieu et à tout moment
NDS-A-6-3	Les utilisateurs doivent être authentifiés selon les systèmes de gestion des utilisateurs.
NDS-A-6-4	La sécurité doit être au cœur des priorités et une préoccupation à tous les niveaux : -> Métier -> Applicatif (<i>fonctionnalités et données informatique</i>) -> Infrastructure -> Opérationnel

6.7. Compatibilité

IDENTIFIANT	INTITULÉ
NDS-A-7-1	Prise en charge des ordinateurs fixes et portables.
NDS-A-7-2	Prise en charge des terminaux mobiles IOS (Apple) : smartphones et tablettes.
NDS-A-7-3	Prise en charge des terminaux mobiles Android (Google) : smartphones et tablettes.
NDS-A-7-4	Prise en charge des environnements OS (ordinateurs) : -> Windows, -> Mac, -> Linux.
NDS-A-7-5	Prise en charge des environnements OS (terminaux) : -> Android, -> IOS.
NDS-A-7-6	Prise en charge des navigateurs Web : -> Edge (Microsoft), -> Chrome (Google), -> Firefox (Mozilla), -> Safari (Apple).
NDS-A-7-7	La future plateforme doit s'exécuter dans un environnement Cloud.

7. Contrats de service

7.1. Business

Le niveau de service a été défini dans le troisième livrable traitant du contrat d'architecture avec les utilisateurs business à la section « **8. Niveau de service** ».

7.2. Application

À la page suivante se trouve le catalogue du niveau de service applicatif attendu.

IDENTIFIANT	NIVEAU DE SERVICE	DESCRIPTION / DÉTAIL	MOYEN / MESURE
NDS-A-1	Évolutivité	La nouvelle architecture, découlant de la nouvelle plateforme de commerce électronique, doit être évolutive (scalable), doit s'adapter aux particularités locales et répondre aux exigences d'utilisation des clients.	Toute nouvelle fonctionnalité business doit pouvoir être implémentée.
NDS-A-3	Évolutivité	Le délai de déploiement d'une fonctionnalité ne doit pas excéder 15 jours.	Mesure du délai moyen entre la date de l'expression finalisée du besoin et celle du déploiement effectif (MEP/MES).
NDS-A-4	Disponibilité	Le temps d'interruption si nécessaire et partiel, id. pour limité à la fonctionnalité ou au processus métier considéré, pour le déploiement d'une nouvelle version ne doit pas excéder 30 minutes.	Mesure du délai moyen entre le début et la fin du déploiement (MEP/MES).
NDS-A-5	Disponibilité	Tout implémentation doit être en conformité avec les standards et charte de développement définis par l'architecte.	Supervision de la production logicielle et de l'exploitation de la plateforme, avec des outils analytique vérifiant : -> la conformité de codage au standard attendu, -> la non regression fonctionnelle, -> la vulnérabilité du code. -> la vulnérabilité de la plateforme.
NDS-A-6	Maintenabilité	L'implémentation d'une nouvelle fonctionnalité ou version de la plateforme ne doit pas généré de bug majeur ou critique à l'issue de son déploiement en production (MEP/MES).	Mesure du nombre de tickets majeur et critique. Audit de l'implémentation considérée devant respecter : -> les standards établis par l'architecte, -> les tests de recettes utilisateurs, -> les tests de non régression.
NDS-A-7	Sécurité	L'implémentation d'une fonctionnalité ne doit pas induire de vulnérabilité.	Supervision de la production logicielle et de la plateforme avec des outils analytique de vulnérabilité. Audit de l'implémentation considérée devant respecter les standards de sécurité en vigueur, définis / relayés par l'architecte.
NDS-A-8	Compatibilité	Les navigateurs web disposant d'une part de marché > 2% des utilisateurs (<i>mesures Foosus ou GlobalStats, la plus haute étant retenue</i>) mobile ou desktop seront supportés.	Analyse statistique.

8. Lignes directrices pour l'implémentation

Les lignes directrices pour l'implémentation définissent les principes à mettre en œuvre pour la conception et le déploiement de la nouvelle solution.

L'objectif, outre de répondre aux besoins et niveaux de service attendu par le métier / client / utilisateur, est de garantir le bon alignement stratégique entre la vision business et la technique desservant cette vision. Autrement dit, l'architecture se doit d'être au service des fonctions métiers, et d'une manière générale de toutes les couches : applicative¹, infrastructure et opérationnelle qui ont déjà abordées dans d'autres livrables², vis-à-vis de la couche métier.

Aussi pour ce faire, les lignes directrices doivent tenir compte tout à la fois :

- des besoins définis par le métier ;
- du niveau de service attendus par le métier de la plateforme ;
- des standards et des bonnes pratiques, représentant l'état de l'art de l'architecture et de l'implémentation technique.

Et c'est donc sur les critères de référence suivants qu'ont été basées lesdites lignes directrices :

- Évolutivité
- Disponibilité
- Accessibilité
- Simplicité
- Performance
- Maintenabilité
- Sécurité

Lesdits critères sont à considérés comme condition sine qua non à la réussite de la réalisation, et que pris dans leur ensemble permettront de limiter les écueils à l'usage et permettant d'éviter des conséquences catastrophiques sur les activités de l'entreprise.

8.1. Évolutivité

Afin de garantir le maintien de l'alignement entre le business et la technique, il est indispensable de pouvoir faire évoluer fonctionnellement la plateforme e-commerce de Foosus. Ceci étant, il est tout autant indispensable de tenir également compte de l'évolution technologique, sans quoi le système pourrait accuser une dette technique pouvant nuire au bon déroulement des activités de l'entreprise ; ne serait-ce que de considérer à titre d'exemple les aspects liés à la sécurité.

L'usage de standards technologiques par pile, écosystème, ou bien encore de framework permet de concilier ces deux types d'évolution.

8.1.1. Fonctionnelle

D'un point de vue fonctionnel, l'emploi de standards comme susmentionné permet au système de pouvoir répondre aux besoins toujours croissant du métier, en facilitant son évolution.

8.1.2. Technologique

D'un point de vue technologique, l'emploi de standards comme susmentionné permet également, à l'instar du point de vue fonctionnel, de faciliter l'évolution du système, mais en permettant cette fois le maintien technologique de celui-ci, dans un souci de contrer la dette technique, avec un support à long terme, et en disposant de mises à jour améliorant sa robustesse sans à devoir fréquemment tout remettre en question.

¹ L'architecture applicative mentionnée fait partie de la couche éponyme.

² Notamment dans le premier livrable traitant de la déclaration de travail architectural à la section « 7. Architecture Cible ».

8.2. Simplicité

Un système simple permet de garantir la maintenabilité et donc la maîtrise des coûts. Il permet également, il est important de le souligner, d'assurer son bon fonctionnement.

Aussi est-il important de bien respecter ce principe, si l'on ne veut pas en perdre le contrôle, ce qui serait catastrophique pour les activités de l'entreprise.

8.3. Maintenabilité

Permettre d'assurer l'évolution de la plateforme sous toutes ses formes³, et que la version de production soit la plus optimale qu'il est possible de l'être pour permettre la bonne exécution des activités de l'entreprise. C'est un critère important à ne pas négliger dans la mesure où celui-ci peut fortement impacter les autres critères, et avec les conséquences inhérentes portant atteintes aux activités de l'entreprise, voire la raison d'être de cette dernière.

8.4. Disponibilité

La disponibilité est un enjeu majeur pour Foosus au vu des attentes en termes de besoin et de niveau de service définis en amont.

En effet, assurer la disponibilité de la plateforme c'est avant tout s'assurer de la bonne continuité de service des activités de l'entreprise. Et pour ce faire, il est important de garantir aux différents utilisateurs l'accès aux fonctionnalités / services de la plateforme.

8.5. Performance

À l'instar du critère de disponibilité, dont il est fortement lié, la performance est également un enjeu majeur pour Foosus au vu des attentes en termes de besoin et de niveau de service définis en amont. En effet, assurer la performance de la plateforme c'est avant tout s'assurer de la pérennité des activités de l'entreprise. Et pour ce faire, il est important de garantir aux différents utilisateurs de la plateforme un environnement fiable leur permettant d'accomplir leurs activités.

8.6. Sécurité

Il s'agit d'un critère très important pour toute entreprise et de loin, par conséquent, l'une de ses priorités si ce n'est la priorité. En effet les données étant au cœur des activités de l'entreprise, elles ont donc une grande valeur pour cette dernière, représentant un aspect financier non négligeable, et qu'elle est tenue de préserver de l'extérieur afin de ne pas nuire à ses activités.

Aussi à ce titre, l'entreprise se doit d'avoir un système garantissant tout à la fois la sécurité des données :

- personnes des utilisateurs,
- de ses activités.

³ Cela inclut tout à la fois, l'évolution par les projets, que celle intervenant en production par l'impérieuse nécessité d'éradiquer tout problème survenant sur la plateforme.

8.7. Accessibilité

Ce critère est un polysème traitant de deux sujets :

1. Accessibilité de la plateforme par les utilisateurs.
2. Accessibilité des données métier dans le cadre de leurs activités, appelée communément "Business Intelligence" ou BI.

Le premier fait référence au besoin et niveau de service définis attendu par les utilisateurs concernant l'accessibilité de la plateforme quel que soit la situation.

Tandis que le second se réfère aux attentes des métiers d'accéder aux données de production dans le cadre de leurs activités, à des fins de suivi, compréhension et amélioration du business.

9. Spécifications pour l'implémentation

Les spécifications pour l'implémentation doivent répondre aux lignes directrices pour l'implémentation.

D'une manière générale, la mise en place d'une gouvernance IT composé des différents référents de chaque domaine (*architecture logicielle, infrastructure, ...*) pour la production logicielle, avec pour mission de :

- définir les différents référentiels ;
- contrôler la conformité vis-à-vis desdits référentiels ;
- définir les processus IT nécessaires ;
- assurer les moyens nécessaires à ladite production.

9.1. Évolutivité

La définition d'un référentiel des bonnes pratiques de conception et d'implémentation.

La mise en place d'une chaîne de production efficiente éliminant la perte de temps à produire des évolutions pour la plateforme. Aussi pour ce faire, cette chaîne de production doit inclure :

- les tests unitaires et d'intégration automatisés ;
- le build continue ;
- et le déploiement automatisé.

9.2. Simplicité

L'emploi de standards de conception et technologique permet de construire un système simple, car ils sont généralement, à défaut d'être parfaitement maîtrisé, relativement bien compris par la majorité des professionnels ; d'où leur utilisation du reste.

En outre, il est également possible de s'inscrire dans un écosystème, ou à défaut dans une pile technologique cohérente.

9.3. Maintenabilité

À l'instar du critère de simplicité, l'emploi de standards de conception et technologiques restent le meilleur atout pour répondre à celui de maintenabilité. À cela s'ajoute également un référentiel de production et la nécessité de définir des processus IT de production, et notamment ceux traitant de la reprise sur panne. Sans oublier la capitalisation documentaire et une veille technique quotidienne des solutions adoptées pour se tenir informé, ce qui est très utile notamment en cas de faille de sécurité.

9.4. Disponibilité

Afin de satisfaire le niveau de service attendu, entre autres pour rappel une disponibilité 24/7 de la plateforme, il est nécessaire de définir un référentiel basé sur des standards éprouvés (*encore eux !*), définissant la conformité attendue.

Celui-ci peut se baser sur les principes de redondances par l'adoption du pile technique définissant une redondance sur chacun des composants du système défini dans le premier livrable traitant de la déclaration de travail architectural à la section « **7.3. Couche Infrastructure** ». Ce qui fait que l'on aura une redondance sur :

- chacun des serveurs : Frontend, d'entrée et équilibrage et Backend ;
- chacune des machines virtuelles portant le(s) composant(s) de service : Authentification, Compte utilisateur, Firewall, Système de gestion commerciale, ...

À cela s'ajoute la mise en place d'une supervision et d'indicateur de qualité de service, pour s'assurer d'un retour du service rendu.

9.5. Performance

Afin de satisfaire le niveau de service attendu, entre autres pour rappel la plateforme doit absorber le trafic et gérer les augmentations de charges, tout en supportant une augmentation quotidienne de 10% des adhésions d'utilisateurs.

Une nouvelle fois la définition d'un référentiel basé sur les standards et normes actuelles, et préconisant notamment l'emploi d'une architecture orientée service, ainsi que des principes pour l'implémentation technique. Sans oublier de définir des processus IT inhérents à la performance. À cela s'ajoute la mise en place d'une infrastructure de production standardisée de type PaaS⁴ supervisée et avec des indicateurs de performance (*KPI QoS*).

9.6. Sécurité

La mise en place d'un référentiel de sécurité basé sur les standards et normes actuelles définissant :

- la vision stratégique à adopter ;
- les processus inhérents, et notamment pour :
 - limiter la vulnérabilité et la compromission du système ;
 - réagir et savoir quoi faire en cas de problème de vulnérabilité du système.
- les parties prenantes et leur(s) responsabilité(s).

L'adoption d'outils, de solutions conformément aux principes des standards et normes actuelles, pour vis-à-vis du système :

- le préserver de toute faille ou vulnérabilité de sécurité ;
- le superviser ;
- lui permettre de communiquer de manière sécurisée ;
- lui permettre d'assurer la continuité des activités de l'entreprise ;
- ...

À cela s'ajoute la mise en place d'une supervision et d'indicateurs de suivi (*KPI QoS*).

⁴ PaaS pour Platform as a Service, est un service distant proposant une infrastructure et un environnement de serveurs applicatifs.

9.7. Accessibilité

Toujours la nécessité de mettre en place un référentiel basé sur les standards et normes actuelles définissant :

- l'organisation des utilisateurs, leur(s) rôle(s) et leur(s) attribution(s) vis-à-vis du système, id. qui accède à quoi ;
- les parties prenantes et leur(s) responsabilité(s) vis-à-vis des données, id. qui produit quoi et dans quel cadre ;
- les parties prenantes et l'usage qu'ils font des données, id. qui utilise quoi et à quelle fin ;
- ...

Sans oublier les processus IT inhérent à l'usage des données et à leur sécurité (*voir la section sur la sécurité vue plus haut*). Et à cela s'ajoute la mise en place d'une supervision et d'indicateurs de suivi des données (*KPI QoD*).

10. Standards pour l'implémentation

Les standards d'implémentation pour l'architecture de la nouvelle solution Foosus définissent un premier niveau de standards qui seront appliqués en vue de la réalisation. Ces standards seront donc à parfaire par la suite par l'instance de gouvernance IT.

10.1. Évolutivité

En ce qui concerne la conception, l'emploi d'une architecture microservices, dérivée directement de l'architecture Service-Oriented Architecture ou SOA, permettra, par sa structure de composants indépendants de par leurs attributions fonctionnelles, de pouvoir faciliter l'évolution la future plateforme.

Pour ce qui est de l'implémentation technique (*conception et développement*), outre l'emploi des principes **KISS**, **DRY**, **SOLID**, ..., l'emploi d'une pile technologique cohérente et Open source telle que : Java, Maven, Spring et AngularJS, participera à garantir l'évolutivité de la plateforme.

De plus, un outil de "versioning" tel que GitHub est impératif, sans quoi faire évoluer la plateforme sera compliqué.

Pour finir, sur les aspects de production, l'emploi d'un environnement containerisé Docker pour faciliter la mise en production.

10.2. Simplicité

En ce qui concerne la conception, l'application de patterns architecturaux correspondant à l'activité de l'entreprise, ici orienté service, permettent de construire un système simple. Aussi dans le cadre de ce projet, l'emploi d'un pattern de type Service-Oriented Architecture pour sa simplicité structurelle est requis.

Pour ce qui est de l'implémentation technique (*conception et développement*), l'emploi des principes **KISS**, **DRY**, **SOLID** et la règle des trois de **Martin Flower**, déjà explicité dans le premier livrable traitant de la déclaration de travail à la section « **5.5. Développement** », participe grandement à ce critère de simplicité.

À ces principes s'ajoutent ceux concernant le codage en lui-même, et notamment :

- Organiser
- Simplifier
- Réutiliser
- Réorganiser / optimiser
- Commenter & documenter

L'idéal serait de définir une charte d'implémentation reprenant tous les éléments susmentionnés.

10.3. Maintenabilité

Pour tout ce qui a attrait à l'organisation, la gouvernance IT :

- l'adoption de recommandations de l'ANSSI⁵, l'INRIA⁶, du CNRS⁷ et des différents consortium d'acteurs industriels, et ONG de la technologie et du numérique, et notamment tout ce qui tourne autour du libre / Open source ;
- la définition de processus de repise sur panne ;
- la standardisation des logs pour faciliter et réduire le temps d'investigation ;
- l'adoption d'un ensemble cohérent de solutions libre⁸.

En ce qui concerne les aspects techniques pour l'implémentation, l'emploi de solutions telles que Java, Maven, Spring et AngularJS.

Par ailleurs, une mise en place d'une solution de "ticketing", afin de référencer et suivre tout incident de production, et donc de permettre de maintenir et d'avoir une plateforme de services optimum pour les métiers / clients / utilisateurs.

10.4. Disponibilité et performance

Emploi d'un environnement PaaS⁹ de type Cloud Computing avec AWS, auquel viendra s'ajouter la virtualisation de serveur avec VMware et une containerisation avec Docker.

Ainsi qu'une solution de supervision et de "ticketing" Trac, afin de référencer et suivre tout incident de production.

⁵ ANSSI pour l'Agence Nationale de la Sécurité des Systèmes d'Information, dont la principale mission est d'œuvrer à la protection de ces dernières. Elle apporte son expertise et son assistance technique aux administrations et aux entreprises. Cependant elle crée une tribune appelée « Agora 41 » un espace de libre expression réunissant 41 personnalités d'expertise et discipline diverses, et publie les réflexions émergentes.

⁶ INRIA pour Institut National de Recherche en Informatique et en Automatique, qui a pour mission le développement de la recherche et de la valorisation en sciences et techniques de l'information et de la communication.

⁷ CNRS pour Centre National de la recherche technologique, dont la mission principale est de coordonner les activités de recherche en vue d'en tirer le meilleur rendement.

⁸ Logiciel libre = Open source + éthique sociale, id. la liberté d'utiliser, de modifier et d'adapter, autrement dit s'approprier, un logiciel.

⁹ PaaS pour Platform as a Service, est un service distant proposant une infrastructure et un environnement de serveurs applicatifs.

10.5. Sécurité

Pour tout ce qui a attrait à l'organisation, la gouvernance IT doit impérativement se prémunir de tout problème de sécurité par :

- l'adoption de recommandations de l'ANSSI et des différents consortium d'acteurs industriels, et ONG de la technologie et du numérique, et notamment tout ce qui tourne autour de la cybersécurité ;
- la définition de processus de
 - mise en sécurisation du système en cas de détection de vulnérabilité ;
 - reprise sur panne ;
- la standardisation des logs pour faciliter et réduire le temps d'investigation ;
- l'authentification :
 - à multiples facteurs pour les interfaces utilisateurs ;
 - OAuth¹⁰ pour l'authentification au niveau de la passerelle.
- la sensibilisation :
 - des utilisateurs par la formation et les exercices ;
 - de l'ingénierie logicielle à la sécurisation de leurs développements ;
 - de l'ingénierie infrastructure à la sécurisation de celles-ci.

En ce qui concerne les aspects techniques, la sécurité passe par :

- la mise en place d'une passerelle pour séparer le back office du front office ;
- l'emploi de protocoles de communication sécurisés : HTTPS (SSL¹¹ ou TLS¹²), SSH¹³ (*tunneling*), ... ;
- l'emploi de chiffrement de données : AES-256¹⁴, RSA¹⁵, ...
- l'emploi d'un pare-feu ;

10.6. Accessibilité

Ce critère étant intimement lié à celui sur la sécurité, il en reprend donc les standards, en y ajoutant les UI/UX, qui permettront de définir des interfaces utilisateurs efficaces et répondant aux besoins de leurs activités.

¹⁰ OAuth est un protocole libre permettant d'autoriser un site Web, un logiciel, ou une application dite "consommateur" l'utilisation de l'API sécurisée d'un autre site Web dit "fournisseur".

¹¹ SSL pour Secure Sockets Layer ou couche de sockets sécurisée est un protocole de sécurisation des échanges par réseau informatique.

¹² TLS pour Transport Layer Security ou sécurité de la couche de transports est un protocole de sécurisation des échanges par réseau informatique.

¹³ SSH pour Secure SHell est à la fois un programme et un protocole de communication sécurisé.

¹⁴ AES-256 pour Advanced Encryption Standard ou standard de chiffrement avancé est un algorithme de chiffrement symétrique d'une clé de 256 bits.

¹⁵ RSA nommé par les initiales de ses trois inventeurs : Ronald RIVEST, Adi SHAMIR et Leonard ADLEMAN est un algorithme de chiffrement asymétrique basé sur deux clés, l'une publique pour chiffrer et l'autre privée pour déchiffrer.

11. Conditions requises pour l'interopérabilité

L'interopérabilité aussi bien interne qu'externe du système est indispensable à son évolutivité.

Aussi le tableau ci-dessous représente le catalogue contenant l'ensemble des conditions requises pour l'interopérabilité.

IDENTIFIANT	INTITULÉ	DESCRIPTION
CR-IR-1	Format des échanges	Les interfaces API Web seront construites selon le principe de conception REST.
CR-IR-2	Format des données	Le format des données en transit sera basé sur le langage JSON.
CR-IR-3	Format de l'encodage	L'encodage de tous les documents et chaîne de caractère sera au format UTF-8, le standard permettant d'assurer une haute compatibilité entre les fichiers informatiques aussi bien internes qu'externes.
CR-IR-4	Compatibilité	La compatibilité avec l'ancien système (<i>format des échanges, nomenclatures, ...</i>) doit être maintenu tant que faire ce peut, afin de faciliter la migration. Aussi pour ce faire, des interfaces de conversions pourront être déployés le cas échéant.

12. Conditions requises pour le management du service IT

Le cadre de management du service IT vise à présenter les grandes lignes directrices de la collaboration entre les pôles de compétences dudit service.

Afin de respecter et s'inscrire plus avant dans la culture Lean de la société Foosus, cette méthodologie sera appliquée au domaine des systèmes d'information et donc à l'IT, avec le Lean IT, que ce soit pour le développement d'applications, le management de projets, l'infrastructure, ... dont le but, tout comme l'est globalement la méthode Lean, reste l'élimination de toute perte dans le travail et qui n'apporte aucune « valeur métier » à un produit ou à un service.

Pour information, ou tout du moins rappel, le Lean IT repose sur cinq principes :

1. L'identification des processus IT et détermination de la valeur ajoutée du point de vue des clients.
2. Définir la chaîne de valeur ajoutée, appelée Value Stream Mapping (VSM).
3. Assurer le flux continu des produits, services et informations, de bout en bout en éliminant tout gaspillage.
4. Le client est le demandeur, aussi rien n'est fait en amont du processus tant que celui-ci ne montre pas ses besoins en aval.
5. Mettre en œuvre une amélioration continue pérenne.

Une démarche Lean IT peut être déclinée sous deux formes, permettant la déclinaison de la stratégie de l'entreprise :

1. **L'amélioration avec rupture.**
2. **L'amélioration continue.**

Aussi afin de correspondre à la fois à l'organisation et aux attentes de la société, les deux approches seront utilisées. La première pour l'adoption d'une organisation plus efficiente et des processus inhérents adaptés. Puis, par l'amélioration continue visant à faire fructifier l'existant par l'amélioration des processus en place, et donc de l'organisation inhérente.

Toutefois concernant la gouvernance IT, la démarche Lean IT implique la nécessité de changements dans les organisations, à savoir :

- des équipes plus autonome pour la résolution des problèmes au plus près du terrain ;
- des équipes pluridisciplinaires pour résoudre les problèmes en cycles courts.

12.1. Product management

La gestion du produit, au sens large du terme, est au cœur des activités de la société Foosus, eus égard à la nature orientée service de ces dernières. Ce rôle est dévolu au Responsable Produit ou Product Owner, qui est le garant des fonctionnalités vis-à-vis du métier / client ou leur représentant, le Business Owner, et qui a la charge de l'expression de besoin visant à faire évoluer le produit, ici la plateforme e-commerce de Foosus. Aussi pour ce faire, le Product Owner organise, priorise et suit, en accord avec à la fois avec le métier / client et l'équipe de développement, le travail pour la bonne réalisation des fonctionnalités à produire.

12.2. Gouvernance architecturale

La gouvernance d'architecture est assurée par l'Architecte Logiciel, lequel est accompagné par les différents référents techniques, et doit répondre aux fonctions de :

- définition de l'architecture ;
- définition des standards d'implémentation ;
- suivi et contrôle de la conformité ;
- veille technologique ;
- conception de solutions ;
- maintien, id. mise à jour, du référentiel d'architecture.

12.3. Production logicielle

La production logicielle est assurée par les différentes équipes techniques, qui doivent s'inscrire à la fois dans :

- la démarche Lean,
- la démarche Scrum, équipe technique pluridisciplinaires appelée « **Dev. Team** ».
- les standards d'implémentation définis par l'Architecte logiciel.

Dans le cadre du présent projet, la Dev. Team, aura bien évidemment la tâche de développer l'ensemble des composants de la nouvelle architecture de la plateforme e-commerce de Foosus. Et qu'elle aura à le faire dans un cadre de travail quotidien (Daily Scrum) et itératif (Sprint).

12.4. Infrastructure

L'infrastructure a pour rôle de mettre en place tous les moyens techniques nécessaires, afin de supporter et permettre la mise en place de l'architecture logicielle.

À l'instar de la gouvernance architecturale, l'infrastructure est assurée également par un Architecte inhérent aux fonctions attendues, présentement l'Architecte Infrastructure. Cependant, dans des organisations à taille contenue, ce rôle est généralement dévolu à l'Architecte Logiciel, qui en plus de ses propres attributions se verra également confié les fonctions suivantes, afin de répondre aux attentes / exigences de l'architecture logicielle :

- définir l'architecture de l'infrastructure ;
- définir les standards d'infrastructure ;
- l'application ou tout du moins de la prise en compte de standards de l'architecture logicielle ;
- suivre et contrôler la conformité ;
- faire de la veille technologique ;
- maintien, id. mise à jour, du référentiel infrastructure.

12.5. Exploitation

L'exploitation, appelée également « production », vise à ce que l'ensemble, architecture logicielle et infrastructure, fonctionne correctement et ce de manière cohérente conformément aux attentes définies en amont. Mais aussi de s'assurer que toute nouvelle mise en production ne vienne pas mettre en péril cet ensemble.

Aussi pour ce faire, l'équipe de production / exploitation, au travers de son référent / responsable exploitation, va mettre en place une supervision avec des outils dédiés. Dès lors qu'un souci est détecté en production, de manière proactive ou réactive, l'équipe en charge de la supervision va immédiatement le remonter aux instances concernées.

13. Contraintes

Le tableau ci-dessous représente le catalogue contenant l'ensembles des contraintes projet transmises par la société pour le chantier d'architecture.

IDENTIFIANT	INTITULÉ
CTR-PRJ-1	Le projet est approuvé pour un coût de 50 000 USD (45 190 €).
CTR-PRJ-2	Le projet est approuvé pour un délai de 6 mois.
CTR-PRJ-3	L'architecture doit permettre d'obtenir le meilleur rapport qualité-coût.
CTR-PRJ-4	Pas de mise à jour ou de reprise de la plateforme actuelle. Celle-ci est conservée en mode maintenance uniquement.
CTR-PRJ-5	L'architecture peut inclure de nouveaux composants personnalisés ou des composants du commerce pour favoriser la flexibilité, la stabilité et l'extensibilité.
CTR-PRJ-6	Les solutions open source sont préférables aux solutions payantes.
CTR-PRJ-7	Le support continu des composants doit être pris en compte lors de leur sélection ou lors des prises de décision de création ou d'achat.
CTR-PRJ-8	Toutes les solutions du commerce ou open source doivent, dans la mesure du possible, faire partie d'une même pile technologique, afin de réduire les coûts de maintenance et de support continus.
CTR-PRJ-9	Les livrables doivent pouvoir être fournis à intervalles réguliers pour que le nouveau système soit rapidement opérationnel et puisse être doté de nouvelles fonctionnalités au fil du temps.
CTR-PRJ-10	Dissocier les nouvelles livraisons de l'architecture et de l'infrastructure existantes afin de limiter les interruptions de service.
CTR-PRJ-11	Respecter la culture du Lean de Foosus.

14. Hypothèses de solutions

Au regard des problématiques de la section « **4.3. Méthode des 5 pourquoi** » plus haut, des pistes de travail et attentes à considérer.

14.1. Taux d'inscriptions utilisateurs

Pour pallier à la chute d'inscriptions, nécessaires pour rassurer les investisseurs de la société :

- Ajouter une fonctionnalité de géociblage pour toucher une gamme plus large d'utilisateurs.
- Définir une architecture se mettant à l'échelle en fonction de la base clients.

14.2. Alignement commercial & technique

Pour pallier à la complexité de la plateforme et soutenir la croissance :

- Garder impliqué les équipes techniques et les guider vers une standardisation architecturale.
- Définir un périmètre architectural clair et concis.

14.3. Compétitivité concurrentiel

Pour pallier la dette technique, le manque d'extensibilité (*verrouillage*) et de l'instabilité de l'architecture :

- Définir une architecture évolutive / extensible, et avec un degré de granularité pertinent / justifié.
- Mettre en place des bonnes pratiques à la fois architecturales et techniques (*conception & développement*).

14.4. Alignement technico-commercial de la performance

Pour pallier à la problématique de visibilité et du manque de clarté de l'intelligence business, id. l'alignement entre la performance technique et commercial :

- La nouvelle architecture doit permettre de fournir les informations utiles à l'analyse de l'intelligence business.
- La plateforme quant à elle doit offrir des outils permettant l'analyse de l'intelligence business, tel que par exemple la possibilité de produire à la demande des tableaux de bord, ou bien encore des KPI ou Key Performance Indicator (*indicateurs clés de performance*).

14.5. La réputation de la marque

Pour pallier au manque de qualité des livraisons, les interruptions dues à ces dernières ainsi qu'aux pannes :

- Mettre en place un processus de qualité des livraisons pour éviter les pannes et donc les interruptions de service.
- Définir une architecture permettant lors des livraisons :
 - d'éviter lesdites interruptions de service, ou tout du moins les minimiser ;
 - de revenir en arrière en cas de problème.
- Définir une architecture se mettant à l'échelle en fonction de la base clients.