



Contrat de Conception et de Développement de l'Architecture

- *Projet* : Concevez une nouvelle architecture afin de soutenir le développement de votre entreprise
- *Client* : Foosus
- *Préparé par* : Henrick AGNAME

Table des matières

1. Introduction et contexte
2. La nature de l'accord
3. Objectifs et périmètre
4. Partie prenante, préoccupations et visions
5. Description de l'architecture, principes stratégiques et conditions requises
6. Reference aux conditions requises pour l'architecture
7. Livrables architecturaux
8. Plan de communication
9. Phases de livrables définies

Introduction et Contexte

Foosus est une entreprise dynamique qui s'est rapidement imposée sur le marché grâce à sa capacité à soutenir la consommation de produits alimentaires locaux et à faciliter la connexion entre les clients et les producteurs et artisans locaux. L'objectif de Foosus est d'améliorer la compétitivité de l'entreprise face aux géants du commerce électronique mondial en développant une nouvelle plateforme d'e-commerce axée sur la localisation géographique.

La plateforme actuelle de Foosus a connu une croissance rapide, mais elle a atteint un point où elle ne peut plus répondre efficacement aux projets d'expansion de l'entreprise. Par conséquent, il est devenu impératif de repenser l'architecture de la plateforme afin de garantir sa scalabilité, sa stabilité et sa capacité à répondre aux besoins des utilisateurs commerciaux de Foosus.

La Nature de l'accord

La nature du contrat d'architecture de développement fourni par Henrick AGNAME est un accord juridique et contractuel entre Henrick AGNAME, en tant que fournisseur de services de développement de logiciels interne, et Foosus qui veut développer une application basé sur une architecture de microservices. Cet accord définit les droits, les responsabilités, et les obligations des deux parties concernant l'architecture de développement, y compris la conception, la mise en œuvre, les tests, et le déploiement des microservices, ainsi que leur intégration dans des systèmes plus larges.

Il s'agit d'un contrat mutuellement bénéfique qui vise à établir une collaboration efficace entre Henrick AGNAME et Foosus, afin de créer une solution de logiciel basée sur des microservices qui est performante, fiable, et adaptée aux besoins spécifiques du client.

Les éléments clés de cet accord comprennent:

1. **Portée du Projet:** Définition claire de la portée du projet, y compris les spécifications des microservices à développer, les exigences fonctionnelles et non fonctionnelles, et les objectifs du projet.
2. **Responsabilités de Henrick AGNAME:** la responsabilités de Henrick AGNAME dans la conception de l'architecture, le développement des microservices, l'intégration, les tests, et le support après le déploiement. Cela inclut également l'assurance de la conformité aux normes de sécurité et de qualité.
3. **Responsabilités du Client:** Fournir des informations pertinentes, collaborer avec Henrick AGNAME lors de la définition des exigences, et participer aux revues de projet. Le client est également responsable de l'acceptation des livrables et de la communication en temps opportun.
4. **Droits de Propriété Intellectuelle:** Termes relatifs à la propriété des codes sources, des documentations, et de tout autre matériel développé dans le cadre du projet.
5. **Paielements et Rémunérations:** Conditions de paiement pour les services de développement, y compris les tarifs, les échéances de paiement, et les conditions en cas de modification de la portée du projet.
6. **Gestion des Risques et des Changements:** Processus de gestion des risques et des changements, y compris la manière dont les modifications de la portée du projet sont traitées et comment les risques sont identifiés et atténués.
7. **Confidentialité et Sécurité des Données:** Les dispositions concernant la confidentialité des informations sensibles partagées entre les parties, ainsi que les engagements en matière de protection des données.

8. **Résiliation et Conditions de Clôture:** Conditions sous lesquelles le contrat peut être résilié par l'une ou l'autre des parties, et les obligations à remplir en cas de clôture du projet.
 9. **Règlement des Litiges:** Mécanismes de règlement des différends en cas de désaccord entre les parties concernant l'exécution du contrat.
-

Objectifs et périmètre

Objectifs

<i>Objectif Business</i>	<i>Notes</i>
Maintien de la solution existante pendant le processus de réalisation de la nouvelle solution	Eviter la mise hors service de la plateforme et d'optimiser le taux d'adhésion des fournisseurs et des consommateurs.
Augmentation des campagne marketing	Mettre en place des campagnes marketing de grande envergure ciblées par zone géographique tout en supportant la charge.
Élasticité(scalabilité) de la pile technologique	Le système doit être capable de répondre aux besoins et aux exigences de la clientèle en constante évolution.
Reduction des indisponibilités de services lié au flux de connexion et aux demandes de service	Absorber les pics d'utilisateurs et le lancement de campagnes marketing
Amélioration de l'accès au système en cas de surcharge	En cas de surcharge du système, les utilisateurs connectés doivent avoir une accessibilité réduite ou dégradée, ce qui signifie que les services devraient fonctionner plus lentement, être moins réactifs ou présenter des limitations temporaires.

Une plateforme géographiquement disponible	Les particularités locales peuvent inclure les différences culturelles, linguistiques, réglementaires ou autres spécificités liées à une région ou un pays. La plateforme devra permettre une personnalisation en fonction des besoins spécifiques de chaque marché local pour répondre aux exigences des utilisateurs. De plus, les facteurs de réseau (connexion lente, haut débit) ne devraient pas entraîner une interruption de l'accès aux différents services.
Une disponibilité accrue	La plateforme devra rester disponible lors de la livraison de nouvelles versions ou lors de la modification du schéma de base de données.
Une expérience utilisateur commune et unique	Tous les utilisateurs devront bénéficier d'une expérience utilisateur identique, peu importe leur localisation géographique.
Réduction des pannes liées à l'intégration et aux tests de nouvelles versions.	Le temps nécessaire pour examiner et tester chaque nouvelle version dans un environnement de Pré-production devra être accéléré avant sa mise en production, afin de garantir une meilleure qualité et performance du produit final.
Améliorer les produits en fonction des besoins des utilisateurs et optimiser leur satisfaction tout en respectant les contraintes budgétaires.	Mettre en place un système de suivi et d'analyse de l'utilisation des solutions par les utilisateurs afin de mieux comprendre leurs besoins, ce qui permettra de faire évoluer les produits et de mieux répondre aux attentes des utilisateurs. Il convient également de mettre en place une plateforme de tests pour les nouveaux produits, permettant de recueillir des retours d'expérience proches de ceux des utilisateurs finaux, et ainsi d'améliorer les produits avant leur lancement sur le marché.

Périmètre

<i>Périmètre</i>	<i>Notes</i>
Recherche de fournisseurs alimentaires	L'architecture doit prendre en charge la fonctionnalité de recherche de fournisseurs alimentaires, en permettant aux utilisateurs commerciaux de trouver facilement et efficacement les producteurs et artisans locaux.
Tri des offres alimentaires	L'architecture doit inclure un processus de tri des offres alimentaires, permettant aux utilisateurs commerciaux de sélectionner et d'ajouter des produits alimentaires au panier, de rechercher des accords de paiement et de spécifier les instructions de livraison.
Intégration de prestataires de paiement tiers	L'architecture doit permettre l'intégration transparente de prestataires de paiement tiers, afin de faciliter les transactions entre les utilisateurs commerciaux et les fournisseurs alimentaires.
Communication avec les fournisseurs alimentaires	L'architecture doit prendre en compte la communication entre les utilisateurs commerciaux et les fournisseurs alimentaires, en incluant des fonctionnalités telles que l'envoi d'instructions de livraison et l'émission de factures de commission par e-mail.
Évolutivité du système	L'architecture doit être conçue de manière à pouvoir évoluer et se développer au fur et à mesure de la croissance de Foosus, en permettant le déploiement de nouveaux services et fonctionnalités dans différentes régions géographiques.
Facilité d'utilisation	L'architecture doit garantir une expérience utilisateur conviviale et intuitive pour les

	utilisateurs commerciaux, en tenant compte des besoins spécifiques de leur utilisation professionnelle.
Sécurité des données	L'architecture doit intégrer des mesures de sécurité robustes pour protéger les données des utilisateurs commerciaux, en assurant la confidentialité et l'intégrité des informations sensibles et respect des RGPD.
Performance du système	L'architecture doit permettre des performances optimales du système, en garantissant une réponse rapide aux requêtes des utilisateurs commerciaux et en minimisant les temps d'indisponibilité du service.
Collaboration interne	L'architecture doit faciliter la collaboration entre les différentes équipes de Foosus, notamment les équipes business, design et développement, en permettant une intégration transparente des processus et une communication efficace.

Parties prenantes, préoccupations et visions

Partie prenante	Préoccupation	Vision
Natasha Jarson, CIO Daniel Antony, CPO Jo Kumar CFO	Innover dans le périmètre d'une Architecture d'Entreprise	Il sera nécessaire d'établir des processus de travail clairs et de capitaliser sur les données et les ressources pour soutenir l'innovation au sein de l'architecture d'entreprise et assurer la réussite à long terme de l'entreprise.

CMO, Natasha Jarson CIO, Daniel Antony CPO, Jo Kumar CFO	Soutenir l'innovation technique rapide et l'expérimentation	Mise en place d'une architecture micro-service avec des services à responsabilités uniques, standardisation des processus (pratiques agiles), mise en place de l'intégration continue et du déploiement continu, et choix d'un stack technique optimal pour un hébergement cloud.
CMO, CPO, Jack Harkness COO (Directeur des opérations)	Visibilité de la plateforme	<ul style="list-style-type: none"> • Visibilité sur l'utilisation des logiciels : mettre en place des outils de suivi et d'analyse pour comprendre comment les utilisateurs interagissent avec les logiciels, afin d'identifier les points d'amélioration et d'optimiser l'expérience utilisateur. • Capacité d'inversion des décisions d'architecture avec un coût minimal : mettre en place une architecture micro services qui permet de remplacer ou de modifier un composant sans impacter l'ensemble du système. • Réplication sur une plateforme pour tester de nouveaux produits : mettre en place un environnement de test qui reproduit l'infrastructure de production et permet d'expérimenter de nouvelles fonctionnalités de produits ou de services en toute sécurité. Cela servira à valider les innovations avant de les déployer sur la plateforme principale, garantissant ainsi leur compatibilité avec les objectifs commerciaux fondamentaux
Ash Callum, CEO	Taux d'inscription de nouveaux	<ul style="list-style-type: none"> • Le nouveau système devra être disponible partout dans le monde et fournir une expérience utilisateur

Jo Kumar CFO	utilisateurs	<p>similaire, peu importe la localisation de l'utilisateur.</p> <ul style="list-style-type: none"> Le nouveau système devra fournir une fonctionnalité de géolocalisation. Le nouveau système devra être scalable et s'adapter à l'évolution de la base de clientèle (mise à l'échelle).
TODO	Améliorer la réputation de Foosus sur le marché grâce à la stabilité.	<ul style="list-style-type: none"> Mise en place de tests automatisés (tests unitaires, d'intégration et de bout en bout). Automatisation du processus de déploiement et d'intégration de nouvelles fonctionnalités et correctifs pour minimiser les erreurs humaines. Tester les nouvelles versions du logiciel avec un petit groupe d'utilisateurs avant de les déployer sur l'ensemble des utilisateurs. Mettre en place un processus de gestion des incidents pour traiter rapidement les problèmes qui se produisent, ce qui inclut la détection, la communication et la résolution des incidents, ainsi qu'une capitalisation de ces données pour éviter que des problèmes similaires ne se reproduisent. Mise à l'échelle horizontale de l'architecture afin de supporter la charge due à l'augmentation du nombre d'utilisateurs et de zones géographiques sans impacter le service. Mise en place d'un déploiement séquentiel par région, ce qui permettra de détecter puis résoudre d'éventuels problèmes potentiels sans affecter

		l'ensemble des utilisateurs.
--	--	------------------------------

Description de l'architecture, principes stratégiques et conditions requises

Description

L'architecture microservices proposée pour l'entreprise sera composée de services indépendants qui sont déployés comme des unités autonomes, chacun réalisant une fonction spécifique. Ces services communiqueront entre eux via des API bien définies. Les composants principaux de cette architecture incluent :

1. **Infrastructure Runtime:** Des conteneurs seront utilisés pour exécuter chaque microservice de manière isolée, assurant ainsi une gestion efficace des ressources et un déploiement rapide.
2. **Hébergement sur le Cloud:** Les microservices seront hébergés dans un environnement cloud pour bénéficier d'une mise à l'échelle automatique, d'une haute disponibilité et d'une réduction des coûts d'infrastructure.
3. **Extension de Base de Données avec Clusterisation:** Au lieu d'utiliser une base de données distincte pour chaque microservice, une base de données centralisée sera mise en place. Cette base de données sera clusterisée sur 3 nœuds pour assurer la haute disponibilité et la tolérance aux pannes.
4. **Équilibrage de Charge entre Base de Données et Microservices:** Un équilibreur de charge sera déployé entre les microservices et la base de données pour distribuer efficacement les requêtes et optimiser les temps de réponse. Cela permettra également de réduire les points de contention et d'améliorer l'évolutivité.
5. **Interfaces Front-end:** Des interfaces utilisateur seront développées pour faciliter l'interaction avec les systèmes de recherche, de commande, d'inventaire, et autres.
6. **Système d'Orchestration:** Pour gérer, surveiller et mettre à l'échelle les microservices, un système d'orchestration comme Kubernetes sera utilisé.
7. **Systèmes Fonctionnels:** Comprend des microservices spécifiques tels que le système d'inventaire, le système de commande, le système de recherche, le système de facturation, et le système de mail.
8. **Sécurisation des Transactions:** Des mécanismes de sécurité tels que TLS, authentification et autorisation seront mis en place pour sécuriser les transactions et protéger les données sensibles.
9. **Système d'Analyse de Données:** Un système d'analyse sera utilisé pour collecter et analyser les données, permettant une meilleure prise de décision et l'optimisation des services.

Principes stratégiques

1. **Autonomie des Services:** Chaque microservice devrait être indépendant, capable de fonctionner et de se mettre à l'échelle de manière autonome.
 2. **Découplage:** Les microservices devraient être faiblement couplés, avec des dépendances minimales entre eux, pour permettre une flexibilité et une maintenabilité maximales.
 3. **Réactivité et Performance:** Les microservices devraient être capables de répondre rapidement aux demandes et de gérer efficacement les ressources pour maintenir de hautes performances.
 4. **Sécurité par Conception:** La sécurité doit être intégrée à chaque étape du développement et du déploiement des microservices.
 5. **Intégration Continue / Déploiement Continu (CI/CD):** Adopter des pratiques CI/CD pour assurer des mises à jour rapides et fiables.
 6. **Observabilité:** Mettre en place une surveillance, des journaux et des traces pour faciliter le débogage et la surveillance de l'état de santé des microservices.
-

Conditions Requises:

1. **Infrastructure Adéquate:** Avoir accès à une infrastructure cloud ou sur site qui puisse soutenir la mise en œuvre
-

Référence aux Conditions requises pour l'architecture

1. **TOGAF (The Open Group Architecture Framework) :** Utilisation de TOGAF comme cadre de référence pour la méthodologie d'architecture d'entreprise. Il fournit une approche structurée pour le développement, la maintenance et la gouvernance de l'architecture d'entreprise.
2. **ITIL (Information Technology Infrastructure Library) :** Utilisation des meilleures pratiques ITIL pour la gestion des services informatiques. ITIL permet de garantir l'alignement des services informatiques sur les besoins de l'entreprise et d'améliorer la qualité du service.
 1. **Stratégie de Service:** L'entreprise doit d'abord évaluer ses besoins et ses objectifs. Cela peut inclure l'expansion du marché, l'amélioration de l'expérience utilisateur, ou l'augmentation de la vitesse de déploiement. Les services qui seront offerts à travers le nouveau système doivent être définis et alignés avec les

objectifs de l'entreprise.

2. Conception de Service: L'entreprise conçoit l'architecture en gardant à l'esprit les exigences de l'entreprise, les SLAs, la sécurité, la conformité, et les besoins de performance. Par exemple, pour l'extension de la base de données, ils pourraient choisir une base de données distribuée qui peut être mise à l'échelle horizontalement.

3. Gestion de la transition de services:

- **Gestion des actifs et des configurations:** Avant le déploiement, un système de gestion de la configuration (CMDB) est mis en place pour documenter tous les microservices et leurs configurations.
- **Gestion des changements:** L'entreprise adopte une approche de déploiement graduel pour minimiser les risques. Les microservices sont d'abord déployés dans un environnement de test ou de staging.
- **Évaluation et validation du service:** Les tests sont effectués pour s'assurer que les microservices fonctionnent comme prévu et répondent aux exigences définies.

4. Opération de Service:

- **Gestion des incidents:** Un processus de gestion des incidents est mis en place pour assurer une réponse rapide en cas de problèmes avec l'un des microservices, tels que des problèmes d'interface ou de performance.
- **Gestion des événements:** Les outils de surveillance sont utilisés pour surveiller en temps réel les microservices et les infrastructures sous-jacentes, et pour déclencher des alertes ou des actions automatiques en cas d'événements spécifiques.
- **Gestion des requêtes et des accès:** Des procédures sont mises en place pour gérer les requêtes des utilisateurs et pour contrôler l'accès aux services, notamment la sécurisation des transactions.

5. Amélioration continue des services:

- **Feedback et Analyse de données:** Après le lancement du nouveau système, l'entreprise collecte des commentaires des utilisateurs et utilise des outils d'analyse de données pour évaluer les performances du système.
- **Optimisation et améliorations:** Sur la base des données collectées et des retours d'expérience, des améliorations continues sont apportées aux microservices. Par exemple, si l'interface de recherche n'est pas aussi réactive que souhaité, des optimisations peuvent être apportées.

3. **Agile/Scrum** : Utilisation des méthodologies agiles, telles que Scrum, pour favoriser l'innovation, la collaboration et la flexibilité dans le développement de projets.
4. **DevOps** : Application des principes DevOps pour faciliter l'intégration continue et le déploiement continu des services et des applications, améliorant ainsi l'efficacité opérationnelle et la qualité des livrables.
5. **ISO/IEC 27001**: Il s'agit d'une norme internationale pour la gestion de la sécurité de l'information (ISMS). Elle établit les exigences pour la mise en place, la

maintenance et l'amélioration continue d'un système de gestion de la sécurité de l'information.

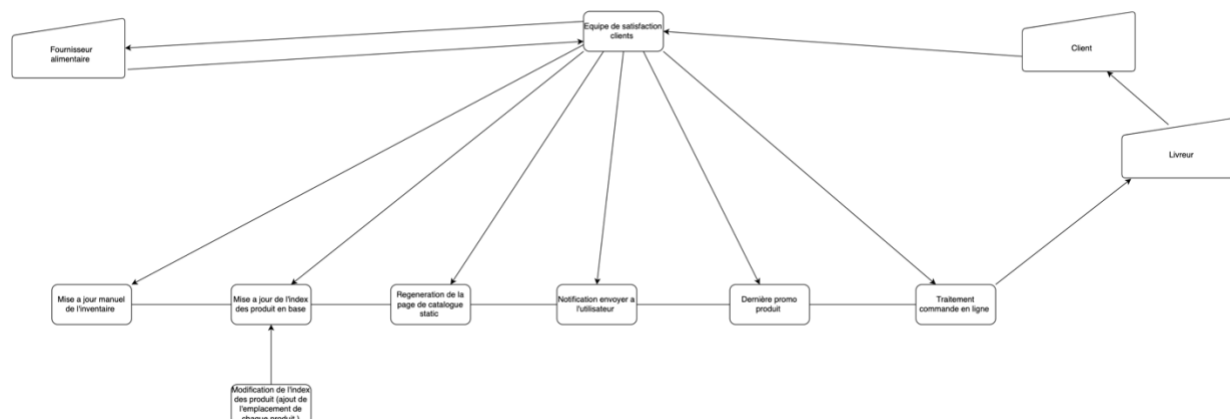
6. **NIST (National Institute of Standards and Technology) Cybersecurity Framework:** Ce cadre fournit des lignes directrices pour la gestion des risques liés à la cybersécurité et la protection des infrastructures critiques. Il vise à aider les organisations à mieux comprendre, gérer et réduire les risques liés à la cybersécurité.
7. **OWASP (Open Web Application Security Project):** L'OWASP est une organisation à but non lucratif qui vise à améliorer la sécurité des logiciels. Ils fournissent des outils, des ressources et des bonnes pratiques pour aider les organisations à sécuriser leurs applications web.
8. **GDPR (General Data Protection Regulation):** Le GDPR est un règlement de l'Union européenne qui vise à protéger les données personnelles des citoyens de l'UE. Il impose des exigences strictes en matière de protection des données et de notification en cas de violation de données.
9. **HIPAA (Health Insurance Portability and Accountability Act):** Pour les organisations du secteur de la santé, la conformité à la loi américaine HIPAA est essentielle pour protéger les informations de santé et les données des patients.

Livrables architecturaux

Livrables architecturaux qui satisfont aux conditions requises pour le business.

Développement de l'architecture

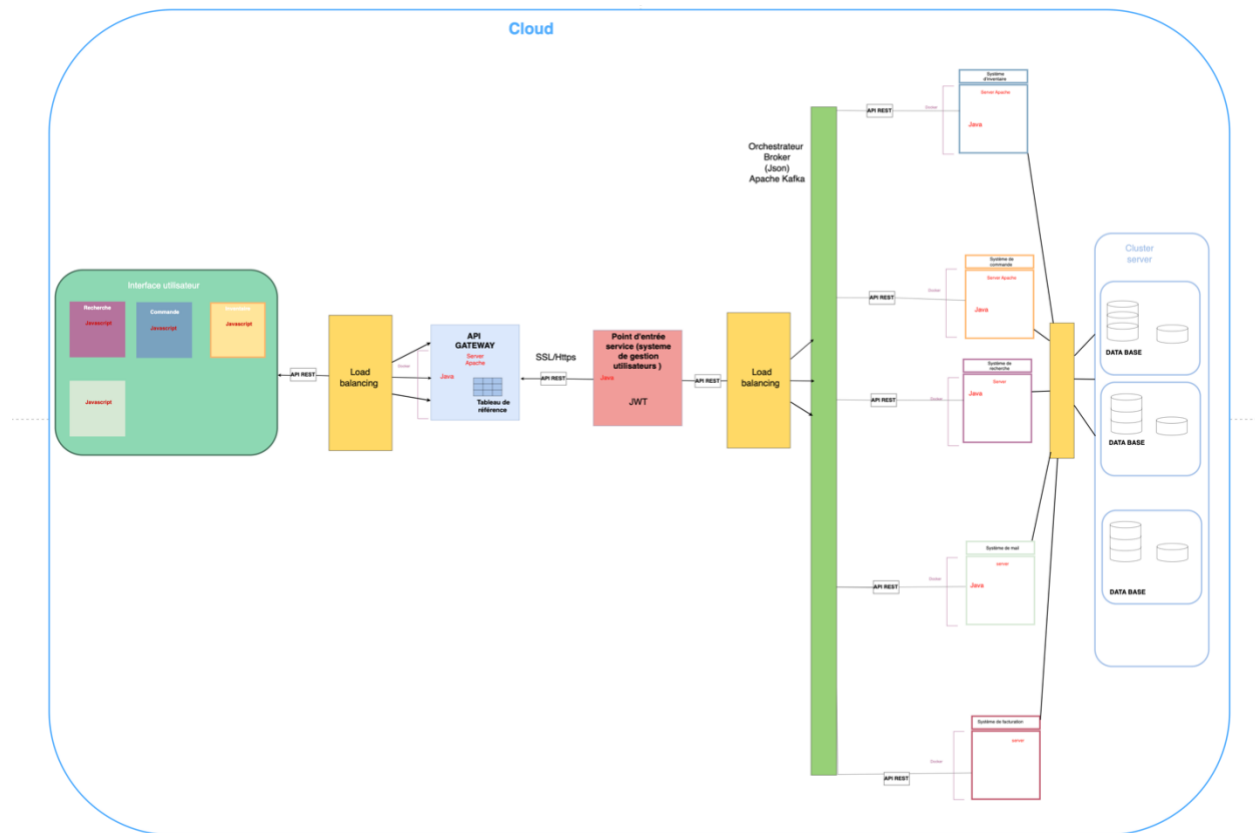
Ce référer au document de vision d'architecture.



La modification de l'index des produits en base de données sera nécessaire pour optimiser la

chaîne d'approvisionnement. En effet l'emplacement des produits sera ajouter à l'index des produits en base de données cela permettra au système de savoir exactement ou se trouve chaque produit. Ainsi Lorsqu'un client se connectera à la plateforme, le système récupérera ces informations de géolocalisation (avec son consentement), puis fera une compare entre son emplacement et celui du stock de chaque produit. Ensuite, le système générera une liste de produits qui sont le plus proche de l'emplacement du client. Cette liste sera affichée sur la page du catalogue lorsque le client consulte la plateforme.

Schéma de la nouvelle architecture :



Plan de communication

Décrit le rythme, les canaux et les formats approuvés utilisés pour la communication

Évènements

- Lancement du projet
- Réunion de suivis hebdomadaire
- Revue du projet à mi-parcours
- Déploiement en production
- Réunion de clôture et de retour d'expérience

Canaux

- E-mails pour les communications officielles
- Plateforme de messagerie interne (ex: Skype) pour discussions en temps réel vidéoconférences
- Jira

Formats

- Courriels pour les annonces
- Présentations PowerPoint pour les réunions
- Documentations écrites et tutoriels pour les formations

Contenu

- Présentation de la vision architecturale
- Chronologie du projet et jalons clés
- Rôles et responsabilités des membres de l'équipe
- Rapports de l'état d'avancement du projet
- Formation sur les outils et technologies utilisés
- Feedback et suggestions des équipes

Rythme de communication

- Annonce initiale - Une fois, lors du lancement du projet.
- Réunions de suivi - Hebdomadaires.

Phases de livrables définies

Roadmap

