

Spécification des Conditions Requises pour l'Architecture



Projet : Foosus

Client : Foosus

Sommaire

| | |
|---|----------|
| Objet de ce document | 3 |
| Conditions requises pour l'architecture | 3 |
| Lignes directrices pour l'implémentation | 4 |
| Mesures du succès | 4 |
| Contrat de service business | 5 |
| Accords de niveau de service | 5 |
| Contrat de service application | 5 |
| Objectifs de niveau de service | 5 |
| Indicateurs de niveau de service | 6 |
| Spécifications pour l'implémentation | 6 |
| Principes et objectifs | 6 |
| Standards pour l'implémentation | 7 |
| Conditions requises pour l'interopérabilité | 7 |
| Conditions requises pour le management du service IT | 8 |
| Contraintes | 8 |
| Hypothèses | 9 |
| Architecture orientée service | 9 |
| Comparaison des fournisseurs de services | 9 |
| Coûts | 9 |
| Performances | 10 |
| Cycle de vie de l'implémentation de l'architecture | 11 |

Objet de ce document

La Spécification des Conditions Requises pour l'Architecture fournit un ensemble de déclarations quantitatives qui dessinent ce que doit faire un projet d'implémentation afin d'être conforme à l'architecture.

Une Spécification des Conditions Requises pour l'Architecture constitue généralement un composant majeur du contrat d'implémentation, ou du contrat pour une Définition de l'Architecture plus détaillée.

Comme mentionné ci-dessus, la Spécification des Conditions Requises pour l'Architecture accompagne le Document de Définition de l'Architecture, avec un objectif complémentaire.

Le Document de Définition de l'Architecture fournit une vision qualitative de la solution et tâche de communiquer l'intention de l'architecture.

La Spécification des Conditions Requises pour l'Architecture fournit une vision qualitative de la solution, énumérant des critères mesurables qui doivent être remplis durant l'implémentation de l'architecture.

Conditions requises pour l'architecture

L'architecture actuelle ne peut plus soutenir la croissance de l'entreprise. C'est pour cela qu'il faut concevoir une nouvelle architecture. Cette partie sera le listing des conditions requises pour cette nouvelle architecture.

La nouvelle architecture doit réduire les risques techniques au maximum afin d'assurer une stabilité des systèmes d'informations.

L'architecture doit assurer la continuité des services en limitant les risques d'interruption afin d'améliorer la réputation de l'entreprise qui a été trop souvent entachée par des interruptions.

L'architecture doit être évolutive. Pour que l'application soit compétitive, il est important de pouvoir ajouter des nouvelles fonctionnalités facilement et rapidement, tout en évitant les interruptions de service.

La nouvelle architecture doit permettre l'augmentation des taux d'adhésion des utilisateurs et des produits alimentaires. Car ces métriques sont très importantes aux yeux des investisseurs, c'est avec ces dernières qu'ils évaluent la valeur de Foosus.

L'architecture doit mettre en place des mesures de bonnes pratiques qui vont permettre aux développeurs d'être guidés en gardant une certaine liberté.

Lignes directrices pour l'implémentation

Il est important de garder en vue les principes d'architectures TOGAF du projet qui sont les suivants :

- Principe 1 : Primauté des principes
- Principe 2 : Maximiser les avantages pour l'entreprise (rentabilité)
- Principe 4 : Continuité des activités
- Principe 19 : Gestion réactive du changement

Ces principes vont servir de pilier pour la prise de décision lors de l'implémentation de l'architecture.

Le respect des accords de niveau de service est une priorité absolue, qui passe par l'atteinte des objectifs de niveau de service.

Mesures du succès

La mesure du succès du projet se fera par rapport aux indicateurs de réussite suivant :

- Le nombre d'adhésions d'utilisateurs par jour doit croître de 10% dans les six mois suivants le déploiement de la nouvelle solution. Pour vérifier cet indicateur, la moyenne du nombre d'adhésion par jour sera calculée tous les mois pour voir s'il y a une augmentation et si elle tend vers 10%.
- Il faut que le taux d'adhésion de producteurs alimentaires passe de 1,4 par mois à 4 par mois dans les six mois après le déploiement de la nouvelle solution.
- Réduction des délais moyens de parution des mises à jour de l'application de 3,5 par semaine à moins d'une semaine dès le déploiement de la nouvelle solution.
- Le taux d'incidents en Production doit être drastiquement réduit, dès le déploiement de la nouvelle solution, de plus de 25 par mois à moins de 1 par mois. Chaque incident doit être sujet à un rapport d'erreur détaillé identifiant les causes de l'incident. Ainsi, il sera facile de quantifier les incidents et d'éviter la récurrence.

Si tous ces indicateurs sont vérifiés, on pourra alors définir le projet comme une réussite.

Contrat de service business

Accords de niveau de service

Les investisseurs mesurent la valeur de Foosus sur sa capacité à maintenir un taux positif d'inscriptions de nouveaux utilisateurs. Cette métrique a chuté rapidement au cours des derniers mois et doit être améliorée en priorité.

La chute du nombre d'inscriptions est en partie dû à la mauvaise image que porte l'application à cause des soucis d'interruption de service. Il est donc très important que la nouvelle architecture prévienne ces interruptions.

Ainsi, l'image de l'application s'en verra grandir, ce qui sera un premier pas vers l'augmentation du taux d'inscription.

La diminution d'interruptions de services passe par la réduction d'accidents en production. Pour cela, il faut mettre en place des tests efficaces en amont pour limiter les problèmes lors des déploiements.

Pour limiter les interruptions, il faut que la nouvelle architecture mette en place un système permettant de mettre à jour l'application en gardant les services en ligne.

Les mises à jour doivent être de petite taille, mais régulières. Des mises à jour de taille modérée et régulière vont aussi favoriser la diminution des erreurs en production.

Contrat de service application

Objectifs de niveau de service

- Objectif 1 : Le nombre d'adhésions d'utilisateur par jour doit augmenter de 10%.
- Objectif 2 : Le taux d'adhésion de producteur alimentaire doit passer à quatre par mois.
- Objectif 3 : Le délai moyen de parution de mise à jour doit être inférieur à une semaine.
- Objectif 4 : Le taux d'incident en production doit être inférieur à 1 par mois.

Indicateurs de niveau de service

Les indicateurs de niveau service qui permettront de valider la réussite des objectifs de niveau de service sont les suivants :

- Le nombre d'adhésions d'utilisateur par jour.
- Le taux d'adhésion de producteur alimentaire.
- Le délai moyen de parution de mise à jour.
- Le taux d'incident en production.

Spécifications pour l'implémentation

Principes et objectifs

| Objectifs / Principes | Principe 1 Primauté des principes | Principe 2 Maximiser les avantages | Principe 4 Continuité des activités | Principe 19 Gestion réactive du changement |
|---|--------------------------------------|---------------------------------------|--|---|
| Augmentation du nombre d'adhésions | | | | |
| Augmentation du taux d'adhésion de producteur alimentaire | | | | |
| Délai de parution de mise à jour | | | | |
| Diminution du taux d'incident en production | | | | |
| Mise en place des bonnes pratiques | | | | |

Les objectifs fixés pour la nouvelle architecture respectent les principes choisis.

Standards pour l'implémentation

Durant l'implémentation, il est important de suivre les normes et bonnes pratiques du métier.

Les normes comme REST vont permettre le développement d'une solution stable et sécurisée côté serveur.

Tandis que les techniques de développement AJAX vont permettre de fournir une fluidité lors de l'utilisation de l'application à l'utilisateur.

Il est impératif d'instaurer un style de programmation (coding style) que chaque développeur devra suivre afin que tout le code produit au sein de l'entreprise soit normé.

Tout ce code normé sera par conséquent facile à maintenir et/ou modifier par n'importe quelle personne connaissant la norme.

Les solutions open source sont préférables aux solutions payantes.

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.

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, et également de favoriser l'interopérabilité (capacité d'outils à communiquer entre eux. Notion à ne pas confondre avec la compatibilité, qui caractérise un outil fonctionnant dans un environnement précis, en respectant toutes les règles que ce dernier impose).

Conditions requises pour l'interopérabilité

Chaque composant de l'architecture doit être indépendant et remplaçable.

Pour se faire, les communications entre les composants se feront à l'aide d'API REST.

La norme REST devra être respectée de manière générale lors du développement du projet. Ainsi, les interfaces du système seront normées et l'interopérabilité des composants sera assurée.

La qualité de la documentation des interfaces est essentielle à une bonne interopérabilité.

Toutes les API doivent posséder une documentation riche permettant une utilisation simple et efficace de ces dernières, ainsi, il sera facile de connecter les composants ensemble.

Pour une bonne interopérabilité, une architecture orientée service (SOA) sera un très gros atout. Les services utilisent des protocoles qui décrivent leur façon de communiquer entre eux, ce qui offre une grande facilité de communication entre ces derniers.

Conditions requises pour le management du service IT

Dans le respect d'une organisation agile et de l'approche Kanban, dans chaque équipe de développement des réunions seront organisées toutes les une à deux semaines (selon la durée des tâches à élaborer), pour définir les tâches à ajouter au tableau Kanban et pour vérifier la conformité des fonctionnalités développées au préalable avec la politique de l'entreprise.

La communication entre les équipes de développement se fera lors de réunions qui seront fixées à l'intervalle régulier par le plan de communication de la déclaration de travail d'architecture. Le plan de communication permettra de s'assurer que les bonnes informations arrivent aux bonnes personnes et que personne ne soit surchargé avec des informations inutiles.

Contraintes

Le projet a été approuvé pour un coût de 50 000 USD et une période de six mois est prévue pour définir l'architecture et préparer un projet de suivi afin de développer un prototype.

L'architecture doit permettre d'obtenir le meilleur rapport qualité/coût.

L'architecture peut inclure de nouveaux composants personnalisés ou des composants du commerce pour favoriser la flexibilité, la stabilité et l'extensibilité.

Hypothèses

Architecture orientée service

Les premières études sur les meilleures pratiques en matière d'architecture en font apparaître plusieurs qui représentent des risques techniques réduits. Il s'agit notamment de micro-services potentiels, de normes prenant en charge des solutions Web et mobiles, de base de données standards et d'autres approches similaires.

À l'aide des résultats de ces études, on peut imaginer une hypothèse d'implémentation utilisant une architecture orientée service, qui serait hébergée par Amazon-Web-Service (AWS).

Dans cette hypothèse, la sécurité des accès aux différents composants de l'application serait assurée par AWS, grâce au service IAM.

Le service IAM va permettre de gérer les utilisateurs, les identifiants comme les mots de passe, clés d'accès et les politiques d'autorisation qui contrôlent les services et ressources d'AWS auxquels les utilisateurs peuvent accéder.

Ainsi, toutes les ressources sont protégées par les rôles des utilisateurs.

AWS va également offrir des options pour mettre à l'échelle l'architecture grâce à des services prêt à l'emploi comme des répartiteurs de charge, ou encore des serveurs qui s'adaptent automatiquement à la charge de travail.

Les communications entre les services étant normées, il est très facile de les connecter ensemble ce qui permet d'en ajouter de nouveau ou d'en changer certains. L'architecture orientée service est donc très évolutive.

Cette agilité que donnent les services à l'architecture nous permettrait de s'adapter très vite au marché et de changer de direction assez facilement si besoin.

Nous avons pris ici AWS pour exemple, mais ce n'est pas le seul fournisseur de service, il y a d'autres solutions comme Google Cloud ou Microsoft Azure, cependant AWS reste le leader du marché.

Comparaison des fournisseurs de services

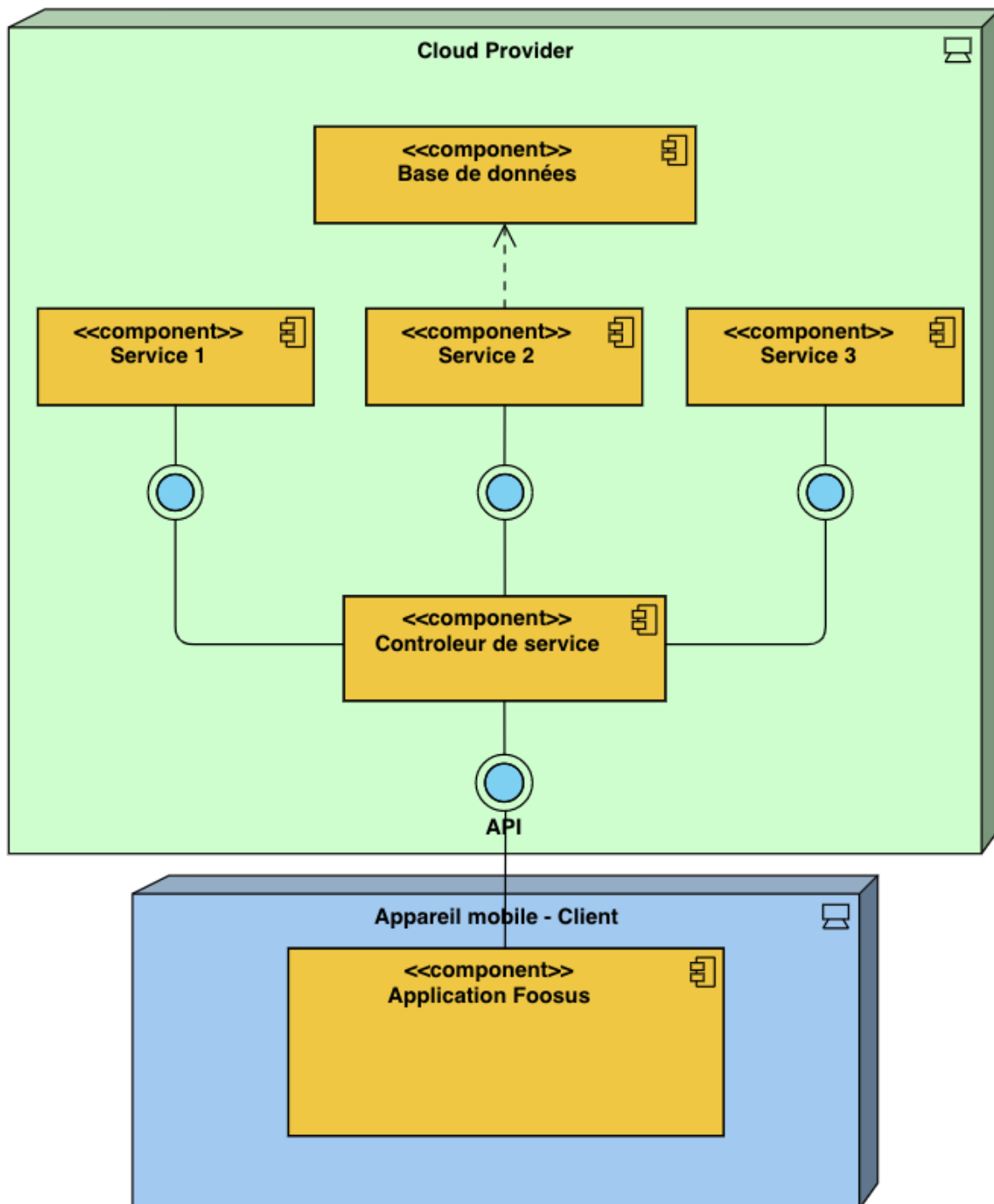
Coûts

Les trois plateformes utilisent un système de facturation à l'utilisation, c'est-à-dire que le coût dépendra de facteurs comme le nombre de requêtes ou le nombre de création d'instance. Ce système est très avantageux, car peu coûteux, mais il rend compliqué l'estimation des coûts.

Performances

Les trois plateformes proposent des performances à peu près équivalentes, une étude plus approfondie est requise pour déterminer laquelle est la plus adaptée au projet. On peut tout de même noter que AWS se démarque en tant que leader et sera plus stable et plus simple d'utilisation que ces concurrents.

Voici un exemple de ce à quoi pourrait ressembler le diagramme de déploiement avec une architecture orientée service :



Cycle de vie de l'implémentation de l'architecture

1. Analyse de l'architecture existante
2. Mise en place de normes et de bonnes pratiques
3. Conception
4. Développement
5. Intégration et test
6. Mise en production