



Spécification des Conditions requises pour l'Architecture

—
Projet : XXXX

Client : Foosus

Préparé par : Théo VACANT

—
N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l'Architecture

Date de Version du Document : 15/09/2023

Revu par :

Date de Révision :

—

Table des Matières

1. Objet de ce document
2. Mesures du succès
3. Conditions requises pour l'architecture
4. Contrats de service business

5. Contrats de service application
6. Lignes directrices pour l'implémentation
7. Spécifications pour l'implémentation
8. Standards pour l'implémentation
9. Conditions requises pour l'interopérabilité
10. Conditions requises pour le management du service IT
11. Contraintes
12. Hypothèses

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'architecte.

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

Mesures du succès

1. Le score de satisfaction client (CSAT)

Le KPI le plus populaire pour mesurer la satisfaction des clients est celui du CSAT. Il consiste à demander directement aux utilisateurs d'évaluer leur satisfaction à l'égard de notre entreprise et de notre service. Il peut être demandé tout au long du cheminement de l'utilisateur sur le site internet, par exemple pour les conseils apportés par nos équipes lors de l'achat, de l'évaluation du produit, des nouvelles fonctionnalités et pour le service client. Notre score correspondra à la moyenne de toutes les réponses de nos utilisateurs pour chacune des thématiques mesurées.

$$\text{CSAT (moyenne)} = \frac{\text{Somme de toutes les notes}}{\text{Nombre de notations}}$$

$$\text{CSAT (pourcentage)} = \frac{\text{Nombre de notations positives}}{\text{Nombre de notations}} \times 100$$

Objectif

- Nombre d'adhésion d'utilisateur par jour → supérieur à 80%
- Adhésion de producteurs alimentaires → supérieur à 80%
- Délai moyen de parution des produits → supérieur à 80%

Conditions requises pour l'architecture

1. Évolutivité de l'architecture :

L'architecture mise en place doit anticiper la croissance potentielle de l'entreprise, tant en termes d'utilisateurs que de fonctionnalités. Elle devra être conçue de manière modulaire, permettant l'ajout de nouveaux composants ou modules sans perturber les opérations existantes. Cette évolutivité garantira que l'infrastructure pourra gérer une augmentation du volume de données et des demandes des utilisateurs sans dégradation notable des performances.

2. Sécurité de l'architecture :

La sécurité est une priorité absolue, notamment en raison des données sensibles traitées par l'entreprise. L'architecture doit intégrer des mécanismes robustes de protection des données, comme le chiffrement, ainsi que des protocoles d'authentification solides. De plus, elle devra être conçue pour résister aux différentes menaces, telles que les attaques par déni de service, tout en garantissant la confidentialité, l'intégrité et la disponibilité des informations.

3. Flexibilité pour l'intégration :

L'architecture doit être suffisamment flexible pour permettre l'intégration de nouvelles fonctionnalités ou technologies à mesure que les besoins de l'entreprise évoluent. Cela signifie que les composants du système doivent être interopérables et construits sur des standards ouverts, facilitant ainsi leur intégration avec d'autres systèmes ou technologies à l'avenir.

4. Intégration avec les bases de données de produits alimentaires :

Afin de fournir des informations précises et à jour sur les sources de nourriture, l'architecture doit être capable de s'intégrer efficacement avec diverses bases de données externes. Cela nécessitera des interfaces bien définies, des protocoles d'échange de données standardisés, et une capacité à traiter rapidement et efficacement d'importants volumes de données.

5. Localisation des sources de nourriture :

L'architecture doit intégrer des outils et technologies de géolocalisation pour identifier et présenter aux utilisateurs les sources de nourriture à proximité. Cela implique non seulement la capacité de traiter des données de localisation, mais aussi d'offrir des fonctionnalités de cartographie et de navigation.

6. Intégration avec les systèmes existants :

L'architecture doit coexister et interagir harmonieusement avec les systèmes et technologies déjà en place au sein de l'entreprise. Cela nécessite des protocoles d'intégration clairs, des interfaces API bien définies, et une compréhension profonde des workflows et processus existants.

7. Gestion d'un grand volume de transactions :

Avec l'expansion de l'entreprise, l'architecture doit être capable de gérer une augmentation significative du nombre de transactions. Des mécanismes tels que le load balancing, la réplication de données, et la mise en cache peuvent être utilisés pour garantir une réponse rapide même sous un volume de requêtes élevé.

8. Prise en charge des intégrations multi-plateformes :

À mesure que l'écosystème technologique évolue, l'architecture doit être en mesure de s'intégrer avec diverses plateformes, qu'il s'agisse de systèmes d'exploitation, de dispositifs IoT, ou de plateformes mobiles. Cela garantira que l'entreprise reste agile et capable de s'adapter aux nouvelles tendances technologiques.

9. Adaptabilité aux besoins changeants :

L'environnement des affaires est en constante évolution, et l'architecture doit être conçue pour s'adapter rapidement à ces changements. Qu'il s'agisse d'introduire de nouvelles fonctionnalités pour répondre à la demande des utilisateurs, ou de modifier des workflows en réponse à des changements réglementaires, l'architecture doit offrir la flexibilité nécessaire pour répondre à ces besoins.

Contrats de service business

1. **Fourniture de services :** Le département IT s'engage à fournir et maintenir les services, applications et infrastructures essentiels. Par exemple, si le département commercial a besoin d'une nouvelle application pour suivre les leads, le département IT s'engage à développer ou acquérir, puis à déployer cette application dans les délais convenus.
2. **Support :** Un support technique est mis à disposition des utilisateurs pendant les heures de travail. Cela signifie que si un employé du service marketing rencontre un problème avec un outil de gestion de campagne, il peut contacter le support IT qui s'engagera à résoudre le problème dans un délai défini, par exemple dans les 2 heures.

3. **Développement** : Le département IT collaborera étroitement avec les unités opérationnelles pour comprendre leurs besoins. Si le département des ressources humaines souhaite une nouvelle fonctionnalité pour le suivi des formations, le département IT établira un calendrier pour sa mise en œuvre et assurera son déploiement.

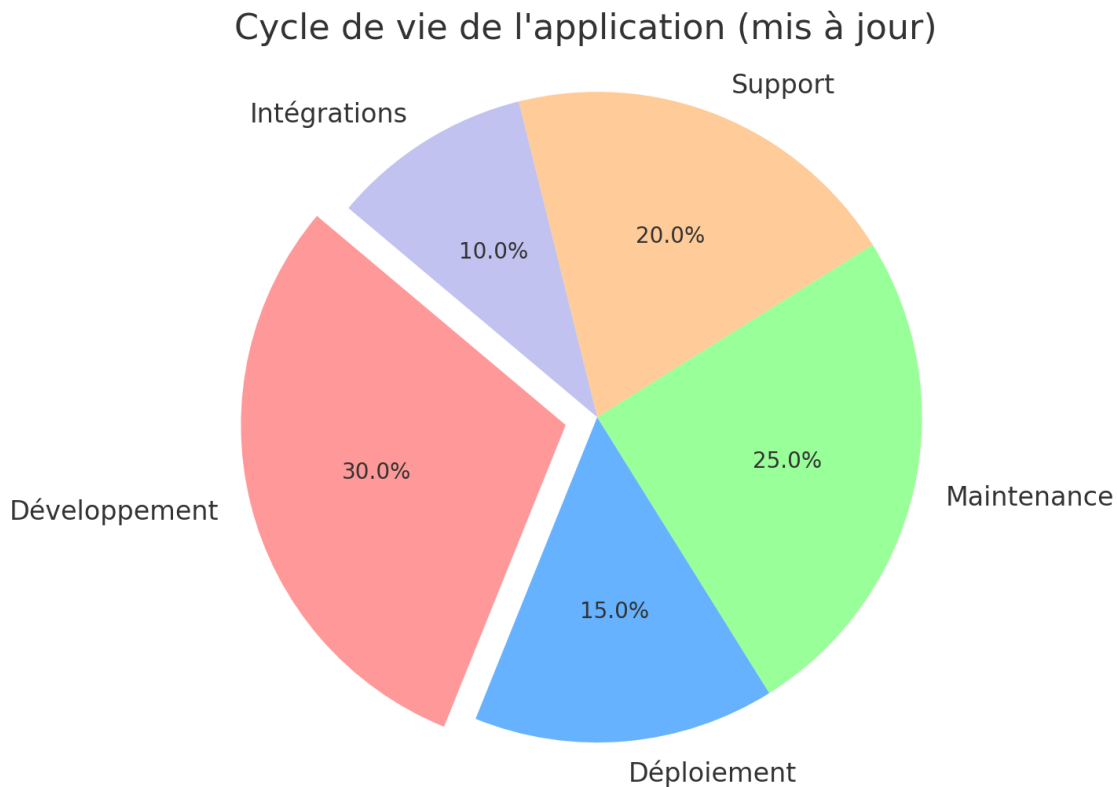
Accords de niveau de service

1. **Temps de réponse** : Garantir un temps de réponse rapide est crucial pour maintenir la productivité. Par exemple, lorsqu'un utilisateur accède à une application de gestion de la relation client (CRM), le temps de chargement de la page ne doit pas dépasser 3 secondes, garantissant ainsi une expérience utilisateur fluide.
2. **Mise à jour des informations** : Les informations, en particulier celles concernant les sources de nourriture, doivent être actualisées régulièrement. Par exemple, si un nouveau fournisseur est ajouté au réseau, ses détails doivent être mis à jour dans la base de données dans un délai de 24 heures.
3. **Disponibilité** : Assurer une disponibilité constante est essentiel pour les opérations. Par exemple, si des employés à l'étranger accèdent au système en dehors des heures de bureau locales, le système doit toujours être opérationnel, même pendant les mises à jour ou la maintenance.
4. **Gestion des incidents** : Les incidents doivent être classés par gravité. Par exemple, si un incident empêche tous les utilisateurs d'accéder à une application cruciale, il est classé comme majeur et doit être résolu en priorité. Les procédures claires de triage, de communication et de résolution doivent être en place pour garantir une réponse rapide et efficace.

Contrats de service application

1. **Performance** : Chaque application doit répondre aux critères de performance définis, tels que le temps de chargement, le temps de réponse, et la capacité à gérer un certain nombre d'utilisateurs simultanés. Par exemple, une application de gestion des stocks devrait être capable de traiter 10 000 transactions par heure sans ralentissement.
2. **Maintenance** : Le département IT s'engage à effectuer régulièrement des mises à jour et des corrections pour assurer le bon fonctionnement des applications. Par exemple, des mises à jour de sécurité pourraient être appliquées tous les mois pour une application financière.

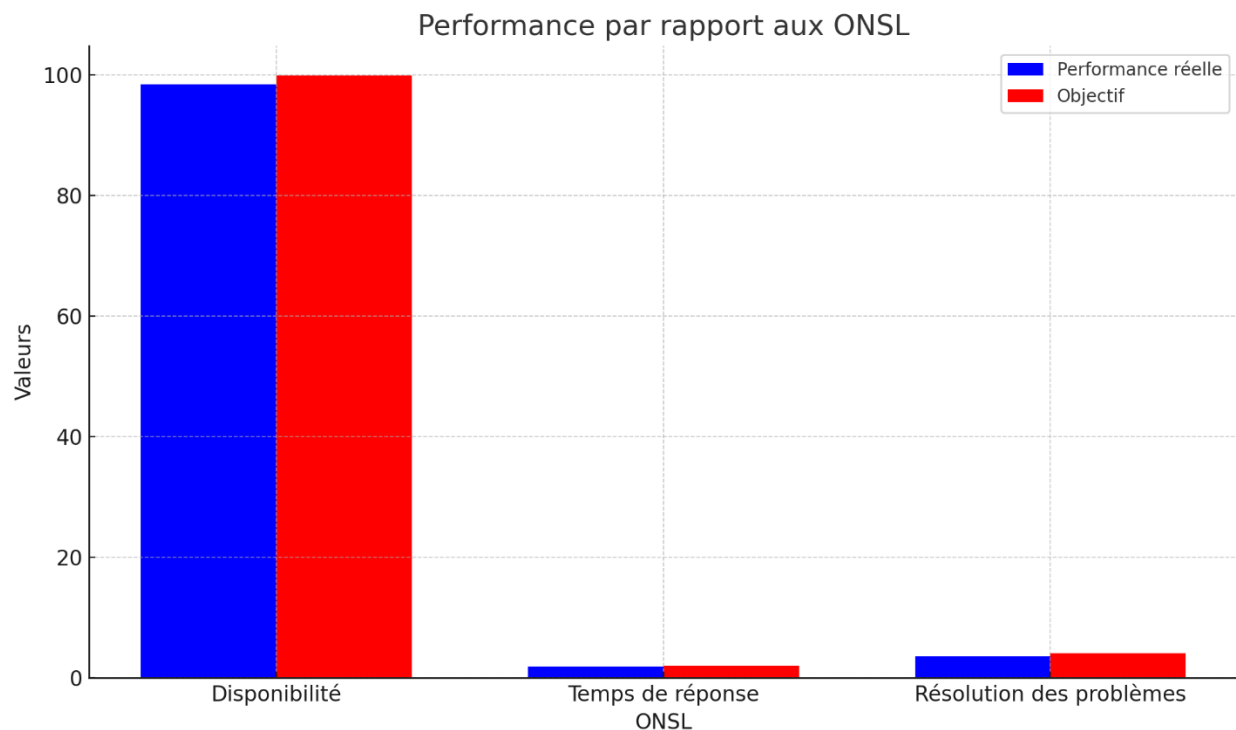
3. **Support** : En cas de problèmes avec une application, le support IT doit être disponible pour aider les utilisateurs. Cela peut inclure la résolution de bugs, l'aide à l'utilisation ou la formation sur les nouvelles fonctionnalités.
4. **Intégrations** : Si l'application doit s'intégrer à d'autres systèmes ou plateformes, le contrat détaillera les responsabilités liées à cette intégration.



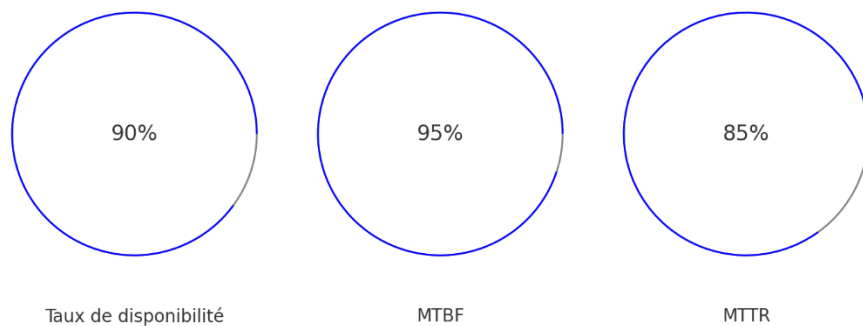
Objectifs de niveau de service

L'ONSL détaille les standards de performance et de disponibilité que le département IT s'engage à maintenir.

1. **Disponibilité** : L'application doit être disponible un certain pourcentage du temps. Par exemple, une disponibilité de 99,9% signifie que l'application peut être indisponible environ 9 heures par an.
2. **Temps de réponse** : L'application doit répondre aux requêtes des utilisateurs dans un temps défini, par exemple en moins de 2 secondes pour 95% des requêtes.
3. **Résolution des problèmes** : Les problèmes signalés doivent être résolus dans un certain délai, par exemple, les problèmes critiques doivent être résolus en moins de 4 heures.



Indicateurs de niveau de service



Lignes directrices pour l'implémentation

1. Documentation complète : Tout au long du processus d'implémentation, une documentation complète doit être maintenue. Cela comprend la documentation technique, les spécifications des utilisateurs, les manuels d'utilisation et les procédures de test. Cette documentation garantira une référence claire pour les futurs développements et aidera à la maintenance.

2. Formation : Avant le déploiement complet de tout nouveau système ou mise à jour, une formation appropriée doit être fournie aux utilisateurs finaux et au personnel technique. Cela garantira une adoption plus fluide des nouvelles solutions.

3. Tests rigoureux : Avant la mise en production, toutes les solutions doivent être soumises à des tests rigoureux, comprenant des tests unitaires, des tests d'intégration et des tests d'acceptation par les utilisateurs. Cela permettra d'identifier et de corriger les problèmes avant qu'ils n'affectent l'environnement de production.

4. Planification de la mise en production : Toute mise en production doit être soigneusement planifiée pour minimiser les perturbations. Cela comprend la planification des sauvegardes, la définition d'une fenêtre de déploiement et la communication avec les parties prenantes concernées.

5. Revue post-implémentation : Après toute mise en œuvre majeure, une revue doit être réalisée pour évaluer le succès du projet, identifier les leçons apprises et recommander des améliorations pour les futurs projets.

6. Conformité aux normes : Toutes les solutions doivent être conformes aux normes de l'industrie et aux réglementations pertinentes. Cela garantira que les solutions sont sécurisées, fiables et conformes.

7. Collaboration interdépartementale : Durant l'implémentation, une collaboration étroite entre les départements IT, opérationnels et autres parties prenantes est essentielle. Cela garantira que les solutions répondent aux besoins de l'entreprise tout en étant techniquement réalisables.

Spécifications pour l'implémentation

- 1. Architecture système :** Un schéma montrant comment les différentes composantes du système interagissent entre elles. Cela pourrait inclure des serveurs, des bases de données, des interfaces utilisateurs, et d'autres systèmes avec lesquels il s'intègre.
- 2. Flux de données :** Un schéma illustrant comment les données circulent à travers le système, depuis leur saisie initiale jusqu'à leur stockage, traitement et éventuellement leur sortie.
- 3. Processus d'intégration :** Un diagramme montrant comment le nouveau système s'intégrera avec les systèmes existants.
- 4. Vue d'ensemble de la sécurité :** Un schéma montrant les différentes couches de sécurité en place, comme les pare-feux, les protocoles d'authentification, et le chiffrement.

Standards pour l'implémentation

1. Normes de codage : Les développeurs doivent suivre des conventions et des normes de codage spécifiques pour garantir la lisibilité, la maintenance et la sécurité du code. Cela peut inclure des conventions de dénomination, des commentaires, des structures de code et des principes de programmation.

2. Contrôle de version : Tout le code et la documentation doivent être stockés dans un système de contrôle de version, tel que Git. Cela permet de suivre les modifications, de collaborer efficacement et de revenir à des versions précédentes si nécessaire.

3. Tests automatiques : L'adoption de pratiques de développement piloté par les tests (TDD) et l'utilisation d'outils d'intégration continue sont essentielles pour garantir que le code est fonctionnel et exempt d'erreurs.

4. Normes de sécurité : Il est essentiel de suivre les meilleures pratiques de sécurité, telles que le principe du moindre privilège, la validation et la désinfection des entrées, et le respect des normes de l'industrie comme OWASP pour les applications web.

5. Documentation technique : Toute l'implémentation doit être accompagnée d'une documentation technique complète, décrivant les architectures, les flux de données, les dépendances, les configurations et tout autre détail pertinent.

6. Interopérabilité : Les solutions doivent être conçues pour être interopérables avec d'autres systèmes, en respectant les normes d'interface et de communication courantes, telles que REST pour les API ou SQL pour les bases de données.

7. Accessibilité : Les interfaces utilisateurs doivent respecter les normes d'accessibilité, comme WCAG, pour garantir que les solutions sont utilisables par tous, y compris les personnes handicapées.

8. Efficacité : Les solutions doivent être optimisées pour garantir des temps de réponse rapides, une utilisation minimale des ressources et une évolutivité pour faire face à des volumes de données ou d'utilisateurs croissants.

Conditions requises pour l'interopérabilité

- 1. Protocoles standards :** Les systèmes doivent adhérer aux protocoles et normes de communication couramment acceptés, tels que HTTP/HTTPS pour les services web, SQL pour les bases de données, ou SMTP pour l'e-mail.
- 2. Formats de données communs :** Pour assurer un échange de données fluide, les systèmes doivent utiliser des formats de données communs et largement reconnus, tels que XML, JSON ou CSV.
- 3. API bien documentées :** Si l'interaction se fait via des API, elles doivent être bien documentées, avec des spécifications claires sur les méthodes, les paramètres, les réponses et les codes d'erreur.
- 4. Authentification et autorisation :** Les mécanismes d'authentification (comme OAuth) et d'autorisation doivent être en place pour garantir que seules les entités autorisées puissent accéder aux données ou aux fonctionnalités.
- 5. Adaptabilité :** Les systèmes doivent être conçus pour s'adapter facilement à de nouveaux formats, protocoles ou normes qui pourraient émerger à l'avenir.
- 6. Gestion des erreurs :** Des mécanismes robustes pour la gestion des erreurs doivent être en place pour traiter les problèmes d'interopérabilité, tels que les données mal formées, les temps d'attente ou les défaillances de communication.
- 7. Performance :** L'interopérabilité ne doit pas compromettre les performances. Les systèmes doivent être optimisés pour traiter efficacement les demandes et les réponses entre différentes plateformes.
- 8. Respect de la confidentialité :** Lors de l'échange de données entre systèmes, il est essentiel de garantir que toutes les normes de confidentialité et de sécurité sont respectées, en particulier si des données sensibles sont transférées.
- 9. Tests réguliers :** Des tests d'interopérabilité doivent être effectués régulièrement pour identifier et résoudre tout problème potentiel.

Conditions requises pour le management du service IT

- 1. Framework ITSM :** L'organisation doit adopter un cadre ITSM reconnu, comme ITIL (Information Technology Infrastructure Library) ou COBIT (Control Objectives for Information and Related Technologies), pour standardiser et guider ses processus de service IT.
- 2. Outils de gestion des services :** Des outils comme les systèmes de gestion des services d'assistance (par exemple, ServiceNow ou JIRA Service Desk) doivent être en place pour gérer les demandes, les incidents, les changements et d'autres aspects du service IT.
- 3. SLAs et OLAs :** Des accords de niveau de service (SLA) et des accords de niveau opérationnel (OLA) doivent être définis et régulièrement revus pour garantir que les services IT répondent aux attentes des utilisateurs et des parties prenantes.
- 4. Gestion des incidents :** Un processus clair pour la détection, l'enregistrement, la classification, l'escalade et la résolution des incidents IT doit être en place.
- 5. Gestion des changements :** Toute modification apportée à l'environnement IT doit suivre un processus de gestion des changements pour minimiser les perturbations et les risques.
- 6. Continuité des services :** Des plans de continuité d'activité et de reprise après sinistre doivent être établis et testés régulièrement pour garantir la disponibilité des services IT en cas d'incidents majeurs.
- 7. Gestion des capacités et des performances :** Les services IT doivent être surveillés et évalués régulièrement pour s'assurer qu'ils fonctionnent à un niveau optimal et peuvent répondre aux besoins futurs.
- 8. Formation et sensibilisation :** Le personnel IT et les utilisateurs finaux doivent recevoir une formation régulière sur les outils, les processus et les meilleures pratiques.
- 9. Feedback et amélioration continue :** Un processus doit être en place pour recueillir des commentaires sur les services IT et pour mettre en œuvre des améliorations continues basées sur ces commentaires.
- 10. Conformité et audits :** Les services IT doivent être régulièrement audités pour garantir leur conformité aux réglementations, normes et meilleures pratiques.

Contraintes

- 1. Budget :** La mise en œuvre et la maintenance des solutions peuvent être limitées par le budget disponible. Cela peut affecter le choix des technologies, la portée du projet et les ressources disponibles.
- 2. Délais :** Des échéances strictes peuvent nécessiter des compromis en termes de fonctionnalités ou de qualité.
- 3. Ressources :** Le manque de personnel qualifié ou de ressources techniques peut limiter les options disponibles pour la mise en œuvre.
- 4. Technologie existante :** Les systèmes et technologies existants peuvent imposer des restrictions sur les solutions qui peuvent être intégrées ou déployées.
- 5. Réglementations :** Des réglementations ou des normes spécifiques à l'industrie peuvent imposer des restrictions sur la manière dont les données sont traitées, stockées ou transmises.
- 6. Interopérabilité :** Les systèmes doivent souvent fonctionner avec d'autres systèmes existants, ce qui peut introduire des contraintes sur les technologies ou les méthodes utilisées.
- 7. Géographie :** La distribution géographique des utilisateurs ou des centres de données peut affecter la performance, la disponibilité ou la conformité des solutions.
- 8. Besoins spécifiques des utilisateurs :** Des besoins ou des exigences spécifiques de certains groupes d'utilisateurs peuvent limiter les options de conception ou d'implémentation.
- 9. Sécurité :** Les exigences de sécurité peuvent imposer des restrictions sur les méthodes de développement, les technologies utilisées ou la manière dont les données sont accessibles.
- 10. Durabilité :** Des considérations environnementales ou de durabilité peuvent limiter les choix technologiques ou opérationnels.