|  |  |  |  |
| --- | --- | --- | --- |
|  | **Date** | **Version** | **Commentaire** |
| JQ | 06/01/2025 | 1.0 | Création du document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table des matières

[Table des matières 1](#_Toc190093673)

[1. Introduction 2](#_Toc190093674)

[1.1. Objet du document 2](#_Toc190093675)

# Contexte

Dans un environnement où les échanges de données entre différents systèmes (Espace Client, Évaluation, NADS, Philia DOM, etc.) sont réalisés via un bus de messages asynchrone (RabbitMQ), il est crucial d’assurer la cohérence et la synchronisation des informations relatives aux membres d’entourage. Dans ce contexte, **VITALIS** agit comme le « cœur applicatif » en centralisant ces données dans sa base officielle et en orchestrant leur propagation vers l’ensemble des modules concernés. Cette approche permet notamment de garantir que chaque système dispose toujours d’une version à jour et validée des données d’entourage pour un usager

# Objectif

Concevoir et implémenter une architecture qui permette de :

 • Centraliser la gestion et la mise à jour des membres d’entourage via VITALIS.

 • Assurer l’échange asynchrone des informations via RabbitMQ.

 • Garantir la synchronisation et la cohérence des données sur l’ensemble des systèmes (Espace Client, Évaluation, NADS, Philia DOM, etc.) lors de la création ou de la mise à jour des informations d’entourage

# Consignes

## Définition des besoins et architecture

* Rédigez un schéma ou un diagramme illustrant l’architecture globale de la solution.
* Identifiez clairement les responsabilités de chaque composant :
  + VITALIS : Gestion, validation et intégration des données d’entourage dans la base.
  + RabbitMQ : Hub asynchrone permettant l’échange de messages entre les systèmes.
  + Interfaces applicatives : Consommation ou envoi de données via VITALIS et RabbitMQ

## Gestion des données d’entourage

* Décrivez comment VITALIS crée ou met à jour les membres d’entourage d'un usager dans sa base de données
* Expliquez la logique de validation des messages reçus et la manière dont VITALIS intègre ou met à jour les données.

## Mise en place des échanges via RabbitMQ

* Définissez le format des messages échangés (par exemple, type d’action, identifiant de l'usager, informations d’entourage, etc.).
* Implémentez le traitement asynchrone des messages :
  + Réception et vérification par VITALIS.
  + Mise à jour de la base de données.
  + Envoi éventuel d’un second message pour propager les informations vers des systèmes spécifiques (ex. Philia DOM).

## Mise en œuvre du scénario de création d’un membre d’entourage en deux séquences

### Séquence 1 : Traitement et validation des messages d’insertion par VITALIS

#### Initiation :

Un utilisateur, déclenche la création d’un nouveau membre d’entourage pour un usager spécifique.

#### Envoi du message d’insertion :

L’application concernée envoie un message via RabbitMQ indiquant qu’un nouveau membre d’entourage doit être inséré. Ce message inclut le type d’action « insertion », l’identifiant unique usager et les données nécessaires.

#### Traitement par VITALIS :

VITALIS récupère le message dans RabbitMQ, effectue une vérification des données (typage, présence des champs obligatoires, cohérence des informations, etc.), puis intègre ou met à jour les données dans sa base.

#### Confirmation de l’insertion :

Une fois les données validées et intégrées, VITALIS peut enregistrer un statut de confirmation (via un log ou un message de retour) pour indiquer la réussite de l’opération.

### Séquence 2 : Propagation vers un service tiers (par exemple, Philia DOM)

#### Déclenchement de la propagation :

Suite à la validation et l’insertion réussie par VITALIS, une logique métier détermine qu’il est nécessaire de synchroniser les données avec un service tiers.

#### Préparation du message de propagation :

* + VITALIS prépare un nouveau message destiné au service tiers. Ce message, envoyé via RabbitMQ, peut contenir les mêmes données d’entourage ou un sous-ensemble pertinent pour le service concerné, avec un type d’action « propagation ».

#### Envoi du message de propagation :

Le message est publié dans RabbitMQ pour être consommé par le service tiers.

#### Traitement par le service tiers :

Le service tiers (ex. Philia DOM) récupère le message, valide les informations et intègre les données dans sa propre base, assurant ainsi une synchronisation avec la référence officielle de VITALIS.

#### Confirmation de la synchronisation :

Le service tiers peut renvoyer un accusé de réception ou mettre à jour un log pour confirmer que la propagation a été effectuée avec succès.

## Documentation et réflexion

* Rédigez une courte documentation expliquant le fonctionnement global de votre solution, en mettant en lumière la centralisation des données via VITALIS et le rôle de RabbitMQ dans la synchronisation asynchrone.
* Réfléchissez aux avantages et aux défis de cette approche (notamment en termes de résilience, de gestion des erreurs et de scalabilité)

# Bonus :

Proposez des stratégies pour la gestion des erreurs (ex. messages erronés, défaillance d’un système, perte de messages) et décrivez comment elles pourraient être intégrées dans l’architecture pour améliorer la robustesse du système.