Document de définition d’architecture

Sommaire :

Introduction

État des lieux

L’entreprise LAE (Les Assureurs Engagés) est une entreprise d’assurance spécialisée dans les assurances-vie. Son engagement repose sur la satisfaction des clients, via son service client et sur la sécurité des données, notamment bancaires et à caractère privé.

Dans le cadre de l’évaluation d’un contrat, l’entreprise recueille des données sensibles tel que les maladies dont souffre le client, ses antécédent judiciaires, et son addiction à la cigarette et/ou à l’alcool.

L’entreprise est divisé en quatre services distinct :

- Le service client, qui est en communications direct avec les clients pour redirigé toute demande de leurs part vers le bon services ou pour modifier des données sur le client dans les bases de données de client/facturation/vente. Il peut également demander au service légal d’apporter une modification à la base de données légal via un appel téléphonique ou un mail.
- Le service de vente, qui a pour fonction de créer des plan d’action dont la réalisation consiste à sélectionner des clients et de leurs proposes des produits adaptés qui, s’ils sont intéressé déclenche la rédaction d’un contrat par le service légal.
- Le service légal, quant à lui, va enquêter sur les clients afin de déterminé les termes du contrat, ainsi que ses riques puis rédiger ce derniers, et cela suite à une demande du service de vente.
- Le service de facturation. Il intervient à la suite de la signature d’un contrat, afin de rédiger une facture et de l’envoyer au client.

Chacun des services présentés ci-dessus utilise une application unique, dont les technologies à leur origines sont toute différente. Cela complexifie la maintenance du système dans sa globalité. De plus les processus ne sont pas fluide, par exemple si un opérateur du service client doit apporté un changement à la base de données légal, alors il doit transmettre sa demande par téléphone ou mail au service légal étant donné qu’il est le seul à y avoir accès.

La base de données du service légal est sauvegardé toutes les semaines, mais cette sauvegarde est réalisé à la main par le responsable informatique de l’entreprise uniquement. Cela implique que lorsque ce derniers est indisponible, aucune sauvegarde n’est réalisé et ces dernières ne sont pas accessible non plus.

Le microsoft access quant à lui recueil l’historique des appels et des mails des clients consigné par les opérateurs du service client. Cependant, il est accessible sans réel vérification et consiste en un répertoire partagé synchronisé entre tous les opérateurs.

Modification de l’architecture existante

Framework d’architecture

La première différence notable avec l’architecture d’origine (figure 1 de l’annexe) est dans le fait qu’il n’y a plus qu’une seule application pour les différents services. Lors de l’utilisation de l’application, l’utilisateur sera invité à se connecter, ce qui permettra de l’identifier et de fournir les vues/accès auxquelles il a le droit.

La deuxième est que nous optons pour une architecture hybride micro-service/événementiel (figure 2 de l’annexe). Dans le but de réduire les délais de traitements ainsi que les erreurs de transmission de données entre les services, un bus d’événement sera placé entre l’interface utilisateur et les bases de données. L’interface utilisateur demandera à un service de produire un événement indiquant quelle donnée(s) doit être changer en précisant des étiquettes correspondantes aux bases de données touchées par cet événement. Une fois l’événement produit et envoyé dans le bus, des services consommateurs reliés à chaque base de données vérifierons de l’étiquette leur correspondant, auquel cas ils la consommeront puis appliqueront le changements à leur base de données. Avec un tel système, nous nous assurons d’une transmission d’information de qualité ainsi que de bases de données synchronisé.
Les changement concernant la base de données légal aura cependant besoin de la confirmation du service légal afin d’être validé.

La troisième modifications apporté concerne la sauvegarde de la base de données légal effectué à la main jusqu’à lors par le responsable informatique. Un service sera créer pour effectuer un dump de la base de données de manières périodique afin de fiabiliser ce processus. Par ailleurs ce service pourrait être étendue aux autres bases de données si le souhait en est émis.

Composants et interactions

Le point d’entrée de la nouvelle architecture sera une interface homme machine, hébergé sur un serveur, qui demandera à l’utilisateur de se connecter à son compte via une session qui lui sera distribué. Une fois la connexion effectuer l’interface chargera les vues correspondant aux accès/besoins de l’utilisateur. Si ce derniers a besoin d’effectuer une modifications à une base de données, alors les informations nécessaires seront émis à un service producteur qui va générer un événement sur mesure.

Cette création va devoir respecter plusieurs règles afin de convenir à l’architecture. Tout d’abord l’événement devra contenir des étiquettes détaillants quelle(s) base(s) de données est(sont) concernée(s) par ce derniers. Cela permettra de garder actif un événement tant que tous les consommateurs concerné n’aient été atteint, de plus l’identité de l’utilisateur émetteur de l’événement sera consigné dans ce dernier pour une meilleure traçabilité et pour s’assurer que, si la modifications porte sur des données légales, la modification soit d’abord accepté par un membre de ce service.

Un fois qu’un événement est créé, il est envoyé dans le bus où il demeurera tant que toutes ses étiquettes ne soit consommées. Les consommateurs désignés par ces étiquettes vont alors recevoir l’événement et le traduire en modification qu’ils opéreront sur leurs base de données respectives.

En parallèles des intervention manuels, deux comportement automatisé surviendront :

- Le premier est un service de sauvegarde cyclique qui, toute les semaines, effectuera une sauvegarde d'un dump de la base de données légal.
- Le second est le service de centralisation et d'archivage des logs retournés par les producteurs et consommateurs d'événements.

Mise en place des changements

Dans le but de mettre cette nouvelle architecture en place tout en réduisant au maximum l'impact sur le travail des employés pendant la transition, nous allons commencé par déployé une CI/CD (Intégration continue/Livraison continue) qui nous permettra de validé les fonctionnalité via des procédures de tests avant leurs déploiement dans le système. Ensuite la nouvelle architecture reposant grandement sur le bus d'événement, il devra être déployé en second, suivi des outils de monitoring dans le but de s'assurer du bon fonctionnement des nouvelles fonctionnalités.

Une fois la nécessaire réalisé, nous pourrons commencé à développé les fonctionnalité les unes après les autres. Leurs déploiement sera progressif afin de s'assurer de la validité de chaque fonctionnalité avant de passer à la suivante pour ne pas bloqué le fonctionnement de l'entreprise.

Pour le déploiement des nouvelles fonctionnalités, nous allons nous appuyés sur deux points afin de s'assurer que leur introduction soit réalisé avec le moins d'accro possible :

- Le premier point est que l'IHM (Interface Homme Machine) sera disponible en parallèle de l'ancienne application pour chaque service. Elle ne contiendra pas toutes les fonctionnalités dans un premier temps comme décris précédemment et s'en verra agrémenter au fur et à mesure du développement. Cela permettra de familiarisé les employés progressivement à la nouvelle application.
- Le deuxième point est que nous allons installé pour la phase de transition un service RBAC (Contrôle d'Accès Basé sur le Rôle) en amont de l'IHM permettant de choisir l'attribution des fonctionnalités avant leur déploiement final à des rôles qui sont répartis parmi les employés, ce service pourra également permettre de choisir la version désiré de la fonctionnalité testée. Ce service permettra de réalisé des pré-déploiement sur une faible portion d'employés pour un test dont les retombées seront minimisé. De plus c'est également une manières d'impliquer les employés dans le développement en récupérant leurs retour d'expérience sur les nouvelles fonctionnalités. Le bonus est que si jamais un pré-déploiement se passe mal, un retour en arrière vers la dernière version stable sera simple à réaliser.

Nouvelle solution de l'architecture

Bonnes pratiques implémentées

L'utilisation d'une architecture hybride permet de prendre les avantages des architectures micro-services et événementiel de la manières la plus adaptée au projet. De cette manières nous pouvons facilement mettre en places plusieurs pratiques importantes tel que la synchronisation des

bases de données, afin d'éviter toute divergence qui pourrait être nuisible aux clients ou aux processus de l'entreprise.

Ce type d'architecture nous permet également d'améliorer grandement la traçabilité des modifications apporté aux différentes bases de données par le biais des événements. Avec le service de monitoring du système, il sera facile d'identifier un comportement inattendu/indésirable ou une erreur commise par un utilisateur ainsi que les données affectées et à quelle moment.

Maintenabilité

La maintenabilité du système global va être grandement amélioré par plusieurs points de la nouvelle architecture.

Pour commencer le fait de fusionner les applications en une seul va permettre de diviser par quatre les efforts de maintenance. D'autant plus que nous n'utiliserons qu'une seule technologie là où il y en avait quatre à l'origine.

Ensuite grâce à la CI/CD qui sera mise en place nous allons pouvoir effectuer des procédures de tests poussés dans le but de fiabiliser les fonctionnalités avant leurs déploiement final pour réduire aux maximum l'émergence de bug lorsque celui ci sera effectuer. L'option de l'IHM permettant de choisir quelle version de chaque fonctionnalité est utilisé permettra de pouvoir faire machine arrière sur une version précédente stable en attendant que la nouvelle soit corrigé, minimisant ainsi l'impact d'un bug bloquant sur la continuité de service.

Sécurité

En ce qui concerne la sécurité, le premier point est qu'avec la gestion des rôles, un utilisateur n'aura accès qu'à ce dont il a droit. Ensuite nous pouvons séparer la sécurité en quatre aspects :

- Disponibilité : L'utilisation de l'architecture hybride micro-service/événementiel permet de bénéficié d'un suivi de l'utilisation des différents services et cloisonné les bugs s'il en survient afin de ne pas impacter l'ensemble du système à chaque fois.

- Intégrité des données : Via le service de sauvegarde cyclique, nous avons à disposition des dumps de la BDD (base de données) légale. Si cette dernières se retrouve corrompu, nous serons en mesures de rétablir la dernière sauvegarde réalisé.

- Confidentialité des données : Nous allons chiffrés les communications inter-service, ainsi que les BDD. Bien entendu toutes les BDD n'ont pas le même niveau de criticité, notamment la BDD Légal, donc si son chiffrements est prioritaire, la méthode employés pour cette dernières pourrait être inadapté à une utilisation généralisé due à un problème d'efficacité et/ou de capacité. Par ailleurs nous allons nous assurez de pseudonymisé les données afin d'éviter que, si une fuite de données survient, elles ne puissent être utilisé de quelque façons que ce soit.

- Traçabilité : Les services consommateur/producteur vont emmètrent des logs vers le service de centralisation, ce qui permettra de suivre tout le cycle de vie des événement afin de détecter au mieux un comportement inattendu ou un événement non traité. De plus un outil de monitoring offrira la possibilité de garder un œil sur la solution en entière aux administrateurs.

Le deuxième point réside dans la connexion à une session via l'IHM. Cette mesure permet d'éviter une persistance des données en local sur les machines des collaborateurs tout en ajoutant la possibilité de configurer des déconnexions pour inactivité ou autre bonne pratique.

Intégration

L'utilisation de la nouvelle solution sera disponible pour tous les services de l'entreprise au fur et à mesure de son développement. L'IHM sera disponible en parallèle des anciennes applications tant que les fonctionnalités de ces dernières ne seront pas totalement remplacées, comme précisé plus haut.

Une architecture de transition (figure 3 de l'annexe) sera nécessaire pour effectuer un déploiement fluide de la nouvelle solution. Elle nous permettra de déployer les fonctionnalités de la nouvelle solution au fur et à mesure, mais pour cela nous avons besoin de trois ajout temporaire absent de l'architecture d'origine.

Le premier est le RBAC (Contrôle d'Accès Basé sur le Rôle) qui sera le service nous permettant d'associer les différentes fonctionnalités à des rôles, puis d'attribuer ces rôles aux utilisateurs selon les besoins des tests de déploiement. Ainsi, nous pourrons mener des tests à petite échelle pour valider la qualité d'une fonctionnalité avant son déploiement final.

Le deuxième est le service de conversion. Il nous servira à convertir les informations reçu du consommateur au format de la base de données d'origine, si cette dernière est d'un type différent de la nouvelle.

Et enfin le troisième ajout est le service de synchronisation qui aura pour fonction de s'assurer que la nouvelle base de données reçoivent bien les changement qui pourrait être effectuer depuis l'application d'origine. Le but étant que les tests des fonctionnalités par un échantillon d'employé ne soit pas une activité séparé de leurs tâches habituelles, au contraire, ils s'agit simplement d'une nouvelle façon de les effectuer.

Avec cette architecture de transition, nous nous offrons la possibilité de fiabilisé le développement de la solution par des tests en condition réel sans perturber outre-mesure les employés des différent services.

Cette architecture de transition sera l'une des étape du processus de CI/CD et surviendra une fois que la fonctionnalité à tester aura validé les tests qui lui sont propre (test unitaires, de performance, etc.) pour valider le bon fonctionnement de cette dernière en condition réel.

Métriques

Dans le but de mesurer l'efficacité de la nouvelle solution, nous allons mettre en place et suivre l'évolution de différentes métriques que voici :

- Temps d'exécution des processus opérationnel : L'un des principaux maux de l'entreprise est la durée de traitement du fait que les informations clients doivent être modifié à plusieurs endroits, que des répertoire partagé doivent se synchroniser, que certaines informations sont synchronisées à la main ou encore que les demandes clients sont redirigé par téléphone ou email.
- Fréquence d'incident de désynchronisation entre des BDD : Avec l'architecture d'origine les

informations clients peuvent ne pas être mise à jour partout, nous allons donc mesurer la fréquence d'incident issue d'information contraire dans les BDD.

- Fréquence d'incident de sécurité : La solution traitant des données sensibles des clients de l'entreprise, la sécurité est un point essentiel. Aussi bien que des tests seront effectués afin d'éviter le déploiement de fonctionnalité présentant des failles, il est prudent de vérifier via cette métrique qu'aucune faille ne soit passé entre les mailles du filet.

Dette technique

L'architecture d'origine présente plusieurs points problématiques dont découle des choix pour la nouvelle architecture. Tout d'abord, chaque service possède une application qui lui est propre développé dans des technologie uniques dont certaines sont obsolètes ce qui complexifie inutilement l'effort de maintenance requis.

Pour réduire cette effort de maintenance, les applications vont être fusionné en une seule développé dans une technologie pertinente.

Annexe :

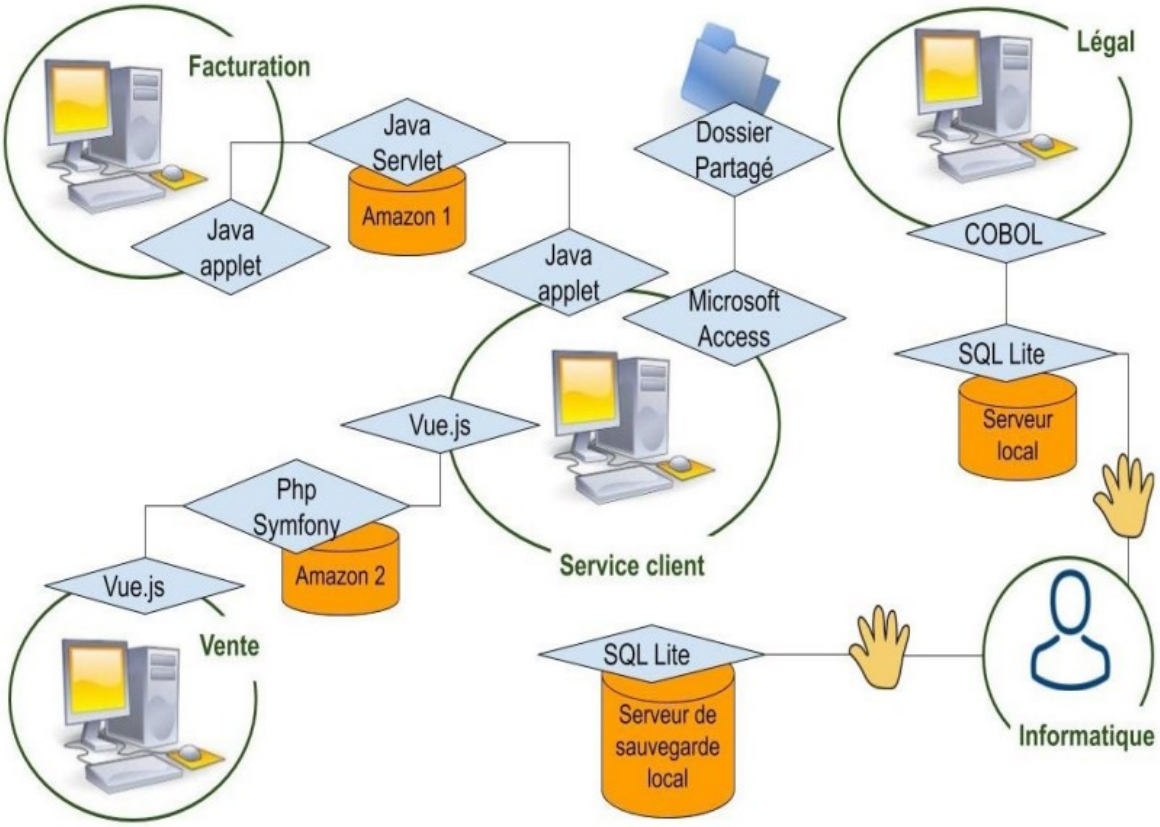

Figure 1: Architecture d'origine

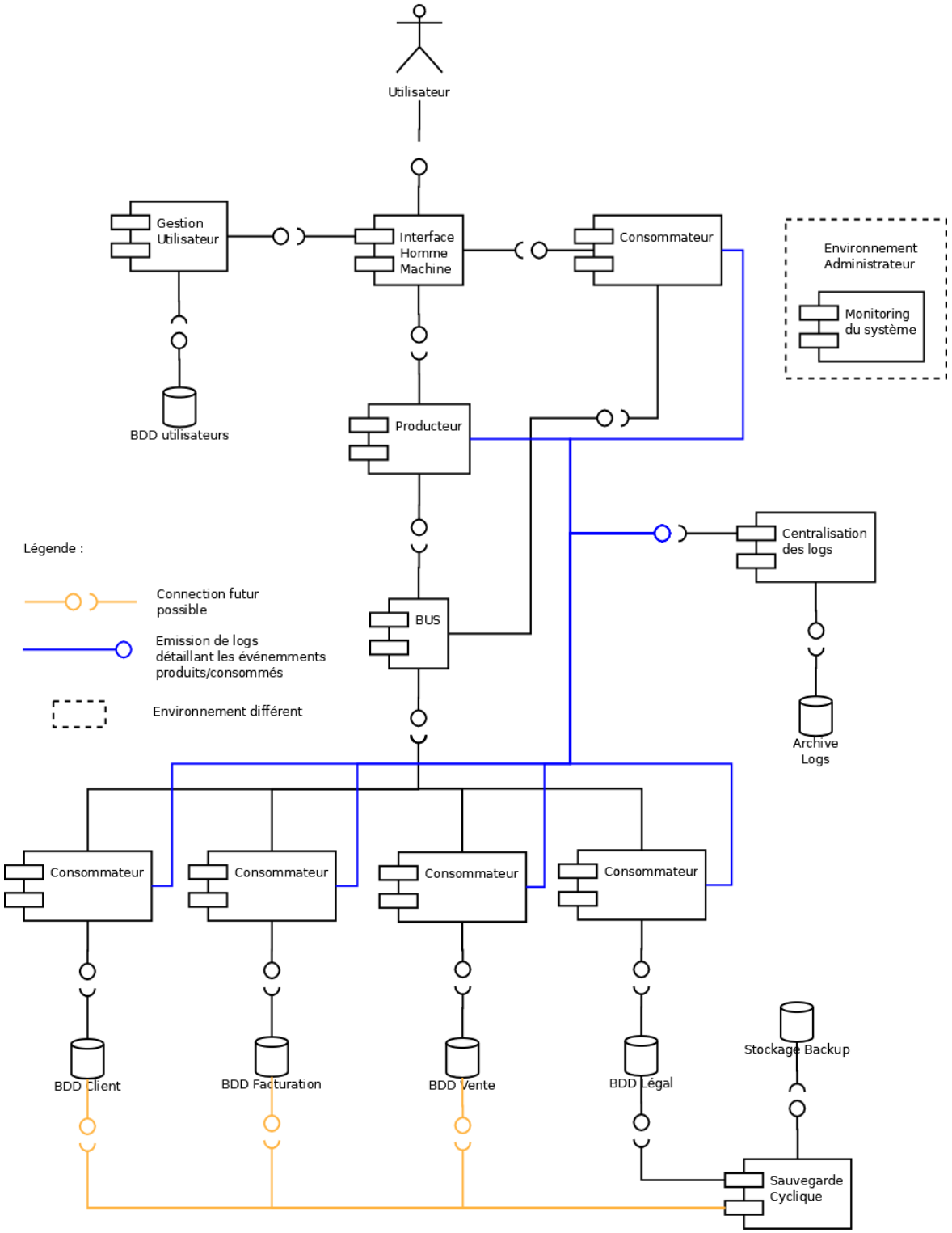

Figure 2: Architecture cible

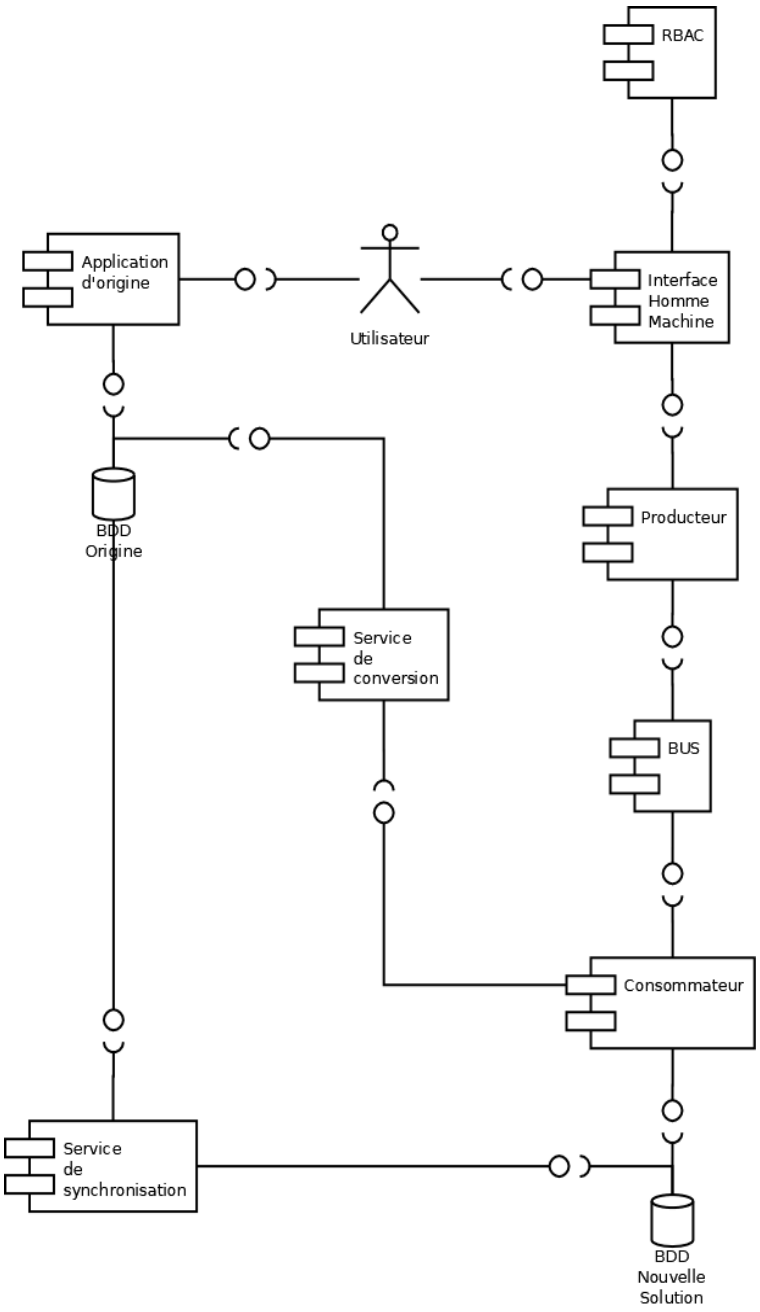

Figure 3: Architecture de transition