Document de définition d’architecture

Sommaire :

Introduction

Contexte

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’architecture actuelle du système informatique de l’entreprise est complexe, engendrant des ralentissement des processus ou bien des risques de perte de données.

L’état actuelle de l’architecture ne pouvant perdurer, un projet de refonte du système à vue le jour. Les différents objectifs soulevés sont de :

- Réduire le temps de traitement de requêtes client
- Augmenter la fiabilité du système
- Augmenter la robustesse des informations du service légal
- Augmenter la sécurité
- Réduire l’effort de maintenance

Afin de répondre à ces besoins, nous avons concus une nouvelle solutions sur mesure. Pour en avoir un détail approfondie, veuillez vous réferrer au livrable « Tailored Architecture Framework » joint à ce document. Vous retrouvez cependant en annexe de ce document l’architecture d’origine (figure 1), l’architecture cible (figure 2), ainsi que l’architecture de transition (figure 3) afin de mieux visualiser, au besoin, les sujets traités dans ce document.

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 données personnels comprises dans 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. Les données des BDD, ainsi que les dumps de sauvegardes, seront chiffrés via un algorithme AES 256. Il s'agit d'un algorithme de chiffrement symétruque, ce qui signifie qu'une seule clef sert à chiffrer/déchiffrer les données. Ce faisant cette clef devra être stocker au moyen d'un Hardware Security Module, un composant permettant de stocker la clef de chiffrement de manières sécurisé.

- 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.

Analyse des risques

Voici la liste des principaux risques identifiés pour ce projet :

ID	Risque	Facteur de mitigation
R1	Fuite de données sensibles	• Chiffrement des communications inter-service • Chiffrement des bases de données
R2	Rejet de la nouvelle solution par les employés	• Implications des équipes pendant le développement (déploiement à échelle réduite) • Campagne de communication • Sessions de formation interactives
R3	Insertion d'erreur dans les BDD via une fonctionnalité défaillante	• Centralisation des logs (traçabilité de l'erreur) • Procédure de tests en amont du déploiement • Retirer la fonctionnalité défaillante
R4	Perte de données	• Sauvegarde séparé de l'hébergement courant
R5	Désynchronisation des BDD	• Modifications des BDD via les événements

Et en voici la matrice Probabilité/Impact :

	Impact					
Probabilité		Négligeable	Mineur	Modéré	Majeur	Critique
	Très probable					
	Probable		R3			
	Peu probable				R4	R1
	Improbable			R5	R2	

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.

Premièrement, 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.

Deuxièmement, le service client enregistre des données essentiel sur un dossier partagé étant stocké par tous les collaborateurs en local et synchronisé entre eux. Cela allonge la durée de traitement des requêtes client à cause de la durée de synchronisation.

Troisièmement, le service légal dispose d'une base de données hébergée sur un serveur local, un dispositif efficace pour protéger les données sensibles des attaques extérieures. Cette configuration présente toutefois deux inconvénients majeurs :

- La sauvegarde de la base de données est manuelle et assurée par le seul responsable informatique, ce qui peut engendré des risques de pertes de données en cas d'absence. D'autant plus qu'il est le seul à avoir accès au stockage des sauvegarde.

- Une communication par email ou téléphone est nécessaire pour toute demande de modification en provenance du service client, ce qui ralenti le processus.

L'un des objectifs principal du projet étant de réduire cette dette technique accumulé, voici comment la nouvelle solution va traiter ces différents points :

- Les différentes applications vont êtres fusionnées en une seule IHM adaptative afin de réduire la complexité technologique et l'effort de maintenance.
- Le service client va utiliser une base de données unique pour tous le service dont l'édition se fera via la création d'événement, cela permettra d'augmenter l'efficacité des opérateurs puisque les modifications pourrons s'enchaîner sans que les collaborateurs n'aient à patienter pour cause de synchronisation.
- Le service légal verra sa base de données inscrite dans l'architecture hybride avec un service de sauvegarde cyclique, effectuant et sauvegardant des dumps de la BDD, et un consommateur spécial à l'écoute d'une modification en provenance du service client afin que le service légal puisse d'abord l'autoriser.

Livraison

Pour ce projet nous allons utiliser la méthodologie Agile. User de cette pratique va offrir plusieurs point intéressant. L'entreprise souffrant actuellement d'une dette technique prononcé, il est préférable de s'orienter vers un développement avec des cycles courts centrés sur une fonctionnalité. Cela en éprouvera la fiabilité, via des séries de tests à chaque itération, ainsi que la conformité, notamment grâce aux nombreux retours que nous pourrons récolter à la fin d'une itération. En appliquant cette méthodologie, nous pourrons donc minimiser au maximum la génération de dette technique supplémentaire lors du déploiement final de chaque fonctionnalité.

Pour appliquer au mieux ces directives nous allons également mettre en place une CI/CD comme déjà énoncé précédemment. Ce processus servira à automatisé la procédure de tests (test unitaires, fonctionnels, etc.) qui assurera une qualité minimal des fonctionnalité avant de les validés grâce à des tests ciblé rendu possible par la mise en place du RBAC.

Vous pouvez retrouver un diagramme de gantt découper en quatre partie (la première pour détailler les tâches à réalisé et les trois autres pour la représentation de ces dernières ainsi que leurs dépendance/hiérarchie) qui sont les figures 4, 5, 6, et 7 en annexe de ce document.
Le projet va être séparé en deux phases distinct :

- Préparation de l'environnement : L'objectif de cette phase est de mettre en place tous les outils nécessaire à la création de la solution.
- Développement de fonctionnalité : L'objectif de cette phase est de développer dans l'IHM les différentes fonctionnalités que contenait les applications d'origine de chaque services.

La première tâche qui devra être réalisé est la création de la CI/CD, puisqu'elle servira à tout le reste. En partant de là deux séries de tâches pourront être réalisé en parallèle afin de réduire le temps de développement (cela va néanmoins nécessité une ressource humaine supplémentaire, on y reviendra dans la prochaine section). La première série correspond à la mise en place du BUS d'événements suivi de la création du service « Producteur » et des « Consommateur ». La deuxième série, quant à elle, à pour but de créer les services de bases de la solution, ainsi que ceux nécessaires à l'architecture de transition, soit, dans l'ordre, l'IHM, le RBAC, les nouvelles BDD, le service de conversion (d'un événement vers l'ancienne BDD), et le service de synchronisation (de l'ancienne BDD vers la nouvelle). Ces deux série ont une durée estimé de respectivement 40 et 34 jours travaillés, d'où l'utilité de les développer en parallèle afin de réduire la durée de développement.

Une fois les deux série terminées, une troisième commence avec la deuxième phase. Il s'agit de l'apport des fonctionnalités des applications d'origine de chaque service à l'IHM en commençant par le service client, puis le service légal, le service de vente et enfin le service de facturation. Cela représente un travail de 20 jours par service pour un total de 80 pour la série.

En cumulant la première phase (42 jours puisque la création de la CI/CD est estimé à 2 jours et que la première série est de 40 jours) et la deuxième phase (80 jours), nous obtenons une estimation de la durée du projet de 122 jours. Étant donné que la durée limite du projet est de 7 mois, nous avons environs 213 jours de disponible. Donc en suivant les estimations le projet devrait respecter les délais.

Chiffrage de la solution

LAE met à disposition pour ce projet, en dehors la collaboration entière de toute l'équipe, un ingénieur généraliste en informatique qui aura pour tâche de développer la plupart des tâche comprises dans ce projet. Ce faisant, comme expliquer dans le détail des tâches de la section Livraison, en parallèle de la création de l'IHM, du RBAC, etc, il y a trois tâches portant sur la création du bus d'événement. Ces tâches sont respectivement la création du bus, du producteur, et des consommateurs. Cette série de tâches représente un total de 40 jours, si l'on considère le taux journalier moyen d'un ingénieur DevOps comme étant à ~550€ (moyenne réalisé à partir des données du site « Malt »), cela représente un investissement total de 22 000€.

D'autre part pour une raison de mitigation des risques il est préférable de ne pas conserver les dumps de sauvegarde des BDD au même endroit que l'hébergement courant de la solution. Aussi, comme il est préférable d'héberger la solution avec des serveur en interne pour garantir un contrôle dessus, nous avons réalisé une estimation d'une location de stockage longue durée pour archivage via Google. Cette estimation à été faite pour des serveurs en Belgique avec un total de 100TB (taille pouvant être modifié selon le besoin) au mois pour 117,89€, donc si nous partons pour un premier devis avec 10 ans de stockage soit 120 mois, le coût total s'élève à 14 146,8€.

Avec ces deux points, la solution consomme 36 146,8€ du budget alloué de 200 000€, il resterait donc un total de 163 853,2€ de budget. Comme précisé un peu plus tôt nous recommandons d'héberger la solutions sur site afin de garder un meilleur contrôle sur cette dernières, ce faisant les serveurs de l'entreprise devront être adapté à cette fin, et le budget restant devrait permettre de couvrir ces frais.

Annexe :

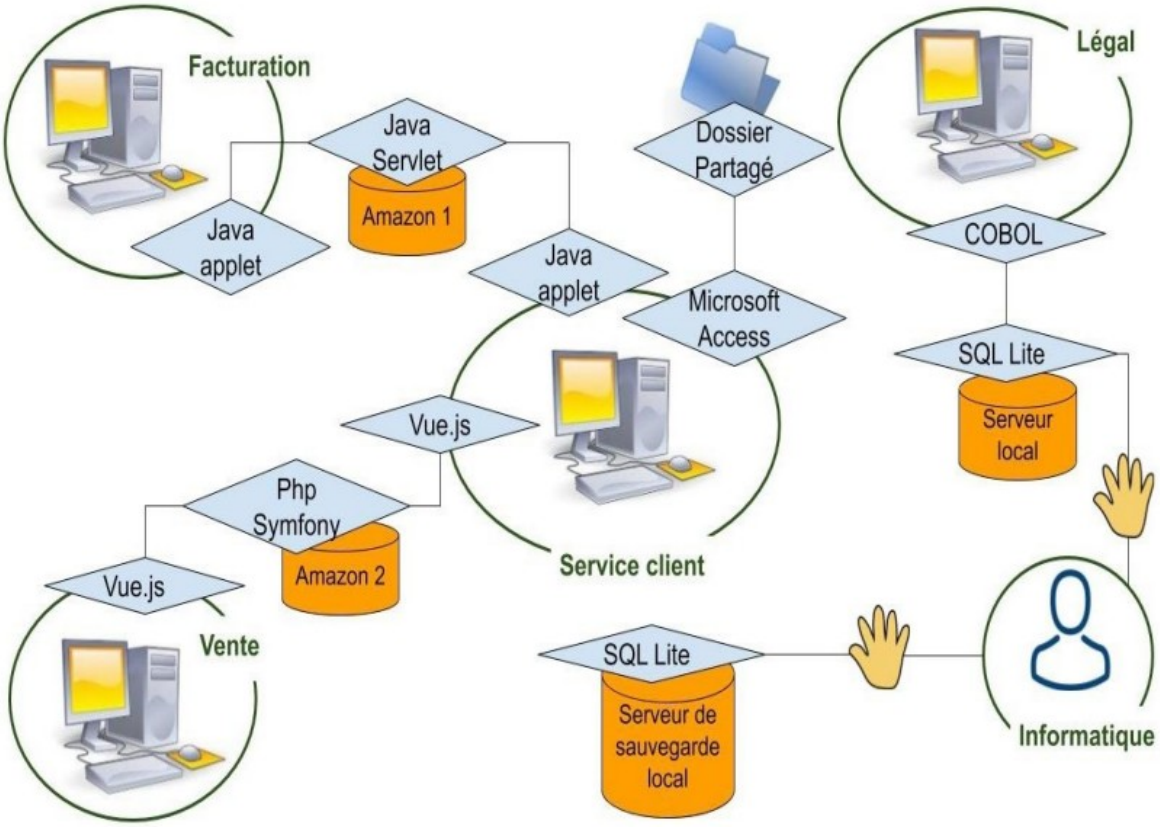

Figure 1: Architecture d'origine

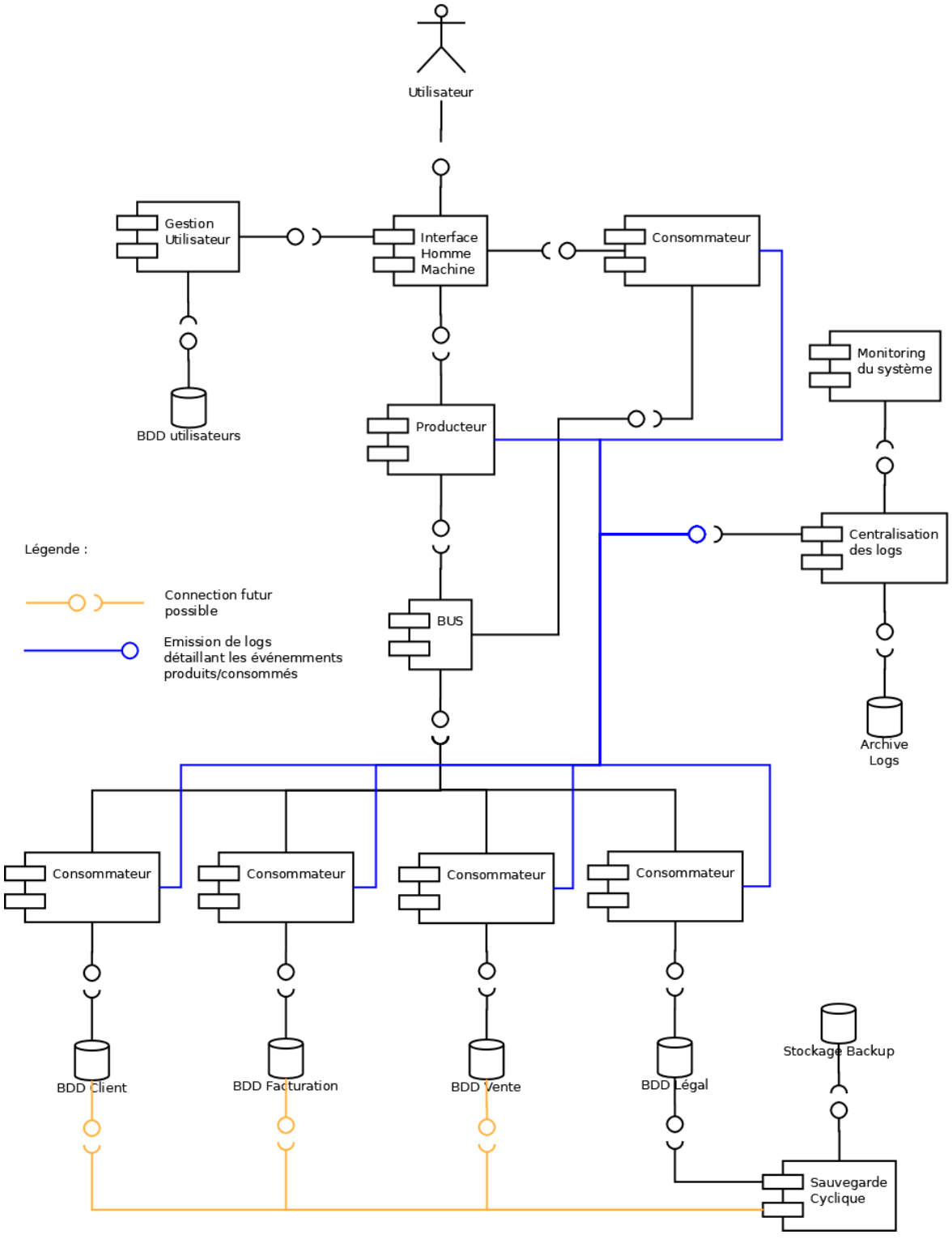

Figure 2: Architecture cible

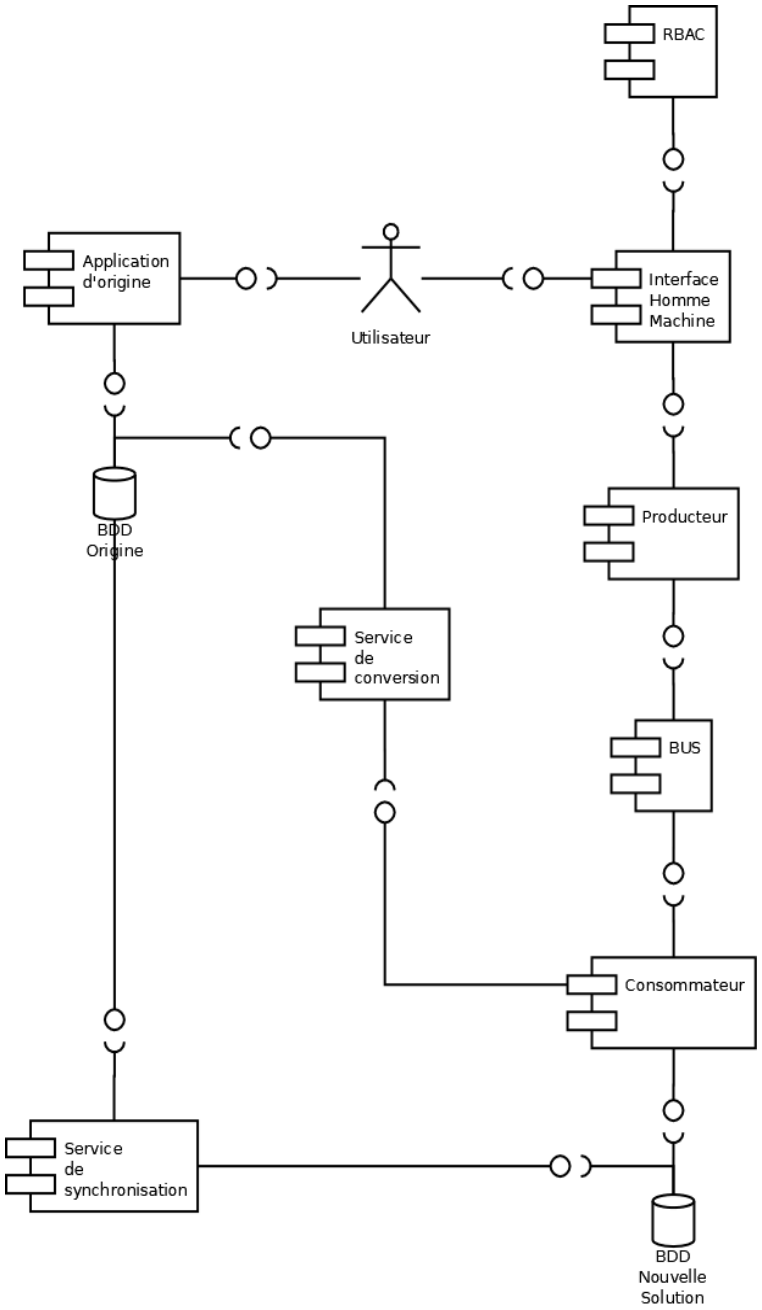

Figure 3: Architecture de transition

Préparation de l'environnement	28/01/2025	26/03/2025
Création de la CICD	28/01/2025	29/01/2025
Création du BUS	30/01/2025	12/02/2025
Création du Producteur	13/02/2025	26/02/2025
Création des consommateurs	27/02/2025	26/03/2025
Création IHM	30/01/2025	12/02/2025
Création RBAC	13/02/2025	17/02/2025
Création des BDD	18/02/2025	20/02/2025
Création Service de Conversion	21/02/2025	06/03/2025
Création du service de synchronisation	07/03/2025	18/03/2025
Développement de fonctionnalité	27/03/2025	16/07/2025
Service Client	27/03/2025	23/04/2025
Service Légal	24/04/2025	21/05/2025
Service Vente	22/05/2025	18/06/2025
Service Facturation	19/06/2025	16/07/2025

Figure 4: Détail des tâches du diagramme de Gantt, représenté dans les figure 5, 6, et 7

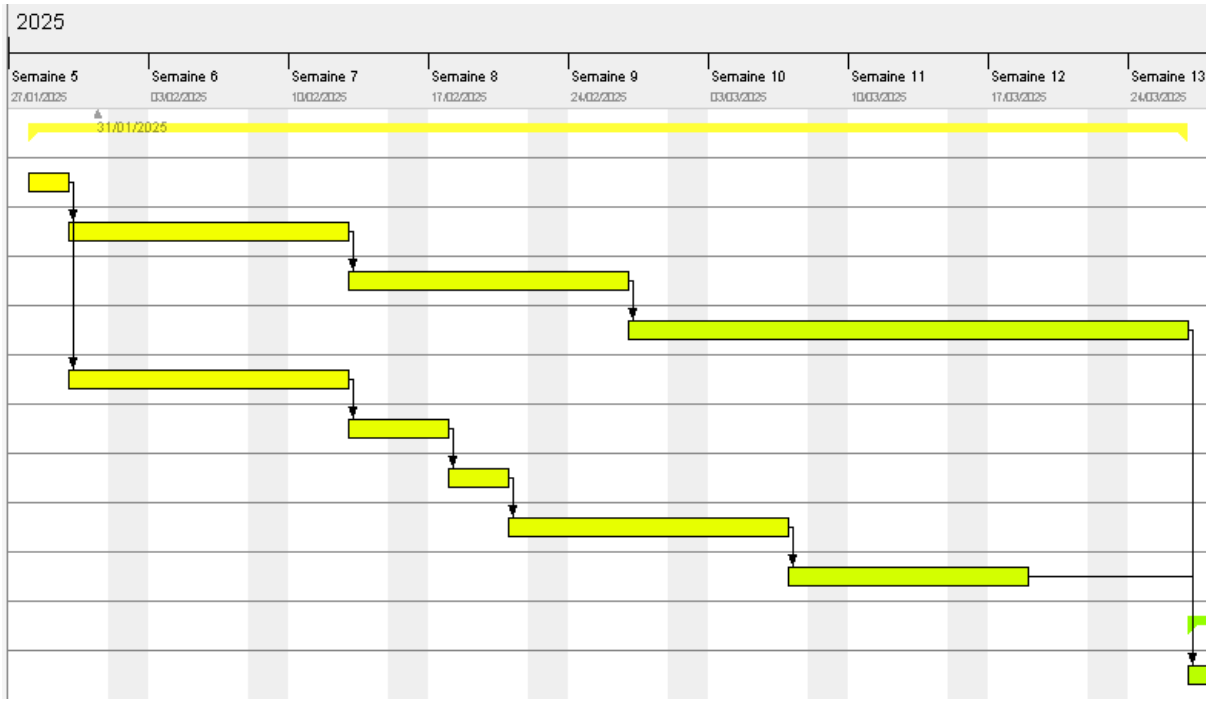

Figure 5: Première section du diagramme de Gantt représentant le phase de préparation de l'environnement

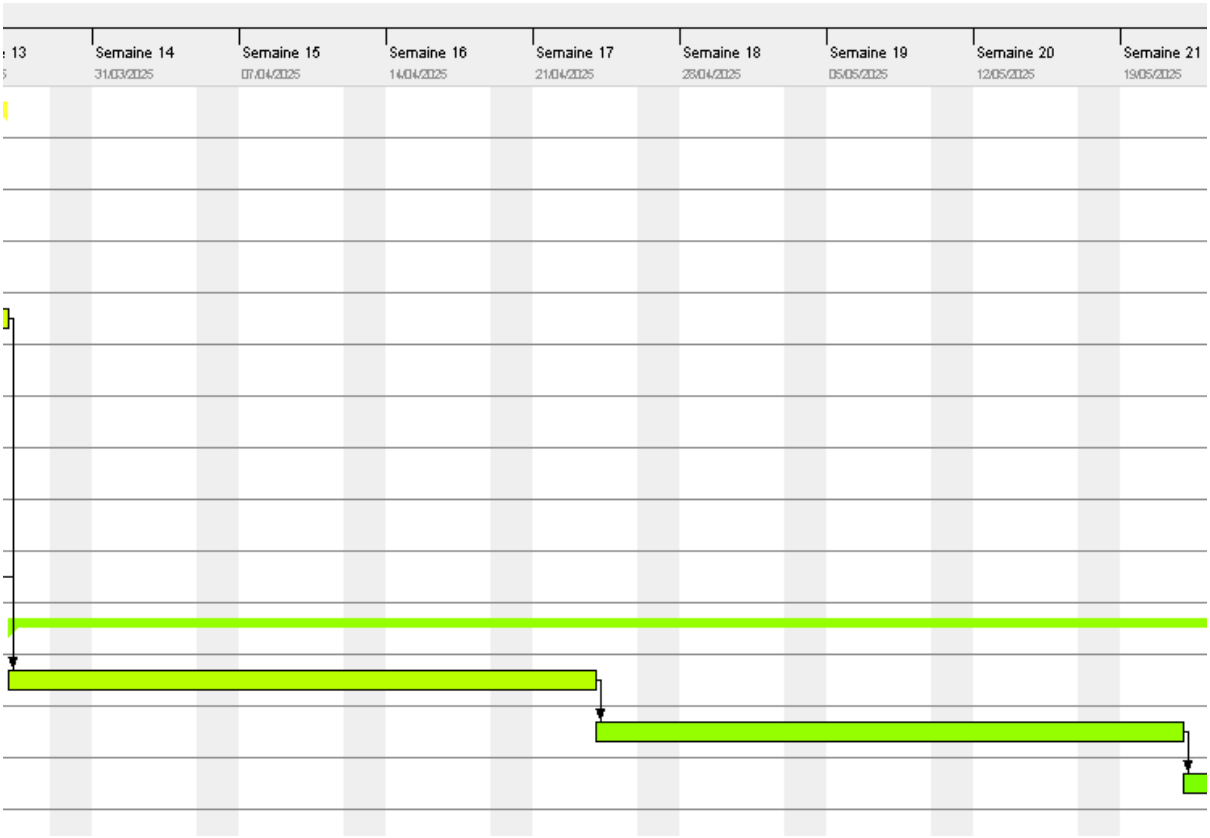

Figure 6: Deuxième section du diagramme de Gantt représentant le début de la phase de développement de fonctionnalité

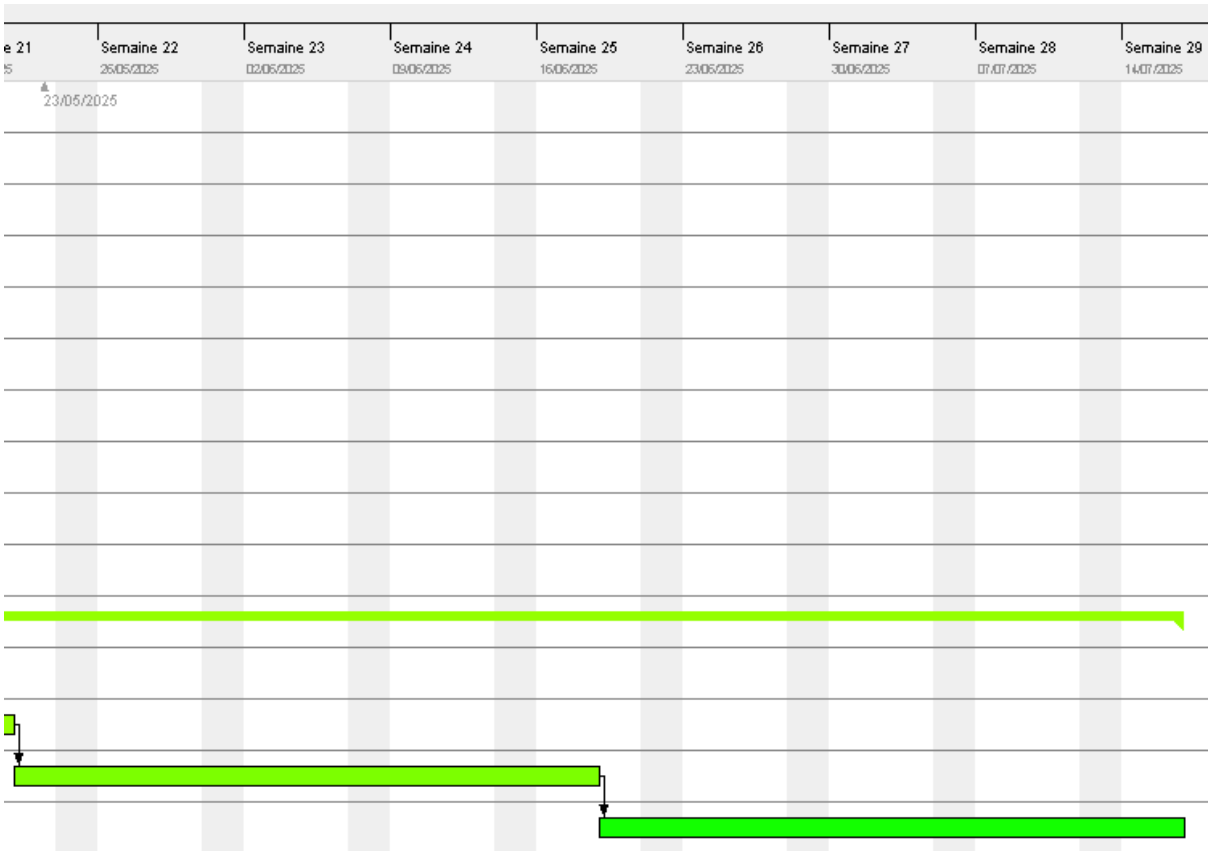

Figure 7: Troisième et dernière section du diagramme de Gantt représentant la fin de la phase de développement de fonctionnalité