Définition d'Architecture

1. Introduction et Objectifs
2. Besoins et exigences
3. Analyse de l'État Actuel
4. Architecture Cible
5. Analyse de l’Écart
6. Analyse de l’Impact
7. Transition
8. Justification

Introduction

Les Assureurs Engagés (LAE) est une entreprise dynamique dans le secteur de l'assurance-vie, reconnue pour son approche centrée sur le client et son engagement envers la qualité de service. LAE est confrontée à des défis croissants liés à son système d'information hétérogène, qui limite sa capacité à répondre efficacement aux besoins de ses clients et à se développer dans un marché compétitif.

Objectifs du Projet

Le projet vise à transformer l'architecture informatique de LAE pour renforcer son efficacité opérationnelle, améliorer la satisfaction client et assurer une croissance soutenue. Les objectifs spécifiques comprennent :

1. **Amélioration de l'Efficacité du Système d'Information** : Moderniser l'infrastructure informatique pour améliorer les performances et la fiabilité.
2. **Réduction de la Dette Technique** : Mettre à jour les technologies obsolètes et intégrer les systèmes fragmentés pour réduire les coûts de maintenance et augmenter la flexibilité.
3. **Amélioration de l'Expérience Utilisateur** : Fournir une interface utilisateur plus intuitive et des processus simplifiés pour améliorer l'engagement et la satisfaction des clients et des employés.

Ces objectifs s'alignent avec la vision stratégique de LAE pour devenir un leader dans le secteur de l'assurance-vie, en utilisant la technologie comme levier clé pour l'innovation et l'excellence opérationnelle.

Besoins et Exigences

Besoins fonctionnels

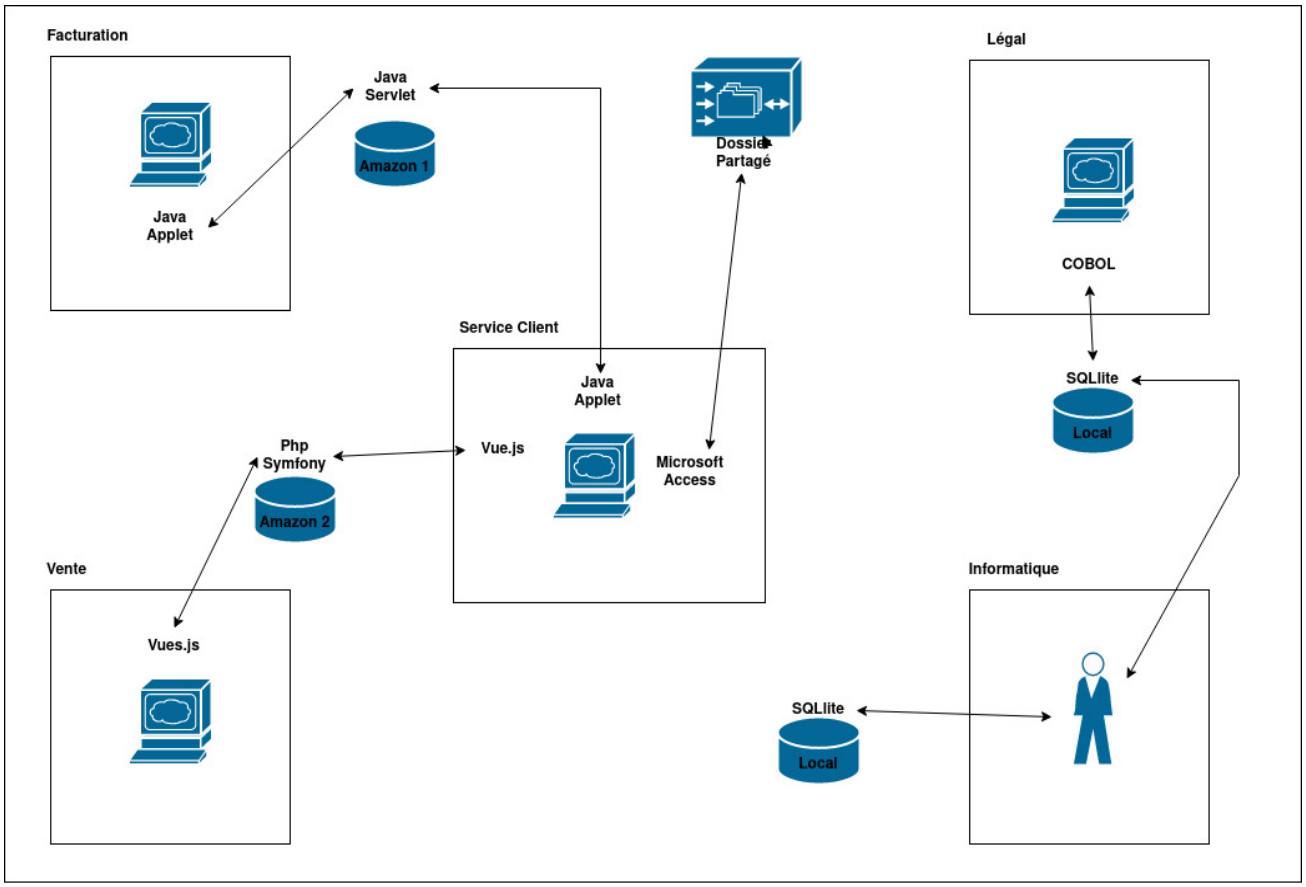
- **Intégration des systèmes** : Assurer une intégration fluide entre les différents systèmes et technologies utilisés par les services de l'entreprise pour améliorer la communication et l'efficacité opérationnelle.
- **Automatisation des processus** : Identifier et automatiser les processus manuels récurrents pour réduire les délais de traitement et minimiser les erreurs.
- **Gestion de la relation client** : Développer des outils ou intégrer des solutions existantes pour améliorer le suivi et la gestion des informations clients.
- **Rapports et analyses** : Mettre en place des solutions de business intelligence pour la génération automatique de rapports et l'analyse de données, afin d'aider à la prise de décision.

Exigences techniques

- **Sécurité des données** : Renforcer la sécurité des données pour protéger les informations sensibles des clients et se conformer aux réglementations en vigueur.
- **Performance et fiabilité** : Assurer que le système est capable de gérer les charges de travail actuelles et futures avec des temps de réponse rapides et une disponibilité élevée.
- **Scalabilité** : La solution doit être facilement scalable pour s'adapter à la croissance de l'entreprise sans nécessiter de refonte majeure.
- **Maintenance et support** : Prévoir des outils et pratiques pour faciliter la maintenance du système et offrir un support technique efficace.

Analyse de l'État Actuel

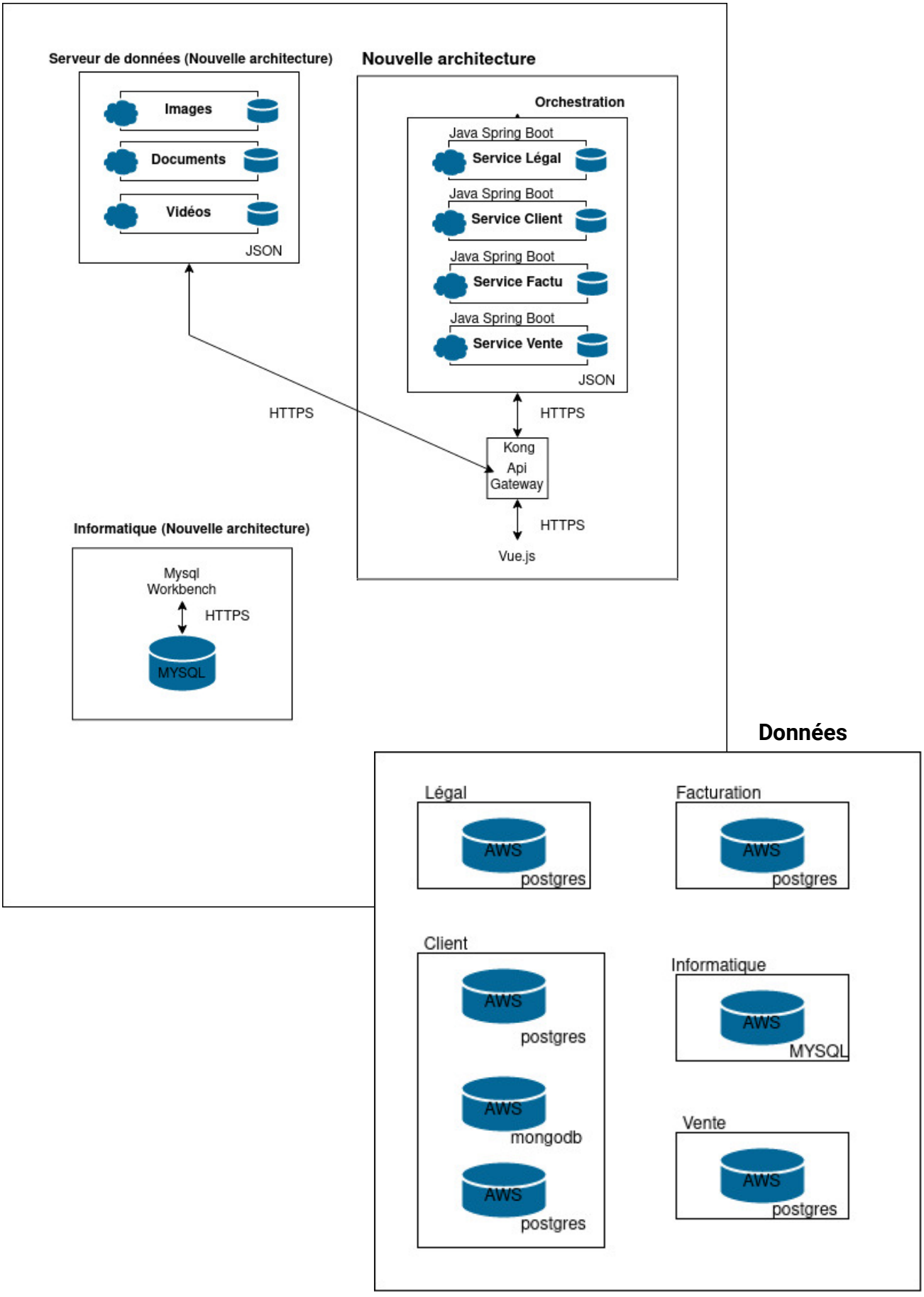
Aspect	Description Actuelle
Technologies Employées	Système composite avec COBOL pour le service légal, PHP et JavaScript pour le service vente, et bases de données isolées pour chaque service.
Sécurité des Données	Utilisation de technologies obsolètes présentant des failles de sécurité. Sauvegardes manuelles sujettes à erreurs et à des risques de perte de données.
Satisfaction Client	Temps de réponse allongé et informations parfois incohérentes dues à la non-synchronisation entre les systèmes.
Défis Opérationnels	Complexité et coûts de maintenance élevés en raison de la diversité des technologies et des processus manuels pour la synchronisation des données.
Dossier partagé	Utilisation d'un dossier partagé qui n'inclut pas de permissions strictes et qui pourrait remettre en cause l'intégrité des données.



Architecture Cible

Aspect	Technologie recommandée	Raison
Conteneurisation	Docker	Encapsulation de microservices dans des conteneurs.
Orchestration	Kubernetes	Gestion automatique des conteneurs Docker.
Communication inter-services	REST / gRPC	REST pour les API HTTP simples, gRPC pour la performance.
API Gateway	Kong / Apigee	Gestion centralisée des requêtes, sécurité, surveillance.
Sécurité	OAuth 2.0 / JWT	Authentification et autorisation.
Base de données	MongoDB / PostgreSQL	MongoDB pour le stockage de documents, PostgreSQL pour les relations complexes.
Monitoring et logistique	Prometheus / Grafana	Collecte de métriques et visualisation.
Développement et déploiement CI/CD	Jenkins / GitLab CI	Automatisation CI/CD.
Stockage et gestion des configurations	Consul / etcd	Stockage de configuration et découverte de services.
Backend	Java (Spring Boot)	Robustesse, portabilité, écosystème riche pour microservices.
Frontend	Vue.js	Simplicité, réactivité, intégration facile avec microservices.

Application et Technologies



Analyse de l’écart

Aspect	Ancienne Architecture	Nouvelle Architecture	Écart Identifié
Technologies Employées	Utilisation de COBOL, PHP, JavaScript et bases de données isolées.	Adoption de Docker, Kubernetes, REST/gRPC, MongoDB, PostgreSQL, Jenkins/GitLab CI.	Transition vers des technologies modernes et unifiées nécessitant une mise à jour des compétences et des outils.
Sécurité des Données	Failles dues à l'utilisation de technologies obsolètes. Sauvegardes manuelles.	Sécurité renforcée avec des protocoles modernes (OAuth/JWT) et des bases de données sécurisées.	Nécessité d'améliorer les pratiques de sécurité des données et d'automatiser les processus de sauvegarde.
Intégration des Systèmes	Systèmes hétérogènes difficilement intégrables.	Microservices facilitant l'intégration entre les différents services via des API.	Reconfiguration nécessaire pour l'intégration fluide des services et la communication entre microservices.
Scalabilité	Difficulté à évoluer en raison de la fragmentation et de l'obsolescence.	Architecture conçue pour être scalable grâce à l'utilisation de conteneurs et d'orchestrateurs.	Adaptation de l'infrastructure pour permettre une scalabilité horizontale et verticale.
Maintenance et Support	Complexité et coûts élevés de maintenance en raison de la diversité des technologies.	Simplification et réduction des coûts grâce à l'unification des technologies et à l'automatisation des déploiements.	Mise en place de pratiques de maintenance et de support adaptées aux nouvelles technologies.
Performance et Fiabilité	Performances inégales et fiabilité parfois compromise.	Amélioration de la performance et de la fiabilité grâce à des technologies et des pratiques modernes.	Surveillance et optimisation continues pour garantir la performance et la fiabilité des services.
Expérience Utilisateur	Interface utilisateur et interactions non	Interfaces modernes et processus utilisateurs	Conception et implémentation

Aspect	Ancienne Architecture	Nouvelle Architecture	Écart Identifié
	optimisées.	simplifiés avec Vue.js et des pratiques de développement agiles.	d'interfaces utilisateurs plus intuitives et réactives.
Gestion des Données	Fragmentation des bases de données entraînant des incohérences.	Adoption de modèles de données et technologies optimisés.	Migration et consolidation des données dans des systèmes scalables et optimisés.

Analyse de l’impact

Aspect	Impact de la Nouvelle Architecture
Fiabilité Technologique	**Modernisation et Unification :** Réduit la dette technique, améliore la flexibilité et l'efficacité des déploiements.
Sécurité	**Renforcement de la Sécurité :** Améliore la protection des données sensibles et assure la conformité réglementaire.
Satisfaction Utilisateur	**Amélioration de l'Expérience Utilisateur :** Augmente la satisfaction et la fidélité des utilisateurs grâce à des services plus rapides et fiables.
Rapidité Opérationnelle	**Optimisation des Opérations :** Réduit les coûts de maintenance et améliore la performance globale du système.
Maintenance et Performance	**Flexibilité et Scalabilité :** Permet une adaptation rapide aux besoins changeants du marché et aux nouvelles opportunités.
Sauvegarde automatique	**Résilience et Sauvegarde :** Augmente la capacité du système à faire face aux problèmes en sauvegardant régulièrement les données.

Transition

Stratégie de transition des données partagées

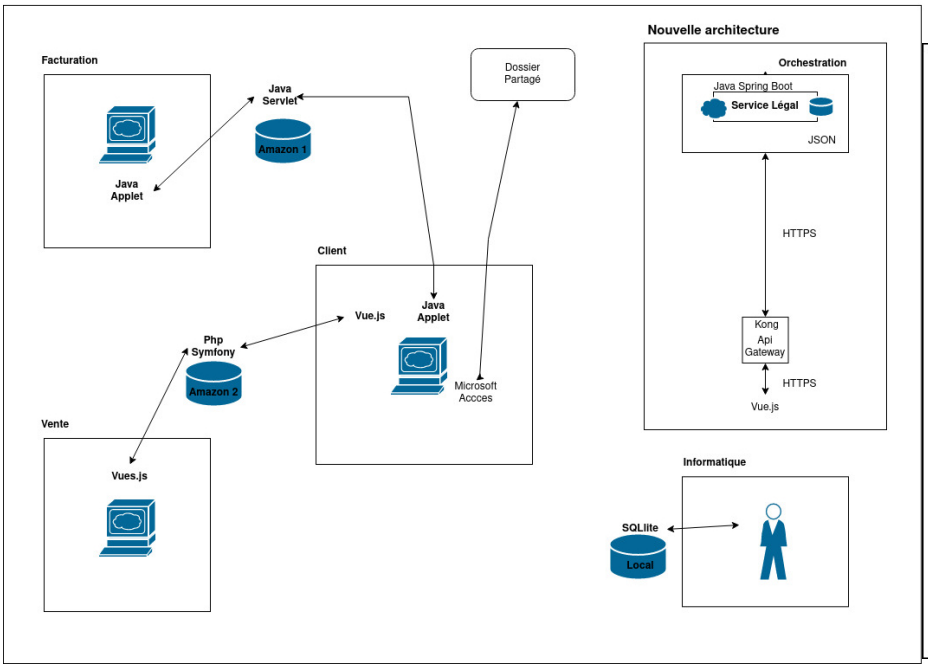
Étape	Objectif	Actions
Audit et Classification des Données	Identifier et classer les données.	- Inventaire des dossiers partagés. - Classification des données selon la sensibilité, fréquence d'accès.
Conception de Services de Gestion des Données	Développer des microservices pour la gestion des données.	- Définir des API pour l'accès et la manipulation des données. - Concevoir des services pour différents types de données (documents, images).
Mise en Place d'une Plateforme de Stockage Centralisée	Centraliser le stockage des données.	- Sélectionner une solution de stockage adaptée (base de données NoSQL, système de fichiers distribué). - Migrer les données vers cette plateforme.
Intégration et Interopérabilité	Assurer l'intégration des microservices.	- Utiliser des API REST pour l'interaction avec les applications. - Mettre en place des mécanismes d'authentification et d'autorisation.
Migration et Transition	Organiser la migration des données et la transition.	- Planifier la migration par phases. - Développer des scripts ou utiliser des outils de migration pour transférer les données.
Suivi et Optimisation	Évaluer les performances et l'adoption par les utilisateurs.	- Mettre en place des outils de monitoring et de logging. - Recueillir les retours des utilisateurs pour optimiser les services.

Plan de mise en œuvre de la nouvelle architecture

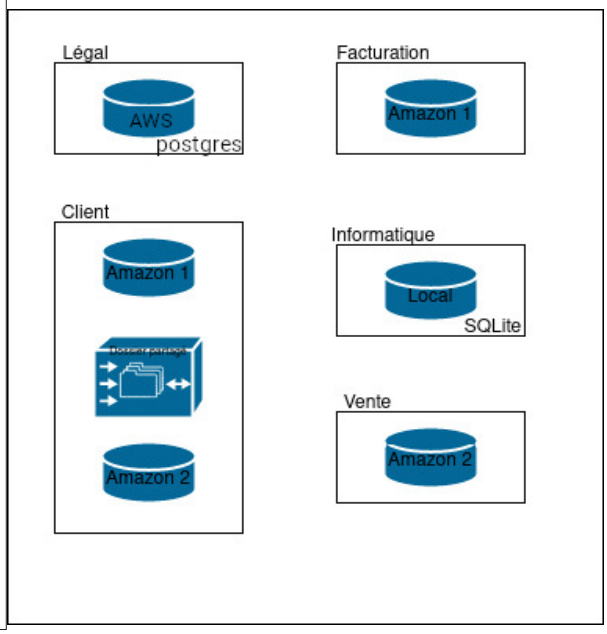
Phase	Étapes	Description
1. Préparation	Analyse du système actuel, Mise en place de l'infrastructure de base	Comprendre les flux actuels des services et préparer l'environnement de conteneurisation.
2. Phase Pilote - Service Légal	Développement des microservices légaux, Intégration et test, Déploiement et évaluation	Construire et intégrer un microservice légal non critique, tester son intégration et évaluer les résultats.Examiner les besoins du service client, créer des microservices appropriés, et les déployer suite aux tests.
3. Service Client	Analyse du service client, Développement des microservices client, Intégration, test et déploiement	Examiner les besoins du service client, créer des microservices appropriés, et les déployer suite aux tests.
4. Service Facturation	Analyse de la facturation, Développement des microservices de facturation, Intégration, test et déploiement	Évaluer les processus de facturation, développer des microservices dédiés, et les intégrer après tests.
5. Service Ve,te	Analyse pour vente, Développement des microservices de vente, Intégration, test et déploiement	Analyser les systèmes de vente, développer des microservices pour les fonctionnalités de vente, et déployer après tests.
6. Service Informatique et Transition Complète	Mise en place MysqlWorkBench, Migration des données, Déploiement final, Retrait des anciens systèmes	Développer le système dédiés au pole Informatique. Migrer les données, déployer tous les microservices et retirer les anciens systèmes.
7. Formation et Support	Formation des utilisateurs, Support continu	Former les utilisateurs à l'utilisation des nouveaux microservices et fournir un support technique.
8. Révisions et Améliorations	Évaluations régulières, Améliorations itératives	Évaluer et améliorer continuellement les microservices en fonction des retours et des données de performance.

2. Schéma – Phase Légal

Application et Technologies

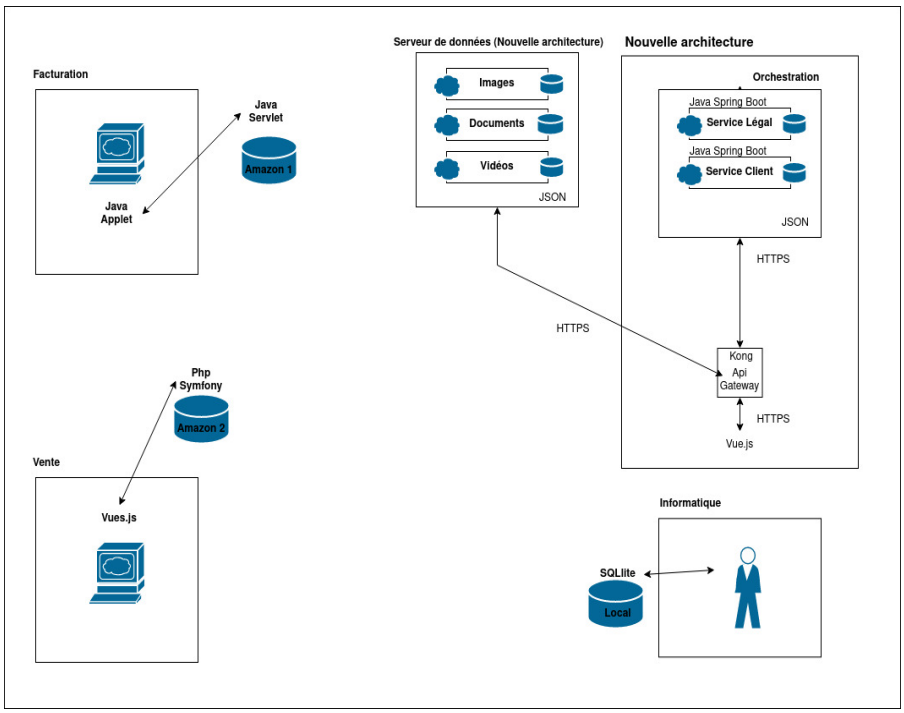


Données

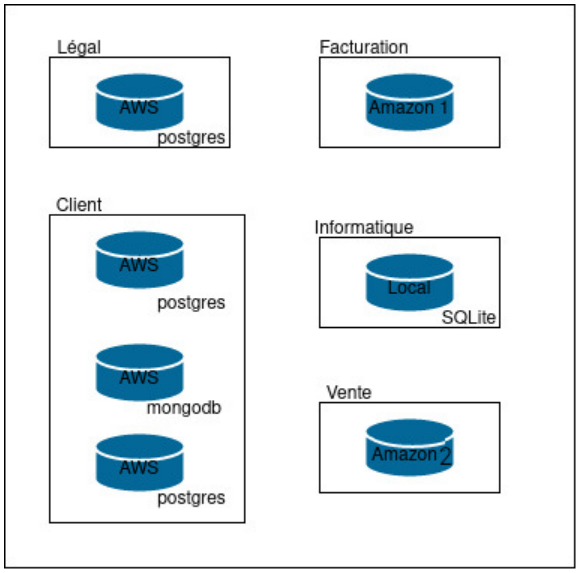


3. Schéma – Phase Service Client

Application et Technologies

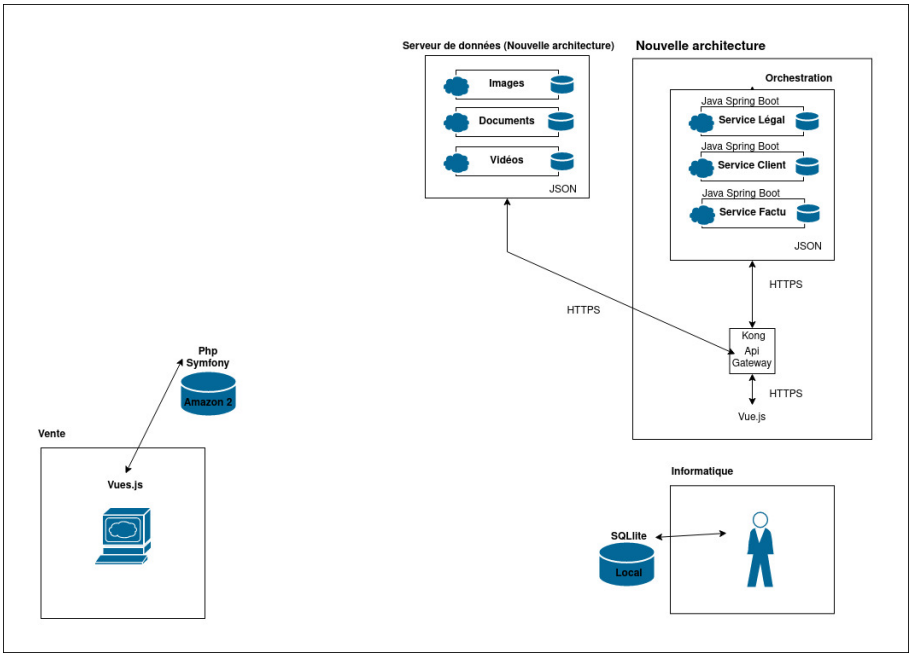


Données

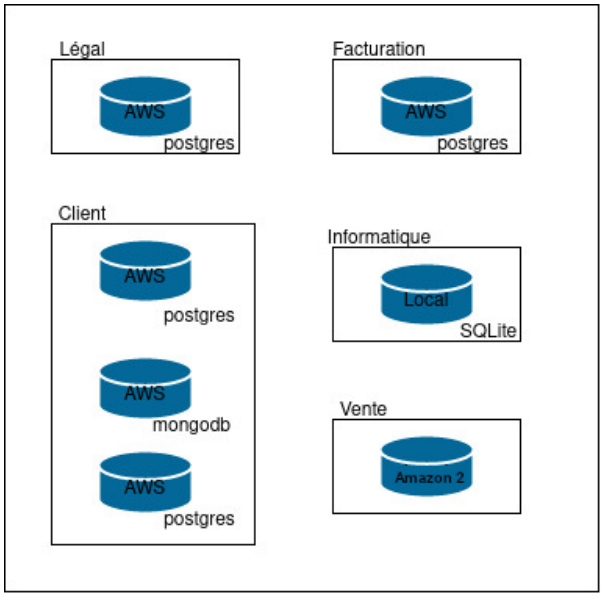


4. Schéma – Phase Facturation

Application et Technologies

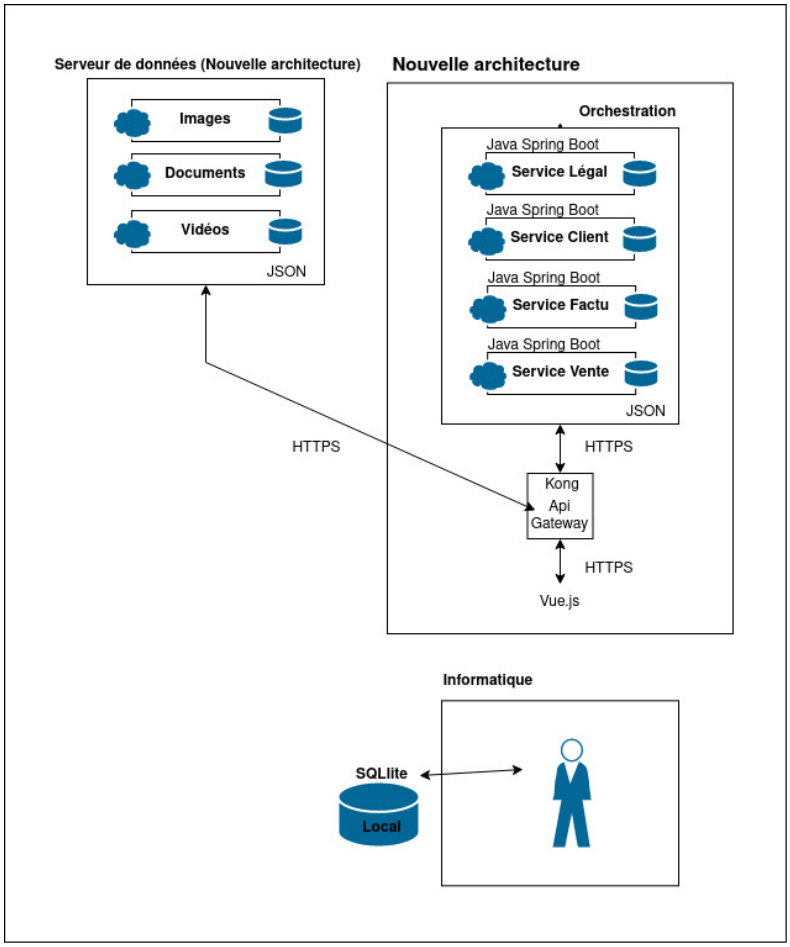


Données

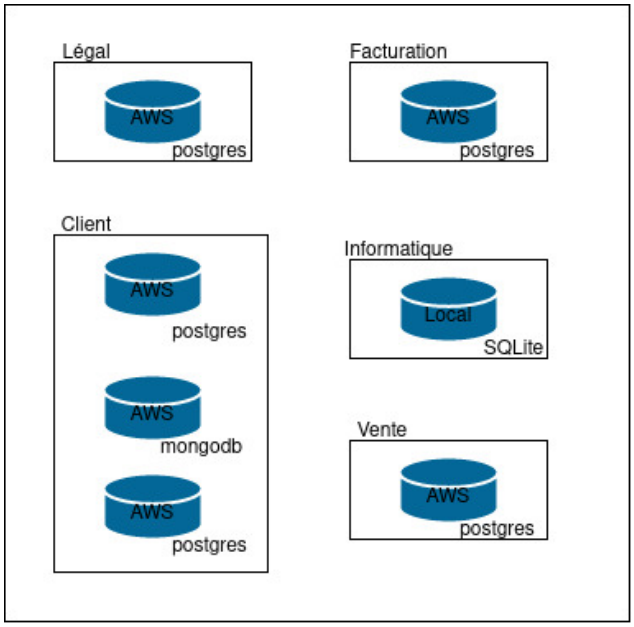


5. Schéma – Phase Vente

Application et Technologies

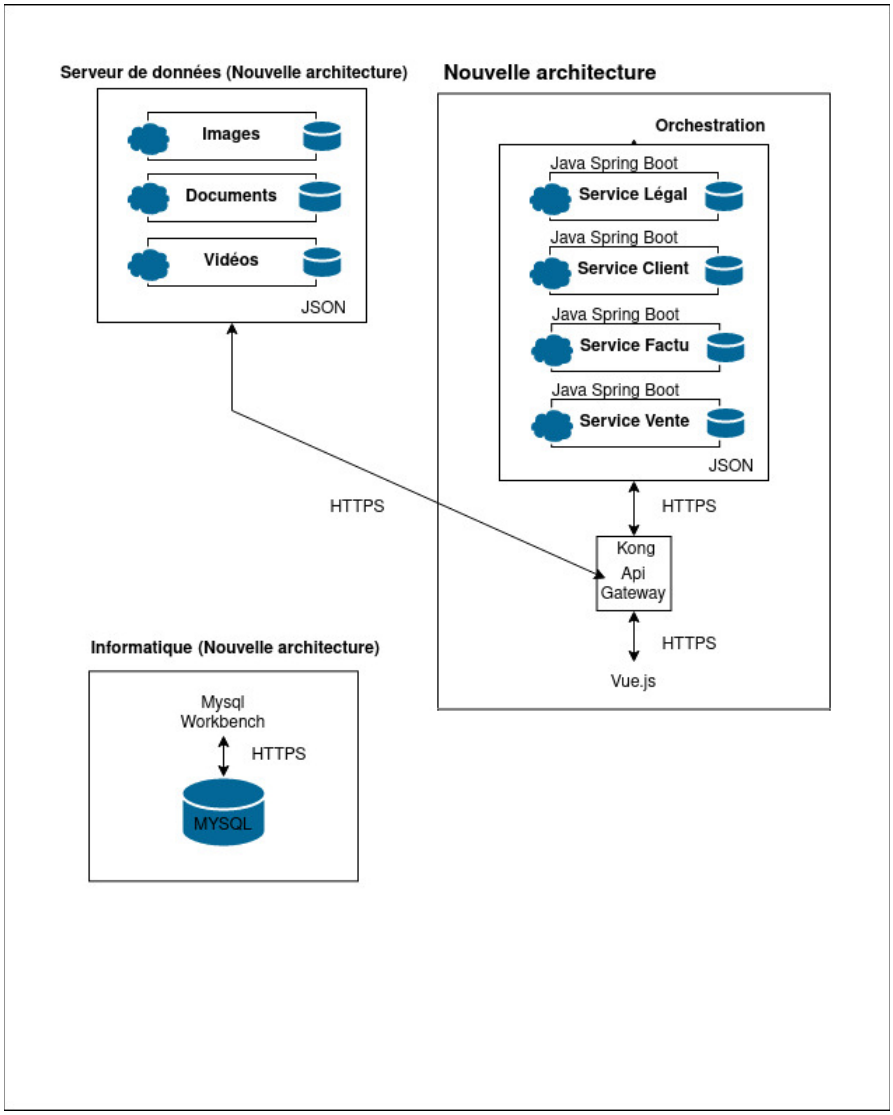


Données

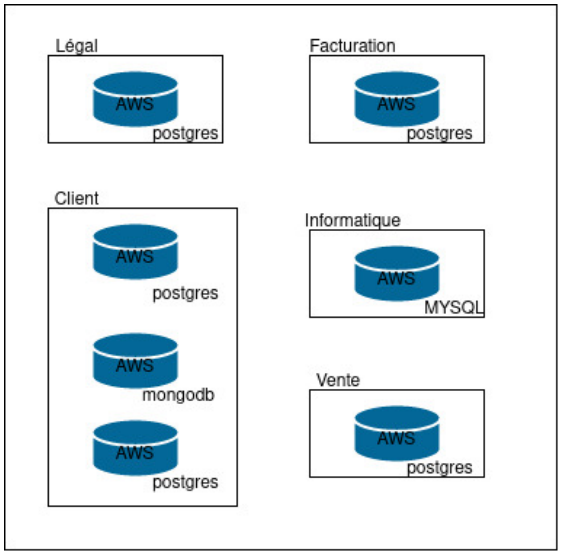


6. Schéma – Phase Informatique et État Final

Application et Technologies



Données



Justification de l'approche architecturale

En conclusion, la transition vers une architecture basée sur les microservices représente un investissement stratégique pour "Les Assureurs Engagés". Cette nouvelle architecture répond non seulement aux exigences actuelles de l'entreprise en termes d'intégration, de performance et de sécurité, mais elle offre également la flexibilité nécessaire pour s'adapter aux changements futurs et aux nouvelles opportunités du marché.