International Journal of Innovation Scientific Research and Review
Vol. 08, Issue, 09, pp.10416-10421, September 2026
Available online at http://www.journalijisr.com
SJIF Impact Factor 2025: 7.913

Research Article

ISSN: 2582-6131

PERFORMANCE ET SECURITE DES SYSTEMES DISTRIBUES: SYNTHESE ET EVALUATION DU MODELE HYBRIDE MARIAMOD

Marie-Jeanne TUDIMUENE MUNDA BOMBESHA
mjyamunda@gmail.com
Chercheure en Génie Logiciel
UNIVERSITE PEDAGOGIQUE DE KANANGA/RÉPUBLIQUE DÉMOCRATIQUE DU CONGO.

Received 09th July 2026; Accepted 10th August 2026; Published online 16th September 2026

RÉSUMÉ

Les organisations doivent traiter des volumes croissants de données tout en garantissant la continuité du service, la cohérence des copies et la protection contre les accès non autorisés. Cet article synthétise une recherché consacrée à l'amélioration de la performance d'un système distribué pour la gestion sécurisée des données. La contribution centrale est le modèle hybride Mariamod-H, qui combine équilibrage adaptatif de charge, traitement parallèle, fragmentation, réplication contrôlée, synchronisation automatique, mise en cache, supervision continue et mécanismes de sécurité. L'étude adopte une démarche de modélisation et d'expérimentation par scénarios dans un contexte de gestion pharmaceutique. Les résultats chiffrés rapportés constituent principalement des objectifs et des valeurs attendues : à 500 utilisateurs simultanés, le temps de réponse attendu passe de 9 200 ms à 1 800 ms, le débit de 85 à 520 requêtes par seconde et le taux d'erreur de 25% à 5%. Le modèle vise également une disponibilité de 99–99,5%, une cohérence rétablie à 99–100% après synchronisation et un basculement en 3–5 secondes. L'article distingue explicitement ces résultats attendus des mesures observées et discute les conditions nécessaires à leur validation reproductible.

Mots-clés: distributed systems; secure data management; adaptive load balancing; replication; synchronization; AES-256; RSA; RBAC; pharmaceutical information system.

INTRODUCTION

Les données organisationnellessont devenues une ressource stratégique. Elles soutient la décision, la coordination des activités, la traçabilité des opérations et la continuité des services. Cette centralité crée une double exigence. Le système doit répondre rapidement à des requêtes concurrentes. Il doit également protéger les informations contre la perte, la divulgation, l'altération et l'usage abusif.

Un système distribué répond à cette exigence en répartissant les traitements, les données et les ressources sur plusieurs nœuds qui coopèrent pour fournir un service cohérent. Cette distribution réduit la dépendance à un serveur unique et facilite la montée en charge. Elle introduit toutefois des difficultés de coordination: latence réseau, pannes partielles, concurrence, synchronisation, cohérence des répliques et contrôle des accès [1][2].

Le problème étudié par la recherché synthétisée est celui d'un système centralisé ou faiblement distribué qui devient lent lorsque la charge augmente et vulnérable lorsque les mécanismes de sécurité restent périphériques. La question de recherche est ainsi reformulée: **comment améliorer la rapidité d'un système distribué sans affaiblir la disponibilité, la cohérence, l'intégrité et la confidentialité des données ?**

L'hypothèse est qu'une architecture intégrée, combinant optimisation et sécurité, peut produire un meilleur compromis qu'une optimisation isolée. L'approche proposée ne traite donc pas le chiffrement, la réplication ou l'équilibrage de charge comme des fonctions indépendantes. Elle les articule au sein d'un même modèle opératoire.

L'objectif de cet article est double. Il s'agit de formaliser l'architecture et les mécanismes du modèle Mariamod-H, et d'interpréter les chiffres expérimentaux avec prudence, en séparant les résultats mesurés des objectifs annoncés.

Cette étude met en évidence une architecture hybride à six niveaux: interface, sécurité, orchestration des requêtes, traitement distribué, gestion des données et supervision. Elle propose une lecture unifiée des indicateurs de performance et de sécurité.

Question	**Comment obtenir simultanément performance, disponibilité, cohérence et sécurité ?**
Hypothèse	La combinaison des mécanismes distribués et cryptographiques améliore le compromis global.
Objet	Le modèle hybride Mariamod-H appliqué à la gestion sécurisée de données pharmaceutiques.
Méthode	Analyse documentaire, modélisation, prototypage et expérimentation par scénarios.

1. Cadre conceptuel et état de l'art

La performance d'un système distribué ne se réduit pas à la vitesse d'exécution. Elle recouvre le temps de réponse, le débit, la latence, la disponibilité, la consommation de ressources, le taux d'erreur et la stabilité sous charge [3]. La sécurité, de son côté, repose sur la confidentialité, l'intégrité et la disponibilité, auxquelles s'ajoutent l'authenticité, la traçabilité et la non-répudiation [4][5].

Ces deux dimensions peuvent entrer en tension. Le chiffrement ajoute un coût de calcul. L'authentification et le contrôle d'accès introduisent des vérifications. La réplication améliore la disponibilité mais augmente le coût de synchronisation. La cohérence forte réduit les anomalies mais peut augmenter la latence d'écriture. Une architecture réaliste doit donc rendre explicites les compromis plutôt que prétendre les supprimer.

Les travaux mobilisés montrent quatre familles de solutions. Les systèmes distribués classiques organisent la communication entre nœuds et la tolérance aux pannes [1][6]. Les bases de données distribuées traitent la fragmentation, l'allocation, la réplication, les transactions et la cohérence [7]. Les travaux sur la performance proposent des métriques et des protocoles de mesure [3]. Les travaux de sécurité structurent l'authentification, le chiffrement, la gestion des clés, la journalisation et le contrôle d'accès [4][8].

La lacune retenue est l'intégration insuffisante de ces familles. Une stratégie de réplication seule ne garantit pas que les copies sont protégées. Un équilibreur de charge seul ne garantit pas que l'utilisateur est autorisé. Un chiffrement seul ne garantit ni la disponibilité ni la cohérence. Mariamod-H se positionne donc comme une architecture de synthèse plutôt que comme un nouvel algorithme cryptographique.

Fig. 1. – Architecture distribuée. L'équilibreur, les nœuds de traitement et la base distribuée forment le socle de la répartition.

La contribution scientifique doit être comprise à ce niveau : le modèle assemble des mécanismes connus dans une chaîne de décision cohérente, adaptée à un contexte organisationnel où les transactions de stock doivent rester traçables et contrôlées.

2. Méthodologie de synthèse et protocole de recherché

La recherche mobilise une approche mixte. L'approche systémique considère le système comme un ensemble de composants interconnectés. L'approche analytique identifie les sources de lenteur, de surcharge et de vulnérabilité. L'approche comparative oppose le système existant au modèle proposé. La modélisation traduit la solution sous forme d'architecture et d'algorithmes. Enfin, l'expérimentation vérifie la faisabilité à travers des scénarios de requêtes, de pannes et d'attaques.

Pour transformer cette démarche en protocole publiable, l'unité expérimentale doit être une exécution complète d'un scénario sous une charge donnée. Chaque condition doit être répétée au moins trente fois ou jusqu'à stabilisation de l'estimation. Les mesures devraient être rapportées par médiane, moyenne, écart interquartile et intervalle de confiance. L'étude fournit les indicateurs et les valeurs attendues, mais pas l'ensemble de ces informations statistiques. Les chiffres de la présente synthèse sont donc présentés comme des objectifs issus de la source, non comme une nouvelle campagne de mesure. Le contexte applicatif est la gestion distribuée de stocks et de commandes pharmaceutiques.

Les scénarios couvrent la consultation de stock, l'enregistrement simultané de commandes, la détection de produits expirés, la panne d'un serveur, le retour d'un dépôt après coupure, la tentative de modification non autorisée, l'altération d'une réplique, le rappel d'un lot et la sauvegarde-restauration.

Scénario	Famille	Indicateurs principaux
Consultation de stock	Performance	Temps de réponse, latence, debit
Commandes simultanées	Concurrence	Erreurs, stock négatif, coherence
Panne d'un nœud	Disponibilité	Temps de détection, basculement, requêtes perdues
Dépôt déconnecté	Synchronisation	Conflits, opérations résolues, cohérence finale
Accès non autorisé	Sécurité	Refus, journalisation, alertes
Réplique altérée	Intégrité	Détection, exclusion, restauration
Rappel de lot	Traçabilité	Temps de localisation et de blocage
Sauvegarde	Continuité	Intégrité, restauration, resynchronization

Le protocole distingue trois niveaux de validation. Le niveau fonctionnel vérifie que les règles métiers sont respectées. Le niveau de performance compare les temps, débits et taux d'erreur. Le niveau de sécurité vérifie les refus d'accès, l'intégrité, la traçabilité et la restauration. Cette séparation évite de conclure à la sécurité sur la seule base d'un gain de vitesse.

3. Architecture du modèle Mariamod-H

Mariamod-H signifie **Modèle hybride d'optimisation distribuée pour la gestion sécurisée des données**. Son architecture comporte une interface client, un gestionnaire de requêtes, un module de sécurité, un équilibreur adaptatif, des nœuds de traitement, une base distribuée, des mécanismes de réplication et de synchronisation, un cache, une supervision et une administration.

Le principe d'orchestration est le suivant. La requête est d'abord authentifiée. Le rôle de l'utilisateur est ensuite vérifié. Le gestionnaire qualifie la demande: lecture, écriture, recherche, rappel, administration ou restauration. L'équilibreur choisit un nœud selon la charge, la proximité des données et la disponibilité. Le traitement est exécuté localement ou distribué. Les écritures critiques sont soumises à une règle de quorum. Enfin, l'opération est journalisée et les répliques sont synchronisées selon la politique applicable.

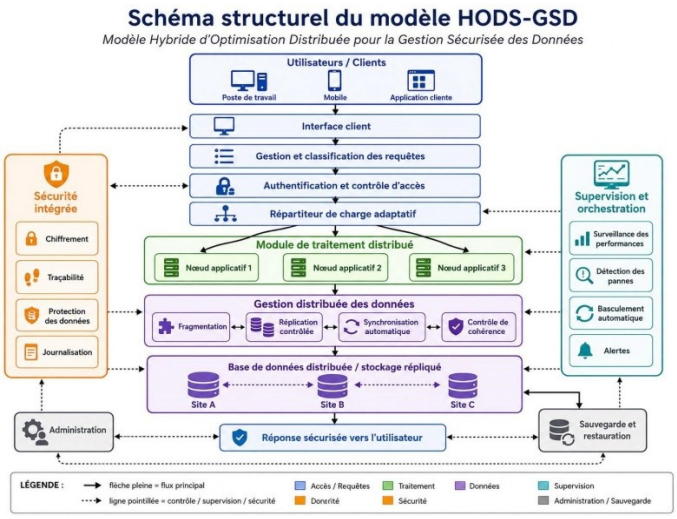

Fig. 2. – Schéma structurel du modèle hybride. Les modules de sécurité et d'optimisation entourent la gestion distribuée des données.

La séparation des responsabilités constitue une propriété importante. Le module de sécurité décide si l'opération est autorisée. L'équilibreur décide où elle doit être traitée. Le module de données décide comment elle est stockée et répliquée. La supervision observe l'état du système et déclenche les mécanismes de basculement. Cette séparation facilite l'audit et limite les dépendances implicites.
L'architecture ne suppose pas que tous les nœuds sont équivalents. Un nœud régional peut privilégier la proximité et le cache. Un nœud central peut gérer la coordination et les opérations sensibles. Une réplique peut être temporairement exclue si son empreinte d'intégrité ne correspond pas à la valeur attendue.

4. Algorithme d'orchestration et complexité

L'algorithme Mariamod reçoit une requête, identifie l'utilisateur, vérifie ses droits, évalue l'état des nœuds, choisit une cible, traite l'opération et consigne le résultat. Pour une lecture, il privilégie le cache ou le nœud le plus proche. Pour une écriture, il sélectionne les nœuds admissibles, applique le verrouillage ou le quorum, réplique la modification et vérifie la cohérence finale.

Pseudo-code conceptuel.

Recevoir(requête) → Authentifier(utilisateur) → Autoriser(rôle, opération) → Évaluer(état des nœuds) → Choisir(cible) → Exécuter(requête) → Répliquer si écriture → Synchroniser → Journaliser → Répondre.

La recherche estime la complexité du cas normal à $O(n + t + \log f + k)$, où n représente le nombre de nœuds ou d'éléments parcourus, t le coût du traitement, f le nombre de fragments et k le nombre de répliques mobilisées. Dans le pire cas, lorsque la synchronisation implique un grand nombre de nœuds et de répliques, la complexité peut atteindre $O(n^2 + t + \log f + k)$.

Cette notation est utile mais ne suffit pas à prédire le temps de réponse réel: celui-ci dépend aussi de la bande passante, des files d'attente, du moteur de stockage, des opérations cryptographiques et du taux de cache.

Une propriété intéressante est la dégradation contrôlée. Si un nœud disparaît, le service peut basculer vers une réplique. Si une réplique est incohérente, elle peut être isolée avant resynchronisation. Si le cache est invalide, la requête revient à la base distribuée. La performance est donc pensée avec des chemins de repli plutôt qu'avec un seul chemin nominal.

Étape	Décision	Risque traité
Authentification	Vérifier l'identité et la session	Usurpation
Autorisation	Comparer rôle et permission	Abus de privilège
Routage	Choisir nœud selon état et charge	Surcharge
Écriture	Appliquer quorum et verrouillage	Conflit
Réplication	Propager vers copies admissible	Perte
Audit	Conserver l'événement signé	Déni d'action

5. Optimisation de la performance

L'équilibrage adaptatif constitue le premier levier. Contrairement à une répartition statique, il prend en compte la charge courante des nœuds, la proximité des données, l'état des connexions et la disponibilité des répliques. Il vise à éviter qu'un serveur principal absorbe toutes les requêtes tandis que d'autres restent sous-utilisés.
Le traitement distribué constitue le second levier. Une requête complexe peut être découpée en sous-tâches exécutées en parallèle, puis regroupée. Cette stratégie est adaptée aux recherches sur plusieurs dépôts, à la localisation d'un lot ou à l'agrégation de stocks. Elle exige toutefois une procédure claire de fusion et de résolution des réponses contradictoires. La fragmentation réduit le volume de données parcouru par chaque opération. Elle doit être conçue à partir de la distribution fonctionnelle et géographique des activités. La réplication contrôlée apporte de la disponibilité sans multiplier inutilement les copies. Le cache évite de solliciter la base pour les informations stables ou fréquemment consultées.

La synchronisation automatique est le mécanisme qui transforme une réplication en état cohérent. L'étude prévoit une comparaison des versions, la détection des conflits, l'application d'une règle de résolution et une vérification finale. Cette séquence doit être accompagnée de journaux permettant de reconstruire la causalité d'une divergence.

Mécanisme	Gain recherché	Condition de réussite
Équilibrage	Réduire la surcharge	État des nœuds disponible en temps utile
Parallélisme	Augmenter le débit	Fusion déterministe des resultants
Fragmentation	Réduire le volume lu	Clé de partition adaptée aux requêtes
Réplication	Améliorer disponibilité	Nombre de copies maîtrisé
Cache	Réduire la latence	Invalidation et cohérence définies
Supervision	Accélérer le basculement	Détection fiable et alerte

La recherche souligne que l'optimisation n'est pas gratuite. La multiplication des copies augmente la synchronisation. Le cache peut servir une donnée périmée. Le parallélisme peut saturer le réseau. L'équilibrage peut déplacer une charge vers un nœud dont les données sont moins proches. L'intérêt de Mariamod-H réside dans la combinaison adaptative de ces mécanismes, non dans l'activation indistincte de tous les leviers.

6. Sécurité intégrée des données

Le modèle retient une chaîne de sécurité couvrant l'identité, les permissions, la confidentialité, l'intégrité, la traçabilité, la réplication et la sauvegarde. L'authentification établit l'identité du pharmacien, du gestionnaire ou de l'administrateur. Le contrôle d'accès basé sur les rôles, ou RBAC, associe chaque rôle à un ensemble de permissions [9]. Une opération est refusée lorsque le rôle ne l'autorise pas, même si l'utilisateur est authentifié.

La solution cryptographique associe RSA et AES-256. RSA est utilisé pour protéger l'échange ou l'encapsulation de clés. AES-256 chiffre effectivement les données. Cette combinaison hybride est cohérente avec la distinction entre échange sécurisé de clés et chiffrement de volume. Elle doit néanmoins préciser le mode d'opération, la gestion des nonces, la rotation des clés, la protection des clés privées et la résistance aux compromissions.

L'étude décrit une clé maître et des sous-clés fonctionnelles: Kauth pour l'authentification, Kdata pour les données sensibles, Ksession pour les communications, Khmac pour l'intégrité des journaux, Krep pour la réplication et Kbackup pour les sauvegardes. Cette séparation limite la portée d'une compromission, à condition que la dérivation, le stockage et la rotation soient formellement spécifiés.

La journalisation rend les opérations sensibles traçables. Un événement doit au minimum enregistrer l'identité, le rôle, l'action, la ressource, l'horodatage, le résultat et l'empreinte d'intégrité. Les journaux eux-mêmes doivent être protégés contre l'altération. La sauvegarde chiffrée complète le dispositif en fournissant un mécanisme de reprise après perte ou corruption.

Contrôle	Fonction	Test associé
Identité	Vérifier l'utilisateur	Connexion légitime
RBAC	Limiter l'action au rôle	Modification interdite
AES-256	Protéger les données	Lecture sans clé
RSA	Protéger l'échange de clé	Transmission
HMAC / empreinte	Détecter l'altération	Réplique modifiée
Journal	Prouver l'action	Audit
Backup chiffré	Restaurer après perte	Récupération

7. Architecture de sécurité et gestion des incidents

La sécurité est intégrée au cycle de vie de la requête. Avant traitement, le système identifie l'utilisateur et vérifie la permission. Pendant le traitement, les communications sont protégées et les données sensibles sont chiffrées. Après traitement, l'opération est journalisée et les répliques sont contrôlées. En cas d'anomalie, la supervision déclenche une alerte et peut isoler le nœud concerné.

Le scénario d'altération d'une réplique illustre ce fonctionnement. La copie principale contient 1 250 boîtes. Une réplique contient 2 500 boîtes après modification frauduleuse. Le système compare les versions et les empreintes, exclut la copie divergente, restaure la valeur correcte et journalise l'incident. Le résultat attendu est le retour de toutes les copies à 1 250 boîtes.

Le scénario d'accès non autorisé met en jeu une tentative de modification de prix, de suppression de lot ou de création d'un administrateur par un magasinier. La décision attendue est le refus de l'opération et la conservation d'une trace. Plusieurs tentatives répétées doivent provoquer une alerte, sans bloquer indistinctement les utilisateurs légitimes.

Le scénario de rappel de lot teste la traçabilité de bout en bout. Le système doit rechercher le lot dans tous les dépôts, identifier les quantités et les structures livrées, bloquer toute distribution supplémentaire et notifier les parties concernées. La performance se mesure alors en secondes ou minutes de localisation, mais la sécurité exige aussi que la décision soit autorisée et auditable.

La supervision continue est ainsi un mécanisme de sécurité autant qu'un mécanisme de performance. Elle observe la charge, les erreurs, la latence, les échecs d'authentification, l'état des répliques et les événements de synchronisation. Son efficacité dépend de seuils correctement calibrés et d'une procédure d'escalade documentée.

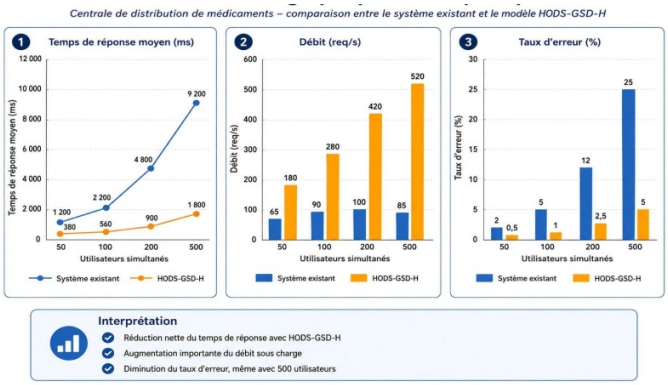

Fig. 3. – Synthèse graphique des valeurs : temps de réponse, débit et taux d'erreur. Ces valeurs sont reproduites comme objectifs, et non comme mesures indépendamment vérifiées.

8. Protocole expérimental et critères de validation

L'expérimentation compare un système existant à Mariamod-H. Les conditions de charge annoncées sont 50, 100, 200 et 500 utilisateurs simultanés. Les indicateurs de performance sont le temps de réponse, le débit et le taux d'erreur. Les indicateurs de résilience sont le temps de détection, le temps de basculement, le nombre de requêtes perdues et la durée d'interruption. Les indicateurs de sécurité sont le taux de refus des accès non autorisés, l'intégrité des copies, la traçabilité et la réussite de la restauration.

Dimension	Mesure	Système existant	Mariamod-H
Performance	Temps	Élevé sous charge	Réduit par cache et routage
Capacité	Débit	Limité par serveur unique	Distribué entre nœuds
Fiabilité	Disponibilité	94–96 % attendu	99 –99,5 % attendu
Résilience	Basculement	Arrêt ou manuel	3 –5 s attendues
Cohérence	Répliques	Variable	99–100 % après synchronization
Sécurité	Accès interdits	Partiellement bloqués	Proche de 100 % attend
Traçabilité	Opérations sensible	Limitée	100 % attendue

Les critères de validation sont fonctionnels et quantitatifs. Une solution satisfaisante doit répondre plus rapidement, traiter davantage de requêtes, éviter les stocks négatifs, bloquer les lots expirés, continuer après la panne d'un nœud, restaurer les répliques, refuser les accès non autorisés, journaliser les opérations importantes et restaurer les données après perte.

9. Résultats de performance rapportés

Le tableau suivant reprend fidèlement quatre valeurs à prendre en charge. La colonne d'amélioration est calculée à partir des valeurs de la source et doit être interprétée comme un gain cible.

Utilisateurs	Existant	Mariamod-H	Réduction temps	Nature
50	1 200 ms	380 ms	68,3 %	Objectif
100	2 200 ms	560 ms	74,5 %	Objectif
200	4 800 ms	900 ms	81,3 %	Objectif
500	9 200 ms	1 800 ms	80,4 %	Objectif

La tendance attendue est non linéaire. Le système existant se dégrade fortement lorsque la charge augmente, alors que Mariamod-H conserve un temps de réponse inférieur grâce à la répartition et au cache. Le gain relatif est maximal à 200 utilisateurs dans le tableau, mais le gain absolu devient le plus important à 500 utilisateurs: 7 400 ms sépareraient les deux solutions.

Le débit attendu est également supérieur. La source rapporte les couples suivants : 65 contre 180 requêtes par seconde en charge faible, 90 contre 280 en charge moyenne, 100 contre 420 en charge élevée et 85 contre 520 en charge critique. Ces nombres traduisent l'effet recherché de la parallélisation, mais ils ne peuvent être qualifiés de résultats expérimentaux définitifs sans préciser la charge exacte et la méthode de mesure.

Charge	Existant	Mariamod-H	Gain relatif
Faible	65 req/s	180 req/s	176,9 %
Moyenne	90 req/s	280 req/s	211,1 %
Élevée	100 req/s	420 req/s	320,0 %
Critique	85 req/s	520 req/s	511,8 %

Le gain attendu en débit est cohérent avec une architecture multi-nœuds, mais sa validité dépend de l'absence de goulot d'étranglement au niveau de la base, du réseau ou du chiffrement. Une mesure robuste devra donc inclure l'utilisation CPU, la mémoire, le réseau, la taille des files et la distribution des latences, notamment le 95e et le 99e percentile.

10. Taux d'erreur, disponibilité et résilience

Cette recherche rapporte les taux d'erreur attendus suivants : 2% contre 0,5 % à 50 utilisateurs, 5% contre 1% à 100, 12% contre 2,5% à 200 et 25% contre 5% à 500. Le modèle vise ainsi à limiter les erreurs liées à la surcharge, aux délais d'attente, aux connexions perdues, aux conflits de commandes et à l'indisponibilité d'un serveur.

Utilisateurs	Existant	Mariamod-H	Réduction relative
50	2 %	0,5 %	75 ,0 %
100	5 %	1 %	80 ,0 %
200	12 %	2,5 %	79 ,2 %
500	25 %	5 %	80 ,0 %

Les indicateurs globaux attendus sont les suivants. La disponibilité passerait de 94–96% à 99–99,5%. Le basculement, auparavant total ou manuel, serait réalisé en 3–5 secondes. Le taux de succès du cache serait nul dans le système existant et atteindrait 70–85% avec Mariamod-H. La cohérence des répliques, variable dans le système existant, serait rétablie à 99–100% après synchronisation. Les lots expirés seraient bloqués à 100%, les commandes créant un stock négatif seraient nulles et les opérations sensibles seraient toutes journalisées.

Ces chiffres décrivent une amélioration fonctionnelle ambitieuse. Ils ne démontrent pas, à eux seuls, la robustesse statistique du modèle. La disponibilité doit être mesurée sur une durée suffisante. Le basculement doit être répété sur différents nœuds et sous différentes charges. La cohérence doit être évaluée après des écritures concurrentes, des coupures et des retards réseau.

La sécurité doit être testée par des scénarios d'abus et non uniquement par des cas nominaux.

La force du protocole est d'associer performance et résilience. La faiblesse est que la source ne documente pas suffisamment la matérialisation des mesures. La contribution est donc mieux présentée comme un modèle validé par scénarios et par objectifs quantitatifs, en attendant une réplication expérimentale complète.

DISCUSSION

La synthèse permet de dégager trois résultats conceptuels. Premièrement, le gain de performance provient d'une combinaison de mécanismes: un cache sans invalidation est dangereux, une réplication sans contrôle est incohérente et un équilibrage sans visibilité peut déplacer le problème. Deuxièmement, la sécurité est distribuée dans tout le cycle de traitement. Elle ne peut pas être ajoutée comme une couche décorative après l'optimisation. Troisièmement, la gestion des incidents est une propriété de l'architecture, pas une procédure extérieure.

Les valeurs attendues suggèrent un effet croissant sous forte charge. Cette propriété est importante pour la gestion de stocks, car la simultanéité des commandes peut provoquer une quantité négative ou une double attribution si les écritures ne sont pas coordonnées. Le quorum, le verrouillage et la synchronisation doivent donc être évalués conjointement avec le débit. L'usage d'AES-256 et de RSA est conceptuellement adapté, mais il ne suffit pas à caractériser une sécurité publiable. Il faudrait documenter le mode AES, la source d'aléa, la rotation, la destruction des clés, la protection des sauvegardes et la gestion des certificats. De même, RBAC devrait être décrit par une matrice de permissions et par des tests de séparation des tâches. Le modèle peut être rapproché des principes de conception des systèmes fiables, scalables et maintenables [2][7]. Il reprend aussi les bonnes pratiques de l'analyse de performance expérimentale [3]. Son originalité n'est pas une primitive isolée, mais une intégration orientée vers un cas de gestion opérationnelle. Cette orientation est pertinente pour une revue appliquée, à condition de mieux distinguer les éléments démontrés, simulés, attendus et proposés.

Aspect	Apport de Mariamod-H	Condition de publication
Performance	Répartition, cache, parallélisme	Mesures reproductibles et percentiles
Disponibilité	Réplication et basculement	Campagnes de pannes répétées
Cohérence	Quorum et synchronisation	Jeu de données et conflits décrits
Sécurité	RBAC, chiffrement, journal	Modèle de menace et tests adversariaux
Traçabilité	Journalisation des opérations	Intégrité et rétention des logs

Implications pratiques et scientifiques

Pour une organisation, Mariamod-H fournit une grille de conception. La première étape consiste à cartographier les données, les rôles, les opérations sensibles et les dépendances de disponibilité. La deuxième consiste à définir les fragments et les répliques à partir des requêtes réelles. La troisième consiste à intégrer l'authentification, le RBAC, le chiffrement et les journaux avant la mise en production. La quatrième consiste à tester les pannes et les restaurations, et non seulement les opérations nominales.

Pour l'ingénierie des données, le modèle rappelle qu'une réplication efficace n'est pas seulement une duplication. Elle doit être gouvernée par une politique de placement, une stratégie de synchronisation et une procédure de résolution des conflits. Pour l'ingénierie de sécurité, le modèle montre que la protection des données inclut le stockage, le transit, la journalisation, les sauvegardes et les copies temporaires.

Pour la recherche, plusieurs prolongements sont possibles. Une première piste consiste à comparer plusieurs politiques d'équilibrage sous charges hétérogènes. Une deuxième consiste à modéliser le coût cryptographique par taille de message et par type de stockage. Une troisième consiste à étudier la cohérence adaptative : forte pour les écritures critiques, éventuellement différée pour les données non sensibles à la latence. Une quatrième consiste à formaliser la détection d'anomalies et la prédiction des attaques annoncées.

La contribution est particulièrement pertinente dans les environnements où une erreur de donnée a une conséquence opérationnelle immédiate. Dans une chaîne pharmaceutique, une quantité incorrecte, un lot expiré non bloqué ou une commande perdue peut affecter la continuité du service. La sécurité et la performance doivent alors être évaluées ensemble, selon des critères métiers compréhensibles.

Partie prenante	Bénéfice attend
Administrateur	Supervision, audit, restauration et contrôle des rôles
Gestionnaire	Accès plus rapide et continuité du service
Pharmacien	Consultation et commande avec contrôle de coherence
Responsable sécurité	Chiffrement, traçabilité et détection d'anomalies
Chercheur	Cadre d'évaluation intégrant performance et sécurité

CONCLUSION

Cette étude consacrée à l'amélioration de la performance d'un système distribué pour la gestion sécurisée des données. La réponse proposée est le modèle Mariamod-H, une architecture hybride qui associe équilibrage adaptatif, traitement distribué, fragmentation, réplication contrôlée, synchronisation, cache, supervision et sécurité intégrée. Elle montre que la contribution est cohérente avec les besoins d'un système de gestion pharmaceutique. Elle vise à réduire le temps de réponse et le taux d'erreur, à augmenter le débit, à maintenir les répliques cohérentes, à assurer le basculement après panne et à refuser les opérations non autorisées. Les données indiquent des objectifs ambitieux : à 500 utilisateurs, 1 800 ms contre 9 200 ms, 520 contre 85 requêtes par seconde et 5 % contre 25 % d'erreurs. Ils doivent toutefois être rapportés comme des valeurs attendues tant qu'une campagne de mesures documentée n'est pas disponible. La principale conclusion scientifique est que performance et sécurité doivent être conçues comme des propriétés conjointes. La distribution améliore la capacité mais crée des problèmes de cohérence. La réplication augmente la disponibilité mais impose une synchronisation. Le chiffrement protège les données mais ajoute un coût. Le modèle proposé est intéressant parce qu'il rend ces interactions explicites au sein d'une architecture unique.Un système distribué performant n'est publiable comme système fiable que si son gain de vitesse, sa résilience et sa sécurité sont mesurés avec un protocole reproductible.

BIBLIOGRAPHIE

[1] G. Coulouris, J. Dollimore, T. Kindberg, et G. Blair, Distributed Systems: Concepts and Design, 5e éd. Addison-Wesley, 2012.

[2] A. S. Tanenbaum et M. Van Steen, Distributed Systems: Principles and Paradigms, 2e éd. Pearson Prentice Hall, 2007.

[3] M. Kleppmann, Designing Data-Intensive Applications. O'Reilly Media, 2017.

[4] W. Stallings, Cryptography and Network Security: Principles and Practice, 7e éd. Pearson, 2017.

[5] M. E. Whitman et H. J. Mattord, Principles of Information Security, 6e éd. Cengage Learning, 2018.

[6] A. D. Kshemkalyani et M. Singhal, Distributed Computing: Principles, Algorithms, and Systems. Cambridge University Press, 2008.

[7] M. T. Ozsu et P. Valduriez, Principles of Distributed Database Systems, 4e éd. Springer, 2020.

[8] M. T. Goodrich et R. Tamassia, Introduction to Computer Security. Addison-Wesley, 2011.

[9] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, et C. E. Youman, « Role-Based Access Control Models », IEEE Computer, vol. 29 , no 2, 1996.

[10] R. Jain, The Art of Computer Systems Performance Analysis. Wiley, 1991.

[11] K. Hwang, G. C. Fox, et J. J. Dongarra, Distributed and Cloud Computing. Morgan Kaufmann, 2012.

[12] J. F. Kurose et K. W. Ross, Computer Networking: A Top-Down Approach, 8e éd. Pearson, 2021.

[13] M. Bishop, Computer Security: Art and Science. Addison-Wesley, 2003.

[14] B. Schneier, Applied Cryptography, 2e éd. Wiley, 1996.

[15] C. P. Pfleeger, S. L. Pfleeger, et J. Margulies, Security in Computing, 5e éd. Pearson, 2015.

[16] A. Silberschatz, H. F. Korth, et S. Sudarshan, Database System Concepts, 7e éd. McGraw-Hill, 2020.

[17] N. A. Lynch, Distributed Algorithms. Morgan Kaufmann, 1996.

[18] H. Attiya et J. Welch, Distributed Computing: Fundamentals, Simulations, and Advanced Topics, 2e éd. Wiley, 2004.

[19] E. Rescorla, SSL and TLS: Designing and Building Secure Systems. Addison-Wesley, 2001. [20] G. Weikum et G. Vossen, Transactional Information Systems. Morgan Kaufmann, 2002.

****