Développement d'une API sécurisée multi-tenant avec chiffrement et recherche

Une entreprise vous a missionné pour concevoir le cœur d'une plateforme SaaS destinée à plusieurs organisations (tenants). Chaque tenant doit accéder à ses propres données, sans risque d'interférence ou d'accès non autorisé. Certaines données sont sensibles et doivent impérativement être chiffrées. Néanmoins, les utilisateurs doivent pouvoir effectuer des recherches simples sur ces données.

Missions à accomplir

1. Modélisation multi-tenant
    - Choix du mode d'isolation (champ tenant_id, schéma, ou base séparée)
    - Implémentation de l'isolation logique
2. Chiffrement des données
    - Sélection des champs sensibles à protéger
    - Mise en place d'un chiffrement applicatif
    - Gestion sécurisée des clés
3. Recherche sur données protégées
    - Maintien d'une capacité de recherche (sur champs dérivés ou non chiffrés)
    - Justification des compromis sécurité / fonctionnalité
4. Développement de l'API
    - Authentification (JWT, middleware tenant-aware)
    - Endpoints CRUD
    - Endpoint(s) de recherche sécurisée
5. Documentation & Présentation
    - Documentation claire et complète
    - Présentation en soutenance

Points d'avancement recommandés (jalons)

Ces étapes ne sont pas imposées mais servent de repères pour organiser votre travail :

🕐 Jour 1 après-midi :
- Compréhension du sujet et constitution des groupes
- Choix du type d'architecture multi-tenant
- Premiers schémas de la base de données

🕑 Jour 2 matin :
- Implémentation du modèle multi-tenant dans la base
- Mise en place de l'initialisation des données par tenant

🕑 Jour 2 après-midi :
- Implémentation du chiffrement des champs sensibles
- Test d'un CRUD simple avec chiffrement côté serveur

🕓 Jour 3 matin :
- Implémentation d'une stratégie de recherche (hash, champs dérivés...)
- Tests sur les performances ou la pertinence de la recherche

🕓 Jour 3 après-midi :
- Ajout de l'authentification (JWT ou autre)
- Sécurisation des accès tenant-aware

🕐 Jour 4 matin :
- Finitions du code et tests globaux
- Préparation de la soutenance (slides éventuelles, plan de démo)

🕐 Jour 4 après-midi :
- Soutenance et démonstration par groupe

Livrables attendus

Code (archive zip)
Documentation incluant :
- Instructions d'installation
- Description du modèle
- Détail du chiffrement et de la recherche

Rendu complet sur la dernière journée du module à 12h30 (Fixe : votre formateur devra avoir reçu les livrables à cette date).

Une démo fonctionnelle de l'API (via Postman ou autre) sera à présenter en soutenance.

Grille d’évaluation

Critère	Points
Conception multi-tenant *Structure, cohérence, clarté des règles d'isolement*	5
Implémentation du chiffrement *Pertinence de la stratégie, bonne utilisation des outils*	5
Capacité de recherche *Recherche fonctionnelle et adaptée*	4
Fonctionnalités API *CRUD, auth, sécurité d'accès*	4
Qualité du code & doc *Organisation, clarté, lisibilité*	2
Démo & présentation *Clarté, pertinence des choix, capacité à répondre aux questions*	5

Ressources

1. Multi-tenancy : approches et modèles
https://www.prisma.io/docs/concepts/more/multi-tenancy
https://docs.djangoproject.com/en/4.2/topics/db/multi-db/
https://www.techtarget.com/searchdatamanagement/definition/multi-tenant-architecture
https://docs.nestjs.com/recipes/mongodb-multi-tenancy

2. Chiffrement des données
https://www.mongodb.com/docs/manual/core/security-client-side-encryption/
https://blog.cryptographyengineering.com/2015/08/13/what-is-field-level-encryption/
https://www.vaultproject.io/docs/secrets/kv/kv-v2
https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html

3. Recherche sur données chiffrées
https://blog.adelean.com/2021/10/searching-on-encrypted-data/
https://docs.microsoft.com/en-us/azure/architecture/patterns/search-secured-data
https://signal.org/docs/
https://blog.cryptomove.com/searching-over-encrypted-data-57db531c3d60
https://www.cs.utexas.edu/~shmat/shmat_cacm12.pdf (Modern Cryptography and Search)
https://eprint.iacr.org/2013/169.pdf (Practical Techniques for Searches on Encrypted Data)
https://privacytools.seas.harvard.edu/files/privacytools/files/search_on_encrypted_data.pdf
https://www.researchgate.net/publication/339508651_Searching_Encrypted_Data_Overview

Exemple de contexte projet : Plateforme de gestion santé pour mutuelles

Votre équipe travaille sur une plateforme SaaS en marque blanche destinée à des mutuelles partenaires. Chaque mutuelle dispose d'un espace dédié pour :

- créer et publier ses offres de remboursement santé,
- enregistrer ses clients et leurs demandes de remboursement,
- suivre les échanges liés aux dossiers (pièces justificatives, messages…).

La plateforme est utilisée par plusieurs mutuelles simultanément. L'application doit garantir que :

- une mutuelle ne peut jamais voir les données des autres,
- les données sensibles (santé, RIB, historique médical, etc.) sont chiffrées,
- les mutuelles peuvent rechercher facilement des demandes ou des assurés dans leur propre périmètre.

🔐 Exemples de données sensibles

- Numéro de sécurité sociale
- Détail des soins (optique, dentaire, hospitalisation…)
- Justificatifs de remboursement (documents médicaux)
- Coordonnées bancaires

🔎 Exemples de recherche

- Rechercher les demandes d'un client par nom ou n° de sécurité sociale
- Lister toutes les demandes en attente sur une période
- Rechercher un type de soin précis (optique, dentaire…)