



Table des matières

| | |
|---------------------------------------|---|
| Dictionnaire de données | 2 |
| Modèle Logique de données : | 2 |
| Modèle Conceptuel de données : | 3 |
| Schéma D'architecture Technique | 4 |
| Sécurité et chiffrement | 4 |

Dictionnaire de données

| Libellé | Type | Taille | Description |
|----------------------|-------------|--------|--|
| idNote | Primary key | | Id d'une note |
| note_title | VARCHAR | 250 | Titre d'une note |
| note_content | TEXT | | Contenu d'une note |
| note_creation_date | Date/Time | | Date de création d'une note |
| note_update_date | Date/Time | | Date de dernière modification d'une note |
| note_delete_date | Date/Time | | Date de suppression d'une note |
| isDeleted | TINYINT | | Est en corbeille ou non |
| idUser | Primary key | | Id d'un utilisateur |
| user_login | VARCHAR | 50 | Login d'un utilisateur |
| user_password | VARCHAR | 190 | Mot de passe d'un utilisateur |
| user_secret_sentence | VARCHAR | 190 | Phrase secrète de récupération |
| idColor | Primary key | | Id d'une couleur |
| color_name | VARCHAR | 20 | Nom d'une couleur |
| color_hex | VARCHAR | 20 | Hex d'une couleur |
| idImage | Primary key | | Id d'une image |
| image_name | VARCHAR | 250 | Nom d'une image |
| Image_data | LOB | | Données binaires de l'image |
| Image_salt | VARCHAR | 80 | Sel de l'image |
| Image_iv | VARCHAR | 24 | IV de l'image |
| Image_mime_type | VARCHAR | 250 | Informations/type d'image |
| Image_creation_date | DATE | | Date de création de l'image |
| idAlbum | Primary key | | Id d'un album |
| album_name | VARCHAR | 100 | Nom d'un album |

Modèle Logique de données :

users = (idUser INT, user_login VARCHAR(50), user_password VARCHAR(190), user_secret_sentence VARCHAR(200));

colors = (idColor INT, color_name VARCHAR(20), color_hexa VARCHAR(16));

albums = (idAlbum INT, album_name VARCHAR(100), #idColor);

notes = (idNote INT, note_title VARCHAR(250), note_content TEXT, note_creation_date DATETIME, note_update_date DATETIME, note_delete_date DATETIME, #idColor, #idUser);

images = (idImage INT, image_name VARCHAR(250), image_data TEXT, image_salt VARCHAR(250), image_iv VARCHAR(24), image_mime_type VARCHAR(250), image_creation_date DATE, #idAlbum*, #idUser);

Modèle Conceptuel de données :

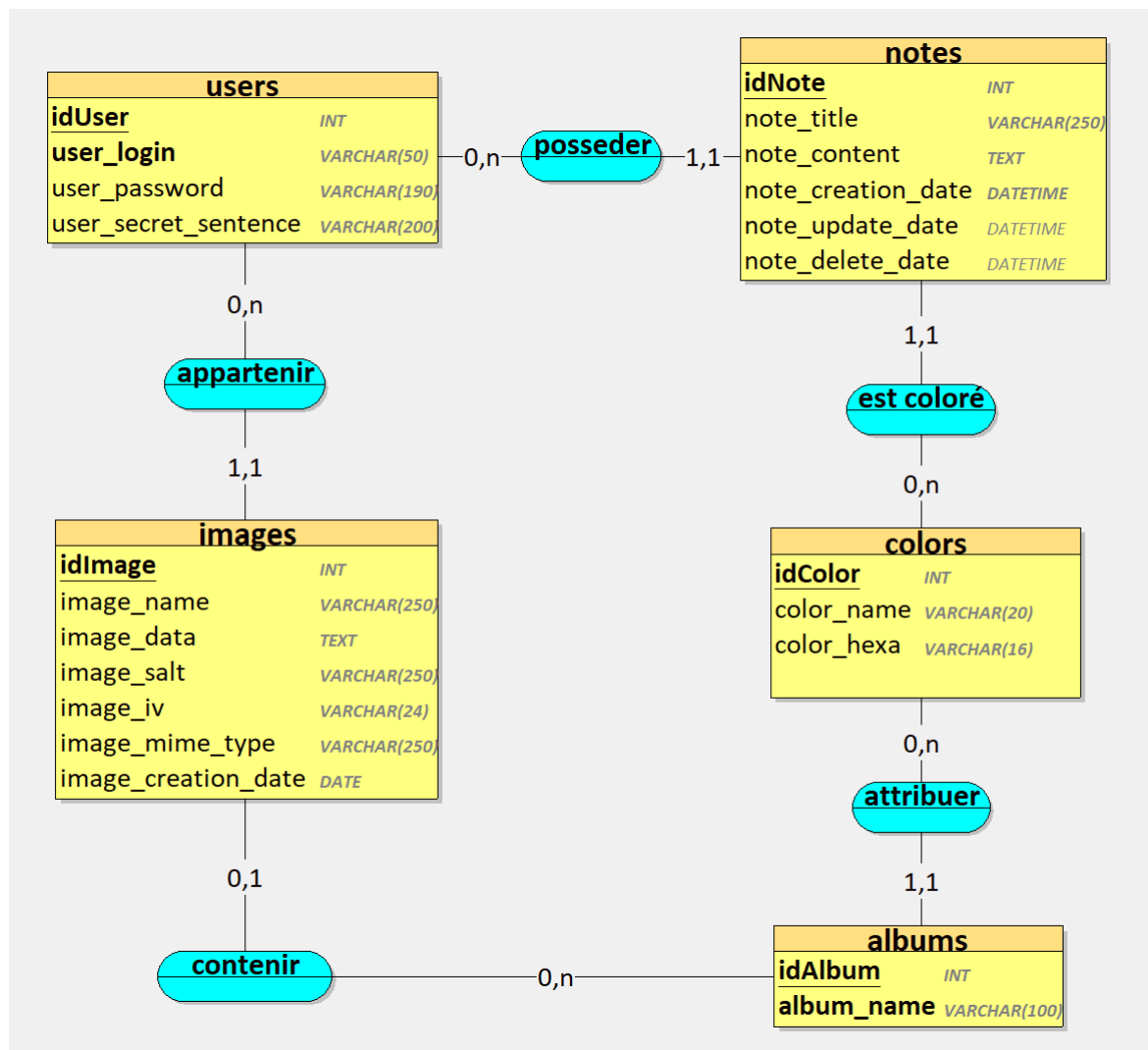
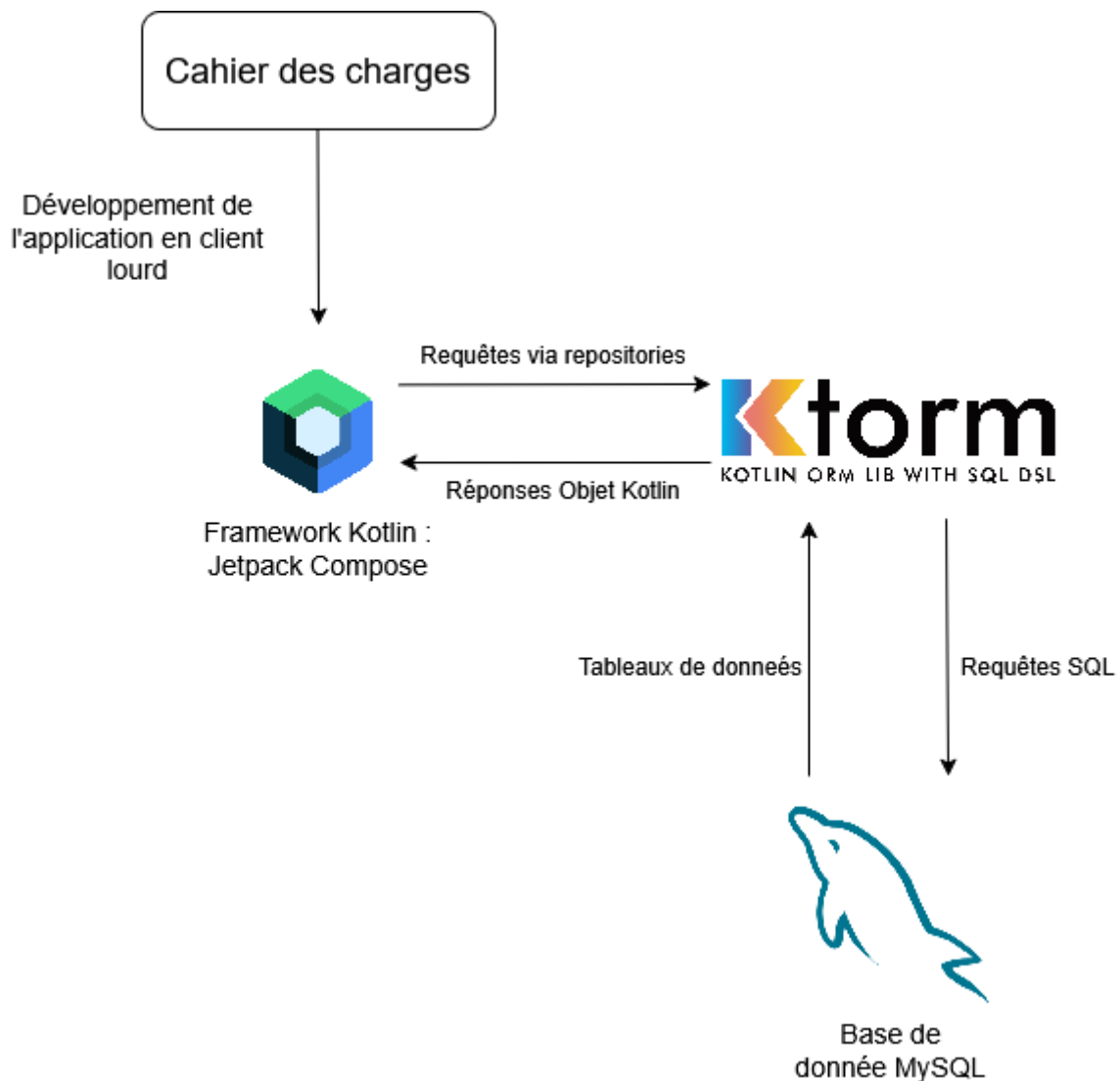


Schéma D'architecture Technique



Sécurité et chiffrement

Les données sensibles de l'utilisateur (ex. : contenu des notes, donnée des images) sont chiffrées côté serveur à l'aide de l'algorithme AES-256 en mode GCM (Galois/Counter Mode). Ce mode garantit à la fois la confidentialité (personne ne peut lire les données sans autorisation) et l'intégrité (toute modification ou falsification est automatiquement détectée grâce au tag d'authentification GCM).

La clé AES-256 utilisée pour le chiffrement n'est jamais stockée. Elle est dérivée dynamiquement à partir du mot de passe de l'utilisateur, via l'algorithme PBKDF2WithHmacSHA256 (65 536 itérations), combiné à un sel (salt) aléatoire de 16 octets.

Le mot de passe utilisateur est protégé avec Argon2, un algorithme de hachage de dernière génération conçu pour résister aux attaques par GPU et les attaques de brute-force. Il impose un coût de calcul élevé par tentative, ce qui ralentit considérablement les attaques.

Contenu d'une donnée chiffrée

Chaque note et image chiffrée contient :

- le salt (16 octets),
- un IV (vecteur d'initialisation) (12 octets),
- et le texte/les données binaires chiffré avec tag d'authentification, encodé en Base64.

Ces éléments sont visibles mais ne permettent en aucun cas de retrouver la clé de chiffrement sans le mot de passe utilisateur.

Estimation de la résistance au brute-force

La politique de mot de passe de l'application est qu'il doit faire 10 caractères minimum, avec la complexité suivante :

- 2 majuscules
- 2 minuscules
- 2 chiffres
- 2 caractères spéciaux

Calcul de la complexité de ce mot de passe :

- Majuscules : 26 lettres en majuscule (A-Z)
- Minuscules : 26 lettres en minuscule (a-z)
- Chiffres : 10 chiffres (0-9)
- Caractères spéciaux : 32 symboles spéciaux possibles (par exemple, !@#\$%^&*()_+[]{}|:;,.<>?)

Nombre total de possibilités pour chaque caractère :

- Majuscules : 26
- Minuscules : 26

- Chiffres : 10
- Caractères spéciaux : 32

Le mot de passe a 10 caractères. La complexité totale de ce mot de passe devient :

Total des combinaisons possibles = $(26+26+10+32)10=94^{10}\approx 5.5\times 10^{19}$ combinaisons

Coût de calcul par tentative :

- Argon2 : Chaque tentative prend environ 200 ms pour être traitée sur un serveur standard.

Attaque avec 10 machines :

- Chaque machine peut effectuer environ 5 tentatives par seconde (en raison de la puissance de calcul limitée par l'algorithme Argon2).
- Avec 10 machines, le taux d'attaque total devient 50 tentatives par seconde.

Temps nécessaire pour tester toutes les combinaisons :

- Le temps nécessaire pour tester toutes les combinaisons devient :

$(5.5\times 10^{19})/50=1.1\times 10^{18}$ secondes

Ce qui équivaut à environ **35 milliards d'années**.

En résumé

- Même si un attaquant obtient les données chiffrées, il ne peut pas déchiffrer sans connaître le mot de passe utilisateur.
- La combinaison Argon2 + PBKDF2 + AES-256-GCM constitue une protection extrêmement robuste, avec des milliards d'années nécessaires pour brute-forcer un mot de passe complexe de 10 caractères.
- En cas de vol de la base de données, les données sensibles restent complètement inaccessibles sans le mot de passe utilisateur.

Conclusion : *Cette approche rend pratiquement impossible pour un attaquant de récupérer les données, même avec un ensemble de 10 machines performantes et des mots de passe complexes.*