

Projet Gestion de Ruchers - Résumé Complet

Projet Gestion de Ruchers - Résumé Complet

Contexte du Projet

Objectif principal : Répondre aux problématiques apportées par le changement climatique et les maladies aux colonies d'abeilles en aidant les apiculteurs à mieux prévenir et intervenir. Faciliter le suivi des cheptel et ruches.

Problématique : Les apiculteurs rencontrent des difficultés de plus en plus récurrentes et impactantes. Le projet vise à développer une application pour assurer un suivi de l'état de santé des ruches ainsi que des interventions sur chacune.

Approche : Collaboration avec un apiculteur client qui guidera l'équipe dans la découverte du métier et toutes les étapes de développement.

Équipe Projet

- **Sayfoutdinov Timour** - Chef de projet
 - **Joly Dorian** - Développement Front-end
 - **Gesse Corentin** - Développement Back-end
 - **Ait Ouarab Mélissa** - DevOps
-

Fonctionnalités Clés de l'Application

Gestion Générale

- **Interface intuitive** : Application très facile d'utilisation
- **Gestion par rucher**: Organisation des ruches en groupes de 20 à 50 ruches
- **Gestion par ruche individuelle** : Suivi détaillé de chaque ruche avec immatriculation unique
- **Géolocalisation** : Placement et suivi GPS des ruchers
- **Prévention des vols** : Système d'alerte (une ruche vaut ~150€, un essaim ~150€, ruche complète ~400€)

Suivi Sanitaire

- **Suivi de santé** : Monitoring de l'état des colonies
- **Gestion des traitements** : Traçabilité des interventions (Apivar, acide oxalique pour le Varroa)
- **Gestion des maladies** : Identification et suivi des pathologies
- **Principe de quarantaine** : Isolation virtuelle des ruches malades (statut "Malade")
- **Sélection génétique** : Identification des colonies résistantes pour l'élevage futur

Traçabilité

- **Généalogie des abeilles** : Suivi de la lignée des essaims
- **Suivi des reines** : Âge (code couleur international), lignée, note de douceur
- **Historique des interventions** : Toutes les actions effectuées sur chaque ruche

Production

- **Rendement en miel** : Suivi de la production (15-25 kg par ruche/an en moyenne)
- **Gestion des récoltes** : Entre 0 et 2 récoltes par an (juin, mi-août, début septembre)

Fonctionnalités Avancées

- **Actions groupées** : Gain de temps majeur - possibilité de sélectionner plusieurs ruches pour enregistrer une action commune
- **Monitoring IoT** : Intégration de capteurs (balances, température interne)
- **Anti-vol GPS** : Traceurs GPS avec alerte en cas de déplacement non autorisé
- **Tableau de bord "Elite"** : Classement des meilleures ruches selon productivité et résistance

Découverte du Métier d'Apiculteur

Principe Général

Pas de routine : Chaque saison est différente, le travail varie selon les périodes de l'année.

Cycle Annuel

Hiver (Surveillance Passive)

- Les abeilles ne font rien, meurent petit à petit
- La reine ne pond pas
- Les abeilles se mettent en grappe pour survivre
- **Actions de l'apiculteur :**
 - Nourrir avec des suppléments liquides protéinés (sirop différent de l'été)
 - Traiter contre le Varroa (parasite principal) : 2 traitements (Apivar et acide oxalique selon la période)

Fin d'Hiver (environ 15°C)

- **Vérifications :**
 - Survie de la colonie
 - Si morte : désinfecter la ruche
 - Si vivante : vérifier l'âge de la reine, la population, la santé
 - État des réserves de nourriture

Printemps et Été (Gestion Active - Mars à Juin)

- **Interventions tous les 7 jours** pour :
 - Gestion de l'essaimage : détruire les cellules royales ou laisser faire
 - Nourrissement
 - Surveillance du semage (ponte)
- **Deux options lors de l'essaimage :**
 - Soit la reine part avec la moitié de l'essaim
 - Soit elles tuent la reine et en choisissent une nouvelle

Avril à Fin de Saison

- **Transhumance** : Déplacement des cheptels d'un point GPS à l'autre pour changer de fleurs et obtenir des miels spécifiques (Lavande, Sapin, Acacia, etc.)

Récolte

- **Périodes** : Juin, mi-août, début septembre
- **Fréquence** : Entre 0 et 2 récoltes par an
- **Rendement** : 15-25 kg de miel par ruche/an (jusqu'à 80 kg pour les meilleures)

Démographie des Abeilles

- **Ouvrière** : Vit environ 3 semaines en été (plusieurs mois en hiver)
- **Reine** : Vit 4-5 ans (souvent remplacée après 2 ans si la ponte baisse)
- **Identification de l'âge de la reine** : Code couleur marqué par l'apiculteur (5 couleurs selon l'année de naissance - code international)

Structure d'une Ruche

Le Corps de Ruche

- Base de la ruche
- Contient les larves, la reine, les ouvrières

- Contient le miel pour la nourriture de la colonie
- **On ne récolte jamais le miel du corps**
- Grille plus petite pour empêcher la reine de passer dans les hausses

Les Hausses

- Cadres de stockage posés sur le corps
- C'est le surplus destiné à la récolte
- Environ 15 kg de miel par hausse
- Une ruche performante peut avoir jusqu'à 4 hausses

Types de Ruches

- **Dadant** (le plus courant)
- **Langstroth**
- **Voirnot**
- **Warré**
- **Kenyane**
- Format et endroit changent selon le type

Capacité

- Une ruche : environ 15-25 kg de miel par an
 - Les meilleures : jusqu'à 80 kg par cheptel
-

Santé et Maladies

Le Varroa (Parasite Principal)

- Parasite qui s'attaque aux larves
- **Traitements** : Apivar et acide oxalique selon la période
- Respect des temps d'attente avant la récolte

Autres Maladies

- Différents types de maladies nécessitant différents traitements
- Parfois, seule solution : détruire et brûler la ruche
- Existence de vaccins dans certains cas

Gestion Sanitaire

- Une ruche malade n'implique pas la contamination systématique du rucher
- Importance de la sélection génétique pour identifier les colonies résistantes

Organisation des Ruchers

Cheptel

- Regroupement de 20 à 50 ruches
- Permet une gestion optimisée des interventions
- Pas d'intervention pour une seule ruche en particulier

Immatriculation

- Chaque ruche a un numéro d'immatriculation unique (obligatoire pour les professionnels)
- Permet le suivi individuel et la traçabilité

Races d'Abeilles

- Différentes races d'abeilles selon les régions et objectifs
 - Buckfast, Noire, etc.
-

Besoins Spécifiques Identifiés

1. **Visualisation des cheptels** pour accéder facilement aux ruches
 2. **Suivi de la transhumance** : où sont placés les cheptels
 3. **Traçabilité de la généalogie** des essaims
 4. **Identification des ruches performantes** : celles qui meurent et celles qui survivent, pour gérer la duplication
 5. **Cahier de miellerie numérique** : remplacement du suivi papier actuel
 6. **Actions groupées** : gain de temps majeur pour les interventions sur plusieurs ruches
 7. **Aide à la décision** : suggestions de ruches à diviser au printemps pour multiplier le cheptel
-

Entités Principales de l'Application

Rucher

- Emplacement GPS
- Type de flore

- Altitude
- Nombre de ruches

Ruche

- Numéro d'immatriculation unique
- Type (Dadant, Warré, Langstroth, etc.)
- Race d'abeille (Buckfast, Noire, etc.)
- Statut (Active, Malade, Morte)
- Historique des interventions

Reine

- Année de naissance (code couleur international)
- Généalogie
- Note de douceur
- Historique de ponte

Intervention

- Type (Visite, Nourrissement, Traitement, Récolte)
- Date
- Observations
- Ruches concernées

Cheptel

- Nom/Identifiant
 - Localisation GPS
 - Liste des ruches
 - Historique des transhumances
-

Fonctionnalités IoT et Sécurité

Monitoring

- **Capteurs de poids** : Balances pour suivre l'évolution de la production
- **Capteurs de température** : Température interne pour la santé du couvain
- **Données en temps réel** : Alertes en cas d'anomalie

Anti-vol

- **Traceurs GPS discrets** à l'intérieur des corps de ruche

- **Accéléromètre** : Détection de déplacement non autorisé
 - **Alertes immédiates** en cas de mouvement suspect
 - **Justification** : Une ruche complète vaut environ 400€
-

Système de Classement "Elite"

Critères de Sélection

1. **Productivité** : Nombre de hausses récoltées
2. **Résistance** :
 - Faible taux de mortalité
 - Bonne santé hivernale
 - Résistance naturelle aux maladies

Utilisation

- Identification des meilleures colonies pour l'élevage
 - Suggestions de ruches à diviser au printemps
 - Amélioration génétique du cheptel
 - Marquage des essaims "Non reproductibles" si la reine meurt prématurément (moins de 2 ans)
-

Contraintes et Enjeux

Contraintes Métier

- Pas de routine : adaptation constante aux saisons
- Interventions fréquentes au printemps (tous les 7 jours)
- Gestion complexe de la généalogie
- Traçabilité obligatoire pour les professionnels

Enjeux Économiques

- Valeur des ruches : 150€ la ruche + 150€ l'essaim
- Risque de vol important
- Production variable selon les années
- Colonies d'abeilles se vendent

Enjeux Sanitaires

- Varroa : menace principale

- Maladies pouvant nécessiter la destruction de ruches
- Importance de la prévention
- Sélection génétique pour améliorer la résistance

Enjeux Environnementaux

- Changement climatique impactant les colonies
 - Importance de la transhumance pour la pollinisation
 - Adaptation aux floraisons
-

Valeur Ajoutée de l'Application

1. **Gain de temps** : Actions groupées, interface intuitive
2. **Meilleure traçabilité** : Historique complet, généalogie
3. **Aide à la décision** : Classement des ruches, suggestions d'actions
4. **Prévention** : Alertes, suivi sanitaire
5. **Sécurité** : Anti-vol GPS
6. **Optimisation** : Suivi de la production, sélection génétique
7. **Professionnalisation** : Cahier de miellerie numérique conforme

CdC

Cahier des charges

Application de gestion et de suivi de ruchers

1. Présentation générale du projet

1.1 Contexte

Le changement climatique, la pression sanitaire (notamment le Varroa) et les vols rendent la gestion apicole de plus en plus complexe. Les apiculteurs doivent aujourd’hui suivre précisément l’état sanitaire, la production, les déplacements et la généalogie de leurs colonies, tout en respectant des obligations de traçabilité encore largement gérées sur papier.

Le projet vise à **numériser et centraliser la gestion des ruchers** afin d'aider les apiculteurs à prévenir, décider et intervenir efficacement.

1.2 Objectif principal

Développer une application permettant :

- Le suivi sanitaire et productif des ruches,
- La traçabilité complète des interventions,
- remplacer le cahier de miellerie papier par un outil numérique conforme.
- La sécurisation des ruches contre le vol.
- Gagner du temps via des actions groupées.

1.3 Objectifs secondaires

- L'aide à la décision (sélection, division, traitements),
- Identifier les colonies performantes et résistantes.
- Faciliter la gestion de la transhumance.
- Intégrer des données issues de capteurs IoT.

1.4 Approche projet

Le développement se fera en **collaboration directe avec un apiculteur client**, garantissant :

- la pertinence métier,
- l'adéquation avec la réalité du terrain,
- une validation continue des fonctionnalités.

2. Équipe projet

- **Chef de projet** : Sayfoudinov Timour
 - **Développement Front-end** : Joly Dorian
 - **Développement Back-end** : Gesse Corentin
 - **DevOps / Infrastructure** : Ait Ouarab Mélissa
-

3. Périmètre fonctionnel

3.1 Gestion générale

- Interface simple et intuitive, utilisable sur le terrain.
 - Gestion par **cheptel** (groupes de 20 à 50 ruches). -> Gestion par rucher
 - Gestion par **ruche individuelle** avec immatriculation unique.
 - Géolocalisation des ruchers (GPS).
 - Historique des déplacements (transhumance).
 - Système d'alertes anti-vol.
-

3.2 Suivi sanitaire

- Suivi de l'état de santé des colonies.
 - Gestion des maladies et pathologies.
 - Traçabilité des traitements (Apivar, acide oxalique, etc.).
 - Respect des périodes de traitement et des délais avant récolte.
 - Statut sanitaire des ruches :
 - Active
 - Malade
 - Morte
 - Mise en quarantaine virtuelle des ruches malades.
 - Identification des colonies résistantes (sélection génétique).
-

3.3 Traçabilité et généalogie

- Historique complet des interventions par ruche.
- Suivi des reines :
 - année de naissance (code couleur international),
 - lignée génétique,
 - note de douceur,
 - historique de ponte.

- Généalogie des essaims.
 - Marquage des colonies non reproductibles (reine morte avant 2 ans).
-

3.4 Production et récoltes

- Suivi du rendement en miel par ruche et par cheptel.
 - Gestion des récoltes (0 à 2 par an).
 - Quantité par hausse (~15 kg).
 - Historique annuel de production.
 - Identification des ruches les plus productives.
-

3.5 Actions groupées

- Sélection multiple de ruches.
 - Enregistrement d'une intervention commune :
 - nourrissement,
 - traitement,
 - visite,
 - récolte.
 - Objectif : gain de temps massif en période de forte activité.
-

3.6 Tableau de bord “Elite”

- Classement automatique des ruches selon :
 - productivité,
 - survie hivernale,
 - résistance sanitaire.
 - Identification des colonies à :
 - conserver,
 - reproduire,
 - exclure de la reproduction.
 - Aide à la décision pour la division printanière.
-

3.7 Gestion des reines – élevage et exploitation mixte

3.7.1 Cas 1 – Apiculteurs éleveurs de reines uniquement (

- La gestion d'un **bloc de cupule de reines**
- Le suivi précis des **cycles d'élevage** :
 - greffage,
 - cellules royales,
 - fécondation,
 - mise en ruchette,
 - reine vendue / perdue / non fécondée.
- L'identification unique de chaque reine avec :
 - année de naissance (code couleur international),
 - lignée génétique,
 - ruche ou élevage,
 - statut (en élevage, fécondée, disponible à la vente, éliminée).
- La traçabilité complète exigée pour :
 - la sélection génétique,
 - la certification,
 - la vente de reines.
- L'historique sanitaire et comportemental :
 - douceur,
 - vitalité,
 - régularité de ponte.
- La gestion de lots de reines (par greffage ou série).

3.7.2 Cas 2 – Apiculteurs à activité mixte (miel + élevage de reines)

- Le **lien explicite entre reines et ruches de production**
- La gestion des reméragés :
 - remplacement volontaire,
 - perte de reine,
 - sélection interne.
- L'identification des ruches servant :
 - de souches reproductrices,
 - de ruches éleveuses,
 - de ruches de production non reproductibles.
- Le suivi de l'impact des reines sur :
 - la productivité,
 - la résistance sanitaire,
 - la survie hivernale.
- Le marquage automatique des colonies :
 - reproductibles,
 - non reproductibles (reine trop âgée, défauts, mortalité précoce).

4. Fonctionnalités IoT et sécurité

4.1 Monitoring IoT

- Capteurs de poids (balances).
- Capteurs de température interne.
- Données en temps réel.
- Alertes en cas d'anomalie (chute de poids, température anormale).

4.2 Anti-vol

- Traceurs GPS intégrés aux ruches.
 - Détection de mouvement (accéléromètre).
 - Alertes immédiates en cas de déplacement non autorisé.
 - Justification économique : une ruche complète ≈ 400 €.
-

5. Entités principales du système

5.1 Rucher

- Identifiant
- Localisation GPS
- Type de flore
- Altitude
- Nombre de ruches

5.2 Cheptel

- Nom / identifiant
- Localisation GPS
- Liste des ruches
- Historique des transhumances

5.3 Ruche

- Numéro d'immatriculation unique
- Type (Dadant, Warré, Langstroth, etc.)
- Race d'abeilles
- Statut sanitaire
- Historique des interventions

5.4 Reine

- Année de naissance (code couleur)
- Généalogie

- Note de douceur
- Historique de ponte

5.5 Intervention

- Type (visite, nourrissement, traitement, récolte)
 - Date
 - Observations
 - Ruches concernées
-

6. Contraintes

6.1 Contraintes métier

- Forte saisonnalité.
- Absence de routine.
- Interventions fréquentes au printemps (tous les 7 jours).
- Traçabilité obligatoire pour les professionnels.
- Gestion complexe de la généalogie.

6.2 Contraintes sanitaires

- Varroa comme menace principale.
- Maladies pouvant imposer la destruction de ruches.
- Respect strict des protocoles de traitement.

6.3 Contraintes économiques

- Valeur élevée du cheptel.
 - Risque de vol.
 - Production fortement variable selon les années.
-

7. Valeur ajoutée de la solution

- Gain de temps opérationnel.
- Centralisation et fiabilité des données.
- Aide à la décision concrète.
- Prévention sanitaire et sécuritaire.
- Optimisation de la production.
- Professionnalisation de la gestion apicole.
- Adaptation aux enjeux climatiques et environnementaux.

US

Épics

1. **Onboarding & Organisation** (ruchers, cheptels, ruches)
 2. **Suivi sanitaire & interventions**
 3. **Production & récoltes**
 4. **Traçabilité & généalogie**
 5. **Transhumance & géolocalisation**
 6. **Sécurité anti-vol**
 7. **Dashboard & aide à la décision (Elite)**
 8. **IoT monitoring**
 9. **Qualité produit (UX, export, offline, sécurité)**
-

EPIC 1 — Onboarding & Organisation

US-1.1 — Crée un rucher (MVP)

En tant qu' apiculteur

Je veux créer un rucher avec ses infos (GPS, flore, altitude)

Afin de localiser et organiser mes ruches.

Acceptation

- Je peux saisir : nom, coordonnées GPS (ou choisir sur carte), flore, altitude,, notes.
- Le rucher apparaît dans la liste et sur la carte.
- Je peux modifier/supprimer un rucher.

Annulé - US-1.2 — Crée un cheptel (MVP)

En tant qu' apiculteur

Je veux créer un cheptel (groupe de 20–50 ruches)

Afin de gérer mes interventions par lot.

Acceptation

- Je peux définir : nom/identifiant, rucher associé, notes.
- Je peux voir le nombre de ruches dans le cheptel.
- Je peux déplacer un cheptel d'un rucher à un autre (liée à la transhumance si activée).

US-1.3 — Ajouter une ruche avec immatriculation unique (MVP)

En tant qu' apiculteur pro
Je veux enregistrer une ruche avec une immatriculation unique
Afin de respecter la traçabilité et suivre chaque ruche précisément.

Acceptation

- L'immatriculation est obligatoire et unique (blocage si doublon).
- Je peux renseigner type de ruche, race, statut initial (Active).
- La ruche est rattachée à un cheptel.

US-1.4 — Voir la fiche détaillée d'une ruche (MVP)

En tant qu' apiculteur
Je veux ouvrir une fiche ruche
Afin de consulter son statut, sa reine, ses interventions et sa production.

Acceptation

- Affiche : immatriculation, type, race, statut, localisation, dernier contrôle, dernière intervention.
 - Onglets/sections : interventions, santé, production, reine/généalogie.
-

EPIC 2 — Suivi sanitaire & interventions

US-2.1 — Enregistrer une intervention sur une ruche (MVP)

En tant qu' apiculteur
Je veux enregistrer une intervention (visite, nourrissement, traitement, récolte)
Afin de garder un historique fiable.

Acceptation

- Champs : type, date, observations, ruches concernées.
- Visible immédiatement dans l'historique de la ruche.
- La date par défaut = aujourd'hui, modifiable.

US-2.2 — Faire une action groupée sur plusieurs ruches (MVP)

En tant qu' apiculteur
Je veux sélectionner plusieurs ruches et appliquer une intervention commune
Afin de gagner du temps sur le terrain.

Acceptation

- Sélection multi-ruches (filtre par cheptel/rucher/statut).
- Une intervention créée référence toutes les ruches sélectionnées.

- Je peux retirer/ajouter une ruche avant validation.

US-2.3 — Marquer une ruche “Malade” et isoler (MVP)

En tant qu’ apiculteur

Je veux passer une ruche en statut “Malade” (quarantaine virtuelle)

Afin de suivre les cas à part et éviter les oubli.

Acceptation

- Le statut “Malade” est visible partout (liste, fiche, carte).
- Les ruches “Malades” sont filtrables.
- Un motif/diagnostic peut être renseigné (optionnel).

US-2.4 — Enregistrer un traitement Varroa (MVP)

En tant qu’ apiculteur

Je veux enregistrer un traitement (Apivar, acide oxalique)

Afin de respecter le protocole et les délais avant récolte.

Acceptation

- Choix du produit + date + dosage/notes.
- Le traitement apparaît dans l'historique “traitements”.
- Une alerte “récolte déconseillée” peut s'afficher si traitement récent (règles paramétrables en V1).

US-2.5 — Déclarer une ruche “Morte” (MVP)

En tant qu’ apiculteur

Je veux marquer une ruche comme “Morte” et enregistrer la cause probable

Afin de tracer la mortalité et planifier la désinfection/remplacement.

Acceptation

- Passage à statut “Morte” + date + cause (liste + texte libre).
- La ruche “Morte” est exclue par défaut des actions groupées (paramétrable).
- Historique conserve tout.

EPIC 3 — Production & récoltes

US-3.1 — Enregistrer une récolte (MVP)

En tant qu' apiculteur
Je veux saisir une récolte avec quantité/nb de hausses
Afin de suivre la production.

Acceptation

- Champs : date, nb hausses, kg (ou estimation), notes.
- Rattachée à une ruche OU un groupe (action groupée).
- Total saison affiché.

US-3.2 — Voir la production annuelle par ruche et cheptel (V1)

En tant qu' apiculteur
Je veux voir la production annuelle par ruche et par cheptel
Afin de comparer les performances.

Acceptation

- Filtre par année.
 - Tri : meilleur rendement, pire rendement.
 - Export possible (V1/V2 selon priorités).
-

EPIC 4 — Traçabilité & généalogie

US-4.1 — Créer/éditer la fiche Reine (MVP)

En tant qu' apiculteur
Je veux enregistrer l'année de naissance de la reine (code couleur), lignée, douceur
Afin de suivre la qualité et l'âge des reines.

Acceptation

- Année obligatoire ou "inconnue".
- Code couleur affiché automatiquement selon année.
- Note de douceur (ex: 1–5) + texte.

US-4.2 — Relier un essaim à une lignée (V1)

En tant qu' apiculteur
Je veux enregistrer l'origine d'un essaim (ruche mère)
Afin de suivre la généalogie.

Acceptation

- Je peux sélectionner la ruche mère.
- La ruche fille affiche sa lignée.

- Historique de reproduction consultable.

US-4.3 — Marquer “Non reproductible” (V1)

En tant qu’ apiculteur

Je veux marquer une colonie “Non reproductible” si reine morte avant 2 ans

Afin de éviter de propager une mauvaise génétique.

Acceptation

- Flag visible sur fiche et liste.
 - Les suggestions de division ignorent ces colonies.
-

EPIC 5 — Transhumance & géolocalisation

US-5.1 — Visualiser ruchers et cheptels sur une carte (MVP)

En tant qu’ apiculteur

Je veux voir mes ruchers/cheptels sur une carte

Afin de savoir où sont mes colonies.

Acceptation

- Carte affiche points GPS + nom + nb ruches.
- Clic ouvre la fiche.
- Filtre par rucher/cheptel/statut.

US-5.2 — Enregistrer une transhumance (V1)

En tant qu’ apiculteur

Je veux enregistrer un déplacement de cheptel (d’un point GPS à un autre)

Afin de tracer les miels et les floraisons.

Acceptation

- Date + origine + destination + flore cible (optionnel).
 - Historique consultable par cheptel.
 - Localisation actuelle mise à jour.
-

EPIC 6 — Sécurité anti-vol

US-6.1 — Déclarer une ruche “sécurisée” avec traceur GPS (V1)

En tant qu' apiculteur

Je veux associer un traceur GPS à une ruche

Afin de activer les alertes.

Acceptation

- Champ identifiant traceur (unique).
- Statut “sécurisée” visible.
- Paramètres d’alerte configurables (V1/V2).

US-6.2 — Recevoir une alerte de déplacement non autorisé (V2)

En tant qu' apiculteur

Je veux recevoir une alerte si une ruche se déplace sans intervention planifiée

Afin de réagir vite en cas de vol.

Acceptation

- Notification push/email (selon canal).
 - Alert contient : ruche, dernière position connue, heure.
 - Un déplacement “autorisé” peut être déclaré.
-

EPIC 7 — Dashboard & Aide à la décision (“Elite”)

US-7.1 — Dashboard synthèse (MVP)

En tant qu' apiculteur

Je veux un tableau de bord avec l’état global (actives/malades/mortes + dernières interventions)

Afin de savoir où agir en priorité.

Acceptation

- Compteurs : Active/Malade/Morte.
- Liste “à traiter bientôt” (ex: pas de visite depuis X jours).
- Liste “alertes” (malades, mortalité récente).

US-7.2 — Classement Elite (V1)

En tant qu' apiculteur
Je veux un classement des meilleures ruches
Afin de sélectionner celles à reproduire.

Acceptation

- Score calculé (prod + survie + santé).
- Filtre par saison/année.
- Une ruche “Non reproductible” est exclue du top.

US-7.3 — Suggestions de division au printemps (V2)

En tant qu' apiculteur
Je veux des suggestions de ruches à diviser
Afin de multiplier le cheptel intelligemment.

Acceptation

- Basé sur règles simples (prod, survie, reine, santé).
 - Suggestions modifiables/validables.
 - Historique des divisions proposées.
-

EPIC 8 — IoT Monitoring

US-8.1 — Associer des capteurs (V2)

En tant qu' apiculteur
Je veux associer des capteurs (poids/température) à une ruche
Afin de suivre les indicateurs sans ouvrir.

Acceptation

- Ajout capteur avec identifiant unique.
- Historique de mesures consultable.
- Indication “capteur actif/inactif”.

US-8.2 — Alertes d'anomalies IoT (V2)

En tant qu' apiculteur
Je veux être alerté si poids/température sort des seuils
Afin de intervenir avant que la colonie décroche.

Acceptation

- Seuils configurables.
- Alerte déclenchée + horodatage + valeur.

- L'apiculteur peut "accuser réception".
-

EPIC 9 — Qualité produit (UX, export, offline, sécurité)

US-9.1 — Recherche & filtres rapides (MVP)

En tant qu' apiculteur

Je veux filtrer rapidement mes ruches (cheptel, statut, type, race)

Afin de trouver une ruche en 3 secondes, pas en 3 minutes.

Acceptation

- Barre de recherche immatriculation/nom.
- Filtres combinables.
- Résultats instantanés.

US-9.2 — Mode terrain (offline) (V2)

En tant qu' apiculteur

Je veux pouvoir consulter/saisir des interventions sans réseau

Afin de travailler en rucher isolé.

Acceptation

- Les saisies se stockent localement.
- Synchronisation automatique quand réseau dispo.
- Gestion de conflit basique (timestamp + priorité utilisateur).

US-9.3 — Export "cahier de miellerie" (V1/V2)

En tant qu' apiculteur pro

Je veux exporter un cahier de miellerie numérique

Afin de remplacer le papier et répondre aux obligations.

Acceptation

- Export PDF/CSV par période.
 - Contient ruches, traitements, récoltes, dates, observations.
 - Format lisible et imprimable.
-

EPIC 10 - Ajout de dernière minute.

US-10.1 — Gestion avancée des reines (élevage & exploitation mixte) (V2)

En tant qu' apiculteur

Je veux gérer précisément les reines, qu'elles soient destinées à l'élevage, à la vente ou à la production de miel

Afin d'assurer une traçabilité complète, une sélection génétique maîtrisée et une exploitation cohérente de mon cheptel.

Acceptation

- Une reine ne peut jamais perdre son historique.
- Tous les changements de statut sont horodatés.
- La traçabilité est exploitable sans retraitement manuel.
- Le système est utilisable aussi bien par :
 - un éleveur pur,
 - un apiculteur mixte,
 - un apiculteur évoluant d'un modèle à l'autre.

Stack Techno

Stack Technique - Projet Gestion de Ruchers

Architecture Globale

Architecture : Application web full-stack avec API GraphQL

Séparation : Front-end et Back-end découplés communiquant via GraphQL

Back-end

Framework & Langage

- **Langage :** Python 3.11+
- **Framework :** Django 5.x
- **API :** GraphQL avec Graphene-Django

Bibliothèques Principales

- Django 5.x (Framework web)
- Graphene-Django (Implémentation GraphQL pour Django)
- Django-filter (Filtrage des données)
- Django-cors-headers (Gestion CORS)
- Pillow (Gestion des images)
- Python-decouple (Variables d'environnement)

Base de Données

- **SGBD :** PostgreSQL 15+ (recommandé pour la production)
- **Alternative développement :** SQLite (pour tests locaux)
- **ORM :** Django ORM

Avantages de Django pour ce projet

- ORM puissant pour gérer les relations complexes (Rucher → Ruche → Reine → Interventions)
- Admin Django pour gestion rapide des données
- Système d'authentification intégré
- Migrations de base de données automatiques

- Écosystème mature et stable
-

Front-end

Framework & Langage

- **Framework** : Next.js 14+ (App Router)
- **Langage** : TypeScript
- **Runtime** : Node.js 18+

Bibliothèques Principales

- Next.js 14+ (Framework React)
- React 18+
- TypeScript (Typage statique)
- Apollo Client (Client GraphQL)
- TailwindCSS (Styling)
- Shadcn/ui ou MUI (Composants UI)
- React Hook Form (Gestion des formulaires)
- Zod (Validation des données)
- Recharts ou Chart.js (Graphiques et statistiques)

Avantages de Next.js pour ce projet

- SSR (Server-Side Rendering) pour de meilleures performances
 - Routing intégré et optimisé
 - Optimisation automatique des images
 - Support TypeScript natif
 - API Routes pour fonctionnalités supplémentaires si besoin
 - PWA possible pour mode hors-ligne
-

API GraphQL

Pourquoi GraphQL ?

Avantages pour ce projet :

- **Flexibilité des requêtes** : Le front peut demander exactement les données nécessaires
- **Réduction du sur-fetching** : Pas de données inutiles transférées
- **Une seule endpoint** : `/graphql` pour toutes les opérations

- **Typage fort** : Schéma auto-documenté
- **Requêtes imbriquées** : Parfait pour Rucher → Ruches → Reine → Interventions en une seule requête

Implémentation

Back-end (Django) :

```
# Graphene-Django
- Définition du schéma GraphQL
- Types (RucherType, RucheType, ReineType, InterventionType)
- Queries (lecture des données)
- Mutations (création, modification, suppression)
- Subscriptions (temps réel pour IoT - optionnel)
```

Front-end (Next.js) :

```
// Apollo Client
- Client GraphQL configuré
- Queries et Mutations typées avec TypeScript
- Cache automatique des données
- Gestion des états de chargement/erreur
```

Exemple de Schéma GraphQL

```
type Rucher {
  id: ID!
  nom: String!
  emplacementGPS: String!
  typeFlore: String
  altitude: Int
  ruches: [Ruche]!
  createdAt: DateTime!
}
```

```
type Ruche {
  id: ID!
  immatriculation: String!
  type: TypeRuche!
  raceAbeille: String
  statut: StatutRuche!
  rucher: Rucher!
  reine: Reine
  interventions: [Intervention]!
}
```

```

type Query {
  ruchers: [Rucher!]!
  rucher(id: ID!): Rucher
  ruches(rucherId: ID): [Ruche!]!
  interventions(rucherId: ID, dateDebut: Date, dateFin: Date): [Intervention!]!
}

type Mutation {
  createRucher(input: RucherInput!): Rucher!
  updateRuche(id: ID!, input: RucheInput!): Ruche!
  createIntervention(input: InterventionInput!): Intervention!
}

```

Structure des Projets

Back-end (Django)

```

backend/
  ├── manage.py
  └── config/
    ├── settings.py
    ├── urls.py
    └── wsgi.py
  └── apps/
    ├── ruchers/
    │   ├── models.py
    │   ├── schema.py (GraphQL)
    │   ├── mutations.py
    │   └── types.py
    ├── ruches/
    ├── reines/
    ├── interventions/
    └── users/
  └── requirements.txt
└── .env

```

Front-end (Next.js)

```

frontend/
  └── src/
    ├── app/
    │   ├── layout.tsx
    │   ├── page.tsx
    │   ├── ruchers/
    │   └── ruches/

```

```
|- interventions/
|- components/
  |- ui/ (composants réutilisables)
  |- forms/
  |- charts/
|- lib/
  |- apollo-client.ts
    |- graphql/
      |- queries/
      |- mutations/
|- types/
  |- graphql.ts (types générés)
|- package.json
|- tsconfig.json
|- .env.local
```

Communication Front ↔ Back

Next.js (Frontend)



Apollo Client



HTTP POST /graphql



Django (Backend)



Graphene-Django



Django ORM



PostgreSQL

Fonctionnalités IoT (Optionnel)

Capteurs & Temps Réel

- **WebSockets** : Django Channels + GraphQL Subscriptions
 - **MQTT** : Pour communication avec capteurs IoT
 - **Redis** : Message broker pour temps réel
-

Déploiement

Développement

- **Back-end :** `python manage.py runserver` (port 8000)
- **Front-end :** `npm run dev` (port 3000)
- **BDD :** PostgreSQL local ou Docker

Production (Suggestions)

- **Back-end :**
 - Heroku / Railway / Render
 - DigitalOcean / AWS EC2
 - Gunicorn + Nginx
- **Front-end :**
 - Vercel (recommandé pour Next.js)
 - Netlify
- **BDD :**
 - PostgreSQL managé (AWS RDS, Supabase, Neon)

Sécurité

Authentification

- **JWT (JSON Web Tokens)** pour l'authentification
- **Django-graphql-jwt** ou **Django REST framework + GraphQL**
- Tokens stockés en httpOnly cookies (front)

Autorisations

- Permissions Django par utilisateur
- Vérification des droits dans les resolvers GraphQL
- CORS configuré pour autoriser uniquement le front

Avantages de cette Stack

- ✓ **Django** : Robuste, sécurisé, admin intégré, ORM puissant
- ✓ **Next.js** : Performances optimales, SEO, TypeScript
- ✓ **GraphQL** : Flexibilité, typage, requêtes optimisées
- ✓ **PostgreSQL** : Fiable, relationnel, performant

- ✓ **TypeScript** : Moins d'erreurs, meilleure maintenabilité
 - ✓ **Écosystème mature** : Beaucoup de documentation et support
-

Points d'Attention

- ⚠ **Courbe d'apprentissage GraphQL** : Si l'équipe n'est pas familière
 - ⚠ **Complexité initiale** : Setup Apollo Client + Graphene
 - ⚠ **Mode hors-ligne** : Nécessite stratégie de cache (Apollo Cache + PWA)
 - ⚠ **N+1 queries** : Attention aux performances GraphQL (utiliser DataLoader)
-

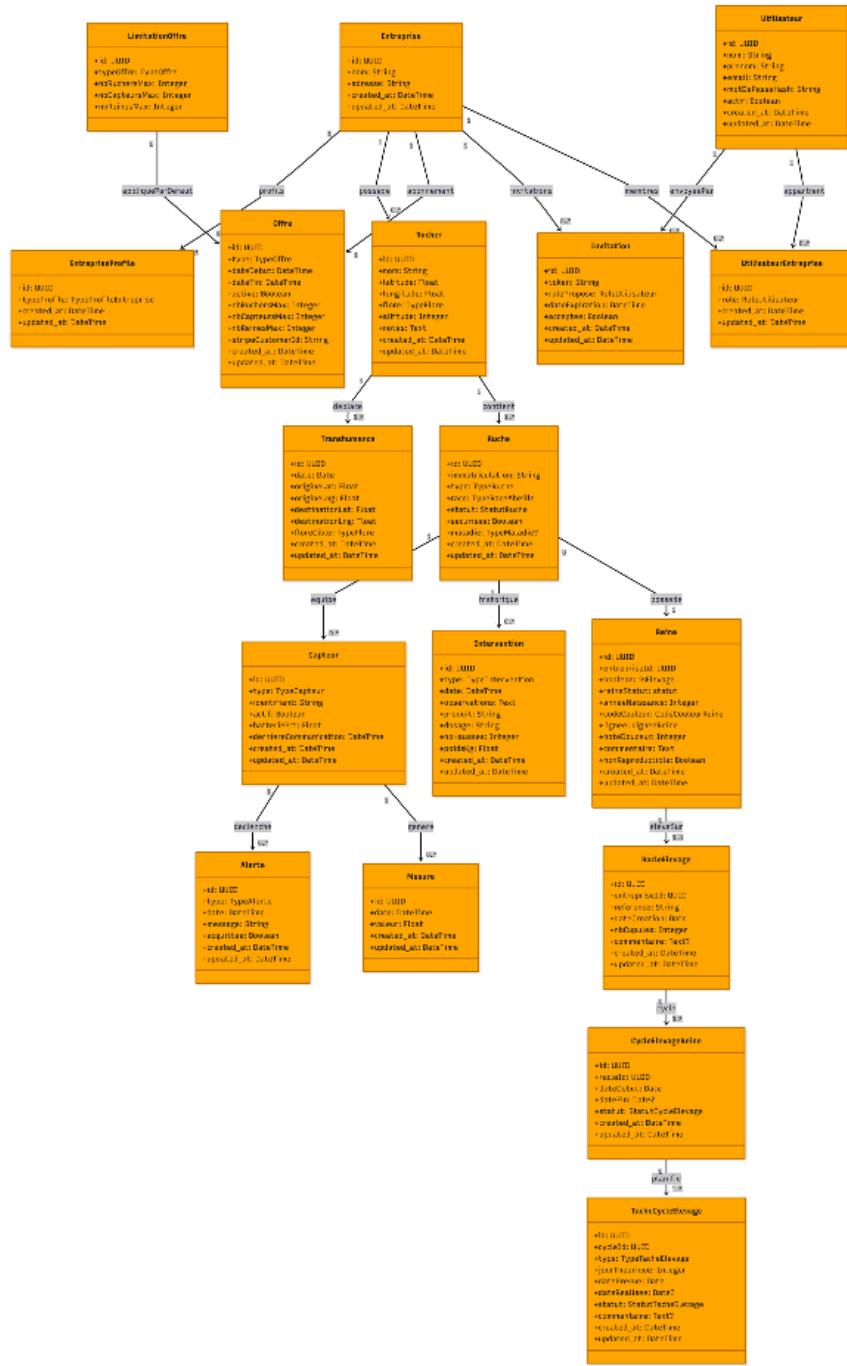
Conclusion

Cette stack **Django + Next.js + GraphQL + PostgreSQL** est parfaitement adaptée pour :

- Gérer des relations complexes (ruchers, ruches, reines, interventions)
- Offrir une interface utilisateur moderne et performante
- Permettre des évolutions futures (IoT, mobile, temps réel)
- Assurer la maintenabilité et la scalabilité du projet

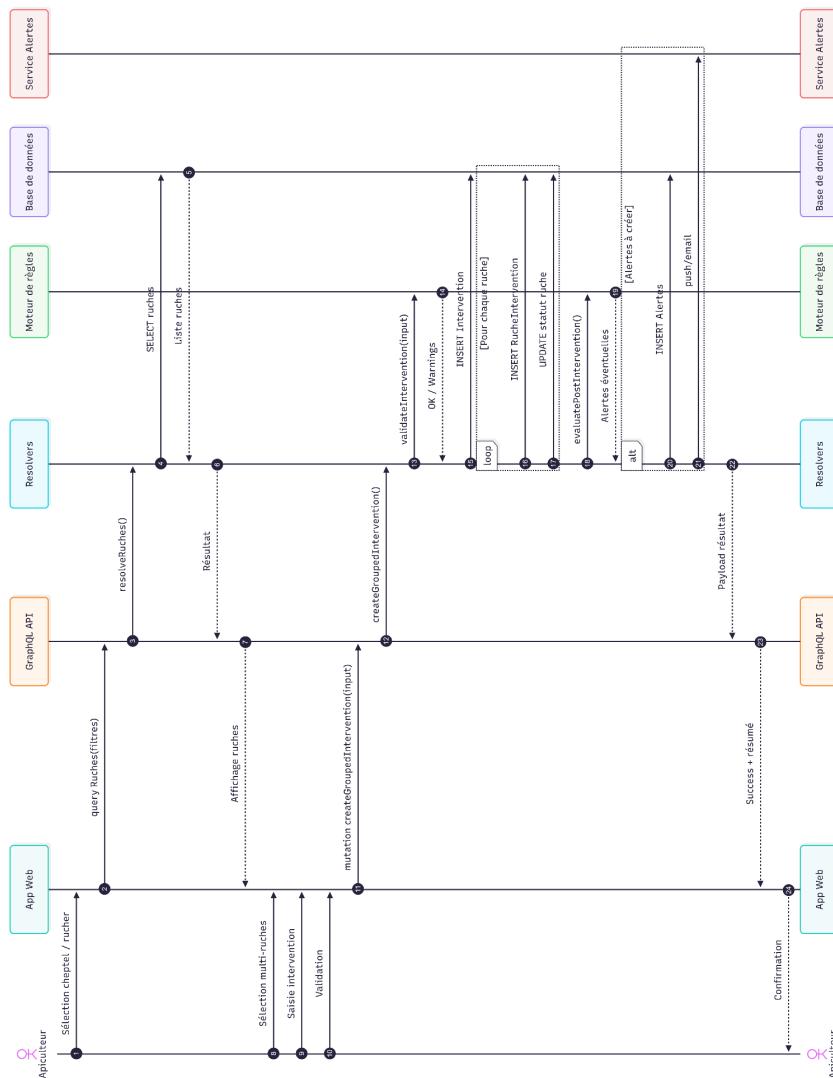
Schémas

Diagramme de classes



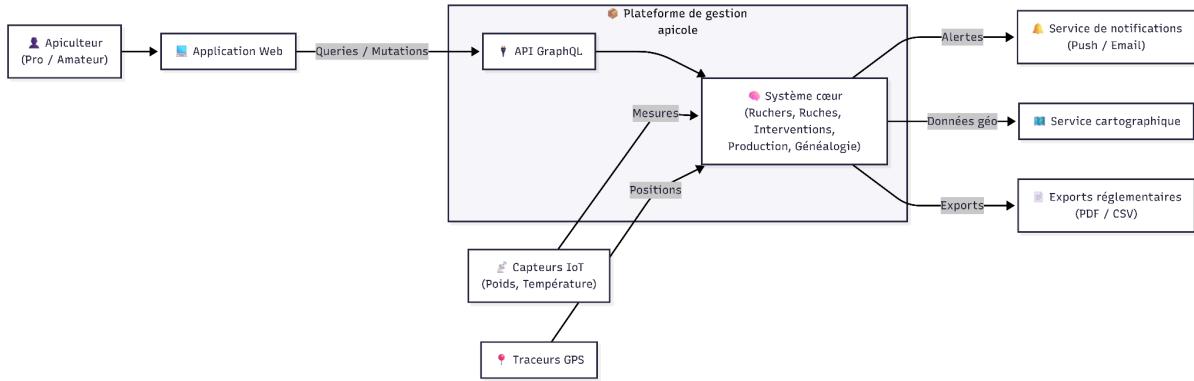
<https://mermaid.ai/d/6df72726-f11a-4a0f-bddb-1e282cf3fd3b>

Diagramme de séquence



<https://mermaid.ai/d/f2291c10-c32a-4435-ae34-3d0461708b3e>

Diagramme de contexte



<https://mermaid.ai/d/1d72474e-e790-4aee-a40a-012a9f05acf5>

IOT

passe par des reseaux a basse frequence

sigfox

mettre en place une gateway (boitier qui peux recevoir et emettre a une certaine distance avec une carte sim

donnees iot stocker en timeseries (bdd timeseries) timescaledb

simuler une gateway et les equipements iot broker mqtt

donnes envoyer sur le broker de message (sur les canaux)