



DIGICHEESE



Fromagerie DIGICHEES:

Concevoir et développer une application informatique

Amos DORCEUS

Idayat NOUFOU

Anna SYTIUK

Plan de la presentation

- Context
- Specification pour la nouvelle solution
- Architecture et Sécurité
- Scénarios de Tests et Résultats
- Conclusion et Perspectives
- Feedback sur le travail
- Références et Livrables

Contexte du Projet



Etat actuel

Application Interne

(sous Microsoft Access/VBA)

- Obsolète depuis >20 ans
- Instabilité (bugs)
- Maintenance difficile
- Faible possibilité de développement
- Manque d'accessibilité et fluidité



Objectif de la mission

Moderniser et optimiser la performance du système d'information de la Fromagerie DIGICHEES



Ambition

Une solution intranet moderne, fluide et sécurisée « DIGICHEES.com »



Valeur ajoutée

Amélioration de l'efficacité opérationnelle pour la gestion de la boutique physique et de la boutique en ligne.

Specification pour la nouvelle solution

- Authentification : page de connexion du site Internet, accessible à tous les utilisateurs
- L'application est centrée sur trois rôles utilisateurs spécifiques, pouvant être cumulés selon les besoins



Administrateur (Admin)

Garant de l'intégrité du système

Gere :

- les utilisateurs
- les paramètres métiers (poids, communes, objets)
- la configuration globale de la base de données.



Opérateur Colis (OP-colis)

Pivot de la relation client et des expéditions

Gere :

- les clients
- les commandes
- les calculs de conditionnement
- le mailing
- l'extraction des statistiques de vente



Opérateur Stock (OS-stock)

Responsable de la logistique interne

Assure :

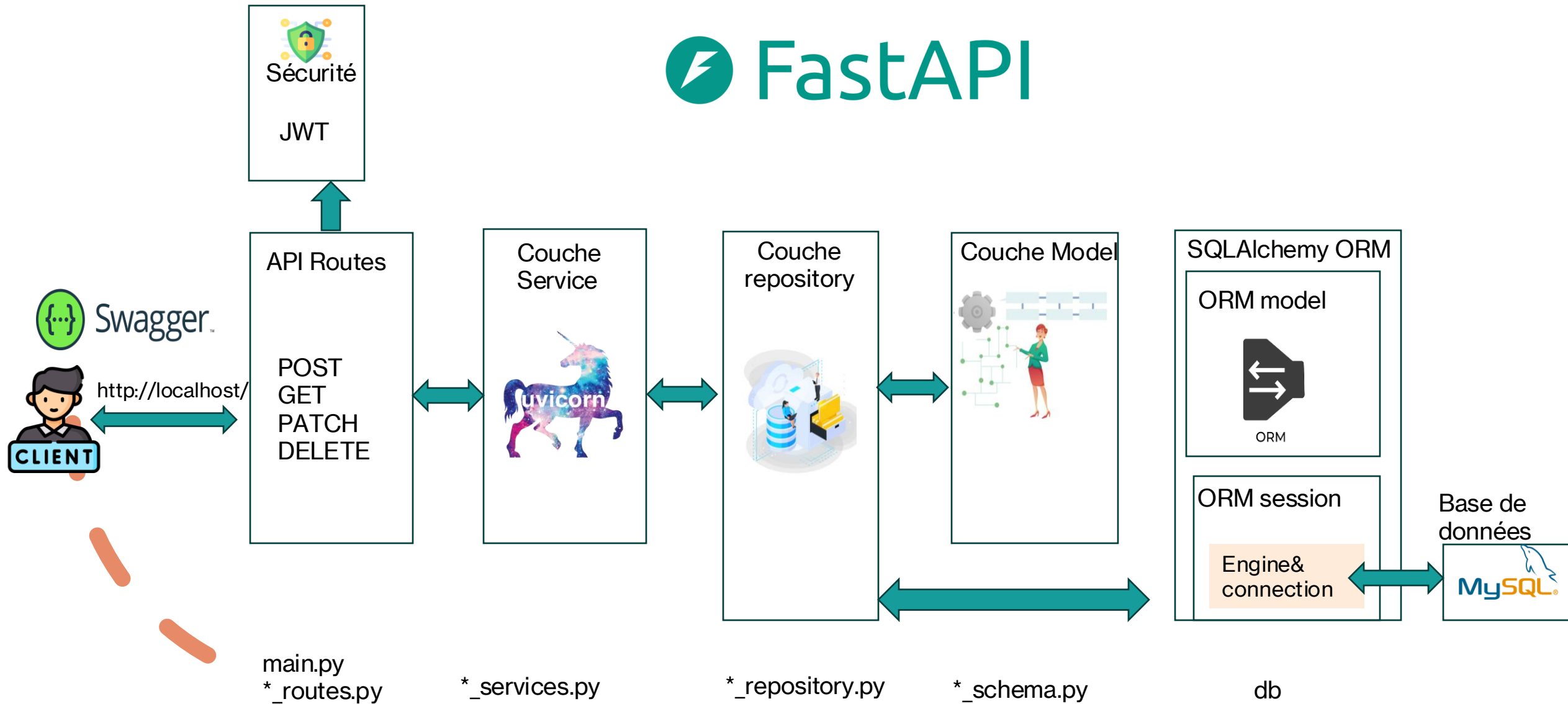
- la saisie de l'inventaire
- la mise à jour annuelle des stocks

Dispose d'une
vue globale sur l'état des marchandises.



Cette version de la solution se focalise sur la gestion de rôle Administrateur

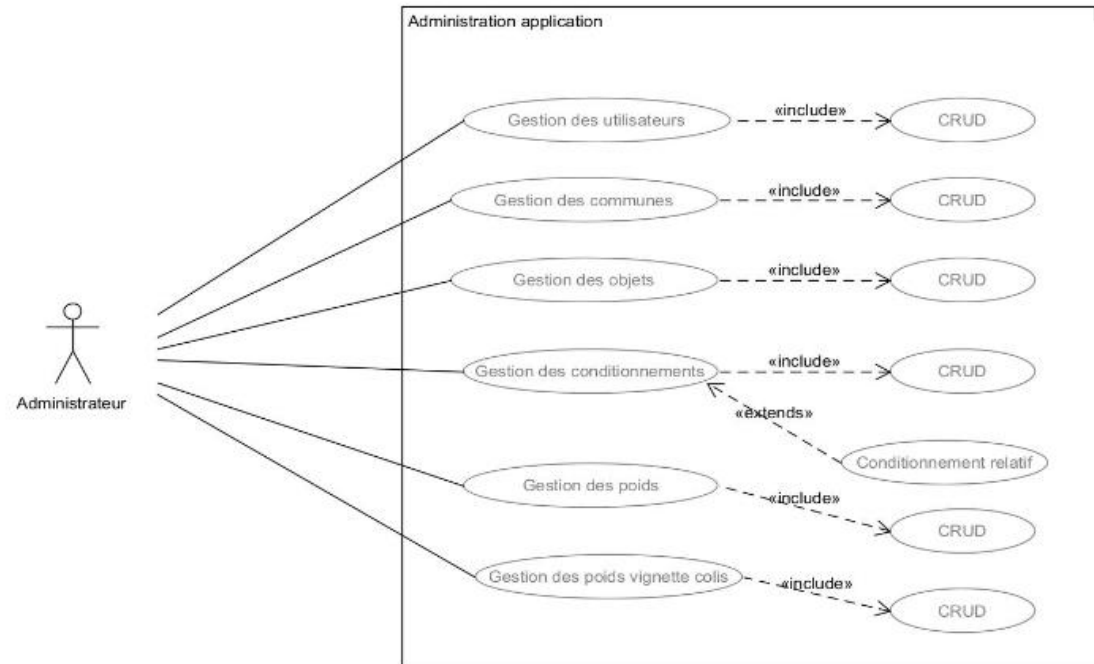
Architecture globale: API Backend + MySQL



CRUD Réalisés - Gestion Administrateur

- Commune
- Département
- Conditionnement
- Objet
- Utilisateur
- Role
- Vignette

3. USE CASE RÔLE ADMINISTRATEUR



Architecture et Sécurité - Standards OWASP

- Broken Access Control – Gestion des accès et authentification
- Cryptographic Failures – Protection des identifiants
- Security Misconfiguration – Configuration sécurisée de l'API
- Logging & Monitoring – Traçabilité et supervision
- Conception non sécurisée

https://owasp.org/Top10/2021/A02_2021-Cryptographic_Failures/

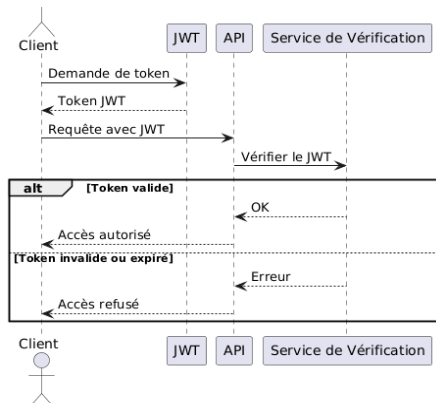
Architecture et Sécurité - Standards OWASP

Gestion des accès et authentification

Risque: Accès non autorisé aux ressources protégé

Mesures implémentées:

- Authentification basée sur JWT
- Vérification du token à chaque requête protégée
- Expiration automatique du token (TTL)
- Séparation des rôles et permissions

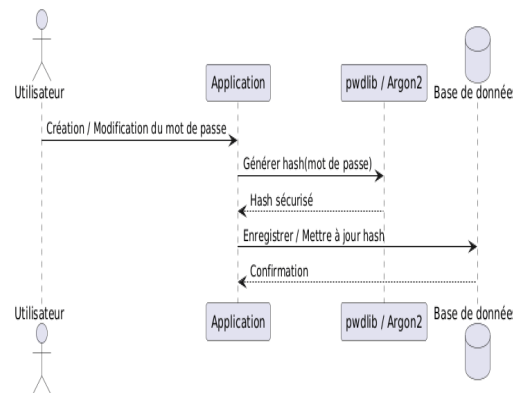


Protection (identifiants, Données)

Risque: Exposition de données sensibles comme les mots de passe en clair

Mesures implémentées:

- Aucun mot de passe stocké en clair
- Hashage sécurisé avec algorithme (pwdlib[argon2])
- Classes spécifiques pour lecture, écriture

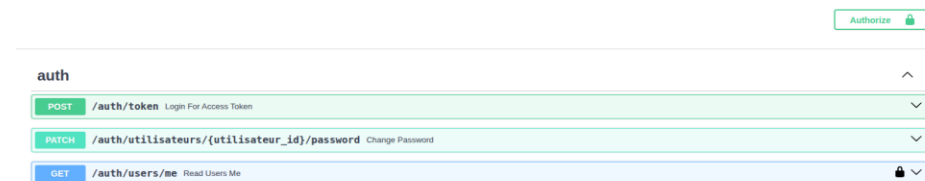


Security Misconfiguration

Risque: Mauvaise configuration exposant des fonctionnalités internes

Mesures implémentées:

- Séparation des environnements
- Variables d'environnement
- Désactivation de Swagger UI en environnement de production



Architecture et Sécurité - Standards OWASP

Traçabilité et supervision

Risque: Absence de détection des attaques et comportements suspect

Mesures implémentées:


- Journalisation des événements critiques
 - CRUD utilisateurs
 - Connexions réussies / échouées

Conception non sécurisée

Risque: Absence de détection des attaques et comportements suspect

Mesures implémentées:

- Architecture en couches
 - Routes -> Service -> Repositories
- JWT + rôles utilisateurs



```
2026-01-20 01:46:45,824 INFO security 82944 logging:52 {"action": "list", "user_id": "amosdorceus", "resource": "ut
2026-01-20 01:47:34,959 INFO security 82944 logging:52 {"action": "list", "user_id": "amosdorceus", "resource": "ut
2026-01-20 01:48:35,713 INFO security 83336 logging:52 {"action": "list", "user_id": "amosdorceus", "resource": "ut
2026-01-20 01:50:25,090 INFO security 83501 logging:52 {"action": "list", "user_id": "amosdorceus", "resource": "ut
2026-01-20 01:52:14,649 INFO security 83662 logging:52 {"action": "list", "user_id": "amosdorceus", "resource": "ut
2026-01-20 01:55:43,285 INFO security 83843 logging:52 {"action": "get", "user_id": "amosdorceus", "resource": "ut
2026-01-20 01:58:11,618 INFO security 83843 logging:52 {"action": "delete", "user_id": "amosdorceus", "resource": "
2026-01-20 13:21:12,697 INFO security 119598 logging:52 {"action": "create", "user_id": null, "resource": "utilisat
2026-01-20 13:23:36,401 INFO security 120012 logging:52 {"action": "get", "user_id": "amosdorceus", "resource": "ut
2026-01-20 13:44:27,535 INFO security 122016 logging:52 {"action": "create", "user_id": null, "resource": "utilisat
2026-01-20 13:48:49,684 INFO security 122652 logging:52 {"action": "create", "user_id": null, "resource": "utilisat
2026-01-20 14:16:20,938 INFO security 124872 logging:52 {"action": "create", "user_id": null, "resource": "utilisat
2026-01-20 14:21:21,967 INFO security 125706 logging:52 {"action": "create", "user_id": null, "resource": "utilisat
2026-01-21 03:03:43,037 INFO security 102703 logging:52 {"action": "list", "user_id": "admin", "resource": "utilisat
2026-01-21 03:41:54,052 INFO security 10538 logging:52 {"action": "list", "user_id": "admin", "resource": "utilisat
2026-01-21 04:36:49,074 INFO security 15664 logging:52 {"action": "get", "user_id": "admin", "resource": "utilisat
2026-01-21 11:03:44,977 INFO security 37857 logging:52 {"action": "get", "user_id": "admin", "resource": "utilisat
2026-01-21 11:07:59,632 INFO security 37857 logging:52 {"action": "list", "user_id": "admin", "resource": "utilisat
2026-01-21 11:09:22,551 INFO security 37857 logging:52 {"action": "create", "user_id": null, "resource": "utilisat
2026-01-21 11:09:58,384 INFO security 37857 logging:52 {"action": "list", "user_id": "amosdorceus", "resource": "ut
2026-01-21 11:10:32,894 INFO security 37857 logging:52 {"action": "list", "user_id": "admin", "resource": "utilisat
2026-01-21 11:11:18,391 INFO security 37857 logging:52 {"action": "update", "user_id": "admin", "resource": "utilisat
2026-01-21 11:11:36,080 INFO security 37857 logging:52 {"action": "get", "user_id": "admin", "resource": "utilisat
```

```
▼ DIGICHEES_APP
  > .venv
  ▼ logs
    ≡ security.log
  ▼ src
    > __pycache__
    > conf
    > core
    > models
    > repositories
    > routes
    > services
    > tests
    > utils
    • _init_.py
    • main.py
    • settings.py
  • .env.dev
  • .gitignore
  • README.md
  ≡ requirements.txt
  ≡ test.db
```

Qualité Logicielle et Qualité de Code

- **Normes de développement** : Utilisation de standards de codage pour assurer une maintenabilité à long terme.
- **Corrections de fiabilité** : Interprétation des résultats de tests unitaires/intégration et présentation des corrections appliquées pour stabiliser l'application.

Scénarios de Tests et Résultats

- **Fonctionnalités Critiques :** tests couvrant les points vitaux

Les tests portent sur les CRUD de l'utilisateur administrateur (Gestion des utilisateurs, des communes, des objets, des poids, des poids vignettes colis, des conditionnements).

- **Résultat des tests:**

```
platform win32 pytest 9.0.2 python 3.14.2
root C:\Users\Anna SYTIUK\Documents\GitHub\DIGICHEES_APP
plugins anyio-4.12.1, cov-7.0.0, html-4.2.0, metadata-3.1.1, rich-0.2.0, tldr-0.2.6
rootdir: C:\Users\Anna SYTIUK\Documents\GitHub\DIGICHEES_APP
plugins: anyio-4.12.1, cov-7.0.0, html-4.2.0, metadata-3.1.1, rich-0.2.0, tldr-0.2.6
```

collected 37 items

```
[100%] src/tests/test_conditionnement_route.py ✓✓✓✓✓
[100%] src/tests/test_objet_route.py ✓✓✓✓✓
[100%] src/tests/test_poids_route.py ✓✓✓✓✓
[100%] src/tests/test_role_route.py ✓✓✓✓✓
[100%] src/tests/test_utilisateur_route.py ✓✓✓✓✓
[100%] src/tests/test_conditionnement_route.py ✓✓✓✓✓
[100%] src/tests/test_objet_route.py ✓✓✓✓✓
[100%] src/tests/test_poids_route.py ✓✓✓✓✓
[100%] src/tests/test_role_route.py ✓✓✓✓✓
[100%] src/tests/test_utilisateur_route.py ✓✓✓✓✓
[100%] src/tests/test_vignette_route.py ✓✓✓✓✓
Percent: 100%
```

Summary

```
37 Total Tests
37 Passed (100.0%)
```

Fonctionnalité	Scénario testé	Résultat attendu
Création	Données valides	201 Created
Lecture	ID existant	200 OK
Modification	ID existant	200 OK
Suppression	ID existant	204 Deleted

Conclusion et Perspectives

- **Résumé des gains**

- **Modernisation** : Migration d'Access/VBA vers API Python/MySQL.
- **Fiabilité** : Élimination des bugs par une architecture robuste.
- **Sécurité** : Conformité aux normes OWASP et RGPD.
- **Évolutivité** : Structure prête pour le Web/Mobile via Swagger.

- **Perspectives d'Évolution**

- **Développement du Front-end** : Création d'une interface utilisateur moderne (React/Vue) utilisant l'API actuelle.
- **Monitoring Avancé** : Déploiement d'outils de surveillance des performances (KPIs) en temps réel
- **Extension Fonctionnelle** : Finalisation des modules optionnels pour les rôles "Opérateur Stock" et "Opérateur Colis".

Feedback sur le travail

- **Collaboration Agile** : Workflow Git optimisé (branches dev/test/prod) sans conflits.
- **Montée en compétences** : Maîtrise des environnements virtuels (VENV) et tests automatisés.
- **Efficacité Dev** : Standardisation du projet (requirements.txt) et documentation structurée.

**Merci pour votre
attention!**

Références

- [OWASP](#)