BTS SIO — Option SLAM

Procédure Technique

Déploiement et maintenance de l'application Papie

Candidat	Cyrille Cour
Etablissement	Aristée
Entreprise d'accueil	UPV (Union Patronale du Var)
Date	Juin 2026

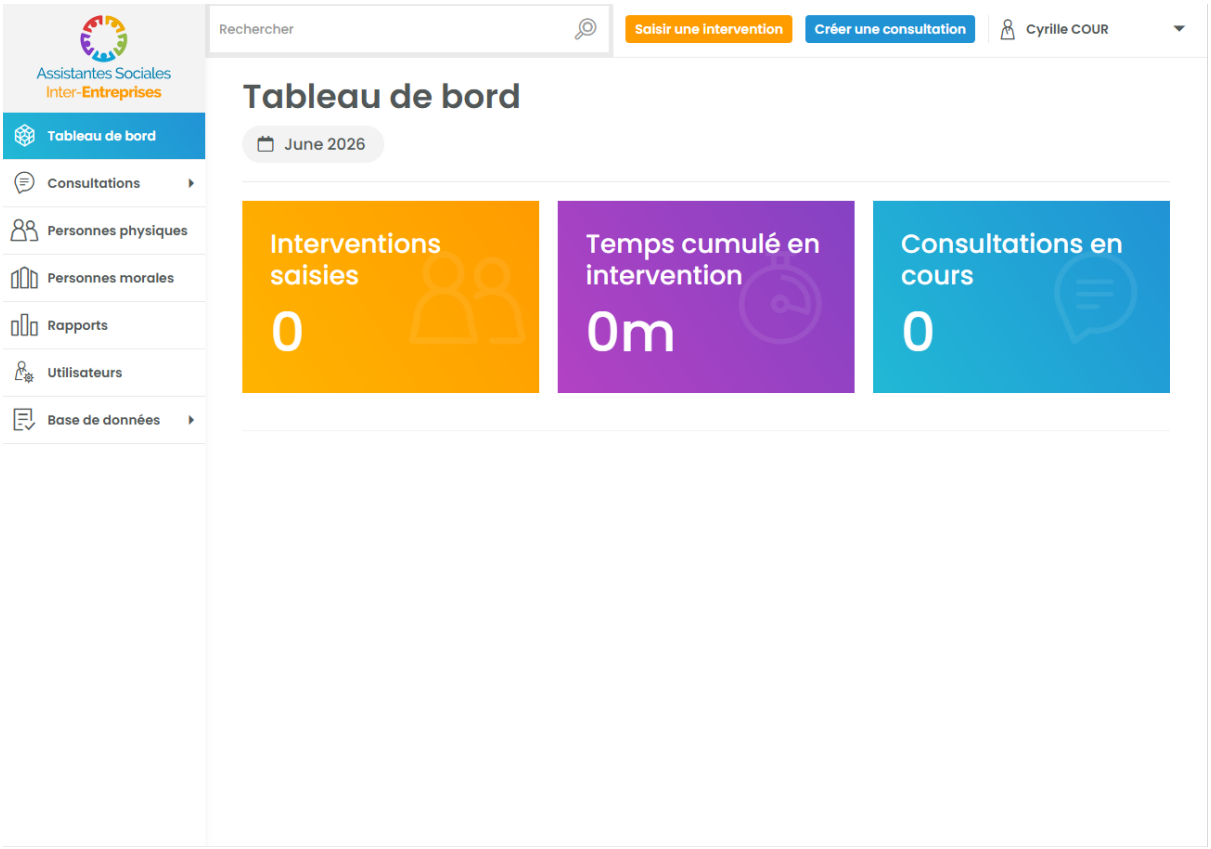
Sommaire

1. Contexte et présentation du projet

L'UPV (Union Patronale du Var) dispose d'un service social qui utilisait jusqu'ici l'application Metycea pour le suivi de ses adhérents (personnes physiques et entreprises). N'ayant pas accès aux fichiers sources de Metycea, il a été décidé de développer une application équivalente : Papie.

Papie est une application web développée sous Symfony 6 / PHP 8.1 qui permet :

- La gestion des entreprises adhérentes (Corporations)
- La gestion des personnes physiques liées à ces entreprises
- Le suivi des consultations et interventions du service social
- La synchronisation automatique avec Odoo (ERP de l'UPV) via Power Automate



2. Architecture technique

Composant	Détail
Framework	Symfony 6
Langage	PHP 8.1
Base de données	MariaDB 10.5
Administration BDD	PhpMyAdmin
Serveur web	Apache (Docker en préprod, Evolix/OVH en prod)
Automatisation	Microsoft Power Automate
ERP source	Odoo (JSON-RPC API)
Déploiement fichiers prod	WinSCP

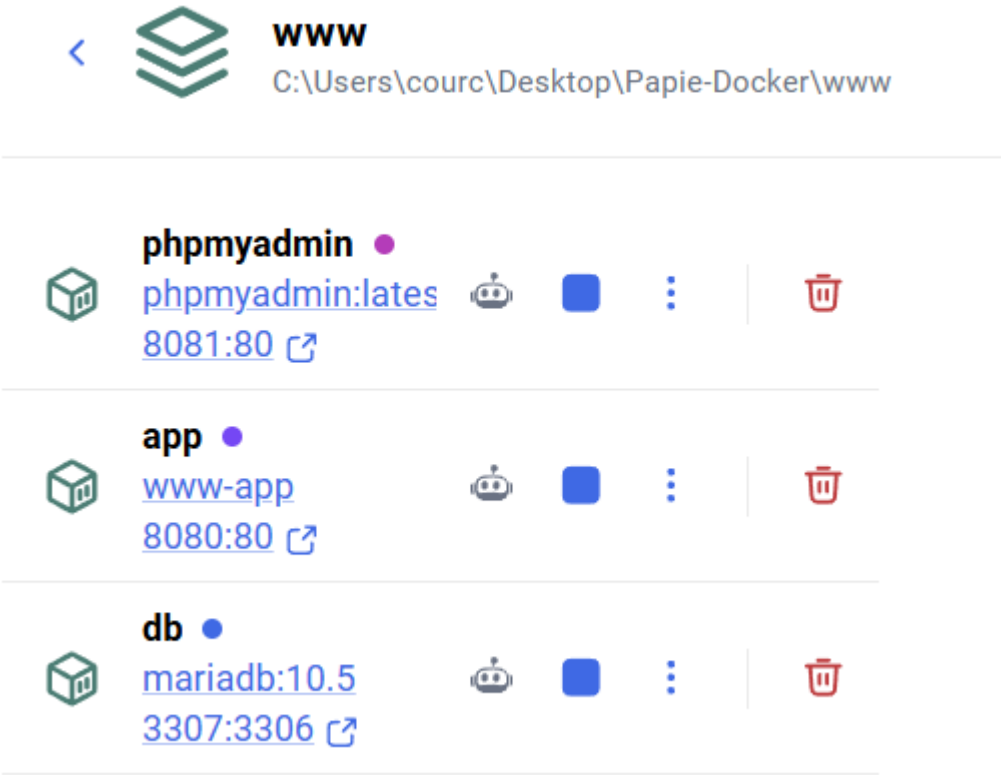
3. Mise en place des environnements

3.1 Environnement de préprod (Docker)

Avant toute intervention en production, un environnement de préprod identique a été mis en place en local via Docker. Cela permet de tester chaque modification sans risque pour les données de production.

Le fichier docker-compose.yml configure trois services :

- papie : conteneur PHP/Apache avec l'application Symfony
- papie_database : conteneur MariaDB 10.5
- papie_phpmyadmin : interface d'administration de la BDD



Commandes utilisées pour démarrer l'environnement :

```
# Démarrer les conteneurs
docker-compose up -d

# Entrer dans le conteneur PHP
docker exec -it papie bash

# Lancer les migrations Doctrine
php bin/console doctrine:migrations:migrate
```

3.2 Environnement de production (Evolix / OVH)

L'environnement de production est hébergé chez OVH avec une infrastructure gérée par Evolix. Le déploiement des fichiers se fait via WinSCP (transfert SFTP).

 Note : Toute modification est d'abord validée en préprod avant d'être appliquée en production. Cette règle a été respectée pour l'ensemble des interventions décrites dans ce document.

4. Récupération de la base de données

La base de données de Metycea a été récupérée et importée dans Papie à l'aide de l'outil BigDump. BigDump est un script PHP permettant d'importer des fichiers SQL volumineux en plusieurs passes, contournant les limitations de timeout des hébergeurs.

Procédure d'import :

- Dépôt du fichier bigdump.php et du dump SQL sur le serveur via WinSCP
- Accès à bigdump.php via le navigateur
- Lancement de l'import en plusieurs passes automatiques
- Vérification de l'intégrité des données dans PhpMyAdmin

5. Intégration de l'API Power Automate

L'UPV utilise Odoo comme ERP pour gérer ses entreprises adhérentes. Pour maintenir Papie synchronisé avec Odoo, un flux Power Automate a été mis en place pour synchroniser automatiquement chaque soir la liste des entreprises.

 Note : La procédure détaillée de création du flux Power Automate est disponible dans une procédure dédiée sur le portfolio.

5.1 Fonctionnement du flux Power Automate

Le flux se déclenche chaque soir et effectue deux appels HTTP :

- Appel 1 — Récupération des entreprises depuis Odoo via JSON-RPC :

```
POST https://[instance].odoo.com/web/dataset/call_kw
{
  "jsonrpc": "2.0",
  "method": "call",
  "params": {
    "service": "object",
    "method": "execute_kw",
    "args": ["instance", uid, "token",
      "res.partner", "search_read",
      [[["is_company", "=", true]]],
      { "fields": [...], "limit": 2000 }
    ]
  }
}
```

- Appel 2 — Envoi des données vers l'endpoint Papie :

```
POST https://[domaine]/api/sync/corporations
Headers:
  Content-Type: application/json
  X-Sync-Token: [token]

Body: { "result": @{body('HTTP')?['result']} }
```

5.2 Modifications des fichiers Symfony

Deux fichiers ont été créés ou modifiés pour accueillir l'API :

Corporation.php — Ajout des champs Odoo

L'entité Corporation a été enrichie de nouveaux champs pour stocker les données provenant d'Odoo :

- odoo_id : identifiant unique Odoo (clé de déduplication principale)
- last_synced_at : date/heure de la dernière synchronisation
- nom_commercial, forme_juridique, naf, secteur_activite
- effectif, gsc, territoire, assistante_sociale
- telephone, mobile, email, site_web
- est_adherent, est_membre, date_adhesion, reducbox, service_social

```php
    // ── Getters / Setters nouveaux champs Odoo ──────────────────────────

    public function setOdooId(?int $odooId): self
    {
        $this->odooId = $odooId;
        return $this;
    }

    public function getOdooId(): ?int
    {
        return $this->odooId;
    }

    public function setLastSyncedAt(?\DateTime $lastSyncedAt): self
    {
        $this->lastSyncedAt = $lastSyncedAt;
        return $this;
    }

    public function getLastSyncedAt(): ?\DateTime
    {
        return $this->lastSyncedAt;
    }

    public function setNomCommercial(?string $nomCommercial): self
    {
        $this->nomCommercial = $nomCommercial;
        return $this;
    }

    public function getNomCommercial(): ?string
    {
        return $this->nomCommercial;
    }

    public function setFormeJuridique(?string $formeJuridique): self
    {
        $this->formeJuridique = $formeJuridique;
        return $this;
    }
```

SyncController.php — Endpoint de synchronisation

Un nouveau contrôleur a été créé pour recevoir les données de Power Automate. La logique de matching des entreprises est la suivante :

Priorité	Condition	Action
1	odoo_id connu en BDD	PATCH — mise à jour directe
2	SIRET valide (non vide, non 000...)	Recherche par SIRET → PATCH du plus ancien
3	SIRET invalide	Recherche par nom (insensible à la casse) → PATCH du plus ancien
4	Aucun match	CREATE — création d'une nouvelle entrée

```php
// ── Mapping des champs ──────────────────────────────────
$corp->setOdooId($odooId);
$corp->setName($name ?? 'Sans nom');
$corp->setSiret($siret ?? '');
$corp->setNomCommercial($val($item['x_studio_nom_commercial'] ?? null));
$corp->setFormeJuridique($label($item['x_studio_forme_juridique'] ?? null));
$corp->setNaf($label($item['x_studio_naf'] ?? null));
$corp->setSecteurActivite($label($item['x_studio_secteur_dactivit'] ?? null));
$corp->setEffectif(isset($item['x_studio_effectif']) && $item['x_studio_effectif'] !== false ? (int)$item['x_studio_effectif'] : null);
$corp->setGsc(!empty($item['x_studio_gsc']));
$corp->setTerritoire($label($item['x_studio_territoire'] ?? null));
$corp->setAssistanteSociale($label($item['x_studio_assistante_sociale'] ?? null));
$corp->setTelephone($val($item['phone'] ?? null));
$corp->setMobile($val($item['mobile'] ?? null));
$corp->setEmail($val($item['email'] ?? null));
$corp->setSiteWeb($val($item['website'] ?? null));
$corp->setEstAdherent(!empty($item['x_studio_est_adherent']));
$corp->setEstMembre(!empty($item['x_studio_est_membre']));
$corp->setReducbox(!empty($item['x_studio_reducbox_active']));
$corp->setServiceSocial(!empty($item['x_studio_ss']));
```

5.3 Modifications de la base de données

Les nouvelles colonnes ont été ajoutées à la table corporation via une migration Doctrine (préprod) et via PhpMyAdmin en production :

```
-- Ajout de la colonne last_synced_at
ALTER TABLE corporation ADD COLUMN last_synced_at DATETIME DEFAULT NULL;

-- Les autres colonnes Odoo ont été ajoutées via migration Doctrine
php bin/console doctrine:migrations:migrate
```

6. Correction du widget Temps total interventions

Suite à une demande des utilisateurs, le cadre affichant le temps total des interventions sur l'écran d'accueil ne retournait pas les bonnes valeurs. Une intervention sur le fichier InterventionRepository a été nécessaire.

La correction a consisté à revoir la requête DQL (Doctrine Query Language) dans InterventionRepository.php pour agréger correctement les secondes d'intervention par période.

```php
public function sumTimeByUser(User $user, InclExcl $inclExcl = null): int
{
    $result = 0;
    try {
        $conn = $this->getEntityManager()->getConnection();    Undefined method 'getEntityManager'.
        $result = (int) $conn->executeQuery(
            'SELECT SUM(HOUR(time) * 3600 + MINUTE(time) * 60 + SECOND(time))
         FROM intervention
         WHERE user_id = :id',
            ['id' => $user->getId()]
        )->fetchOne();
    } catch (\Exception $e) {
    }
    return $result;
}
```

7. Audit et nettoyage de la base de données

7.1 Constat : doublons dans la table corporation

Après le premier lancement de l'API, un audit de la table corporation a révélé un nombre important de doublons. Les causes identifiées :

- SIRETs invalides (placeholders avec des 0000...) saisis par les utilisateurs
- Même entreprise saisie plusieurs fois avec des variantes de nom
- L'ancienne logique de l'API matchait uniquement par SIRET, créant des doublons en cas de SIRET incorrect

Requête d'audit initiale pour identifier les doublons :

```
SELECT name, siret, COUNT(DISTINCT id) as nb_corporations,
       COUNT(DISTINCT pc.id) as nb_persons,
       GROUP_CONCAT(DISTINCT id ORDER BY id) as ids
FROM corporation c
LEFT JOIN person_corporation pc ON pc.corporation_id = c.id
WHERE odoo_id IS NULL
GROUP BY name, siret
HAVING COUNT(DISTINCT id) > 1
ORDER BY nb_persons DESC;
```

7.2 Analyse de l'impact sur les données

Avant toute suppression, il était crucial d'analyser l'impact sur les entités liées. Le schéma de données est le suivant :

```
Corporation
    ^
    | PersonCorporation (table pivot Person <-> Corporation)
    ^
    Person
    ^
    PersonFlat (snapshot figé au moment de la consultation)
    ^
    Consultation
    ^
    Intervention(s)
```

Conclusions de l'analyse :

- PersonFlat stocke une copie des données Corporation au moment de la consultation → données historiques préservées même en cas de suppression
- PersonCorporation pointe directement vers Corporation (nullable=false) → risque de blocage FK en cas de suppression directe
- 7124 PersonCorporation pointaient vers des corporations potentiellement à supprimer → suppression directe impossible sans réattribution préalable

 Note: La stratégie choisie est la fusion (réattribution) et non la suppression directe, afin de préserver tous les liens entre personnes et entreprises.

7.3 Stratégie de nettoyage

La stratégie en 4 phases :

Phase	Action	Détail
1	Corriger l'API	Nouvelle logique odooId > SIRET > NOM pour éviter tout nouveau doublon
2	Lancer l'API	Peupler les odoo_id sur les corporations existantes (1917 updatées, 383 créées)
3	Fusionner	Réattribuer PersonCorporation et PersonFlat vers les corporations propres (avec odoo_id)
4	Supprimer	Supprimer les doublons orphelins (sans odoo_id correspondant à une PM Odoo)

7.4 Exécution du nettoyage

Résultats du premier lancement API

```
{
  "success": true,
  "created": 383,
  "updated": 1544,
  "total": 1927
}
```

Réattribution des PersonCorporation

```
-- 52 lignes réattribuées vers les corporations propres
UPDATE person_corporation pc
INNER JOIN corporation c_sale ON pc.corporation_id = c_sale.id
INNER JOIN corporation c_propre
    ON LOWER(c_propre.name) = LOWER(c_sale.name)
    AND c_propre.siret = c_sale.siret
    AND c_propre.odoo_id IS NOT NULL
SET pc.corporation_id = c_propre.id
WHERE c_sale.odoo_id IS NULL;
```

Réattribution des PersonFlat

```
-- 13/14 lignes réattribuées
UPDATE person_flat pf
INNER JOIN corporation c_sale ON pf.corporation_id = c_sale.id
INNER JOIN corporation c_propre
    ON LOWER(c_propre.name) = LOWER(c_sale.name)
    AND c_propre.siret = c_sale.siret
    AND c_propre.odoo_id IS NOT NULL
SET pf.corporation_id = c_propre.id
WHERE c_sale.odoo_id IS NULL;
```

Suppression des doublons orphelins

```
-- Vérification préalable : 0 ligne retournée = safe
SELECT pc.id FROM person_corporation pc
INNER JOIN corporation c ON pc.corporation_id = c.id
INNER JOIN corporation c_propre
    ON LOWER(c_propre.name) = LOWER(c.name)
    AND c_propre.odoo_id IS NOT NULL
WHERE c.odoo_id IS NULL;

-- Suppression : 17/19 lignes supprimées
DELETE FROM corporation WHERE odoo_id IS NULL
AND id IN (SELECT id_sale FROM (...) as tmp);
```

PAPIE

Résultat final

Métrique	Préprod	Production
Total corporations	4254	4256
Avec odoo_id (PMs Odoo)	1918	1917
Sans odoo_id (PMs locales)	2336	2339
Doublons odoo_id	0	0
Doublons supprimés	19	17
PersonCorporation réattribuées	54	52
PersonFlat réattribuées	14	13