

## Cahier des Charges Technique

Version 1.0 - Phase Cadrage

# Conception et implémentation d'une Pipeline d'extraction automatique de CV PDF → JSON

## Équipe Projet

LIMI Zakaria

NAHLI Ghita

OUTZOULA Abderrazzak

SALEHI Abderrahmane

SAADIOU Badreddine

# Table des matières

<b>1 Présentation globale du projet ATS</b>	<b>3</b>
1.1 Contexte général . . . . .	3
1.2 Objectifs d'un Applicant Tracking System . . . . .	3
1.3 Chaîne fonctionnelle globale de l'ATS . . . . .	4
1.4 Schéma macro du flux global . . . . .	4
1.5 Positionnement du sous-projet : pipeline PDF → JSON . . . . .	5
<b>2 Cahier des charges du sous-système : Pipeline PDF → JSON</b>	<b>6</b>
2.1 Contexte et vision . . . . .	6
2.1.1 Contexte métier . . . . .	6
2.1.2 Vision produit . . . . .	6
2.1.3 Environnement et écosystème SI . . . . .	7
2.2 Acteurs et Cas d'usage . . . . .	7
2.2.1 Acteurs du système . . . . .	7
2.2.2 Cas d'usage (Use Cases) . . . . .	8
2.2.3 Diagramme de cas d'usage (UML) . . . . .	9
2.2.4 User Stories associées . . . . .	9
2.3 Périmètre et Hors-périmètre . . . . .	9
2.3.1 Définition du périmètre . . . . .	9
2.3.2 Synthèse du périmètre projet . . . . .	10
2.3.3 Hypothèses de cadrage . . . . .	10
2.3.4 Contraintes projet . . . . .	10
2.4 Exigences fonctionnelles détaillées . . . . .	11
2.4.1 Vue globale du pipeline . . . . .	11
2.4.2 Module M1 : Ingestion et pré-traitement PDF . . . . .	11
2.4.3 Module M2 : OCR multilingue . . . . .	11
2.4.4 Module M3 : Layout Parsing . . . . .	11
2.4.5 Module M4 : Extraction sémantique (NLP) . . . . .	12
2.4.6 Module M5 : Normalisation et mapping JSON . . . . .	12
2.4.7 Module M6 : API REST et supervision . . . . .	13
2.5 Exigences non fonctionnelles . . . . .	13

2.5.1	Performance . . . . .	13
2.5.2	Fiabilité et disponibilité . . . . .	13
2.5.3	Sécurité et conformité RGPD . . . . .	13
2.5.4	Scalabilité et industrialisation . . . . .	13
2.5.5	Observabilité, logs et supervision . . . . .	13
2.5.6	Expérience utilisateur . . . . .	13
2.5.7	Synthèse des objectifs non fonctionnels . . . . .	14
2.6	Modèle de données JSON (vue macro) . . . . .	14
2.7	Flux de traitement et gestion des erreurs . . . . .	15
2.7.1	Flux nominal de traitement . . . . .	15
2.7.2	Gestion des erreurs et scénarios alternatifs . . . . .	16
2.7.3	Traçabilité et auditabilité . . . . .	17
2.8	Stratégie de tests et critères d'acceptation . . . . .	17
2.8.1	Stratégie de tests . . . . .	17
2.8.2	Matrice de couverture des tests . . . . .	17
2.8.3	Critères d'acceptation . . . . .	18
2.9	Planning prévisionnel et gouvernance . . . . .	18
2.9.1	Macro-planning (indicatif) . . . . .	18
2.9.2	Gouvernance projet . . . . .	18
2.10	Risques majeurs et plans d'atténuation . . . . .	18

# Chapitre 1

## Présentation globale du projet ATS

### 1.1 Contexte général

Dans un contexte de transformation digitale des fonctions Ressources Humaines, les entreprises font face à une augmentation continue du volume de candidatures, à une diversification des canaux de recrutement et à une complexité croissante dans la gestion des données candidats.

Les processus traditionnels, souvent fragmentés et fortement manuels, ne permettent plus :

- d'assurer un traitement rapide et homogène des candidatures ;
- de valoriser efficacement le vivier de talents existant ;
- de produire des indicateurs fiables pour le pilotage RH ;
- de garantir une expérience fluide pour les recruteurs et managers.

C'est dans ce contexte que s'inscrit le projet de développement d'un **ATS – Applicant Tracking System**, visant à structurer, automatiser et industrialiser l'ensemble du cycle de gestion des candidatures.

### 1.2 Objectifs d'un Applicant Tracking System

Un ATS est un système permettant de centraliser et d'orchestrer les activités liées au recrutement, depuis la réception des candidatures jusqu'à la décision et au reporting.

Les objectifs principaux sont :

- **Centraliser** toutes les candidatures entrantes, quelle que soit leur source ou format ;
- **Structurer** l'information issue des CV pour constituer une base exploitable (CV-thèque) ;
- **Accélérer** le tri et la création de short-lists via automatisation et critères métier ;
- **Faciliter** la collaboration RH / managers via workflow et suivi d'avancement ;

- **Piloter** la performance recrutement via indicateurs et reporting.

## 1.3 Chaîne fonctionnelle globale de l'ATS

Le fonctionnement global d'un ATS peut être décrit selon les étapes suivantes :

### E1 Centralisation des candidatures

Collecte des candidatures depuis différentes sources : job boards, candidatures spontanées, cooptations, événements campus, partenaires, etc. Les CV peuvent être multi-formats et hétérogènes.

### E2 Analyse et structuration des CV

Transformation des documents bruts en données exploitables, permettant la constitution d'une base globale interrogeable (CVthèque).

### E3 Job board et filtrage automatisé

Publication des offres sur les canaux sélectionnés, puis filtrage des candidatures pour générer une *short-list* cohérente avec les attentes du poste.

### E4 Gestion collaborative des évaluations

Mise en place d'un workflow de suivi et d'évaluation partagé : recruteurs et managers suivent l'avancement, échangent et tracent leurs décisions.

### E5 Analyse et reporting

Production d'indicateurs : efficacité des sources, délais de traitement, conversion par étape, qualité des candidatures, performance des campagnes.

## 1.4 Schéma macro du flux global

Le flux global peut être résumé par la logique suivante :

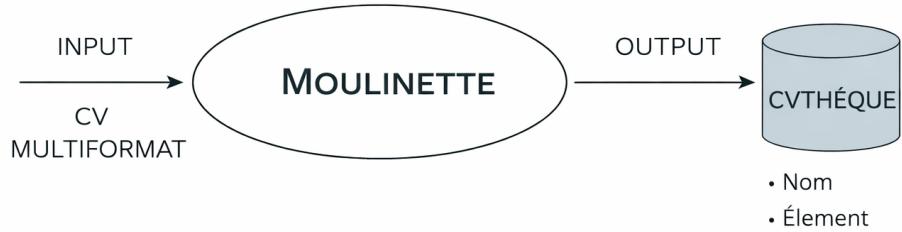


FIGURE 1.1 – Vue macro du pipeline de structuration des CV : entrée multiformat, traitement par la moulinette et alimentation de la CVthèque.

## 1.5 Positionnement du sous-projet : pipeline PDF → JSON

Au cœur de l’ATS, un composant critique consiste à transformer les CV bruts en données structurées et normalisées. Ce sous-projet correspond à la **brique “analyse et structuration des CV”** (la “moulinette”).

- **Entrée** : CV reçus sous forme de fichiers PDF (texte ou scans), formats variés, multilingues.
- **Sortie** : un **JSON structuré** conforme au schéma défini, directement intégrable dans une CVthèque.

Le présent cahier des charges formalise donc les exigences de conception, développement, tests et industrialisation de cette brique, tout en s’inscrivant dans la vision ATS globale.

# Chapitre 2

## Cahier des charges du sous-système : Pipeline PDF → JSON

### 2.1 Contexte et vision

#### 2.1.1 Contexte métier

Les processus actuels de gestion des candidatures au sein de Forvis Mazars impliquent une grande variété de formats de CV reçus depuis différents canaux : plateformes de recrutement, candidatures spontanées, cooptations, événements campus. Cette diversité engendre :

- une hétérogénéité des contenus (qualité variable, différentes langues : FR/EN/AR) ;
- une grande disparité des mises en page (multi-colonnes, tableaux, icônes) ;
- un traitement manuel chronophage et source d'erreurs de saisie ;
- une exploitation limitée des données pour la recherche de talents et le reporting.

Ces contraintes nuisent à la performance des équipes RH, ralentissent la création des short-lists et limitent la valorisation du vivier de candidats dans le système d'information Forvis.

#### 2.1.2 Vision produit

L'objectif du projet est de fournir un **pipeline automatisé de conversion de CV PDF vers un JSON structuré**, intégrable directement dans la CVthèque et les outils analytiques internes.

Ce pipeline doit permettre :

- d'augmenter le taux de structuration des données candidats ;
- d'accélérer l'intégration et le traitement des nouvelles candidatures ;

- de fiabiliser les informations clés (coordonnées, expérience, formation) ;
- d'améliorer la capacité d'analyse du vivier (indicateurs, reporting RH).

Le produit constitue un **socle technologique** réutilisable, extensible et industrialisable visant à automatiser et standardiser à terme l'ensemble du parsing documentaire RH.

### 2.1.3 Environnement et écosystème SI

La solution s'intègre dans le SI RH existant, en s'interfaçant principalement avec :

- le système ATS Forvis (*Applicant Tracking System*) ;
- les outils de suivi des candidatures et de reporting RH ;
- les sources externes de candidatures (jobboards, partenaires).

La figure ?? illustre la position du pipeline dans l'écosystème global.

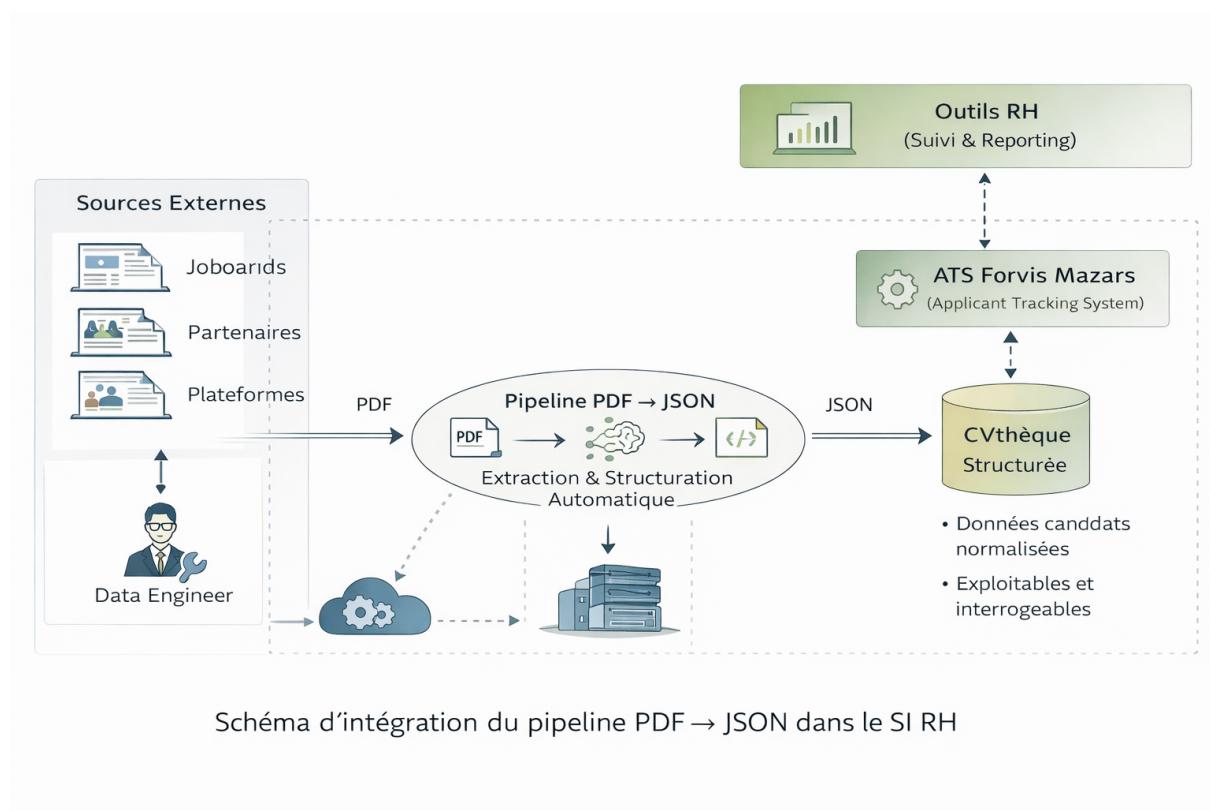


FIGURE 2.1 – Positionnement du pipeline PDF → JSON dans le SI RH.

## 2.2 Acteurs et Cas d'usage

### 2.2.1 Acteurs du système

Les acteurs impliqués dans l'utilisation ou la gestion de la solution sont décrits dans le tableau suivant :

Acteur	Rôle et Interaction avec le système
Recruteur / Talent Acquisition	Soumet les CV au système (via interface interne Forvis), consomme les données enrichies pour prise de décision.
Administrateur SI RH	Gère l'intégration technique avec l'ATS Forvis, surveille l'état des traitements, gère les droits d'accès et les quotas.
Data Engineer / Équipe IT	Déploie, maintient, supervise la solution et son observabilité, analyse les erreurs et améliore les modèles.
Pipeline d'extraction (Système)	Automatise le traitement des CV, orchestrant OCR, segmentation, extraction et normalisation JSON.
Sources externes (Job-boards, partenaires)	Fournissent des CV dans des formats variés à traiter par l'API.

TABLE 2.1 – Tableau des acteurs du système et leurs responsabilités.

### 2.2.2 Cas d'usage (Use Cases)

- **CU1 : Soumettre un CV et obtenir un JSON structuré**

Le recruteur fournit un fichier PDF et reçoit un objet JSON compatible Forvis.

- **CU2 : Traitement de CV en batch**

L'administrateur envoie un ensemble de CV pour intégration massive dans la CV-thèque (campagnes de recrutement).

- **CU3 : Consultation de l'état d'un traitement**

L'administrateur récupère le statut d'un job : succès, échec, logs et métadonnées processing.

- **CU4 : Rejet et diagnostic d'un parsing**

Le data engineer rejoue une extraction pour debug (problème OCR ou NLP).

### 2.2.3 Diagramme de cas d'usage (UML)

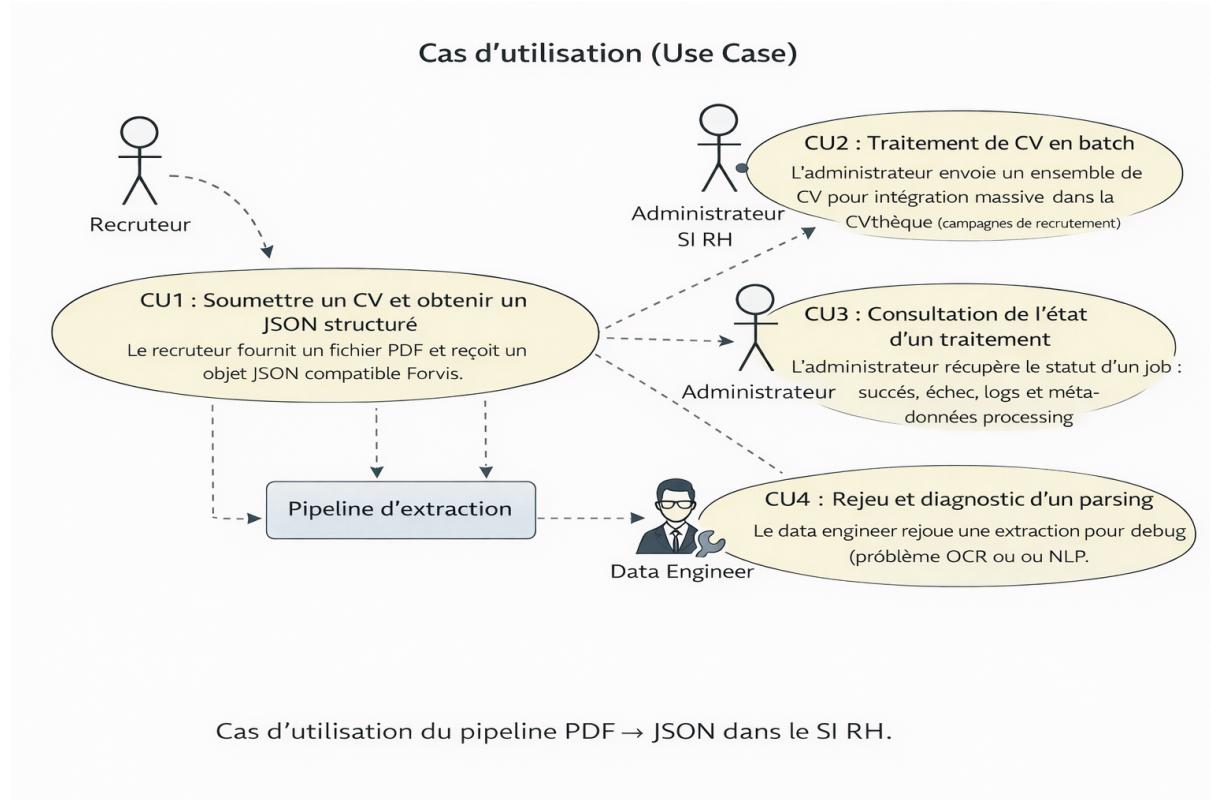


FIGURE 2.2 – UML

### 2.2.4 User Stories associées

- **US1** : En tant que *recruteur*, je souhaite soumettre un CV afin de créer automatiquement une fiche candidat pré-remplie.
- **US2** : En tant qu'*administrateur*, je veux suivre l'état des traitements pour garantir la qualité opérationnelle.
- **US3** : En tant que *data engineer*, je veux rejouer une extraction avec logs détaillés afin de diagnostiquer la cause d'une erreur de parsing.
- **US4** : En tant que *système ATS*, je veux recevoir un JSON standardisé pour alimenter automatiquement ma base candidats.

## 2.3 Périmètre et Hors-périmètre

### 2.3.1 Définition du périmètre

Le projet couvre l'ensemble des activités nécessaires à la mise en place d'un pipeline automatisé permettant de convertir les CV au format PDF en un objet JSON standar-

disé conforme aux exigences de Forvis. Ce périmètre intègre les étapes de conception, développement, tests, documentation et préparation à l'industrialisation.

### 2.3.2 Synthèse du périmètre projet

Inclus dans le périmètre	Exclus du périmètre
Développement du pipeline d'extraction PDF → JSON (OCR + Layout + NLP + Mapping)	Développement d'une interface graphique finale destinée aux recruteurs
Support multilingue : FR, EN, AR	Gestion avancée de l'écriture manuscrite (arabe stylisé, notes manuscrites)
Prise en charge des formats PDF textes et scannés	Parsing de CV aux formats non standards (images réseaux sociaux, CV vidéo)
Déploiement d'une API REST + logs + suivi des traitements	Refonte ou modification du SI RH Forvis
Création d'un corpus annoté pour entraînement et validation	Moteurs de matching candidat / poste ou scoring d'éligibilité
Documentation technique et support à l'intégration	Exploitation analytique avancée (BI, dashboards)
Tests fonctionnels, techniques et de performance	Intégration directe emailing / workflows externes

TABLE 2.2 – Périmètre fonctionnel et exclusions du projet.

### 2.3.3 Hypothèses de cadrage

- Les documents fournis sont exclusivement des CV en PDF.
- Le schéma JSON final est validé conjointement avec Forvis en amont des développements.
- L'accès au SI RH et aux environnements de tests sera assuré par Forvis.
- La performance cible est basée sur un volume standard ( $CV \leq 5$  pages).

### 2.3.4 Contraintes projet

- Conformité RGPD et politique DLP (Data Loss Prevention).
- Respect du planning et des jalons de validation RH.
- Sécurité et traçabilité des traitements (auditabilité interne).
- Architecture adaptable aux évolutions futures (scalabilité).

## 2.4 Exigences fonctionnelles détaillées

### 2.4.1 Vue globale du pipeline

Le pipeline se décompose en modules :

**M1 Ingestion et pré-traitement PDF**

**M2 OCR multilingue**

**M3 Layout Parsing**

**M4 Extraction sémantique (NLP)**

**M5 Normalisation & mapping JSON**

**M6 API REST & supervision**

### 2.4.2 Module M1 : Ingestion et pré-traitement PDF

- EF1.1 Le système doit accepter en entrée des fichiers PDF de taille maximale configurable (par défaut 10 Mo).
- EF1.2 Le système doit détecter automatiquement si le PDF contient du texte ou uniquement des images.
- EF1.3 Le système doit, si nécessaire, générer des images page par page pour alimenter l'OCR.
- EF1.4 Le système doit réaliser des pré-traitements d'image configurables : binarisation, redressement, réduction du bruit, amélioration du contraste.
- EF1.5 Le système doit conserver un identifiant unique de traitement afin de tracer chaque CV de bout en bout.

### 2.4.3 Module M2 : OCR multilingue

- EF2.1 Le système doit réaliser une reconnaissance optique de caractères pour les langues FR, EN et AR.
- EF2.2 Le système doit être capable de détecter la langue dominante par bloc ou par page.
- EF2.3 Le système doit retourner pour chaque *token* : le texte reconnu, la langue estimée et la **boîte englobante** (bounding box).

### 2.4.4 Module M3 : Layout Parsing

- EF3.1 Le système doit segmenter chaque page en zones logiques (blocs, colonnes, sections).
- EF3.2 Le système doit classifier les blocs selon des types métier (*header, contact, résumé, expériences, formations, compétences, langues, centres d'intérêt, divers, etc.*

EF3.3 Le système doit gérer les layouts multi-colonnes et déterminer l'ordre de lecture pertinent.

EF3.4 Le système doit être capable d'ignorer les éléments purement décoratifs (icônes, pictogrammes sans texte, logos).

EF3.5 Le système doit produire un **document intermédiaire** structuré (par exemple en JSON) décrivant la hiérarchie : page → bloc → ligne → token.

#### **2.4.5 Module M4 : Extraction sémantique (NLP)**

EF4.1 Le système doit extraire les informations d'identité : nom, prénom, civilité, éventuellement date de naissance.

EF4.2 Le système doit extraire les coordonnées : email, téléphone, localisation principale (ville, pays).

EF4.3 Le système doit extraire les expériences professionnelles :

- intitulé de poste ;
- nom de l'entreprise ;
- dates de début et de fin (gestion des formats variés) ;
- lieu ;
- description des missions et réalisations.

EF4.4 Le système doit extraire les formations : diplômes, écoles / universités, spécialités, dates, niveau (licence, master, grande école, etc.).

EF4.5 Le système doit extraire les compétences techniques et fonctionnelles (par ex. langages, outils, méthodologies).

EF4.6 Le système doit extraire les langues parlées et niveau estimé si présent.

EF4.7 Le système doit extraire les certifications / projets / publications lorsqu'ils sont identifiés comme tels dans le CV.

#### **2.4.6 Module M5 : Normalisation et mapping JSON**

EF5.1 Le système doit mapper chaque information extraite vers un schéma JSON défini conjointement avec les équipes Forvis.

EF5.2 Le système doit normaliser certains champs :

- pays et villes selon une nomenclature ;
- niveaux d'études selon une grille interne ;

EF5.3 Le système doit gérer la présence d'informations manquantes ou incertaines en les marquant explicitement (`null`, `unknown`, ou indicateur de confiance).

## 2.4.7 Module M6 : API REST et supervision

EF6.1 L'API doit proposer au minimum :

- un endpoint de soumission synchrone d'un CV ;
- un endpoint de soumission asynchrone / batch (optionnel) ;
- un endpoint de consultation de l'état d'un traitement.

EF6.2 L'API doit retourner des codes HTTP cohérents (2xx, 4xx, 5xx) et un corps de réponse explicite.

EF6.3 Le système doit enregistrer les métadonnées de chaque traitement : horodatage, durée, taille du document, résultat (succès / échec).

## 2.5 Exigences non fonctionnelles

### 2.5.1 Performance

- **ENF1.1** : Temps moyen de traitement  $\leq$  10 secondes par CV (format 1–5 pages).
- **ENF1.2** : Traitement en parallèle permettant jusqu'à 50 CV/minute en mode batch.

### 2.5.2 Fiabilité et disponibilité

- **ENF2.1** : Disponibilité minimale 99 % en horaire de bureau.
- **ENF2.2** : Mise en file d'attente automatique en cas de surcharge.

### 2.5.3 Sécurité et conformité RGPD

- **ENF3.1** : Communications chiffrées en HTTPS/TLS.
- **ENF3.2** : Suppression automatique des CV sources après traitement selon politique RGPD.

### 2.5.4 Scalabilité et industrialisation

- **ENF4.1** : Architecture modulaire (remplacement des modèles utilisés OCR / NLP).

### 2.5.5 Observabilité, logs et supervision

- **ENF5.1** : Logs exportables vers monitoring (ELK, Grafana...).
- **ENF5.2** : Exposition de métriques : temps de traitement, taux d'échec, volume.

### 2.5.6 Expérience utilisateur

- **ENF6.1** : Messages d'erreurs explicites et exploitables.

### 2.5.7 Synthèse des objectifs non fonctionnels

Critère	Objectif mesurable	Priorité
Performance	$\leq 10 \text{ s} / \text{CV}$ (moyenne)	Haute
Disponibilité	$\geq 99\%$	Moyenne
Précision sémantique	$\geq 90\%$ extraction correcte	Haute
Sécurité	Chiffrement + Authentification forte	Haute
Scalabilité	50 CV/min en batch	Moyenne
Conformité RGPD	Suppression auto + audit logs	Haute

TABLE 2.3 – KPI et objectifs associés aux exigences non fonctionnelles.

## 2.6 Modèle de données JSON (vue macro)

```
{
  "candidate_id": "string",
  "source": "string",
  "personal_info": {
    "first_name": "string",
    "last_name": "string",
    "email": "string",
    "phone": "string",
    "location": {
      "city": "string",
      "country": "string"
    }
  },
  "experiences": [
    {
      "title": "string",
      "company": "string",
      "location": "string",
      "start_date": "YYYY-MM",
      "end_date": "YYYY-MM or null",
      "description": "string"
    }
  ],
  "education": [
    {
      "degree": "string",
      "field": "string"
    }
  ]
}
```

```

    "school": "string",
    "specialization": "string",
    "start_date": "YYYY-MM",
    "end_date": "YYYY-MM"
  }
],
"skills": [
  {
    "name": "string",
    "category": "technical|functional|other",
    "level": "string (optional)"
  }
],
"languages": [
  {
    "name": "string",
    "level": "string (optional)"
  }
],
"meta": {
  "processing_time_ms": 1234,
  "confidence_global": 0.93
}
}

```

## 2.7 Flux de traitement et gestion des erreurs

### 2.7.1 Flux nominal de traitement

1. Soumission du fichier PDF via l'API.
2. Validation du fichier (format, taille, lisibilité).
3. Pré-traitement éventuel (image preprocessing).
4. OCR multilingue avec extraction des tokens textuels.
5. Segmentation du layout et classification des blocs.
6. Extraction sémantique des informations clés.
7. Normalisation et mapping des données dans le schéma JSON.
8. Retour du JSON structuré au système Forvis.

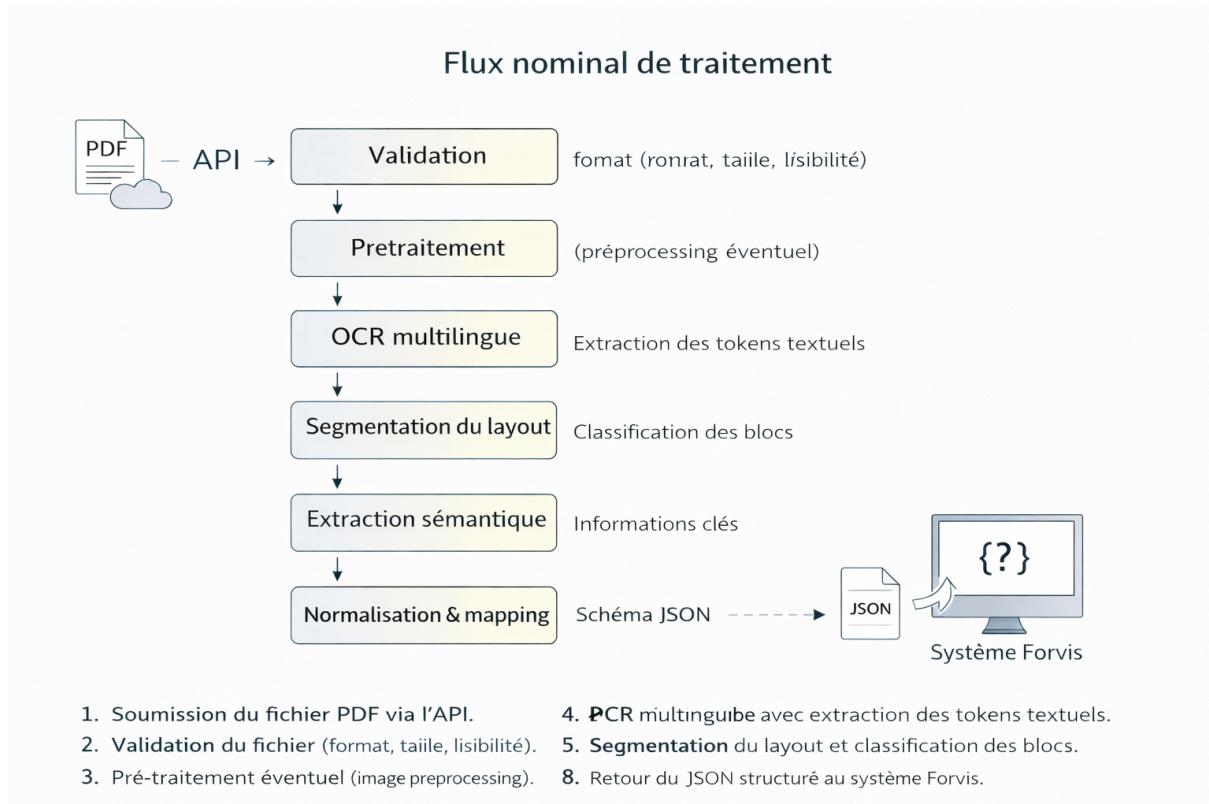


FIGURE 2.3 – Flux de traitement

### 2.7.2 Gestion des erreurs et scénarios alternatifs

Erreur détectée	Conséquence	Traitement attendu
Document non lisible ou corrompu	Impossible d'extraire le contenu	Code 400 + message explicite « document illisible »
OCR de faible confiance	Risque d'informations erronées	Retour JSON avec indicateurs de confiance + log détaillé
Champs critiques manquants (nom, email...)	Fiche candidat incomplète	Signalement + mise en quarantaine si seuil non atteint
Dépassement du temps de traitement	Surcharge ou PDF trop complexe	Timeout + possibilité de relance
Défaillance d'un module du pipeline	Blocage de la chaîne	Retry auto + isolation du job défaillant

TABLE 2.4 – Catalogue des principales erreurs et stratégies de mitigation.

### 2.7.3 Traçabilité et auditabilité

Pour chaque CV traité, le système doit enregistrer :

- l'identifiant unique du traitement ;
- les métriques (durée, scores de confiance) ;
- les logs détaillés des modules activés ;
- la cause d'un échec le cas échéant.

## 2.8 Stratégie de tests et critères d'acceptation

### 2.8.1 Stratégie de tests

- **Tests unitaires** : OCR, NLP, Layout, mapping JSON.
- **Tests d'intégration** : pipeline complet PDF → JSON.
- **Tests de performance** : temps de traitement, mémoire, taux de succès selon volumes.
- **Tests de robustesse** : cas limites, multi-colonnes, pictogrammes, formats exotiques.
- **Validation métier** : comparaison avec échantillon annoté par les équipes RH.

### 2.8.2 Matrice de couverture des tests

Exigence testée	Type de test	Critère de réussite
Extraction identité	Unitaire + métier	100 % : nom + email extraits correctement
OCR FR/EN/AR	Performance	Confiance moyenne $\geq$ 90 %
Segmentation multi-colonnes	Intégration	Ordre de lecture correct dans 95 % des cas
Mapping JSON	Intégration	Zéro écart au schéma JSON
Temps de traitement	Performance	< 5 sec / CV (moyenne)
Tolérance aux erreurs	Robustesse	Pas de crash bloquant, erreurs isolées et traitées

TABLE 2.5 – Matrice de couverture des tests et critères associés.

### 2.8.3 Critères d'acceptation

- **CA1** : Conformité au schéma JSON validé par Forvis.
- **CA2** : Précision extraction infos clés  $\geq 90\%$ .
- **CA3** : Temps moyen  $\leq 5$  secondes par CV.
- **CA4** : Messages d'erreurs clairs et exploitables.
- **CA5** : Aucun échec systémique bloquant en production.
- **CA6** : Documentation complète livrée.

## 2.9 Planning prévisionnel et gouvernance

### 2.9.1 Macro-planning (indicatif)

- **Décembre (S1–S4)** : cadrage, collecte corpus, benchmark OCR/Layout/NLP, architecture, POC bout en bout.
- **Janvier (S5–S8)** : développement modules, API, dataset annoté, tuning.
- **Février (S9–S12)** : tests, amélioration performance, documentation, préparation soutenance/livraison.

### 2.9.2 Gouvernance projet

- **Product Owner** : vision, priorisation, arbitrages fonctionnels.
- **Rituels :**
  - réunions hebdomadaires d'avancement ;
  - revues de sprint avec démonstration ;
  - rétrospectives.

## 2.10 Risques majeurs et plans d'atténuation

Risque	Impact	Plan d'atténuation
Qualité faible des scans	OCR imprécis	Pré-traitement avancé + politique de rejet + retour explicite aux recruteurs.
Variabilité extrême des CV	Extraction insuffisante	Enrichir le corpus, couvrir principaux templates, itérations de tuning.
Manque de disponibilité métier	Retards validation	Planifier ateliers, valider panel prioritaire, nommer référent métier.

Contraintes RGPD mal maîtrisées	Risque non-conformité	Collaboration DPO, durées conservation, anonymisation jeux de tests/logs.
---------------------------------	-----------------------	---

Ce cahier des charges sert de référence partagée entre équipes métier et techniques pour piloter la conception, le développement et l'industrialisation de la solution.