Schéma des workflows métier

Application : dirCops
Projet : Suivi facturation / échéancier / paiement (EMGI)
Date : 2026-02-09
Objectif : Décrire de façon détaillée tous les workflows (états, transitions, acteurs, règles). En MVP il n'y a qu'un seul cycle de vie formalisé : celui de la facture (état dérivé de l'encours).

---

1. Vue d'ensemble

Dans ce projet, les **workflows sont volontairement simples** : pas de chaîne de validation (RH, IT), pas de statuts stockés en base pour les factures, pas de SLA ni relances. Le seul cycle de vie explicite est celui de la **facture** (état calculé : en attente / payée). Les autres entités (client, paiement, écheancier) n'ont pas de machine à états.

Workflow / entité	États ?	Stocké en base ?	Acteur	Détail ci-dessous
Facture (suivi)	Oui (2 états dérivés)	Non (calculé)	Commercial / système	§ 2
Paiement	Non	—	Commercial	Création directe, pas de statut
Écheancier	Non	—	Commercial	Création directe, pas de statut
Client	Non	—	—	Pas de cycle de vie métier

---

2. Workflow « Facture (état de suivi) »

L'état d'une facture est **dérivé** des paiements (non stocké en base). Il sert à l'affichage (Payé / En attente) et au filtrage (échéances à venir = factures en attente).

2.1 Entité concernée

- **Table :** `factures` (+ agrégation des `paiements`).
- **Champ état :** aucun. L'état est calculé : `encours_facture = montant_ht - SUM(paiements.montant)` ; si `encours_facture <= 0` → **Payée**, sinon → **En attente**.

2.2 États

Code état	Libellé	Condition (calcul)
`en_attente`	En attente	encours_facture > 0
`paye`	Payée	encours_facture ≤ 0

2.3 Transitions

Transition	Depuis	Vers	Déclencheur	Acteur	Règle métier
Paiement enregistré	En attente	Payée*	POST paiement (montant ≥ encours restant)	Commercial	Automatique après enregistrement du paiement. Pas de validation.

*Une facture peut rester « En attente » après un paiement partiel (plusieurs paiements possibles).

2.4 Diagramme d'états (machine à états simplifiée)



2.5 Règles complémentaires

- **Transition retour possible** : la suppression d'un paiement (DELETE) fait recalculer l'encours ; une facture « Payée » peut redevenir « En attente » si le dernier paiement est supprimé.
- **Pas de validation métier** : la saisie d'un paiement est directe (pas d'approbation RH/IT).
- **Pas de SLA, relances, escalades** : hors périmètre MVP.
- **Historisation** : les dates de création sont dans `factures.created_at` et `paiements.created_at` ; pas d'audit log « qui a changé quel statut ».

---

3. Entités sans workflow (création directe)

3.1 Paiement

- **Création** : formulaire « Enregistrer un paiement » → POST `/api/paiements.php` (facture_id, montant, date_paiement). Acteur : Commercial.

- **Suppression** : bouton Supprimer → confirmation → DELETE `/api/paiements.php?id=`. Acteur : Commercial. L'encours de la facture est recalculé à la volée après suppression.
- **Pas d'états** : pas de statut « brouillon », « validé », etc. Enregistrement direct.

3.2 Écheancier

- **Création** : formulaire « Ajouter un écheancier » → POST `/api/echeanciers.php` (facture_id, nb_echeances, date_debut_echeancier). Acteur : Commercial.
- **Suppression** : bouton Supprimer → confirmation → DELETE `/api/echeanciers.php?id=`. Acteur : Commercial.
- **Pas d'états** : une facture a ou n'a pas d'écheancier ; pas de statut « en cours », « terminé » sur l'écheancier en MVP.
- **Évolution possible** (post-MVP) : statut par échéance (à venir / payée / partielle) si le client le demande.

3.3 Client

- **Création** : formulaire « Ajouter un client » (accueil) → POST `/api/clients.php` (nom). Acteur : Commercial.
- **Suppression** : bouton Supprimer → confirmation → DELETE `/api/clients.php?id=`. En cascade : factures, paiements, écheanciers du client. Vérification applicative possible (refus si factures existantes selon règle métier). Acteur : Commercial.
- Pas de cycle de vie métier (pas de statut actif/archivé en MVP).

---

4. Suppressions (actions destructives)

Entité	Action	Déclencheur	Confirmation	Effet
Client	Suppression	Bouton Supprimer (liste clients)	Oui (avec avertissement cascade)	DELETE clients.php?id= ; cascade factures → paiements, écheanciers. Vérification possible côté API (refus si factures).
Facture	Suppression	Bouton Supprimer (liste factures, fiche client)	Oui	DELETE factures.php?id= ; cascade paiements, écheancier.
Paiement	Suppression	Bouton Supprimer (liste paiements, fiche client)	Oui	DELETE paiements.php?id= ; encours recalculé à la volée.
Écheancier	Suppression	Bouton Supprimer (liste écheanciers, fiche client)	Oui	DELETE echeanciers.php?id= ; facture inchangée.

Aucun workflow de validation : la confirmation utilisateur (dialog) suffit ; pas d'audit log des suppressions en MVP.

---

5. Ce qui n'est pas dans le périmètre workflow (MVP)

- **Validation en plusieurs étapes** (ex. brouillon → nouveau → en attente lettre → en traitement IT → terminé) : non.
- **Statuts RH / IT distincts** : non.
- **SLA, relances, escalades** : non.
- **Audit log des changements d'état** : non.
- **Champ `global_status` / `hr_status` / `it_status` en base** : non (seul l'état dérivé « en attente / payée » pour la facture, calculé à la volée).

---

6. Synthèse pour le chiffrage

Élément	Présent en MVP ?	Détail
États facture (en attente / payée)	Oui	Dérivés, non stockés ; diagramme § 2.4.
Transitions avec acteur	Oui	1 transition : enregistrement paiement → passage à Payée si encours ≤ 0.
Workflow de validation (approbation)	Non	Saisie directe.
Statuts stockés en base	Non	Uniquement calcul à l'affichage.
SLA / relances / escalades	Non	Hors scope.
Historisation / audit log	Non	Uniquement created_at.

Les workflows sont donc **bien détaillés en schéma** pour ce qui existe (facture) ; le reste du projet repose sur des créations directes sans machine à états.