Guide de l’agent comptable avancé

Fonctionnalités, architecture des sous-agents, mise en œuvre et jeux de données de test

ADVANCED ACCOUNTING AGENT / WEST AFRICA

1. Présentation générale

L’agent comptable avancé est une solution d’assistance par intelligence artificielle destinée aux procédures comptables en Afrique de l’Ouest, avec une priorité initiale donnée à la Côte d’Ivoire. Il transforme les reçus, factures, relevés bancaires et tableurs en propositions d’écritures comptables en partie double, structurées et vérifiables, selon les conventions du SYSCOHADA et des contrôles fiscaux configurés par pays.

Il ne se limite pas à automatiser Excel : il rapproche les faits extraits des pièces justificatives, des comptes appropriés et des traitements fiscaux pertinents. Un comptable qualifié doit vérifier et approuver les écritures avant leur comptabilisation définitive.

2. Fonctionnalités principales

- Lecture des documents : prise en charge des PDF, images JPEG et fichiers XLSX ; extraction des émetteurs, dates, lignes de facturation, montants, taxes et références de paiement.
- Normalisation des données : harmonisation des dates, montants en FCFA (XOF), fournisseurs, centres de coûts et correspondances avec le plan comptable validé.
- Contrôles de conformité : détection des mentions obligatoires manquantes, vérification des calculs et application de règles configurables relatives au SYSCOHADA, à la TVA et à la facturation normalisée électronique (FNE) en Côte d’Ivoire.
- Production des résultats : génération de projets d’écritures équilibrées (débit/crédit), rattachées aux pièces d’origine, pour contrôle et importation ultérieure.

3. Sous-agents et traitement parallèle

Pour des tâches indépendantes, comme analyser une facture pendant le rapprochement d’un relevé bancaire, une file de tâches au niveau de l’application est préférable à une exécution strictement séquentielle. Le programme d’orchestration peut affecter des tâches à des travailleurs en nombre limité et regrouper leurs résultats.

- Architecture conseillée : file de tâches, groupe de travailleurs à concurrence limitée, mécanisme de relance, identifiants d'idempotence et suivi persistant de l'état des opérations.
- Parallélisme dans une session : `Promise.allSettled` convient pour agréger des appels JavaScript asynchrones indépendants, lorsque l'environnement le permet. Éviter les écritures simultanées sur le même grand livre ou classeur.
- Optimisation mémoire : envisager des modèles locaux quantifiés compatibles, limiter le contexte envoyé au modèle, mettre en cache les données extraites et réserver les calculs et rapprochements déterministes au code classique.
- Contrôle interne : signaler les extractions incertaines, prévoir une validation humaine et conserver un journal d'audit avec les références des justificatifs.

4. Comparaison technique : Python et Rust

Critère	Python	Rust
Développement	Prototypage rapide ; nombreuses bibliothèques OCR, données et IA	Apprentissage plus exigeant ; mise en œuvre plus longue
Exécution	Bonnes performances avec bibliothèques optimisées	Excellente maîtrise des performances et de la mémoire
Exploitation	Intégration simple de pandas, NumPy et pydantic	Intéressant pour certains calculs intensifs
Choix recommandé	Langage principal de l'agent	À envisager après mesure d'un goulot d'étranglement

Conclusion : privilégier Python. La lecture des fichiers, l'inférence des modèles et les échanges externes constituent souvent les principales sources de latence. Mesurer les performances réelles avant d'envisager une réécriture en Rust.

5. Démarrage rapide

Prérequis : environnement Python, dépendances installées, solution de lecture de documents et modèle de langage adapté (local ou API), ainsi qu'un référentiel de règles comptables et fiscales validé.

```
pip install -r requirements.txt
python advanced_accounting_skill.py --receipt /data/recu.jpg --doc-type Receipt
```

Remarque : cette commande constitue un exemple d'interface en ligne de commande. Elle suppose que le fichier `advanced_accounting_skill.py` et ses paramètres ont été implémentés.

6. Données de test : novembre 2026

Scénario : un responsable transmet plusieurs dépenses et factures fournisseurs du mois de novembre 2026. L’agent doit détecter les doublons, vérifier les informations fiscales, classer les opérations et proposer des écritures équilibrées en FCFA. Les données ci-dessous sont fictives et purement illustratives.

Pièce / type	Données reçues	Résultat attendu
FAC-001 / facture	Fournitures de bureau : HT 100 000 FCFA ; TVA indiquée 18 000 ; TTC 118 000	Vérifier les totaux ; ventiler charge et TVA ; créditer le fournisseur pour 118 000, sous réserve du droit à déduction
REC-002 / reçu	Transport : 12 500 FCFA ; règlement en espèces ; aucune TVA mentionnée	Enregistrer la charge brute et la sortie de caisse ; ne pas inventer une TVA déductible
BQ-003 / banque	Débit de 118 000 FCFA ; référence FAC-001	Rapprocher le paiement fournisseur ; ne pas enregistrer une seconde charge
FAC-004 / facture	Même fournisseur, même date, même référence et 118 000 FCFA que FAC-001	Signaler un doublon potentiel ; suspendre la comptabilisation avant vérification

Exemples d’écritures proposées

Pièce	Débit	Crédit
FAC-001	Charge de fournitures : 100 000 ; TVA récupérable : 18 000 (si admissible)	Fournisseur : 118 000
REC-002	Charge de transport : 12 500	Caisse : 12 500
BQ-003	Fournisseur : 118 000	Banque : 118 000

Vérifications attendues : (1) équilibre de chaque écriture ; (2) rapprochement du règlement sans double charge ; (3) alerte sur FAC-004 ; (4) aucune création de donnée fiscale absente ; (5) conservation des identifiants de source et du statut de validation.

7. Limites et précautions réglementaires

Le SYSCOHADA constitue un cadre comptable, tandis que la TVA et les obligations FNE relèvent de règles nationales de fiscalité et de facturation. La solution doit utiliser des règles maintenues, datées et vérifiées pour chaque juridiction. Elle ne garantit pas, à elle seule, la conformité réglementaire et ne remplace pas la validation professionnelle.