Grille d’auto-évaluation

Partie 3 : Labos 6 et 7 — LOG430

# Niveaux d’évaluation

Chaque critère est évalué selon l’un des quatre niveaux suivants :

— Excellent : Travail dépassant les attentes, complet et justifié.

— Suffisant : Répond aux attentes avec des justifications claires.

— Suffisant avec améliorations : Efforts visibles, mais lacunes à combler.

— Insuffisant : Critère absent ou mal exécuté.

1

# Grille d’évaluation détaillée

2

Critère Excellent Suffisant Suffisant avec Insuffisant

améliorations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Définition du scénario métier | Processus métier en moins que 4 étapes, diagramme et README com-  plet | Scénario clair,  étapes identifiées | Scénario partiel, diagramme manquant | Scénario absent ou incomplet | |
| 2. Saga orchestrée (Labo  6) | Orchestrateur  central fonctionnel, appels synchrones com-  plets | Orchestrateur implémenté, flux basique | Orchestrateur  partiel, flux  incomplet | Pas de saga or- | |
| chestrée |  |
| 3. Machine d’état (Labo 6) | Diagramme + code clair, persistance de l’état et logs complets | Diagramme fourni et persistance basique | Modèle partiel, persistance manquante | Pas de  d’état | machine |
| 4. Gestion des échecs & compensation (Labo 6) | Rollback/compensationCompensation documentés, tests implémentée, d’échec inclus tests manquants | | Gestion d’erreurs minimale | Pas de  d’échecs | gestion |
| 5. Simulation de cas  d’échec (Labo 6) | Tests automatisés Quelques scripts d’échec et robus- ou scénarios ma-  tesse nuels | | Tests partiels, non automatisés | Aucun  d’échec | test |
| 6. Producteurs d’événements (Labo 7) | Événements Événements cor-  JSON horodatés, rects, sans replay  ID unique, retry ni DLQ  + DLQ | | Producteurs partiels, schéma incomplet | Pas de  teurs | produc- |

Critère Excellent Suffisant Suffisant avec Insuffisant améliorations

3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7. Topics/Queues &  schéma (Labo 7) | Nommage clair,  versionnement, gestion de schéma | Nommage cohérent, schéma de  base | Organisation  confuse, pas de  version | Pas de to-  pics/queues |
| 8. Consommateurs idempotents (Labo 7) | Plus que 2  consommateurs, idempotence  et logs JSON  structurés | 1 consommateur ou idempotence  partielle | Consommateurs minimaux, sans idempotence | Pas de consom-  mateurs |
| 9. Event Store & replay  (Labo 7) | Stockage complet, endpoint de replay, reconstruction OK | Stockage et replay basique | Stockage sans replay | Pas d’Event Store |
| 10. CQRS & read models  (Labo 7) | Séparation  claire Com-  mand/Query, projections fonctionnelles | CQRS partiel, read models  limités | Read models  manquants | Pas de CQRS |
| 11. Observabilité & traça-  bilité | Prometheus exposé, Grafana, trace & corrélation IDs | Métriques ou logs structurés seuls | Observabilité limitée | Pas d’observabi-  lité |
| 12. Tests automatisés & e2e | CI complète, tests bout à bout, scénarios échec/replay | Tests unitaires + quelques e2e | Tests manuels, pas d’automatisation | Pas de tests |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Critère | Excellent | Suffisant | Suffisant avec améliorations | Insuffisant |
| 13. Documentation &  ADR | README détaillé, guide de déploiement, plus  de 2 ADR | README et rap-  port succincts | Documentation partielle, ADR manquants | Pas de documen-  tation |
| 14. Structure du dépôt &  CI/CD | Répertoires lab6/, lab7/, pipelines CI/CD, manifests k8s | Organisation de base, CI partielle | Structure confuse,  CI manquante | Pas de CI/CD |
| 15. Saga chorégraphiée  (Labo 7) | Flux d’événements complet, coordination sans orchestrateur, diagramme clair | Saga implémentée, coordination partielle | Saga partielle, non testée | Pas de saga |
| 16. Événements compensatoires (Labo 7) | Événements de  compensation robustes, testés et idempotents | Présents mais sans test automa-  tisé | Schéma incomplet ou non testé | Aucun événement compensatoire |

4

# Autoévaluation à compléter par l’étudiant(e)

Nom :Maksym Pravdin Date : 23 juillet 2025

Pour chaque critère, cochez la case correspondant à votre niveau et ajoutez un court commentaire :

Critère 1. Définition claire du scénario métier

(✓) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant

Commentaire : La saga choisie est celle de la création d’une commande en ligne avec le panier d’un client. ÉTAPE 1 : on récupère le panier du client avec le service stocks. ÉTAPE 2 : On enregistre la vente avec le service des ventes ÉTAPE 3 : On fait la mise à jour de l’inventaire avec le service stocks ÉTAPE 4 : On augmente la réputation/rank de l’utilisateur qui a fait la vente avec le service auth.

Critère 2. Saga Orchestrée structurée (Pub/Sub)

(✓) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant

Commentaire : Pour la saga orchestrée c’est le service « orchestr-sales » qui est le publisher qui orchestre toute la saga et les services auth, stocks (les 2 instances) et sales sont ces subscribers qui sont traitées dans la saga de manière séquentielle.

Critère 3. Utilisation de RabbitMQ pour la communication entre services

(□) Excellent (✓) Suffisant (□) À améliorer (□) Insuffisant

Commentaire : Pour le labo 6, je n’ai pas utilisé RabbitMQ (il est utilisé dans le labo 7). À la place, j’ai codé un service orchestrateur qui fait la même chose, car RabbitMQ n’était pas mentionné dans l’énoncé du labo 6.

Critère 4. Implémentation d’un ou plusieurs producteurs d’événements (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 5. Création de consommateurs idempotents et découplés

(□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 6. Définition des schémas d’événements clairs et cohérents (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 7. Persistance des événements dans un Event Store

(□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 8. Fonction de relecture (event replay) fonctionnelle

(□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 9. Implémentation du pattern CQRS (Command / Query) (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 10. Lecture à partir de read models cohérents (projection) (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 11. Observabilité : logs structurés et métriques exposées

(□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 12. Visualisation dans Grafana ou autre outil de monitoring (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 13. Tests de bout en bout (e2e) ou automatisés des flux d’événements (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 14. Structure du projet et documentation claire (README, ADR) (□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 15. Implémentation d’une saga chorégraphiée

(□) Excellent (□) Suffisant (□) À améliorer (□) Insuffisant Commentaire :

Critère 16. Déploiement sur Kubernetes (section optionnelle)

(□) Réalisé (□) Partiellement (□) Non réalisé (□) Non tenté Commentaire :