

# **Architecture Microservices – Kafka & Saga Orchestration**

**Documentation Technique – Version Corporate**

# Table des matières

1. Page de garde
2. Introduction
3. Explication du schéma d'architecture micro-services
4. Les micro-services principaux
5. Le rôle central de Kafka
6. Orchestration Saga
7. Gestion des erreurs
8. Résilience & Bonnes pratiques
9. Diagramme d'Architecture
10. Diagramme de Séquence
11. Résumé visuel et conceptuel

## Introduction

Cette documentation présente le fonctionnement complet d'une architecture micro-services basée sur Apache Kafka et le pattern Saga...

## Explication du schéma d'architecture micro-services

L'architecture illustre un système distribué où chaque service est autonome et communique via Kafka...

## Les micro-services principaux

Order Service : reçoit commande, publie order.placed, consomme stock.checked et payment.result...

Stock Service : consomme order.placed, vérifie produits, publie stock.checked...

Billing Service : consomme payment.request, traite les paiements, publie payment.result...

Notification Service : consomme order.confirmed / order.failed, informe le client.

## Le rôle central de Kafka

Kafka assure le découplage total, la résilience, la rétention, le replay, la scalabilité horizontale...

## Orchestration Saga

Étapes normales : order.placed → stock.check.request → payment.request → order.confirmed...

## Gestion des erreurs

Stock insuffisant : order.failed directement.

Paiement échoué : compensation stock.release puis order.failed.

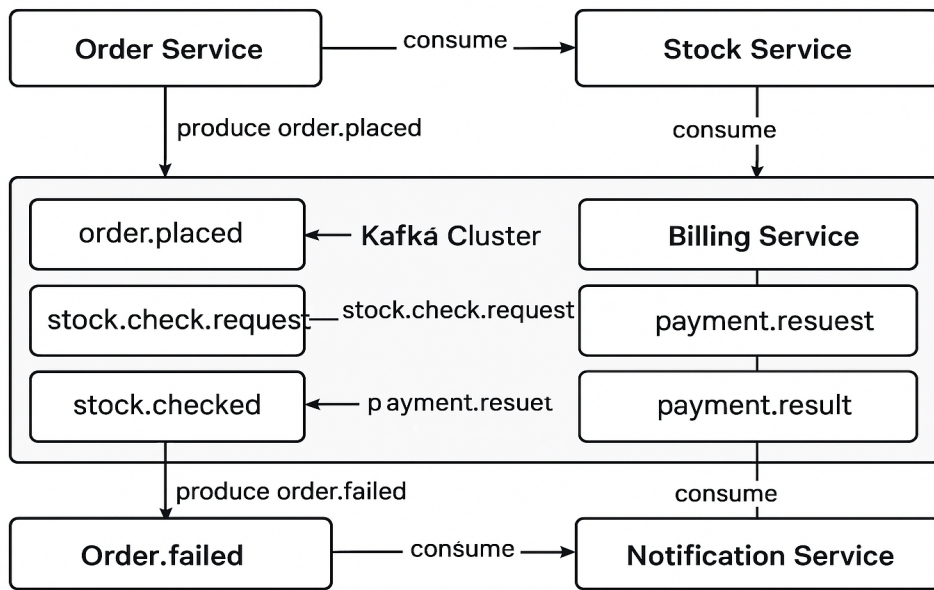
## Résilience & bonnes pratiques

Idempotence, retries, DLQ, observabilité, traces distribuées, correlationId...

## Résumé visuel et conceptuel

Architecture entièrement asynchrone, orchestrateur Saga fiable, compensations robustes, services découplés.

## Diagramme d'Architecture



# Diagramme de Séquence

