

Social Commerce Platform (Intro)

Nexgate is a unified social commerce platform that combines eCommerce, event management, and social networking into a single ecosystem. Built for the East African market, Nexgate enables users to buy, sell, connect, and experience — all in one place.

The Concept

Nexgate reimagines how commerce happens online. Instead of separating shopping, events, and social media into different apps, Nexgate brings them together.

User creates → Product or Event → **Posts to Social** → Followers see in feed → **Purchase or Book**

A product isn't just listed in a shop — it's shared, liked, and commented on. An event isn't just a ticket page — it's a post that spreads organically through your network.

Platform Architecture

Nexgate follows a microservices architecture with nine independent services communicating via RabbitMQ message broker and backed by PostgreSQL databases.

Core Services

1. **Authentication** — User registration, login, JWT tokens, OAuth, device management
2. **User Profile** — Profiles, followers/following, privacy settings, trust scores
3. **File Manager** — Media uploads, image processing, BlurHash, video transcoding via FFmpeg
4. **Payment** — Payment methods, M-Pesa/Selcom/Tembo Plus integration, escrow, transactions
5. **Direct Message** — Real-time private messaging, conversations, media sharing

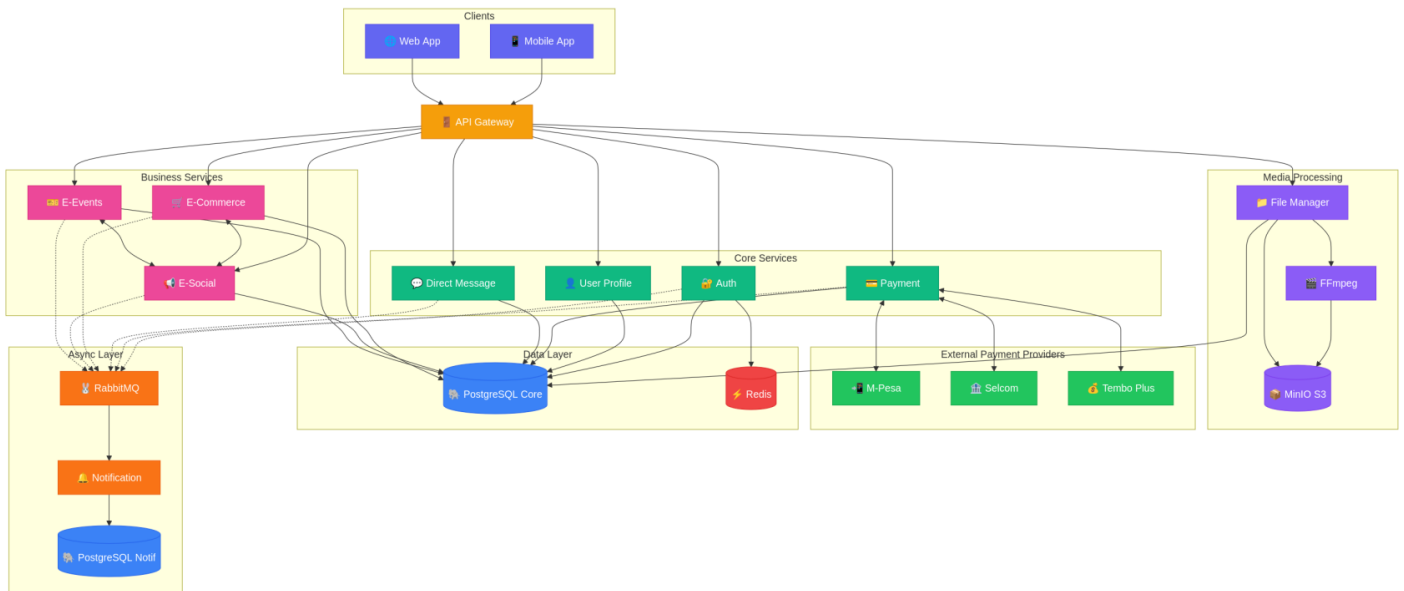
Business Services

- 6. **E-Commerce** — Shops, products, orders, cart, checkout, reviews, seller verification
- 7. **E-Events** — Event creation, ticketing, RSA-signed tickets, check-ins, organizer ratings
- 8. **E-Social** — Posts, feeds, likes, comments, reposts, polls, mentions, hashtags

External Service

- 9. **Notification** — Push notifications, in-app alerts, email, SMS (async via RabbitMQ)

Architecture Diagram



Service Relationships

Service	Role	Connects To
Authentication	Identity, tokens, sessions	Redis, PostgreSQL Core, RabbitMQ
User Profile	Profiles, follows, trust scores	PostgreSQL Core, RabbitMQ

Service	Role	Connects To
File Manager	Media uploads, processing	FFmpeg, MinIO, PostgreSQL Core
Payment	Transactions, escrow	M-Pesa, Selcom, Tembo Plus, PostgreSQL Core, RabbitMQ
Direct Message	Private conversations	PostgreSQL Core, RabbitMQ
E-Commerce	Shops, products, orders	E-Social (embed products in posts), PostgreSQL Core, RabbitMQ
E-Events	Events, tickets, check-ins	E-Social (embed events in posts), PostgreSQL Core, RabbitMQ
E-Social	Posts, feeds, interactions	E-Commerce, E-Events (receives embeds), PostgreSQL Core, RabbitMQ
Notification	Push, email, SMS	RabbitMQ (consumes events), PostgreSQL Notif

Data Flow Summary

Flow	Path
User creates product	E-Commerce → Post to E-Social → Followers see in feed
User creates event	E-Events → Post to E-Social → Followers see in feed
User uploads media	File Manager → FFmpeg (if video) → MinIO → BlurHash generated
User makes payment	Payment → External Provider (M-Pesa/Selcom/Tembo) → Callback → Update order
Any action triggers notification	Service → RabbitMQ → Notification Service → Push/Email/SMS

Infrastructure Components

Component	Purpose
PostgreSQL (Core)	Shared DB for Auth, User, File, Payment, DM, E-Commerce, E-Events, E-Social
PostgreSQL (Notif)	Separate DB for Notification service
Redis	Session storage, caching, rate limiting
RabbitMQ	Async messaging, triggers notifications
MinIO	S3-compatible object storage for all media

Component	Purpose
FFmpeg	Video transcoding, thumbnails, compression

Tech Stack

Layer	Technology
Backend	Java 21, Spring Boot 3.x
Database	PostgreSQL (Core + Notification)
Cache	Redis
Messaging	RabbitMQ
Storage	MinIO (S3-compatible)
Media Processing	FFmpeg
Payments	M-Pesa, Selcom, Tembo Plus
Security	JWT, RSA signatures, device fingerprinting

Revision #3

Created 11 December 2025 07:38:39 by Admin Qbit

Updated 11 December 2025 08:10:37 by Admin Qbit