

E-Commerce Platform Architecture

System Design Document v2.0

1. System Overview

This document describes the architecture of our e-commerce platform. The system handles user authentication, product catalog, shopping cart, order processing, and payment integration.

2. System Components

- Web Frontend:	React SPA served via CloudFront CDN
- API Gateway:	Kong API Gateway with rate limiting and JWT validation
- Auth Service:	Node.js service handling OAuth2, JWT tokens, MFA
- Product Service:	Python FastAPI for catalog management
- Order Service:	Python FastAPI for order processing
- Payment Service:	Node.js integration with Stripe API
- Notification Service:	Go service for email/SMS via SendGrid
- PostgreSQL Database:	Primary data store with read replicas
- Redis Cache:	Session storage and API response caching
- Elasticsearch:	Product search and analytics

3. Data Flows

- User -> Web Frontend -> API Gateway -> Auth Service (login/register)
- Web Frontend -> API Gateway -> Product Service (browse catalog)
- Web Frontend -> API Gateway -> Order Service -> Payment Service (checkout)
- Payment Service -> Stripe API (process payment)
- Order Service -> Notification Service -> SendGrid (order confirmation)
- All Services -> PostgreSQL (persist data)
- API Gateway -> Redis (cache responses)

4. Trust Boundaries

- Internet Zone: Web Frontend, CDN
- DMZ: API Gateway, Load Balancer
- Internal Network: All microservices
- Data Zone: PostgreSQL, Redis, Elasticsearch
- External Services: Stripe, SendGrid

5. Security Controls

- TLS 1.3 for all external communications

JWT tokens with 15 minute expiry