Sample Microservices Architecture

This document outlines a sample microservices architecture for a scalable e-commerce platform.

1. Frontend Services:
- Web Dashboard (React/Next.js): User interface for customers and admins.
- Mobile App API (BFF - Backend for Frontend): Optimized API for mobile clients.

2. Backend Services:
- User Service (Python/FastAPI): Handles authentication, user profiles.
- Product Service (Node.js/Express): Catalog management, search.
- Order Service (Go/Gin): Order processing, checkout flow.
- Payment Service (Java/Spring Boot): Payment gateway integration.

3. Data Layer:
- Users DB: PostgreSQL (Relational data for users)
- Product DB: MongoDB (Document store for flexible product schema)
- Orders DB: PostgreSQL (Transactional integrity for orders)
- Cache: Redis (Session management, product caching)

4. Messaging & Async:
- RabbitMQ: For order processing events and notifications.
- Kafka: Using for high-throughput event streaming and analytics.

5. Infrastructure:
- Kubernetes (K8s) for orchestration.
- Docker containers for all services.
- AWS ECS/EKS or Google GKE for managed hosting.

Constraints:
- High Availability: 99.9% uptime required.
- Scalability: Auto-scaling enabled for peak loads (Black Friday).
- Security: OAuth2/OIDC for authentication.

This architecture is designed to handle high traffic loads while maintaining modularity.