Event Ticketing Platform - Scenario Flow

Scenario: Seat Booking Flow with Temporary Seat Lock

1. User selects seat from Android/iOS/Web client and sends booking request.
2. Request passes through Route53 (DNS) and CDN before reaching API Gateway.
3. API Gateway validates authentication (via User Service).
4. API Gateway routes request to Booking Service.
5. Booking Service validates event details using Event Service.
6. Booking Service requests seat lock from SeatLock Service.
7. SeatLock Service performs atomic lock using Redis with TTL (e.g., 10 minutes).
8. If lock successful, Booking Service creates a PENDING booking record in MySQL.
9. Payment processing occurs (external integration).
10. On success, booking is CONFIRMED and seat lock is consumed/released.
11. Response returned back through API Gateway to the client.

Failure Paths

1. Seat already locked → return 409 Conflict.
2. Lock expires automatically via Redis TTL.
3. Payment failure → booking marked FAILED and seat released.

Architectural Notes

1. Seat availability = Total seats − Confirmed bookings − Active Redis locks.
2. SeatLock Service isolates concurrency control.
3. Services communicate via unidirectional dependencies.
4. API Gateway acts as single entry point.