Scalable Event Ticketing & Seat Allocation System

Submitted to: Gaurav M

By: I Jaswanth

The Challenge: High Demand, Critical Performance

High Traffic Surges

Managing massive concurrent user loads during event launches without system collapse.

Preventing Oversells

Ensuring no more tickets are sold than available to maintain customer trust and operational integrity.

Low-Latency Checkout

Providing a swift and seamless purchase experience, critical for user satisfaction during peak demand.

High Availability

Guaranteeing continuous service operation to maximize sales opportunities and user access.

Made with GAMMA

Key Stakeholders in Our Ecosystem

Understanding the diverse needs of those interacting with the system is crucial for a robust design.

Buyers

Seeking a fast, reliable, and user-friendly experience for ticket discovery and purchase.

Event Organizers

Requiring tools for event creation, inventory management, and comprehensive sales reporting.

Payments & Finance

Needing secure, idempotent transaction processing and accurate financial reconciliation.

Customer Support

Access to booking history and tools for efficient issue resolution and customer assistance.

Made with GAMMA

User Stories: A Glimpse into Interactions

How different users interact with the system to achieve their goals.

Buyer Perspective

- As a buyer, I want to **browse events** to find what interests me.
- As a buyer, I want to **select my preferred seats** from an interactive map.
- As a buyer, I want to **reserve seats temporarily** to complete my purchase without losing them.
- As a buyer, I want to **pay securely** to finalize my ticket acquisition.

Organizer & Support Perspective

- As an organizer, I want to **create new events** and define seating configurations.
- As an organizer, I want to **view sales reports** to monitor event performance.
- As support, I want to **view a customer's booking history** to assist with inquiries or changes.

Made with GAMMA

Critical Non-Functional Requirements

Defining the performance, reliability, and consistency standards for our system.

Rapid Checkout

99th percentile checkout time of less than 2 seconds.

High Concurrency

Ability to gracefully handle over 200,000 concurrent users during peak event launches.

Exceptional Uptime

Achieving 99.95% system uptime to ensure continuous service availability.

Strong Seat Consistency

Guaranteeing strong consistency for seat commit operations to prevent overselling.

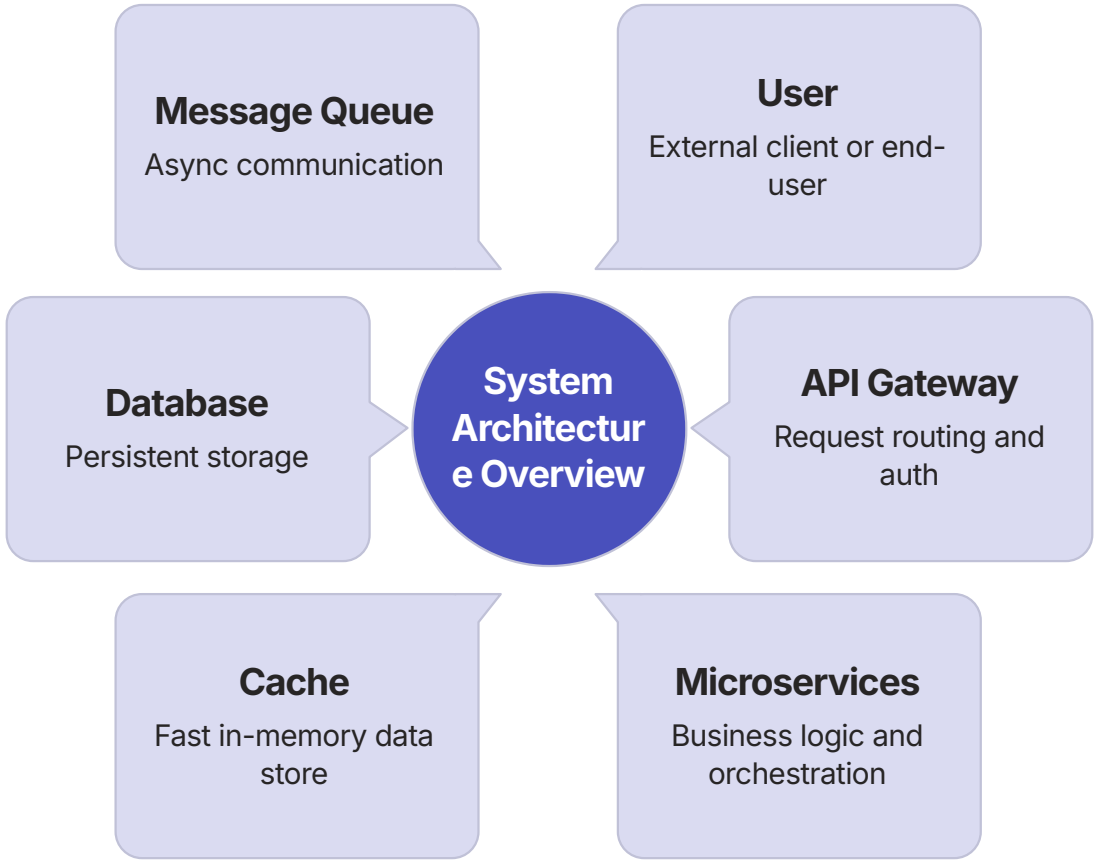
Idempotent Payments

Processing payments in an idempotent manner to avoid duplicate charges or incorrect states.

Made with GAMMA

System Architecture Overview

A high-level view of the components and their interactions within the ticketing system.



Core Microservices: The Heart of the System

Each service is designed for specific functionalities, promoting modularity and scalability.

1

Event Service

Manages event creation, details, scheduling, and overall event lifecycle.

2

Seat Service

Handles seat inventory, availability, and manages the seat map for various venues.

3

Reservation Service

Manages temporary seat locks, reservations, and ensures atomic operations for seat allocation.

4

Payment Service

Processes all financial transactions securely, handles refunds, and integrates with payment gateways.

5

Notification Service

Sends transactional emails, SMS, and in-app notifications for bookings, updates, and reminders.

Made with GAMMA

Essential Supporting Components

Robust infrastructure components that enable high performance and resilience.

Redis Cache

High-speed in-memory data store for frequently accessed data like seat availability, reducing database load.

Kafka/SQS Queues

Asynchronous message brokers for reliable communication between services and handling spikes in traffic.

Sharded SQL/NoSQL DB

Scalable and highly available data storage solution, optimized for both transactional and analytical workloads.

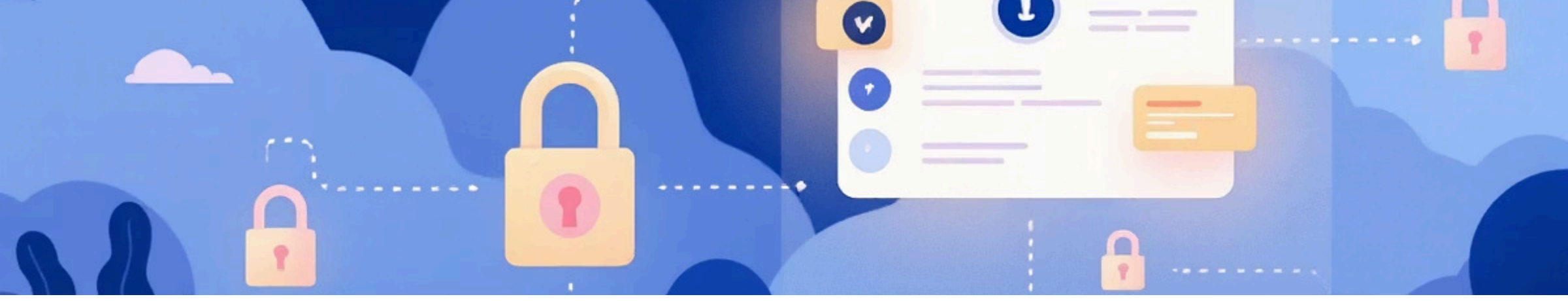
Load Balancer (LB)

Distributes incoming network traffic across multiple servers to ensure optimal resource utilization and prevent overload.

API Gateway

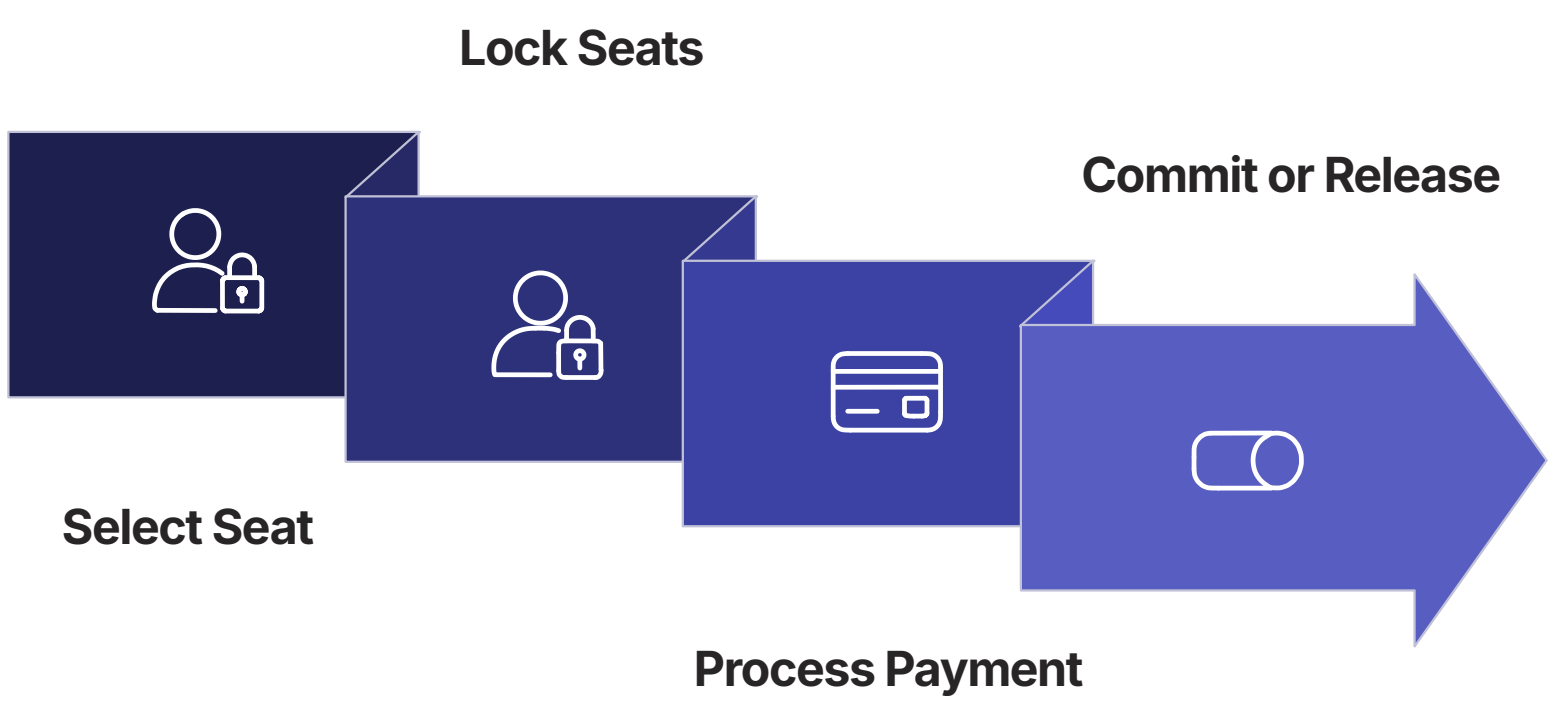
Single entry point for all client requests, handling routing, authentication, and rate limiting.

Made with GAMMA



Seat Reservation Flow

A step-by-step process outlining how seats are securely reserved and committed.



Made with GAMMA

API Contracts: Defining Interactions

Examples of key API endpoints for managing seat reservations.

Initiate Seat Reservation

```
POST /seats/reserve
{
  "eventId": "UUID",
  "seatIds": ["seat-A1", "seat-A2"],
  "userId": "UUID",
  "expiryInMinutes": 5
}
```

This endpoint attempts to lock specified seats for a given user for a limited time.

Commit Reserved Seats

```
POST /seats/commit
{
  "reservationId": "UUID",
  "paymentToken": "string",
  "amount": {
    "currency": "USD",
    "value": 150.00
  }
}
```

Confirms the reservation upon successful payment, finalizing the seat allocation.

Made with GAMMA