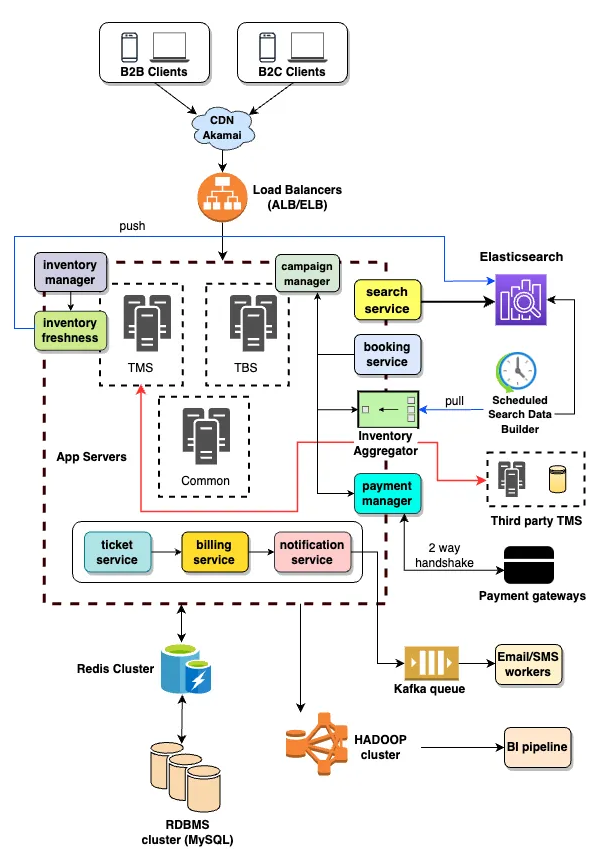
**Movie Ticket Booking Platform System Architecture**



High-Level Components

1. **Clients**:
   * **B2B Clients**: Represent theatre partners who can manage their theatres, shows, inventory, etc.
   * **B2C Clients**: Represent end users who are searching and booking tickets via the platform.
2. **CDN (Content Delivery Network)**:
   * **Akamai CDN** is used to deliver static content faster by caching it close to the clients for both B2B and B2C. This ensures that the platform scales well under heavy traffic.
3. **Load Balancers (ALB/ELB)**:
   * Load Balancers distribute traffic to the application servers to ensure scalability and high availability. **ALB (Application Load Balancer)** or **ELB (Elastic Load Balancer)** is typically used in cloud environments such as AWS to route traffic efficiently.

Core Microservices Services (Application Servers)

These services make up the primary operations of the platform and are divided into **TMS (Theatre Management System)** and **TBS (Ticket Booking System)**.

1. **Inventory Manager**:
   * Manages the theatre's inventory, including configuring theatre details, shows, seat layouts, campaigns, etc., for B2B partners.
2. **Inventory Freshness**:
   * This ensures that the inventory is always up-to-date in **Elasticsearch** and is pushed in real-time whenever a change is made (e.g., seats sold, new shows added).
3. **Campaign Manager**:
   * Configures offers and deals for both internal teams and theatre operators. These deals can be applied during searches and bookings.
4. **Search Service**:
   * Allows both B2B and B2C clients to search for available movies, theatres, and shows. It retrieves data from **Elasticsearch** and applies filters such as location, language, and ongoing offers.
5. **Booking Service**:
   * Manages the booking workflow, including reserving seats, payment initiation, and confirmation of bookings. Real-time data is pulled from the **Inventory Aggregator**.
6. **Inventory Aggregator**:
   * Combines data from local **TMS** systems and third-party **TMS** systems. It standardizes the data and provides it to the **Booking Service**.
7. **Payment Manager**:
   * Handles all the payment-related workflows and integrates with external payment gateways. It performs a two-way handshake with the gateway for payment confirmation.
8. **Ticket Service**:
   * Generates the tickets after successful booking and payment. It also handles ticket modifications, cancellations, and triggers workflows for the next steps (e.g., invoicing).
9. **Billing Service**:
   * Manages the ledger for every transaction, tracking payments, commissions, offers, etc. It supports payouts for theatre operators and the organization.
10. **Notification Service**:
    * Sends out notifications via email, SMS, or push notifications triggered by events such as booking confirmation, ticket cancellation, etc.

Data & Caching Layers

1. **Redis Cluster**:
   * Used as an in-memory data store and cache for improving performance by reducing database load and providing faster data retrieval for frequently accessed information.
2. **RDBMS Cluster (MySQL)**:
   * This is the primary relational database where persistent data like user information, bookings, transactions, and inventory details are stored.
3. **Hadoop Cluster & BI Pipeline**:
   * **Hadoop Cluster** is used for big data analytics, storing vast amounts of historical booking, transactional, and user behavior data.
   * The **BI Pipeline** analyzes the data from Hadoop and provides insights into performance, campaigns, user behavior, etc., to internal teams.

Message Queuing & Communication

1. **Kafka Queue**:
   * A distributed messaging system that allows services like the **Notification Service**, **Ticket Service**, and others to communicate asynchronously. It ensures that events like successful bookings are queued and handled efficiently by downstream systems (e.g., email workers, BI analytics).

External Dependencies

1. **Third-Party TMS**:
   * The platform can integrate with third-party theatre management systems to pull inventory, shows, and seat availability data through the **Inventory Aggregator**.
2. **Payment Gateways**:
   * The platform interacts with external payment gateways to process payments securely. A two-way handshake ensures that payments are processed and verified.

**Data Pipeline & Analytics**

1. **Email/SMS Workers**:
   * These workers are responsible for sending out communication (e.g., booking confirmation, promotional messages) asynchronously, triggered via the **Kafka Queue**.
2. **BI Pipeline**:
   * This pipeline processes data stored in **Hadoop**, analyzing it to provide insights to stakeholders for business decisions, marketing strategies, and operational improvements.

Data Base ER Diagram



Relationships

* **User** can make multiple **Bookings**.
* **Theatre** can have multiple **Shows**.
* **Movie** can have multiple **Shows**.
* **Show** occurs in a specific **Screen** within a **Theatre**.
* **Screen** contains multiple **Seats**.
* **Booking** is associated with a specific **Show** and can include multiple **BookingDetails**.
* **BookingDetail** is linked to a specific **Seat**.
* **Payment** is linked to a **Booking**.
* **Theatre** has KYC information stored in **KYCInfo**.
* **Notification** is associated with a **User**.