High-Level Architecture for Multi-Tenant Booking Platform

Contents

[1️. Architectural Requirements 1](#_Toc190628232)

[2. User Categories 1](#_Toc190628233)

[3. Features 2](#_Toc190628234)

[4. Data Storage 3](#_Toc190628235)

[5. Communication / Integration styles 4](#_Toc190628236)

[6. Technology stack summary 4](#_Toc190628237)

## 1️. Architectural Requirements

* **Multi-Tenancy:** The solution or platform should support multi-tenancy in terms of service consumption, tenant specific features, data isolation.
* **Windows Platform & Cloud Agnostic:** Initially target windows on-premises. We should be able to transfer the workload from on-premise to other cloud providers.
* **Scalable & Performant:** Requires service to the scalable and performant for regional and global expansion and support unpredictable traffic.
* **Security & Compliance:**
  + **Data Protection**: Encryption at rest and in transit
  + **Compliance with Regional Laws**: GDPR, CCPA, PCI-DSS for payments
  + **Audit: Logging & Monitoring** for threat detection
  + **Application:** Application needs to be secure to different OWSAP vulnerabilities.
  + **Network:** system components must be within a private network and only required public access must be exposed.
  + **Access:** User and services mush have only necessary access to execute actions based on roles and permissions.

## 2. User Categories

* **Movie Goers (End Users) -** Individuals who use the platform to browse, book, and manage movie tickets and related purchases. These users should be able to Register, browse movies, book tickets, and purchase eatables via a web browser and mobile devices.
* **Movie Managers (Administrators)** Personnel responsible for managing movies, schedules, seating, and inventory. These users should be able to Manage movie listings, show schedules, and seat availability. Oversee eatables inventory and pricing. Administer overall operational aspects of the movie theater. The operations should be performed only when they connect to the organization’s private network.

## 3. Features

The following features need to be modular, so that they can be deployed and scaled separately, they can evolve separately.

To keep the system maintainable while allowing future microservices transition, we merged 7 services into 5.

|  |  |
| --- | --- |
| **Features/Modules** | **Responsibilities** |
| User management | Registration, login, preferences, settings. |
| Identity and access management | Authentication and authorization, Access control, User management, groups |
| Booking | Seat reservations and booking |
| Payments | Payments processing, multiple payment provide support |
| Movie Management | Movie/show schedules, seat availability, eatables inventory |
| Notification | Movie tickets, offers, promotions, Booking confirmation and cancellation. Emails, SMS |
| Reporting & Analytics | Business analytics, performance tracking |

4. User Interface

We need different types of user interface based on the type of user base and the level of security and preferences.

|  |  |  |
| --- | --- | --- |
| **UI stack** | **User base** | **Tech stack** |
| Web based UI | Movie goers- preferring browser-based application | Angular / react |
| Mobile App | Movie goes - preferring low latency and custom features | .Net MAUI |
| Desktop app | Movie Admins - preferring view to secure information and admin capabilities to set up movie details | .Net MAUI |

5. Backend Services

The backend services need to be modular, independently deployable and scalable and should be able to communicate with one another

|  |  |  |
| --- | --- | --- |
| **Service** | **Responsibilities** | **Tech stack** |
| User and Identity service | Registration, login, preferences, settings. Authentication and authorization | .net 8 |
| Booking and Payment service | Seat reservations and booking. Payment processing, multiple payment provide support | .net 8 |
| **Event & Inventory Service** | Emails, SMS, push notifications | .net 8 |
| Notification Service | Payment processing, multiple payment provide support | .net 8 or node js |
| **Reporting & Analytics Service** | Business analytics, performance tracking | .net 8 |

## 4. Data Storage

A hybrid storage approach for structured, semi-structured, and unstructured data with multi-tenancy considerations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Data Type** | **Recommended Storage** | **Multi-Tenancy Strategy** |
| Users, bookings, payments | Structured Data | SQL Database (PostgreSQL, SQL Server) | Schema-per-tenant or shared DB with tenant column |
| Movie metadata, show details, user preferences | Semi-Structured Data | NoSQL Document DB (MongoDB) | Partition per tenant |
| Movie poster, trailers, promotional content | Unstructured Data | Object Storage (File Server) | Tenant-specific storage buckets |
| Live seat availability, active bookings | Real-Time Data | In-Memory Cache (Redis) | Per-tenant caching where necessary |
| Payment transactions, user activity logs | Transactional Logs | Event Store (Kafka, Append-Only SQL Table) | Tenant-tagged log entries |
| Historical trends, revenue reports | Analytics & Reporting | Data Warehouse (SQL server column store, Azure synapse) | Tenant-aware partitioning |
| Customer Behavior analysis, recommendations | Big Data & AI | Data Lake (AWS S3, Azure Data Lake) | Tagged datasets for multi-tenancy |

## 5. Communication / Integration styles

We need to choose the appropriate communication model and style, based on the type of components and need.

|  |  |  |
| --- | --- | --- |
| **Actors involved** | **Style** | **Benefits** |
| User to Service | Rest API | Request-response, Sync |
| User to Service | GraphQL | Request-response, query based, Sync |
| Service to user | Websockets | Push, bidirectional. |
| Service to Service | Grpc | Request-response, Sync |
| Service to Service | AMQP, MQTT | Real-time, Async |
| Database to Datawarehouse | Change data capture | Real time, Async |
| Database to Datawarehouse | ETL based | Batch |

## 6. Technology stack summary

| **Technology Layer** | **Choice** | **Comments** |
| --- | --- | --- |
| UI | .NET MAUI (Mobile & Desktop), Angular/React (Web) | .NET MAUI is best for cross-platform mobile/desktop apps on Windows, and Angular/React are industry standards for responsive web apps. |
| API Gateway | Azure APIM (Self-hosted) / Kong | Azure APIM provides deep integration with Microsoft ecosystems, and Kong is a strong alternative for flexibility. |
| Backend | .NET 8 for all backend, Node.js for notifications | .NET 8 offers high performance, strong Windows support, and long-term stability. Node.js is ideal for handling real-time notifications. |
| Message Broker | RabbitMQ | A robust message broker that runs well on Windows and supports event-driven microservices. |
| Identity Management | Active Directory (AD) | Best choice for Windows environments, supports LDAP, Kerberos, and AD FS for SSO. |
| Cache | Redis | High-performance in-memory caching, supports distributed caching and runs on Windows via WSL or containers. |
| SQL Database | SQL Server / PostgreSQL | SQL Server is optimized for Windows and works well with SSIS, Azure Synapse, and Power BI. PostgreSQL offers a strong alternative for open-source support. |
| NoSQL Database | MongoDB | Document-based NoSQL store, supports Windows installations and change data capture (CDC) for event-driven architectures. |
| File Storage | Windows File Server, DFS, NFS | Windows File Server supports NTFS-based storage, DFS enables distributed file systems, and NFS is useful for compatibility. |
| Event Store | Kafka | Industry-standard event store, supports real-time streaming, works well on Windows using Kafka for Windows or WSL. |
| Data Warehouse | SQL Server Columnstore, Azure Synapse | SQL Server Columnstore is best for analytical workloads, and Azure Synapse offers cloud-based big data analytics. |
| ETL (Extract, Transform, Load) | SSIS, Azure Data Factory | SSIS is a Windows-native ETL tool, while Azure Data Factory is great for cloud/hybrid workflows. |
| Analytical Tools | Power BI, Tableau, SSRS | Power BI is best for Microsoft integration, Tableau is strong for visual analytics, and SSRS supports report generation. |

**Security and Compliance**

* **Identity & Access Management (IAM)**: OAuth, JWT, Multi-Factor Authentication (MFA)
* **Data Protection**: Encryption at rest and in transit
* **Compliance with Regional Laws**: GDPR, CCPA, PCI-DSS for payments
* **Audit Logging & Monitoring** for threat detection