



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment 4

**Student Name:** Umang Kaushik

**Branch:** BE CSE

**Semester:** 6th

**Subject Name:** SD

**UID:** 23BCS10712

**Section/Group:** KRG-1-B

**Date of Performance:** 2<sup>nd</sup> Feb

**Subject Code:** 23CSH-314

**1. Aim:** To design an video-streaming platform (OTT - Netflix, Prime, Hotstar)

### **2. Objectives:**

- Low Latency: Ensure "instant-start" playback and minimize buffering.
- High Availability: Maintain 99.99% uptime even during peak traffic (e.g., major sports events).
- Device Agnosticism: Provide a seamless experience across mobile, web, and smart TVs.
- Personalization: Leverage data to offer relevant content recommendations.

### **3. Procedure:**

- The workflow follows a standard pipeline from the studio to the user's screen:
- Content Ingestion: Raw video files are uploaded to the server.
- Transcoding: The video is compressed and encoded into multiple formats (MP4, WebM) and resolutions (1080p, 720p, etc.).
- Chunking: The video is broken into small 2-10 second segments for Adaptive Bitrate Streaming (ABR).
- Distribution: These segments are cached across a Content Delivery Network (CDN) to bring the data closer to the user.

- Consumption: The client-side player selects the best chunk based on current internet speed.

#### **4. Functional Requirements:**

- User Management: Registration, login, and multi-profile support.
- Content Catalog: A searchable database of movies and shows.
- Video Player: Support for subtitles, multiple audio tracks, and playback speed control
- Watchlist & History: Tracking progress across devices.
- Subscription Engine: Integration with payment gateways for billing.

#### **5. Non-Functional Requirements:**

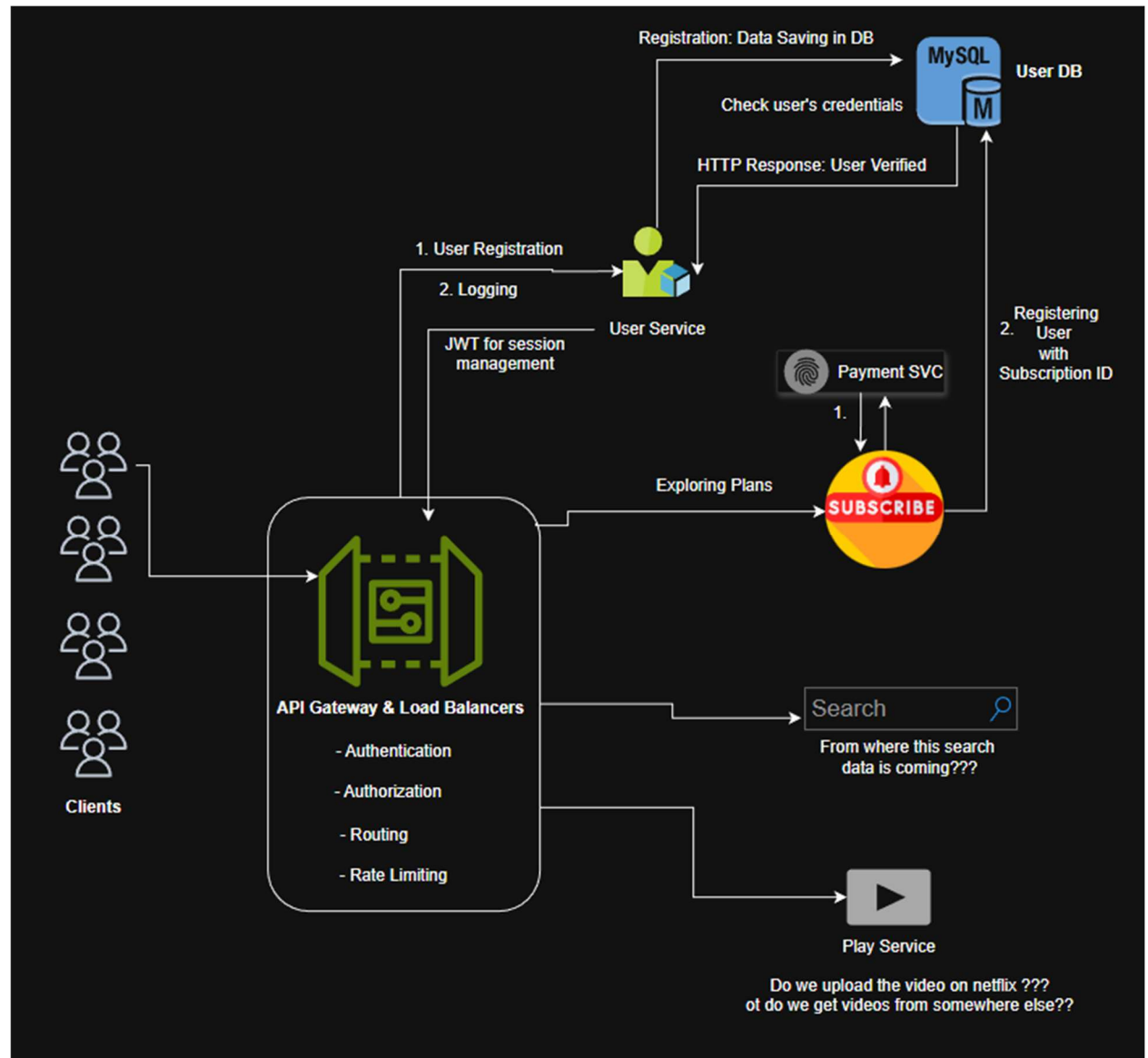
- Scalability: Ability to scale horizontally to handle millions of concurrent users.
- Reliability: No single point of failure; data must be replicated across regions.
- Security: Implementation of DRM (Digital Rights Management) to prevent piracy.
- Performance: Minimal "Start-up Delay" and smooth transitions during network fluctuations.

#### **6. Outcome:**

A robust, distributed system blueprint that ensures users can binge-watch their favorite shows without the dreaded "spinning circle," regardless of their location or device.

#### **7. Required System Design:**

## HLD:



## KAFKA:

