International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 11 | Nov - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Spring Boot MVC Video Live Streaming on Web server With Nginx RTMP Server and OBS Studio

Gopal Nilkanth Ghuge [1] , Dr.Sachin A Vyavhare [2] , Prof. Sapna R Tayde[3] ,

[1]*Gopal Nilkanth Ghuge ME(csit) & Sanmati engineering Collage Washim*
[2]Dr. Sachin A Vyavhare & Sanmati engineering Collage Washim

[3]Prof. Sapna R Tayde & Sanmati engineering Collage Washim

Abstract - This paper presents a web-based live video streaming framework developed using **Spring Boot MVC**, **NGINX RTMP server**, and **OBS (Open Broadcaster Software)**. The goal is to create a stable, flexible, and efficient system for real-time broadcasting over the web. OBS captures and encodes the live video stream, which is then transmitted to the NGINX RTMP server. The server handles the streaming process and distributes the video to connected web clients using HLS (HTTP Live Streaming).

Spring Boot MVC acts as the backend control layer, managing user sessions, stream details, and playback integration. The system provides a reliable way to broadcast and view live content with minimal delay. This architecture can be used for online classes, live events, tutorials, and remote conferences where performance and simplicity are equally important.

1.INTRODUCTION

In recent years, live video streaming has become an essential part of online communication and media sharing. Platforms like YouTube, Twitch, and Facebook Live demonstrate the demand for real-time content delivery. However, building a custom streaming platform requires efficient backend processing, stable media handling, and a smooth user interface.

The proposed system combines three open-source technologies to achieve this goal:

1. **OBS Studio** – captures live audio and video from the user and sends it using the RTMP protocol.
2. **NGINX RTMP Server** – receives, processes, and redistributes the live stream to multiple web clients.
3. **Spring Boot MVC** – serves as the web application layer that manages viewers, stream data, and playback on the browser.

This combination creates a simple yet powerful framework for live video delivery. It eliminates the need for expensive cloud-based streaming solutions and provides flexibility for developers to extend the system with features like authentication, analytics, and recording. The design ensures **low latency**, **scalability**, and **high reliability**, making it suitable for academic, professional, and entertainment use.

2. Body of Paper

System Architecture

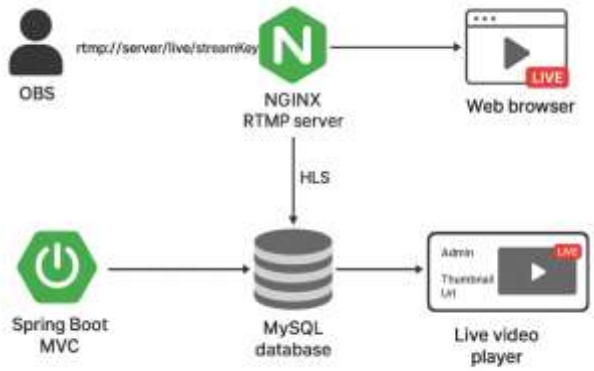

Figure 1: System architecture of the live streaming platform.

Working of the System

The system is designed to provide smooth and real-time live video streaming by integrating three key components: **OBS Studio**, **NGINX RTMP server**, and **Spring Boot MVC web application**.
Each component performs a specific role in capturing, transmitting, managing, and displaying the live stream.

---

1. OBS Studio (Broadcaster Side)

- The broadcaster uses **OBS Studio** to capture video and audio from the local camera and microphone.
- OBS compresses and encodes the captured video stream using formats like H.264 for video and AAC for audio.
- The encoded stream is then sent to the **NGINX RTMP server** through the **RTMP (Real-Time Messaging Protocol)**.
- Example RTMP URL:
- rtmp://localhost/live/streamkey
- OBS continuously sends the live data packets to the NGINX server for distribution.

2. NGINX RTMP Server (Streaming Layer)

- The **NGINX RTMP module** acts as the media server that receives the live stream from OBS.
- It stores the incoming stream temporarily in memory and converts it into **HLS (HTTP Live Streaming)** segments for browser compatibility.
- These HLS files (.m3u8 and .ts chunks) are served over HTTP, which allows smooth playback on web browsers and mobile devices.
- The server can handle multiple streams simultaneously and distribute them to multiple viewers without performance loss.
- NGINX also manages buffering, caching, and bitrate adjustments to ensure continuous playback even under variable network conditions.

---

3. Spring Boot MVC Application (Web Interface & Control Layer)

- The **Spring Boot MVC** web application acts as the user-facing platform.
- It provides:
  - **Admin interface** for creating or managing live stream sessions.
  - **User interface** for viewing live video streams directly through the browser.
- The backend stores information about each live stream such as:
  - Stream key, title, description, thumbnail, and status (live/offline).
- The application fetches and displays the HLS stream URL generated by NGINX using <video> tags or JavaScript HLS players.
- It also tracks **viewer count**, **stream duration**, and **connection status** in real time using **WebSocket or REST APIs**.

---

4. End-User Side (Viewer Experience)

- Viewers access the live stream through a web browser using the Spring Boot MVC web interface.
- The browser requests the .m3u8 playlist from the NGINX server, which is then played using an HTML5 video player.
- The video loads automatically and plays continuously with minimal buffering or delay.
- If the broadcaster stops or reconnects, the Spring Boot layer updates the stream status instantly to maintain a smooth user experience.

3. CONCLUSIONS

The integration of **Spring Boot MVC**, **NGINX RTMP**, and **OBS Studio** provides a complete, flexible, and efficient solution for live video streaming over the web. This system successfully combines the power of a robust backend framework, a lightweight streaming server, and an open-source broadcasting tool to deliver real-time, low-latency video to viewers.

By using **OBS**, live content can be easily captured and transmitted. The **NGINX RTMP server** efficiently handles the distribution and conversion of streams into browser-compatible formats like HLS, while **Spring Boot MVC** manages user interaction, session control, and playback interfaces. This modular design ensures scalability, allowing multiple broadcasters and viewers to operate simultaneously without system degradation.

ACKNOWLEDGEMENT

The heading should be treated as a 3rd level heading and should not be assigned a number.

REFERENCES

- **Pivotal Software, Inc.** — *Spring Boot Reference Documentation*, Version 3.x, Available at: https://spring.io/projects/spring-boot

- **NGINX, Inc.** — *NGINX RTMP Module Documentation*, Available at: https://github.com/arut/nginx-rtmp-module

- **OBS Project** — *Open Broadcaster Software (OBS Studio) Documentation*, Available at: https://obsproject.com/docs