iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** I **Month of publication:** January 2025

DOI: https://doi.org/10.22214/ijraset.2025.66742

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

Stream Rider: A Developer's Perspective on Live Streaming

Dhanraj Sardana[1], Dr. Pinki Nayak[2], Dr. Jyoti Parashar[3]
[1]*ADGIPS*
[2]*Associate Professor, ADGIPS*
[3]*Associate Professor, ADGIPS*

Abstract: ***This paper presents the development and analysis of a live streaming application designed to provide a seamless, low-latency streaming experience. The project leverages modern web development frameworks and streaming technologies, including React for the frontend, Node.js for the backend, and MongoDB for efficient database management. To address challenges like scalability and latency reduction, the application integrates RTMP and HLS protocols with a robust architecture powered by cloud services and a Content Delivery Network (CDN).***
The system delivers real-time video streaming with interactive features such as live chat, catering to a wide range of use cases, including gaming, entertainment, educational, and professional live events. Key challenges, such as ensuring scalability, reducing latency, and maintaining security, were addressed through efficient design and risk mitigation strategies.
The proposed solution demonstrates significant improvements in streaming quality, system scalability, and user experience, meeting both user expectations and industry standards. By offering insights into building scalable and reliable live streaming platforms, this project contributes to the growing demand for high-quality real-time communication solutions. Preliminary performance results further validate the application's ability to deliver low-latency, high-quality streaming while ensuring optimal resource utilization.
Keywords: CDN, HLS, RTMP, React, Node.js, MongoDB, Zegocloud

I. INTRODUCTION

The rapid growth in live-streamed content consumption across sectors such as entertainment, education, gaming, and corporate communication has underscored the increasing demand for reliable and high- quality streaming platforms. However, developing such platforms presents significant challenges, including high latency, suboptimal video quality, scalability limitations, and the need to manage large numbers of concurrent viewers. Furthermore, ensuring seamless streaming experiences across diverse devices and network conditions compounds these challenges.

A. Objective

This research focuses on the development of a dynamic live streaming application Delivery Network (CDN) to deliver low- latency, high-quality streaming. designed to facilitate real-time video streaming with a focus on low latency and interactive features. The platform is inspired by existing solutions like Twitch but seeks to address specific shortcomings while enhancing user satisfaction.
The primary objective is to enable content creators to deliver high-quality live streams effortlessly while engaging their audiences through real-time features such as live chat, reactions, and screen sharing. The application supports multiple use cases, including gaming, entertainment, educational content, and professional events. The system is designed with a user-friendly interface and a robust backend architecture to ensure reliability and ease of use for both streamers and viewers

B. Limitations of Existing Platforms

Existing platforms such as Twitch provide comprehensive features but are constrained by several limitations:
1) Platform Dependency: Streamers are highly reliant on a single platform for their content delivery.
2) Limited Customization: Streamers have minimal control over branding and personalization.
3) Complex Onboarding: The initial setup process is often intricate, posing barriers for new users.
These limitations necessitate a solution that prioritizes user flexibility and control while maintaining high performance.

©IJRASET: All Rights are Reserved

C. Project Proposition

This research proposes a scalable and adaptive live streaming platform leveraging modern web and streaming technologies. The system integrates RTMP and HLS protocols with a scalable architecture supported by cloud services and a Content
These limitations highlight the need for a more adaptable and efficient solution, especially for developers and organizations.
The application utilizes a contemporary technology stack, including React for efficient frontend development, a UI Kit for consistent design, and CDN for reliable content delivery. Secure user authentication is implemented using JWT, while OBS integration facilitates effective stream management. By supporting adaptive streaming protocols such as HLS and RTMP, the platform ensures optimal performance across devices and network conditions. This research not only addresses the limitations of existing platforms but also contributes to the broader understanding of scalable, user- centric live streaming systems

II. RELATED WORK

A. Existing Solutions

Popular live streaming platforms, such as Twitch and YouTube Live, have become industry leaders, offering robust features and scalable architectures. These platforms effectively cater to a wide range of use cases, including gaming, entertainment, education, and professional events. Their implementations typically rely on centralized servers and proprietary technologies to handle large-scale streaming demands.

However, despite their success, these platforms exhibit several limitations:

1) High Latency: The delay between the streamer and the viewer remains a persistent issue, impacting real- time interaction and engagement.
2) Server Dependency: Centralized server architectures can lead to bottlenecks under high traffic loads, limiting scalability.
3) Limited Flexibility: These platforms provide minimal customization options for streamers, constraining branding and feature adaptability.
4) Secure Data Management: MongoDB's NoSQL architecture supports flexible data models, seeking to build tailored live streaming platforms.

B. Research Gap

While proprietary platforms address scalability and performance challenges, there is a lack of open-source solutions that effectively integrate advanced streaming technologies like HLS, RTMP, and CDN. Existing open-source frameworks often fail to provide a holistic approach that combines low-latency streaming, real-time interaction, and robust content delivery.
Moreover, the absence of a unified architecture that seamlessly integrates frontend, backend, and streaming protocols has created a gap in the ecosystem. This gap hinders the ability of smaller organizations and developers to create custom, high- performance live streaming applications.

C. Motivation

This research is motivated by the potential to bridge the aforementioned gap through the integration of modern technologies. The combination of **React** for the frontend, **Node.js** for the backend, and **MongoDB** for database management offers a lightweight and scalable architecture.
These technologies are well-suited for:

1) Real-Time Updates: React's component-based design ensures smooth and responsive user interfaces, enhancing viewer experience.
2) Scalability: Node.js, with its event- driven architecture, enables efficient handling of concurrent requests, making it ideal for high-traffic streaming platforms.

Following is the detailed description of Fig 3.1 accommodating diverse user interactions like chat messages and viewer analytics.
Furthermore, the integration of HLS and RTMP protocols, supported by a Content Delivery Network (CDN), ensures high- quality, low-latency streaming across multiple devices and varying network conditions. By addressing the limitations of existing platforms and incorporating advanced streaming techniques, this project aims to contribute a scalable, flexible, and user-centric solution to the live streaming domain.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

III. SYSTEM ARCHITECTURE

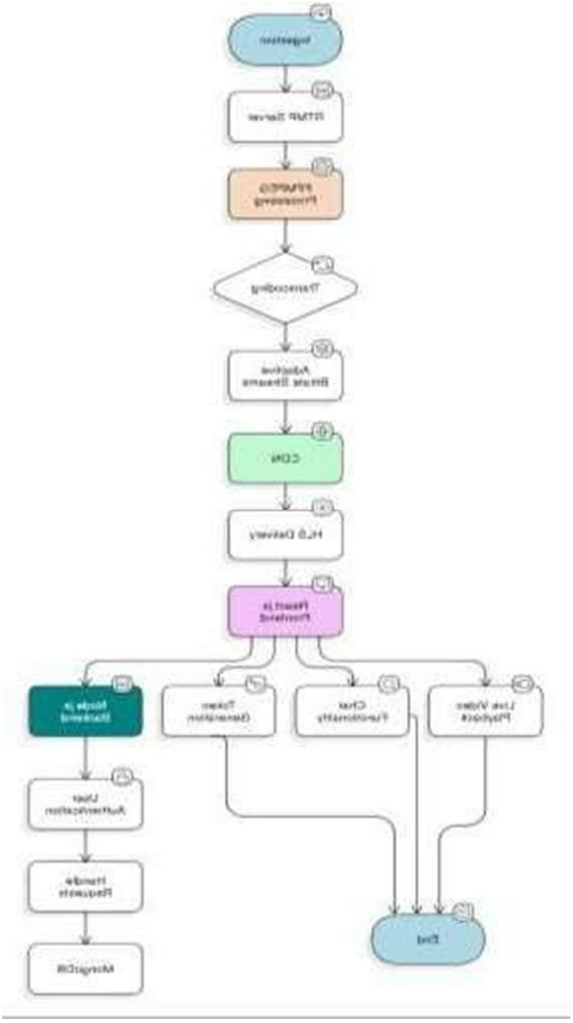


Fig 3.1 System Flow

1) *Ingestion:* This step marks the beginning of the video streaming process. The RTMP Server receives live video streams from the streamer's device using the Real- Time Messaging Protocol (RTMP) and OBS server.
2) *FFMPEG Processing:* ffmpeg is used to process and encode the raw video stream. It handles video transcoding converting the stream into different formats and bitrates. Example 1080 px encoding ffmpeg -i some_fun_video_name.mp4- profile:v baseline-level3.0-s 1920x1080-start_number 0- hls_time 10-hls_list_size 0-f hls./media /some_fun_video_name /hls/ 1080_out.m3u8
3) *Transcoding:* Converts the video stream into multiple quality levels i.e. adaptive bitrate streams to accommodate users with varying network conditions. Example ffmpeg-I "test.mov" -vcodec h264 -b:v 20971520-s 1998x1080-r 24-acodec aac -b:a 327780 -ar 95000 "test.mp4"
4) *Adaptive Bitrate Streams:* Enables smooth playback by dynamically adjusting the video quality based on the user's internet speed.
5) *Content Delivery Network (CDN):* The transcoded streams are distributed globally using a CDN for low-latency and high-performance video delivery.
6) *HLS Delivery:* HTTP Live Streaming (HLS) delivers the video segments to users in a format compatible with various devices and platforms.
7) *React.js (Frontend):* The user interface is built with React.js, which allows seamless interaction with the platform. It Includes features like live video playback, chat, and user controls.

D. Frontend Features

1) Live Video Playback: Users watch the live stream in real-time.
2) Chat Functionality: Enables interaction between users and streamers via live chat.
3) Token Generation: Manages secure access to streams.
4) Node.js (Backend): Handles backend logic, including:
5) User Authentication: Ensures secure access.
6) Token Generation: Generates tokens for secure streaming sessions.
7) Handle Requests: Manages various API requests.
8) Database (MongoDB): Stores user and streaming data

IV. CHALLENGES AND LIMITATIONS

A. Technical Challenges

The development of a live streaming platform involves addressing various technical complexities. Key challenges faced during this research include:

1) Integration of HLS and RTMP Protocols: Ensuring seamless integration of HLS (HTTP Live Streaming) and RTMP (Real-Time Messaging Protocol) posed significant challenges. These protocols, while essential for delivering high-quality and low latency streams, require meticulous synchronization and optimization to maintain consistent performance across devices and varying network conditions.
2) Performance Issues with Zegocloud API and CDN: The reliance on Zegocloud for real-time communication and on CDNs for content delivery introduced challenges during periods of high traffic. Peak load conditions occasionally resulted in latency spikes or reduced quality of service, necessitating careful load balancing and resource allocation strategies.

B. Limitations

Despite the robustness of the proposed solution, certain limitations exist due to the scope and focus of this project:

1) Restricted Feature Scope: The project emphasizes core functionalities such as live video streaming, real-time chat, and content delivery. Advanced features like AI- driven recommendations, dynamic bitrate adjustment, or personalized content curation are beyond the current scope and could be explored in future research.
2) Regional Constraints: The application's scalability and performance heavily depend on the availability and efficiency of CDN nodes in specific regions. Limited CDN coverage in remote areas may impact the quality of service and overall user experience.
3) High Dependency on a Single Platform: Streaming platforms like Twitch and YouTube Live centralize control, leaving creators reliant on their infrastructure for hosting, audience reach, and monetization. This dependency limits creators' autonomy, restricts revenue diversification, and reduces access to audience analytics, making streamers vulnerable to policy changes, account issues, or platform outages.
4) Limited Control Over Customization and Branding: Streamers have minimal flexibility in personalizing their streams, as platforms dictate the user interface and available features. This lack of customization prevents creators from establishing a unique brand identity, limiting their ability to differentiate themselves in a competitive market or tailor the experience to audience needs.
5) Higher Entry Barriers for New Users: The onboarding process for new streamers is often complex, involving technical configurations, stringent verification processes, and high thresholds for monetization eligibility. Limited support and scattered resources further discourage non-technical users, creating accessibility challenges for new entrants in the streaming space.

V. CONCLUSION

A. Summary

This study presents a comprehensive framework for building a scalable, low- latency live streaming application leveraging modern technologies such as HLS, RTMP, CDN, React, Node.js, and MongoDB. By addressing the limitations of existing platforms, the proposed system ensures seamless content delivery, robust real-time interaction features, and enhanced scalability. The integration of adaptive streaming protocols, coupled with efficient cloud-based architecture, demonstrates, significant improvements in system performance, including reduced latency improved video quality, and reliable handling of concurrent users.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

B. Impact

The proposed solution contributes to the field of live streaming by providing a modular and extensible architecture that balances performance, scalability, and user experience. By leveraging open-source tools and modern frameworks, this work facilitates the creation of customizable live streaming platforms tailored to diverse use cases. Additionally, the emphasis on low- latency streaming and real-time engagement addresses critical challenges in the industry, making this framework relevant for applications in education, gaming, corporate communication, and beyond.

C. Call to Action

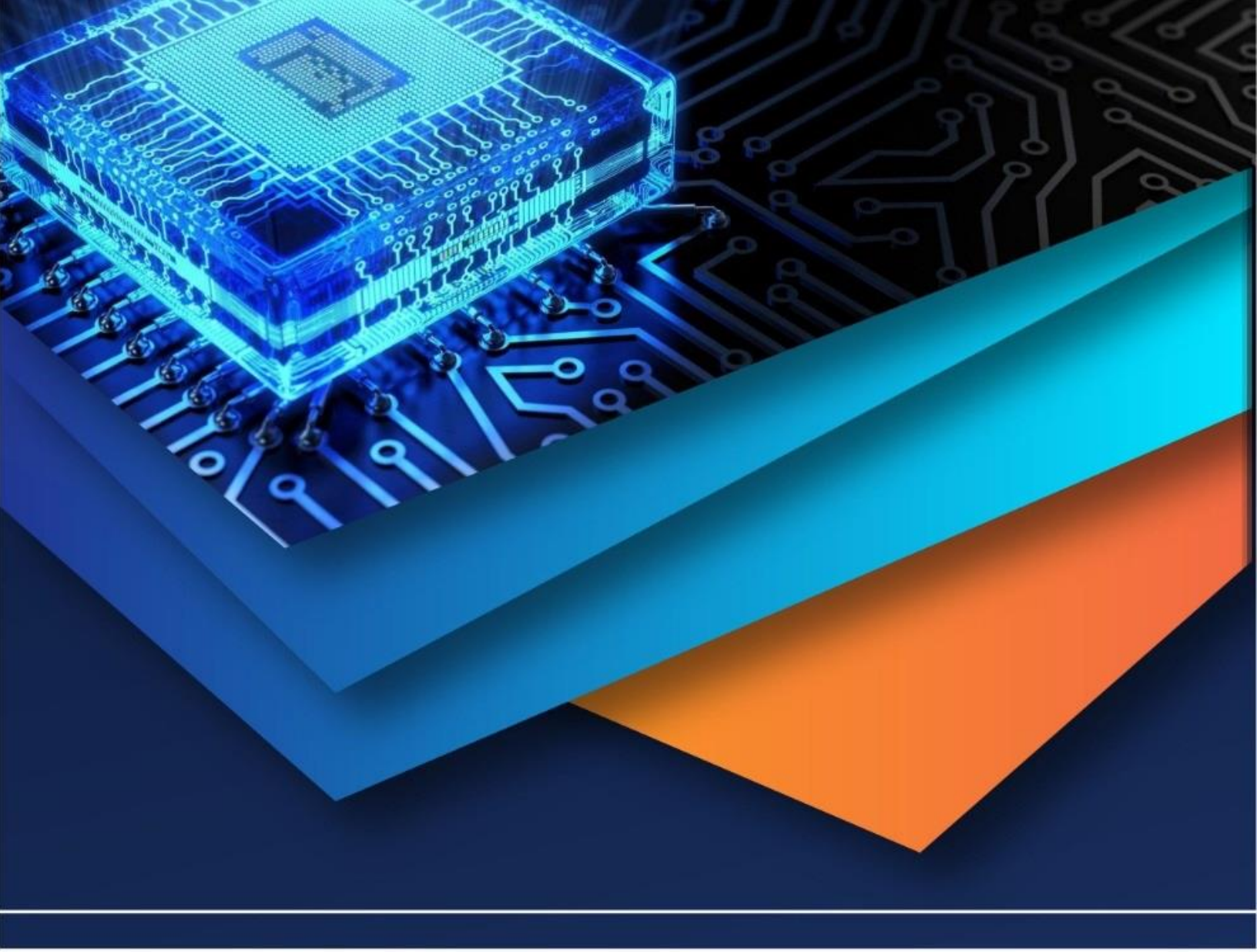
This research underscores the potential of integrating advanced streaming technologies with scalable web development practices. Developers and researchers are encouraged to adopt and further refine this framework to explore emerging trends such as AI-driven analytics, enhanced content personalization, and advanced security mechanisms. By building upon this foundation, future systems can achieve greater efficiency, adaptability, and innovation in the dynamic domain of live streaming.

VI. ACKNOWLEDGEMENT

The note starts with thanks to Almighty who actually created this piece of work and helped us when things were not easy for us. We are very grateful and indebted to my supervisor Dr. Pinki Nayak ma'am who immensely helped and rendered her valuable advice, precious time, knowledge and relevant information regarding the collection of material. She has been a major source of inspiration throughout the research paper as she not only guided us but also encouraged us to solve problems that arose during the research. Her guidance and suggestions about the research have really enlightened us. It has been a great help to support to have her around. And finally, we would like to mention appreciation to our parents and friends who have been instrumental throughout this period by providing unrelenting encouragement.

REFERENCES

[1] Kaplan, A. M. & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. Business Horizons, 53, 59 - 68.
[2] Kim, J. (2021). Advertising in the Metaverse: Research Agenda", Journal of Interactive Advertising, 21(3), 141-
[3] 144.
[4] Li, B., Hou, F., Guan, Z., & Chong, A. (2018) What drives people to purchase virtual gifts in live streaming? the mediating role of flow, PACIS 2018 Proceedings, 3434-3447.
[5] Li, R., Lu, Y., Ma, J., & Wang, W. (2020) Examining gifting behavior on live streaming platforms: An identity- based motivation model, Information & Management, 103406.
[6] Mikko V. J. Heikkinen Thomas, C. & Fabio, H. (2010), Value analysis of centralized and distributed communications and video streaming, Info, 12, 5, 42 – 58.
[7] Mueser, D. & Vlachos, P. (2018) Almost like being there? A conceptualisation of live-streaming theatre, International Journal of Event and Festival Management, 9, 183-203.
[8] Novak, C.C. (2012) Live versus recorded: exploring television sales presentations, ProQuest Dissertations Publishing, University of South Florida, 1515667.
[9] Scheibe, K., K. J. Fietkiewicz and W. G. Stock (2016) Information Behavior on Social Live Streaming Services, Journal of Information Science Theory & Practice, 4 (2), 6-20.
[10] Sjöblom, M., Törhönen, M., Hamari, J., & Macey, J. (2019) The ingredients of Twitch streaming: Affordances of game streams, Computers in Human Behavior, 92, 20 – 28.
[11] Sun, J., Shao, X., Li, X. Guo, Y. & Nie, K. (2019) How live streaming influences purchase intentions in social commerce: An IT affordance perspective, Electronic Commerce Research and Applications, 37, 100886.
[12] Tang, J. C., G. Venolia and K. M. Inkpen (2016) Meerkat and periscope: i stream, you stream, apps stream for live streams, In Proceedings of the 2016 Conference on Human Factors in Computing Systems, San Jose, California.
[13] X. Gao et al. How the live streaming commerce viewers process the persuasive message: An ELM perspective and the moderating effect of mindfulness
[14] Electronic Commerce Research and Applications (2021)
[15] Y. Guo et al. Way to success: Understanding top streamer's popularity and influence from the perspective of source characteristics
[16] Journal of Retailing and Consumer Services (2022)
[17] G. Hufnagel et al. Seeking the perfect price: Consumer responses to personalized price discrimination in e- commerce
[18] Journal of Business Research (2022)
[19] A Khan et al. Social commerce advertising avoidance and shopping cart abandonment: A fs/QCA analysis of German Consumers Journal of Retailing and Consumer Services (2022)
[20] K. Kang et al. The dynamic effect of interactivity on customer engagement behavior through tie strength: Evidence from live streaming commerce platforms International Journal of Information Management (2021)
[21] X. Ma et al. Do long-life customers pay more in pay-what-you-want pricing? Evidence from live streaming Journal of Business Research (2022)
[22] E.W. Mainardes et al. Antecedents and consequents of consumers not adopting e-commerce Journal of Retailing and Consumer Services (2020)
[23] M. Nouri-Harzvili et al. Optimizing the competitive service and pricing decisions of dual retailing channels: A combined coordination model Computers and Industrial Engineering (2022)
[24] H. Albanna et al. An integrated model for using social media applications in non-profit organizations International Journal of Information Management (2022)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)