

# Kappa: A Synchronized Video Viewer

Zhan Xiong Chin  
chinz@seas.upenn.edu  
Univ. of Pennsylvania  
Philadelphia, PA

David Liao  
liaod@seas.upenn.edu  
Univ. of Pennsylvania  
Philadelphia, PA

Yoo Jin Kim  
yoojkim@seas.upenn.edu  
Univ. of Pennsylvania  
Philadelphia, PA

**Abstract**—E-sports is a market that is rapidly becoming popular and widespread, yet often does not see high quality software solutions catering to it. Professional e-sports teams, whose members typically play remotely, do not have good ways to analyze or review their games together. Instead, they rely on ad-hoc solutions using a combination of existing software, leaving considerable room for improvement.

To address this need, we propose Kappa, a synchronized all-in-one video viewing platform that allows e-sports teams to view, annotate, and analyze video replays of games remotely. The main technical contribution is designing both a scalable backend that supports our custom syncing protocol as well as pioneering a frontend UI that solves existing complications that arise when integrating with other third party systems. We achieved the creation of a fluid and responsive UI that is intuitive and integrates well with Youtube. With scalability and cost in mind, the UI is designed to complement the backend by helping minimize server-side upstream communication. The backend architecture emphasizes flexibility and reliability, with communication latencies of less than 100ms to all clients under all loads using techniques that exploit differential payloads and minimize overhead communication costs.

## I. INTRODUCTION AND PROBLEM DEFINITION

E-sports is a rapidly growing market, with a burgeoning number of professional teams entering the field to compete for multi-million dollar prize pools in online and offline competitions. With so much at stake, these professional teams spend a considerable amount of their time practicing and attempting to improve their skills. For instance, a typical day for a professional player would be to spend the morning playing a number of matches, then reviewing their video replays as a team in the afternoon. Similar to athletes in other sports, any improvements to the training process can yield a significant impact on their success rates.

However, despite the importance of such training and analysis time, the processes that teams use for these are often woefully inadequate. Unlike a conventional professional sporting team, e-sports teams are often decentralized in nature, with their members potentially being in completely different cities or countries for most of the year. Thus, any shared analysis time tends to rely on ad-hoc solutions. For example, a team might communicate via multiple screenshots edited with Microsoft Paint, or by manually communicating to each other the relevant video timings in a video replay. Such ad-hoc solutions slow down the analysis process and significantly increase the chance of a miscommunication, while making it difficult to remember the details of a previous analysis session.

## II. SOLUTION OUTLINE

### A. Overview

Our proposed solution to this need is Kappa, an all-in-one web platform that aims to make collaborative video analysis sessions easy to run. Our platform allows videos to be loaded from Youtube, synchronizing their playback across multiple browsers. A shared canvas overlaid across the videos allows for annotations and other details to be easily communicated to the other users of the platform.

### B. Architecture

The main site of the platform is in charge of handling login and account registration processes. It is hosted using a custom web server written in Go. It also allows the user to create rooms (each room is a single shared video instance) and invite friends and teammates to join the relevant room.

The synchronized video viewer is in charge of tracking changes to the video's state (play/pause, volume, current playback time) and sending them to the sync server (described below). It is built in JavaScript and React on top of the YouTube IFrame Player. Making use of the YouTube player rather than writing our own player (which we did for earlier versions of the platform) enables users to more easily make use of existing video resources such as previously hosted YouTube videos. This not only leverages YouTube's high performance content delivery network across all users viewing the same video, but also makes it easy for teams to analyze video replays from other competitions or tournaments, which are frequently posted to YouTube.

The shared canvas provides a layer on top of the video which allows for annotations and drawings to be overlaid over the video. Any drawing or annotation made on the canvas is immediately synchronized with other clients via the sync server as well. It is also built in JavaScript and React.

The sync server is in charge of passing messages back and forth between the various clients to synchronize their video and canvas state. It is also written in Go and relies on WebSockets to provide a reliable and high-performance communication channel between itself and its clients.

### III. DESIGN DECISIONS

### IV. TECHNICAL DEPTH

### V. EVALUATION OF SOLUTION

### VI. IMPLICATIONS OF PROJECT

### VII. FUTURE WORK

### VIII. CONCLUSION

In conclusion, we have built an easy-to-use and scalable solution for a target demographic.

Furthermore, despite our focus on the gaming and e-sports market, our platform is fairly general in nature. We can foresee numerous use cases outside of e-sports, even with minimal tweaks to the system. For example, the platform could be easily adapted to education, by providing a shared virtual classroom setting with a teacher teaching via video stream while making use of the annotation feature to easily communicate information to students.

### ACKNOWLEDGMENT

We would like to thank our senior design advisor Professor Boon Thau Loo for the invaluable support and advice that he has provided us throughout this project.

### REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.