Kollani: A Distributed Tool for Real-Time Collaborative Reviews of 3D Assets

F. Andreussi, C. Hickstein and M. Minsel

RnDeep GmbH, Stuttgart, Germany

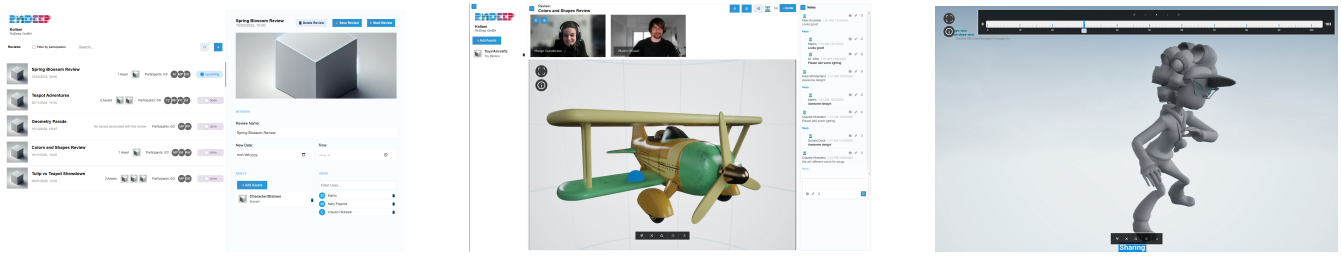

Figure 1: *Screenshots of the Kollani webapp, showcasing its features and layout. From left to right: the homepage, an open review with notes, a review of an animated scene (cropped in)*

Abstract

We present a novel web-based platform for reviewing 3D assets in a collaborative, interactive, and unified way. Currently, these reviews have to be carried out across several different platforms, while the application we developed integrates multiple tools (i.e. an interactive 3D environment, a video-conferencing feed, and a chat) in one coherent interface that can also be connected to popular DCCs, like Maya, in order to create a quick, accessible, and expressive way to discuss and review 3D models of any nature for any user, regardless of their own technical background and level of familiarity with modelling software.

CCS Concepts

• ***Human-centered computing*** → *Synchronous Editors;* • ***Applied computing*** → *Document management;* • ***Networks*** → *Network services;*

1. Introduction

Having witnessed the rise in popularity of remote working and the ever increasing demand for digital collaboration tools, we noticed that discussing 3D assets and scenes still required the use of multiple services not really designed for this purpose: one for the video-call, one for tickets and notes, and one for visualising the assets.
Our proposed solution for this issue is **Kollani**: a web-based platform that aims to provide a unified and coherent experience for reviewing 3D scenes among distributed teams and presenting proposals to remote clients. Its main innovative contribution are:

- The development of a 3D environment with **Unreal Engine** that is streamed to a web interface.
- The full support of animated assets.
- An annotation system, in which comments can be permanently pinned to specific mesh vertices even when the mesh is animated or transformed and embed the POV of the annotator.
- The example implementation of an integration of our software within a commercial DCC (i.e. Maya).

2. Functionality Overview

The review page integrates the video feeds from the participants up top, an editable list of all the assets displayed in the scene, and a comments panel, where the user can write new notes and reply to other user's messages.
The most interesting part of the interface is arguably, though, in the center of the screen, where an Unreal Engine viewport finds its place. This is a departure from similar previous works [NLHK15, DLJ15], which rlied on WebGL for the 3D rendering limiting the detail and complexity of the scene. Every participant has full access and autonomy in navigating and create notes in the scene; whenever the collective attention of the callers has to be focussed on a particular detail, the camera point-of view can be

DOI: 10.2312/egp.20261001

shared across all clients by a single user. Annotations get pinned on the asset's geometry and visualised as blue dots. By clicking on these dots, the camera snaps to the same position and orientation, in which the annotator was when they wrote it, and a pop-up reveals its contents.
In contrast with [NLHK15, DLJ15, GMM*14], Kollani was developed with animated assets in mind and provides the users with a timeline and playback commands. Moreover, the notes follow the mesh's animation and can be configured to be visible only in a specific frame range.
A further aspect that previous works did not consider, is the need for professionals to make substantial edits on the assets in a familiar environment. We, therefore, propose of a plug-in for Maya in which the assets can be imported, edited using its established interface, and sent to be viewed in Kollani in real-time. We believe this enables an agile, informal, and iterative workflow suited to all stages of a product's development.

3. Platform Architecture

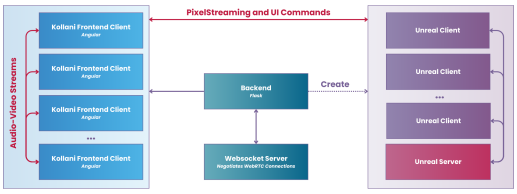

Figure 2: *Schematic representation of Kollani's architecture*

Kollani's **frontend** is built using **Angular** providing a user interface that is accessible across multiple devices and web browsers.
Its backend, on the other hand, is scripted in **Python** and makes extensive use of **Flask**, a lightweight framework well suited for rapid prototyping and modular service design. Flask provides the central logic of the system, exposing to the frontend the endpoints for creating reviews, managing assets as well as users, and storing annotations in the database, while also handling the negotiation of video rooms and signalling for WebRTC connections. It acts as a bridge between the web frontend and the Unreal clients, which it managesand coordinates, enabling scalability across multiple machines and ensuring a reliable foundation for both cloud-based and on-premises deployments.

3.1. Unreal Engine

The Unreal instances used to render the assets, run on their own dedicated process parallel to the Angular frontend, the Python backend, and the **Pixel Streaming** server. Pixel Streaming is an Unreal Engine feature that streams rendered frames and audio from the engine to a web browser in real time, enabling high-fidelity visualization without requiring local rendering power. These clients can be deployed, either on a physical machine or on AWS using **Terraform**, a managing tool that automates provisioning and scaling of cloud resources to support large numbers of simultaneous review sessions. This distributed setup allows scaling across multiple machines and ensures reliable operation in cloud environments.

3.2. WebRTC

Kollani's interactive features are powered by the WebRTC (Web Real Time Connections) communication protocol, which provides a reliable and open-source API to establish peer-to-peer and low-latency streaming of video and audio feeds, simplifying the infrastructure required for exchanging data. [JCBB25] Moreover, its ability to handle complex JSON structures, makes the messaging structure convenient to understand, expand, and manage.
Due to its versatility, not only mediates the videocalls, but also all the interactions between the users and Unreal. The connections are managed by the backend, which establishes the necessary signalling processes, while the Pixel Streaming handles both the delivery of the game engine's output into the webpage and the user interactions between frontend and Unreal.

4. Future Work

Additonal planned features are the possibility of showing or hiding specific geometries in the scene, colour coding notes, and enabling multimodal annotations similarly to [GMM*14] like audio and visual recordings or sketches on the 3D scene. These inputs are already partially supported thanks to the possibility of attaching any file to a comment although the UX can be greatly improved.
Also, drawing inspiration from [SHS*18, DLJ15], we would like to add limited scene editing tools to the Unreal client.Finally, we are considering investigating emerging user behaviours and quantify user satisfaction.

5. Acknowledgement

Kollani was developed in collaboration with Hochschule der Medien Stuttgart and Seru Film with the support of the Invest BW funding project finances by the Ministry for Business, Work and Tourism of Baden-Württemberg.

References

[DLJ15] DESPRAT C., LUGA H., JESSEL J.-P.: Hybrid client-server and p2p network for web-based collaborative 3d design. In *23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (June 2015). URL: http://wscg.zcu.cz/WSCG2015/CSRN-2501.pdf. 1, 2

[GMM*14] GUERREIRO J., MEDEIROS D., MENDES D., SOUSA M., JORGE J., RAPOSO A., SANTOS I.: Beyond Post-It: Structured Multimedia Annotations for Collaborative VEs. In *ICAT-EGVE 2014 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments* (2014), Nojima T., Reiners D., Staadt O., (Eds.), The Eurographics Association. doi:10.2312/ve.20141365. 2

[JCBB25] JENNINGS, CASTELLI, BOSTRÖM, BRUAROEY: *WebRTC: Real-Time Communication in Browsers*. Tech. rep., W3C WebRTC Working Group, March 2025. URL: https://www.w3.org/TR/webrtc/. 2

[NLHK15] NYAMSUREN P., LEE S.-H., HWANG H.-T., KIM T.-J.: A web-based collaborative framework for facilitating decision making on a 3d design developing process. In *Journal of Computational Design and Engineering* (February 2015). doi:10.1016/j.jcde.2015.02.001. 1, 2

[SHS*18] SPIELMANN, HELZLE, SCHUSTER, TROTTNOW, GÖTZ, ROHR: Vpet: virtual production editing tools. In *ACM SIGGRAPH 2018 Emerging Technologies* (2018). URL: https://doi.org/10.1145/3214907.3233760. 2