

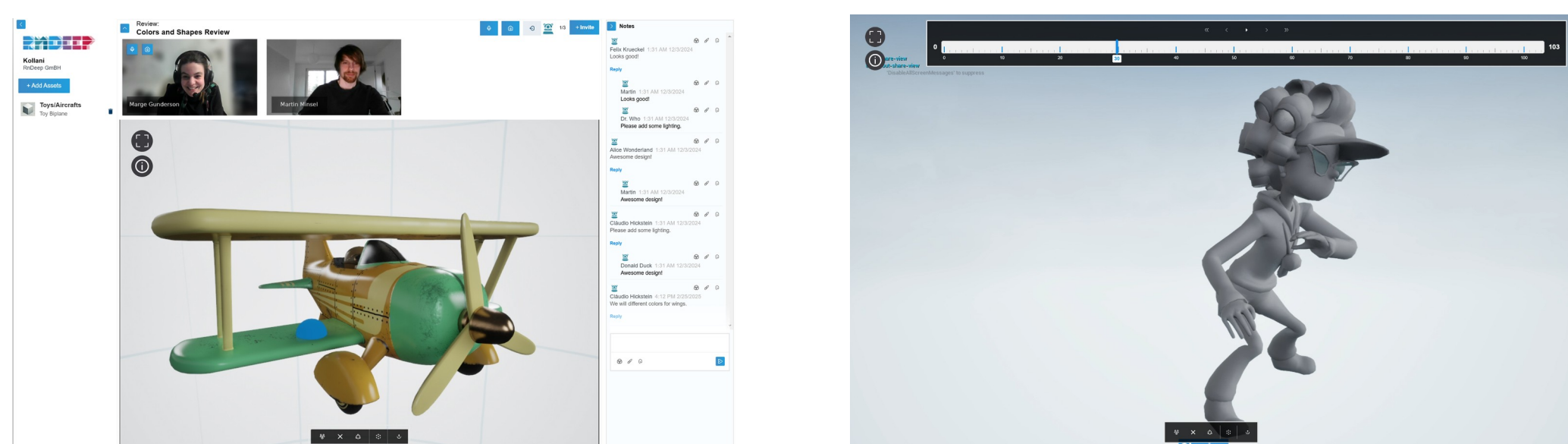
## KOLLANI: A DISTRIBUTED TOOL FOR REAL-TIME COLLABORATIVE REVIEWS OF 3D ASSETS

Francesco Andreussi, Claudio Hickstein and Martin Minsel  
RnDeep GmbH, Stuttgart

### PROBLEM

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.



### RELATED WORK

Our work stems from our collaboration and contacts with local companies and institutions, like the Filmakademie Baden-Württemberg[4] and especially SeRu Films and Hochschule der Medien in Stuttgart, which are our partners in this project.

However, Kollani also shares some concepts and technologies with previous works, in particular [1] and [2]. The main innovation we introduce is the use of Unreal Engine instead of WebGL to render the 3D scene. This creates some communication overhead but it allows us to handle more complex environments and increase the quality of the rendering regardless of the users' hardware limitations.

### OVERVIEW

Within a review, every participant has full access and autonomy in navigating and create notes in the scene; although, the camera point-of view can also be shared across all clients by a single user. Annotations get pinned on the asset's geometry and visualised as blue dots and can be viewed from the same position and orientation as the annotator.

Kollani was developed with animated assets in mind and provides the users with a timeline and playback commands, while the notes will track the mesh after any animation or transformation.

Finally we 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 to enable an agile, informal, and iterative workflow that we believe is suited to all stages of a product's development.

### METHODOLOGY

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 manages and coordinates, enabling scalability across multiple machines and ensuring a reliable foundation for both cloud-based and on-premises deployments.

The **Unreal** clients used to render the assets, run on their own dedicated process in parallel to an Unreal server, which coordinate dataflows and synchronisation between clients, 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 visualisation without requiring local rendering power. These instances can be deployed either on a physical machine or in the cloud. This distributed setup allows scaling across multiple machines and ensures reliable operation in cloud environments.

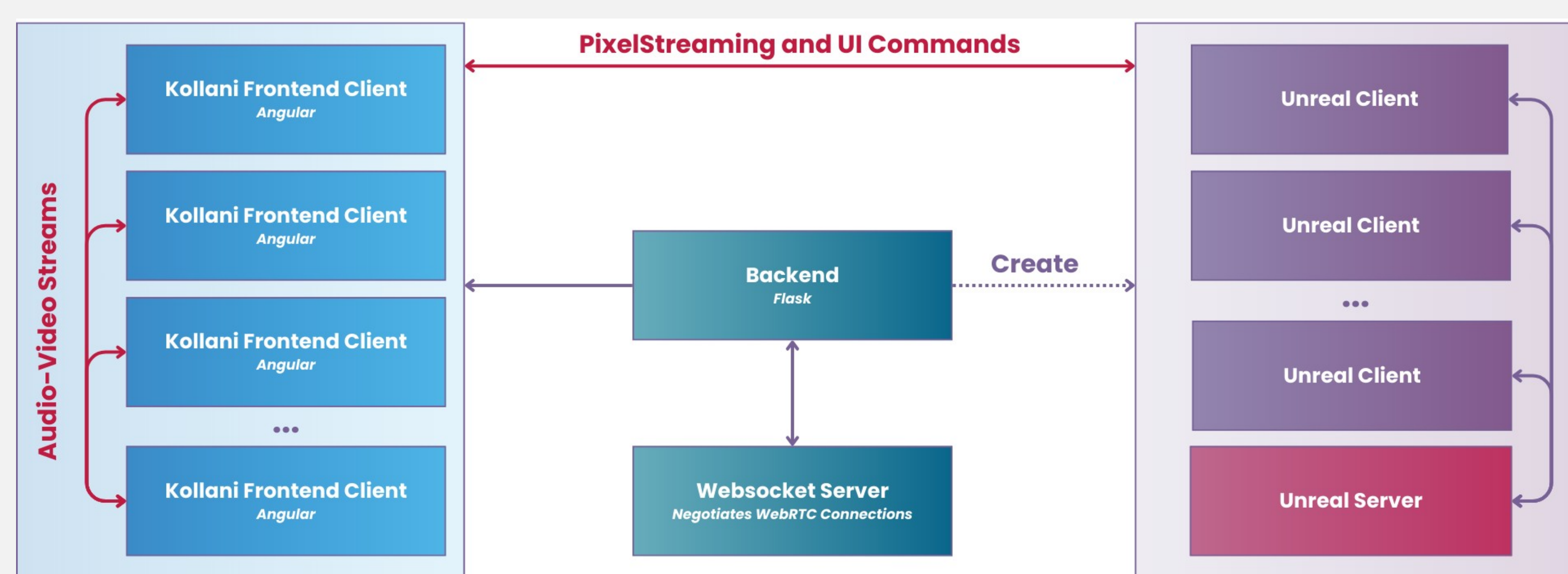
Kollani's interactive features are powered by the **WebRTC** (Web Real Time Connections) communication protocol, which provides a reliable and open-source API to establish peer-to-peer and low-latency streaming of video and audio feeds, simplifying the infrastructure required for exchanging data. [3] Moreover, its ability to handle complex JSON structures, makes the messaging structure convenient to understand, expand, and manage.

Due to its versatility, it not only mediates the video-calls, 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.

### RESULTS

As we enter the first phase of public testing, we believe we have created a viable web service that enables robust peer-to-peer connections between web clients, which are also all connected to Unreal instances. The users are able to perform basic interactions within the Unreal environment like moving around, leaving and consulting annotations, sharing their camera, and scrubbing animations. The Maya plug-in we present allows for a streamlined workflow, in which transforms and edits applied to the asset are immediately visible in the web app, while, at the same time, delivering a simpler, more accessible interface for non-technical users. We have been able to test our infrastructure on real-world assets provided by some of our clients and we were able to work with them without any real degradation of the rendering performance.

Already planned features are the possibility of showing or hiding specific geometries in the scene, colour coding notes, and enabling multimodal annotations similarly to [2] like audio and visual recordings or sketches on the 3D scene. Also, drawing inspiration from [4,1], 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.



### AFFILIATIONS



### REFERENCES

- [1] 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>.
- [2] 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.
- [3] JENNINGS, CASTELLI, BOSTROM, BRUAROEY: WebRTC: Real-Time Communication in Browsers. Tech. rep., W3C WebRTC Working Group, March 2025. URL: <https://www.w3.org/TR/webrtc/>.
- [4] 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>