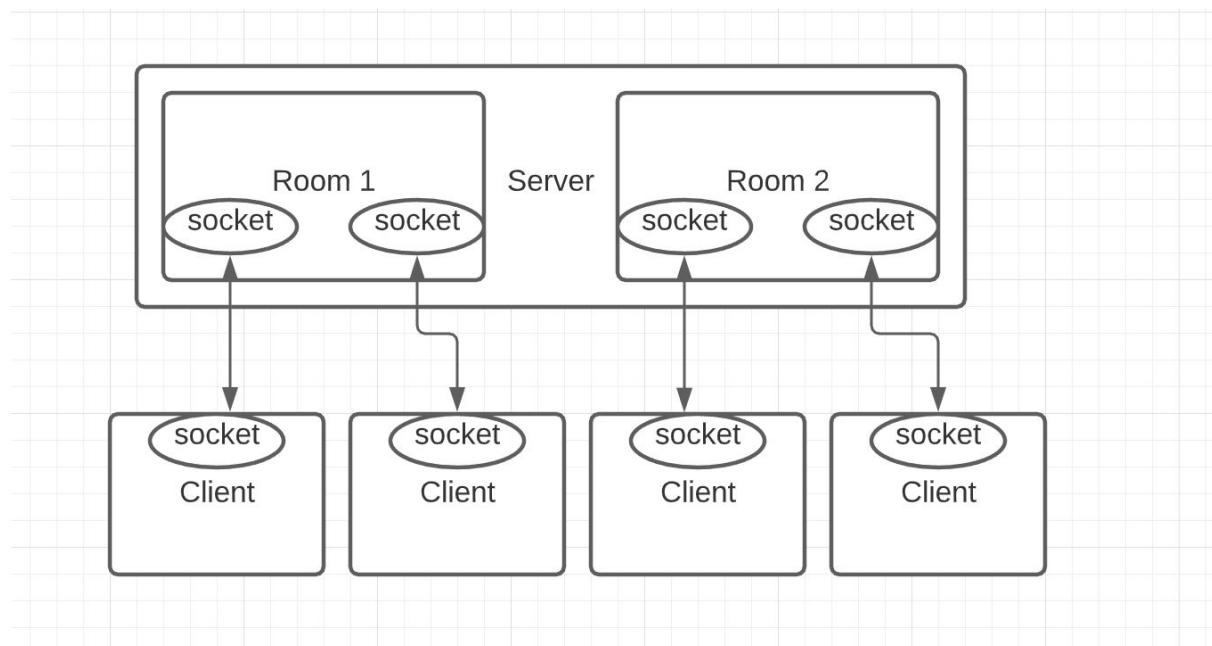


Design Document

Application Overview: 3313 Super Chat Room is designed as a Proof of Concept of an online discussion chatroom meant to support student's remote learning experience via discussion rooms centered around OS concepts and language specifics.

Functionality: Upon loading the main page, the student would be greeted by a number of possible chatroom options based on different topics in the 3313 curricula (multiple possible concurrent transactions). They will also be greeted by a "Professor Grolinger Bot" acting as a system administrator who will announce personal as well as worldwide updates. Students can send messages to the chat room and see their messages reflected. Upon exit of the chat room, they will also see their exit reflected on other students' clients.

Technology: The language used for backend scripting in Javascript with Node.js v15.4.0 as its runtime engine. Socket.io is used to manage the Websocket connection as well as maintain synchronous multi-threaded communication between potential clients. The application is currently running as a single process Javascript event loop on our runtime engine. However, it can be easily loaded balanced/scaled up using either Nginx or Redis for load balancing



Backend Design Diagram: concurrent transactions are managed by the chat room server, which separates the socket connection of different clients abstracted into different levels of transaction separation. The client is aware of the possible room options and can disconnect as well as connect to wildly different rooms, engaging in discussions that interest them.

Peaceful Termination: The peaceful termination is triggered via a shell script generated by the server at the inception of this service. Depending on whether we shut down the socket connection or the server connection, the interaction would be wildly different on the client end. Nevertheless, an alert should pop up for connected users, reminding them that the application is closing. It will also close the window using javascript on the client end.

Moving forward:

Necessary for Production:

1. An eventual consistent message platform backed by a database management system that will save the conversation for later revisit
2. Admin console for removing specific comments in the chat room
3. A more robust authentication system.

Nice to Haves:

1. Custom greetings, alerts, comments for the system administrator/Professor Grolinger Bot (Implementation is 25% finished).
2. Distributed locks in Redis to ensure that a single IP address can only make a single socket connection.
3. Automatic publishing tools that can be used to cutout highlighted discussions and integrate with Owl Discussion Forums