Random Peer to Peer Chatting Application using WebRTC

Prof. Rushali Patil[1], Satish Kumar Patra[2], Krishnanunni R S[3], P Deepak Raj[4], Mahima Yadav[5]
Department of Computer Engineering
Army Institute of Technology Pune, India

Abstract: *Because of its ease for real-time communication and simple features, messaging has become an inextricable aspect of our lives. Because of their quickness and potential, messaging or chat services have become an indisputable part of how people communicate in today's world. By exchanging messages in real time, the chat program makes it simple to communicate with individuals all over the world. People are increasingly using immersive experience to connect with one another. Chat apps are rising in popularity because they can virtually retain the feeling of real-time interaction, from group messaging in live chat to e-learning and team collaboration through chat rooms to file sharing between coworkers. However, when it comes to deciding how to create the app, the consumer experience is critical. Our goal is to create a chat software with real-time messaging services that provide users with a genuine and highly interactive engagement. Our goal is to create an application that is dependable, safe, and runs in real time, without concern for user volatility or concurrent constraints. Make sure the chat application has the right security safeguards when working with sensitive material like regulatory requirements and confidential user information. As a solution to the above-mentioned requirements, we propose a random peer-to-peer chatting application using technologies like WebRTC and PeerJS. The recent improvements in WebRTC show how efficient this implementation could become. Here in this project, We propose PeerJS, which is a wrapper in JavaScript to use WebRTC without considering connection drops as everything is handled properly. At the same time, the signaling needed by WebRTC is provided by PeerJS although a custom signaling server can be created from scratch in a matter of minutes*

Keywords: WebRTC

REFERENCES

[1]. Shi Yuzhuo1 , Hao Kun2 1 Tianjin College of Commerce, Tianjin, China, 2011 IEEE. Design and Realization of Chatting Tool Based on Web

[2]. Bita Kheibari Uluslararası Bilgisayar Enstitüsü Ege Üniversitesi 2020 IEEE. A WebRTC Architecture Assisted by Software Defined Networks

[3]. Alexandre Gouaillard, Ludovic Roux CoSMo Software Consulting, Singapore, 2019 IEEE. Real-Time Communication Testing Evolution with WebRTC 1.0

[4]. R. Arda Kirmizioglu and A. Murat Tekalp Multi-party WebRTC Services using Delay and Bandwidth Aware SDN Assisted IP Multicasting of Scalable Video over 5G Networks .Web Application for Social Networking using RTC GeorgeSuciu R&D Department BEIA Consult International   Marian Ceaparu R&D Department

[5]. BEIA Consult International, WebRTC role in real-time communication and video conferencing

[6]. Samuel Micka, Utkarsh Goel, Hanlu Ye,† Mike P. Wittie, and Brendan Mumey Department of Computer Science, Montana State University, Bozeman, MT 59717

[7]. Roberto Beraldi, Gabriele Proietti Mattia DIAG, La Sapienza University of Rome, Italy. Power of random choices made efficient for fog computing

[8]. Shi Yuzhuo1 , Hao Kun2 1 Tianjin College of Commerce, Tianjin, China. Design and Realization of Chatting Tool Based on Web

[9]. Sunghyun Yoon, Taeheum Na, and Ho-Yong Ryu Smart Network Research Department, Electronics and Telecommunications Research Institute, Daejeon, Korea. An Implementation of Web-RTC based Audio/Video Conferencing System on Virtualized Cloud

[10]. Alexandru C. 2014. Impact of WebRTC (P2P in the Browser). inB. Stiller et al. (Eds.). Internet Economics VIII. Technical Report. Department of Informatics, University of Zurich.

[11]. Vanessa W., Salim F. and Moskovits P. 2013. The definitive guide to HTML5 WebSocket. New York: Apress.

[12]. Beizer B. 1995. Black-box testing: techniques for functional testing of software and systems. Canada: John Wiley and Son.