
PROXIMA

Furkan Bayansar
Emre Yiğit Alparslan
Mehmet Emin Bakır
Mazlum Ferhat Arslan

Introduction

- What was the problem ?
 - Lack of Interaction with participant in webinar tools
- What was current solutions ?
 - Using instant messaging for interaction
- Our solution
 - Face-to-face interaction between all participant



Proxima

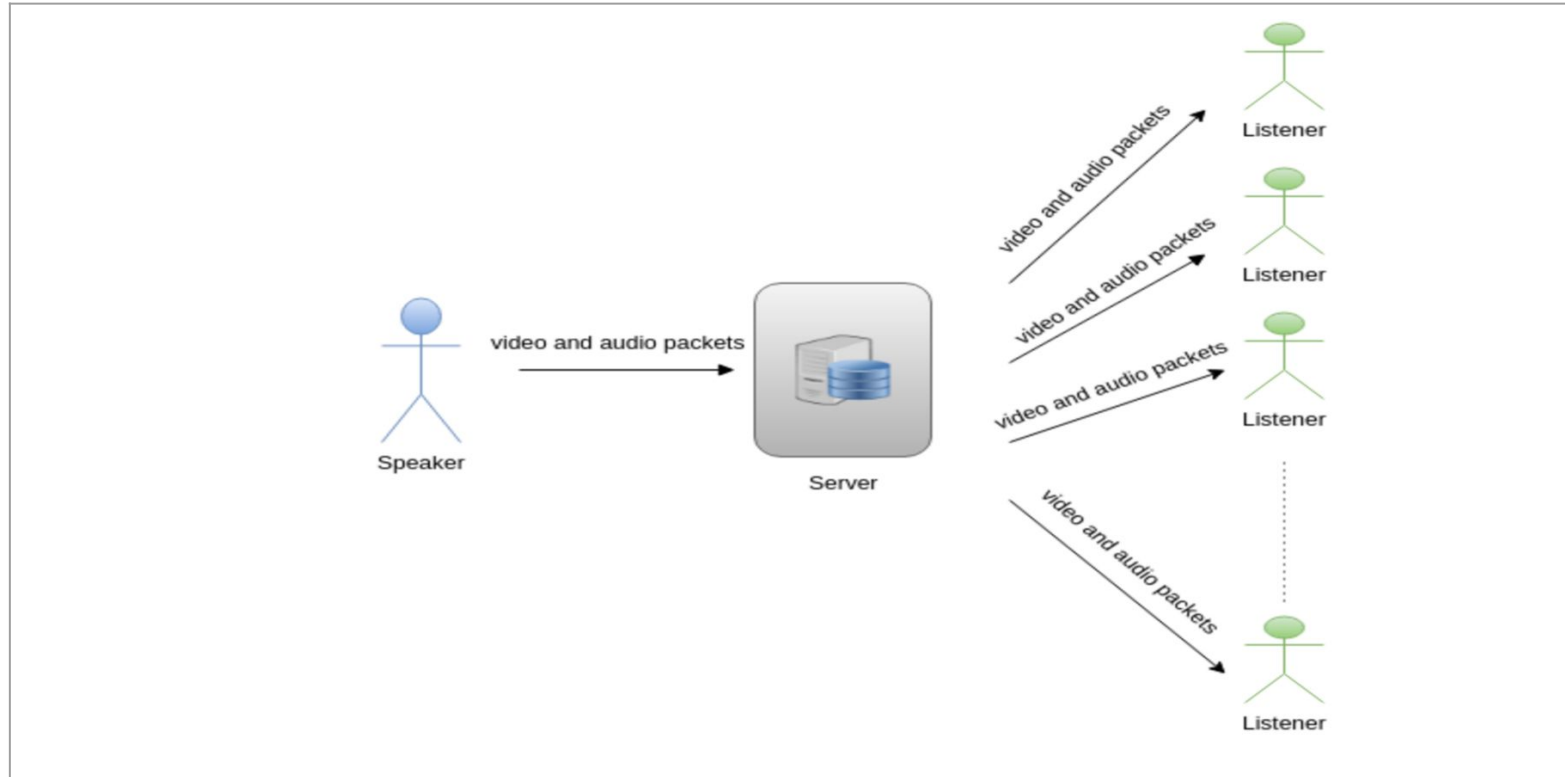
Part I: Broadcasting

- This part uses server-to-client communication.
- 4 seconds video and audio packets are sent to the server.
- Server sends these packets to participants separately.
- Packets are sent through socket.io connection.
- This process causes at least 4-5 seconds delay. However, this is not a big concern (ex. Ustream has 30 seconds delay).
- Scalability tests could not be completed for large amount of participants.

Part I: Broadcasting(cont.)

- Broadcasters can record broadcast, if they want.
- Recording starts, when broadcaster click on “Start Recording” button.
- If broadcast was recorded, “Watch Recorded Broadcast” button will appear to all participants.
- They can watch recorded broadcast whenever they want.
- RecordRTC have been used to record broadcast.
- Recorded video is sent to server through socket.io, and then it is saved into server folders for later watching.

Part I: Broadcasting Diagram



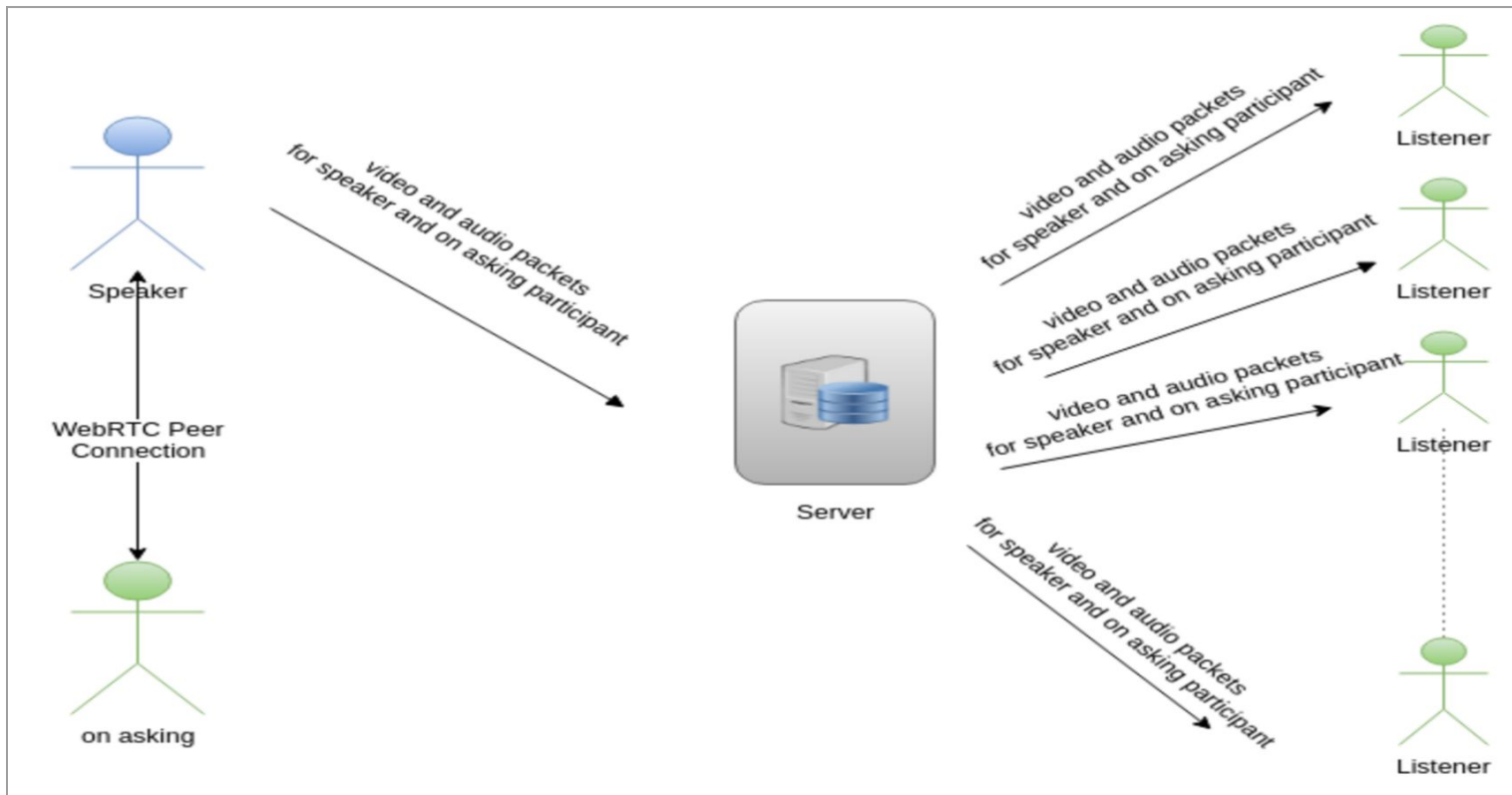
Part II: Asking Question

- When a participant wants to ask a question, sends request to speaker.
- Required data, to establish peer-to-peer communication is sent through socket.io after speaker lets participants ask question.
- WebRTC peer-to-peer connection is started between speaker and questioner.
- Video and audio packets of the questioner is sent by speaker to server.
- Server sends questioner's and speaker's packets together to all other participants.

Part II: Asking Question(cont.)

- p2p connection between speaker and the questioner provides communication without delay.
- Questioner and speaker data packets reach each one of the participants at the same time. Therefore, participants can follow speaker and questioner simultaneously.
- This server-to-client communication is same as in broadcasting part.

Part II: Asking Question Diagram



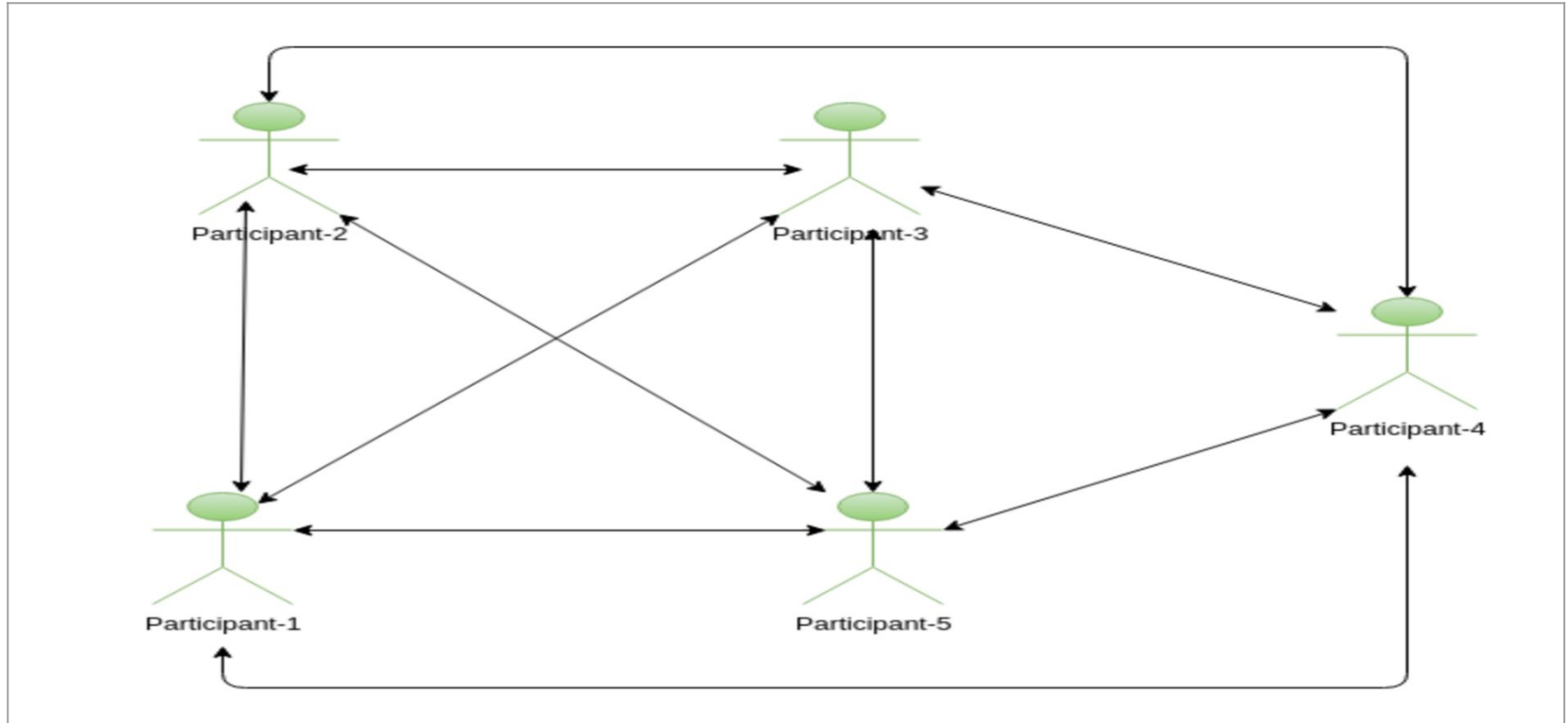
Part III: Video Chat Room

- This part implemented in order to increase interaction among participants.
- After one of the participants opens a video chat room, up to 4 people can join that room.
- Communication among those participants will be p2p.
- Required data, to establish peer-to-peer communication is sent through socket.io after a participant wants to join a room.

Part III: Video Chat Room(cont.)

- WebRTC peer-to-peer connection is started among participants who are in the room.
- This communication has good quality.
- In testing process, there was no problem, if participants have the Internet connection, which is good enough.
- This part is successfully completed.

Part III: Video Chat Room Diagram



Technologies



socket.io



WebRTC



express



Bootstrap

