

Architecture

Client-Server Architecture:

WebRTC

Overview:

WebRTC is a collection of communication protocols and APIs that enable real-time peer-to-peer communication directly in web browsers without the need for plugins or additional software. It provides functionalities such as audio and video streaming, data transfer, and peer-to-peer communication.

Components:

- **Web Browser:** The client-side application runs in a web browser, which supports WebRTC APIs.
- **WebRTC APIs:** JavaScript APIs provided by the web browser for establishing peer connections, managing media streams, and exchanging data.
- **Signaling Server:** While WebRTC facilitates peer-to-peer communication, a signaling server is required for initial connection establishment, negotiation, and coordination between clients. Signaling can be implemented using various protocols such as WebSocket or HTTP.

Architecture

- **Client-Side Web Application:** The client-side application is developed using HTML, CSS, and JavaScript, utilizing WebRTC APIs for audio streaming.
- **WebRTC APIs Usage:** The application utilizes WebRTC APIs such as **getUserMedia** for accessing the user's audio input, **RTCPeerConnection** for establishing peer-to-peer connections, and **MediaStream** for managing audio streams.

- **Peer Connection Establishment:** Once signaling is complete, the clients establish a direct peer-to-peer connection using WebRTC. They exchange session descriptions (SDP) containing information about their audio capabilities and network configurations.
- **STUN/TURN Server Usage:** If direct peer-to-peer communication is not possible due to network restrictions, the clients can use STUN/TURN servers to relay audio data through a third-party server.

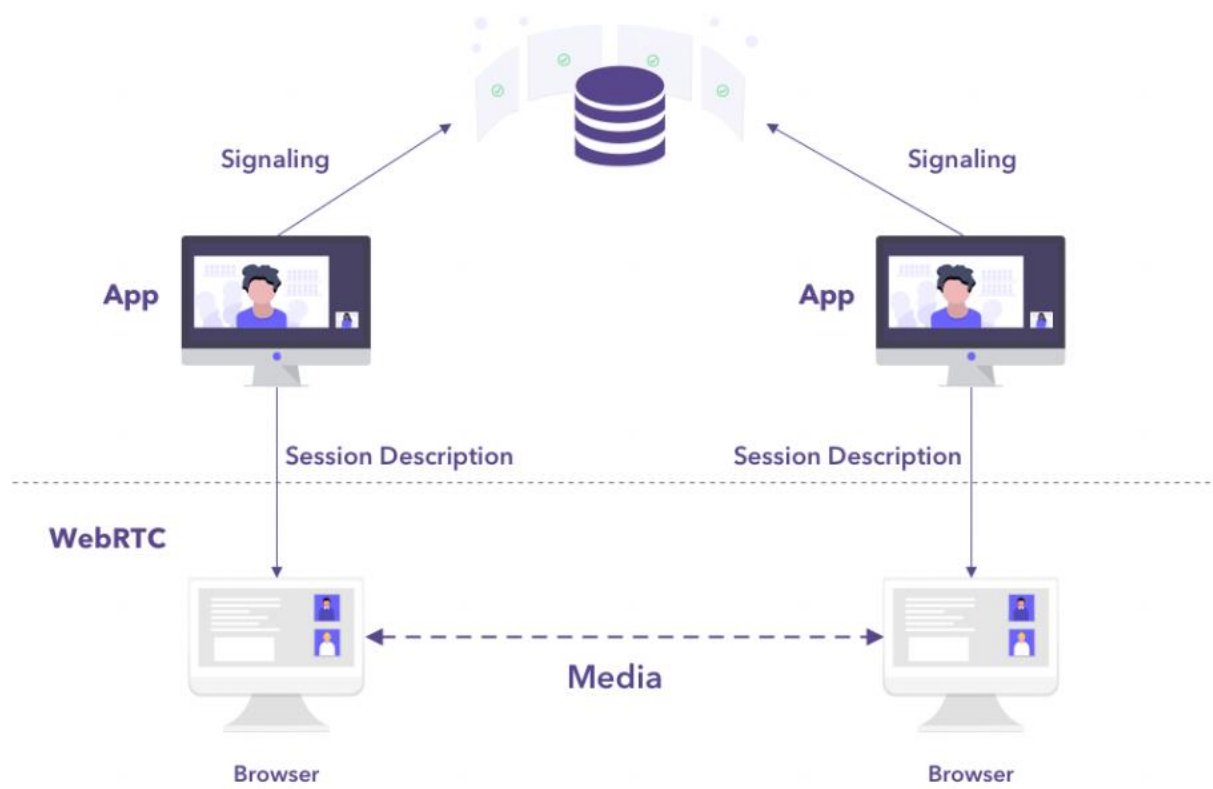
Benefits of WebRTC

- **Built-in Browser Support:** Most modern web browsers support WebRTC, allowing for seamless integration without the need for plugins or additional software.
- **Low Latency:** WebRTC is designed for real-time communication, offering low-latency audio streaming suitable for applications requiring high-quality, bi-directional audio.
- **Security:** WebRTC provides built-in encryption and security features, ensuring the privacy and integrity of audio streams.

Considerations:

- **Browser Compatibility:** While most modern browsers support WebRTC, it's essential to consider browser compatibility and provide fallback mechanisms for older browsers if necessary.
- **Audio Quality:** Optimizing audio settings such as codec selection, bitrate, and echo cancellation to ensure high-quality audio streaming while minimizing latency.
- **Network Conditions:** Handling network fluctuations and implementing mechanisms for adaptive bitrate adjustment and packet loss mitigation to maintain audio quality under varying network conditions.

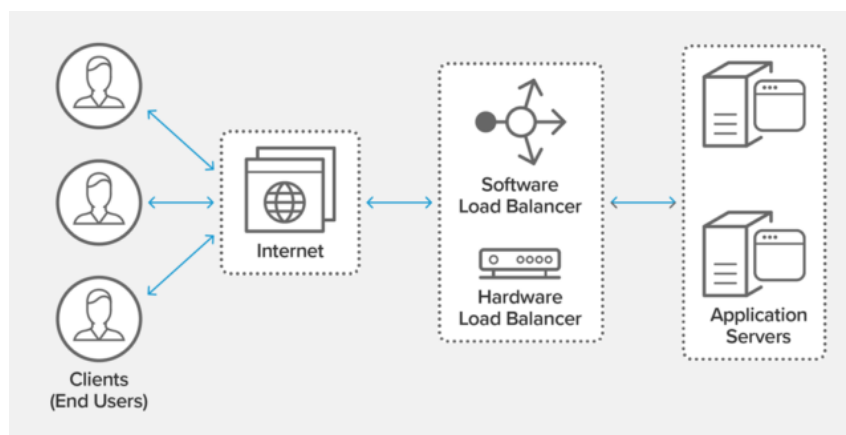
By leveraging WebRTC on the client-side, you can develop a real-time bi-directional audio streaming service with minimal overhead and complexity, offering low-latency, high-quality communication directly in web browsers.



Server-Side Architecture:

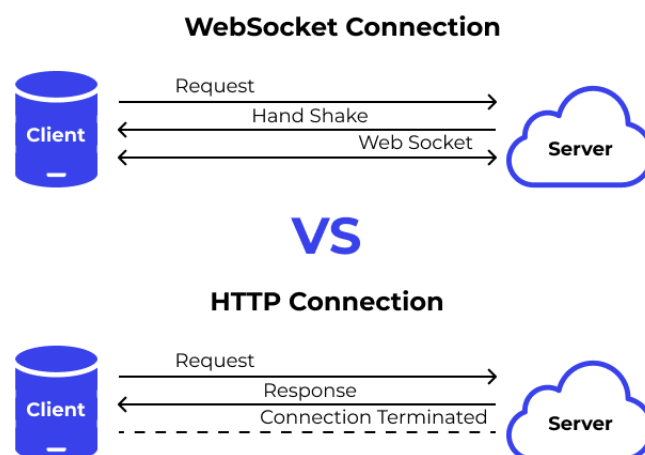
Load Balancer:

Fronting the server-side infrastructure to distribute incoming connections across multiple instances for scalability and availability.



Connection Manager:

Responsible for handling incoming connection requests, establishing and managing bi-directional audio streams between clients.



Database:

Stores user authentication data, user preferences, and session information.



- MongoDB follows a document data model
- MongoDB is useful for scaling large data volumes
- in-memory storage engine for a hybrid approach
- You can use memory caching for data you frequently access

Cloud hosting platforms:

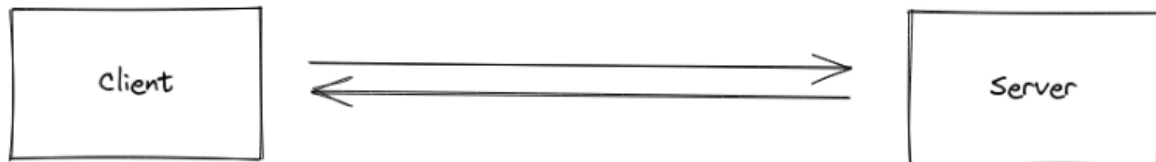
Cloud hosting platforms offer scalable infrastructure and services, including virtual servers, storage, databases, and networking, enabling businesses to deploy, manage, and scale applications and services globally. Leading providers include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and IBM Cloud.



Technologies:

1. Socket.IO

Socket.IO is a library that enables **low-latency**, **bidirectional** and **event-based** communication between a client and a server.



Socket.IO is NOT a WebSocket implementation.

Socket.IO indeed uses WebSocket for transport.

2. WebRTC

WebRTC (Web Real-Time Communications) is an open source project that enables real-time voice, text and video communications capabilities between web browsers and devices.

WebRTC provides software developers with application programming interfaces (APIs) written in JavaScript.

Developers use these APIs to create peer-to-peer (P2P) communications between internet web browsers and mobile applications without worrying about compatibility and support for audio-, video- or text-based content.