## What is WebRTC?

WebRTC stands for Web Real-Time Communication. It is a set of APIs and protocols that allow web applications to exchange multimedia data (audio, video) and arbitrary data directly between peers. This technology is built into modern browsers, meaning that real-time communications can be achieved without installing extra software.

## How WebRTC Works

WebRTC uses several components and protocols to establish and maintain direct connections

1. Peer-to-Peer Communication
   1. RTCPeerConnection API: Manages the connection between peers, handling the streaming of audio and video.
   2. MediaStream API: Captures media from the user’s device (using getUserMedia) and streams it to the remote peer.
2. Signaling Process
   1. Before a direct connection is established, peers must exchange control messages (e.g., session descriptions and ICE candidates).
   2. Signaling Server: Although WebRTC handles the media and data transfer, it does not specify the signaling mechanism. Developers typically use WebSocket, XHR, or any custom protocol to exchange signaling messages (offers, answers, and ICE candidates).
3. NAT Traversal:
   1. STUN (Session Traversal Utilities for NAT): Helps clients discover their public IP addresses when behind a NAT.
      1. A STUN server is primarily used for NAT traversal. it helps a device discover its public IP address and port as seen on the internet. Each peer independently contacts the STUN server to learn its own external mapping, and then this information is exchanged between peers via the signaling channel. In other words, the STUN server itself does not directly exchange the public IP addresses between peers; it simply provides each peer with the necessary information to establish a direct connection.
   2. TURN (Traversal Using Relays around NAT): Used as a fallback to relay media if a direct peer-to-peer connection cannot be established.
      1. While STUN (Session Traversal Utilities for NAT) provides an efficient method for NAT traversal and establishing peer-to-peer connections in numerous cases, it faces significant challenges with stringent NAT setups or intricate firewall restrictions. TURN (Traversal Using Relays around NAT) servers become essential in such situations. Acting as intermediaries, TURN servers facilitate the relay of traffic between devices unable to form a direct connection.