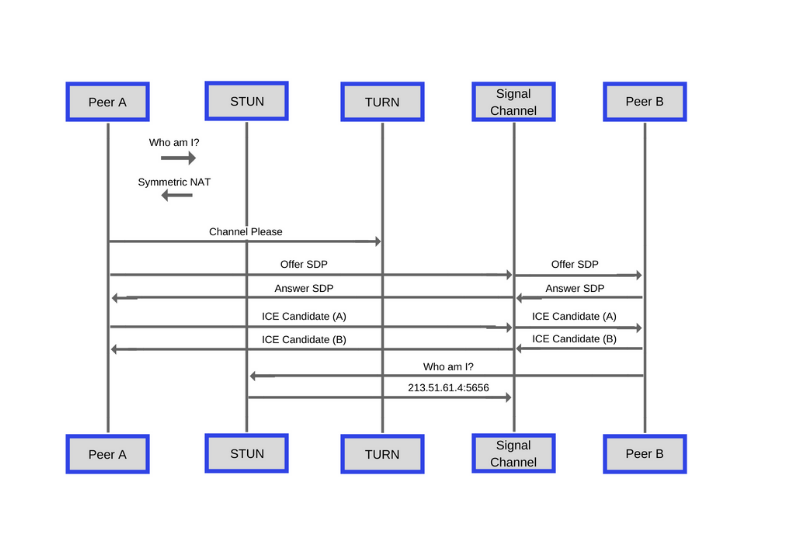
**Architecture Design**

1. **Jitsi Applications**

[Collection of Open Source projects](https://jitsi.github.io/handbook/docs/architecture) which provide state-of-the-art video conferencing capabilities that are secure, easy to use and easy to self-host.

1. **Jitsi Architecture**
2. [Jitsi Meet](https://jitsi.org/jitsi-meet) - WebRTC compatible JavaScript application that uses Jitsi Videobridge to provide high quality, scalable video conferences.

* **WebRTC:** is [open-source](https://webrtc.googlesource.com/src/) and supported by Apple, Google, Microsoft.., amongst others. This page is maintained by the Google WebRTC team. you can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions.

**There are 3 primary components in WebRTC**

* **MediaStream API**

The MediaStream API provides the functionality to access camera, microphone or screen using javascript.

* **RTCPeerConnection API**

The RTCPeerConnection API takes care of the NAT traversal, Codec processing, SDP negotiation, Media transferring and much more on handling the secure connection between peers.

* **RTCDataChannel API**

The RTCDataChannel API allows to setup bidirectional data transfer channel between peers.

1. [Jitsi Videobridge (jvb)](https://jitsi.org/jitsi-videobridge) - WebRTC compatible server designed to route video streams amongst participants in a conference

* Jitsi Videobridge is an XMPP server component that allows for multiuser video communication. Jitsi Videobridge does not mix the video channels into a composite video stream, but only relays the received video channels to all call participants. And it’s fully open source and WebRTC compatible.

1. [Jitsi Conference Focus (jicofo)](https://github.com/jitsi/jicofo) - server-side focus component used in Jitsi Meet conferences that manages media sessions between each of the participants and the videobridge

* It is responsible for managing media sessions between each of the participants and the videobridge. Whenever new conference is about to start an IQ is sent to the component to allocate new focus instance.
* After that special focus participant joins Multi User Chat room. It will be creating Jingle session between Jitsi videobridge and the participant.
* Although the session in terms of XMPP is between focus user and participant the media will flow between participant and the videobridge. That's because focus user will allocate Colibri channels on the bridge and use them as it's own Jingle transport.

1. [Jitsi Gateway to SIP (jigasi)](https://github.com/jitsi/jigasi) - server-side application that allows regular SIP clients to join Jitsi Meet conferences

* **SIP (Session Initiation Protocol):**  is a protocol used in [VoIP communications](https://www.lifewire.com/getting-started-with-voip-3426746) allowing users to make voice and video calls, mostly for free.

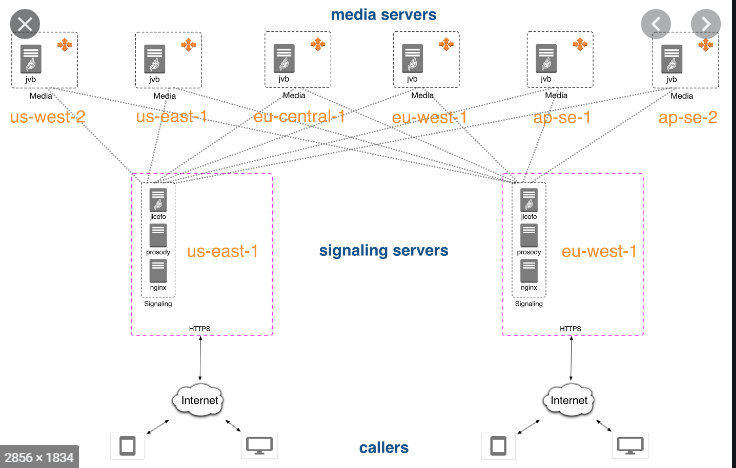
1. [Jibri](https://github.com/jitsi/jibri) - set of tools for recording and/or streaming a Jitsi Meet conference that works by launching a Chrome instance rendered in a virtual framebuffer and capturing and encoding the output with ffmpeg

* It works by launching a Chrome instance rendered in a virtual framebuffer and capturing and encoding the output with ffmpeg. It is intended to be run on a separate machine (or a VM), with no other applications using the display or audio devices. Only one recording at a time is supported on a single jibri.

1. **Server**

The prosody xmpp server is central in the jitsi-meet systemarchitecture. jicofo is configured to be an administrator on the prosody xmpp server.

The video bridge and other jitsi modules such as jicofo are users on the prosody xmpp server. The system can be expanded to cover various usecases.



1. **Design model**

**+, Web UI: Jitsi have used Nginx to build to web UI.**

* + Nginx: is an HTTP reverse proxy server, a mail proxy server, and a general-purpose TCP/UDP proxy server.

+, **Prosody: the XMPP server**

* + Prosody / XMPP serves as user authentication and helps provide chat and messaging services.

+, **Jitsi meet:** the core of Jitsi

Its mission is to:

1. Create a layer for the XMPP signal.
2. Create connection peers.
3. Accepting media / messages and files.

Once the application starts to execute and process the signal, the user can communicate via the Jicofo as it is responsible for signaling and forwards the JVB.

+, **Jicofo**: the XMPP focus component.

* + It is responsible for the management of communications between each participant and the JVB.

1. Manage conference: who joins the room, who goes out of the room.
2. Manage Colibri channels for participants and set up communication streams to the JVB.
3. Load balancing Jitsi video bridge based on conference number.
4. Handle client authentication (optional).
5. Participate in some functions such as mute, recording.

+, **Jvb**: Jitsi Videobridge, the video router.

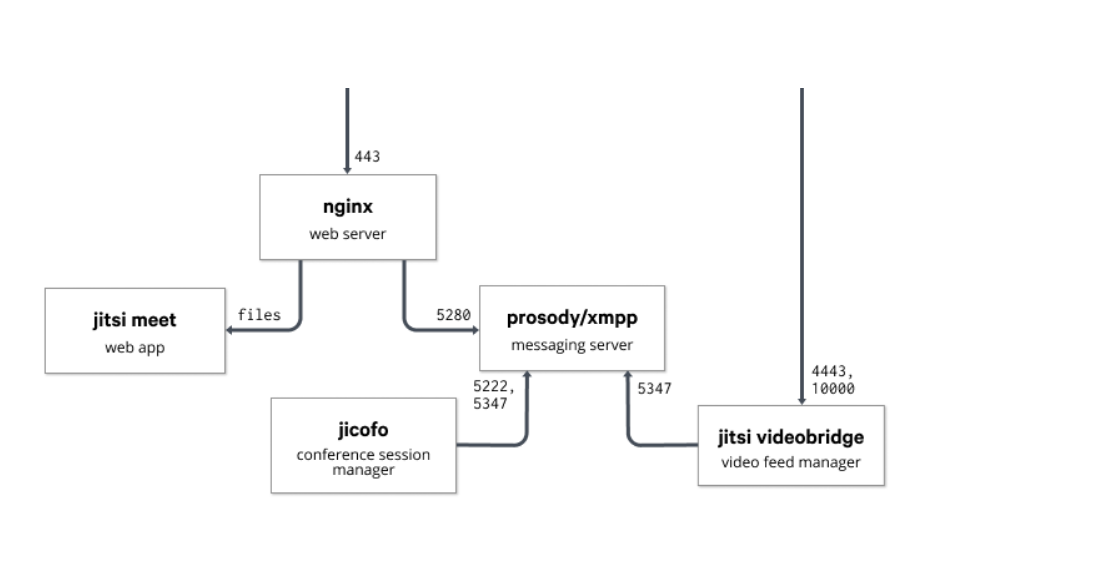
* + This is a core component that allows multiple users to communicate with each other via video conferance.
  + Jitsi Videobridge SFU (Selective Forwarding Units) for media distribution, does not mix video channels into an aggregated video stream, but only forwards received video channels to all call participants.

+, **Jigasi:** the SIP (audio only) gateway.

* + An affiliate application that allows regular SIP clients to participate in Jitsi Meet conferences hosted by Jitsi Videobridge.

+, **Jibri**: the broadcasting infrastructure.

* + Jibri offers recording or streaming services in Jitsi Meet conference.



# **Cloud API**

The Jitsi Meet Cloud API is a specification for services which can support the integration of Jitsi Meet into other applications, for mapping conferences for dial-in support, and for supporting directory search and user invitations to conferences.

