Looking at building educational tools

My background is in instructional design and technology. Every so often I talk to old classmates and the topic always gets around to technology and how much it sucks from their perspective. I mean, yes, we can do large scale courses online but it's far from optimal… the difference of opinion comes as to best implement these technologies in ways that students will want to use them.

The post will discuss some low level infrastructure and learning tools that we can build with them.

Some of these ideas may be rough and not fully fleshed out.

Before

Chat, Collaboration and Video Conferencing

Most modern multiuser audio and video systems use [WebRTC](#) as the backbone for the system. The technologies are open source and quite powerful but, depending on the use case, they can be quite complex to set up.

See the [WebRTC APIs](#) codelab from Google for a good overview to the technologies involved.
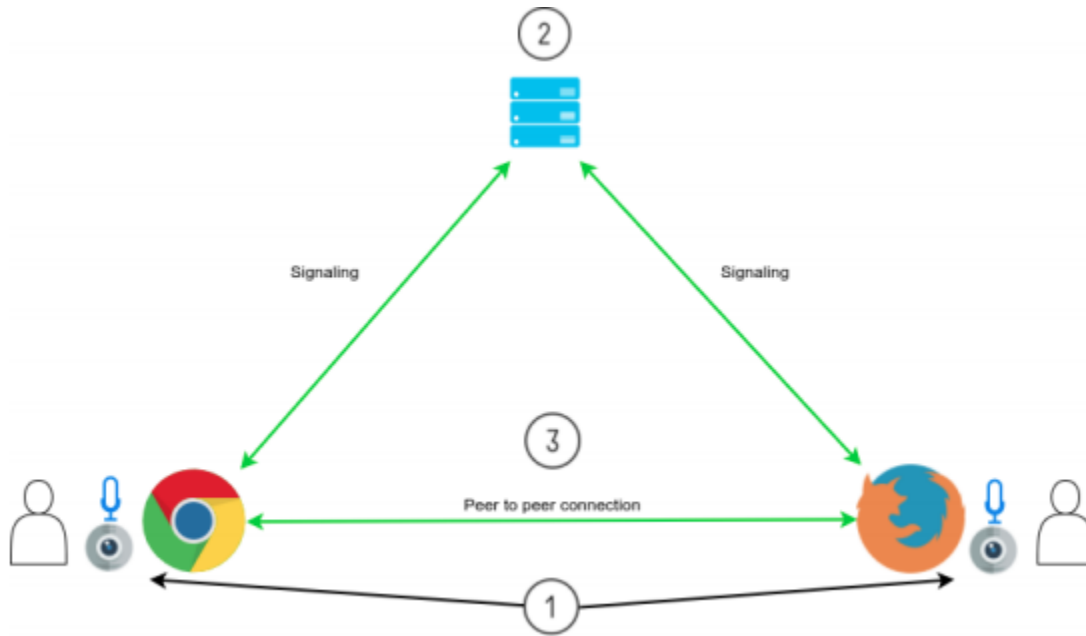
Basic Audio and Video Communication

Figure 1: Basic WebRTC communication flow. Taken from How to Build a Video Conference Application with WebRTC

At the most basic level, the process for enabling communication on WebRTC is a three step flow:

1. The browser gets access to the media devices (camera and microphone0)
2. Each peer exchanges information about itself with all other peers through a signaling process
3. After signaling, peers can connect directly and the communication begins

To make all these possible we need the following elements:

- A WebCamera and microphone for each participant
- A signaling server for information exchange between peers
- A pair of STUN / TURN servers are required for NAT traversal and to relay media if direct communication is not possible

See Signaling and video calling for more information on creating a WebRTC application and

This will take care of the basic multi-user communication flow, it doesn't deal with sharing data or side-channel communication.

The data channel

Some existing tools

Some examples of WebRTC frameworks:

- [Pion](#)
- [Media Soup](#)
- [Jitsi](#)

Google Docs-like experience

- Pre-packaged solutions
    - Etherpad
        - [Website](#)
        - [Etherpad Lite Github Repo](#)
- Build from scratch
    - [WebSockets - A Conceptual Deep Dive](#)
    - References
        - [The WebSocket API](#)
        - [The WebSocket Protocol](#)
        - [WebSockets - MDN](#)
    - Older links I still found useful as conceptrual models
        - [Real-time Collaborative Editing with Web Sockets, Node.js & Redis](#)
        - [Introducing WebSockets: Bringing Sockets to the Web](#)
    - (Older) Examples
        - [Plink](#)
        - [Paint With Me](#)
        - [Pixelatr](#)
        - [Dashed](#)
        - [Massively multiplayer online crossword](#)
        - [Ping server](#)

Using Google Maps For Learning Games

Technical issues

Authentication

Data Collection, Retention and Security