Case Report

WebRTC: delivering telehealth in the browser

Arin W. Sime

WebRTC.ventures, Charlottesville, Virginia, USA

Correspondence to: Arin W. Sime. CEO/Founder of WebRTC.ventures, Charlottesville, Virginia, USA. Email: arin@agilityfeat.com.

Abstract: WebRTC is enabling a new generation of Telehealth applications and will be an important part of the future of Telehealth. WebRTC allows for web applications to control a user's microphone and video camera from the browser. In this viewpoint, the author presents the pros and cons of WebRTC for Telehealth applications.

Keywords: WebRTC; telehealth; browser

Received: 22 February 2016; Accepted: 24 March 2016; Published: 13 April 2016.

doi: 10.21037/mhealth.2016.03.08

View this article at: http://dx.doi.org/10.21037/mhealth.2016.03.08

Telehealth is a rapidly expanding part of healthcare globally, with the potential to save costs and better serve patients with specialist care, regardless of where they live (http://www.hhnmag.com/articles/3648-telehealth-promises-to-reshape-health-care, "Telehealth Promises to Reshape Health Care", *Hospitals & Health Networks Magazine*, March 10, 2015).

Although the Internet has been widely available to the global public for decades now, it's only recently that several factors have converged to make delivering telehealth through the web browser feasible. Until recently, there was not the right combination of high bandwidth and security needed to enable telehealth.

In 2012 a new web standard was proposed as part of the larger HTML5 standard. WebRTC stands for "Web Real Time Communications", and it allows for web sites to get access to a user's camera and microphone on their computer (with user permission of course), and then to connect that user's browser with other users.

Put more simply, WebRTC allows us to build video chat applications directly into websites. This gives users functionality like Skype and Google Hangouts, but without any downloaded applications necessary.

WebRTC is a protocol for allowing to browsers to connect to each other in a Peer-to-Peer (P2P) fashion, and then exchange streams of video, audio, and data directly between the two browsers (*Figure 1*).

P2P means there is no intermediary server transferring the data between the two users. This has very important privacy considerations, as I'll explain in a moment. The two users must use a website to make the initial connection to each other—a simple example is to imagine a patient browsing a directory of medical specialists on a website, and then choosing to call a particular doctor.

In that example, the web application will help the patient to call the doctor, just like a receptionist may help you establish a connection to a particular office extension. Once that connection is established through a process called "Signaling", none of the ensuing voice or video data will be sent through the web server however—it's now a direct connection between the two users' browsers.

This is a big deal for telehealth, because it means that confidential patient information is never stored on other computers or media servers. In addition, the WebRTC protocol encrypts all the video and voice communications between the two users automatically, which means that patient information is encrypted in transit over the Internet.

That P2P architecture and encrypted data transit is what makes WebRTC so attractive to telehealth. Those characteristics of WebRTC make it easier to build highly secure and HIPAA compliant remote teleconsultation services.

So far everything I've described is how WebRTC connects two users' browsers on their laptops, but with a little extra work it's also possible to use WebRTC to connect mobile devices too. The same applications can be built to connect a tablet or smartphone to others on desktop computers or mobile devices.

Traditional IT solutions for Unified Communications or remote telehealth involve large software packages with

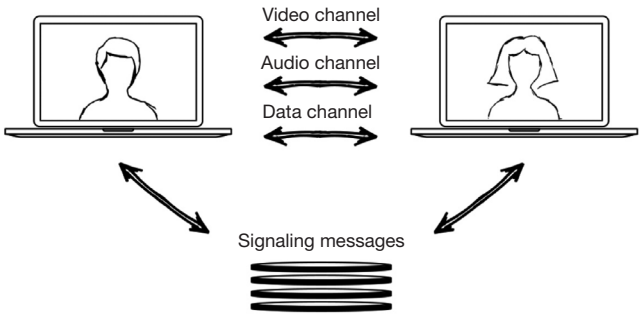

Figure 1 Overview of peer-to-peer WebRTC connection between two users. Signaling messages are used to establish the Peer Connection, and after that, the users can exchange live video, audio, and data with each other without an intermediary server.

high licensing costs. WebRTC is an open standard which is much easier to implement, which means that WebRTC also has a democratizing effect, making it possible to build less expensive telehealth solutions that individual doctors and family practices may be able to afford to incorporate into their offerings, without the buying power of a large hospital or health care company.

Given all of these advantages of privacy, security, and cost, will WebRTC-based video chat be the future of telehealth? There's no doubt in my mind that it's an important component of mobile health, but of course there's more to it.

WebRTC has a few challenges that you need to consider before rushing to build a mobile health application with it.

First, the P2P nature is both an advantage and a disadvantage. We already discussed its advantage: no intermediary media server means participants in a WebRTC video chat have a higher assurance of privacy because the video and audio streams are shared directly with each other.

The disadvantage of this P2P architecture is that it doesn't scale well to large conversations. If your telehealth use case requires a larger number of participants, then WebRTC may not handle the network load well in its "pure" P2P format. Every participant in the call must have a P2P connection with every other participant, which causes a large increase in bandwidth and processing power needed by each participant's computer.

If you plan to build a remote surgery teaching application, with dozens of students watching live, then WebRTC may not scale well.

There is a way around this—many WebRTC based conferencing tools will use a media server in the middle of the conversation that allows for combining of the video streams into single streams. This server in the middle means that your WebRTC implementation is no longer purely P2P, and therefore has lost a little of the privacy allure of "pure" WebRTC. This architecture also adds a lot of technical complexity and cost, but that complexity can be mitigated by building on top of commercial WebRTC platforms like TokBox, who can provide the media servers for you at a per-minute charge.

The second downside of WebRTC is that it is not supported natively in all web browsers or mobile devices yet. Apple devices don't support it yet in the phone browser, which means your users will have to download a native mobile application to join in a telehealth experience. That is a bit of a hassle, and also adds significant development cost since you may need to hire mobile developers as well as web developers if you are building a telehealth app from scratch.

On desktop computers, you need to use the Chrome, Firefox, or Opera browsers in order to use a WebRTC application right now. Microsoft's Edge browser is expected to support it in the future, but Internet Explorer is going to be phased out and will not ever support WebRTC.

A final challenge to be aware of with WebRTC is that it may be difficult to establish a call successfully behind some hospital networks. If your hospital has a very restrictive network security policy, it may be difficult for two web browsers using WebRTC to establish the P2P connection necessary to hold a call. You can test this on your corporate network by trying out a free WebRTC based video conferencing tool such as Appear.in.

Although I am a big fan of WebRTC, and the revolutionary impact that I believe it will have on telehealth, it's important to remember that it is not a panacea. Technology rarely is a perfect solution unfortunately, but when we take into account all the pros and cons of telehealth, then I am willing to wager that in the coming years you will soon join in a WebRTC based video chat also.

Acknowledgements

None.

Footnote

Conflicts of Interest: The author has no conflicts of interest to declare.

doi: 10.21037/mhealth.2016.03.08
Cite this article as: Sime AW. WebRTC: delivering telehealth in the browser. mHealth 2016;2:11