

2017 WebRTC vs Adobe Flash: improving streams transfer

The existing [technological solution](#) of OpenMeetings has both pros and cons. The advantage is Flash-player availability for all popular browsers, either by default or as a downloadable plug-in. Accordingly, user has only to go to the correct link to start a web conference in OpenMeetings. Another advantage is that the OM application works directly in a browser, thus, on most modern operating systems.

Now about limitations. The current implementation is limited to video codecs supported by Adobe Flash, there are just few and most of them are private. Since 2011 Flash support for mobile platforms has been stopped, which costs great effort for mobile developers to implement adequate functionality utilizing Flash on mobile devices.

Since the ten years of the project life new technologies implementing real-time interaction has appeared, options are now being considered that can adequately replace Flash elements of the interface with another mechanism for transmitting video / audio streams.

One of these alternatives is the use of **WebRTC** technologies set that allow you to work with video and audio streams directly in browser. At the moment, WebRTC is supported by most modern desktop and mobile browsers.

WebRTC includes three different APIs for processing data streams: 1) `getUserMedia`, which receives user data within a browser, for example, a video stream from webcam; 2) `RTCPeerConnection` that transmits streams over the network; 3) `RTCDataChannel` - set of APIs designed to help open UDP connections between two browsers.

The main advantages of WebRTC vs Flash are that you do not need to install any plug-ins or extensions to the browser because now it is supported out of the box. Also, WebRTC is more resistant to "crashes" and uses html5 directly unlike Flash technology.

However, webRTC is a fairly young technology and is still weakly optimized compared to Flash. But since it has already become a de facto standard for streaming data processing, the emergence of high-performance implementation of this technology is just around the corner.

In OpenMeetings, we started the transition to WebRTC technology in June 2015 as a students' project within Google Summer of Code summer internship program. To date, we have our own [signal server](#) written in Java and a [prototype peer-to-peer network](#) that allows you to transfer audio / video. [Converters](#) for flv <-> webm / mkv are already coded to make the transition to WebRTC smoother, leaving Flash support for needy users.

At the moment, we are striving to move to WebRTC prototype with client-server architecture. This will result in better performance for webinars and web lectures. Having one powerful server or cluster with a high bandwidth can support a large number of listeners who have poorer throughput and computing resources.

We focus on WebRTC technology, since it is developed under the patronage of W3C web consortium, which means it will inevitably become the official standard of web-transfer of media content.

As soon as Flash technology will be left behind, OpenMeetings will make a qualitative leap, namely:

1. development of mobile clients will be simplified by an order of magnitude, since this will allow using only a browser without any additional technological requirements (Java, Flash);
2. stability will increase, because WebRTC has much less crashes than Flash;
3. new protocols used in the technology WebRTC, will speed up the transfer of video / audio.