## Platform/browser supportability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Web Standards** | **Firefox** | **Chrome** | **Safari** | **IE** | **Opera** |
| WebGL | 4 | 9 | 5.1 (Disabled) | No support | 11 |
| Canvas | 2 | 4 | 3.1 | 9 | 10 |
| WebSocket | 11 | 16 | Nightly | 10 | No support |
| **Web client support** | C:\Users\benxiao\Documents\Cisco\Demo\Checkmark.png | C:\Users\benxiao\Documents\Cisco\Demo\Checkmark.png | C:\Users\benxiao\Documents\Cisco\Demo\Checkmark.png(Nightly) | C:\Users\benxiao\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\M6T1XM4P\MC900432537[1].png | C:\Users\benxiao\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\M6T1XM4P\MC900432537[1].png |

## Other dependencies

* Broadway JS decoder
* Websocket support on Webex side

## Pros

* No download
* Codec implementation is within Javascript code, so we are not limited by browser codec support. More flexibility
* Ability to use socket based data transfer, unlike with <video>, which can only take HTTP requests.

## Cons

* Performance is not as fast as native code, and degrades with higher resolutions and large frame changes.
* JS code is open to the public
* Lack of widespread browser support for required web standards prevents easy cross-browser compatibility.
* Websocket is TCP based rather than UDP based, making it more unsuitable for video. The PeerConnection API addresses this, but it is not a mature standard yet.

## Current problems/difficulties

* Performance is slow for large changes in screen. Could be due to copying Javascript array buffers around.
* Screen corruption with 1920x1080 resolution. JS decoder might be too slow to keep up. We might need to drop frames.
* 1-2 second delay.

## Other comments

* Chrome has a MediaSource API that lets us transfer Javascript arraybuffers directly to the video element for decode. This could be a very viable solution to the performance issues with see with the Javascript decoder.