

# Comparing apples to oranges: A comparison of AudioWorklet polyfills

Christoph Guttandin  
Media Codings  
Berlin  
chrisguttandin@media-  
codings.com

## ABSTRACT

The specification of the Web Audio API is nearing the V1 milestone but there is still only one implementation of the AudioWorklet so far in Chromium based browsers. Firefox is expected to catch up soon but there is uncertainty whether or not the AudioWorklet will be implemented in Safari. For a foreseeable future Web Audio developers can't rely on the AudioWorklet to be available in all browsers.

But as many other new browser technologies the AudioWorklet can be used in browsers which do not support it natively by utilizing a polyfill as fallback strategy. Three polyfills are currently available which do all provide the same API as the native implementation but do of course have other performance characteristics. Every polyfill follows a slightly different approach.

This talk will incorporate a comparison of the native AudioWorklet and all available polyfills. They do for example differ in the way in which they emulate the AudioWorkletGlobalScope. One uses a Web Worker, the other uses an iframe and the third one is evaluating the AudioWorklet code on the main thread. This has not only implications on the

performance but can also be a security risk when running third party AudioWorklets from untrusted sources. On the other hand, executing the code on the main thread will save some time because it avoids passing around costly messages. In some scenarios it might be possible to achieve a lower latency using that technique. But this assumption falls short in case the main thread is very busy and can't invoke the process() function in a timely manner. In such a situation it is better to offload the work to an iframe or a Web Worker to execute the code in parallel.

For some web apps it might be possible to avoid a polyfill at all. If for example Web MIDI is a hard requirement it can be assumed that the AudioWorklet is available as well. For other web apps it could be beneficial to keep using the now deprecated ScriptProcessorNode even when the AudioWorklet is available. At least when running on underpowered low-end mobile devices this could make sense.

As with many other decisions choosing the right AudioWorklet polyfill is a decision depending on many factors. This talk will provide an opinionated checklist which is supposed to help to choose the most appropriate polyfill for any use case.



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** owner/author(s).

*Web Audio Conference WAC-2019, December 4–6, 2019, Trondheim, Norway.*

© 2019 Copyright held by the owner/author(s).