



**HAL**  
open science

## Web Audio Modules: Swiss knife for audio plugin developments on the Web Platform

Michel Buffa, Antoine Vidal-Mazuy, Quentin Plet

### ► To cite this version:

Michel Buffa, Antoine Vidal-Mazuy, Quentin Plet. Web Audio Modules: Swiss knife for audio plugin developments on the Web Platform. Internet of Sounds 2023, Oct 2023, Pise, Italy. 10.5281/zenodo.6769408 . hal-04335580

**HAL Id: hal-04335580**

**<https://inria.hal.science/hal-04335580>**

Submitted on 11 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Web Audio Modules: Swiss knife for audio plugin developments on the Web Platform

Michel Buffa, Antoine Vidal-Mazuy  
Université Côte d'Azur,  
(buffa, antoine-vidalmazuy)@univ-cotedazur.fr

Quentin Plet  
Université Jean Monnet,  
quentin.plet@etu.univ-st-etienne.fr

## ABSTRACT

We propose to present in this demo a large set of web audio plugins (aka web audio modules), real-time effects (most classic audio effects, tube guitar amplifier simulations, etc.), instruments (synthesizers, samplers,) along with an open source DAW (wam-studio) and a real time collaborative host called sequencer.party. Web Audio Modules (WAM) is a standard for developing Web Audio plugins and hosts first proposed in 2015, that recently had a major update in 2021 for its 2.0 version.

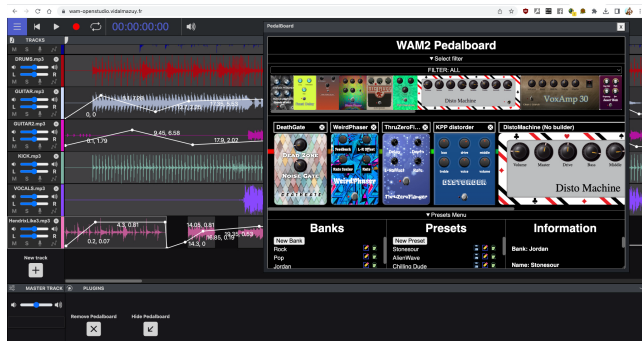


Figure 1: Wam-Studio, an open source DAW, written to showcase Web Audio Modules.

## 1. Web Audio, Web Midi, Web Assembly

Introduced in 2012 by the World Wide Web Consortium (W3C), the Web Audio API reached the "Recommendation" status in June 2021 (Official Standard). This JavaScript API allows the developer to manipulate a set of "Audio Nodes" that process or produce sound and that can be connected to form an "audio graph". Audio Nodes process the sound in the audio thread, and the algorithms used cannot be changed, only parameter changes are allowed. The recent addition (2018) of the AudioWorklet node provided a solution for implementing custom low-level audio processing running in the audio thread [5]. Furthermore, it became possible to cross compile existing C/C++/Rust/Domain Specific Language code to WebAssembly, a byte code representation supported by web browsers, facilitating the port of existing DSP code to the web platform.

Over the years, lots of high-level audio effects and instruments have been developed using a combination of such audio nodes. Also, the WebMidi API, recently supported by all Major browsers, mobile and desktop ones, brings MIDI support to the browser and combined with Web Audio it became possible to control

instruments using external devices (midi piano keyboard, control surfaces, pads etc.).

## 2. Web Audio Modules: VST for the Web

In 2015, Jari Kleimola and Olivier Larkin proposed the Web Audio Modules (WAM)[1], a standard for Web Audio plugins and DAWs. The 2.0 version of Web Audio Modules has been released in 2021 as a group effort by a large set of people [2] and since then, multiple plugins and hosts have been published, mostly as open source and free software. WAM 2.0 comes with a SDK, an abstract API, several open source repositories with dozen of plugins, tutorials, and two hosts for demonstrating the capabilities of WAMs. When we designed WAM 2.0 we wanted to support as many development workflow as possible, from the simple web developer who code on plain JavaScript, the React developer used to build his applications, to the C/C++ developer who is used to pro DSP development and who will cross compile his existing code to WebAssembly and reuse a minimal set of features from the WAM SDK.



Figure 2: sequencer.party: a collaborative online host. All you see are WAM plugins.

Another concern was performance. The WAM framework uses the original design for handling the communication between plugins and host applications that do not rely on the low-level parameter management provided by the Web Audio API. The main reason for that is to allow high performances in the case where both a DAW and plugins are implemented as AudioWorklets. At the time the WebAudio API has been designed, AudioWorklets did not exist and some use cases could not be taken into account. Indeed, if the DAW is built using AudioWorklet nodes for processing audio, then some parts of the code run in the high priority / audio thread. Then, if a WAM plugin is associated with a given track in a DAW project (Figure 1), and if the plugin is itself built using an AudioWorklet

node, it also has custom code running in the audio thread. The WAM framework has been designed to handle this particular case and will enable DAW/plugins communication without crossing the audio thread barrier. Let us take one example: while playing, a MIDI track sends notes to a virtual instrument plugin, and changes some of the parameters of this plugin at the frequency rate. Remember that a DAW can have multiple tracks associated to dozen of plugins, and each plugin can have dozens of parameters. The WAM framework detects this case, and will seamlessly use Shared Array Buffers and a ring buffer, without crossing the audio thread barrier. No need to send events from the control/GUI thread, that would have been mandatory if the Web Audio API audio node parameter management was used. To sum up, the WAM framework streamlines the creation of plugins and host applications and enables highly efficient communication between hosts and plugins. A video of WAMs in action is available online<sup>1</sup>.



Figure 3: some WAM plugins developed by the community.



Figure 4: a WAM sampler that can be MIDI controlled, used freesounds.org for finding samples.

Today, a WAM community Github repository proposes dozen of plugins (Figure 3: effects, instruments, utility plugins), an open source DAW called wam-studio<sup>2</sup> (Figure 1) is being proposed as a demonstrator for WAM plugins [4], and a real-time collaborative host called sequencer.party<sup>3</sup> is a great showcase for the WAM standard (Figure 2). We propose to demonstrate the WAM ecosystem at the conference, using an electric guitar [3], microphones, midi controllers, pads for controlling a WAM sampler (Figure 4) etc.

### 3. References

- [1] Kleimola, J. and Larkin, O. "WebAudioModules". In *Proceedings of the Sound and Music Computing Conference*. 2015.
- [2] Buffa, M., Ren, S., Campbell, O. and al. "Web Audio Modules 2.0, an open Web Audio plugin standard", In *Companion Proceedings (Developer's track) of the The Web Conference 2022 (WWW '22, ranked A\*)*, April 2022, Lyon, France. Accepted, to be presented.
- [3] J. Lebrun, M. Buffa, and J. Sintes, "From blues to rock to jazz: three different Web Audio tube guitar amplifier simulator plugins," in *Proc. 7th Web Audio Conference (WAC 2022)*, (Cannes, France) (2022 Jul.). <https://dx.doi.org/10.5281/zenodo.6769408>
- [4] Buffa, M. and Vidal-Mazuy, A., 2023, April. WAM-studio, a Digital Audio Workstation (DAW) for the Web. In *Companion Proceedings of the ACM Web Conference 2023* (pp. 543-548).
- [5] Ren, S., Letz, S., Orlarey, Y., Michon, R., Fober, D., Buffa, M. and Lebrun, J. Using Faust DSL to develop custom, sample accurate DSP code and audio plugins for the Web browser. *J. Audio Eng. Soc.*, 68(10), (2020) pp.703-716. (hal-03087763)

<sup>1</sup> [https://www.youtube.com/watch?v=w7a\\_Kbx7nA8](https://www.youtube.com/watch?v=w7a_Kbx7nA8)

<sup>2</sup> <https://wam-studio.i3s.univ-cotedazur.fr/>

<sup>3</sup> <https://sequencer.party/>