# Host and Plugins : Parameter Automation Without Crossing the Audio Thread Barrier

Dylan Batisse, Antoine Vidal-Mazui, Jean-Philippe Carlens, Antoine Cousson, Michel Buffa

▶ **To cite this version:**

Dylan Batisse, Antoine Vidal-Mazui, Jean-Philippe Carlens, Antoine Cousson, Michel Buffa. Host and Plugins : Parameter Automation Without Crossing the Audio Thread Barrier. WAC 2022 - Web Audio Conference 2022, Jul 2022, Cannes, France. hal-03812916

**HAL Id: hal-03812916**
**https://inria.hal.science/hal-03812916v1**

Submitted on 13 Oct 2022

**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.

# Host and Plugins : Parameter Automation Without Crossing the Audio Thread Barrier

Dylann Batisse, Antoine Cousson, Antoine Vidal Mazuy, Jean-Philippe Carlens, Michel Buffa
Université Côte d'Azur
(firstName.lastName)@univ-cotedazur.fr

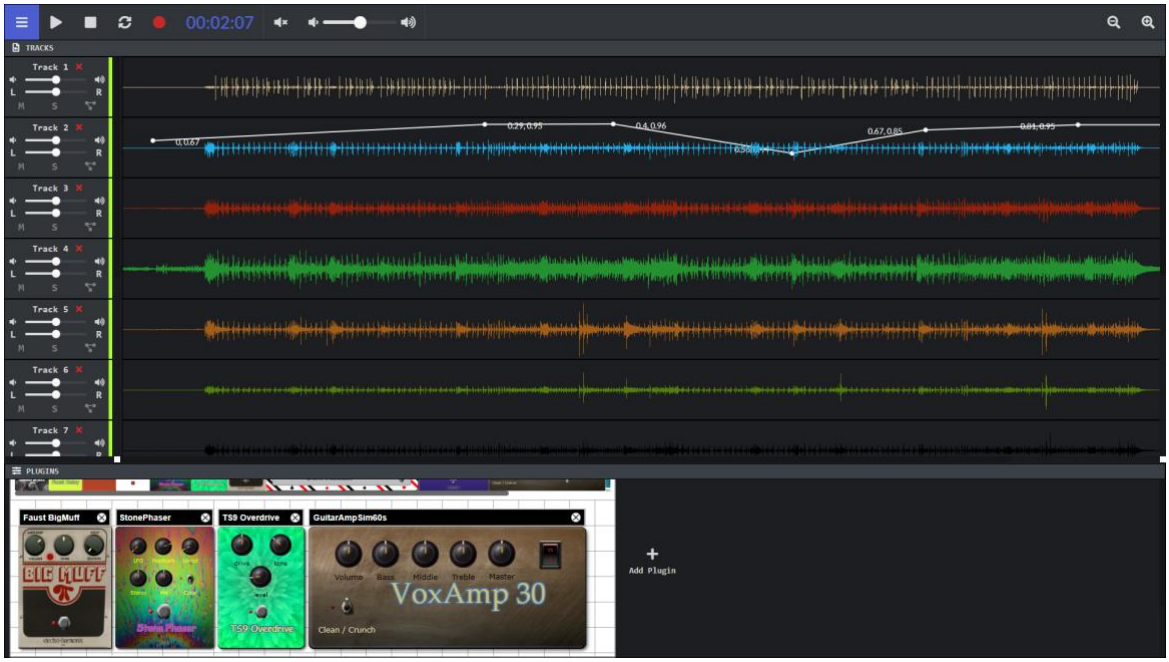


**Figure 1: A multi-track DAW prototype with a pedalboard WAM plugin in the plugin chain associated to the selected track.**

## ABSTRACT

WebAudio Modules 2.0 (aka WAM) is a standard for developing interoperable host and plugins that can be written using different development approaches (JavaScript, build systems, TypeScript, FAUST, Csound, front-end frameworks, etc.). The WAM standard has been designed with high performance host-plugin communication in mind when both the host and plugins are running in the audio thread. This is the case when the host is written as an Audio Worklet (AW) and when plugins are also AW based. We have developed for the WAM standard an example of such a host: a very simple DAW using Audio Worklets to manage the audio buffers for each track. In a typical way, each track is associated to a chain of plugins, to bring for example real time effects on the signal. This DAW is an interesting example to study for developers who would like to write this type of program: not only does it show how to manage audio buffers with sample accuracy (in order to have very precise loops), but it also shows how, thanks to the tools offered by the WAM standard, you can automate a large number of plugin parameters at the sample rate in a very efficient way (using Shared Array Buffers and Ring Buffers), much more than what is possible with the management of the parameters offered by the WebAudio API.

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

*Web Audio Conference WAC-2022, July 6–8, 2022, Cannes, France.*

© 2022 Copyright held by the owner/author(s).

## 1. Introduction

In 2015 Jari Kleimola and Oliver Larkin created Web Audio Modules (WAM) [2], a standard for creating reusable and interoperable plugins implemented as WebComponents, the core of which was written in C/C++.

This initiative was aimed primarily at developers of native plugins (i.e. VST plugins). Since then, this standard has evolved in a new "Web Audio Modules 2.0" version [1], taking into account feedback from developers of the first version and the evolution of web standards (with WebAssembly, WebComponents, WebMidi etc.). The WAM distribution is open source and distributed as four

GitHub repositories and as npm modules (MIT License)[1]. One of these repository, the wam-example github repository, contains different plugins and hosts that cover a variety of programming languages and approaches. WAM v2.0 creators asked us to develop some host examples written as AudioWorklets (AW), with and without the use of WebAssembly modules, to show some features of the standard, in particular how WAM events can be used to automate a large amount of plugin parameters *without leaving the audio thread* (this is the case when both the host and plugins are written as AW running in the audio thread).

## 2. WebAudio API limitations for parameter handling

Since the WAM standard aims to allow maximum flexibility for developers, WAM authors have decided that WebAudio API's AudioParams should not appear explicitly in the WAM API. Some WAM design patterns do not make use of any stock WebAudio API nodes, and while it is possible to define custom parameters for AudioWorklet, in many cases it would be too heavy / cumbersome to expose the potentially hundreds of parameters residing in WebAssembly code via that API. Furthermore, the parts of the WebAudio API having to do with parameters were conceived before developers had any direct access to the audio thread, forcing parameter updates to be scheduled asynchronously from the main thread far enough in advance to account for crossing the thread barrier. This aspect of the WebAudio API is not compatible with the goal to support synchronous, 'just in time' interaction between hosts and plugins on the audio thread as in native plugin environments. To get around these limitations the WAM standard has its own WamParameter API to handle parameter updates, designed to facilitate many different WAM design patterns and interaction between hosts and plugins on either thread.

## 3. Plugin parameters and host-plugin communication

Sample-accurate event scheduling is a critical requirement for professional audio applications. The WebAudio API does allow for sample-accurate scheduling of AudioParams, and this capability is leveraged in the WAM SDK to facilitate WAMs that incorporate one or more stock Web Audio nodes. However, as mentioned before the automation API for AudioParams is not useful for many WAM design patterns, and we wanted to show how WAM supports sample-accurate processing of other kinds of events such as MIDI messages or parameter automation messages. The WAM API facilitates event scheduling through a unified interface *that is mirrored on both the main thread and audio thread*. Hosts operating entirely on the main thread will still be required to schedule events with some lookahead to ensure that they are processed at the intended time, as these messages must still cross the thread barrier. However, *hosts with a presence on the audio thread can schedule events at the beginning of the rendering block in which the events should occur*. This is the case of the DAW we will demo.

## 4. Related works

At the time of writing (April 2022) the most advanced DAWS available on the Web platform are commercial. The oldest one, oundtrap.com belongs to Spotify, and is mainly based on high level WebAudio API nodes, and is not open to third-party plugins. The same code base seems to be used for a mobile version. Bandlab.com belongs to an Asian company that sells music gear on the Web, and is also available on mobile devices. The technology used is not public, but it seems that the mobile version and the Web version do not share the same code base. AmpedStudio.com uses AudioWorklet and C++ cross compiled plugins a lot, and does not have yet a specific version for mobile devices…It supports different sorts of plugins, including the WAM standard. We have been working with a developer's version of AmpedStudio and managed to load successfully some WAM 2.0 plugins in it. All these DAWs support audio and MIDI tracks, plugin chains for instruments and effects, automation, and some of them are collaborative.

Sequencer.party is a mix between Ableton Live and Google Docs, it's a sequencer 100% collaborative. The sequencer itself is a WAM host and most modules are WAM2 plugins, also available on GitHub.

Gridsound is a more traditional open source DAW, that has been developed since 2015. It supports audio and MIDI, but does not yet supports an interoperable plugin format. It is based on "components" whose format is specific to Gridsound.

There are also popular JS librairies for developing a multitrack player/recorder, such as wavesurfer.js[2] or waveform-playlist[3], and some audacity like audio buffer editors such as AudioMass[4].

None of the open source DAWs have their core audio player based on AudioWorklet, and only AmpedStudio.com takes care about the audio thread isolation optimization for host-plugins communications. This is why our prototype can be a good way to see how to implement such features.

## 5. Settings for the demo

Our host can record/play/edit different tracks with a customizable plugin chain associated to each track. We propose to show some audio projects we prepared, that use a large number of plugins and parameters with automation. During the demo, attendees will also be able create projects from scratch and / or edit existing projects. We also propose to show some source code extracts and explain how this DAW uses the features of the WAM2 standard for host-plugins interaction (discovering/managing plugins and automation curves, etc.).

## 6. REFERENCES


[1] M. Buffa, S. Ren, O. Campbell, T. Burns, S. Yi, J. Kleimola, O. Larkin, "Web Audio Modules 2.0, an open Web Audio plugin standard", In Companion Proceedings (Developer's track) of the The Web Conference 2022, April 2022, Lyon, France.

[2] J. Kleimola and O. Larkin. Web audio modules. *12th Sound and Music Computing Conference (SMC15). Maynooth, Ireland.*


[1] https://github.com/webaudiomodules and https://www.npmjs.com/settings/webaudiomodules/packages

[2] https://wavesurfer-js.org/

[3] https://github.com/naomiaro/waveform-playlist

[4] https://github.com/pkalogiros/AudioMass