



HAL
open science

High-performance Web-Audio applications, the challenges of writing a DAW

Michel Buffa, Antoine Vidal-Mazuy

► **To cite this version:**

Michel Buffa, Antoine Vidal-Mazuy. High-performance Web-Audio applications, the challenges of writing a DAW. Réunion de lancement du projet ANR DOTS, IRCAM, Oct 2023, paris, France. hal-04337523

HAL Id: hal-04337523

<https://inria.hal.science/hal-04337523>

Submitted on 12 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

Applications Web-Audio hautes performances, challenges à relever pour écrire un DAW

Journée ANR DOTS

13 Octobre 2023

(Lien de la présentation :

<https://tinyurl.com/4yj2ppkt>)



Michel Buffa, Antoine Vidal-Mazuy

Université Côte d'Azur / Laboratoire I3S / INRIA, WIMMICS group,

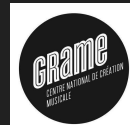
michel.buffa@univ-cotedazur.fr, @micbuffa



WAMs



UNIVERSITÉ
CÔTE D'AZUR



53JS

Qui suis-je ?

- Professeur à l'Université Côte d'Azur (UCA), France
 - Membre de l'équipe WIMMICS commune à INRIA et au laboratoire I3S du CNRS
 - **W3C Advisory Committee Representative** pour UCA
 - **Membre du W3C WebAudio Working Group depuis 2014**
 - michel.buffa@univ-cotedazur.fr, @micbuffa
- Membre du groupe qui a créé les Web Audio Modules aka **“Plugins audio pour la plateforme Web”**
 - Sorte de “VST pour le Web”



WebAudio, WebMidi, etc.

Concepts de l'API WebAudio

Opérations audio dans un **Audio Context**

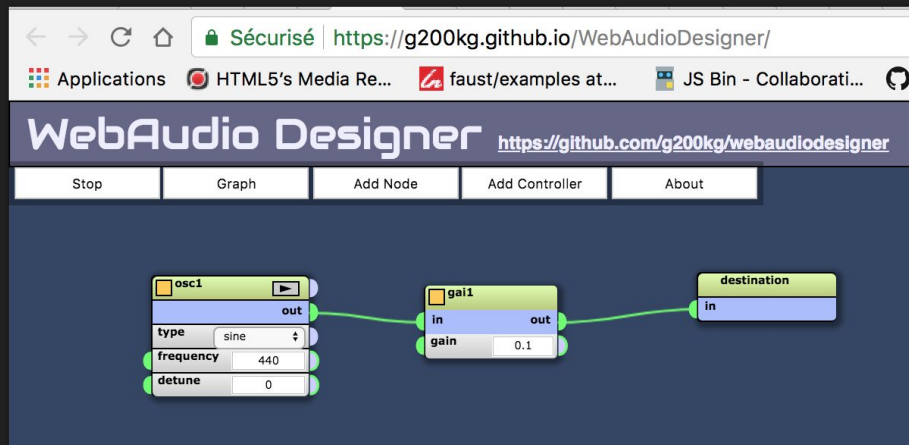
```
let ctx = new AudioContext();
```

Conception modulaire :

- Opérations audio : **audio nodes**
...qui forment un **graphe audio**

```
let osc1 = ctx.createOscillator();  
osc1.frequency.value = 440;  
let gain1 = ctx.createGain();  
gain1.gain.value = 0.1;
```

```
osc1.connect(gain1).connect(ctx.destination);
```



Architecture interne

- **Un thread de rendu** appelé "**audio thread**" est seul en charge du rendu sonore du graphe audio
 - Contraintes temps réel fortes : thread de haute priorité. Si il ne parvient pas à fournir le prochain bloc d'échantillons sonores à temps : glitches, dropouts etc.
- **Un thread de contrôle**, exécutant en général du code JavaScript code pour la GUI, fait les appels à l'API WebAudio, gère la création et modification du graphe audio.
 - connexion/déconnexion des nœuds, modifications de leurs paramètres.
- **Les nœuds "standards" traitent le son dans l'audio thread**
 - Les algorithmes DSP sont fixes, seuls les paramètres peuvent être modifiés.
- **Le nœud AudioWorklet (2018) fournit une solution pour implémenter des traitements DSP personnalisés dans l'audio thread.**
 - **AudioWorkletNode** code: main thread
 - **AudioWorkletProcessor** code: audio thread (avec contraintes/limitations)

WebAudioModules (WAM) audio plugins pour le Web

WAMs dans des logiciels hôtes

<https://ampedstudio.com>



<https://sequencer.party/>

mainline.i3s.unice.fr/wam2/packages/_/

Applications Midjourney Show... An evil monkey in... MHKG program an... (8) Comment utili... Formalités admin... maquette MIAGE [...]

Example WAM Host

NEW: See WAM 2.0 API docs
NEW: See WAM 2.0 Parameter Manager docs

0:00 / 2:56

Select MIDI input device: Select...
Select live input device: Par défaut - MacBook Pro Microphone (Built-in)
Live input: NOT ACTIVATED, click to toggle on/off

Enter any WAM Plugin URL
Jobx/index.js

LOAD PLUGIN

Midi

- Simple MIDI Keyboard
- Midi Virtual keyboard No Sound (just emits events)
- Random note generator
- Simple Transport
- MIDI Sequencer

Pure JavaScript

- Quadrafuzz
- Quadrafuzz without builder
- Disto Machine without builder
- PimpFongDelay
 - Graphic Equalizer (pureJS)
 - WAM Example (pureJS)
- TimSyynth, a GM synth without GUI. Accepts program changes, volume change, notes...
- Wam Event Viewer
- WebMIDI Output

Fast

- FastPimpFongDelay
- FastPimpFongDelayDefaultUI
- Guitar AmpSim 60s (generated by fast IDE)
- Stone Phaser (generated by fast IDE)
- TSP Overdrive (generated by fast IDE)
- BigMuff Fuzz (generated by fast IDE)
- Fast Flute MIDI

Typescript

- LiveGain
- Oscilloscope
- Spectroscope
- Spectrogram

CsSound

- CsSound PitchShifter

A screenshot of a virtual synthesizer interface. It features a piano keyboard at the top, a 'Program Change' slider, and a 'Volume' knob. Below the keyboard is a complex control panel with various sections: 'MASTER' (Volume, Tune, Transpose), 'GLOBAL' (Spread, Unison, Glide), 'OSCILLATORS' (OSC1, OSC2, OSC3), 'MIX' (Mix, Filter, Amp), 'FILTER' (Cutoff, Resonance, Env), 'FILTER ENVELOPE' (A, D, S, R), 'AMPLIFIER ENVELOPE' (A, D, S, R), 'MODULATION' (LFO, Sync, Mod, Trig), 'VOICE VARIATION' (Pitch, Filter, Env), and 'VOICE PAN' (L, C, R). The interface is light blue and black.

https://mainline.i3s.unice.fr/wam2/packages/_/

```
SAVE STATE RESTORE STATE
+ instance.descriptor : {"name":"OBXD","vendor":"Jari Kleimola 2017-2020 (jari@webaudiomodules.org)","description":"","version":"1.0.0","apiVersion":1,"isInstrument":true,"website":"","hasAudioInput":true,"hasAudioOutput":true}
+ gui.properties.dataWidth.value : undefined, (you should define get properties in Gui.js)
+ gui.properties.dataHeight.value : undefined, (you should define get properties in Gui.js)
+ instance.audioNode.getParameterInfo() :
+ 0 :
```

DAWs commerciaux

File Edit Settings Tutorials Save Song

Show Instrument

Drums - Mambo

Piano - Moon [Em]

Volume

+ Automation

Soundtrap.com (spotify)

File Edit View Settings Help

Nuzzle Puzzle - Classic Rap Type Beat in Eb Minor - 90BPM

00:00.0 E^b min 90 bpm 4/4

Main Voc

Keys (L)

Keys (L) Copy

Synth Line

Bass Drum

Bandlab.com

0:00:00.00 180 bpm 4/4

Bass

Chords

Drums

Lead

Percussion

VIRTUAL KEYBOARD

OCTAVE

DEVICE CHAIN

Ampedstudio.com

Soundation - Feed

https://chrome.soundation.com

File Edit View Settings Soundation Help

SPC

Sample

SAM-1

Soundation.com

DAWs: le photoshop de l'audio

Enregistrer, éditer,
mixer des pistes
audio et MIDI

Pistes associées à
des plug-ins (effets
audio, instruments)

Automation des
paramètres

Gérer le temps-réel
et la latence!



WAM-Studio: un DAW **open source** compatible WAM

!

Motivation:

- Démonstrateur des APIs du W3C
- Pas de DAW web open source complet disponible
- Développer un DAW web: problèmes spécifiques lié à l'environnement sandboxé



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

Challenge 1: concevoir un framework host/plugin Web

Groupe de travail sur les plugins WebAudio

Journées d'Informatique Musicale **2023**

(lien présentation: <https://tinyurl.com/5n62b6hn>)

Membres : Michel Buffa et Jerome Lebrun (Université Côte d'Azur / Laboratoire I3S),
Stéphane Letz (GRAME), Laurent Pottier et Shihong Ren (Université de Saint-Etienne
/ Laboratoire ECLLA)

Collaborateurs externes: Steven Yi (stevenyi@gmail.com), Owen Campbell
(owen.campbell@gmail.com), Jari Kleimola (jari.kleimola@webaudiomodules.org),
Hugo Mallet (hugo.mallet@53js.fr), Tom Burns (tom@burns.ca), Quentin Plet
(quentin.plet@etu.univ-st-etienne.fr), Antoine Vidal-Mazuy
(antoine.vidal-mazuy@etu.univ-cotedazur.fr)



Web Audio Modules 2.0 An Open Standard for WebAudio Plugins

The Web Conference 2022
Web Developer and W3C track

April 26, 2022

(presentation link: <https://tinyurl.com/2p8jvu44>)

Michel Buffa (Université Côte d'Azur / Laboratoire I3S / INRIA, WIMMICS group),
Shihong Ren (Université de Saint-Etienne / Laboratoire ECLLA),
Owen Campbell (owen.campbell@gmail.com)

Tom Burns (tom@burns.ca), Steven Yi (stevenyi@gmail.com),
Jari Kleimola (jari.kleimola@webaudiomodules.org),



ADC²²

WEBAUDIO MODULES 2.0:

AUDIO PLUGINS FOR THE WEB PLATFORM!

MICHEL BUFFA



Challenge 2 : host/plugin communication (i.e 44.1Khz)

Contrôle bas-niveau de la lecture nécessaire.

Chaque piste est un AudioWorklet
(WAMNode+WAMProcessor)

- Le processeur d'une piste communique avec les plugins à la fréquence d'échantillonnage (WAM SDK) sans quitter le thread audio !
 - Automation des paramètres : [DEMO](#)
 - Envoyer des notes MIDI : [DEMO](#)

The image shows the OBXO virtual analog synthesizer interface, which is a complex panel of controls. It includes sections for MASTER (VOLUME, TUNE, TRANSPOSE), GLOBAL (SPREAD, LINSON, GLEDE), CONTROL (KEEP ALL, LEGATO, VOICES), OSCILLATORS (OSC 1, OSC 2), MIX, FILTER, FILTER ENVELOPE, AMPLIFIER ENVELOPE, MODULATION, VOICE VARIATION, and VOICE PAN. Below the interface is a MIDI piano roll showing a sequence of notes on a piano keyboard. The notes are represented by green bars on a grid, with the C7 and C6 keys labeled on the left. The tempo is set to 120 bpm, and the MIDI input device is set to 'Select...'. The piano roll shows a sequence of notes starting with a C7 chord, followed by a sequence of notes that move down the scale, ending with a C6 chord.

The image shows a DAW interface with two audio tracks. The top track is labeled 'GUITAR.mp3' and has a volume knob set to 100%. The bottom track is labeled 'GUITAR2.mp3' and has a volume knob set to 100%. The tracks are stereo and merged L/R. The waveform for the top track shows a signal with a peak at 8.65, 0.89 and a trough at 20.15, 0.47. The waveform for the bottom track shows a signal with a peak at 0.35, 0.11.



No need to schedule ahead in time like in the famous article "[A tale of two clocks](#)" by C.Wilson.

Challenge 3 : gérer la latence

Latence = RETARD !

1 - Latency Round-trip : la durée qu'il faut pour qu'un signal, tel qu'une voix chantée ou un solo de guitare, passe d'une entrée analogique sur une interface audio, à travers les convertisseurs analogique-numérique, jusqu'au DAW, puis retourne ressorte par l'interface audio. Si cette durée est élevée, cela impacte négativement la capacité du musicien à jouer en rythme, par exemple. Elle devrait être < 20 ms.

2 - Latence de sortie : temps mis à un son émis par le DAW pour atteindre la sortie audio

3 - latence d'entrée : temps écoulé avant que l'enregistrement ne démarre dans le DAW lorsque l'on joue d'un instrument ou que l'on chante dans un microphone. Doit être prise en compte lors de l'enregistrement !

Latence d'entrée = latence round-trip - latence de sortie

Challenge 3: compenser la latence d'entrée

1. Mesurer la latence round-trip :

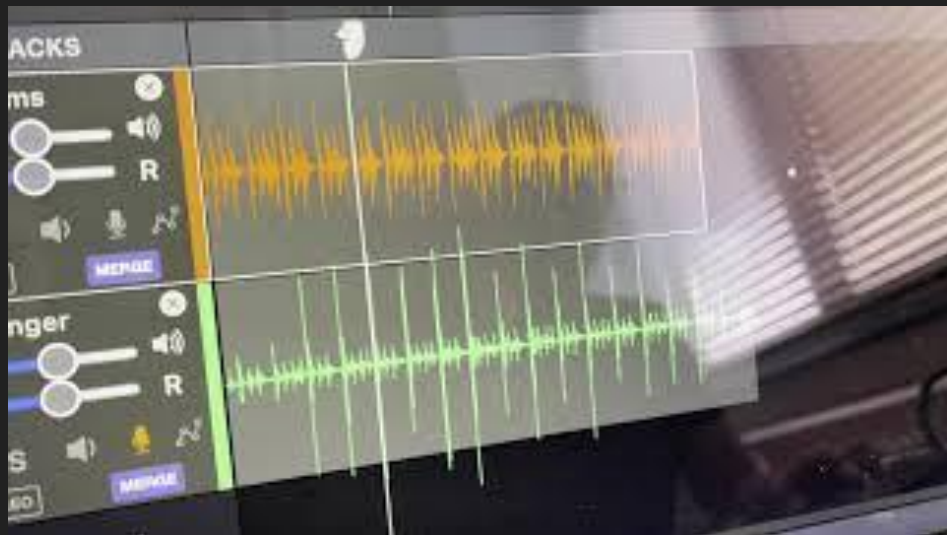
- Solution matérielle** (micros en entrée et en sortie, enregistreur numérique),
- Solution logicielle** : enregistre un son du DAW, l'enregistre et mesure le temps écoulé (vidéo ci-dessous)

2. La latence de sortie est fournie par la WebAudio API,

3. Calculer la latence d'entrée,

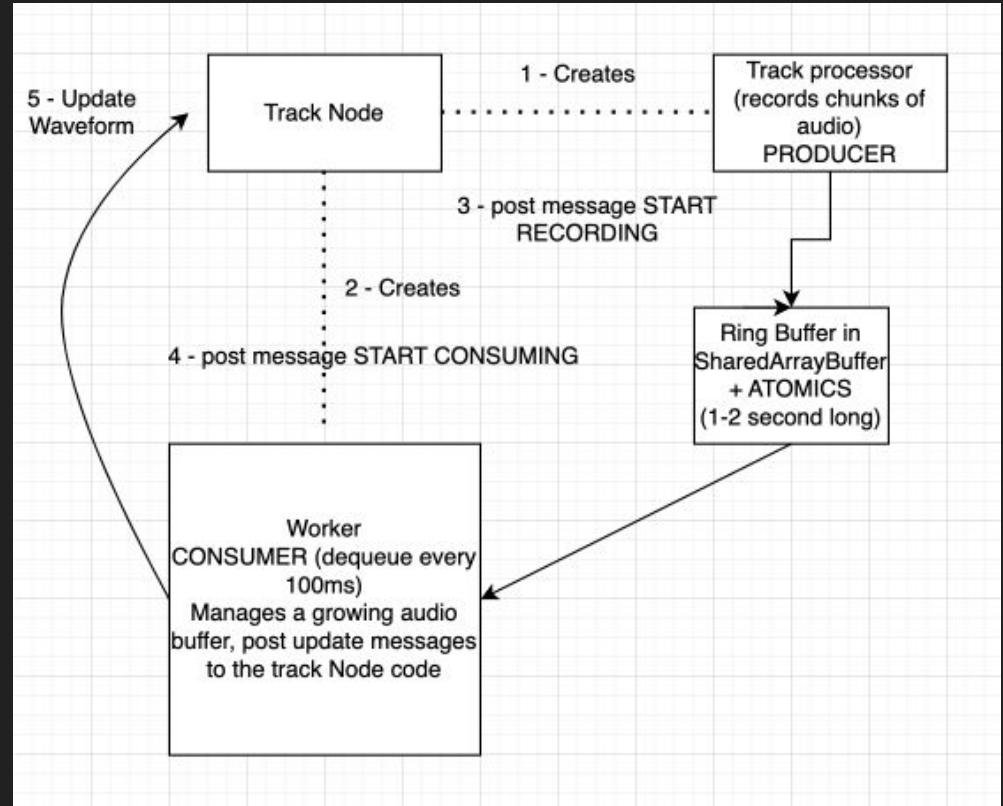
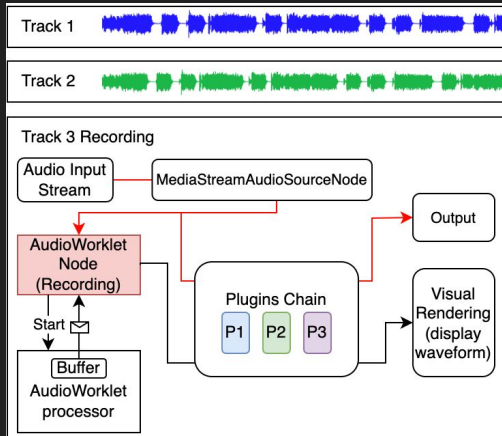
$\text{Entrée} = \text{Round-trip} - \text{Sortie}$

4. Décaler en arrière dans le temps les buffers audio enregistrés.

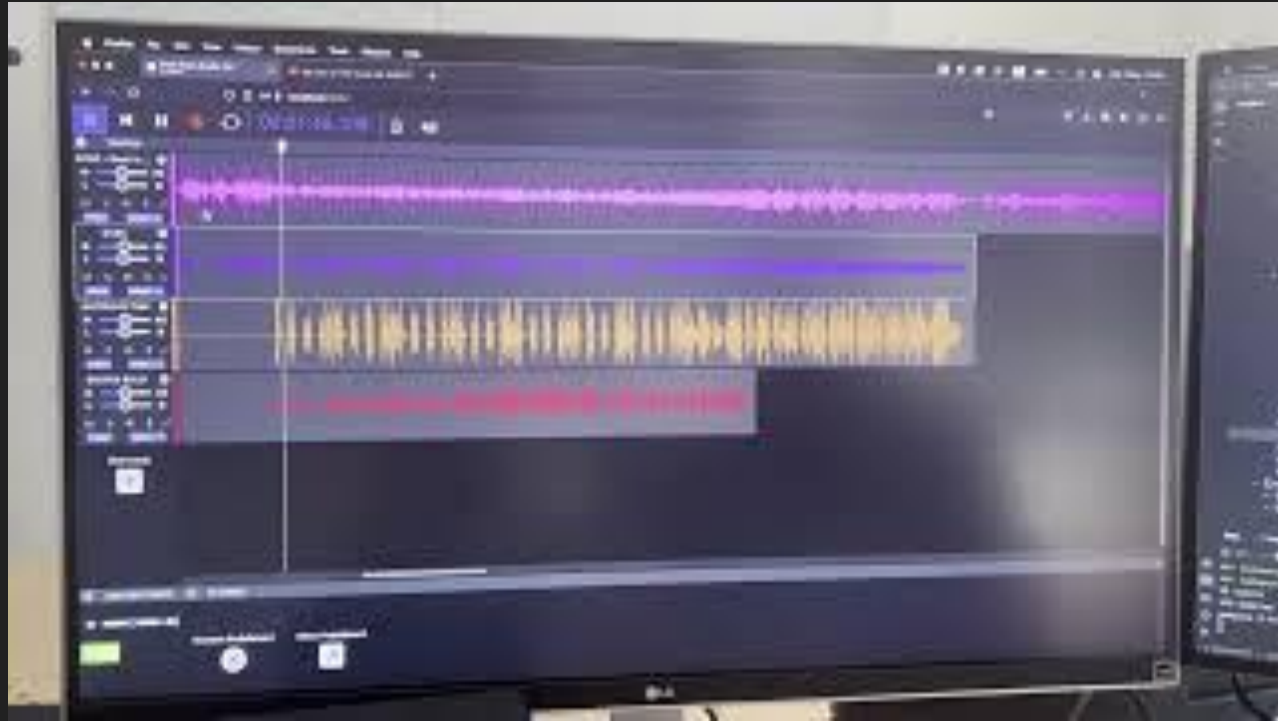


SOLUTION INTÉGRÉE DANS LE DAW !

Challenge 4: enregistrer plusieurs pistes en même temps de manière robuste, pendant la lecture d'autres pistes...



A small projet from recording to mixing



CONCLUSION + WAM-Studio disponibilité

WAM-Studio est en développement mais fonctionnel et stable

- Bonne démonstration des technologies du W3C pour développeurs audio
- La conception et le code montrent des bonnes pratiques pour enregistrer et gérer la latence dans un environnement sandboxé.

WAM-Studio a des mises à jour hebdomadaires...

- Code source et image Docker [disponibles sur GitHub](#) (front-end et back-end sont dans le même projet, image docker fournie, prêt à déployer)
- [Démo en ligne](#)
- Les parties principales de WAM-studio font l'objet de [tutoriaux en ligne](#).
- Tous les plugins présents dans WAM-Studio sont open-source et publiés sur le dépôt GitHub [wam-community](#). Ils sont aussi disponibles dans d'autres hôtes (sequencer.party par ex.), et via [URLs prêts à l'emploi](#) demo [ici](#)!
- WAM-Studio plugin manager aussi disponible séparément ([source](#) and [demo](#))

WAM-Studio travail en cours...

Fini

- Sauvegarde des projets dans le cloud,
- Viewports sur les pistes (zoom, support pour pistes de longues durées)
- Macros par type de piste (presets voix/guitare etc.)
- Grilles de mesures + tempo et key signatures
- (presque fini) : split/merge, normalize, reverse sur pistes audio

A venir (S4 2023)

- CTRL-Z / undo
- GUI pour le routage audio (support de cartes sons > 4 entrées/sorties)
- Pistes MIDI (déjà fonctionnelles en dehors du DAW)
- Smart cache (PouchDB/IndexedDB) pour sauvegarde plus rapide des projets

Plus tard (2024)

- Support des [WebCodecs](#)
- Fonctionnalités collaboratives